

**Oracle9i**

レプリケーション・マネージメント API リファレンス

リリース 2 (9.2)

2002 年 7 月

部品番号 : J06293-01

**ORACLE®**

---

Oracle9i レプリケーション・マネージメント API リファレンス, リリース 2 (9.2)

部品番号 : J06293-01

原本名 : Oracle9i Replication Management API Reference, Release 2 (9.2)

原本部品番号 : A96569-01 (Vol.1)、A96570-01 (Vol.2)

原本著者 : Ted Burroughs

原本協力者 : Randy Urbano, Valarie Moore, N. Arora, S. Balaraman, Y. Chan, A. Demers, A. Downing, C. Elsbernd, Y. Feng, J. Galagali, D. Goddard, L. Kaplan, V. Krishnamurthy, A. Lakshminath, P. Lane, J. Liu, E. Lu, P. McElroy, M. Pratt, A. Rajaram, N. Shodhan, W. Smith, J. Stamos, J. Stern, M. Subramaniam, E. Vandeveld, L. Wong, D. Zhang

Copyright © 1996, 2002 Oracle Corporation. All rights reserved.

Printed in Japan.

制限付権利の説明

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

当プログラムのリバース・エンジニアリング等は禁止されております。

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に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その他の会社名および製品名は、あくまでその製品および会社を識別する目的にのみ使用されており、それぞれの所有者の商標または登録商標です。

---

# 目次

はじめに .....	xvii
------------	------

## 第 I 部 レプリケーション環境の構成

### 1 レプリケーションの概要

レプリケーション環境構築の概要 .....	1-2
作業を開始する前に .....	1-4

### 2 レプリケーション・サイトの構築

レプリケーション・サイト設定の概要 .....	2-2
マスター・サイトの設定 .....	2-4
orc1.world の設定 .....	2-4
orc2.world の設定 .....	2-8
orc3.world の設定 .....	2-12
マスター・サイト間のスケジュール・リンクの作成 .....	2-16
マテリアライズド・ビュー・サイトの設定 .....	2-20
mv1.world の設定 .....	2-21
mv2.world の設定 .....	2-27

### 3 マスター・グループの作成

マスター・グループ作成の概要 .....	3-2
作業を開始する前に .....	3-3
マスター・グループの作成 .....	3-5

## 4 配置テンプレートの作成

Oracle 配置テンプレートの概念 .....	4-2
配置テンプレートを作成する前に .....	4-3
配置テンプレートの作成 .....	4-3
配置テンプレートをパッケージ化してインスタンス化 .....	4-13
配置テンプレートのパッケージ化 .....	4-14
インスタンス化スクリプトのファイルへの保存 .....	4-16
インスタンス化ファイルの配布 .....	4-18
配置テンプレートのインスタンス化 .....	4-18
インスタンス化後のリフレッシュ・グループのリフレッシュ .....	4-21

## 5 マテリアライズド・ビュー・グループの作成

マテリアライズド・ビュー・グループ作成の概要 .....	5-2
マテリアライズド・ビュー・グループの作成 .....	5-4
mv1.world にマテリアライズド・ビュー・グループを作成 .....	5-4
mv2.world にマテリアライズド・ビュー・グループを作成 .....	5-12

## 6 競合解消の構成

競合解消の準備 .....	6-2
更新の競合の競合解消メソッドの作成 .....	6-3
上書きおよび廃棄による競合解消メソッド .....	6-3
最小および最大の競合解消メソッド .....	6-6
タイムスタンプによる競合解消メソッド .....	6-9
加算による競合解消メソッドと平均による競合解消メソッド .....	6-14
優先グループによる競合解消メソッド .....	6-17
サイトの優先順位による競合解消メソッド .....	6-21
一意性競合の競合解消メソッドの作成 .....	6-26
削除の競合防止メソッドの作成 .....	6-32
動的所有権の競合防止メソッド .....	6-37
ワークフロー .....	6-37
トークン渡し .....	6-38
行の所有者の検索 .....	6-40
所有権の取得 .....	6-40
変更の適用 .....	6-41

競合解消の監査 .....	6-42
競合解消統計の収集 .....	6-42
競合解消統計の表示 .....	6-42
競合解消統計の取消し .....	6-42
統計情報の消去 .....	6-43

## 第 II 部 レプリケーション環境の管理と監視

### 7 マスター・レプリケーション環境の管理

マスター定義サイトの変更 .....	7-2
オプション 1: すべてのマスター・サイトが利用可能 .....	7-2
オプション 2: 旧マスター定義サイトが利用不可 .....	7-3
新しいマスター・サイトの追加 .....	7-4
マスター・グループを静止することなく、新しいマスター・サイトを追加 .....	7-4
静止中のマスター・グループに新しいマスター・サイトを追加 .....	7-30
マスター・グループからマスター・サイトの削除 .....	7-40
使用不可能なマスター・サイトの削除 .....	7-42
データ・ディクショナリ・ビュー内のコメント・フィールドの更新 .....	7-43
プロシージャ・レプリケーションの使用 .....	7-45
プロシージャ・レプリケーションの制限事項 .....	7-45
ユーザー定義型とプロシージャ・レプリケーション .....	7-47
トランザクションのシリアル化 .....	7-48
レプリケート・プロシージャのサポートの生成 .....	7-48

### 8 マテリアライズド・ビュー・レプリケーション環境の管理

マテリアライズド・ビューのリフレッシュ .....	8-2
マテリアライズド・ビュー・グループのマスター・サイトの変更 .....	8-2
マテリアライズド・ビュー・グループおよびオブジェクトの削除 .....	8-3
配置テンプレートで作成されたマテリアライズド・ビュー・グループの削除 .....	8-3
手動で作成されたマテリアライズド・ビュー・グループまたはオブジェクトの削除 .....	8-9
マスター・サイトまたはマスター・マテリアライズド・ビュー・サイトのクリーン・アップ .....	8-11
マテリアライズド・ビュー・ログの管理 .....	8-16
マテリアライズド・ビュー・ログの変更 .....	8-16
マテリアライズド・ビュー・ログ領域の管理 .....	8-17

マテリアライズド・ビュー・ログが含まれるマスター表の再編成 .....	8-20
マテリアライズド・ビュー・ログの削除 .....	8-24
エクスポート/インポートによるマテリアライズド・ビュー・サイトのオフライン・ インスタンス化 .....	8-25
マテリアライズド・ビュー・グループに対するグループ所有者の使用 .....	8-36

## 9 レプリケーション・オブジェクトおよびキューの管理

レプリケート・オブジェクトの変更 .....	9-2
静止中のマスター・グループにあるレプリケート・オブジェクトの変更 .....	9-2
変更をレプリケートしない表の変更 .....	9-5
レプリケーションの使用禁止 .....	9-6
レプリケーション機能の再有効化 .....	9-6
レプリケート・トリガーの起動を1回のみにする .....	9-7
レプリケート表での LONG 列の LOB 列への変換 .....	9-7
レプリケート表間の違いの判別 .....	9-9
DIFFERENCES プロシージャの使用 .....	9-9
RECTIFY プロシージャの使用 .....	9-9
遅延トランザクション・キューの管理 .....	9-13
遅延トランザクション・キューのプッシュ .....	9-13
遅延トランザクション・キューのページ .....	9-14
AnyData 型を使用した遅延コールの引数の値の判定 .....	9-15
エラー・キューの管理 .....	9-18
受信者としてのエラー・トランザクションの再実行 .....	9-18
代替ユーザーとしてのエラー・トランザクションの再実行 .....	9-19

## 10 レプリケーション環境の監視

マスター・レプリケーション環境の監視 .....	10-2
マスター・サイトの監視 .....	10-2
マスター・グループの監視 .....	10-4
マスターの監視 .....	10-6
マテリアライズド・ビュー・サイトの監視 .....	10-11
マテリアライズド・ビュー・サイトの一般情報のリスト表示 .....	10-11
マテリアライズド・ビュー・グループの一般情報のリスト表示 .....	10-12
マテリアライズド・ビューの情報のリスト表示 .....	10-13
マテリアライズド・ビュー・サイトにあるリフレッシュ・グループの情報のリスト表示 .....	10-15

マテリアライズド・ビュー・サイトにある各リフレッシュ・ジョブのジョブ ID の判定 .....	10-16
現在リフレッシュしているマテリアライズド・ビューの判定 .....	10-17
<b>管理要求の監視 .....</b>	<b>10-17</b>
管理要求の一般情報のリスト表示 .....	10-17
管理要求エラーの原因の判定 .....	10-18
管理要求を実行するジョブの一般情報のリスト表示 .....	10-19
<b>遅延トランザクション・キューの監視 .....</b>	<b>10-20</b>
トランザクション伝播の監視 .....	10-20
正常に伝播されたトランザクションのページの監視 .....	10-23
<b>エラー・キューの監視 .....</b>	<b>10-26</b>
レプリケーション・サイトにあるエラー・トランザクションの一般情報のリスト表示 .....	10-26
エラー・トランザクションの割合の判定 .....	10-27
各起点マスター・サイトからのエラー・トランザクションの数のリスト表示 .....	10-27
レプリケーション・サイトのエラー・トランザクションに関するエラー・メッセージの リスト表示 .....	10-28
レプリケーション・サイトのエラー操作の判定 .....	10-28
<b>レプリケーション環境のパフォーマンスの監視 .....</b>	<b>10-29</b>
レプリケーション・トランザクションの行変更の平均数の追跡 .....	10-29
遅延トランザクション・キューに入るトランザクションの率の追跡 .....	10-29
トランザクションの伝播のために作成されるネットワークの平均通信量の判定 .....	10-30
リモート・サイトでトランザクションを適用するための平均時間の判定 .....	10-31
パラレル伝播ジョブがスリープに費やす時間の割合の判定 .....	10-31
DEFSCHEDULE ビューのリモート・マスター・サイトに関する統計の消去 .....	10-32
V\$REPLPROP を使用した遅延トランザクションのパラレル伝播の監視 .....	10-32

## 第 III 部 レプリケーション・マネージメント API パッケージ・リファレンス

### 11 レプリケーション・マネージメント API リファレンスの概要

Oracle レプリケーション・マネージメント API の使用例 .....	11-2
レプリケーション・マネージメント API を使用する際の考慮事項 .....	11-3
レプリケーション・マネージメント・ツールとレプリケーション・マネージメント API .....	11-3
日時データ型および期間データ型の略称 .....	11-4

## 12 DBMS\_DEFER

CALL プロシージャ .....	12-2
COMMIT_WORK プロシージャ .....	12-4
<i>datatype_ARG</i> プロシージャ .....	12-5
TRANSACTION プロシージャ .....	12-7

## 13 DBMS\_DEFER\_QUERY

GET_ARG_FORM ファンクション .....	13-2
GET_ARG_TYPE ファンクション .....	13-4
GET_CALL_ARGS プロシージャ .....	13-6
GET_ <i>datatype_ARG</i> ファンクション .....	13-7
GET_OBJECT_NULL_VECTOR_ARG ファンクション .....	13-9

## 14 DBMS\_DEFER\_SYS

ADD_DEFAULT_DEST プロシージャ .....	14-4
CLEAR_PROP_STATISTICS プロシージャ .....	14-5
DELETE_DEFAULT_DEST プロシージャ .....	14-6
DELETE_DEF_DESTINATION プロシージャ .....	14-6
DELETE_ERROR プロシージャ .....	14-7
DELETE_TRAN プロシージャ .....	14-7
DISABLED ファンクション .....	14-8
EXCLUDE_PUSH ファンクション .....	14-9
EXECUTE_ERROR プロシージャ .....	14-10
EXECUTE_ERROR_AS_USER プロシージャ .....	14-11
PURGE ファンクション .....	14-12
PUSH ファンクション .....	14-14
REGISTER_PROPAGATOR プロシージャ .....	14-17
SCHEDULE_PURGE プロシージャ .....	14-18
SCHEDULE_PUSH プロシージャ .....	14-20
SET_DISABLED プロシージャ .....	14-22
UNREGISTER_PROPAGATOR プロシージャ .....	14-24
UNSCHEDULE_PURGE プロシージャ .....	14-25
UNSCHEDULE_PUSH プロシージャ .....	14-25



## 15 DBMS\_MVIEW

BEGIN_TABLE_REORGANIZATION プロシージャ .....	15-3
END_TABLE_REORGANIZATION プロシージャ .....	15-4
EXPLAIN_MVIEW プロシージャ .....	15-4
EXPLAIN_REWRITE プロシージャ .....	15-6
I_AM_A_REFRESH ファンクション .....	15-7
PMARKER ファンクション .....	15-7
PURGE_DIRECT_LOAD_LOG プロシージャ .....	15-8
PURGE_LOG プロシージャ .....	15-8
PURGE_MVIEW_FROM_LOG プロシージャ .....	15-10
REFRESH プロシージャ .....	15-12
REFRESH_ALL_MVIEWS プロシージャ .....	15-14
REFRESH_DEPENDENT プロシージャ .....	15-16
REGISTER_MVIEW プロシージャ .....	15-18
UNREGISTER_MVIEW プロシージャ .....	15-20

## 16 DBMS\_OFFLINE\_OG

BEGIN_INSTANTIATION プロシージャ .....	16-2
BEGIN_LOAD プロシージャ .....	16-4
END_INSTANTIATION プロシージャ .....	16-5
END_LOAD プロシージャ .....	16-7
RESUME_SUBSET_OF_MASTERS プロシージャ .....	16-8

## 17 DBMS\_OFFLINE\_SNAPSHOT

BEGIN_LOAD プロシージャ .....	17-2
END_LOAD プロシージャ .....	17-4

## 18 DBMS\_RECTIFIER\_DIFF

DIFFERENCES プロシージャ .....	18-2
RECTIFY プロシージャ .....	18-5

## 19 DBMS\_REFRESH

ADD プロシージャ .....	19-2
CHANGE プロシージャ .....	19-3
DESTROY プロシージャ .....	19-6
MAKE プロシージャ .....	19-6
REFRESH プロシージャ .....	19-9
SUBTRACT プロシージャ .....	19-10

## 20 DBMS\_REPCAT

ADD_GROUPED_COLUMN プロシージャ .....	20-7
ADD_MASTER_DATABASE プロシージャ .....	20-8
ADD_NEW_MASTERS プロシージャ .....	20-10
ADD_PRIORITY_datatype プロシージャ .....	20-16
ADD_SITE_PRIORITY_SITE プロシージャ .....	20-17
ADD_conflicttype_RESOLUTION プロシージャ .....	20-18
ALTER_CATCHUP_PARAMETERS プロシージャ .....	20-23
ALTER_MASTER_PROPAGATION プロシージャ .....	20-25
ALTER_MASTER_REPOBJECT プロシージャ .....	20-27
ALTER_MVIEW_PROPAGATION プロシージャ .....	20-30
ALTER_PRIORITY プロシージャ .....	20-31
ALTER_PRIORITY_datatype プロシージャ .....	20-33
ALTER_SITE_PRIORITY プロシージャ .....	20-34
ALTER_SITE_PRIORITY_SITE プロシージャ .....	20-36
CANCEL_STATISTICS プロシージャ .....	20-37
COMMENT_ON_COLUMN_GROUP プロシージャ .....	20-38
COMMENT_ON_MVIEW_REPSITES プロシージャ .....	20-39
COMMENT_ON_PRIORITY_GROUP/COMMENT_ON_SITE_PRIORITY プロシージャ .....	20-40
COMMENT_ON_REPGROUP プロシージャ .....	20-41
COMMENT_ON_REPOBJECT プロシージャ .....	20-42
COMMENT_ON_REPSITES プロシージャ .....	20-43
COMMENT_ON_conflicttype_RESOLUTION プロシージャ .....	20-45
COMPARE_OLD_VALUES プロシージャ .....	20-47
CREATE_MASTER_REPGROUP プロシージャ .....	20-49
CREATE_MASTER_REPOBJECT プロシージャ .....	20-50
CREATE_MVIEW_REPGROUP プロシージャ .....	20-54
CREATE_MVIEW_REPOBJECT プロシージャ .....	20-55

DEFINE_COLUMN_GROUP プロシージャ .....	20-59
DEFINE_PRIORITY_GROUP プロシージャ .....	20-60
DEFINE_SITE_PRIORITY プロシージャ .....	20-61
DO_DEFERRED_REPCAT_ADMIN プロシージャ .....	20-62
DROP_COLUMN_GROUP プロシージャ .....	20-63
DROP_GROUPED_COLUMN プロシージャ .....	20-64
DROP_MASTER_REPGROUP プロシージャ .....	20-66
DROP_MASTER_REPOBJECT プロシージャ .....	20-67
DROP_MVIEW_REPGROUP プロシージャ .....	20-69
DROP_MVIEW_REPOBJECT プロシージャ .....	20-70
DROP_PRIORITY プロシージャ .....	20-71
DROP_PRIORITY_GROUP プロシージャ .....	20-72
DROP_PRIORITY_datatype プロシージャ .....	20-73
DROP_SITE_PRIORITY プロシージャ .....	20-75
DROP_SITE_PRIORITY_SITE プロシージャ .....	20-76
DROP_conflicttype_RESOLUTION プロシージャ .....	20-77
EXECUTE_DDL プロシージャ .....	20-79
GENERATE_MVIEW_SUPPORT プロシージャ .....	20-80
GENERATE_REPLICATION_SUPPORT プロシージャ .....	20-82
MAKE_COLUMN_GROUP プロシージャ .....	20-84
PREPARE_INSTANTIATED_MASTER プロシージャ .....	20-85
PURGE_MASTER_LOG プロシージャ .....	20-87
PURGE_STATISTICS プロシージャ .....	20-88
REFRESH_MVIEW_REPGROUP プロシージャ .....	20-89
REGISTER_MVIEW_REPGROUP プロシージャ .....	20-91
REGISTER_STATISTICS プロシージャ .....	20-92
RELOCATE_MASTERDEF プロシージャ .....	20-93
REMOVE_MASTER_DATABASES プロシージャ .....	20-94
RENAME_SHADOW_COLUMN_GROUP プロシージャ .....	20-96
REPCAT_IMPORT_CHECK プロシージャ .....	20-97
RESUME_MASTER_ACTIVITY プロシージャ .....	20-98
RESUME_PROPAGATION_TO_MDEF プロシージャ .....	20-99
SEND_OLD_VALUES プロシージャ .....	20-100
SET_COLUMNS プロシージャ .....	20-102
SPECIFY_NEW_MASTERS プロシージャ .....	20-104
SUSPEND_MASTER_ACTIVITY プロシージャ .....	20-106
SWITCH_MVIEW_MASTER プロシージャ .....	20-107

UNDO_ADD_NEW_MASTERS_REQUEST プロシージャ .....	20-109
UNREGISTER_MVIEW_REPGROUP プロシージャ .....	20-111
VALIDATE ファンクション .....	20-111
WAIT_MASTER_LOG プロシージャ .....	20-114

## 21 DBMS\_REPCAT\_ADMIN

GRANT_ADMIN_ANY_SCHEMA プロシージャ .....	21-3
GRANT_ADMIN_SCHEMA プロシージャ .....	21-4
REGISTER_USER_REPGROUP プロシージャ .....	21-4
REVOKE_ADMIN_ANY_SCHEMA プロシージャ .....	21-6
REVOKE_ADMIN_SCHEMA プロシージャ .....	21-7
UNREGISTER_USER_REPGROUP プロシージャ .....	21-8

## 22 DBMS\_REPCAT\_INSTANTIATE

DROP_SITE_INSTANTIATION プロシージャ .....	22-2
INSTANTIATE_OFFLINE ファンクション .....	22-3
INSTANTIATE_ONLINE ファンクション .....	22-5

## 23 DBMS\_REPCAT\_RGT

ALTER_REFRESH_TEMPLATE プロシージャ .....	23-4
ALTER_TEMPLATE_OBJECT プロシージャ .....	23-6
ALTER_TEMPLATE_PARM プロシージャ .....	23-8
ALTER_USER_AUTHORIZATION プロシージャ .....	23-10
ALTER_USER_PARM_VALUE プロシージャ .....	23-12
COMPARE_TEMPLATES ファンクション .....	23-14
COPY_TEMPLATE ファンクション .....	23-15
CREATE_OBJECT_FROM_EXISTING ファンクション .....	23-17
CREATE_REFRESH_TEMPLATE ファンクション .....	23-18
CREATE_TEMPLATE_OBJECT ファンクション .....	23-20
CREATE_TEMPLATE_PARM ファンクション .....	23-23
CREATE_USER_AUTHORIZATION ファンクション .....	23-25
CREATE_USER_PARM_VALUE ファンクション .....	23-27
DELETE_RUNTIME_PARMS プロシージャ .....	23-29
DROP_ALL_OBJECTS プロシージャ .....	23-30
DROP_ALL_TEMPLATE_PARMS プロシージャ .....	23-31

DROP_ALL_TEMPLATE_SITES プロシージャ .....	23-32
DROP_ALL_TEMPLATES プロシージャ .....	23-33
DROP_ALL_USER_AUTHORIZATIONS プロシージャ .....	23-33
DROP_ALL_USER_PARM_VALUES プロシージャ .....	23-34
DROP_REFRESH_TEMPLATE プロシージャ .....	23-35
DROP_SITE_INSTANTIATION プロシージャ .....	23-36
DROP_TEMPLATE_OBJECT プロシージャ .....	23-37
DROP_TEMPLATE_PARM プロシージャ .....	23-38
DROP_USER_AUTHORIZATION プロシージャ .....	23-39
DROP_USER_PARM_VALUE プロシージャ .....	23-40
GET_RUNTIME_PARM_ID ファンクション .....	23-41
INSERT_RUNTIME_PARMS プロシージャ .....	23-41
INstantiate_OFFLINE ファンクション .....	23-43
INstantiate_ONLINE ファンクション .....	23-45
LOCK_TEMPLATE_EXCLUSIVE プロシージャ .....	23-48
LOCK_TEMPLATE_SHARED プロシージャ .....	23-48

## 24 DBMS\_REPUTIL

REPLICATION_OFF プロシージャ .....	24-3
REPLICATION_ON プロシージャ .....	24-3
REPLICATION_IS_ON ファンクション .....	24-3
FROM_REMOTE ファンクション .....	24-4
GLOBAL_NAME ファンクション .....	24-4
MAKE_INTERNAL_PKG プロシージャ .....	24-5
SYNC_UP_REP プロシージャ .....	24-6

## 第 IV 部 レプリケーション・データ・ディクショナリ・リファレンス

### 25 レプリケーション・カタログ・ビュー

レプリケーション・カタログ・ビューの概要 .....	25-2
DBA_REGISTERED_MVIEW_GROUPS .....	25-5
ALL_REPCAT_REFRESH_TEMPLATES .....	25-6
ALL_REPCAT_TEMPLATE_OBJECTS .....	25-7
ALL_REPCAT_TEMPLATE_PARMS .....	25-9
ALL_REPCAT_TEMPLATE_SITES .....	25-11
ALL_REPCAT_USER_AUTHORIZATIONS .....	25-12

ALL_REPCAT_USER_PARM_VALUES .....	25-13
ALL_REPCATLOG .....	25-15
ALL_REPCOLUMN .....	25-16
ALL_REPCOLUMN_GROUP .....	25-19
ALL_REPCONFLICT .....	25-19
ALL_REPDDL .....	25-20
ALL_REPGENOBJECTS .....	25-21
ALL_REPGROUP .....	25-22
ALL_REPGROUP_PRIVILEGES .....	25-23
ALL_REPGROUPED_COLUMN .....	25-24
ALL_REPKEY_COLUMNS .....	25-25
ALL_REPOBJECT .....	25-26
ALL_REPPARAMETER_COLUMN .....	25-28
ALL_REPPRIORITY .....	25-29
ALL_REPPRIORITY_GROUP .....	25-30
ALL_REPPROP .....	25-31
ALL_REPRESOL_STATS_CONTROL .....	25-32
ALL_REPRESOLUTION .....	25-33
ALL_REPRESOLUTION_METHOD .....	25-34
ALL_REPRESOLUTION_STATISTICS .....	25-34
ALL_REPSITES .....	25-35
DBA_REPCAT_REFRESH_TEMPLATES .....	25-36
DBA_REPCAT_TEMPLATE_OBJECTS .....	25-37
DBA_REPCAT_TEMPLATE_PARS .....	25-37
DBA_REPCAT_TEMPLATE_SITES .....	25-37
DBA_REPCAT_USER_AUTHORIZATIONS .....	25-37
DBA_REPCAT_USER_PARM_VALUES .....	25-37
DBA_REPCATLOG .....	25-38
DBA_REPCOLUMN .....	25-38
DBA_REPCOLUMN_GROUP .....	25-38
DBA_REPCONFLICT .....	25-38
DBA_REPDDL .....	25-38
DBA_REPEXTENSIONS .....	25-38
DBA_REPGENOBJECTS .....	25-41
DBA_REPGROUP .....	25-41
DBA_REPGROUP_PRIVILEGES .....	25-41
DBA_REPGROUPED_COLUMN .....	25-41

DBA_REPKKEY_COLUMNS .....	25-42
DBA_REPOBJECT .....	25-42
DBA_REPPARAMETER_COLUMN .....	25-42
DBA_REPPRIORITY .....	25-42
DBA_REPPRIORITY_GROUP .....	25-42
DBA_REPPROP .....	25-42
DBA_REPRESOL_STATS_CONTROL .....	25-43
DBA_REPRESOLUTION .....	25-43
DBA_REPRESOLUTION_METHOD .....	25-43
DBA_REPRESOLUTION_STATISTICS .....	25-43
DBA_REPSITES .....	25-44
DBA_REPSITES_NEW .....	25-44
USER_REPCAT_REFRESH_TEMPLATES .....	25-45
USER_REPCAT_TEMPLATE_OBJECTS .....	25-45
USER_REPCAT_TEMPLATE_PARMS .....	25-46
USER_REPCAT_TEMPLATE_SITES .....	25-46
USER_REPCAT_USER_AUTHORIZATION .....	25-46
USER_REPCAT_USER_PARM_VALUES .....	25-46
USER_REPCATLOG .....	25-47
USER_REPCOLUMN .....	25-47
USER_REPCOLUMN_GROUP .....	25-47
USER_REPCONFLICT .....	25-47
USER_REPDDL .....	25-48
USER_REPGENOBJECTS .....	25-48
USER_REPGROUP .....	25-48
USER_REPGROUP_PRIVILEGES .....	25-48
USER_REPGROUPED_COLUMN .....	25-48
USER_REPKKEY_COLUMNS .....	25-48
USER_REPOBJECT .....	25-49
USER_REPPARAMETER_COLUMN .....	25-49
USER_REPPRIORITY .....	25-49
USER_REPPRIORITY_GROUP .....	25-49
USER_REPPROP .....	25-50
USER_REPRESOL_STATS_CONTROL .....	25-50
USER_REPRESOLUTION .....	25-50
USER_REPRESOLUTION_METHOD .....	25-50
USER_REPRESOLUTION_STATISTICS .....	25-51
USER_REPSITES .....	25-51

## 26 レプリケーション動的パフォーマンス・ビュー

V\$MVREFRESH .....	26-2
V\$REPLPROP .....	26-2
V\$REPLQUEUE .....	26-4

## 27 遅延トランザクション・ビュー

DEFCALL .....	27-2
DEFCALLDEST .....	27-2
DEFDEFAULTDEST .....	27-3
DEFERRCOUNT .....	27-3
DEFERROR .....	27-3
DEFLOB .....	27-4
DEFPROPAGATOR .....	27-4
DEFSCHEDULE .....	27-5
DEFTRAN .....	27-7
DEFTRANDEST .....	27-8

## 28 マテリアライズド・ビューとリフレッシュ・グループ・ビュー

## 第 V 部 付録

### A セキュリティ・オプション

マルチマスター・レプリケーションのセキュリティ設定 .....	A-2
トラステッド・セキュリティとアントラステッド・セキュリティの比較 .....	A-3
マテリアライズド・ビュー・レプリケーションのセキュリティ設定 .....	A-7
トラステッド・セキュリティとアントラステッド・セキュリティの比較 .....	A-8

### B ユーザー定義の競合解消メソッド

ユーザー定義の競合解消メソッド .....	B-2
競合解消メソッドのパラメータ .....	B-2
更新の競合の解消 .....	B-3
一意性競合の解消 .....	B-3
削除の競合の解消 .....	B-3
複数層マテリアライズド・ビューとユーザー定義の競合解消メソッド .....	B-4



ユーザー定義の競合解消メソッドの制限事項 .....	B-4
ユーザー定義の競合解消メソッドの例 .....	B-6
<b>ユーザー定義の競合通知方法 .....</b>	<b>B-8</b>
競合通知ログの作成 .....	B-8
競合通知パッケージの作成 .....	B-9
<b>競合解消情報の表示 .....</b>	<b>B-11</b>

## 索引



---

# はじめに

『Oracle9i レプリケーション・マネージメント API リファレンス』では、レプリケーション・マネージメント API の特長および機能を説明します。特に、レプリケーション・マネージメント API の各パッケージの参照情報とそれらの使用例を示します。

さらに、このリファレンスには、レプリケーションに重要なレプリケーション・カタログおよびその他のデータ・ディクショナリ・ビューに関する参照情報も含まれています。

ここでは、次の項目を説明します。

- [対象読者](#)
- [このマニュアルの構成](#)
- [関連文書](#)
- [表記規則](#)

## 対象読者

このリファレンスは、レプリケーション環境の開発およびメンテナンスを行うデータベース管理者やアプリケーション開発者を対象としています。管理者やアプリケーション開発者は、次のタスクの1つ以上を実行します。

- レプリケーション・サイトの構成
- マスター・グループの作成
- 配置テンプレートの作成
- マテリアライズド・ビュー・グループの作成
- 競合解消の構成
- レプリケーション環境の管理
- レプリケーション・マネージメント API の使用
- データ・ディクショナリ・ビューを使用したレプリケーション環境の監視
- セキュリティ・オプションの計画と構成

このリファレンスは、各自がリレーショナル・データベースの概念、分散データベースの管理方法、PL/SQL（プロシージャ・レプリケーションを使用する場合）、およびアドバンスド・レプリケーション環境を実行するオペレーティング・システムについてよく理解していることを前提としています。

## このマニュアルの構成

このマニュアルの構成は、次のとおりです。

### 第 I 部「レプリケーション環境の構成」

レプリケーション・マネージメント API を使用してマルチマスター・レプリケーションおよびマテリアライズド・ビュー・レプリケーションを設定する方法を示します。また、競合解消メソッドを構成するための指示およびレプリケーション・マネージメント API を使用したレプリケーション環境を管理するための指示も含まれます。

### 第 1 章「レプリケーションの概要」

レプリケーション・マネージメント API を使用したレプリケーション環境の構築プロセスの概要を説明します。また、レプリケーション環境の構築の前提条件も説明します。

## 第2章「レプリケーション・サイトの構築」

マスター・サイトとマテリアライズド・ビュー・サイトの設定プロセスの詳細を説明します。この章の内容は、新しいレプリケーション環境を構築する場合や、新しいマスター・サイトまたはマテリアライズド・ビュー・サイトを既存のレプリケーション環境に追加する場合に必要なります。

## 第3章「マスター・グループの作成」

マルチマスター・レプリケーション用のマスター・グループの作成方法や、マテリアライズド・ビュー・サイト用のマスターとしてのマスター・グループの作成方法を説明します。第3章ではマスター・グループを作成し、第2章で設定した3つのマスター・サイトの間でデータをレプリケートする方法を説明します。

## 第4章「配置テンプレートの作成」

配置テンプレートを使用してマテリアライズド・ビュー環境を構築する方法を説明します。配置テンプレートを使用すると、任意の数のマテリアライズド・ビュー・サイトに対し、最も効果的にマテリアライズド・ビュー環境を配布できます。

## 第5章「マテリアライズド・ビュー・グループの作成」

マテリアライズド・ビュー・グループを使用してマテリアライズド・ビュー環境を構築する方法を説明します。配置テンプレートのみでは必要な環境が構築できない場合は、第5章を参照してください。この章では、マテリアライズド・ビュー・サイトに対してマテリアライズド・ビュー環境を構築する方法を説明します。

## 第6章「競合解消の構成」

データの競合が発生した場合にすべてのサイトのデータを収束させるために必要な、競合解消メソッドを説明します。

## 第II部「レプリケーション環境の管理と監視」

レプリケーション・マネージメント API を使用したレプリケーション環境の管理方法を示します。

## 第7章「マスター・レプリケーション環境の管理」

マルチマスター・レプリケーション環境を管理するために必要な管理タスクを説明します。内容には、新規マスター・サイトの追加、マスター・グループの管理その他が含まれます。

## 第8章「マテリアライズド・ビュー・レプリケーション環境の管理」

マテリアライズド・ビュー・レプリケーション環境を管理するために必要な管理タスクを説明します。内容には、グループ所有者の使用、マテリアライズド・ビュー・ログの管理、オフライン・インスタンスエーションその他が含まれます。

## 第9章「レプリケーション・オブジェクトおよびキューの管理」

レプリケーション環境を管理するために必要な管理タスクを説明します。内容にはレプリケート・オブジェクトの変更、遅延トランザクション・キューの管理、エラー・キューその他が含まれます。

## 第10章「レプリケーション環境の監視」

レプリケーション環境を監視するための問合せを説明します。

## 第III部「レプリケーション・マネージメント API パッケージ・リファレンス」

次に示すレプリケーション・マネージメント API の参照情報を示します。各パッケージのプロシージャおよびファンクションと、それらのパラメータ、および各プロシージャまたはファンクションから呼び出される例外が含まれます。

## 第11章「レプリケーション・マネージメント API リファレンスの概要」

レプリケーション・マネージメント API の概要と使用例を紹介します。

## 第12章「DBMS\_DEFER」

DBMS\_DEFER パッケージのプロシージャを説明します。

## 第13章「DBMS\_DEFER\_QUERY」

DBMS\_DEFER\_QUERY パッケージのプロシージャおよびファンクションを説明します。

## 第14章「DBMS\_DEFER\_SYS」

DBMS\_DEFER\_SYS パッケージのプロシージャおよびファンクションを説明します。

## 第15章「DBMS\_MVIEW」

DBMS\_MVIEW パッケージのプロシージャおよびファンクションを説明します。

## 第16章「DBMS\_OFFLINE\_OG」

DBMS\_OFFLINE\_OG パッケージのプロシージャを説明します。

## 第17章「DBMS\_OFFLINE\_SNAPSHOT」

DBMS\_OFFLINE\_SNAPSHOT パッケージのプロシージャを説明します。

## 第18章「DBMS\_RECTIFIER\_DIFF」

DBMS\_RECTIFIER\_DIFF パッケージのプロシージャを説明します。

## 第19章「DBMS\_REFRESH」

DBMS\_REFRESH パッケージのプロシージャを説明します。

## 第20章「DBMS\_REPCAT」

DBMS\_REPCAT パッケージのプロシージャおよびファンクションを説明します。

## 第21章「DBMS\_REPCAT\_ADMIN」

DBMS\_REPCAT\_ADMIN パッケージのプロシージャを説明します。

## 第22章「DBMS\_REPCAT\_INSTANTIATE」

DBMS\_REPCAT\_INSTANTIATE パッケージのプロシージャおよびファンクションを説明します。

## 第23章「DBMS\_REPCAT\_RGT」

DBMS\_REPCAT\_RGT パッケージのプロシージャおよびファンクションを説明します。

## 第24章「DBMS\_REPUTIL」

DBMS\_REPUTIL パッケージのプロシージャおよびファンクションを説明します。

## 第IV部「レプリケーション・データ・ディクショナリ・リファレンス」

レプリケーション環境の情報を提供するデータ・ディクショナリ・ビューを説明します。

## 第25章「レプリケーション・カタログ・ビュー」

レプリケーション・カタログを説明します。このカタログには、マスター・サイトやマテリアライズド・ビュー・サイトに対して、レプリケートされるオブジェクトのタイプ、レプリケート先、レプリケーションの実行中にエラーが発生していないかどうか、などを判断するために使用するデータ・ディクショナリ・ビューが含まれます。

## 第26章「レプリケーション動的パフォーマンス・ビュー」

マテリアライズド・ビューが現在リフレッシュ中であるかを判断したり、遅延トランザクション・キューの統計を判断するためにマスター・サイトやマテリアライズド・ビュー・サイトが使用する動的パフォーマンス・ビューを説明します。

## 第27章「遅延トランザクション・ビュー」

遅延トランザクションの情報が含まれるデータ・ディクショナリ・ビューを説明します。これらのビューは、トランザクションの接続先、トランザクションを構成する遅延コール、トランザクションの処理中に発生するエラーなど、遅延トランザクションに関する情報を提供します。

## 第28章「マテリアライズド・ビューとリフレッシュ・グループ・ビュー」

マテリアライズド・ビューおよびマテリアライズド・ビュー・リフレッシュ・グループの情報を備えたデータ・ディクショナリ・ビューを説明します。

## 第 V 部 「付録」

次の付録が含まれます。

### 付録 A 「セキュリティ・オプション」

レプリケーション・マネージメント API を使用したマルチマスター・レプリケーションとマテリアライズド・ビュー・レプリケーションのセキュリティ設定を説明します。

### 付録 B 「ユーザー定義の競合解消メソッド」

レプリケーション・マネージメント API を使用したユーザー定義の競合解消メソッドおよび通知関数関数の作成方法を説明します。

## 関連文書

詳細は、次の Oracle マニュアルを参照してください。

- 『Oracle9i アドバンスド・レプリケーション』
- 『Oracle9i データベース概要』
- 『Oracle9i データベース管理者ガイド』
- 『Oracle9i SQL リファレンス』
- 『PL/SQL ユーザーズ・ガイドおよびリファレンス』（プロシージャ・レプリケーションを使用する場合）

Oracle9i ドキュメント・セットのその他ドキュメントで、特定の項目に関する詳細情報を得ることができます。

このマニュアルに記載されている例の多くは、Oracle のインストール時にデフォルトでインストールされるシード・データベースのサンプル・スキーマを使用しています。これらのスキーマがどのように作成されているか、およびその使用方法については、『Oracle9i サンプル・スキーマ』を参照してください。

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

<http://otn.oracle.co.jp/membership/>

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

<http://otn.oracle.co.jp/document/>



# 表記規則

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

- [本文の表記規則](#)
- [コード例の表記規則](#)
- [Windows オペレーティング・システムの表記規則](#)

## 本文の表記規則

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

表記規則	意味	例
太字	太字は、本文中に定義されている用語または用語集に含まれている用語、あるいはその両方を示します。	この句を指定する場合は、 <b>索引構成表</b> を作成します。
固定幅フォントの大文字	固定幅フォントの大文字は、システムにより指定される要素を示します。この要素には、パラメータ、権限、データ型、Recovery Manager キーワード、SQL キーワード、SQL*Plus またはユーティリティ・コマンド、パッケージとメソッドの他、システム指定の列名、データベース・オブジェクトと構造体、ユーザー名、およびロールがあります。	この句は NUMBER 列に対してのみ指定できます。  BACKUP コマンドを使用すると、データベースのバックアップを作成できます。  USER_TABLES データ・ディクショナリ・ビューの TABLE_NAME 列を問い合わせます。  DBMS_STATS.GENERATE_STATS プロシージャを使用します。
固定幅フォントの小文字	固定幅フォントの小文字は、実行可能ファイル、ファイル名、ディレクトリ名およびサンプルのユーザー指定要素を示します。この要素には、コンピュータ名とデータベース名、ネット・サービス名、接続識別子の他、ユーザー指定のデータベース・オブジェクトと構造体、列名、パッケージとクラス、ユーザー名とロール、プログラム・ユニット、およびパラメータ値があります。  <b>注意：</b> 一部のプログラム要素には、大文字と小文字の両方が使用されます。この場合は記載されているとおりに入力してください。	sqlplus と入力して SQL*Plus をオープンします。  パスワードは orapwd ファイルに指定されています。  /disk1/oracle/dbs ディレクトリ内で、データ・ファイルと制御ファイルのバックアップを作成します。  department_id、department_name および location_id の各列は、hr.departments 表にあります。  初期化パラメータ QUERY_REWRITE_ENABLED を true に設定します。  oe ユーザーで接続します。  これらのメソッドは、JRepUtil クラスで実装されます。

表記規則	意味	例
固定幅フォントの 小文字の イタリック	固定幅フォントの小文字のイタリックは、 プレースホルダまたは変数を示します。	<i>parallel_clause</i> を指定できます。  <i>Uold_release</i> .SQL を実行します。 <i>old_release</i> はアップグレード前にインストールしていたリリースです。

コード例の表記規則

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

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

次の表は、コード例の記載上の表記規則と使用例を示しています。

表記規則	意味	例
[ ]	大カッコで囲まれている項目は、1 つ以上のオプション項目を示します。大カッコ自体は入力しないでください。	DECIMAL ( <i>digits</i> [ , <i>precision</i> ])
{ }	中カッコで囲まれている項目は、そのうちの 1 つのみが必要であることを示します。中カッコ自体は入力しないでください。	{ENABLE   DISABLE}
	縦線は、大カッコまたは中カッコ内の複数の選択肢を区切るために使用します。オプションのうち 1 つを入力します。縦線自体は入力しないでください。	{ENABLE   DISABLE} [COMPRESS   NOCOMPRESS]
...	水平の省略記号は、次のいずれかを示します。 <ul style="list-style-type: none"><li>■ 例に直接関係のないコード部分が省略されていること。</li><li>■ コードの一部が繰り返し可能なこと。</li></ul>	CREATE TABLE ... AS <i>subquery</i> ;  SELECT <i>col1</i> , <i>col2</i> , ... , <i>coln</i> FROM <i>employees</i> ;
. . . .	垂直の省略記号は、例に直接関係のない数行のコードが省略されていることを示します。	SQL> SELECT NAME FROM V\$DATAFILE; NAME ----- /fsl/dbs/tbs_01.dbf /fsl/dbs/tbs_02.dbf . . . /fsl/dbs/tbs_09.dbf 9 rows selected.

表記規則	意味	例
その他の表記	大カッコ、中カッコ、縦線および省略記号以外の記号は、表示されているとおりに入力してください。	acctbal NUMBER(11,2); acct      CONSTANT NUMBER(4) := 3;
イタリック	イタリックの文字は、特定の値を指定する必要があるプレースホルダまたは変数を示します。	CONNECT SYSTEM/system_password DB_NAME = database_name
大文字	大文字は、システムにより指定される要素を示します。これらの用語は、ユーザー定義の用語と区別するために大文字で記載されています。大カッコで囲まれている場合を除き、記載されているとおりの順序とスペルで入力してください。ただし、この種の用語は大 / 小文字区別がないため、小文字でも入力できます。	SELECT last_name, employee_id FROM employees; SELECT * FROM USER_TABLES; DROP TABLE hr.employees;
小文字	小文字は、ユーザー指定のプログラム要素を示します。たとえば、表名、列名またはファイル名を示します。  <b>注意：</b> 一部のプログラム要素には、大文字と小文字の両方が使用されます。この場合は記載されているとおりに入力してください。	SELECT last_name, employee_id FROM employees; sqlplus hr/hr CREATE USER mjones IDENTIFIED BY ty3MU9;

## Windows オペレーティング・システムの表記規則

次の表は、Windows オペレーティング・システムの表記規則と使用例を示しています。

表記規則	意味	例
「スタート」→を 選択	プログラムの起動方法。	Database Configuration Assistant を起動するには、「スタート」→「プログラム」→「Oracle - HOME_NAME」→「Configuration and Migration Tools」→「Database Configuration Assistant」を選択します。
ファイル名と ディレクトリ名	ファイル名とディレクトリ名には、大 / 小文字区別はありません。特殊文字のうち、左山カッコ (<)、右山カッコ (>)、コロン (:)、二重引用符 (")、スラッシュ (/)、パイプ ( ) およびハイフン (-) は使用できません。特殊文字の円記号 (¥) は、引用符で囲まれていても要素のセパレータとして扱われます。¥¥ で始まるファイル名は、Windows では汎用命名規則を使用しているものとみなされます。	c:¥winnt"¥"system32 は C:¥WINNT¥SYSTEM32 と同じです。
C:¥>	現行のハード・ディスク・ドライブの Windows コマンド・プロンプトを表します。コマンド・プロンプトでのエスケープ文字はカレット (^) です。プロンプトには、作業中のサブディレクトリが反映されます。このマニュアルでは、コマンド・プロンプトと呼んでいます。	C:¥oracle¥oradata>
特殊文字	Windows コマンド・プロンプトでは、特殊文字の二重引用符 (") のエスケープ文字として、特殊文字の円記号 (¥) が必要な場合があります。カッコと一重引用符 (') には、エスケープ文字は不要です。エスケープ文字と特殊文字の詳細は、Windows オペレーティング・システムのマニュアルを参照してください。	C:¥>exp scott/tiger TABLES=emp QUERY=¥"WHERE job='SALESMAN' and sal<1600¥" C:¥>imp SYSTEM/password FROMUSER=scott TABLES=(emp, dept)
HOME_NAME	Oracle ホーム名を表します。ホーム名は英数字で 16 文字以内です。ホーム名に使用できる特殊文字は、アンダースコアのみです。	C:¥> net start OracleHOME_NAME_TNSListener

表記規則	意味	例
<code>ORACLE_HOME</code> と <code>ORACLE_BASE</code>	<p>Oracle8 リリース 8.0 以前では、Oracle コンポーネントをインストールすると、すべてのサブディレクトリはデフォルトで次のいずれかの名前を使用して、トップレベルの <code>ORACLE_HOME</code> ディレクトリの下に置かれていました。</p> <ul style="list-style-type: none"><li>■ Windows NT の場合は <code>C:\orant</code></li><li>■ Windows 98 の場合は <code>C:\orawin98</code></li></ul> <p>このリリースは、Optimal Flexible Architecture (OFA) のガイドラインに従っています。すべてのサブディレクトリがトップレベルの <code>ORACLE_HOME</code> ディレクトリの下にあるとはかぎりません。<code>ORACLE_BASE</code> というトップレベルのディレクトリがあり、これはデフォルトでは <code>C:\oracle</code> です。他の Oracle ソフトウェアがインストールされていないコンピュータに最新の Oracle リリースをインストールすると、最初の Oracle ホーム・ディレクトリのデフォルト設定は <code>C:\oracle\orann</code> となります。この場合、<code>nn</code> は最新リリース番号です。Oracle ホーム・ディレクトリは、<code>ORACLE_BASE</code> の直下にあります。</p> <p>このマニュアルに記載されているディレクトリ・パスの例は、すべて OFA の表記規則に準拠しています。</p>	<p><code>%ORACLE_HOME%\rdbms\admin</code> ディレクトリにアクセスします。</p>



# 第 I 部

---

## レプリケーション環境の構成

第 I 部ではレプリケーション・マネージメント API を使用してマルチマスター・レプリケーションおよびマテリアライズド・ビュー・レプリケーションを設定する方法について説明します。また、レプリケーション・マネージメント API を使用して競合解消メソッドを構成する方法についても説明します。

第 I 部は、次の章で構成されています。

- [第 1 章「レプリケーションの概要」](#)
- [第 2 章「レプリケーション・サイトの構築」](#)
- [第 3 章「マスター・グループの作成」](#)
- [第 4 章「配置テンプレートの作成」](#)
- [第 5 章「マテリアライズド・ビュー・グループの作成」](#)
- [第 6 章「競合解消の構成」](#)





---

# レプリケーションの概要

この章では、レプリケーション・マネージメント API を使用してレプリケーション環境を構築するプロセスを説明します。

この章では、次の項目を説明します。

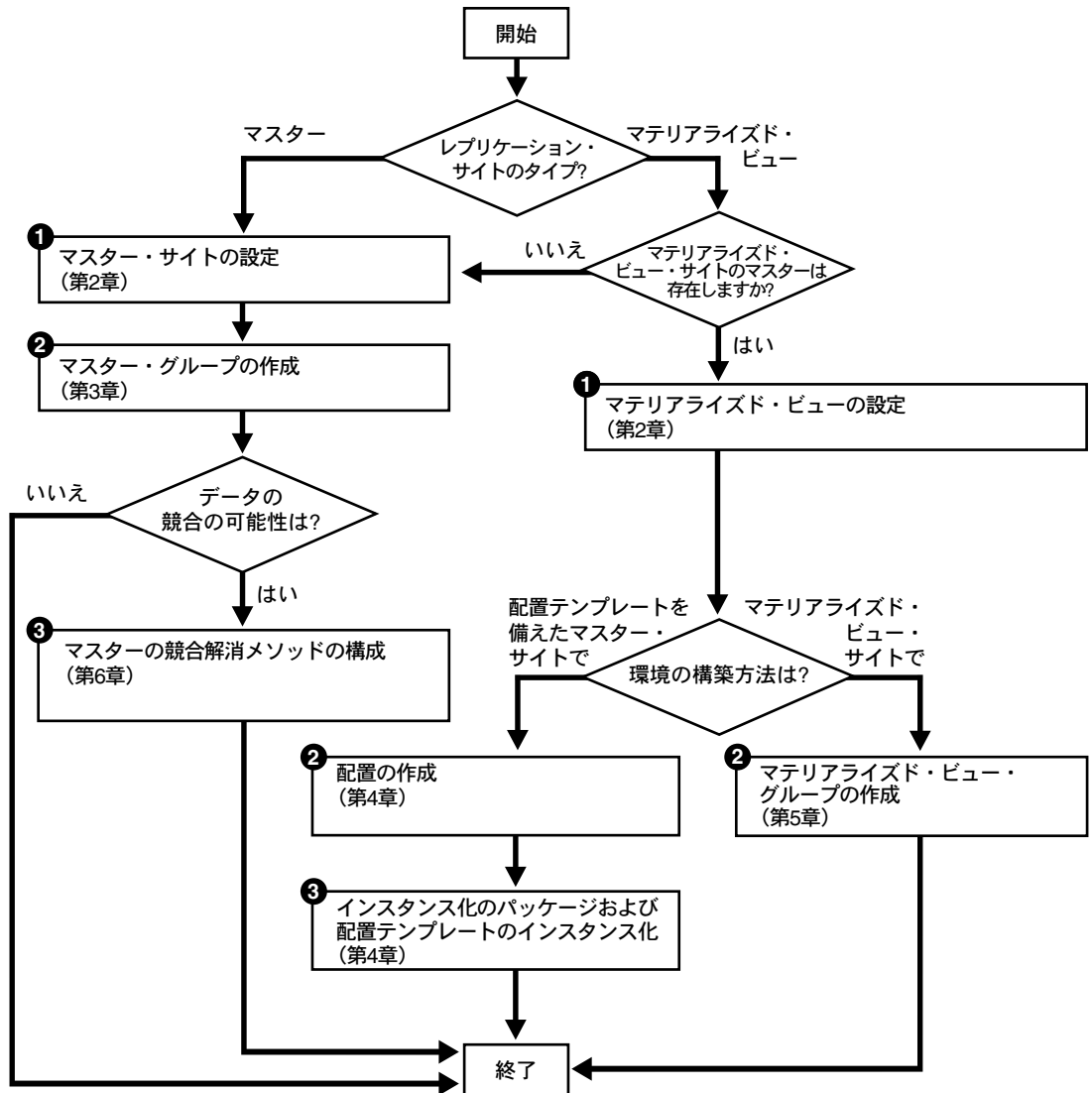
- [レプリケーション環境構築の概要](#)
- [作業を開始する前に](#)

## レプリケーション環境構築の概要

図 1-1 は、レプリケーション環境を構築するための基本的な手順を示します。どのようなタイプのレプリケーション・サイトを構築する場合でも、まずサイトを設定します。

レプリケーション・サイトの設定を終えると、マスター・グループおよびマテリアライズド・ビュー・グループの構築を開始できます。レプリケーション環境の構築後に、必ず第 6 章と第 II 部「[レプリケーション環境の管理と監視](#)」を参照して、競合解消とレプリケーション環境の管理について理解してください。

図 1-1 レプリケーション環境構築のプロセス



## 作業を開始する前に

レプリケーション・サイトを設定する際には、事前にレプリケーション環境が各自のニーズに合うように計画します。計画で考慮する項目を次に示します。

- レプリケート・データベース・オブジェクトを設計します。
- レプリケーションで重要な初期化パラメータの設定を決定します。
- マルチマスター・レプリケーション環境、マテリアライズド・ビュー・レプリケーション環境のどちらを構築するのか、または両レプリケーション環境を組み合わせハイブリッド環境を構築するのかを決定します。
- スケジュール・リンクの構成を決定します。
- スケジュール・ページの構成を決定します。
- シリアル伝播、パラレル伝播のどちらを使用するのかを決定します。
- パラレル伝播を使用する場合は、並列度を決定します。
- マテリアライズド・ビュー環境の構築を計画する場合は、環境構築に配置テンプレートを使用するかどうかを決定します。
- 各自の環境における競合の可能性を分析し、競合の可能性がある場合は、どの競合解消メソッドを使用するかを決定します。
- レプリケーション環境のセキュリティを構成します。
- レプリケーション環境の耐障害性を設計します。

**関連項目：** レプリケーション環境の計画の詳細は『Oracle9i アドバンスド・レプリケーション』を参照してください。

---

## レプリケーション・サイトの構築

この章では、レプリケーション・マネージメント API を使用して、マスター・サイトとマテリアライズド・ビュー・レプリケーション・サイトを設定する方法を説明します。

この章では、次の項目を説明します。

- [レプリケーション・サイト設定の概要](#)
- [マスター・サイトの設定](#)
- [マテリアライズド・ビュー・サイトの設定](#)

## レプリケーション・サイト設定の概要

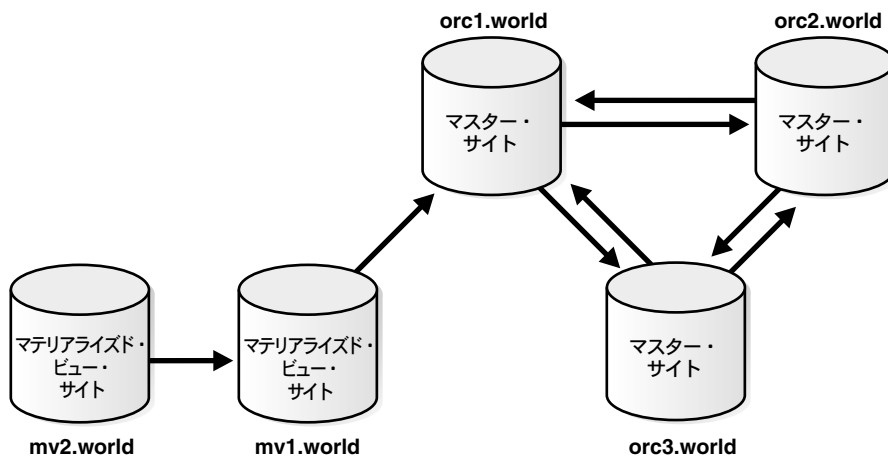
レプリケーション環境を構築する前に、その環境に関係するサイトを設定する必要があります。図 2-2 と図 2-3 に示すように、マスター・サイトとマテリアライズド・ビュー・サイトでは設定方法が異なります。

この章や他の章に含まれる例では、次の 9 つのデータベースを使用します。

- orcl.world
- orc2.world
- orc3.world
- orc4.world
- orc5.world
- mv1.world
- mv2.world
- mv3.world
- mv4.world

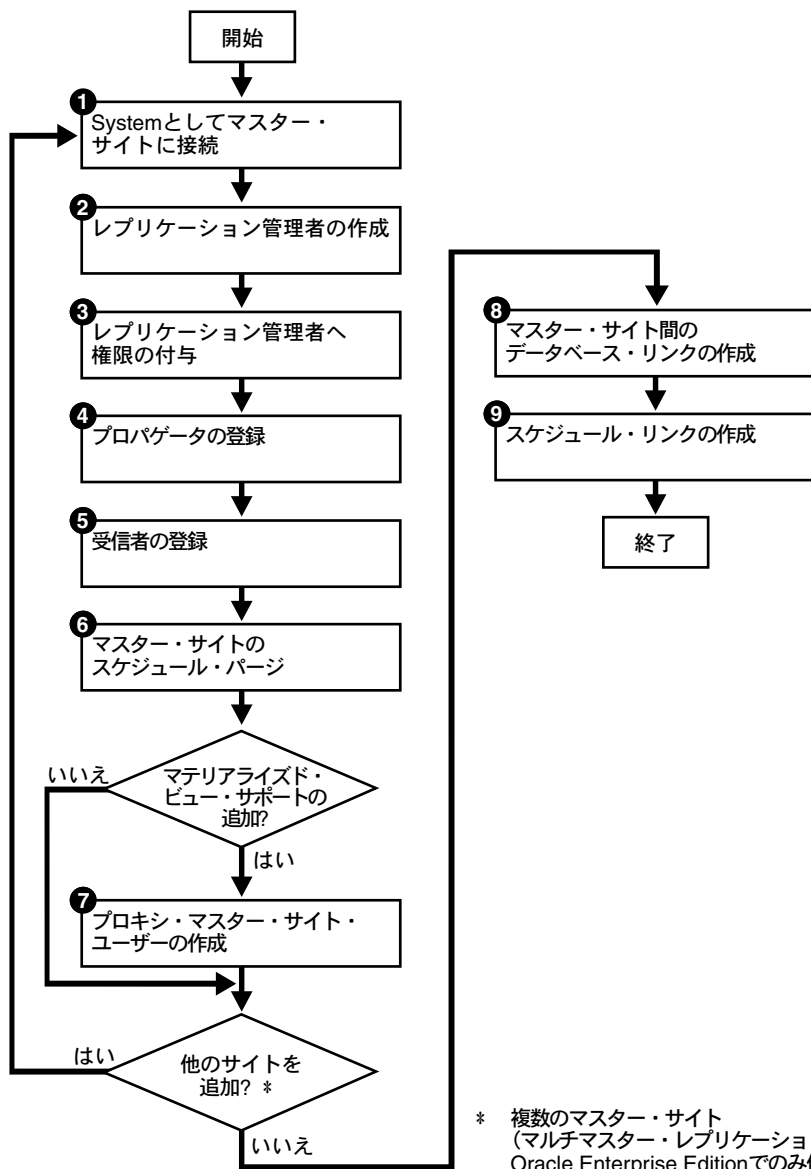
第 2 章～第 6 章では、図 2-1 に示すレプリケーション環境で作業します。この章では、指示に従ってこの環境を構築していきます。mv2.world は mv1.world マテリアライズド・ビューに基づいたマテリアライズド・ビューであり、複数層のマテリアライズド・ビュー環境を構築することに注意してください。図 2-1 の矢印はデータベース・リンクを表しています。

図 2-1 3 つのマスター・サイトと 2 つのマテリアライズド・ビュー・サイト



新しいマスター・サイトを構築する場合は図 2-2、新しいマテリアライズド・ビュー・サイトを構築する場合は図 2-3 の手順に従います。

図 2-2 マスター・サイトの設定



## マスター・サイトの設定

次の項では、サンプル・レプリケーション環境に `orc1.world`、`orc2.world` および `orc3.world` の3つのマスター・サイトを設定するために、段階的に指示が与えられます。

---

**注意：** このドキュメントをオンラインで参照している場合は、次の「BEGINNING OF SCRIPT」の行から 2-19 ページの「END OF SCRIPT」の行までのテキストをテキスト・エディタにコピーして編集し、使用している環境に適したスクリプトを作成します。

---

```
/***** BEGINNING OF SCRIPT *****/
```

### orc1.world の設定

次に示す手順に従って、`orc1.world` マスター・サイトを設定します。

**手順 1 SYSTEM として `orc1.world` のマスター・サイトに接続します。**

SYSTEM として、レプリケーションを設定するデータベースに接続します。`orc1.world` の設定を終えてから、2-8 ページの `orc2.world` サイトの手順 1 および 2-12 ページの `orc3.world` サイトの手順 1 を再開します。

```
*/

SET ECHO ON

SPOOL setup_masters.out

CONNECT system/manager@orc1.world

/*
```

**手順 2 `orc1.world` のレプリケーション管理者を作成します。**

レプリケーション管理者には、レプリケーション環境の構築および管理に必要な権限が付与されます。レプリケーション管理者は、レプリケーション環境に関係する各データベースごとに配置します。

```
*/

CREATE USER repadmin IDENTIFIED BY repadmin;

/*
```



**手順 3 次に示す手順に従って orc1.world のレプリケーション管理者に権限を付与します。**

- a. レプリケーション管理者に強力な権限を付与してレプリケーション環境を構築し、管理するには、GRANT\_ADMIN\_ANY\_SCHEMA プロシージャを実行します。

```
*/

BEGIN
    DBMS_REPCAT_ADMIN.GRANT_ADMIN_ANY_SCHEMA (
        username => 'repadmin');
END;
/

/*
```

- b. repadmin が任意のレプリケート表に対するマテリアライズド・ビュー・ログを作成できるようにするには、repadmin に対し COMMENT ANY TABLE および LOCK ANY TABLE を付与します。

```
*/

GRANT COMMENT ANY TABLE TO repadmin;
GRANT LOCK ANY TABLE TO repadmin;

/*
```

- c. repadmin がレプリケーション・マネージメント・ツールに接続できるようにするには、repadmin に対し SELECT ANY DICTIONARY を付与します。

```
*/

GRANT SELECT ANY DICTIONARY TO repadmin;

/*
```

**手順 4 orc1.world でプロパゲータを登録します。**

プロパゲータは、遅延トランザクション・キューを他のマスター・サイトに伝播します。

```
*/

BEGIN
    DBMS_DEFER_SYS.REGISTER_PROPAGATOR (
        username => 'repadmin');
END;
/

/*
```

**手順 5 orcl.world の受信者を登録します。**

受信者は、他のマスター・サイトからプロパゲータにより伝播された遅延トランザクションを受信します。

```
*/

BEGIN
    DBMS_REPCAT_ADMIN.REGISTER_USER_REPGROUP (
        username => 'repadmin',
        privilege_type => 'receiver',
        list_of_gnames => NULL);
END;
/

/*
```

**手順 6 マスター・サイト orcl.world をスケジュール・ページします。**

チェック中の遅延トランザクション・キューのサイズを一定に保つためには、完了した遅延トランザクションをページします。SCHEDULE\_PURGE プロシージャにより、ページ・プロセスが自動化されます。レプリケーション管理者としてこのプロシージャを実行します。

---

---

**注意：** 日付式は NEXT\_DATE および INTERVAL のパラメータとして使用します。次に例を示します。

- 現在の日付は SYSDATE と指定します。
  - 1 時間の間隔は SYSDATE + 1/24 と指定します。
  - 7 日間の間隔は SYSDATE + 7 と指定します。
- 
- 

```
*/

CONNECT repadmin/repadmin@orcl.world

BEGIN
    DBMS_DEFER_SYS.SCHEDULE_PURGE (
        next_date => SYSDATE,
        interval => 'SYSDATE + 1/24',
        delay_seconds => 0);
END;
/

/*
```

**関連項目：** 日付式の詳細は、『Oracle9i データベース管理者ガイド』および『Oracle9i SQL リファレンス』を参照してください。

**手順 7 このマスター・サイトに基づいてマテリアライズド・ビュー・サイトを構築する場合は、次に示す手順に従って、マテリアライズド・ビュー・サイトのユーザーに該当する orcl.world におけるプロキシ・マスター・サイト・ユーザーを作成します。**

- a. プロキシ・マテリアライズド・ビュー管理者を作成します。

プロキシ・マテリアライズド・ビュー管理者は、マテリアライズド・ビュー・サイトのマテリアライズド・ビュー管理者の代理としてターゲット・マスター・サイトでタスクを実行します。

\*/

```
CONNECT SYSTEM/MANAGER@orcl.world
```

```
CREATE USER proxy_mvviewadmin IDENTIFIED BY proxy_mvviewadmin;
```

```
BEGIN
```

```
    DBMS_REPCAT_ADMIN.REGISTER_USER_REPGROUP (
        username => 'proxy_mvviewadmin',
        privilege_type => 'proxy_snapadmin',
        list_of_gnames => NULL);
```

```
END;
```

```
/
```

```
-- Place GRANT SELECT_CATALOG_ROLE statement here if necessary.
```

/\*

マテリアライズド・ビュー・サイトでマテリアライズド・ビュー管理者が、レプリケーション・マネージメント・ツールを使用した管理運用を行うためには、proxy\_mvviewadmin に対し SELECT\_CATALOG\_ROLE を付与します。

```
GRANT SELECT_CATALOG_ROLE TO proxy_mvviewadmin;
```

レプリケーション・マネージメント・ツールを使用しない場合は、この権限を proxy\_mvviewadmin に対し付与する必要はありません。しかし、レプリケーション・マネージメント・ツールを使用する場合は、GRANT 文を、前行の REGISTER\_USER\_REPGROUP 文の直後の行に移動します。

**関連項目：** A-7 ページ「マテリアライズド・ビュー・レプリケーションのセキュリティ設定」

- b. プロキシ・リフレッシャを作成します。

プロキシ・リフレッシャはマスター・サイトで、マテリアライズド・ビュー・サイトのリフレッシャの代理としてタスクを実行します。

```
*/

CREATE USER proxy_refresher IDENTIFIED BY proxy_refresher;

GRANT CREATE SESSION TO proxy_refresher;
GRANT SELECT ANY TABLE TO proxy_refresher;

/*
```

## orc2.world の設定

次に示す手順に従って、orc2.world マスター・サイトを設定します。

### 手順 1 SYSTEM として orc2.world に接続します。

---

---

**注意：** 複数のマスター・サイト（マルチマスター・レプリケーション）は、Oracle Enterprise Edition でのみ利用できます。Oracle Enterprise Edition を使用していない場合は、2-20 ページの「[マテリアライズド・ビュー・サイトの設定](#)」をスキップします。

---

---

SYSTEM として、レプリケーションを設定するデータベースに接続します。orc2.world の設定後、2-12 ページのサイト orc3.world の手順 1 を開始します。

```
*/

CONNECT system/manager@orc2.world

/*
```

### 手順 2 orc2.world のレプリケーション管理者を作成します。

レプリケーション管理者には、レプリケーション環境の構築および管理に必要な権限が付与されます。レプリケーション管理者は、レプリケーション環境に関係する各データベースごとに配置します。

```
*/

CREATE USER repadmin IDENTIFIED BY repadmin;

/*
```

**手順 3 次に示す手順に従って orc2.world のレプリケーション管理者に権限を付与します。**

- a. レプリケーション管理者に強力な権限を付与してレプリケーション環境を構築し、管理するには、GRANT\_ADMIN\_ANY\_SCHEMA プロシージャを実行します。

```
*/

BEGIN
    DBMS_REPCAT_ADMIN.GRANT_ADMIN_ANY_SCHEMA (
        username => 'reppadmin');
END;
/

/*
```

- b. reppadmin が任意のレプリケート表に対するマテリアライズド・ビュー・ログを作成できるようにするには、reppadmin に対し COMMENT ANY TABLE および LOCK ANY TABLE 権限を付与します。

```
*/

GRANT COMMENT ANY TABLE TO reppadmin;
GRANT LOCK ANY TABLE TO reppadmin;

/*
```

- c. reppadmin がレプリケーション・マネージメント・ツールに接続できるようにするには、reppadmin に対し SELECT ANY DICTIONARY を付与します。

```
*/

GRANT SELECT ANY DICTIONARY TO reppadmin;

/*
```

**手順 4 orc2.world のプロパゲータを登録します。**

プロパゲータは、遅延トランザクション・キューを他のマスター・サイトに伝播します。

```
*/

BEGIN
    DBMS_DEFER_SYS.REGISTER_PROPAGATOR (
        username => 'reppadmin');
END;
/

/*
```

### 手順 5 orc2.world の受信者を登録します。

受信者は、他のマスター・サイトからプロパゲータにより伝播された遅延トランザクションを受信します。

```
*/

BEGIN
    DBMS_REPCAT_ADMIN.REGISTER_USER_REPGROUP (
        username => 'repadmin',
        privilege_type => 'receiver',
        list_of_gnames => NULL);
END;
/

/*
```

### 手順 6 マスター・サイト orc2.world をスケジュール・ページします。

チェック中の遅延トランザクション・キューのサイズを一定に保つためには、完了した遅延トランザクションをページします。SCHEDULE\_PURGE プロシージャにより、ページ・プロセスが自動化されます。レプリケーション管理者としてこのプロシージャを実行します。

```
*/

CONNECT repadmin/repadmin@orc2.world

BEGIN
    DBMS_DEFER_SYS.SCHEDULE_PURGE (
        next_date => SYSDATE,
        interval => 'SYSDATE + 1/24',
        delay_seconds => 0);
END;
/

/*
```

**手順 7 このマスター・サイトに基づいてマテリアライズド・ビュー・サイトを構築する場合は、次に示す手順に従って、マテリアライズド・ビュー・サイトのユーザーに該当する orc2.world におけるプロキシ・マスター・サイト・ユーザーを作成します。**

- a. プロキシ・マテリアライズド・ビュー管理者を作成します。

プロキシ・マテリアライズド・ビュー管理者は、マテリアライズド・ビュー・サイトのマテリアライズド・ビュー管理者の代理としてターゲット・マスター・サイトでタスクを実行します。

```
*/
```

```
CONNECT SYSTEM/MANAGER@orc2.world
```

```
CREATE USER proxy_mvviewadmin IDENTIFIED BY proxy_mvviewadmin;
```

```
BEGIN
```

```
    DBMS_REPCAT_ADMIN.REGISTER_USER_REPGROUP (
        username => 'proxy_mvviewadmin',
        privilege_type => 'proxy_snapadmin',
        list_of_gnames => NULL);
```

```
END;
```

```
/
```

```
-- Place GRANT SELECT_CATALOG_ROLE statement here if necessary.
```

```
/*
```

マテリアライズド・ビュー・サイトでマテリアライズド・ビュー管理者が、レプリケーション・マネージメント・ツールを使用した管理運用を行うためには、proxy\_mvviewadmin に対し SELECT\_CATALOG\_ROLE を付与します。

```
*/
```

```
GRANT SELECT_CATALOG_ROLE TO proxy_mvviewadmin;
```

```
/*
```

レプリケーション・マネージメント・ツールを使用しない場合は、この権限を proxy\_mvviewadmin に対し付与する必要はありません。しかし、レプリケーション・マネージメント・ツールを使用する場合は、GRANT 文を、前行の REGISTER\_USER\_REPGROUP 文の直後の行に移動します。

**関連項目：** A-7 ページ「マテリアライズド・ビュー・レプリケーションのセキュリティ設定」

- b. プロキシ・リフレッシャを作成します。

プロキシ・リフレッシャはマスター・サイトで、マテリアライズド・ビュー・サイトのリフレッシャの代理としてタスクを実行します。

```
*/  
  
CREATE USER proxy_refresher IDENTIFIED BY proxy_refresher;  
  
GRANT CREATE SESSION TO proxy_refresher;  
GRANT SELECT ANY TABLE TO proxy_refresher;  
  
/*
```

## orc3.world の設定

次に示す手順に従って、orc3.world マスター・サイトを設定します。

### 手順 1 SYSTEM として orc3.world に接続します。

---

---

**注意：** 複数のマスター・サイト（マルチマスター・レプリケーション）は、Oracle Enterprise Edition でのみ利用できます。Oracle Enterprise Edition を使用していない場合は、2-20 ページの「[マテリアライズド・ビュー・サイトの設定](#)」をスキップします。

---

---

SYSTEM として、レプリケーションを設定するデータベースに接続します。

```
*/  
  
CONNECT system/manager@orc3.world  
  
/*
```

### 手順 2 orc3.world のレプリケーション管理者を作成します。

レプリケーション管理者には、レプリケーション環境の構築および管理に必要な権限が付与されます。レプリケーション管理者は、レプリケーション環境に関係する各データベースごとに配置します。

```
*/  
  
CREATE USER repadmin IDENTIFIED BY repadmin;  
  
/*
```



**手順 3 次に示す手順に従って orc3.world のレプリケーション管理者に権限を付与します。**

- a. レプリケーション管理者に強力な権限を付与してレプリケーション環境を構築し、管理するには、GRANT\_ADMIN\_ANY\_SCHEMA プロシージャを実行します。

```
*/

BEGIN
    DBMS_REPCAT_ADMIN.GRANT_ADMIN_ANY_SCHEMA (
        username => 'reppadmin');
END;
/

/*
```

- b. reppadmin が任意のレプリケート表に対するマテリアライズド・ビュー・ログを作成できるようにするには、reppadmin に対し COMMENT ANY TABLE および LOCK ANY TABLE を付与します。

```
*/

GRANT COMMENT ANY TABLE TO reppadmin;
GRANT LOCK ANY TABLE TO reppadmin;

/*
```

- c. reppadmin がレプリケーション・マネージメント・ツールに接続できるようにするには、reppadmin に対し SELECT ANY DICTIONARY を付与します。

```
*/

GRANT SELECT ANY DICTIONARY TO reppadmin;

/*
```

**手順 4 orc3.world のプロパゲータを登録します。**

プロパゲータは、遅延トランザクション・キューを他のマスター・サイトに伝播します。

```
*/

BEGIN
    DBMS_DEFER_SYS.REGISTER_PROPAGATOR (
        username => 'reppadmin');
END;
/

/*
```

### 手順 5 orc3.world の受信者を登録します。

受信者は、他のマスター・サイトからプロパゲータにより伝播された遅延トランザクションを受信します。

```
*/

BEGIN
    DBMS_REPCAT_ADMIN.REGISTER_USER_REPGROUP (
        username => 'repadmin',
        privilege_type => 'receiver',
        list_of_gnames => NULL);
END;
/

/*
```

### 手順 6 マスター・サイト orc3.world をスケジュール・ページします。

チェック中の遅延トランザクション・キューのサイズを一定に保つためには、完了した遅延トランザクションをページします。SCHEDULE\_PURGE API により、ページ・プロセスが自動化されます。レプリケーション管理者としてこのプロシージャを実行します。

```
*/

CONNECT repadmin/repadmin@orc3.world

BEGIN
    DBMS_DEFER_SYS.SCHEDULE_PURGE (
        next_date => SYSDATE,
        interval => 'SYSDATE + 1/24',
        delay_seconds => 0);
END;
/

/*
```

**手順 7 このマスター・サイトに基づいてマテリアライズド・ビュー・サイトを構築する場合は、次に示す手順に従って、マテリアライズド・ビュー・サイトのユーザーに該当する orcl.world におけるプロキシ・マスター・サイト・ユーザーを作成します。**

- a. プロキシ・マテリアライズド・ビュー管理者を作成します。

プロキシ・マテリアライズド・ビュー管理者は、マテリアライズド・ビュー・サイトのマテリアライズド・ビュー管理者の代理としてターゲット・マスター・サイトでタスクを実行します。

```
*/
```

```
CONNECT SYSTEM/MANAGER@orc3.world
```

```
CREATE USER proxy_mvviewadmin IDENTIFIED BY proxy_mvviewadmin;
```

```
BEGIN
```

```
    DBMS_REPCAT_ADMIN.REGISTER_USER_REPGROUP (
        username => 'proxy_mvviewadmin',
        privilege_type => 'proxy_snapadmin',
        list_of_gnames => NULL);
```

```
END;
```

```
/
```

```
-- Place GRANT SELECT_CATALOG_ROLE statement here if necessary.
```

```
/*
```

マテリアライズド・ビュー・サイトでマテリアライズド・ビュー管理者が、レプリケーション・マネージメント・ツールを使用した管理運用を行うためには、proxy\_mvviewadmin に対し SELECT\_CATALOG\_ROLE を付与します。

```
*/
```

```
GRANT SELECT_CATALOG_ROLE TO proxy_mvviewadmin;
```

```
/*
```

レプリケーション・マネージメント・ツールを使用しない場合は、この権限を proxy\_mvviewadmin に対し付与する必要はありません。しかし、レプリケーション・マネージメント・ツールを使用する場合は、GRANT 文を、前行の REGISTER\_USER\_REPGROUP 文の直後の行に移動します。

**関連項目：** A-7 ページ「マテリアライズド・ビュー・レプリケーションのセキュリティ設定」

- b. プロキシ・リフレッシャを作成します。

プロキシ・リフレッシャはマスター・サイトで、マテリアライズド・ビュー・サイトのリフレッシャの代理としてタスクを実行します。

\*/

```
CREATE USER proxy_refresher IDENTIFIED BY proxy_refresher;
```

```
GRANT CREATE SESSION TO proxy_refresher;
```

```
GRANT SELECT ANY TABLE TO proxy_refresher;
```

/\*

## マスター・サイト間のスケジュール・リンクの作成

次に示す手順に従って、マスター・サイト間のスケジュール・リンクを作成します。

### 手順1 マスター・サイト間でデータベース・リンクを作成します。

データベース・リンクは必要な配布メカニズムを備えていて、複数のレプリケーション・サイト間でデータをレプリケートできます。プライベート・データベース・リンクを作成する前に、各プライベート・データベース・リンクが使用するパブリック・データベース・リンクを作成する必要があります。次に、設定した各マスター・サイトのすべてのレプリケーション管理者間のデータベース・リンクを作成します。

**関連項目：** データベース・リンクの詳細は、『Oracle9i データベース管理者ガイド』を参照してください。

\*/

```
CONNECT SYSTEM/MANAGER@orc1.world
```

```
CREATE PUBLIC DATABASE LINK orc2.world USING 'orc2.world';
```

```
CREATE PUBLIC DATABASE LINK orc3.world USING 'orc3.world';
```

```
CONNECT repadmin/repadmin@orc1.world
```

```
CREATE DATABASE LINK orc2.world CONNECT TO repadmin IDENTIFIED BY repadmin;
```

```
CREATE DATABASE LINK orc3.world CONNECT TO repadmin IDENTIFIED BY repadmin;
```

```
CONNECT SYSTEM/MANAGER@orc2.world
```

```
CREATE PUBLIC DATABASE LINK orc1.world USING 'orc1.world';
```

```
CREATE PUBLIC DATABASE LINK orc3.world USING 'orc3.world';
```

```
CONNECT repadmin/repadmin@orc2.world
```

```
CREATE DATABASE LINK orc1.world CONNECT TO repadmin IDENTIFIED BY repadmin;
```

```
CREATE DATABASE LINK orc3.world CONNECT TO repadmin IDENTIFIED BY repadmin;
```

```
CONNECT SYSTEM/MANAGER@orc3.world
CREATE PUBLIC DATABASE LINK orc1.world USING 'orc1.world';
CREATE PUBLIC DATABASE LINK orc2.world USING 'orc2.world';

CONNECT repadmin/repadmin@orc3.world
CREATE DATABASE LINK orc1.world CONNECT TO repadmin IDENTIFIED BY repadmin;
CREATE DATABASE LINK orc2.world CONNECT TO repadmin IDENTIFIED BY repadmin;

/*
```

## 手順 2 各データベース・リンクのスケジュールを定義してスケジュール・リンクを作成します。

SCHEDULE\_PUSH プロシージャを実行するときは、データベース・リンクを定義してスケジュール・リンクを作成します。スケジュール・リンクにより、遅延トランザクション・キューが他のマスター・サイトのそれぞれに伝播される回数が決まります。手順 1 で作成した各データベース・リンクごとに SCHEDULE\_PUSH プロシージャを実行します。データベース・リンクは、SCHEDULE\_PUSH プロシージャの接続先パラメータで指定します。

Oracle の非同期レプリケーション・メカニズムを使用する場合でも、スケジュール・リンクを構成して、連続的、リアルタイムなレプリケーションをシミュレートできます。この例におけるスケジュール・リンクでは、連続的なレプリケーションをシミュレートします。

**関連項目：** 連続的レプリケーションのシミュレーションの詳細情報は、『Oracle9i アドバンスド・レプリケーション』を参照してください。

```
*/

CONNECT repadmin/repadmin@orc1.world

BEGIN
  DBMS_DEFER_SYS.SCHEDULE_PUSH (
    destination => 'orc2.world',
    interval => 'SYSDATE + (1/144)',
    next_date => SYSDATE,
    parallelism => 1,
    execution_seconds => 1500,
    delay_seconds => 1200);
END;
/

BEGIN
  DBMS_DEFER_SYS.SCHEDULE_PUSH (
    destination => 'orc2.world',
    interval => 'SYSDATE + (1/144)',
    next_date => SYSDATE,
    parallelism => 1,
```

```
        execution_seconds => 1500,
        delay_seconds => 1200);
END;
/

BEGIN
DBMS_DEFER_SYS.SCHEDULE_PUSH (
    destination => 'orc3.world',
    interval => 'SYSDATE + (1/144)',
    next_date => SYSDATE,
    parallelism => 1,
    execution_seconds => 1500,
    delay_seconds => 1200);
END;
/

CONNECT repadmin/repadmin@orc2.world

BEGIN
    DBMS_DEFER_SYS.SCHEDULE_PUSH (
        destination => 'orc1.world',
        interval => 'SYSDATE + (1/144)',
        next_date => SYSDATE,
        parallelism => 1,
        execution_seconds => 1500,
        delay_seconds => 1200);
END;
/

BEGIN
    DBMS_DEFER_SYS.SCHEDULE_PUSH (
        destination => 'orc3.world',
        interval => 'SYSDATE + (1/144)',
        next_date => SYSDATE,
        parallelism => 1,
        execution_seconds => 1500,
        delay_seconds => 1200);
END;
/

CONNECT repadmin/repadmin@orc3.world

BEGIN
    DBMS_DEFER_SYS.SCHEDULE_PUSH (
        destination => 'orc1.world',
        interval => 'SYSDATE + (1/144)',
        next_date => SYSDATE,
```

```

        parallelism => 1,
        execution_seconds => 1500,
        delay_seconds => 1200);
END;
/

BEGIN
    DBMS_DEFER_SYS.SCHEDULE_PUSH (
        destination => 'orc2.world',
        interval => 'SYSDATE + (1/144)',
        next_date => SYSDATE,
        parallelism => 1,
        execution_seconds => 1500,
        delay_seconds => 1200);
END;
/

SET ECHO OFF

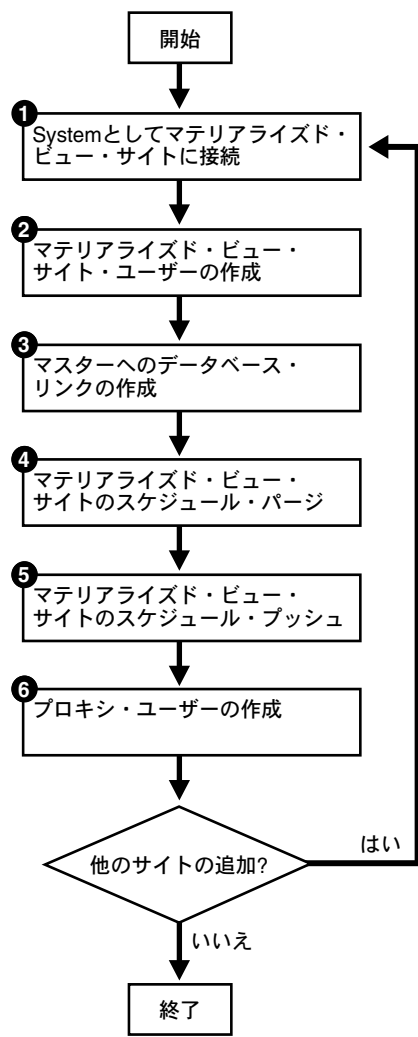
SPOOL OFF

/*****END OF SCRIPT*****/

```

# マテリアライズド・ビュー・サイトの設定

図 2-3 マテリアライズド・ビュー・サイトの設定





## mv1.world の設定

次に示す手順に従って、mv1.world マスター・マテリアライズド・ビュー・サイトを設定します。mv1.world は mv2.world の基準となるため、マスター・マテリアライズド・ビュー・サイトです。

---

---

**注意：** このドキュメントをオンラインで参照している場合は、次の「BEGINNING OF SCRIPT」の行から 2-31 ページの「END OF SCRIPT」の行までのテキストをテキスト・エディタにコピーして編集し、使用している環境に適したスクリプトを作成します。

---

---

```
/***** BEGINNING OF SCRIPT *****/
```

### 手順 1 SYSTEM として mv1.world のマテリアライズド・ビュー・サイトに接続します。

SYSTEM として、マテリアライズド・ビュー・サイトとして設定するデータベースに接続します。

```
*/
```

```
SET ECHO ON
```

```
SPOOL setup_mvs.out
```

```
CONNECT SYSTEM/MANAGER@mv1.world
```

```
/*
```

### 手順 2 mv1.world にマテリアライズド・ビュー・サイトのユーザーを作成します。

マテリアライズド・ビュー・サイトには複数のユーザーを作成します。必要なユーザーは、次のとおりです。

- マテリアライズド・ビュー管理者
- プロパゲータ
- リフレッシャ
- 受信者（サイトが mv1.world のように、他のマテリアライズド・ビューのマスター・マテリアライズド・ビュー・サイトとして機能する場合）

次に示す手順に従って、これらのユーザーを作成します。

- a.** マテリアライズド・ビュー管理者を作成します。

マテリアライズド・ビュー管理者は、マテリアライズド・ビュー・サイトの作成と管理を行います。マテリアライズド・ビュー管理者に適切な権限を付与するには、GRANT\_ADMIN\_ANY\_SCHEMA プロシージャを実行します。

```
*/

CREATE USER mviewadmin IDENTIFIED BY mviewadmin;

BEGIN
    DBMS_REPCAT_ADMIN.GRANT_ADMIN_ANY_SCHEMA (
        username => 'mviewadmin');
END;
/

GRANT COMMENT ANY TABLE TO mviewadmin;

GRANT LOCK ANY TABLE TO mviewadmin;

/*
```

- b.** mviewadmin がレプリケーション・マネージメント・ツールに接続できるようにするには、mviewadmin に対し SELECT ANY DICTIONARY を付与します。

```
*/

GRANT SELECT ANY DICTIONARY TO mviewadmin;

/*
```

- c.** プロパゲータを作成します。

プロパゲータは、遅延トランザクション・キューをターゲット・マスター・サイトに伝播します。

```
*/

CREATE USER propagator IDENTIFIED BY propagator;

BEGIN
    DBMS_DEFER_SYS.REGISTER_PROPAGATOR (
        username => 'propagator');
END;
/

/*
```

d. リフレッシャを作成します。

リフレッシャは、マテリアライズド・ビュー・サイトに対するターゲット・マスター・サイトのレプリケート表に対する変更の取込みを行います。このユーザーは、1 つ以上のマテリアライズド・ビューをリフレッシュします。mviewadmin ユーザーをリフレッシャとする場合、この手順は必須ではありません。

```
*/

CREATE USER refresher IDENTIFIED BY refresher;

GRANT CREATE SESSION TO refresher;

GRANT ALTER ANY MATERIALIZED VIEW TO refresher;

/*
```

e. 受信者を登録します。

受信者は、マテリアライズド・ビュー・サイトからプロパゲータにより伝播された遅延トランザクションを受信します。サイトが、他のマテリアライズド・ビュー・サイトのマスター・マテリアライズド・ビュー・サイトとして機能する場合のみ、受信者が必要です。

```
*/

BEGIN
    DBMS_REPCAT_ADMIN.REGISTER_USER_REPGROUP (
        username => 'mviewadmin',
        privilege_type => 'receiver',
        list_of_gnames => NULL);
END;
/

/*
```

**手順 3 次の手順に従って、マスター・サイトへのデータベース・リンクを作成します。**

- a. パブリック・データベース・リンクを作成します。

```
*/  
  
CONNECT SYSTEM/MANAGER@mv1.world  
  
CREATE PUBLIC DATABASE LINK orcl.world USING 'orcl.world';  
  
/*
```

- b. マテリアライズド・ビュー管理者のデータベース・リンクを作成します。

マテリアライズド・ビュー・サイトのマテリアライズド・ビュー管理者からマスター・サイトのプロキシ・マテリアライズド・ビュー管理者に至るデータベース・リンクを作成します。

```
*/  
  
CONNECT mviewadmin/mviewadmin@mv1.world;  
  
CREATE DATABASE LINK orcl.world  
  CONNECT TO proxy_mviewadmin IDENTIFIED BY proxy_mviewadmin;  
  
/*
```

- c. プロパゲータ受信者のデータベース・リンクを作成します。

マテリアライズド・ビュー・サイトのプロパゲータからマスター・サイトの受信者に至るデータベース・リンクを作成します。マスター・サイトの作成時に受信者は定義されています。

```
*/  
  
CONNECT propagator/propagator@mv1.world  
  
CREATE DATABASE LINK orcl.world  
  CONNECT TO repadmin IDENTIFIED BY repadmin;  
  
/*
```

**関連項目：** 2-10 ページの **手順 5**

**手順 4 mv1.world マテリアライズド・ビュー・サイトをスケジュール・パージします。**

チェック中の遅延トランザクション・キューのサイズを一定に保つためには、完了した遅延トランザクションをパージします。SCHEDULE\_PURGE プロシージャにより、パージ・プロセスが自動化されます。マテリアライズド・ビュー・サイトに含まれるマテリアライズド・ビューが、読取り専用のみの場合は、このプロシージャを実行する必要はありません。

```
*/

CONNECT mviewadmin/mviewadmin@mv1.world

BEGIN
    DBMS_DEFER_SYS.SCHEDULE_PURGE (
        next_date => SYSDATE,
        interval => 'SYSDATE + 1/24',
        delay_seconds => 0,
        rollback_segment => '');
END;
/

/*
```

**手順 5 マテリアライズド・ビュー・サイトがマスター・サイトと常時接続されている場合は、mv1.world マテリアライズド・ビュー・サイトでスケジュール・プッシュのオプションを選択できます。マテリアライズド・ビュー・サイトがマスター・サイトと長期間接続されていない場合は、スケジュール・プッシュは使用しないで必要に応じてリフレッシュを行ってください。スケジュール・プッシュはマスター・サイトを変更します。**

SCHEDULE\_PUSH プロシージャは、遅延トランザクション・キューをターゲット・マスター・サイトに伝播する時期を計画します。

```
*/

CONNECT mviewadmin/mviewadmin@mv1.world

BEGIN
    DBMS_DEFER_SYS.SCHEDULE_PUSH (
        destination => 'orc1.world',
        interval => 'SYSDATE + 1/24',
        next_date => SYSDATE,
        stop_on_error => FALSE,
        delay_seconds => 0,
        parallelism => 0);
END;
/

/*
```

**手順 6 次の手順に従って、mv1.world マテリアライズド・ビュー・サイトのプロキシ・ユーザーを作成します。**

- a. プロキシ・マテリアライズド・ビュー管理者を作成します。

プロキシ・マテリアライズド・ビュー管理者は、このマテリアライズド・ビュー・サイトに基いて、マテリアライズド・ビュー・サイトのマテリアライズド・ビュー管理者にかわってターゲット・マスター・マテリアライズド・ビュー・サイトのタスクを実行します。サイトが、他のマテリアライズド・ビュー・サイトのマスター・マテリアライズド・ビュー・サイトとして機能しない場合、このユーザーは必須ではありません。

\*/

```
CONNECT SYSTEM/MANAGER@mv1.world
```

```
CREATE USER proxy_mviewadmin IDENTIFIED BY proxy_mviewadmin;
```

```
BEGIN
```

```
    DBMS_REPCAT_ADMIN.REGISTER_USER_REPGROUP (  
        username => 'proxy_mviewadmin',  
        privilege_type => 'proxy_snapadmin',  
        list_of_gnames => NULL);
```

```
END;
```

```
/
```

```
-- Place GRANT SELECT_CATALOG_ROLE statement here if necessary.
```

/\*

このマテリアライズド・ビュー・サイトに基いたマテリアライズド・ビュー・サイトのマテリアライズド・ビュー管理者が、レプリケーション・マネージメント・ツールを使用した管理運用を行うためには、proxy\_mviewadmin に対し SELECT\_CATALOG\_ROLE を付与します。

```
GRANT SELECT_CATALOG_ROLE TO proxy_mviewadmin;
```

レプリケーション・マネージメント・ツールを使用しない場合は、この権限を proxy\_mviewadmin に対し付与する必要はありません。しかし、レプリケーション・マネージメント・ツールを使用する場合は、GRANT 文を、前行の REGISTER\_USER\_REPGROUP 文の直後の行に移動します。

- b. プロキシ・リフレッシャを作成します。

プロキシ・リフレッシャは、このマテリアライズド・ビュー・サイトに基づいたマテリアライズド・ビュー・サイトのリフレッシャのかわりに、マスター・マテリアライズド・ビュー・サイトのタスクを実行します。サイトが、他のマテリアライズド・ビュー・サイトのマスター・マテリアライズド・ビュー・サイトとして機能しない場合、このユーザーは必須ではありません。

```
*/
```

```
CREATE USER proxy_refresher IDENTIFIED BY proxy_refresher;
```

```
GRANT CREATE SESSION TO proxy_refresher;
```

```
GRANT SELECT ANY TABLE TO proxy_refresher;
```

```
/*
```

**関連項目：** A-7 ページ「マテリアライズド・ビュー・レプリケーションのセキュリティ設定」

## mv2.world の設定

次の手順に従って、mv2.world マテリアライズド・ビュー・サイトを設定します。

mv2.world は、mv1.world、その他のマテリアライズド・ビューに基づいているため、複数層マテリアライズド・ビュー構成の一部です。

**手順 1** SYSTEM としてレベル 2 マテリアライズド・ビュー・サイト mv2.world に接続します。

SYSTEM として、レベル 2 マテリアライズド・ビュー・サイトとして設定するデータベースに接続します。このサイト、mv2.world は、mv1.world に基づいたマテリアライズド・ビュー・サイトになります。

```
*/
```

```
CONNECT SYSTEM/MANAGER@mv2.world
```

```
/*
```

### 手順 2 mv2.world のレベル 2 マテリアライズド・ビュー・サイトのユーザーを作成します。

レベル 2 マテリアライズド・ビュー・サイトには複数のユーザーを作成します。必要なユーザーは、次のとおりです。

- マテリアライズド・ビュー管理者
- プロパゲータ
- リフレッシャ

次に示す手順に従って、これらのユーザーを作成します。

#### a. マテリアライズド・ビュー管理者を作成します。

マテリアライズド・ビュー管理者は、レベル 2 マテリアライズド・ビュー・サイトの作成と管理を行います。マテリアライズド・ビュー管理者に適切な権限を付与するには、GRANT\_ADMIN\_ANY\_SCHEMA プロシージャを実行します。

```
*/  
  
CREATE USER mviewadmin IDENTIFIED BY mviewadmin;  
  
BEGIN  
    DBMS_REPCAT_ADMIN.GRANT_ADMIN_ANY_SCHEMA (  
        username => 'mviewadmin');  
END;  
/  
  
/*
```

#### b. mviewadmin がレプリケーション・マネージメント・ツールに接続できるようにするには、mviewadmin に対し SELECT ANY DICTIONARY を付与します。

```
*/  
  
GRANT SELECT ANY DICTIONARY TO mviewadmin;  
  
/*
```

#### c. プロパゲータを作成します。

プロパゲータは、遅延トランザクション・キューをターゲット・マスター・マテリアライズド・ビュー・サイトに伝播します。

```
*/  
  
CREATE USER propagator IDENTIFIED BY propagator;  
  
BEGIN
```



```

DBMS_DEFER_SYS.REGISTER_PROPAGATOR (
    username => 'propagator');
END;
/

/*

```

- d. リフレッシャを作成します。

リフレッシャは、レベル 2 マテリアライズド・ビュー・サイトに対するターゲット・マスター・マテリアライズド・ビュー・サイトのレプリケート・マテリアライズド・ビューに対する変更の取込みを行います。

```

*/

CREATE USER refresher IDENTIFIED BY refresher;

GRANT CREATE SESSION TO refresher;
GRANT ALTER ANY MATERIALIZED VIEW TO refresher;

/*

```

**手順 3 次の手順に従って、マスター・マテリアライズド・ビュー・サイトへのデータベース・リンクを作成します。**

- a. パブリック・データベース・リンクを作成します。

```

*/

CONNECT SYSTEM/MANAGER@mv2.world

CREATE PUBLIC DATABASE LINK mv1.world USING 'mv1.world';

/*

```

- b. マテリアライズド・ビュー管理者のデータベース・リンクを作成します。

レベル 2 マテリアライズド・ビュー・サイトのマテリアライズド・ビュー管理者からマスター・サイトのプロキシ・マテリアライズド・ビュー管理者に至るデータベース・リンクを作成します。

```

*/

CONNECT mviewadmin/mviewadmin@mv2.world;

CREATE DATABASE LINK mv1.world
    CONNECT TO proxy_mviewadmin IDENTIFIED BY proxy_mviewadmin;

/*

```

- c. プロパゲータ受信者のデータベース・リンクを作成します。

レベル 2 マテリアライズド・ビュー・サイトのプロパゲータからマスター・マテリアライズド・ビュー・サイトの受信者に至るデータベース・リンクを作成します。マスター・マテリアライズド・ビュー・サイトの作成時に受信者は定義されています。

\*/

```
CONNECT propagator/propagator@mv2.world
```

```
CREATE DATABASE LINK mv1.world
```

```
CONNECT TO mviewadmin IDENTIFIED BY mviewadmin;
```

/\*

### 手順 4 mv2.world のレベル 2 マテリアライズド・ビュー・サイトをスケジュール・パージします。

チェック中の遅延トランザクション・キューのサイズを一定に保つためには、完了した遅延トランザクションをパージします。SCHEDULE\_PURGE プロシージャにより、パージ・プロセスが自動化されます。レベル 2 マテリアライズド・ビュー・サイトに含まれるマテリアライズド・ビューが、読取り専用のみの場合は、このプロシージャを実行する必要はありません。

\*/

```
CONNECT mviewadmin/mviewadmin@mv2.world
```

```
BEGIN
```

```
  DBMS_DEFER_SYS.SCHEDULE_PURGE (  
    next_date => SYSDATE,  
    interval => 'SYSDATE + 1/24',  
    delay_seconds => 0,  
    rollback_segment => '');
```

```
END;
```

```
/
```

/\*

**手順 5 マテリアライズド・ビュー・サイトがマスター・マテリアライズド・ビュー・サイトと常時接続されている場合は、mv2.world マテリアライズド・ビュー・サイトでスケジュール・プッシュのオプションを選択できます。マテリアライズド・ビュー・サイトがマスター・マテリアライズド・ビュー・サイトと長期間接続されていない場合は、スケジュール・プッシュは使用しないで必要に応じてリフレッシュを行ってください。スケジュール・プッシュはマスター・マテリアライズド・ビュー・サイトを変更します。**

SCHEDULE\_PUSH プロシージャは、遅延トランザクション・キューをターゲット・マスター・マテリアライズド・ビュー・サイトに伝播する時期を計画します。

```
*/  
  
CONNECT mvviewadmin/mviewadmin@mv2.world  
  
BEGIN  
  DBMS_DEFER_SYS.SCHEDULE_PUSH (  
    destination => 'mv1.world',  
    interval => 'SYSDATE + 1/24',  
    next_date => SYSDATE,  
    stop_on_error => FALSE,  
    delay_seconds => 0,  
    parallelism => 0);  
END;  
/  
  
SET ECHO OFF  
  
SPOOL OFF  
  
/***** END OF SCRIPT *****/
```



---

## マスター・グループの作成

この章では、マスター・レプリケーション・サイトでマスター・グループを作成する方法を説明します。この章では、次の項目を説明します。

- マスター・グループ作成の概要
- マスター・グループの作成

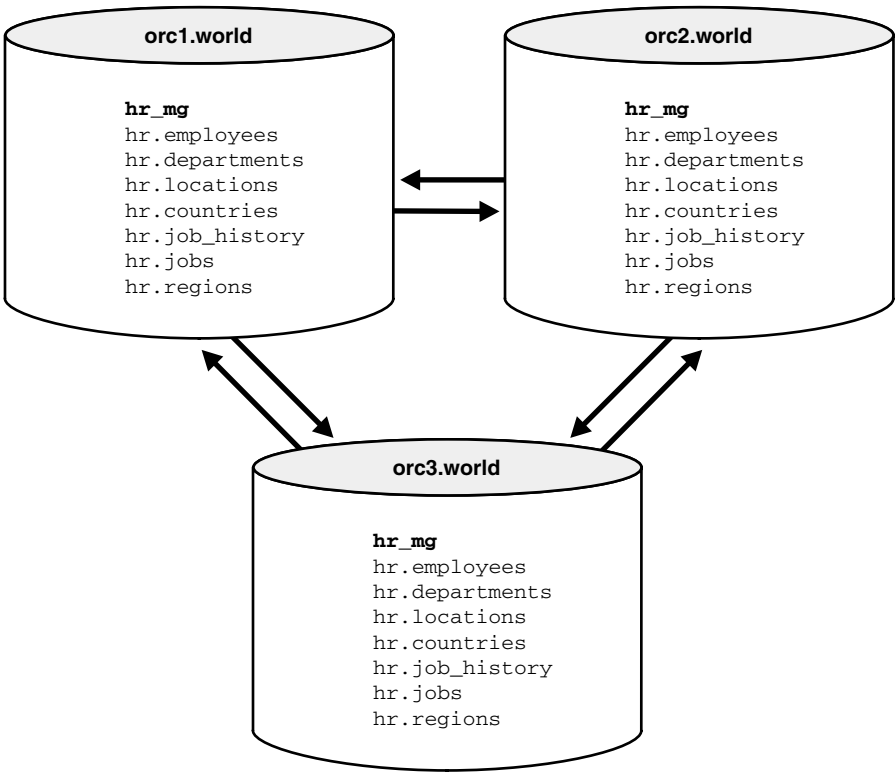
## マスター・グループ作成の概要

マスター・サイトを設定した後、マスター・グループを作成できます。レプリケーション環境を正しく構築するには、[図 3-2](#) に示す手順に従う必要があります。

**関連項目：** マスター・サイトの設定の詳細は、2-1 ページの「[レプリケーション・サイトの構築](#)」を参照してください。

この章では、hr\_repg マスター・グループを作成して、[図 3-1](#) に示すオブジェクトをレプリケートします。

図 3-1 全サイト間の hr スキーマの表をレプリケート



## 作業を開始する前に

この章のスクリプトを設計どおりに動作させるために、`orc1.world`、`orc2.world` および `orc3.world` に `hr` スキーマが存在することを確認してください。 `hr` スキーマには次のデータベース・オブジェクトが含まれます。

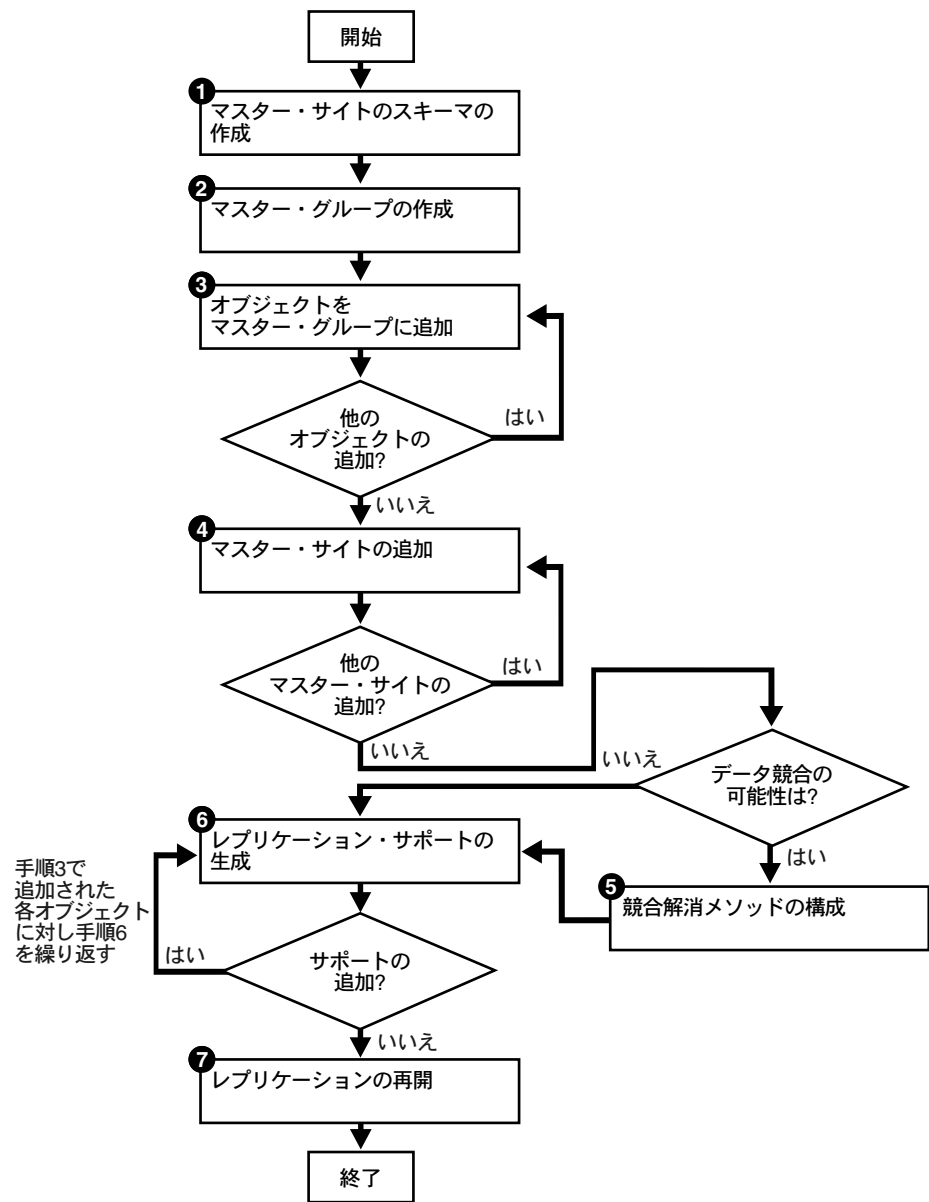
- `countries` 表
- `departments` 表
- `employees` 表
- `jobs` 表
- `job_history` 表
- `locations` 表
- `regions` 表
- `dept_location_ix` 索引
- `emp_department_ix` 索引
- `emp_job_ix` 索引
- `emp_manager_ix` 索引
- `jhist_department_ix` 索引
- `jhist_employee_ix` 索引
- `jhist_job_ix` 索引
- `loc_country_ix` 索引

リストされている索引は、`hr` スキーマの外部キー列を基準とした索引です。親表の更新や削除が許可されていない場合を除き、外部キー参照制約がある表のレプリケーションでは、常に外部キーに索引を作成してからこれらの索引をレプリケートすることをお勧めします。索引は自動でレプリケートされません。

デフォルトでは、**Oracle9i** のインストール時に、`hr` スキーマが自動的にインストールされます。この章のスクリプト例では、すべてのマスター・サイトに `hr` スキーマが存在し、そのスキーマには各サイトのデータベース・オブジェクトのすべてが含まれることを前提としています。また、表には、**Oracle** のインストール時に自動的に挿入されたデータが含まれるものとします。`hr` スキーマがレプリケーション・サイトにインストールされていない場合は、手動でインストールできます。

**関連項目：** `hr` スキーマやその他のサンプル・スキーマの情報およびサンプル・スキーマの手動インストールの情報は、『**Oracle9i** サンプル・スキーマ』を参照してください。

図 3-2 マスター・グループの作成





# マスター・グループの作成

次に示す手順に従って、hr\_repg マスター・グループを作成します。

**注意：** このドキュメントをオンラインで参照している場合は、次の「BEGINNING OF SCRIPT」の行から 3-14 ページの「END OF SCRIPT」の行までのテキストをテキスト・エディタにコピーして編集し、使用している環境に適したスクリプトを作成します。

```
/***** BEGINNING OF SCRIPT *****/
```

## 手順 1 マスター・サイトでスキーマを作成します。

マスター・グループに関係するすべてのマスター・サイトに、スキーマがまだ存在しない場合は、スキーマを作成して、必要な権限を付与します。この例では、hr スキーマを使用します。このスキーマは、Oracle のインストール時にデフォルトでインストールされたスキーマの 1 つです。したがって、hr スキーマはすべてのマスター・サイトに存在します。

```
*/

SET ECHO ON

SPOOL create_mg.out

PAUSE Press <RETURN> to continue when the schema exists at all master sites.

/*
```

## 手順 2 マスター・グループを作成します。

新しいマスター・グループを定義するには、CREATE\_MASTER\_REPGROUP プロシージャを使用します。オブジェクトをマスター・グループに追加する場合または他のレプリケーション管理タスクを実行する場合は、この手順で定義したマスター・グループ名を参照します。この手順はレプリケーション管理者が実行します。

```
*/

CONNECT repadmin/repadmin@orc1.world

BEGIN
    DBMS_REPCAT.CREATE_MASTER_REPGROUP (
        gname => 'hr_repg');
END;

/*
```

### 手順3 マスター・グループにオブジェクトを追加します。

オブジェクトをマスター・グループに追加するには、CREATE\_MASTER\_REOBJECT を使用します。多くの場合、表と索引をマスター・グループへ追加しますが、プロシージャ、ビューその他も追加できます。

```
*/

BEGIN
    DBMS_REPCAT.CREATE_MASTER_REOBJECT (
        gname => 'hr_repg',
        type => 'TABLE',
        oname => 'countries',
        sname => 'hr',
        use_existing_object => TRUE,
        copy_rows => FALSE);
END;
/

BEGIN
    DBMS_REPCAT.CREATE_MASTER_REOBJECT (
        gname => 'hr_repg',
        type => 'TABLE',
        oname => 'departments',
        sname => 'hr',
        use_existing_object => TRUE,
        copy_rows => FALSE);
END;
/

BEGIN
    DBMS_REPCAT.CREATE_MASTER_REOBJECT (
        gname => 'hr_repg',
        type => 'TABLE',
        oname => 'employees',
        sname => 'hr',
        use_existing_object => TRUE,
        copy_rows => FALSE);
END;
/

BEGIN
    DBMS_REPCAT.CREATE_MASTER_REOBJECT (
        gname => 'hr_repg',
        type => 'TABLE',
        oname => 'jobs',
        sname => 'hr',
```

```

        use_existing_object => TRUE,
        copy_rows => FALSE);
END;
/

BEGIN
    DBMS_REPCAT.CREATE_MASTER_REPOBJECT (
        gname => 'hr_repg',
        type => 'TABLE',
        oname => 'job_history',
        sname => 'hr',
        use_existing_object => TRUE,
        copy_rows => FALSE);
END;
/

BEGIN
    DBMS_REPCAT.CREATE_MASTER_REPOBJECT (
        gname => 'hr_repg',
        type => 'TABLE',
        oname => 'locations',
        sname => 'hr',
        use_existing_object => TRUE,
        copy_rows => FALSE);
END;
/

BEGIN
    DBMS_REPCAT.CREATE_MASTER_REPOBJECT (
        gname => 'hr_repg',
        type => 'TABLE',
        oname => 'regions',
        sname => 'hr',
        use_existing_object => TRUE,
        copy_rows => FALSE);
END;
/

BEGIN
    DBMS_REPCAT.CREATE_MASTER_REPOBJECT (
        gname => 'hr_repg',
        type => 'INDEX',
        oname => 'dept_location_ix',
        sname => 'hr',
        use_existing_object => TRUE,
        copy_rows => FALSE);
END;
/

```

```
BEGIN
    DBMS_REPCAT.CREATE_MASTER_REPOBJECT (
        gname => 'hr_repg',
        type => 'INDEX',
        oname => 'emp_department_ix',
        sname => 'hr',
        use_existing_object => TRUE,
        copy_rows => FALSE);
END;
/

BEGIN
    DBMS_REPCAT.CREATE_MASTER_REPOBJECT (
        gname => 'hr_repg',
        type => 'INDEX',
        oname => 'emp_job_ix',
        sname => 'hr',
        use_existing_object => TRUE,
        copy_rows => FALSE);
END;
/

BEGIN
    DBMS_REPCAT.CREATE_MASTER_REPOBJECT (
        gname => 'hr_repg',
        type => 'INDEX',
        oname => 'emp_manager_ix',
        sname => 'hr',
        use_existing_object => TRUE,
        copy_rows => FALSE);
END;
/

BEGIN
    DBMS_REPCAT.CREATE_MASTER_REPOBJECT (
        gname => 'hr_repg',
        type => 'INDEX',
        oname => 'jhist_department_ix',
        sname => 'hr',
        use_existing_object => TRUE,
        copy_rows => FALSE);
END;
/
```

```

BEGIN
    DBMS_REPCAT.CREATE_MASTER_REPOBJECT (
        gname => 'hr_repg',
        type => 'INDEX',
        oname => 'jhist_employee_ix',
        sname => 'hr',
        use_existing_object => TRUE,
        copy_rows => FALSE);
END;
/

BEGIN
    DBMS_REPCAT.CREATE_MASTER_REPOBJECT (
        gname => 'hr_repg',
        type => 'INDEX',
        oname => 'jhist_job_ix',
        sname => 'hr',
        use_existing_object => TRUE,
        copy_rows => FALSE);
END;
/

BEGIN
    DBMS_REPCAT.CREATE_MASTER_REPOBJECT (
        gname => 'hr_repg',
        type => 'INDEX',
        oname => 'loc_country_ix',
        sname => 'hr',
        use_existing_object => TRUE,
        copy_rows => FALSE);
END;
/

/*

```

#### 手順 4 マスター・サイトを追加します。

マスター定義サイト（デフォルトではマスター・グループが作成されたサイトがマスター定義サイトになります）でマスター・グループを定義すれば、レプリケーション環境に関する他のサイトを定義できます。orc2.world および orc3.world サイトをレプリケーション環境に追加する場合があります。この例では、すべてのマスター・サイトにマスター・グループを作成します。ただし、データベースを静止することなく、まず 1 つのマスター・サイトにマスター・グループを作成して、後でマスター・サイトを追加することもできます。この場合には、この手順をスキップします。

**関連項目：** 詳細は、7-4 ページの「[マスター・グループを静止することなく、新しいマスター・サイトを追加](#)」を参照してください。

この例では、hr スキーマがすべてのマスター・サイトに存在することが前提となっているため、ADD\_MASTER\_DATABASE プロシージャの use\_existing\_objects パラメータが TRUE に設定されています。つまり、hr スキーマのオブジェクトは、すべてのマスター・サイトで事前に作成されていることが前提となっています。また、各マスター・サイトの表には同一データが格納されていることが前提となるため、copy\_rows パラメータは FALSE に設定されています。

---

**注意：** 循環依存のある表または自己参照型制約のある表が含まれるマスター・グループにマスター・サイトを追加する場合は、表の定義を事前に作成し、新しいマスター・サイトに手動でデータをロードします。循環依存の例を次に示します。表 A には表 B に対する外部キー制約があり、表 B には表 A に対する外部キー制約があります。

---

```
*/

BEGIN
    DBMS_REPCAT.ADD_MASTER_DATABASE (
        gname => 'hr_repg',
        master => 'orc2.world',
        use_existing_objects => TRUE,
        copy_rows => FALSE,
        propagation_mode => 'ASYNCHRONOUS');
END;
/

/*
```

---

**注意：** DBA\_REPSITES ビューに orc2.world が表示されてから作業を続けます。他の SQL\*Plus セッションで、次の SELECT 文を実行し、orc2.world が表示されることを確認します。

---

```
SELECT DBLINK FROM DBA_REPSITES WHERE GNAME = 'HR_REPG';
```

---

```
*/

PAUSE Press <RETURN> to continue.
```

```
BEGIN
    DBMS_REPCAT.ADD_MASTER_DATABASE (
        gname => 'hr_repg',
        master => 'orc3.world',
        use_existing_objects => TRUE,
        copy_rows => FALSE,
```

```
propagation_mode => 'ASYNCHRONOUS');
END;
/
/*
```

---

---

**注意：** DBA\_REPSITES ビューに orc3.world が表示されてから作業を続けます。他の SQL\*Plus セッションで、次の SELECT 文を実行し、orc3.world が表示されることを確認します。

---

---

```
SELECT DBLINK FROM DBA_REPSITES WHERE GNAME = 'HR_REPG';
```

---

---

```
*/
PAUSE Press <RETURN> to continue.
/*
```

#### 手順 5 競合の可能性がある場合は、競合解消メソッドを設定します。

---

---

**注意：** グループ作成中に 1 つ以上の表をマスター・グループに追加した場合は、レプリケーション・アクティビティの即時再開は行いません。まず、レプリケーションの競合の可能性を検討して、グループのレプリケート表の競合解消を構成します。

---

---

**関連項目：** 競合解消メソッドの構成の詳細は、[第 6 章「競合解消の構成」](#)を参照してください。

```
*/
PAUSE Press <RETURN> to continue after configuring conflict resolution methods or if
no conflict resolution methods are required.
/*
```

**手順 6 レプリケーション・サポートを生成します。**

```
*/

BEGIN
    DBMS_REPCAT.GENERATE_REPLICATION_SUPPORT (
        sname => 'hr',
        oname => 'countries',
        type => 'TABLE',
        min_communication => TRUE);
END;
/

BEGIN
    DBMS_REPCAT.GENERATE_REPLICATION_SUPPORT (
        sname => 'hr',
        oname => 'departments',
        type => 'TABLE',
        min_communication => TRUE);
END;
/

BEGIN
    DBMS_REPCAT.GENERATE_REPLICATION_SUPPORT (
        sname => 'hr',
        oname => 'employees',
        type => 'TABLE',
        min_communication => TRUE);
END;
/

BEGIN
    DBMS_REPCAT.GENERATE_REPLICATION_SUPPORT (
        sname => 'hr',
        oname => 'jobs',
        type => 'TABLE',
        min_communication => TRUE);
END;
/

BEGIN
    DBMS_REPCAT.GENERATE_REPLICATION_SUPPORT (
        sname => 'hr',
        oname => 'job_history',
        type => 'TABLE',
        min_communication => TRUE);
```



```

END;
/

BEGIN
    DBMS_REPCAT.GENERATE_REPLICATION_SUPPORT (
        sname => 'hr',
        oname => 'locations',
        type => 'TABLE',
        min_communication => TRUE);
END;
/

BEGIN
    DBMS_REPCAT.GENERATE_REPLICATION_SUPPORT (
        sname => 'hr',
        oname => 'regions',
        type => 'TABLE',
        min_communication => TRUE);
END;
/

/*

```

---



---

**注意：** DBA\_REPCATLOG ビューが空になってから、マスター・アクティビティを再開します。DBA\_REPCATLOG ビューを監視するには、次の SELECT 文を実行します。

---



---

```

SELECT COUNT(*) FROM DBA_REPCATLOG WHERE GNAME = 'HR_REPG';

```

---



---

```

*/

PAUSE Press <RETURN> to continue.

/*

```

### 手順7 レプリケーションを開始します。

マスター・グループの作成、レプリケーション・オブジェクトの追加、レプリケーション・サポートの生成および追加マスター・データベースの追加を行った後で、レプリケーション・アクティビティを開始します。RESUME\_MASTER\_ACTIVITY プロシージャを使用して、指定したマスター・グループのレプリケーションをオンにします。

```
*/

BEGIN
    DBMS_REPCAT.RESUME_MASTER_ACTIVITY (
        gname => 'hr_repg');
END;
/

SET ECHO OFF

SPOOL OFF

/***** END OF SCRIPT *****/
```

---

## 配置テンプレートの作成

この章では、レプリケーション・マネージメント API を使用して、配置テンプレートを作成する方法を説明します。この章では、次の項目を説明します。

- [Oracle 配置テンプレートの概念](#)
- [配置テンプレートを作成する前に](#)
- [配置テンプレートの作成](#)
- [配置テンプレートをパッケージ化してインスタンス化](#)
- [配置テンプレートのインスタンス化](#)

マテリアライズド・ビュー環境を構築する前に、マスター・サイトの設定、マスター・グループの作成および目的のマテリアライズド・ビュー・サイトの設定を行います。また、作成したマテリアライズド・ビュー・サイトでのアクティビティによって、マスター・サイトで競合が発生する可能性がある場合は、マテリアライズド・ビュー・グループを作成する前に、マテリアライズド・ビューのマスター表に対して競合解消を設定します。

### 関連項目：

- [2-4 ページ「マスター・サイトの設定」](#)
- [3-2 ページ「マスター・グループ作成の概要」](#)
- [2-20 ページ「マテリアライズド・ビュー・サイトの設定」](#)
- [第 6 章「競合解消の構成」](#)

## Oracle 配置テンプレートの概念

Oracle で提供される配置テンプレートを使用すると、データベース管理者はマテリアライズド・ビュー環境をパッケージ化して、その配布とインストールを簡単かつ安全に行ったり、カスタマイズしたりできます。配置テンプレートは、1つのマテリアライズド・ビューと固定データ・セットのみで単純に構成することも、数百のマテリアライズド・ビューと1つ以上の変数に基づく動的データ・セットで複雑に構成することもできます。この目的は、環境を1回定義して、必要に応じてその配置テンプレートを配置することです。Oracle 配置テンプレートの特長を次に示します。

- 集中管理
- マテリアライズド・ビュー環境を繰り返し配置可能
- テンプレート・パラメータを使用したリモート・サイトでのデータのサブセット化
- テンプレートのインスタンス化とデータ・アクセスを制御する許可されたユーザーのリスト

マテリアライズド・ビュー環境を適用するために、DBA がマスター・サイトに配置テンプレートを作成します。このテンプレートには、マテリアライズド・ビュー環境を配置するために必要なすべての情報を格納します。たとえば、リモート・サイトとターゲット・リフレッシュ・グループでオブジェクトを作成するための DDL などを格納します。また、このテンプレートには、ユーザーのセキュリティ情報やカスタム・マテリアライズド・ビューを作成するためのテンプレート・パラメータへのリンクも含まれています。

次のオブジェクト型のインスタンス化では、配置テンプレートは使用できません。

- ユーザー定義型
- ユーザー定義型本体
- ユーザー定義演算子
- 索引タイプ

これらのオブジェクト型に基づいたオブジェクトのインスタンス化でも、配置テンプレートは使用できません。

**関連項目：** 配置テンプレートの概念的な情報の詳細は『Oracle9i アドバンスド・レプリケーション』を参照してください。

## 配置テンプレートを作成する前に

マスター・サイトの1つで、高速リフレッシュのマテリアライズド・ビューをサポートするには、マテリアライズド・ビューにレプリケートされる各マスター表のマテリアライズド・ビュー・ログを作成します。

この章の例では、hr サンプル・スキーマを使用します。hr スキーマの表のマテリアライズド・ビュー・ログを作成するには、次の命令を入力します。

```
CONNECT hr/hr@orc3.world
```

```
CREATE MATERIALIZED VIEW LOG ON hr.countries;  
CREATE MATERIALIZED VIEW LOG ON hr.departments;  
CREATE MATERIALIZED VIEW LOG ON hr.employees;  
CREATE MATERIALIZED VIEW LOG ON hr.jobs;  
CREATE MATERIALIZED VIEW LOG ON hr.job_history;  
CREATE MATERIALIZED VIEW LOG ON hr.locations;  
CREATE MATERIALIZED VIEW LOG ON hr.regions;
```

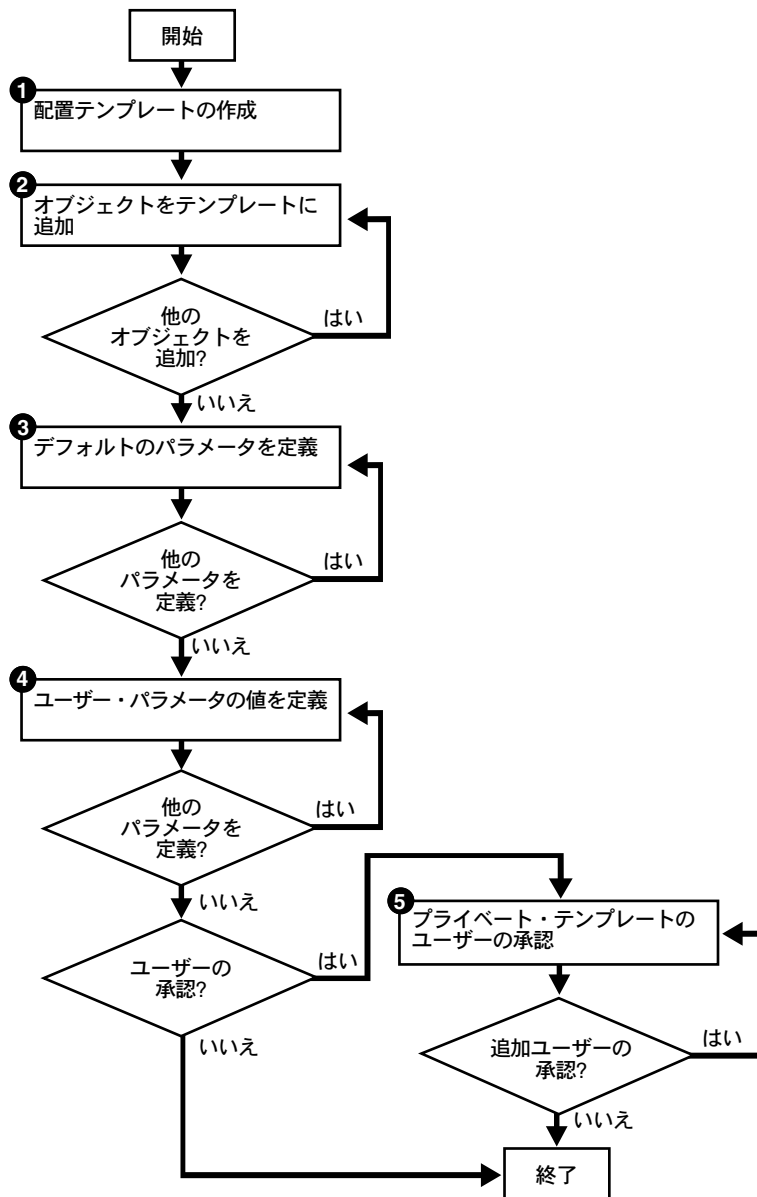
**関連項目：** この SQL 文の詳細は、『Oracle9i SQL リファレンス』の CREATE MATERIALIZED VIEW LOG 文を参照してください。

## 配置テンプレートの作成

この項では、レプリケーション・マネージメント API を使用して配置テンプレートを作成する、完全なスクリプト例を示します。

**関連項目：** 配置テンプレートの概念的情報およびアーキテクチャ的信息は『Oracle9i アドバンスド・レプリケーション』を参照してください。

図 4-1 配置テンプレートの作成



スクリプト内のコメントにも、必ず目を通してください。コメントには、レプリケーション・マネージメント API で配置テンプレートを作成するための重要で役立つ情報が記述されています。

---

---

**注意：** 列のサブセットをマスター表とするマテリアライズド・ビューを作成する場合は、レプリケーション・マネージメント・ツールを使用します。列のサブセット化の詳細は、『Oracle9i アドバンスド・レプリケーション』およびレプリケーション・マネージメント・ツールのオンライン・ヘルプを参照してください。

---

---



---

---

**注意：** このドキュメントをオンラインで参照している場合は、次の「BEGINNING OF SCRIPT」の行から 4-12 ページの「END OF SCRIPT」の行までのテキストをテキスト・エディタにコピーして編集し、使用している環境に適したスクリプトを作成します。

---

---

```
/***** BEGINNING OF SCRIPT *****/
```

このスクリプトでは、4つのテンプレート・オブジェクト、2つのテンプレート・パラメータ、ユーザー・パラメータ値セットおよび許可されたユーザーを含むプライベート配置テンプレートが作成されます。次に示す手順に従って、テンプレートを構築します。

### 手順 1 配置テンプレートを作成します。

配置テンプレートの構成要素を作成する前に、CREATE\_RERESH\_TEMPLATE プロシージャを使用して、配置テンプレートの名前、テンプレートの特性（パブリックまたはプライベートのステータス、ターゲット・リフレッシュ・グループおよび所有者）を定義します。

```
*/

SET ECHO ON

SPOOL create_dt.out

CONNECT repadmin/repadmin@orc3.world

DECLARE
  a NUMBER;
```

```
BEGIN
  a := DBMS_REPCAT_RGT.CREATE_REFRESH_TEMPLATE (
    owner => 'hr',
    refresh_group_name => 'hr_refg',
    refresh_template_name => 'hr_refg_dt',
    template_comment => 'Human Resources Deployment Template',
    public_template => 'N');
END;
/

/*
```

### 手順 2 次の手順に従って、オブジェクトをテンプレートに追加します。

- a. countries\_mv マテリアライズド・ビューを作成します。

```
*/

DECLARE
  tempstring VARCHAR2(3000);
  a NUMBER;
BEGIN
  tempstring := 'CREATE MATERIALIZED VIEW hr.countries_mv
    REFRESH FAST WITH PRIMARY KEY FOR UPDATE AS SELECT
    country_id, country_name, region_id
    FROM hr.countries@:dblink';
  a := DBMS_REPCAT_RGT.CREATE_TEMPLATE_OBJECT (
    refresh_template_name => 'hr_refg_dt',
    object_name => 'countries_mv',
    object_type => 'SNAPSHOT',
    ddl_text => tempstring,
    master_rollback_seg => 'rbs');
END;
/

/*
```

マテリアライズド・ビューを作成するときは、必ず、マテリアライズド・ビューに対する問合せの中で表の所有者のスキーマ名を指定します。前述の例では、countries 表の所有者として hr を指定しました。



- b. departments\_mv マテリアライズド・ビューを作成します。

```
*/

DECLARE
    tempstring VARCHAR2(3000);
    a NUMBER;
BEGIN
    tempstring := 'CREATE MATERIALIZED VIEW hr.departments_mv
REFRESH FAST WITH PRIMARY KEY FOR UPDATE AS SELECT
    department_id, department_name, manager_id, location_id
FROM hr.departments@:dblink';
    a := DBMS_REPCAT_RGT.CREATE_TEMPLATE_OBJECT (
        refresh_template_name => 'hr_refg_dt',
        object_name => 'departments_mv',
        object_type => 'SNAPSHOT',
        ddl_text => tempstring,
        master_rollback_seg => 'rbs');

END;
/

/*
```

- c. employees\_mv マテリアライズド・ビューを作成します。

```
*/

DECLARE
    tempstring VARCHAR2(3000);
    a NUMBER;
BEGIN
    tempstring := 'CREATE MATERIALIZED VIEW hr.employees_mv
REFRESH FAST WITH PRIMARY KEY FOR UPDATE AS SELECT
    employee_id, first_name, last_name, email, phone_number,
    hire_date, job_id, salary, commission_pct, manager_id,
    department_id
FROM hr.employees@:dblink WHERE department_id = :dept';
    a := DBMS_REPCAT_RGT.CREATE_TEMPLATE_OBJECT (
        refresh_template_name => 'hr_refg_dt',
        object_name => 'employees_mv',
        object_type => 'SNAPSHOT',
        ddl_text => tempstring,
        master_rollback_seg => 'rbs');

END;
/

/*
```

- d. jobs\_mv マテリアライズド・ビューを作成します。

```
*/

DECLARE
    tempstring VARCHAR2(3000);
    a NUMBER;
BEGIN
    tempstring := 'CREATE MATERIALIZED VIEW hr.jobs_mv
        REFRESH FAST WITH PRIMARY KEY FOR UPDATE AS SELECT
            job_id, job_title, min_salary, max_salary
        FROM hr.jobs@:dblink';
    a := DBMS_REPCAT_RGT.CREATE_TEMPLATE_OBJECT (
        refresh_template_name => 'hr_refg_dt',
        object_name => 'jobs_mv',
        object_type => 'SNAPSHOT',
        ddl_text => tempstring,
        master_rollback_seg => 'rbs');

END;
/

/*
```

- e. job\_history\_mv マテリアライズド・ビューを作成します。

```
*/

DECLARE
    tempstring VARCHAR2(3000);
    a NUMBER;
BEGIN
    tempstring := 'CREATE MATERIALIZED VIEW hr.job_history_mv
        REFRESH FAST WITH PRIMARY KEY FOR UPDATE AS SELECT
            employee_id, start_date, end_date, job_id, department_id
        FROM hr.job_history@:dblink';
    a := DBMS_REPCAT_RGT.CREATE_TEMPLATE_OBJECT (
        refresh_template_name => 'hr_refg_dt',
        object_name => 'job_history_mv',
        object_type => 'SNAPSHOT',
        ddl_text => tempstring,
        master_rollback_seg => 'rbs');

END;
/

/*
```

- f. locations\_mv マテリアライズド・ビューを作成します。

```
*/  
  
DECLARE  
    tempstring VARCHAR2(3000);  
    a NUMBER;  
BEGIN  
    tempstring := 'CREATE MATERIALIZED VIEW hr.locations_mv  
REFRESH FAST WITH PRIMARY KEY FOR UPDATE AS SELECT  
    location_id, street_address, postal_code, city,  
    state_province, country_id  
FROM hr.locations@:dblink';  
    a := DBMS_REPCAT_RGT.CREATE_TEMPLATE_OBJECT (  
        refresh_template_name => 'hr_refg_dt',  
        object_name => 'locations_mv',  
        object_type => 'SNAPSHOT',  
        ddl_text => tempstring,  
        master_rollback_seg => 'rbs');  
END;  
/  
  
/*
```

- g. regions\_mv マテリアライズド・ビューを作成します。

```
*/  
  
DECLARE  
    tempstring VARCHAR2(3000);  
    a NUMBER;  
BEGIN  
    tempstring := 'CREATE MATERIALIZED VIEW hr.regions_mv  
REFRESH FAST WITH PRIMARY KEY FOR UPDATE AS SELECT  
    region_id, region_name  
FROM hr.regions@:dblink';  
    a := DBMS_REPCAT_RGT.CREATE_TEMPLATE_OBJECT (  
        refresh_template_name => 'hr_refg_dt',  
        object_name => 'regions_mv',  
        object_type => 'SNAPSHOT',  
        ddl_text => tempstring,  
        master_rollback_seg => 'rbs');  
END;  
/  
  
/*
```

### 手順3 パラメータのデフォルト値を定義します。

他の手順では、CREATE\_\* ファンクションおよびプロシージャではなく、ALTER\_TEMPLATE\_PARM プロシージャを使用して、テンプレート・パラメータの値とプロンプト文字列を定義します。実パラメータは、手順 1b および 1c で作成されたので、ALTER\_\* プロシージャを使用します。:dblink および :dept テンプレート・パラメータは ddl\_text パラメータで定義済です。Oracle は DDL 中のこれらのパラメータを検出して、自動的にテンプレート・パラメータを作成します。テンプレート・パラメータ情報の残り（すなわち、デフォルトのパラメータ値とプロンプト文字列）を定義するには、ALTER\_TEMPLATE\_PARM プロシージャを使用します。

次の手順に従って、パラメータのデフォルト値を定義します。

- a. dept パラメータのデフォルト値を定義します。

```
*/

BEGIN
    DBMS_REPCAT_RGT.ALTER_TEMPLATE_PARM (
        refresh_template_name => 'hr_refg_dt',
        parameter_name => 'dept',
        new_default_parm_value => '30',
        new_prompt_string => 'Enter your department number:',
        new_user_override => 'Y');
END;
/

/*
```

- b. dblink パラメータのデフォルト値を定義します。

```
*/

BEGIN
    DBMS_REPCAT_RGT.ALTER_TEMPLATE_PARM (
        refresh_template_name => 'hr_refg_dt',
        parameter_name => 'dblink',
        new_default_parm_value => 'orc3.world',
        new_prompt_string => 'Enter your master site:',
        new_user_override => 'Y');
END;
/

/*
```

**手順4 ユーザー・パラメータ値を定義します。**

個々のリモート・マテリアライズド・ビュー・サイトにおけるカスタム・データ・セットのインスタンス化を自動的に行うためには、指定したユーザーが、ターゲット・テンプレートをインスタンス化した場合に自動的に使用されるユーザー・パラメータ値を定義します。CREATE\_USER\_PARM\_VALUE プロシージャを使用すれば、ユーザー・パラメータに値を割り当てることができます。

次の手順に従って、ユーザー・パラメータ値を定義します。

- a. ユーザー hr に対する dept ユーザー・パラメータ値を定義します。

```
*/  
  
DECLARE  
    a NUMBER;  
BEGIN  
    a := DBMS_REPCAT_RGT.CREATE_USER_PARM_VALUE (  
        refresh_template_name => 'hr_refg_dt',  
        parameter_name => 'dept',  
        user_name => 'hr',  
        parm_value => '20');  
END;  
/  
  
/*
```

- b. ユーザー hr に対する dblink ユーザー・パラメータ値を定義します。

```
*/  
  
DECLARE  
    a NUMBER;  
BEGIN  
    a := DBMS_REPCAT_RGT.CREATE_USER_PARM_VALUE (  
        refresh_template_name => 'hr_refg_dt',  
        parameter_name => 'dblink',  
        user_name => 'hr',  
        parm_value => 'orc3.world');  
END;  
/  
  
/*
```

#### 手順 5 プライベート・テンプレートに対するユーザーを許可します。

これは、プライベート・テンプレートであるため（4-3 ページの手順で定義した DBMS\_REPCAT\_RGT.CREATE\_REFRESH\_TEMPLATE ファンクションの public\_template => 'n'）、dt\_personnel 配置テンプレートのインスタンス化をユーザーに許可する必要があります。権限を与えられたユーザーを作成するには、DBMS\_REPCAT\_RGT パッケージの CREATE\_USER\_AUTHORIZATION ファンクションを使用します。

```
*/

DECLARE
    a NUMBER;
BEGIN
    a := DBMS_REPCAT_RGT.CREATE_USER_AUTHORIZATION (
        USER_NAME => 'hr',
        REFRESH_TEMPLATE_NAME => 'hr_refg_dt');
END;
/

COMMIT;

SET ECHO OFF

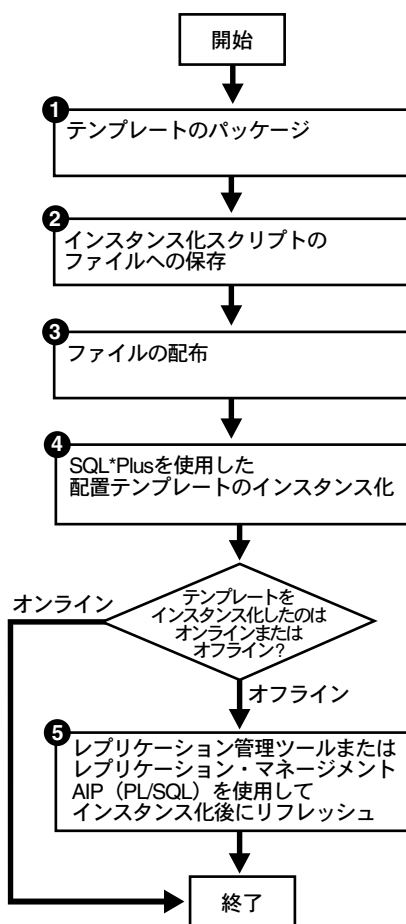
SPOOL OFF

/***** END OF SCRIPT *****/
```

## 配置テンプレートをパッケージ化してインスタンス化

配置テンプレートを作成した後、テンプレートをパッケージ化してインスタンス化を行います。ここでは、オンラインとオフラインのインスタンス化プロシージャの使用例を示します。オンラインとオフラインのインスタンス化プロシージャは非常に類似しており、必要に応じて `INSTANTIATE_ONLINE` ファンクションと `INSTANTIATE_OFFLINE` ファンクションのいずれかを使用します。この項では、2つのタスクを説明します。まず、インスタンス化スクリプトを作成して、次にそのスクリプトをファイルに保存します。

図 4-2 配置テンプレートのパッケージ化およびインスタンス化



## 配置テンプレートのパッケージ化

INstantiate\_offline ファンクションまたは Instantiate\_online ファンクションを実行すると、Oracle では USER\_REPCAT\_TEMP\_OUTPUT データ・ディクショナリ・ビューにスクリプトが移入されて、リモート・マテリアライズド・ビュー環境が構築されます。オンラインとオフラインのスクリプトのいずれにも、配置テンプレートで指定されたオブジェクトを作成するための SQL 文が含まれます。ただし、オフライン・インスタンスエーション・スクリプトには、オブジェクトに移入するデータが含まれます。オンライン・インスタンスエーション・スクリプトには、このデータは含まれません。オンライン・インスタンスエーションでは、マテリアライズド・ビュー・サイトがマスター・サイトに接続して、このデータをダウンロードします。

必要に応じて、「配置テンプレートをパッケージ化してオフライン・インスタンスエーション」または「配置テンプレートをパッケージ化してオンライン・インスタンスエーション」に示すスクリプトを実行してください。これらの項は、Oracle Enterprise Edition、Oracle Standard Edition または Oracle Personal Edition を実行中のマテリアライズド・ビュー・サイトのテンプレートのパッケージ化にのみ適用されます。

---

---

**注意：** Instantiate\_offline および Instantiate\_online ファンクションのいずれかを、特定のマテリアライズド・ビュー・サイトに対し 1 回以上実行する場合は、サイトのテンプレートを再パッケージする前に、DBMS\_REPCAT\_RGT パッケージの DROP\_SITE\_INSTANTIATION プロシージャを実行します。このプロシージャを実行しない場合は、テンプレート・サイトの重複を示すエラーが Oracle から戻されます。

---

---

## 配置テンプレートをパッケージ化してオフライン・インスタンスエーション

Instantiate\_offline ファンクションは、指定された配置テンプレートの内容に従って、マテリアライズド・ビュー環境を構築するスクリプトを生成します。このスクリプトには、マテリアライズド・ビュー環境を構築するための DDL (CREATE 文) の他に、マテリアライズド・ビュー環境に適切なデータ・セットを移入するための DML (INSERT 文) が含まれています。

---

---

**注意：** 配置テンプレートをパッケージ化するマスター・サイトに、配置テンプレート用のターゲット・マスター・オブジェクトが含まれている場合は、ループバック・データベース・リンクを作成する必要があります。

---

---



```
--Use the INSTANTIATE_OFFLINE function to package the
--template for offline instantiation by a remote materialized view
--site. Executing this procedure both creates a script that
--creates that materialized view environment and populates the
--environment with the proper data set. This script is stored
--in the temporary USER_REPCAT_TEMP_OUTPUT view.

CONNECT repadmin/repadmin@orc3.world

SET SERVEROUTPUT ON
DECLARE
    dt_num NUMBER;
BEGIN
    dt_num := DBMS_REPCAT_RGT.INSTANTIATE_OFFLINE(
        refresh_template_name => 'hr_refg_dt',
        user_name => 'hr',
        site_name => 'mv4.world',
        next_date => SYSDATE,
        interval => 'SYSDATE + (1/144)');
    DBMS_OUTPUT.PUT_LINE('Template ID = ' || dt_num);
END;
/
COMMIT;
/
```

dt\_num 変数に戻される値を記録しておきます。この値は、USER\_REPCAT\_TEMP\_OUTPUT データ・ディクショナリ・ビューから選択して、生成されたスクリプトを取り出すときに使用する必要があります。ここに示したスクリプトを実行した後、4-16 ページの「[インスタンス化スクリプトのファイルへの保存](#)」に示す手順に従ってください。このスクリプトは、個々のマテリアライズド・ビュー・サイトに一意であり、他のマテリアライズド・ビュー・サイトでは使用できません。

## 配置テンプレートをパッケージ化してオンライン・インスタンス化

INSTANTIATE\_ONLINE ファンクションは、指定された配置テンプレートの内容に従って、マテリアライズド・ビュー環境を構築するスクリプトを生成します。このスクリプトがリモート・マテリアライズド・ビュー・サイトで実行されると、スクリプト内の DDL (CREATE 文) に従ってマテリアライズド・ビュー・サイトが作成され、マスター・サイトから適切なデータ・セットが移入されます。この処理では、リモート・マテリアライズド・ビュー・サイトが、マスター・サイトにオンラインで接続されている必要があります。

**関連項目：** マテリアライズド・ビュー・サイトのその他の要件は、『Oracle9i アドバンスド・レプリケーション』を参照してください。

```
--Use the INSTANTIATE_ONLINE function to "package" the
--template for online instantiation by a remote materialized view
--site. Executing this procedure creates a script which can
--then be used to create a materialized view environment. This script
--is stored in the temporary USER_REPCAT_TEMP_OUTPUT view.
```

```
CONNECT repadmin/repadmin@orc3.world

SET SERVEROUTPUT ON
DECLARE
    dt_num NUMBER;
BEGIN
    dt_num := DBMS_REPCAT_RGT.INSTANTIATE_ONLINE(
        refresh_template_name => 'hr_refg_dt',
        user_name => 'hr',
        site_name => 'mv4.world',
        next_date => SYSDATE,
        interval => 'SYSDATE + (1/144)');
    DBMS_OUTPUT.PUT_LINE('Template ID = ' || dt_num);
END;
/
COMMIT;
/
```

dt\_num 変数に戻される値を記録しておきます。この値は、USER\_REPCAT\_TEMP\_OUTPUT データ・ディクショナリ・ビューから選択して、生成されたスクリプトを取り出すときに使用する必要があります。ここに示したタスクを実行した後、[「インスタンス化スクリプトのファイルへの保存」](#)に示す手順に従ってください。

## インスタンス化スクリプトのファイルへの保存

USER\_REPCAT\_TEMP\_OUTPUT データ・ディクショナリ・ビューの内容を効率的に保存するには、UTL\_FILE パッケージを使用して、USER\_REPCAT\_TEMP\_OUTPUT ビューの TEXT 列の内容をファイルに保存します。

---

---

**注意：** 次に示すアクションは、INSTANTIATE\_OFFLINE ファンクションまたは INSTANTIATE\_ONLINE ファンクションをコールした直後に行う必要があります。これは、USER\_REPCAT\_TEMP\_OUTPUT データ・ディクショナリ・ビューの内容が、一時的なものであるためです。4-14 ページの[「配置テンプレートのパッケージ化」](#)に示したスクリプトを実行していない場合は、それを実行してから、次に示すアクションを行ってください。

---

---

**関連項目：** UTL\_FILE パッケージの詳細は、『Oracle9i PL/SQL パッケージ・プロシージャおよびタイプ・リファレンス』を参照してください。

次の項目を入力して、配置テンプレート・スクリプトをファイルに保存します。

```
DECLARE
  fh UTL_FILE.FILE_TYPE;
  CURSOR ddlcursor(myid NUMBER) IS
    SELECT TEXT FROM USER_REPCAT_TEMP_OUTPUT WHERE OUTPUT_ID = myid ORDER BY LINE;
BEGIN
  fh := UTL_FILE.FOPEN ('file_location', 'file_name', 'w');
  UTL_FILE.PUT_LINE (fh, 'SET ECHO OFF;');
  FOR myrec IN ddlcursor(template_id) LOOP
    UTL_FILE.PUT_LINE(fh, myrec.text);
  END LOOP;
  UTL_FILE.PUT_LINE (fh, 'SET ECHO ON;');
  UTL_FILE.FFLUSH(fh);
  UTL_FILE.FCLOSE(fh);
END;
/
```

*file\_location*、*file\_name*および*template\_id*はプレースホルダであることに注意してください。使用している環境に合せて正しい値に置換します。

- *file\_location* プレースホルダを、テンプレート・スクリプトの保存先を示す完全なディレクトリ・パスで置換します。

---

**注意：** テンプレート・スクリプトの場所として指定する値は、UTL\_FILE\_DIR 初期化パラメータに登録された場所である必要があります。UTL\_FILE\_DIR 初期化パラメータに登録されていない場所を指定した場合は、テンプレートのスクリプトをファイルに保存する際、Oracle からエラーが返されます。UTL\_FILE\_DIR の詳細は、『Oracle9i アドバンスド・レプリケーション』を参照してください。

---

- *file\_name* プレースホルダを、テンプレートのスクリプトに使用する名前で置換します。
- *template\_id* プレースホルダを、以前テンプレートをパッケージ化した時に、INSTANTIATE\_OFFLINE ファンクションまたは INSTANTIATE\_ONLINE ファンクションから戻された番号で置換します。

たとえば、次のような値があるとします。

プレースホルダ	値
<i>file_location</i>	/home/gen_files/
<i>file_name</i>	sf.sql
<i>template_id</i>	18

これらの値を指定した後、レプリケーション管理者としてマスター・サイトに接続します。次のプロシージャを実行してテンプレートのスクリプトをファイルに保存します。

```
CONNECT repadmin/repadmin@orc3.world

DECLARE
  fh UTL_FILE.FILE_TYPE;
  CURSOR ddlcursor(myid NUMBER) IS
    SELECT TEXT FROM USER_REPCAT_TEMP_OUTPUT WHERE OUTPUT_ID = myid
    ORDER BY LINE;
BEGIN
  fh := UTL_FILE.FOPEN ('/home/gen_files/', 'sf.sql', 'w');
  UTL_FILE.PUT_LINE (fh, 'SET ECHO OFF;');
  FOR myrec IN ddlcursor(18) LOOP
    UTL_FILE.PUT_LINE(fh, myrec.text);
  END LOOP;
  UTL_FILE.PUT_LINE (fh, 'SET ECHO ON;');
  UTL_FILE.FFLUSH(fh);
  UTL_FILE.FCLOSE(fh);
END;
/
```

## インスタンス化ファイルの配布

インスタンス化スクリプトを作成して、それをファイルに保存した後、テンプレートをインスタンス化するリモート・マテリアライズド・ビュー・サイトにこのファイルを配布する必要があります。ファイルを配布するには、FTP サイトにファイルを送信するか、CD-ROM、フロッピー・ディスクなどに保存します。

## 配置テンプレートのインスタンス化

インスタンス化スクリプトがリモート・マテリアライズド・ビュー・サイトに配布された後、そこで配置テンプレートをインスタンス化できます。配置テンプレートをインスタンス化する前に、マテリアライズド・ビュー・サイトが設定されていることを確認してください。

### 関連項目：

- 配置テンプレートをインスタンス化する前に満たす必要があるマテリアライズド・ビュー・サイトの要件は、『Oracle9i アドバンスド・レプリケーション』を参照してください。
- 2-20 ページ「[マテリアライズド・ビュー・サイトの設定](#)」

次のスクリプトは、Oracle Enterprise Edition、Oracle Standard Edition または Oracle Personal Edition がインストールされたリモート・マテリアライズド・ビュー・サイトでインスタンス化処理を行う方法を示しています。

---

---

**注意：** このドキュメントをオンラインで参照している場合は、次の「BEGINNING OF SCRIPT」の行から 4-21 ページの「END OF SCRIPT」の行までのテキストをテキスト・エディタにコピーして編集し、使用している環境に適したスクリプトを作成します。

---

---

/\*\*\*\*\* BEGINNING OF SCRIPT \*\*\*\*\*/

**手順1 マテリアライズド・ビュー・サイトにスキーマが存在しない場合は、スキーマを作成します。**

リモート・マテリアライズド・ビュー・サイトでインスタンス化スクリプトを実行する前に、レプリケート・オブジェクトを含むスキーマを作成します。

次の例は、hr スキーマの作成方法を示します。このスキーマは、データベースにすでに存在する場合があります。

```
*/

SET ECHO ON

SPOOL instant_mv.out

CONNECT system/manager@mv4.world

CREATE TABLESPACE demo_mv
  DATAFILE 'demo_mv.dbf' SIZE 10M AUTOEXTEND ON
  EXTENT MANAGEMENT LOCAL AUTOALLOCATE;

CREATE TEMPORARY TABLESPACE temp_mv
  TEMPFILE 'temp_mv.dbf' SIZE 5M AUTOEXTEND ON;

CREATE USER hr IDENTIFIED BY hr;

ALTER USER hr DEFAULT TABLESPACE demo_mv
  QUOTA UNLIMITED ON demo_mv;

ALTER USER hr TEMPORARY TABLESPACE temp_mv;
```

```
GRANT
    CREATE SESSION,
    CREATE TABLE,
    CREATE PROCEDURE,
    CREATE SEQUENCE,
    CREATE TRIGGER,
    CREATE VIEW,
    CREATE SYNONYM,
    ALTER SESSION,
    CREATE MATERIALIZED VIEW,
    ALTER ANY MATERIALIZED VIEW,
    CREATE DATABASE LINK
TO hr;
```

```
/*
```

### 手順2 データベース・リンクが存在しない場合は、レプリケート・スキーマへのデータベース・リンクを作成します。

配置テンプレートをインスタンス化する前に、レプリケーション・スキーマへの必要なデータベース・リンクが存在することを確認します。マテリアライズド・ビューの所有者は、マスター・サイトの設定時に作成した proxy\_refresher へのデータベース・リンクが必要です。

```
*/
```

```
CREATE PUBLIC DATABASE LINK orc3.world USING 'orc3.world';
```

```
CONNECT hr/hr@mv4.world
```

```
CREATE DATABASE LINK orc3.world
```

```
    CONNECT TO proxy_refresher IDENTIFIED BY proxy_refresher;
```

```
/*
```

**関連項目：** プロキシ・マスター・サイトのユーザー作成の詳細は、2-7 ページの手順7を参照してください。

**手順 3 インスタンス化スクリプトを実行します。**

```
*/
```

```
@d:¥sf.sql
```

```
SET ECHO OFF
```

```
SPOOL OFF
```

```
/*
```

構築するマテリアライズド・ビュー環境の規模やロードされるデータ量によっては、インスタンス化処理にかなり時間がかかる場合があります。

```
***** END OF SCRIPT *****/
```

## インスタンス化後のリフレッシュ・グループのリフレッシュ

オフライン・インスタンス化・メソッドを使用して配置テンプレートをインスタンス化した場合は、次の実行文を発行して、できるだけ早くリフレッシュ・グループをリフレッシュします。

```
CONNECT hr/hr@mv4.world
```

```
EXECUTE DBMS_REFRESH.REFRESH ('hr_refg');
```





---

## マテリアライズド・ビュー・グループの作成

この章では、リモート・マテリアライズド・ビュー・レプリケーション・サイトで、マテリアライズド・ビュー・グループを作成する方法を説明します。この章では、次の項目を説明します。

- [マテリアライズド・ビュー・グループ作成の概要](#)
- [マテリアライズド・ビュー・グループの作成](#)

マテリアライズド・ビュー環境を構築する前に、マスター・サイトの設定、マスター・グループの作成および目的のマテリアライズド・ビュー・サイトの設定を行います。また、作成したマテリアライズド・ビュー・サイトでのアクティビティによって、マスター・サイトで競合が発生する可能性がある場合は、マテリアライズド・ビュー・グループを作成する前に、マテリアライズド・ビューのマスター表に対して競合解消を設定します。

### 関連項目：

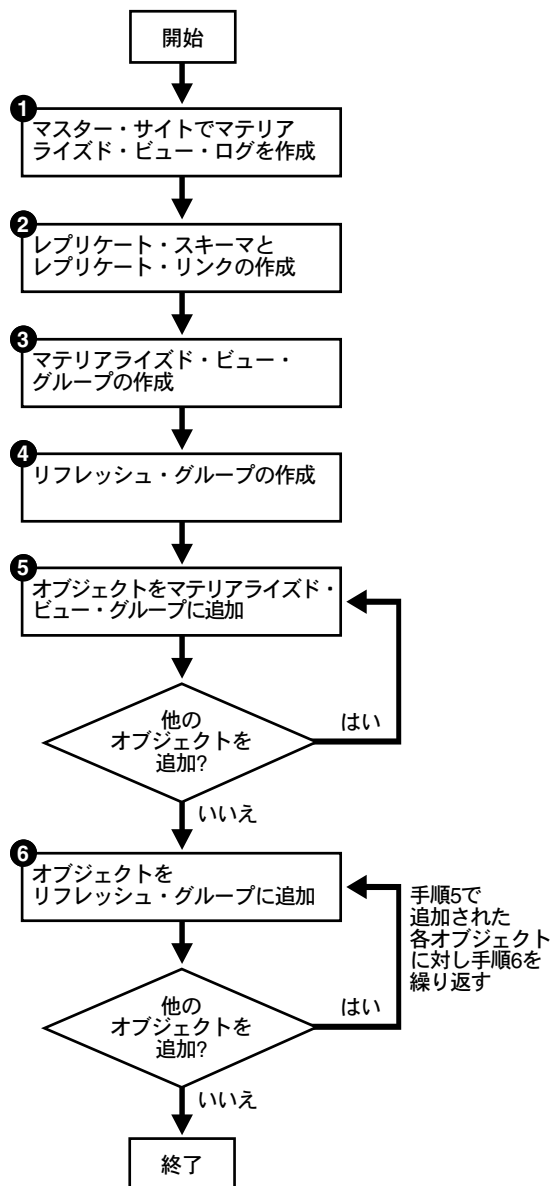
- [2-4 ページ「マスター・サイトの設定」](#)
- [3-2 ページ「マスター・グループ作成の概要」](#)
- [2-20 ページ「マテリアライズド・ビュー・サイトの設定」](#)
- [第 6 章「競合解消の構成」](#)

## マテリアライズド・ビュー・グループ作成の概要

マテリアライズド・ビュー・サイトを設定して、1 つ以上のマスター・グループを作成すれば、リモート・マテリアライズド・ビュー・サイトにマテリアライズド・ビュー・グループを作成できます。[図 5-1](#) では、マテリアライズド・ビュー・グループの作成プロセスを説明しています。

**関連項目：** マテリアライズド・ビュー・サイトの設定の詳細は、[第 2 章「レプリケーション・サイトの構築」](#) を参照してください。マスター・グループの作成の詳細は、[第 3 章「マスター・グループの作成」](#) を参照してください。

図 5-1 マテリアライズド・ビュー・グループの作成



## マテリアライズド・ビュー・グループの作成

この章では、2つのマテリアライズド・ビュー・サイト mv1.world および mv2.world に、2つのマテリアライズド・ビュー・グループを作成するプロセスを説明します。

- mv1.world のマテリアライズド・ビュー・グループは、orc1.world マスター・サイトの hr\_repg マスター・グループのオブジェクトに基づいています。
- mv2.world のマテリアライズド・ビュー・グループは、mv1.world マテリアライズド・ビュー・サイトの hr\_repg マスター・グループのオブジェクトに基づいています。

このため、この章の例では複数層マテリアライズド・ビュー環境の作成方法、1つ以上のマテリアライズド・ビューが基にする他のマテリアライズド・ビューをどこにするかを説明します。

次の手順に従って、これら2つのマテリアライズド・ビュー・グループを作成します。

---

**注意：** このドキュメントをオンラインで参照している場合は、次の「BEGINNING OF SCRIPT」の行から 5-19 ページの「END OF SCRIPT」の行までのテキストをテキスト・エディタにコピーして編集し、使用している環境に適したスクリプトを作成します。

---

/\*\*\*\*\* BEGINNING OF SCRIPT \*\*\*\*\*/

### mv1.world にマテリアライズド・ビュー・グループを作成

次の手順に従って、mv1.world マテリアライズド・ビュー・サイトに hr\_repg マテリアライズド・ビュー・グループを作成します。このマテリアライズド・ビュー・グループは、orc1.world マスター・サイトの hr\_repg マスター・グループに基づいています。

#### 手順1 マスター・サイトにマテリアライズド・ビュー・ログを作成します。

マスター・サイトの1つで、マテリアライズド・ビュー・サイトをサポートするには、マテリアライズド・ビューにレプリケートされる各マスター表のマテリアライズド・ビュー・ログを作成します。orc1.world が mv1.world マテリアライズド・ビュー・サイトのターゲット・マスター・サイトであることを 2-2 ページの図 2-1 で再確認してください。必要なマテリアライズド・ビュー・ログは、orc1.world に作成します。

\*/

SET ECHO ON

SPOOL create\_mv\_group.out

CONNECT hr/hr@orc1.world

```

CREATE MATERIALIZED VIEW LOG ON hr.countries;
CREATE MATERIALIZED VIEW LOG ON hr.departments;
CREATE MATERIALIZED VIEW LOG ON hr.employees;
CREATE MATERIALIZED VIEW LOG ON hr.jobs;
CREATE MATERIALIZED VIEW LOG ON hr.job_history;
CREATE MATERIALIZED VIEW LOG ON hr.locations;
CREATE MATERIALIZED VIEW LOG ON hr.regions;

```

```

/*

```

**関連項目：** この SQL 文の詳細は、『Oracle9i SQL リファレンス』の CREATE MATERIALIZED VIEW LOG 文を参照してください。

**手順 2 レプリケート・スキーマにデータベース・リンクが存在しない場合は、次の手順に従って作成します。マテリアライズド・ビュー・グループを構築する前に、リモート・マテリアライズド・ビュー・サイトにレプリケート・スキーマが存在して、必要なデータベース・リンクが作成されていることを確認します。**

- a. hr スキーマが存在しない場合は、スキーマを作成します。この例では、マテリアライズド・ビュー・サイトに hr スキーマが存在する場合は、手順 b に進んでください。

```

*/

```

```

CONNECT system/manager@mv1.world

```

```

CREATE TABLESPACE demo_mv1
  DATAFILE 'demo_mv1.dbf' SIZE 10M AUTOEXTEND ON
  EXTENT MANAGEMENT LOCAL AUTOALLOCATE;

```

```

CREATE TEMPORARY TABLESPACE temp_mv1
  TEMPFILE 'temp_mv1.dbf' SIZE 5M AUTOEXTEND ON;

```

```

CREATE USER hr IDENTIFIED BY hr;

```

```

ALTER USER hr DEFAULT TABLESPACE demo_mv1
  QUOTA UNLIMITED ON demo_mv1;

```

```

ALTER USER hr TEMPORARY TABLESPACE temp_mv1;

```

```

GRANT
  CREATE SESSION,
  CREATE TABLE,
  CREATE PROCEDURE,
  CREATE SEQUENCE,
  CREATE TRIGGER,
  CREATE VIEW,

```

```
CREATE SYNONYM,  
ALTER SESSION,  
CREATE MATERIALIZED VIEW,  
ALTER ANY MATERIALIZED VIEW,  
CREATE DATABASE LINK  
TO hr;
```

```
/*
```

- b. データベース・リンクが存在しない場合は、レプリケート・スキーマへのデータベース・リンクを作成します。

マテリアライズド・ビュー・グループを構築する前に、レプリケーション・スキーマへの必要なデータベース・リンクが存在することを確認します。マテリアライズド・ビューの所有者は、マスター・サイトの設定時に作成した `proxy_refresher` へのデータベース・リンクが必要です。

```
*/
```

```
CONNECT hr/hr@mv1.world
```

```
CREATE DATABASE LINK orcl.world  
CONNECT TO proxy_refresher IDENTIFIED BY proxy_refresher;
```

```
/*
```

**関連項目：** プロキシ・マスター・サイトのユーザー作成の詳細は、2-7 ページの手順 7 を参照してください。

**手順3 マテリアライズド・ビュー・グループを作成します。**

次のプロシージャは、リモート・マテリアライズド・ビュー・サイトのマテリアライズド・ビュー管理者が実行します。

```
*/
```

```
CONNECT mviewadmin/mviewadmin@mv1.world
```

```
/*
```

gname パラメータで指定したマスター・グループは、ターゲット・マスター・サイトでレプリケートするマスター・グループの名前と一致する必要があります。

```
*/
```

```
BEGIN
```

```
    DBMS_REPCAT.CREATE_MVIEW_REPGROUP (
        gname => 'hr_repg',
        master => 'orcl.world',
        propagation_mode => 'ASYNCHRONOUS');
```

```
END;
```

```
/
```

```
/*
```

**手順4 リフレッシュ・グループを作成します。**

特定のリフレッシュ・グループに追加されたすべてのマテリアライズド・ビューは、同時にリフレッシュされます。このため、リフレッシュ・グループ内の関連するマテリアライズド・ビュー間では、トランザクション一貫性が保たれます。

```
*/
```

```
BEGIN
```

```
    DBMS_REFRESH.MAKE (
        name => 'mviewadmin.hr_refg',
        list => '',
        next_date => SYSDATE,
        interval => 'SYSDATE + 1/24',
        implicit_destroy => FALSE,
        rollback_seg => '',
        push_deferred_rpc => TRUE,
        refresh_after_errors => FALSE);
```

```
END;
```

```
/
```

```
/*
```

**手順 5 次の手順に従って、オブジェクトをマテリアライズド・ビュー・グループに追加します。**

- a. マスター表に基づいてマテリアライズド・ビューを作成します。

マテリアライズド・ビューを作成するときは、必ず、マテリアライズド・ビューに対する問合せの中で表の所有者のスキーマ名を指定します。次の例では、各問合せにおける表の所有者として hr が指定されています。

```
*/

CREATE MATERIALIZED VIEW hr.countries_mv1
  REFRESH FAST WITH PRIMARY KEY FOR UPDATE
  AS SELECT * FROM hr.countries@orc1.world;

CREATE MATERIALIZED VIEW hr.departments_mv1
  REFRESH FAST WITH PRIMARY KEY FOR UPDATE
  AS SELECT * FROM hr.departments@orc1.world;

CREATE MATERIALIZED VIEW hr.employees_mv1
  REFRESH FAST WITH PRIMARY KEY FOR UPDATE
  AS SELECT * FROM hr.employees@orc1.world;

CREATE MATERIALIZED VIEW hr.jobs_mv1
  REFRESH FAST WITH PRIMARY KEY FOR UPDATE
  AS SELECT * FROM hr.jobs@orc1.world;

CREATE MATERIALIZED VIEW hr.job_history_mv1
  REFRESH FAST WITH PRIMARY KEY FOR UPDATE
  AS SELECT * FROM hr.job_history@orc1.world;

CREATE MATERIALIZED VIEW hr.locations_mv1
  REFRESH FAST WITH PRIMARY KEY FOR UPDATE
  AS SELECT * FROM hr.locations@orc1.world;

CREATE MATERIALIZED VIEW hr.regions_mv1
  REFRESH FAST WITH PRIMARY KEY FOR UPDATE
  AS SELECT * FROM hr.regions@orc1.world;

/*
```



- b. オブジェクトをマテリアライズド・ビュー・グループに追加します。

```
*/

BEGIN
    DBMS_REPCAT.CREATE_MVIEW_REPOBJECT (
        gname => 'hr_repg',
        sname => 'hr',
        oname => 'countries_mv1',
        type => 'SNAPSHOT',
        min_communication => TRUE);
END;
/

BEGIN
    DBMS_REPCAT.CREATE_MVIEW_REPOBJECT (
        gname => 'hr_repg',
        sname => 'hr',
        oname => 'departments_mv1',
        type => 'SNAPSHOT',
        min_communication => TRUE);
END;
/

BEGIN
    DBMS_REPCAT.CREATE_MVIEW_REPOBJECT (
        gname => 'hr_repg',
        sname => 'hr',
        oname => 'employees_mv1',
        type => 'SNAPSHOT',
        min_communication => TRUE);
END;
/

BEGIN
    DBMS_REPCAT.CREATE_MVIEW_REPOBJECT (
        gname => 'hr_repg',
        sname => 'hr',
        oname => 'jobs_mv1',
        type => 'SNAPSHOT',
        min_communication => TRUE);
END;
/
```

```
BEGIN
    DBMS_REPCAT.CREATE_MVIEW_REPOBJECT (
        gname => 'hr_repg',
        sname => 'hr',
        oname => 'job_history_mv1',
        type => 'SNAPSHOT',
        min_communication => TRUE);
END;
/

BEGIN
    DBMS_REPCAT.CREATE_MVIEW_REPOBJECT (
        gname => 'hr_repg',
        sname => 'hr',
        oname => 'locations_mv1',
        type => 'SNAPSHOT',
        min_communication => TRUE);
END;
/

BEGIN
    DBMS_REPCAT.CREATE_MVIEW_REPOBJECT (
        gname => 'hr_repg',
        sname => 'hr',
        oname => 'regions_mv1',
        type => 'SNAPSHOT',
        min_communication => TRUE);
END;
/

/*
```

**手順 6 リフレッシュ・グループにオブジェクトを追加します。**

リフレッシュ・グループに追加するすべてのマテリアライズド・ビュー・グループ・オブジェクトは同時にリフレッシュされるので、関連するマテリアライズド・ビュー間での参照整合性が保たれます。

```
*/

BEGIN
    DBMS_REFRESH.ADD (
        name => 'mviewadmin.hr_refg',
        list => 'hr.countries_mv1',
        lax => TRUE);
END;
/

BEGIN
    DBMS_REFRESH.ADD (
        name => 'mviewadmin.hr_refg',
        list => 'hr.departments_mv1',
        lax => TRUE);
END;
/

BEGIN
    DBMS_REFRESH.ADD (
        name => 'mviewadmin.hr_refg',
        list => 'hr.employees_mv1',
        lax => TRUE);
END;
/

BEGIN
    DBMS_REFRESH.ADD (
        name => 'mviewadmin.hr_refg',
        list => 'hr.jobs_mv1',
        lax => TRUE);
END;
/

BEGIN
    DBMS_REFRESH.ADD (
        name => 'mviewadmin.hr_refg',
        list => 'hr.job_history_mv1',
        lax => TRUE);
END;
/
```

```
BEGIN
  DBMS_REFRESH.ADD (
    name => 'mviewadmin.hr_refg',
    list => 'hr.locations_mv1',
    lax => TRUE);
END;
/

BEGIN
  DBMS_REFRESH.ADD (
    name => 'mviewadmin.hr_refg',
    list => 'hr.regions_mv1',
    lax => TRUE);
END;
/

/*
```

## mv2.world にマテリアライズド・ビュー・グループを作成

次の手順に従って、mv2.world マテリアライズド・ビュー・サイトに hr\_repg マテリアライズド・ビュー・グループを作成します。このマテリアライズド・ビュー・グループは、mv1.world マテリアライズド・ビュー・サイトの hr\_repg マテリアライズド・ビュー・グループに基づいています。

### 手順1 マスター・マテリアライズド・ビュー・サイトにマテリアライズド・ビュー・ログを作成します。

マスター・マテリアライズド・ビュー・サイトの1つが他のマテリアライズド・ビュー・サイトをサポートするには、他のマテリアライズド・ビュー・サイトにレプリケートされるマテリアライズド・ビューごとにマテリアライズド・ビュー・ログを作成します。mv1.world が mv2.world マテリアライズド・ビュー・サイトのターゲット・マスター内部化ビュー・サイトであることを 2-2 ページの [図 2-1](#) で再確認してください。必要なマテリアライズド・ビュー・ログは、mv1.world に作成します。

```
*/

CONNECT hr/hr@mv1.world

CREATE MATERIALIZED VIEW LOG ON hr.countries_mv1;
CREATE MATERIALIZED VIEW LOG ON hr.departments_mv1;
CREATE MATERIALIZED VIEW LOG ON hr.employees_mv1;
CREATE MATERIALIZED VIEW LOG ON hr.jobs_mv1;
CREATE MATERIALIZED VIEW LOG ON hr.job_history_mv1;
```

```
CREATE MATERIALIZED VIEW LOG ON hr.locations_mv1;  
CREATE MATERIALIZED VIEW LOG ON hr.regions_mv1;
```

```
/*
```

**関連項目：** この SQL 文の詳細は、『Oracle9i SQL リファレンス』の  
CREATE MATERIALIZED VIEW LOG 文を参照してください。

**手順 2 レプリケート・スキーマにデータベース・リンクが存在しない場合は、次の手順に従って作成します。マテリアライズド・ビュー・グループを構築する前に、リモート・マテリアライズド・ビュー・サイトにレプリケート・スキーマが存在して、必要なデータベース・リンクが作成されていることを確認します。**

- a. この例で、hr スキーマが存在しない場合は、スキーマを作成します。マテリアライズド・ビュー・サイトに hr スキーマが存在する場合は、手順 b に進んでください。

```
*/
```

```
CONNECT system/manager@mv2.world  
CREATE TABLESPACE demo_mv2  
  DATAFILE 'demo_mv2.dbf' SIZE 10M AUTOEXTEND ON  
  EXTENT MANAGEMENT LOCAL AUTOALLOCATE;
```

```
CREATE TEMPORARY TABLESPACE temp_mv2  
  TEMPFILE 'temp_mv2.dbf' SIZE 5M AUTOEXTEND ON;
```

```
CREATE USER hr IDENTIFIED BY hr;
```

```
ALTER USER hr DEFAULT TABLESPACE demo_mv2  
  QUOTA UNLIMITED ON demo_mv2;
```

```
ALTER USER hr TEMPORARY TABLESPACE temp_mv2;
```

```
GRANT  
  CREATE SESSION,  
  CREATE TABLE,  
  CREATE PROCEDURE,  
  CREATE SEQUENCE,  
  CREATE TRIGGER,  
  CREATE VIEW,  
  CREATE SYNONYM,  
  ALTER SESSION,  
  CREATE MATERIALIZED VIEW,  
  ALTER ANY MATERIALIZED VIEW,  
  CREATE DATABASE LINK  
TO hr;
```

```
/*
```

- b. データベース・リンクが存在しない場合は、レプリケート・スキーマへのデータベース・リンクを作成します。

マテリアライズド・ビュー・グループを構築する前に、レプリケーション・スキーマへの必要なデータベース・リンクが存在することを確認します。マテリアライズド・ビューの所有者は、マスター・マテリアライズド・ビュー・サイトの設定時に作成した `proxy_refresher` へのデータベース・リンクが必要です。

```
*/  
  
CONNECT hr/hr@mv2.world  
  
CREATE DATABASE LINK mv1.world  
  CONNECT TO proxy_refresher IDENTIFIED BY proxy_refresher;  
  
/*
```

**関連項目：** プロキシ・マスター・マテリアライズド・ビュー・サイトのユーザー作成の詳細は、2-26 ページの手順 6 を参照してください。

### 手順 3 マテリアライズド・ビュー・グループを作成します。

次のプロシージャは、リモート・マテリアライズド・ビュー・サイトのマテリアライズド・ビュー管理者が実行します。

```
*/  
  
CONNECT mviewadmin/mviewadmin@mv2.world  
  
/*  
  
gname パラメータで指定したレプリケーション・グループは、ターゲット・マスター・マテリアライズド・ビュー・サイトでレプリケートするレプリケーション・グループの名前と一致する必要があります。  
  
*/  
  
BEGIN  
  DBMS_REPCAT.CREATE_MVIEW_REPGROUP (  
    gname => 'hr_repg',  
    master => 'mv1.world',  
    propagation_mode => 'ASYNCHRONOUS');  
END;  
/  
  
/*
```

**手順 4 リフレッシュ・グループを作成します。**

特定のリフレッシュ・グループに追加されたすべてのマテリアライズド・ビューは、同時にリフレッシュされます。このため、リフレッシュ・グループ内の関連するマテリアライズド・ビュー間では、トランザクション一貫性が保たれます。

```

*/

BEGIN
    DBMS_REFRESH.MAKE (
        name => 'mviewadmin.hr_refg',
        list => '',
        next_date => SYSDATE,
        interval => 'SYSDATE + 1/24',
        implicit_destroy => FALSE,
        rollback_seg => '',
        push_deferred_rpc => TRUE,
        refresh_after_errors => FALSE);
END;
/

/*

```

**手順 5 次の手順に従って、オブジェクトをマテリアライズド・ビュー・グループに追加します。**

- a. マスター・マテリアライズド・ビューに基づいてマテリアライズド・ビューを作成します。

他のマテリアライズド・ビューに基づいたマテリアライズド・ビューを作成した場合は必ず、マテリアライズド・ビューの間合せでマテリアライズド・ビュー所有者の名前を指定します。次の例では、各間合せにおけるマテリアライズド・ビューの所有者として hr が指定されています。

```

*/

CREATE MATERIALIZED VIEW hr.countries_mv2
    REFRESH FAST WITH PRIMARY KEY FOR UPDATE
    AS SELECT * FROM hr.countries_mv1@mv1.world;

CREATE MATERIALIZED VIEW hr.departments_mv2
    REFRESH FAST WITH PRIMARY KEY FOR UPDATE
    AS SELECT * FROM hr.departments_mv1@mv1.world;

CREATE MATERIALIZED VIEW hr.employees_mv2
    REFRESH FAST WITH PRIMARY KEY FOR UPDATE
    AS SELECT * FROM hr.employees_mv1@mv1.world;

```

```
CREATE MATERIALIZED VIEW hr.jobs_mv2
  REFRESH FAST WITH PRIMARY KEY FOR UPDATE
  AS SELECT * FROM hr.jobs_mv1@mv1.world;

CREATE MATERIALIZED VIEW hr.job_history_mv2
  REFRESH FAST WITH PRIMARY KEY FOR UPDATE
  AS SELECT * FROM hr.job_history_mv1@mv1.world;

CREATE MATERIALIZED VIEW hr.locations_mv2
  REFRESH FAST WITH PRIMARY KEY FOR UPDATE
  AS SELECT * FROM hr.locations_mv1@mv1.world;

CREATE MATERIALIZED VIEW hr.regions_mv2
  REFRESH FAST WITH PRIMARY KEY FOR UPDATE
  AS SELECT * FROM hr.regions_mv1@mv1.world;

/*
```

- b. マテリアライズド・ビューをマテリアライズド・ビュー・グループに追加します。

```
*/

BEGIN
  DBMS_REPCAT.CREATE_MVIEW_REPOBJECT (
    gname => 'hr_repg',
    sname => 'hr',
    oname => 'countries_mv2',
    type => 'SNAPSHOT',
    min_communication => TRUE);
END;
/

BEGIN
  DBMS_REPCAT.CREATE_MVIEW_REPOBJECT (
    gname => 'hr_repg',
    sname => 'hr',
    oname => 'departments_mv2',
    type => 'SNAPSHOT',
    min_communication => TRUE);
END;
/
```



```
BEGIN
    DBMS_REPCAT.CREATE_MVIEW_REPOBJECT (
        gname => 'hr_repg',
        sname => 'hr',
        oname => 'employees_mv2',
        type => 'SNAPSHOT',
        min_communication => TRUE);
END;
/

BEGIN
    DBMS_REPCAT.CREATE_MVIEW_REPOBJECT (
        gname => 'hr_repg',
        sname => 'hr',
        oname => 'jobs_mv2',
        type => 'SNAPSHOT',
        min_communication => TRUE);
END;
/

BEGIN
    DBMS_REPCAT.CREATE_MVIEW_REPOBJECT (
        gname => 'hr_repg',
        sname => 'hr',
        oname => 'job_history_mv2',
        type => 'SNAPSHOT',
        min_communication => TRUE);
END;
/

BEGIN
    DBMS_REPCAT.CREATE_MVIEW_REPOBJECT (
        gname => 'hr_repg',
        sname => 'hr',
        oname => 'locations_mv2',
        type => 'SNAPSHOT',
        min_communication => TRUE);
END;
/
```

```
BEGIN
    DBMS_REPCAT.CREATE_MVIEW_REPOBJECT (
        gname => 'hr_repg',
        sname => 'hr',
        oname => 'regions_mv2',
        type => 'SNAPSHOT',
        min_communication => TRUE);
END;
/

/*
```

### 手順 6 オブジェクトをリフレッシュ・グループに追加します。

リフレッシュ・グループに追加するすべてのマテリアライズド・ビュー・グループ・オブジェクトは同時にリフレッシュされるので、関連するマテリアライズド・ビュー間での参照整合性が保たれます。

```
*/
```

```
BEGIN
    DBMS_REFRESH.ADD (
        name => 'mviewadmin.hr_refg',
        list => 'hr.countries_mv2',
        lax => TRUE);
END;
/
```

```
BEGIN
    DBMS_REFRESH.ADD (
        name => 'mviewadmin.hr_refg',
        list => 'hr.departments_mv2',
        lax => TRUE);
END;
/
```

```
BEGIN
    DBMS_REFRESH.ADD (
        name => 'mviewadmin.hr_refg',
        list => 'hr.employees_mv2',
        lax => TRUE);
END;
/
```

```
BEGIN
    DBMS_REFRESH.ADD (
        name => 'mviewadmin.hr_refg',
        list => 'hr.jobs_mv2',
        lax => TRUE);
END;
/

BEGIN
    DBMS_REFRESH.ADD (
        name => 'mviewadmin.hr_refg',
        list => 'hr.job_history_mv2',
        lax => TRUE);
END;
/

BEGIN
    DBMS_REFRESH.ADD (
        name => 'mviewadmin.hr_refg',
        list => 'hr.locations_mv2',
        lax => TRUE);
END;
/

BEGIN
    DBMS_REFRESH.ADD (
        name => 'mviewadmin.hr_refg',
        list => 'hr.regions_mv2',
        lax => TRUE);
END;
/

SET ECHO OFF

SPOOL OFF

/***** END OF SCRIPT *****/
```



---

## 競合解消の構成

この章では、レプリケーション環境に競合解消メソッドを定義する方法を説明します。この章では、次の項目を説明します。

- [競合解消の準備](#)
- [更新の競合の競合解消メソッドの作成](#)
- [一意性競合の競合解消メソッドの作成](#)
- [削除の競合防止メソッドの作成](#)
- [動的所有権の競合防止メソッド](#)
- [競合解消の監査](#)

## 競合解消の準備

レプリケーション環境に複数のサイトがある場合、競合が起こることがあります。競合を回避するようにデータベースとフロントエンド・アプリケーションを設計しても、競合を皆無にはできません。レプリケーションで重要な点は、レプリケーション環境に関係するすべてのサイトで、確実にデータを収束させることです。

データの競合が発生した場合、競合がビジネス・ルールに従って解消され、すべてのサイトでデータを正常に収束させるためのメカニズムが必要です。

アドバンスド・レプリケーションでは、ビジネス・ルールに従って競合を解消する、データベース用の競合解消システムを定義できます。Oracle の組込みの競合解消メソッドで競合を解消できない場合は、独自の競合解消メソッドを作成し、それを使用して競合を解消できます。

レプリケート表に競合解消メソッドを実装する前に、システム内のデータを分析し、競合の起こりやすい領域を確認します。たとえば、従業員番号などの静的データは、頻繁には変更されません。このため、競合はあまり起こりません。これに対して、従業員の担当顧客は頻繁に変わるので、データの競合が起こりやすくなります。

競合の起こりやすい領域を特定した後、その解消方法を決定します。たとえば、最新の変更を優先させたり、特定のサイトを優先させたりします。

この章では、各種の競合解消メソッドを説明します。この説明を参考にして、解消メソッドに最も適したメソッドを使用します。そのため各項を読み、発生する可能性のある競合の解消メソッドについて考慮してください。

問題の起こりうる領域を特定し、その問題を解決するビジネス・ルールを決定した後、Oracle の競合解消メソッド（またはユーザー独自のメソッドのいずれか）を使用して競合解消システムを実装します。

**関連項目：** 競合解消メソッドの概要および各メソッドに対するデータ収束の詳細は、『Oracle9i アドバンスド・レプリケーション』を参照してください。

## 更新の競合の競合解消メソッドの作成

データの競合は、複数のサイトでほぼ同時に同じ行が更新された場合、またはあるサイトの遅延トランザクションが他のサイトに正しく伝播される前に最も発生します。

更新の競合を回避する方法の 1 つは、同期レプリケーション環境を実装することです。ただし、この方法は、大量のネットワーク・リソースを必要とします。

もう 1 つの方法は、Oracle 競合解消メソッドを使用することです。これらの方法を使用すると、同じ行が複数の更新を受信した場合に起こる更新の競合に対処できます。

---

**競合解消メソッドの作成と静止：** 次の項の指示に従って、競合解消メソッドを追加するために静止するマスター・グループを指定します。各自のマスター・サイトで、Oracle8i リリース 8.1.7 以上が単一のマスター環境で実行されている場合は、競合解消メソッドを追加するためにマスター・グループを静止する必要はありません。静止が必要でない場合の詳細は、『Oracle9i アドバンスド・レプリケーション』の冒頭の項「Advanced Replication の新機能」を参照してください。

---

## 上書きおよび廃棄による競合解消メソッド

上書きによるメソッドと廃棄によるメソッドは、レプリケーション元またはレプリケーション先のサイトからの値を無視します。このため、複数のマスター・サイトでの確実なデータ収束は保証されません。これらのメソッドは、単一のマスター・サイトと複数のマテリアライズド・ビュー・サイトで使用するか、あるいは、なんらかのユーザー定義の通知機能と使用することを前提としています。

上書きによるメソッドは、レプリケーション元の新しい値で、レプリケーション先のカレント値を置き換えます。これに対して、廃棄によるメソッドは、レプリケーション元からの新しい値を無視します。

**関連項目：** 上書きおよび廃棄の詳細は、20-18 ページの「[ADD\\_conflicttype\\_RESOLUTION プロシージャ](#)」および『Oracle9i アドバンスド・レプリケーション』を参照してください。

次の手順に従って、上書きまたは廃棄による競合解消メソッドを作成します。次の例は、マスター・サイトでの廃棄による競合解消メソッドの使用を示しています。したがって、競合が発生すると、マテリアライズド・ビュー・サイトからのデータは廃棄され、マスター・サイトのデータは保持されます。

---

**注意：** このドキュメントをオンラインで参照している場合は、次の「BEGINNING OF SCRIPT」の行から 6-6 ページの「END OF SCRIPT」の行までのテキストをテキスト・エディタにコピーして編集し、使用している環境に適したスクリプトを作成します。

---

```
/***** BEGINNING OF SCRIPT *****/
```

**手順 1** レプリケーション管理者として接続します。次の手順のプロシージャは、レプリケーション管理者が実行します。

```
*/
```

```
SET ECHO ON
```

```
SPOOL discard_conflictres.out
```

```
CONNECT repadmin/repadmin@orc1.world
```

```
/*
```

**手順 2** 上書きまたは廃棄による競合解消メソッドを定義する前に、競合解消メソッドを適用する表を含むマスター・グループを静止します。単一のマスター・レプリケーション環境においては、マスター・グループの静止は必要ありません。詳細は、6-3 ページの「[競合解消メソッドの作成と静止](#)」を参照してください。

```
*/
```

```
BEGIN
```

```
    DBMS_REPCAT.SUSPEND_MASTER_ACTIVITY (  
        gname => 'hr_repg');
```

```
END;
```

```
/
```

```
/*
```

**手順 3** DBMS\_REPCAT.MAKE\_COLUMN\_GROUP プロシージャを使用して、ターゲット表の列グループを作成します。Oracle の競合解消メソッドはすべて、列グループと呼ばれる論理列グループ化に基づいています。

```
*/
```

```
BEGIN
```

```
    DBMS_REPCAT.MAKE_COLUMN_GROUP (  
        sname => 'hr',  
        oname => 'departments',  
        column_group => 'dep_cg',  
        list_of_column_names => 'manager_id,location_id');
```

```
END;
```

```
/
```

```
/*
```



**手順 4 特定の表に対し競合解消メソッドを設定するには、  
DBMS\_REPCAT.ADD\_UPDATE\_RESOLUTION プロシージャを使用します。この例では、  
OVERWRITE 競合解消メソッドを作成します。**

```
*/  
  
BEGIN  
    DBMS_REPCAT.ADD_UPDATE_RESOLUTION (  
        sname => 'hr',  
        oname => 'departments',  
        column_group => 'dep_cg',  
        sequence_no => 1,  
        method => 'DISCARD',  
        parameter_column_name => 'manager_id,location_id');  
END;  
/  
  
/*
```

**手順 5 競合解消メソッドを受信した表に対するレプリケーション・サポートを再生成します。**

```
*/  
  
BEGIN  
    DBMS_REPCAT.GENERATE_REPLICATION_SUPPORT (  
        sname => 'hr',  
        oname => 'departments',  
        type => 'TABLE',  
        min_communication => TRUE);  
END;  
/  
  
/*
```

**手順 6 レプリケーション・サポートが生成された後に、マスター・アクティビティを再開します。**

```
*/  
  
BEGIN  
    DBMS_REPCAT.RESUME_MASTER_ACTIVITY (  
        gname => 'hr_repg');  
END;  
/  
  
SET ECHO OFF  
  
SPOOL OFF  
  
/***** END OF SCRIPT *****/
```

## 最小および最大の競合解消メソッド

アドバンスド・レプリケーションによって列グループの競合が検出され、最小のメソッドまたは最大のメソッドがコールされると、列グループ内の該当する列で、レプリケーション元の新しい値とレプリケーション先のカレント値が比較されます。競合解消メソッドを定義する場合は、この列を指定します。

レプリケーション先の列の新しい値が、カレント値よりも小さい、または大きい場合（使用する方法により異なります）、レプリケーション元にある列グループの値がレプリケーション先に適用されます。その場合、その行に対しては、他のすべてのエラーが解決されていることを前提とします。そうでない場合は、行が変更されずにそのまま保持されます。

次の手順に従って、最小または最大の競合解消メソッドを作成します。

---

---

**注意：** このドキュメントをオンラインで参照している場合は、次の「BEGINNING OF SCRIPT」の行から 6-9 ページの「END OF SCRIPT」の行までのテキストをテキスト・エディタにコピーして編集し、使用している環境に適したスクリプトを作成します。

---

---

```

/***** BEGINNING OF SCRIPT *****/

```

**手順 1 レプリケーション管理者として接続します。次の手順のプロシージャは、レプリケーション管理者が実行します。**

```

*/

```

```

SET ECHO ON

```

```

SPOOL min_conflictres.out

```

```

CONNECT repadmin/repadmin@orc1.world

```

```

/*

```

**手順 2 最小または最大の競合解消メソッドを定義する前に、競合解消メソッドを適用する表を含むマスター・グループを静止します。単一のマスター・レプリケーション環境においては、マスター・グループの静止は必要ありません。詳細は、6-3 ページの「[競合解消メソッドの作成と静止](#)」を参照してください。**

```

*/

```

```

BEGIN

```

```

    DBMS_REPCAT.SUSPEND_MASTER_ACTIVITY (
        gname => 'hr_repg');

```

```

END;

```

```

/

```

```

/*

```

**手順 3 DBMS\_REPCAT.MAKE\_COLUMN\_GROUP プロシージャを使用して、ターゲット表の列グループを作成します。Oracle の競合解消メソッドはすべて、列グループと呼ばれる論理列グループ化に基づいています。**

```

*/

```

```

BEGIN

```

```

    DBMS_REPCAT.MAKE_COLUMN_GROUP (
        sname => 'hr',
        oname => 'jobs',
        column_group => 'job_minsal_cg',
        list_of_column_names => 'min_salary');

```

```

END;

```

```

/

```

```

/*

```

**手順 4 特定の表に対し競合解消メソッドを設定するには、**  
DBMS\_REPCAT.ADD\_UPDATE\_RESOLUTION プロシージャを使用します。この例では、  
MINIMUM 競合解消メソッドを作成します。

```
*/  
  
BEGIN  
    DBMS_REPCAT.ADD_UPDATE_RESOLUTION (  
        sname => 'hr',  
        oname => 'jobs',  
        column_group => 'job_minsal_cg',  
        sequence_no => 1,  
        method => 'MINIMUM',  
        parameter_column_name => 'min_salary');  
END;  
/  
  
/*
```

**手順 5 競合解消メソッドを受信した表に対するレプリケーション・サポートを再生成します。**

```
*/  
  
BEGIN  
    DBMS_REPCAT.GENERATE_REPLICATION_SUPPORT (  
        sname => 'hr',  
        oname => 'jobs',  
        type => 'TABLE',  
        min_communication => TRUE);  
END;  
/  
  
/*
```

**手順 6 RESUME\_MASTER\_ACTIVITY プロシージャを使用してレプリケーション・アクティビティを再開します。**

```
*/  
  
BEGIN  
    DBMS_REPCAT.RESUME_MASTER_ACTIVITY (  
        gname => 'hr_repg');  
END;  
/  
  
SET ECHO OFF  
  
SPOOL OFF  
  
/***** END OF SCRIPT *****/
```

## タイムスタンプによる競合解消メソッド

最古のタイムスタンプによるメソッドと最新のタイムスタンプによるメソッドは、最小のメソッドと最大のメソッドに類似しています。タイムスタンプによるメソッドを使用するには、DATE 型のレプリケート表にある列を指定する必要があります。アプリケーションで列グループのいずれかの列を更新する場合は、レプリケーション先の **timestamp** 列の値をローカルの SYSDATE 値で更新する必要があります。他のサイトから適用された変更に対しては、タイムスタンプ値をレプリケート元のタイムスタンプ値に設定します。

タイムスタンプによる競合解消が正しく機能するための 2 つの要素は次のとおりです。

- コンピュータ間で同期化された時刻の設定
  - タイムスタンプを自動的に記録するためのタイムスタンプ・フィールドとトリガー
- 次の手順に従って、タイムスタンプによる競合解消メソッドを作成します。

---

---

**注意：** このドキュメントをオンラインで参照している場合は、次の「BEGINNING OF SCRIPT」の行から 6-13 ページの「END OF SCRIPT」の行までのテキストをテキスト・エディタにコピーして編集し、使用している環境に適したスクリプトを作成します。

---

---

```
/****** BEGINNING OF SCRIPT *****/
```

**手順 1 レプリケーション管理者として接続します。次の手順のプロシージャは、レプリケーション管理者が実行します。**

```
*/
```

```
SET ECHO ON
```

```
SPOOL timestamp_conflictres.out
```

```
CONNECT repadmin/repadmin@orc1.world
```

```
/*
```

**手順 2 タイムスタンプによる競合解消メソッドを定義する前に、競合解消メソッドを適用する表を含むマスター・グループを静止します。単一のマスター・レプリケーション環境においては、マスター・グループの静止は必要ありません。詳細は、6-3 ページの「[競合解消メソッドの作成と静止](#)」を参照してください。**

```
*/
```

```
BEGIN
```

```
    DBMS_REPCAT.SUSPEND_MASTER_ACTIVITY (  
        gname => 'hr_repg');
```

```
END;
```

```
/
```

```
/*
```

**手順 3 ターゲット表にタイムスタンプ・フィールドが存在しない場合は、行が挿入または更新されたときにタイムスタンプ値を記録する表に、列を追加します。DDL をターゲット表に適用するには、ALTER\_MASTER\_REPOBJECT プロシージャを使用します。DDL を発行するのみでレプリケート・オブジェクトが無効になります。**

```
*/
```

```
BEGIN
```

```
    DBMS_REPCAT.ALTER_MASTER_REPOBJECT (  
        sname => 'hr',  
        oname => 'countries',  
        type => 'TABLE',  
        ddl_text => 'ALTER TABLE hr.countries ADD (timestamp DATE)');
```

```
END;
```

```
/
```

```
/*
```

**手順 4 変更された表のレプリケーション・サポートを再生成します。**

```
*/  
  
BEGIN  
    DBMS_REPCAT.GENERATE_REPLICATION_SUPPORT (  
        sname => 'hr',  
        oname => 'countries',  
        type => 'TABLE',  
        min_communication => TRUE);  
END;  
/  
  
/*
```

**手順 5 行が挿入または更新されたときにタイムスタンプを記録するトリガーを作成します。**  
記録された値は、タイムスタンプによるメソッドに基づいた競合の解消で使用されます。トリガーを作成して、マスター・グループに追加するには、DDL を直接実行するのではなく、DBMS\_REPCAT.CREATE\_MASTER\_REPOBJECT プロシージャを使用します。

---

---

**注意：** 優先グループによる競合解消には、datetime および interval データ型の列は使用できません。

---

---

```
*/  
  
BEGIN  
    DBMS_REPCAT.CREATE_MASTER_REPOBJECT (  
        gname => 'hr_repg',  
        type => 'TRIGGER',  
        oname => 'insert_time',  
        sname => 'hr',  
        ddl_text => 'CREATE TRIGGER hr.insert_time  
                     BEFORE  
                     INSERT OR UPDATE ON hr.countries FOR EACH ROW  
                     BEGIN  
                         IF DBMS_REPUTIL.FROM_REMOTE = FALSE THEN  
                             :NEW.TIMESTAMP := SYSDATE;  
                         END IF;  
                     END;');  
END;  
/  
  
/*
```

**手順 6** DBMS\_REPCAT.MAKE\_COLUMN\_GROUP プロシージャを使用して、ターゲット表の列グループを作成します。Oracle の競合解消メソッドはすべて、列グループと呼ばれる論理列グループ化に基づいています。

```
*/

BEGIN
    DBMS_REPCAT.MAKE_COLUMN_GROUP (
        sname => 'hr',
        oname => 'countries',
        column_group => 'countries_timestamp_cg',
        list_of_column_names => 'country_name,region_id,timestamp');
END;
/

/*
```

**手順 7** 特定の表に対し競合解消メソッドを設定するには、DBMS\_REPCAT.ADD\_UPDATE\_RESOLUTION プロシージャを使用します。この例では、以前作成した timestamp 列を使用して、LATEST TIMESTAMP 競合解消メソッドを指定します。

```
*/

BEGIN
    DBMS_REPCAT.ADD_UPDATE_RESOLUTION (
        sname => 'hr',
        oname => 'countries',
        column_group => 'countries_timestamp_cg',
        sequence_no => 1,
        method => 'LATEST TIMESTAMP',
        parameter_column_name => 'timestamp');
END;
/

/*
```



**手順 8 競合解消メソッドを受信した表に対するレプリケーション・サポートを再生成します。**

```
*/  
  
BEGIN  
    DBMS_REPCAT.GENERATE_REPLICATION_SUPPORT (  
        sname => 'hr',  
        oname => 'countries',  
        type => 'TABLE',  
        min_communication => TRUE);  
END;  
/  
  
/*
```

**手順 9 RESUME\_MASTER\_ACTIVITY プロシージャを使用してレプリケーション・アクティビティを再開します。**

```
*/  
  
BEGIN  
    DBMS_REPCAT.RESUME_MASTER_ACTIVITY (  
        gname => 'hr_repg');  
END;  
/  
  
SET ECHO OFF  
  
SPOOL OFF  
  
/***** END OF SCRIPT *****/
```

## 加算による競合解消メソッドと平均による競合解消メソッド

加算によるメソッドと平均によるメソッドは、1つの数値列のみから構成される列グループに使用します。この競合解消メソッドは、一方の値のみを受け付けるのではなく、比較対象の2つの値を加算するか、またはその2つの平均を取ります。

次の手順に従って、加算競合解消メソッドまたは平均競合解消メソッドを作成します。この例では、競合が発生した場合に、従業員のコミッション率の平均を取ります。

---

---

**注意：** このドキュメントをオンラインで参照している場合は、次の「BEGINNING OF SCRIPT」の行から 6-16 ページの「END OF SCRIPT」の行までのテキストをテキスト・エディタにコピーして編集し、使用している環境に適したスクリプトを作成します。

---

---

```
/****** BEGINNING OF SCRIPT *****/
```

**手順 1 レプリケーション管理者として接続します。次の手順のプロシージャは、レプリケーション管理者が実行します。**

```
*/
```

```
SET ECHO ON
```

```
SPOOL average_conflictres.out
```

```
CONNECT repadmin/repadmin@orc1.world
```

```
/*
```

**手順 2 加算競合解消メソッドおよび平均競合解消メソッドを定義する前に、競合解消メソッドを適用する表を含むマスター・グループを静止します。単一のマスター・レプリケーション環境においては、マスター・グループの静止は必要ありません。詳細は、6-3 ページの「競合解消メソッドの作成と静止」を参照してください。**

```
*/
```

```
BEGIN
```

```
    DBMS_REPCAT.SUSPEND_MASTER_ACTIVITY (  
        gname => 'hr_repg');
```

```
END;
```

```
/
```

```
/*
```

**手順 3** DBMS\_REPCAT.MAKE\_COLUMN\_GROUP プロシージャを使用して、ターゲット表の列グループを作成します。Oracle の競合解消メソッドはすべて、列グループと呼ばれる論理列グループ化に基づいています。

```
*/  
  
BEGIN  
    DBMS_REPCAT.MAKE_COLUMN_GROUP (  
        sname => 'hr',  
        oname => 'employees',  
        column_group => 'commission_average_cg',  
        list_of_column_names => 'commission_pct');  
END;  
/  
  
/*
```

**手順 4** 特定の表に対し競合解消メソッドを設定するには、DBMS\_REPCAT.ADD\_UPDATE\_RESOLUTION プロシージャを使用します。この例では、sal 列を使用して、ADDITIVE 競合解消メソッドを指定します。

```
*/  
  
BEGIN  
    DBMS_REPCAT.ADD_UPDATE_RESOLUTION (  
        sname => 'hr',  
        oname => 'employees',  
        column_group => 'commission_average_cg',  
        sequence_no => 1,  
        method => 'AVERAGE',  
        parameter_column_name => 'commission_pct');  
END;  
/  
  
/*
```

**手順 5 競合解消メソッドを受信した表に対するレプリケーション・サポートを再生成します。**

```
*/  
  
BEGIN  
    DBMS_REPCAT.GENERATE_REPLICATION_SUPPORT (  
        sname => 'hr',  
        oname => 'employees',  
        type => 'TABLE',  
        min_communication => TRUE);  
END;  
/  
  
/*
```

**手順 6 RESUME\_MASTER\_ACTIVITY プロシージャを使用してレプリケーション・アクティビティを再開します。**

```
*/  
  
BEGIN  
    DBMS_REPCAT.RESUME_MASTER_ACTIVITY (  
        gname => 'hr_repg');  
END;  
/  
  
SET ECHO OFF  
  
SPOOL OFF  
  
/***** END OF SCRIPT *****/
```

## 優先グループによる競合解消メソッド

優先グループでは、特定の列にある値ごとに優先レベルを指定できます。Oracle によって競合が検出されると、優先列で優先順位の低い値を持つ表が、優先順位の高い値を持つ表のデータで更新されます。

次の手順に従って、優先グループによる競合解消メソッドを作成します。

---

---

**注意：** このドキュメントをオンラインで参照している場合は、次の「BEGINNING OF SCRIPT」の行から 6-21 ページの「END OF SCRIPT」の行までのテキストをテキスト・エディタにコピーして編集し、使用している環境に適したスクリプトを作成します。

---

---

```
/***** BEGINNING OF SCRIPT *****/
```

**手順 1 レプリケーション管理者として接続します。次の手順のプロシージャは、レプリケーション管理者が実行します。**

```
*/
```

```
SET ECHO ON
```

```
SPOOL priority_groups_conflictres.out
```

```
CONNECT repadmin/repadmin@orc1.world
```

```
/*
```

**手順 2 優先グループによる競合解消メソッドを定義する前に、競合解消メソッドを適用する表を含むマスター・グループを静止します。単一のマスター・レプリケーション環境においては、マスター・グループの静止は必要ありません。詳細は、6-3 ページの「[競合解消メソッドの作成と静止](#)」を参照してください。**

```
*/
```

```
BEGIN
```

```
    DBMS_REPCAT.SUSPEND_MASTER_ACTIVITY (
        gname => 'hr_repg');
```

```
END;
```

```
/
```

```
/*
```

**手順 3** サイトの優先順位による競合解消メカニズムを使用する列グループに、`job` 列が含まれていることを確認します。このフィールドを既存の列グループに追加するには、`ADD_GROUPED_COLUMN` プロシージャを使用します。列グループを作成していない場合は、`DBMS_REPCAT.MAKE_COLUMN_GROUP` プロシージャを使用して、新しい列グループを作成できます。

```
*/

BEGIN
    DBMS_REPCAT.MAKE_COLUMN_GROUP (
        sname => 'hr',
        oname => 'employees',
        column_group => 'employees_priority_cg',
        list_of_column_names => 'manager_id,hire_date,salary,job_id');
END;
/

/*
```

**手順 4** 優先順位の値を表に割り当てる前に、定義した値を保持する優先グループを作成します。

```
*/

BEGIN
    DBMS_REPCAT.DEFINE_PRIORITY_GROUP (
        gname => 'hr_repg',
        pgroup => 'job_pg',
        datatype => 'VARCHAR2');
END;
/

/*
```

**手順 5** `DBMS_REPCAT.ADD_PRIORITY_datatype` プロシージャは、複数のバージョンが利用できます。利用可能な各データ型にはバージョンがあります (`NUMBER`、`VARCHAR2` など)。利用するすべての表に対して優先順位の値を定義し終えるまで、必要に応じてこのプロシージャを実行します。

**関連項目：** 詳細は、20-16 ページの「[ADD\\_PRIORITY\\_datatype プロシージャ](#)」を参照してください。

```
*/  
  
BEGIN  
    DBMS_REPCAT.ADD_PRIORITY_VARCHAR2(  
        gname => 'hr_repg',  
        pgroup => 'job_pg',  
        value => 'ad_pres',  
        priority => 100);  
END;  
/  
  
BEGIN  
    DBMS_REPCAT.ADD_PRIORITY_VARCHAR2(  
        gname => 'hr_repg',  
        pgroup => 'job_pg',  
        value => 'sa_man',  
        priority => 80);  
END;  
/  
  
BEGIN  
    DBMS_REPCAT.ADD_PRIORITY_VARCHAR2(  
        gname => 'hr_repg',  
        pgroup => 'job_pg',  
        value => 'sa_rep',  
        priority => 60);  
END;  
/  
  
BEGIN  
    DBMS_REPCAT.ADD_PRIORITY_VARCHAR2(  
        gname => 'hr_repg',  
        pgroup => 'job_pg',  
        value => 'pu_clerk',  
        priority => 40);  
END;  
/  
  
BEGIN  
    DBMS_REPCAT.ADD_PRIORITY_VARCHAR2(  
        gname => 'hr_repg',  
        pgroup => 'job_pg',  
        value => 'st_clerk',  
        priority => 20);  
END;  
/  
  
/*
```

**手順 6** PRIORITY GROUP 解消メソッドをレプリケート表に追加します。次の例では、このメソッドは指定した列グループの第 2 の競合解消メソッドです (sequence\_no パラメータ)。

```
*/

BEGIN
    DBMS_REPCAT.ADD_UPDATE_RESOLUTION (
        sname => 'hr',
        oname => 'employees',
        column_group => 'employees_priority_cg',
        sequence_no => 2,
        method => 'PRIORITY GROUP',
        parameter_column_name => 'job_id',
        priority_group => 'job_pg');
END;
/

/*
```

**手順 7** 競合解消メソッドを受信した表に対するレプリケーション・サポートを再生成します。

```
*/

BEGIN
    DBMS_REPCAT.GENERATE_REPLICATION_SUPPORT (
        sname => 'hr',
        oname => 'employees',
        type => 'TABLE',
        min_communication => TRUE);
END;
/

/*
```

**手順 8** RESUME\_MASTER\_ACTIVITY プロシージャを使用してレプリケーション・アクティビティを再開します。

```
*/

BEGIN
    DBMS_REPCAT.RESUME_MASTER_ACTIVITY (
        gname => 'hr_repg');
END;
/
```



```
SET ECHO OFF
```

```
SPOOL OFF
```

```
/***** END OF SCRIPT *****/
```

## サイトの優先順位による競合解消メソッド

サイトの優先順位は、優先グループを特殊化したものです。このため、サイトの優先順位に対応付けられたプロシージャは、優先グループに対応付けられたプロシージャと同じように機能します。競合は、フィールドの値の優先順位ではなく、関係するサイトの優先順位に基づいて解消されます。

たとえば、orc2.world の優先順位が orc1.world よりも高い場合に、この2つのサイトで競合が起これば、orc2.world の値が使用されます。

次の手順に従って、サイトの優先順位による競合解消メソッドを作成します。

---

---

**注意：** このドキュメントをオンラインで参照している場合は、次の「BEGINNING OF SCRIPT」の行から 6-26 ページの「END OF SCRIPT」の行までのテキストをテキスト・エディタにコピーして編集し、使用している環境に適したスクリプトを作成します。

---

---

```
/***** BEGINNING OF SCRIPT *****/
```

**手順 1 レプリケーション管理者として接続します。次の手順のプロシージャは、レプリケーション管理者が実行します。**

```
*/
```

```
SET ECHO ON
```

```
SPOOL site_priority_conflictres.out
```

```
CONNECT repadmin/repadmin@orc1.world
```

```
/*
```

**手順 2** サイトの優先順位による競合解消メソッドを定義する前に、競合解消メソッドを適用する表を含むマスター・グループを静止します。単一のマスター・レプリケーション環境においては、マスター・グループの静止は必要ありません。詳細は、6-3 ページの「**競合解消メソッドの作成と静止**」を参照してください。

```
*/

BEGIN
    DBMS_REPCAT.SUSPEND_MASTER_ACTIVITY (
        gname => 'hr_repg');
END;
/

/*
```

**手順 3** site 列を、サイトの値を格納する表に追加します。DDL をターゲット表に適用するには、DBMS\_REPCAT.ALTER\_MASTER\_REPOBJECT プロシージャを使用します。DDL を発行するのみでレプリケート・オブジェクトが無効になります。

```
*/

BEGIN
    DBMS_REPCAT.ALTER_MASTER_REPOBJECT (
        sname => 'hr',
        oname => 'regions',
        type => 'TABLE',
        ddl_text => 'ALTER TABLE hr.regions ADD (site VARCHAR2(20))');
END;
/

/*
```

**手順 4** 影響のあったオブジェクトのレプリケーション・サポートを再生成します。

```
*/

BEGIN
    DBMS_REPCAT.GENERATE_REPLICATION_SUPPORT (
        sname => 'hr',
        oname => 'regions',
        type => 'TABLE',
        min_communication => TRUE);
END;
/

/*
```

手順 5 行が挿入または更新されたときにサイトのグローバル名を記録するトリガーを作成します。記録された値は、サイトの優先順位によるメソッドに基づいた競合の解消で使われます。トリガーを作成して、マスター・グループに追加するには、DDL を直接実行するのではなく、DBMS\_REPCAT.CREATE\_MASTER\_REPOBJECT プロシージャを使用します。

```
*/

BEGIN
    DBMS_REPCAT.CREATE_MASTER_REPOBJECT (
        gname => 'hr_repg',
        type => 'TRIGGER',
        oname => 'insert_site',
        sname => 'hr',
        ddl_text => 'CREATE TRIGGER hr.insert_site
                    BEFORE
                    INSERT OR UPDATE ON hr.regions FOR EACH ROW
                    BEGIN
                        IF DBMS_REPUTIL.FROM_REMOTE = FALSE THEN
                            SELECT global_name INTO :NEW.SITE FROM GLOBAL_NAME;
                        END IF;
                    END;');

END;
/

/*
```

手順 6 サイトの優先順位による競合解消メカニズムを使用する列グループに、新しい列が含まれていることを確認します。この列を既存の列グループに追加するには、ADD\_GROUPED\_COLUMN プロシージャを使用します。列グループを作成していない場合は、DBMS\_REPCAT.MAKE\_COLUMN\_GROUP プロシージャを使用して、新しい列グループを作成できます。

```
*/

BEGIN
    DBMS_REPCAT.MAKE_COLUMN_GROUP (
        sname => 'hr',
        oname => 'regions',
        column_group => 'regions_sitepriority_cg',
        list_of_column_names => 'region_id,region_name,site');

END;
/

/*
```

**手順 7 サイトの優先順位の値をレプリケート環境のサイトに割り当てる前に、定義した値を保持するサイトの優先グループを作成します。**

```
*/  
  
BEGIN  
    DBMS_REPCAT.DEFINE_SITE_PRIORITY (  
        gname => 'hr_repg',  
        name => 'regions_sitepriority_pg');  
END;  
/  
  
/*
```

**手順 8 DBMS\_REPCAT.ADD\_SITE\_PRIORITY\_SITE プロシージャを使用して、レプリケーション環境のサイトごとに優先順位の値を定義します。レプリケーション環境の各サイトに対し、サイトの優先順位の値を定義し終えるまで、必要に応じてこのプロシージャを実行します。**

```
*/  
  
BEGIN  
    DBMS_REPCAT.ADD_SITE_PRIORITY_SITE (  
        gname => 'hr_repg',  
        name => 'regions_sitepriority_pg',  
        site => 'orcl.world',  
        priority => 100);  
END;  
/  
  
BEGIN  
    DBMS_REPCAT.ADD_SITE_PRIORITY_SITE (  
        gname => 'hr_repg',  
        name => 'regions_sitepriority_pg',  
        site => 'orc2.world',  
        priority => 50);  
END;  
/  
  
BEGIN  
    DBMS_REPCAT.ADD_SITE_PRIORITY_SITE (  
        gname => 'hr_repg',  
        name => 'regions_sitepriority_pg',  
        site => 'orc3.world',  
        priority => 25);
```

```
END;  
/  
  
/*
```

**手順 9** サイトに優先順位の値を割り当てた後、SITE PRIORITY 解消メソッドをレプリケート表に追加します。次の例では、このメソッドは指定した列グループの第 3 の競合解消メソッドです (sequence\_no パラメータ)。

```
*/  
  
BEGIN  
  DBMS_REPCAT.ADD_UPDATE_RESOLUTION (  
    sname => 'hr',  
    oname => 'regions',  
    column_group => 'regions_sitepriority_cg',  
    sequence_no => 1,  
    method => 'SITE PRIORITY',  
    parameter_column_name => 'site',  
    priority_group => 'regions_sitepriority_pg');  
END;  
/  
  
/*
```

**手順 10** 競合解消メソッドを受信した表に対するレプリケーション・サポートを再生成します。

```
*/  
  
BEGIN  
  DBMS_REPCAT.GENERATE_REPLICATION_SUPPORT (  
    sname => 'hr',  
    oname => 'regions',  
    type => 'TABLE',  
    min_communication => TRUE);  
END;  
/  
  
/*
```

**手順 11** RESUME\_MASTER\_ACTIVITY プロシージャを使用してレプリケーション・アクティビティを再開します。

```
*/  
  
BEGIN  
    DBMS_REPCAT.RESUME_MASTER_ACTIVITY (  
        GNAME => 'hr_repg');  
END;  
/  
  
SET ECHO OFF  
  
SPOOL OFF  
  
/***** END OF SCRIPT *****/
```

## 一意性競合の競合解消メソッドの作成

レプリケーション環境では、一意制約に関してデータの挿入を主な原因とする競合が発生する場合があります。重複する行の削除がビジネス・ルールで許可されている場合は、行の削除を Oracle の組込み競合解消メソッドで定義します。

ただし、競合する値を変更して、一意制約に違反しないようにする方が一般的です。競合する値を変更すれば、重要なデータが失われることはありません。Oracle の組込みの一意性競合解消メソッドを使用すると、値にサイト名や順序番号を付けることで、競合する値を一意にできます。

この他に、一意性競合解消メソッドの追加コンポーネントとして、通知機能があります。競合する情報は Oracle によって変更されるので、その情報を表に挿入できます。ただし、その場合、競合を通知して、その通知に基づいて競合を分析し、レコードを削除するのか、データを他のレコードにマージするのか、または競合するデータにまったく新しい値を定義するのかを決定することが必要です。

一意性競合解消メソッドでは列内の競合は UNIQUE 制約により検出され、解消されます。この項の例では、email 列に一意性制約 emp\_email\_uk が設定されている hr サンプル・スキーマの employees 表を使用します。

---

---

**注意：** 列に一意性制約解消メソッドを追加するには、列の一意索引名が一意制約名または主キー制約名と一致する必要があります。

---

---

次の手順に従って、一意性競合解消メソッドを作成します。

---

---

**注意：** このドキュメントをオンラインで参照している場合は、次の「BEGINNING OF SCRIPT」の行から 6-32 ページの「END OF SCRIPT」の行までのテキストをテキスト・エディタにコピーして編集し、使用している環境に適したスクリプトを作成します。

---

---

```
/***** BEGINNING OF SCRIPT *****/
```

#### 手順 1 レプリケーション管理者として接続します。

```
*/  
  
SET ECHO ON  
  
SPOOL unique_conflictres.out  
  
CONNECT repadmin/repadmin@orc1.world  
  
/*
```

#### 手順 2 一意性競合解消メソッドを定義する前に、競合解消メソッドを適用する表を含むマスター・グループを静止します。

```
*/  
  
BEGIN  
    DBMS_REPCAT.SUSPEND_MASTER_ACTIVITY (  
        gname => 'hr_repg');  
END;  
/  
  
/*
```

**手順 3 通知機能から受信したメッセージを格納する表を作成します。この例で使用する表の名前は、conf\_report です。**

```
*/  
  
BEGIN  
  DBMS_REPCAT.EXECUTE_DDL (  
    gname => 'hr_repg',  
    ddl_text => 'CREATE TABLE hr.conf_report (  
      line NUMBER(2),  
      txt VARCHAR2(80),  
      timestamp DATE,  
      table_name VARCHAR2(30),  
      table_owner VARCHAR2(30),  
      conflict_type VARCHAR2(7))');  
END;  
/  
  
/*
```

**手順 4 手順 3 で作成した表の所有者として接続します。**

```
*/  
  
CONNECT hr/hr@orc1.world  
  
/*
```

**手順 5 競合が検出されたときに conf\_report 表へ通知を送信するパッケージを作成します。この例で使用するパッケージの名前は、notify です。**

**関連項目：** 付録 B「ユーザー定義の競合解消メソッド」では、このスクリプトで作成された競合解消通知パッケージを説明しています。

```
*/  
  
CREATE OR REPLACE PACKAGE notify AS  
  FUNCTION emp_unique_violation (email IN OUT VARCHAR2,  
    discard_new_values IN OUT BOOLEAN)  
  RETURN BOOLEAN;  
END notify;  
/  
  
CREATE OR REPLACE PACKAGE BODY notify AS  
  TYPE message_table IS TABLE OF VARCHAR2(80) INDEX BY BINARY_INTEGER;  
  PROCEDURE report_conflict(conflict_report IN MESSAGE_TABLE,  
    report_length IN NUMBER,
```



```

conflict_time IN DATE,
conflict_table IN VARCHAR2,
table_owner IN VARCHAR2,
conflict_type IN VARCHAR2) IS
BEGIN
    FOR idx IN 1..report_length LOOP
        BEGIN
            INSERT INTO hr.conf_report
            (line, txt, timestamp, table_name, table_owner, conflict_type)
            VALUES (idx, SUBSTR(conflict_report(idx),1,80), conflict_time,
                conflict_table, table_owner, conflict_type);
            EXCEPTION WHEN others THEN NULL;
        END;
    END LOOP;
END report_conflict;

FUNCTION emp_unique_violation(email IN OUT VARCHAR2,
    discard_new_values IN OUT BOOLEAN)
RETURN BOOLEAN IS
    local_node VARCHAR2(128);
    conf_report MESSAGE_TABLE;
    conf_time DATE := SYSDATE;
BEGIN
    BEGIN
        SELECT global_name INTO local_node FROM global_name;
        EXCEPTION WHEN others THEN local_node := '?';
    END;
    conf_report(1) := 'UNIQUENESS CONFLICT DETECTED IN EMPLOYEES ON ' ||
        TO_CHAR(conf_time, 'MM-DD-YYYY HH24:MI:SS');
    conf_report(2) := ' AT NODE ' || local_node;
    conf_report(3) := 'ATTEMPTING TO RESOLVE CONFLICT USING' ||
        ' APPEND SITE NAME METHOD';
    conf_report(4) := 'EMAIL: ' || email;
    conf_report(5) := NULL;
    report_conflict(conf_report,5,conf_time,'employees','hr','UNIQUE');
    discard_new_values := FALSE;
    RETURN FALSE;
END emp_unique_violation;
END notify;

/

/*

```

**手順 6 レプリケーション管理者として接続します。**

```
*/  
  
CONNECT repadmin/repadmin@orcl.world  
  
/*
```

**手順 7 すべてのマスター・サイトで通知機能を利用するために、手順 5 で作成したパッケージをレプリケーション環境のすべてのマスター・サイトにレプリケートします。**

```
*/  
  
BEGIN  
    DBMS_REPCAT.CREATE_MASTER_REPOBJECT (  
        gname => 'hr_repg',  
        type =>  'PACKAGE',  
        oname => 'notify',  
        sname => 'hr');  
END;  
/  
  
BEGIN  
    DBMS_REPCAT.CREATE_MASTER_REPOBJECT (  
        gname => 'hr_repg',  
        type =>  'PACKAGE BODY',  
        oname => 'notify',  
        sname => 'hr');  
END;  
/  
  
/*
```

**手順 8 競合解消メソッドの一環として通知機能を追加します。競合の発生を通知するのみの機能でもかまいません。次の例では、USER FUNCTION として通知機能を追加しています。**

```
*/  
  
BEGIN  
    DBMS_REPCAT.ADD_UNIQUE_RESOLUTION(  
        sname => 'hr',  
        oname => 'employees',  
        constraint_name => 'emp_email_uk',  
        sequence_no => 1,  
        method => 'USER FUNCTION',  
        comment => 'Notify DBA',  
        parameter_column_name => 'email',  
        function_name => 'hr.notify.emp_unique_violation');  
END;
```

```
END;  
/
```

```
/*
```

**手順 9** 表に実際の競合解消メソッドを追加します。次の例では、APPEND SITE NAME 一意性競合解消メソッドをレプリケート表へ追加しています。

```
*/
```

```
BEGIN
```

```
  DBMS_REPCAT.ADD_UNIQUE_RESOLUTION(  
    sname => 'hr',  
    oname => 'employees',  
    constraint_name => 'emp_email_uk',  
    sequence_no => 2,  
    method => 'APPEND SITE NAME',  
    parameter_column_name => 'email');
```

```
END;  
/
```

```
/*
```

**手順 10** 競合解消メソッドを受信した表に対するレプリケーション・サポートを再生成します。

```
*/
```

```
BEGIN
```

```
  DBMS_REPCAT.GENERATE_REPLICATION_SUPPORT (  
    sname => 'hr',  
    oname => 'employees',  
    type => 'TABLE',  
    min_communication => TRUE);
```

```
END;  
/
```

```
/*
```

**手順 11** RESUME\_MASTER\_ACTIVITY プロシージャを使用してレプリケーション・アクティビティを再開します。

```
*/  
  
BEGIN  
    DBMS_REPCAT.RESUME_MASTER_ACTIVITY (  
        gname => 'hr_repg');  
END;  
/  
  
SET ECHO OFF  
  
SPOOL OFF  
  
/***** END OF SCRIPT *****/
```

## 削除の競合防止メソッドの作成

行の削除を行う場合は、比較する 2 つの値がある更新の競合とは違って、1 つの値しか存在しないので、前述の更新の競合解消メソッドを適用できません。

レプリケーション環境で行を削除する最善の方法は、削除対象の行をマークし、マークされたすべてのレコードを表から定期的にパージして競合を回避することです。物理的に行が削除されるのではなく、比較される 2 つの値が存在するので、競合が発生しても、すべてのマスター・サイトでデータを収束させることができます。ただし、他のエラーが発生していないことを前提とします。データが収束したことを確認した後、レプリケート・パージ・プロシージャを使用して、マークされた行をパージします。

データベース用のフロントエンド・アプリケーションの開発では、削除対象としてマークされた行にフィルタをかけた方が便利です。このようにすると、ユーザーにはその行が物理的に削除されたように見えます。フィルタをかけるには、単に SELECT 文で削除対象としてマークされた行をデータ・セットから除外します。

たとえば、次のように SELECT 文を指定して、現在の従業員をリストします。

```
SELECT * FROM hr.locations WHERE remove_date IS NULL;
```

次に、削除の競合を回避するレプリケート表を作成するメソッドを示します。また、削除対象としてマークされたレコードをパージするために、プロシージャ・レプリケーションを使用する方法も学びます。

次の手順に従って、削除の競合の競合防止メソッドを作成します。

---

---

**注意：** このドキュメントをオンラインで参照している場合は、次の「BEGINNING OF SCRIPT」の行から 6-37 ページの「END OF SCRIPT」の行までのテキストをテキスト・エディタにコピーして編集し、使用している環境に適したスクリプトを作成します。

---

---

```
/***** BEGINNING OF SCRIPT *****/
```

**手順 1 レプリケーション管理者としてマスター定義サイトに接続します。**

```
*/  
  
SET ECHO ON  
  
SPOOL delete_conflictres.out  
  
CONNECT repadmin/repadmin@orc1.world  
  
/*
```

**手順 2 競合解消メソッドを適用する表が含まれたマスター・グループを静止します。**

```
*/  
  
BEGIN  
    DBMS_REPCAT.SUSPEND_MASTER_ACTIVITY (  
        gname => 'hr_repg');  
END;  
/  
  
/*
```

**手順 3** 削除された記録のマークを格納するレプリケート表に列を追加します。タイムスタンプを使用して削除の記録をマークすることをお勧めします（タイムスタンプでは、削除の記録がマークされた日時が示されます）。タイムスタンプを使用するということで、新しい列のデータ型は `DATE` となります。`remove_date` 列を既存のレプリケート表に追加する場合は、`DBMS_REPCAT.ALTER_MASTER_REPOBJECT` プロシージャを使用します。

```
*/  
  
BEGIN  
    DBMS_REPCAT.ALTER_MASTER_REPOBJECT (  
        sname => 'hr',  
        oname => 'locations',  
        type => 'TABLE',  
        ddl_text => 'ALTER TABLE hr.locations ADD (remove_date DATE)');  
END;  
/  
  
/*
```

**手順 4** 変更された表のレプリケーション・サポートを再生成します。

```
*/  
  
BEGIN  
    DBMS_REPCAT.GENERATE_REPLICATION_SUPPORT (  
        sname => 'hr',  
        oname => 'locations',  
        type => 'TABLE',  
        min_communication => TRUE);  
END;  
/  
  
/*
```

**手順 5 レプリケーション環境のすべてのマスター・サイトにレプリケートされるパッケージを作成します。このパッケージは、指定した表からマーク済の記録をすべてパージします。**

```

*/

BEGIN
  DBMS_REPCAT.CREATE_MASTER_REPOBJECT (
    gname => 'hr_repg',
    type => 'PACKAGE',
    oname => 'purge',
    sname => 'hr',
    ddl_text => 'CREATE OR REPLACE PACKAGE hr.purge AS
                PROCEDURE remove_locations(purge_date DATE);
                END;');

END;
/

BEGIN
  DBMS_REPCAT.CREATE_MASTER_REPOBJECT (
    gname => 'hr_repg',
    type => 'PACKAGE BODY',
    oname => 'purge',
    sname => 'hr',
    ddl_text => 'CREATE OR REPLACE PACKAGE BODY hr.purge AS
                PROCEDURE remove_locations(purge_date IN DATE) IS
                BEGIN
                  DBMS_REPUTIL.REPLICATION_OFF;
                  LOCK TABLE hr.locations IN EXCLUSIVE MODE;
                  DELETE hr.locations WHERE remove_date IS NOT NULL
                     AND remove_date < purge_date;
                  DBMS_REPUTIL.REPLICATION_ON;
                EXCEPTION WHEN others THEN
                  DBMS_REPUTIL.REPLICATION_ON;
                END;
                END;');

END;
/

/*
```

**手順 6** 各パッケージおよびパッケージ本体のレプリケーション・サポートを生成します。レプリケーション・サポートの生成後、使用者用のシノニムが作成されて、レプリケート・オブジェクトとしてマスター・グループに追加されます。このシノニムには、`defer_purge.remove_locations` というラベルが付けられます。

```
*/

BEGIN
    DBMS_REPCAT.GENERATE_REPLICATION_SUPPORT (
        sname => 'hr',
        oname => 'purge',
        type => 'PACKAGE',
        min_communication => TRUE);
END;
/

BEGIN
    DBMS_REPCAT.GENERATE_REPLICATION_SUPPORT (
        sname => 'hr',
        oname => 'purge',
        type => 'PACKAGE BODY',
        min_communication => TRUE);
END;
/

/*
```

**手順 7** 個別の端末ウィンドウで、他のすべてのマスター・サイトにおける任意の管理要求を手動でプッシュします。複数の手順を含む管理操作も存在するため、`DBMS_REPCAT` パッケージの `DO_DEFERRED_REPCAT_ADMIN` プロシージャを数回実行します。次に例を示します。

```
BEGIN
    DBMS_REPCAT.DO_DEFERRED_REPCAT_ADMIN (
        gname => 'hr_repg',
        all_sites => FALSE);
END;
/

/*

PAUSE Press <RETURN> to continue when you have verified that there are no pending
administrative requests in the DBA_REPCATLOG data dictionary view.

/*
```



**手順 8 RESUME\_MASTER\_ACTIVITY プロシーダを使用してレプリケーション・アクティビティを再開します。**

```
*/  
  
BEGIN  
    DBMS_REPCAT.RESUME_MASTER_ACTIVITY (  
        gname => 'hr_repg');  
END;  
/  
  
SET ECHO OFF  
  
SPOOL OFF  
  
/***** END OF SCRIPT *****/
```

## 動的所有権の競合防止メソッド

この項では、アプリケーションの競合を回避するための先進の設計方法について説明しています。トークン渡しとして知られているこの方法は、次に説明するワークフロー・メソッドに似ています。この項では、このメソッドを使用してすべての行の所有権を管理する方法を説明していますが、このメソッドを変更して行内の個々の列グループの所有権を管理することもできます。

ワークフローおよびトークン渡しの両方とも、データの動的所有権に対応しています。動的所有権の行の更新は一度に 1 サイトのみ可能ですが、サイト間での行の所有権の受渡しは可能です。ワークフローおよびトークン渡しの両方とも、1 つ以上の識別子列の値を使用して、現在、行の更新が許可されている者を特定します。

## ワークフロー

ワークフロー・パーティション分割では、サイト間で受け渡されるデータの所有権を考慮します。行の現在の所有者のみが、識別子列の値を変更することで行の所有権を他のサイトに渡せます。

受注、出荷および請求サイト別に簡単な例を紹介します。ここでは、受注のステータスを表示するのに識別子列を使用します。行の更新を許可されたサイトはステータスで特定します。受注サイトのユーザーは受注の入力後、この行のステータスを **ship** へと更新します。所有権が出荷サイトに渡されるため、受注サイトのユーザーはこの行を修正できません。

受注の出荷後、出荷サイトのユーザーは行のステータスを **bill** へと更新するため、所有権は請求サイトその他へ渡されます。

競合を回避するために、動的データ所有権を実装するアプリケーションが、次の条件を満たすことを確認します。

- 行の所有者のみが行を更新可能。
- ある行を 2 つ以上のサイトで共有することは不可。
- すべてのサイトにおける順序付けの競合が解消可能。

ワークフロー・パーティション分割では、行のカレント所有者のみが識別子列を更新することで、行の所有権を次のサイトに渡すことができます。他のサイトが所有権を放棄しないかぎり別のサイトに所有権が渡されることはありません。このため、所有者が 2 人以上となることはありません。

作業の流れは順序付けられているため、作業の流れの中で最後に生じたサイトからの変更を適用することで、順序付けの競合が解消できます。優先順位による競合解消メソッドを使用することで、順序付けの競合が解消できます。この方法では、優先順位の値が各ワークフロー・プロセスのステップとともに増加します。優先順位による競合解消メソッドでは、2 つ以上のマスター・サイトで優先順位の値が常に増加しても問題なくまとめることができます。

## トークン渡し

トークン渡しでは、より一般化されたアプローチを使用してこれらの基準を満たします。トークン渡しを実装する場合は、識別子列ではなくレプリケート表が所有者およびエポック列を保有します。所有者列には、現在、行を所有サイトのグローバル・データベース名を格納します。

トークン渡しメカニズムの設計を終了すると、これを使用してワークフローなど各種データ所有権の動的パーティション化を実装できます。

トークン渡しを自動的に実装するようにアプリケーションを設計します。このアプリケーション以外から所有者またはエポック列を更新しないでください。

行の更新では、アプリケーションは次の条件に従います。

**手順 1 行のカレント所有者の位置を特定します。**

**手順 2 行をロックして、所有権変更時の更新を防止します。**

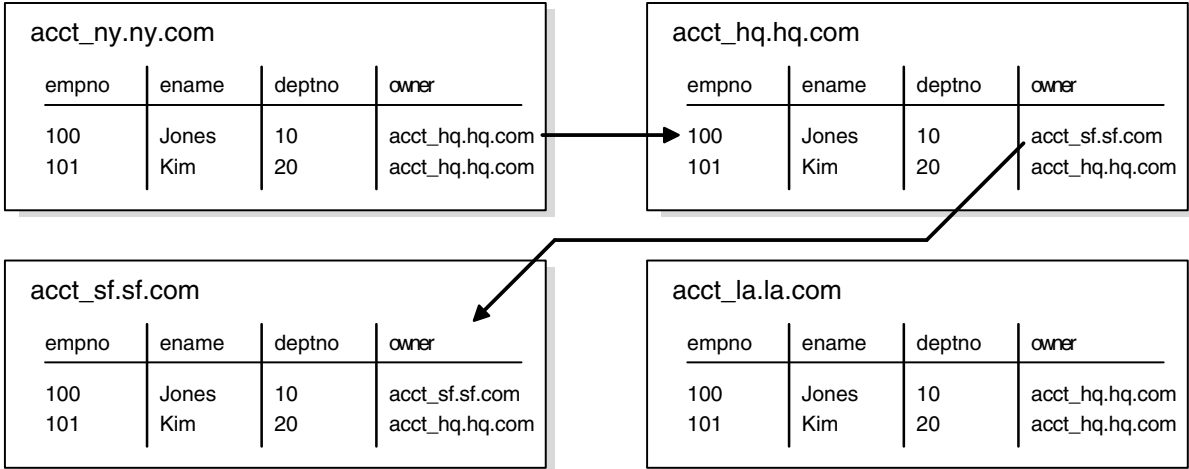
**手順 3 行の所有権を設定します。**

**手順 4 更新を実行します。Oracle は、トランザクションのコミット時にロックを解放します。**

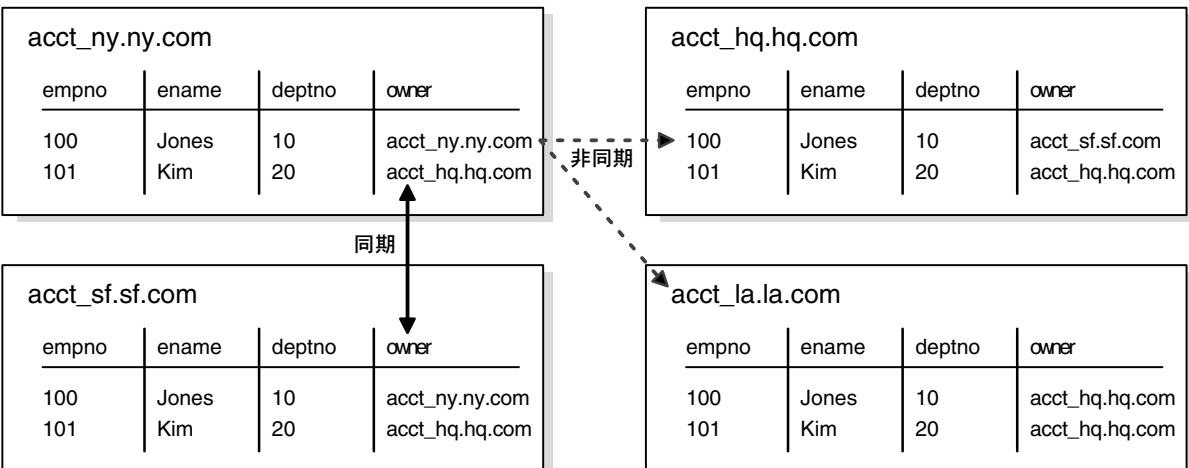
たとえば図 6-1 では、従業員 100 の所有権を `acct_sf` データベースから `acct_ny` データベースへ渡す方法を説明しています。

図 6-1 トークンの取込み

手順1. 真の所有者の識別



手順2. 所有権の取込みとブロードキャスト変更



## 行の所有者の検索

所有権を取得する場合、acct\_ny データベースは単純な再帰的アルゴリズムを使用して、行の所有者を検索します。このアルゴリズムのサンプル・コードを次に示します。

```
-- Sample code for locating the token owner.
-- This is for a table TABLE_NAME with primary key PK.
-- Initial call should initialize loc_epoch to 0 and loc_owner
-- to the local global name.
get_owner(PK IN primary_key_type, loc_epoch IN OUT NUMBER,
          loc_owner IN OUT VARCHAR2)
{
  -- use dynamic SQL (dbms_sql) to perform a select similar to
  -- the following:
  SELECT owner, epoch into rmt_owner, rmt_epoch
    FROM TABLE_NAME@loc_owner
   WHERE primary_key = PK FOR UPDATE;
  IF rmt_owner = loc_owner AND rmt_epoch >= loc_epoch THEN
    loc_owner := rmt_owner;
    loc_epoch := rmt_epoch;
    RETURN;
  ELSIF rmt_epoch >= loc_epoch THEN
    get_owner(PK, rmt_epoch, rmt_owner);
    loc_owner := rmt_owner;
    loc_epoch := rmt_epoch;
    RETURN;
  ELSE
    raise_application_error(-20000, 'No owner for row');
  END IF;
}
```

## 所有権の取得

行の所有権を検索した後、acct\_ny サイトは次の手順に従って acct\_sf サイトから所有権を取得します。

**手順 1** sf サイトの行をロックして、所有権交換で生じる変更を防止します。

**手順 2** sf サイトおよび ny サイトの所有者情報を同期式で更新します。この操作により常に 1 つのサイトのみが所有者となります。sf サイトでの更新を DBMS\_REPUTIL.REPLICATION\_OFF を使用してレプリケートしないでください。手順 4 の ny サイトでレプリケートされた所有権の変更は、影響を一切受けない sf サイトなど、レプリケーション環境の他のすべてのサイトに伝播されます。

**手順 3** 新しい所有者サイト、`ny` の行情報を、カレント所有者サイト、`sf` の情報で更新します。最新のデータであることが保証されます。今回は、`ny` サイトでの変更をレプリケートしないでください。`sf` サイトでこのデータに対して行うキューの変更は、通常の方法で他のすべてのサイトに伝播されます。`sf` の変更が `ny` に伝播されても、次の項目で説明するエポック番号の値を理由として無視されます。

**手順 4** 新しい所有者サイトのエポック番号を以前のサイトの値に 1 を加えた値で更新します。この更新は新しい所有者に対してのみ実行します。更新後は、この更新内容を非同期式で他のマスター・サイトへ伝播します。新しい所有者サイトのエポック番号を増やすことで順序の競合を防止します。

`sf` の変更（前述の手順 2 で示した遅延キューに対する）が `ny` サイトに伝播されたとき、`ny` サイトはこれを無視します。これは、同一のデータに対するエポック番号が、`ny` サイトのエポック番号より小さいためです。

その他、`hq` サイトが `ny` の変更を受信した後に `sf` の変更を受信したものの、`ny` サイトからの変更のエポック番号が大きいため `hq` サイトが `sf` の変更を無視するような例も考えられます。

## 変更の適用

更新を自動的に行うために、アプリケーションにはトークン渡しの実装します。このアプリケーション以外から所有者またはエポック列を更新しないでください。所有権の変更時に取り込んだロックは、実際に更新するときに解放されます。変更された情報は、更新された所有者情報およびエポック情報とともに、通常の方法により非同期式で他のサイトへ伝播されます。

## 競合解消の監査

Oracle によって更新、削除または一意性競合が検出され、正常に解消された場合は、ALL\_REPRESOLUTION\_STATISTICS データ・ディクショナリ・ビューを問い合わせることによって、競合解消に使用された方法の情報を表示できます。このビューは、競合に関連する表に対して競合解消統計収集を有効化した場合にのみ更新されます。

**関連項目：** 詳細は、25-34 ページの「[ALL\\_REPRESOLUTION\\_STATISTICS](#) ビュー」を参照してください。

## 競合解消統計の収集

DBMS\_REPCAT パッケージの REGISTER\_STATISTICS プロシージャを使用して、正常に実行された、表の更新、削除および一意性の競合解消の情報を収集します。次の例では、hr スキーマ内の従業員表の統計が収集されます。

```
BEGIN
    DBMS_REPCAT.REGISTER_STATISTICS (
        sname => 'hr',
        oname => 'employees');
END;
/
```

## 競合解消統計の表示

ある表に対して REGISTER\_STATISTICS をコールすると、その表で正常に解消された各競合は ALL\_REPRESOLUTION\_STATISTICS データ・ディクショナリ・ビューに記録されます。解消不可能な競合の情報は、オブジェクトが登録されているかどうかに関係なく、通常、DEFERROR ビューに記録されます。

**関連項目：** 詳細は、25-34 ページの「[ALL\\_REPRESOLUTION\\_STATISTICS](#) ビュー」および 27-3 ページの「[DEFERROR](#) ビュー」を参照してください。

## 競合解消統計の取消し

正常に実行された、表の更新、削除および一意性の競合解消の情報収集が不要になった場合は、DBMS\_REPCAT パッケージの CANCEL\_STATISTICS プロシージャを使用します。次の例では、hr スキーマ内の employees 表に収集された統計を取り消します。

```
BEGIN
    DBMS_REPCAT.CANCEL_STATISTICS (
        sname => 'hr',
        oname => 'employees');
END;
/
```

## 統計情報の消去

正常に実行された、更新、削除および一意性の競合解消の情報を記録するように表を登録している場合は、DBMS\_REPCAT パッケージの PURGE\_STATISTICS プロシージャをコールすることによって、DBA\_REPRESOLUTION\_STATISTICS データ・ディクショナリ・ビューからこの情報を削除できます。

次の例では、1 月 1 日～3 月 31 日の、employees 表に対する挿入、更新および削除により解消した競合についての統計がパージされます。

```
BEGIN
    DBMS_REPCAT.PURGE_STATISTICS (
        sname => 'hr',
        oname  => 'employees',
        start_date => '01-JAN-2001',
        end_date  => '31-MAR-2001');
END;
/
```





# 第 II 部

---

## レプリケーション環境の管理と監視

第 II 部には、レプリケーション・マネージメント API を使用したレプリケーション環境の管理、データ・ディクショナリを使用したレプリケーション環境の監視に関する指示が含まれています。

第 II 部は次の章で構成されています。

- [第 7 章「マスター・レプリケーション環境の管理」](#)
- [第 8 章「マテリアライズド・ビュー・レプリケーション環境の管理」](#)
- [第 9 章「レプリケーション・オブジェクトおよびキューの管理」](#)
- [第 10 章「レプリケーション環境の監視」](#)



---

## マスター・レプリケーション環境の管理

事業の拡大や縮小、緊急事態などにより、データ配信を変更する場合、レプリケーション環境の構成も変更する必要があります。この章では、レプリケーション環境のマスター・サイトの管理を説明します。この項では、マスター・サイトの変更と再構成を中心に説明しています。

この章では、次の項目を説明します。

- マスター定義サイトの変更
- 新しいマスター・サイトの追加
- マスター・グループからマスター・サイトの削除
- データ・ディクショナリ・ビュー内のコメント・フィールドの更新
- プロシージャ・レプリケーションの使用

## マスター定義サイトの変更

ほとんどのレプリケーション管理タスクは、マスター定義サイトのみで実行できます。マスター定義サイトを他のマスター・サイトへ移動する場合は、DBMS\_REPCAT パッケージの RELOCATE\_MASTERDEF プロシージャを使用します。この API は、マスター定義サイトが利用できなくなったために、新しいマスター定義サイトを指定する必要がある場合に使用します（7-3 ページの「[オプション 2: 旧マスター定義サイトが利用不可](#)」を参照してください）。

### オプション 1: すべてのマスター・サイトが利用可能

すべてのマスター・サイトが利用可能な場合は、この項のアクションを実行して、マスター定義サイトを変更します。これらのアクションを実行するには、次の要件を満たす必要があります。

**実行者:** レプリケーション管理者

**実行場所:** いずれかのマスター・サイト

**レプリケーションの状態:** 実行中（静止中ではない）

次の手順に従います。

**手順 1 レプリケーション管理者としてマスター・サイトに接続します。**

```
CONNECT repadmin/repadmin@orc1.world
```

**手順 2 マスター定義サイトを再配置します。**

```
BEGIN
    DBMS_REPCAT.RELOCATE_MASTERDEF (
        gname => 'hr_repg',
        old_masterdef => 'orc1.world',
        new_masterdef => 'orc2.world',
        notify_masters => TRUE,
        include_old_masterdef => TRUE);
END;
/
```

## オプション 2: 旧マスター定義サイトが利用不可

旧マスター・サイトが利用不可の場合は、この項のアクションを実行して、マスター定義サイトを変更します。これらのアクションを実行するには、次の要件を満たす必要があります。

**実行者:** レプリケーション管理者

**実行場所:** いずれかのマスター・サイト

**レプリケーションの状態:** 通常

次の手順に従います。

**手順 1** レプリケーション管理者としてマスター・サイトに接続します。

```
CONNECT repadmin/repadmin@orc4.world
```

**手順 2** マスター定義サイトを再配置します。

```
BEGIN
  DBMS_REPCAT.RELOCATE_MASTERDEF (
    gname => 'hr_repg',
    old_masterdef => 'orc1.world',
    new_masterdef => 'orc2.world',
    notify_masters => TRUE,
    include_old_masterdef => FALSE);
END;
/
```

## 新しいマスター・サイトの追加

レプリケーション環境が拡張されるにつれ、新しいマスター・サイトをマスター・グループに追加することが必要になります。新しいマスター・サイトは実行中のマスター・グループへも静止中のマスター・グループへも追加できます。マスター・グループが静止中ではない場合は、ユーザーは、新しいマスター・サイトが追加されときのデータに対して、データ操作言語 (DML) を操作できます。ただし、マスター・グループが静止中ではない場合は、新しいマスター・サイトを追加するときに、より管理的なアクションが要求されます。

---

**注意：** 循環依存のある表または自己参照型制約のある表を含むマスター・グループにマスター・サイトを追加する場合は、表の定義を事前に作成して、新しいマスター・サイトに手動でデータをロードします。循環依存の例を次に示します。表 A には表 B に対する外部キー制約があり、表 B には表 A に対する外部キー制約があります。

---

該当する項の指示に従って、新しいマスター・サイトをマスター・グループに追加します。

- [マスター・グループを静止することなく、新しいマスター・サイトを追加](#)
- [静止中のマスター・グループに新しいマスター・サイトを追加](#)

## マスター・グループを静止することなく、新しいマスター・サイトを追加

この項には、新しいマスター・サイトを静止中ではない既存のマスター・グループへ追加するプロシージャが含まれています。これらの新しいサイトは、他のレプリケーション・グループのレプリケーション・サイト（マスター・サイトまたはマテリアライズド・ビュー・サイト）に存在する場合も存在しない場合もあります。

マスター・グループを静止しないで新しいマスター・サイトを追加する場合は、次のいずれかの方法が使用できます。

- 全データベースのエクスポートまたはインポート、あるいは変更ベースのリカバリを使用して、現在、レプリケーション・グループを保有していない新しいマスター・サイトを追加します。指示の詳細は、7-8 ページの「[全データベースのエクスポートまたはインポートの使用、あるいは変更ベースのリカバリの使用](#)」を参照してください。
- オブジェクトレベルのエクスポートまたはインポートを使用して、すでに他のレプリケーション・グループを保有する新しいマスター・サイトを追加、またはレプリケーション・グループを保有していない新しいマスター・サイトを追加します。指示の詳細は、7-19 ページの「[オブジェクトレベルのエクスポート / インポートの使用](#)」を参照してください。

全データベースのエクスポートまたはインポート、および変更ベースのリカバリを使用して、マスター定義サイトのすべてのレプリケーション・グループを新しいマスター・サイトに追加します。この方法を使用する場合は、次の条件が適用されます。

- 新しいマスター・サイトは既存のレプリケーション・グループを保有できません。
- マスター定義サイトは、マテリアライズド・ビュー・グループを保有できません。
- マスター定義サイトはすべてのマスター・グループについて同じである必要があります。これらのマスター・グループの1つ以上で、異なるマスター定義サイトを保有する場合は、全データベースのエクスポートまたはインポート、あるいは変更ベースのリカバリは使用しません。この場合は、オブジェクトレベルのエクスポートまたはインポートを使用します。
- 拡張プロセスが完了したときは、新しいマスター・サイトには、マスター定義サイトのすべてのレプリケーション・グループを含める必要があります。すなわち、マスター定義サイトのマスター・グループのサブセットを新しいマスター・サイトへ追加することはできません。グループのすべてを追加する必要があります。

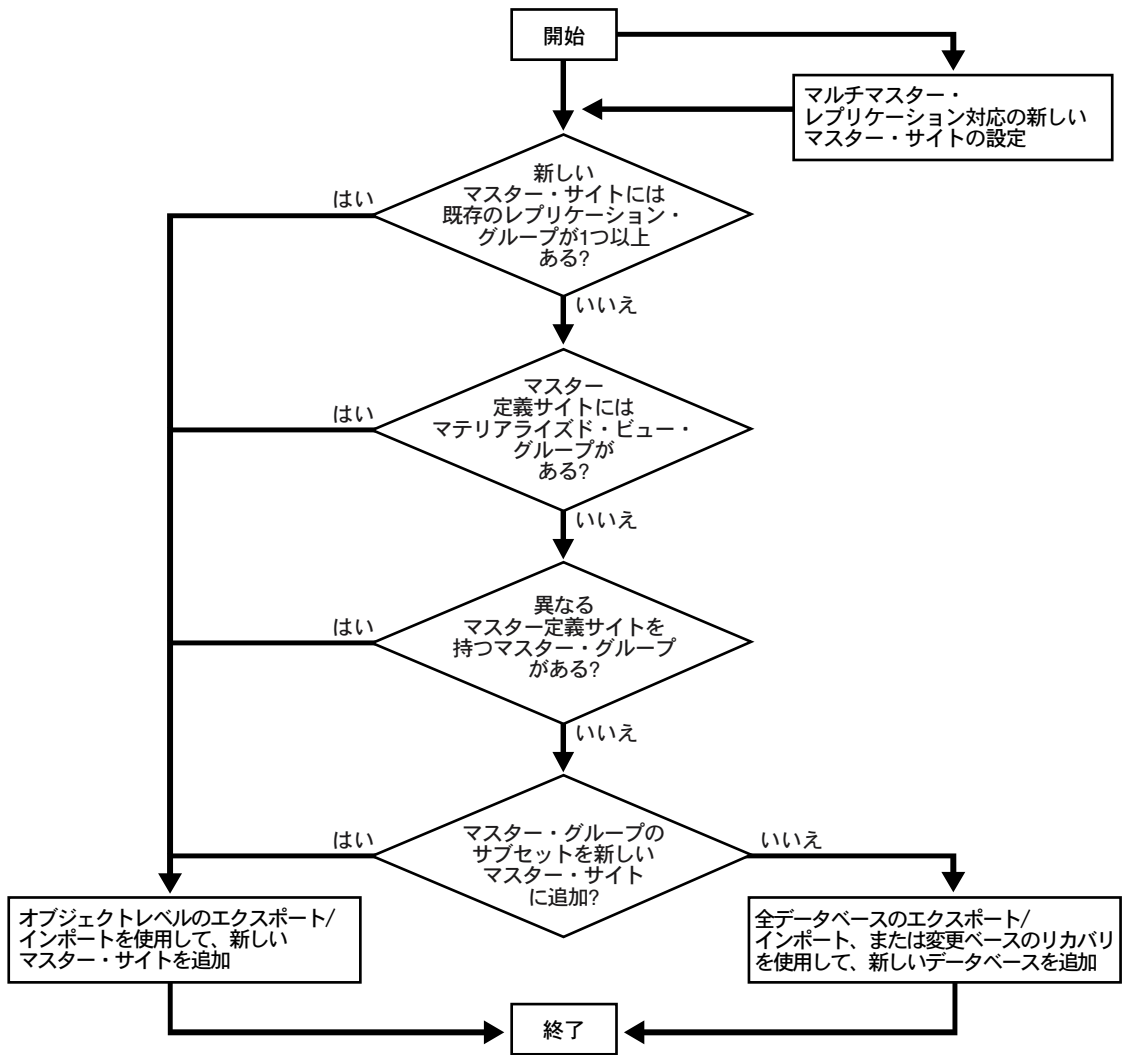
---

**注意：** 変更ベースのリカバリを使用する場合は、既存のマスター・サイトおよび新しいマスター・サイトが同一のオペレーティング・システムで実行されていることが必要です（リリースの相違は許容されます）。この条件は、全データベースのエクスポートまたはインポートへは適用されません。

---

使用している環境がこれらの条件の一部を満たすことができない場合は、オブジェクトレベルのエクスポートまたはインポートを使用して新しいマスター・サイトを追加します。[図 7-1](#) は、これらの条件をまとめたものです。

図 7-1 マスター・サイトの追加に使用する方法を決定





オブジェクトレベルのエクスポートまたはインポートを使用して、マスター・グループをすでに他のレプリケーション・グループを保有する新しいマスター・サイトへ追加、またはマスター・グループをレプリケーション・グループを保有していない新しいマスター・サイトへ追加します。この方法では、1つ以上のマスター・グループを新しいマスター・サイトへ一度に追加できます。また、マスター定義サイトのマスター・グループのサブセットを選択して操作実行時に新しいマスター・サイトへ追加できます。

オブジェクトレベルのエクスポートまたはインポートを使用する場合において、2つ以上のマスター・グループにまたがる整合性制約があり、これらの制約に該当する表が新しいマスター・サイトにすでに存在する場合は、新しいマスター・サイトを追加する表のこれらの整合性制約を一時的に無効にします。最初は、新しいマスター・サイトを参照する DEFSCCHEDULE データ・ディクショナリ・ビューには2つの行があります。伝播を受け取った場合は、このビューの行は1つです。また、すべてのマスター・サイトから新しいマスター・サイトへの伝播を受け取った場合は、無効化した整合性制約を再有効化できます。

マスター・グループを静止することなく新しいマスター・サイトを追加する2つの方法を再度、次に示します。

- 全データベースのエクスポートまたはインポート、あるいは変更ベースのリカバリ
- オブジェクトレベルのエクスポートまたはインポート

いずれかの方法を使用する場合は、新しいマスター・サイトの追加時に、新しいマスター・サイトへの遅延トランザクションの伝播は、部分的または完全に無効化されます。このため、既存の各マスター・サイトには、未伝播の遅延トランザクション・キューの中で最大のものを格納するのに十分な空き領域があることを確認します。

次の制限事項も両方法に適用されます。

- 影響を受けるマスター・グループはすべて、非同期レプリケーションを使用します。同期レプリケーションは使用できません。
- すべてのスケジュール・リンクは、パラレル化を1以上にセットしたパラレル伝播を使用します。
- 影響を受けるすべてのマスター・グループのデータベース・リンクは、接続修飾子を所有しないか、またはすべてのマスター・グループで同一の接続修飾子を所有するかのいずれかです。
- 新しいマスター・サイトを1つ以上のマスター・グループへ追加するプロセスを開始した後は、これらの新しいマスター・サイトが追加されてから、影響を受けるマスター・グループへの追加を開始します。マスター定義サイトの DBA\_NEW\_REPSITES データ・ディクショナリ・ビュー内に、影響を受けるマスター・グループの情報がある場合は、マスター・グループに対して開始した追加プロセスは完了できません。
- 新しいマスター・サイトを1つ以上のマスター・グループへ追加するプロセスを開始した後は、新しいマスター・サイトが追加されるまで、これらのマスター・グループのマスター定義サイトを再配置できません。DBA\_NEW\_REPSITES データ・ディクショナリ・ビュー内に、影響を受けるマスター・グループの情報がある場合は、マスター・グループに対して開始した追加プロセスは完了できません。

- マスター・サイトで許可されるマスター・サイトの追加要求は一度に1つのみです。たとえば、hq1.worldがmgroup1のマスター定義サイトで、hq2.worldがmgroup2のマスター定義サイトである場合、hq1.worldのmgroup2への追加およびhq2.worldのmgroup1への追加は同時には行えません。
- すべてのマスター・サイトは、9.0.0以上の互換性のレベルが必要です。互換性のレベルは、COMPATIBLE 初期化パラメータで管理します。互換性のレベルが9.0.0より低いマスター・サイトがある場合は、マスター・グループを静止して新しいマスター・サイトに拡張する必要があります。この場合は、7-30 ページの「[静止中のマスター・グループに新しいマスター・サイトを追加](#)」の指示に従います。
- オブジェクト・レベルまたは全データベースのエクスポートまたはインポートを使用している場合は、エクスポートのロールバック・セグメントまたは UNDO 表領域に十分な領域があることを確認します。

また、いずれかの方法で新しいマスター・サイトを追加する前に、新しいマスター・サイトがマルチマスター・レプリケーション用に正しく設定されていることを確認します。

---

**注意：** 次の項に記述された手順を続行できなくなった場合は、メッセージのトレース・ファイルおよびアラート・ログをチェックします。

---

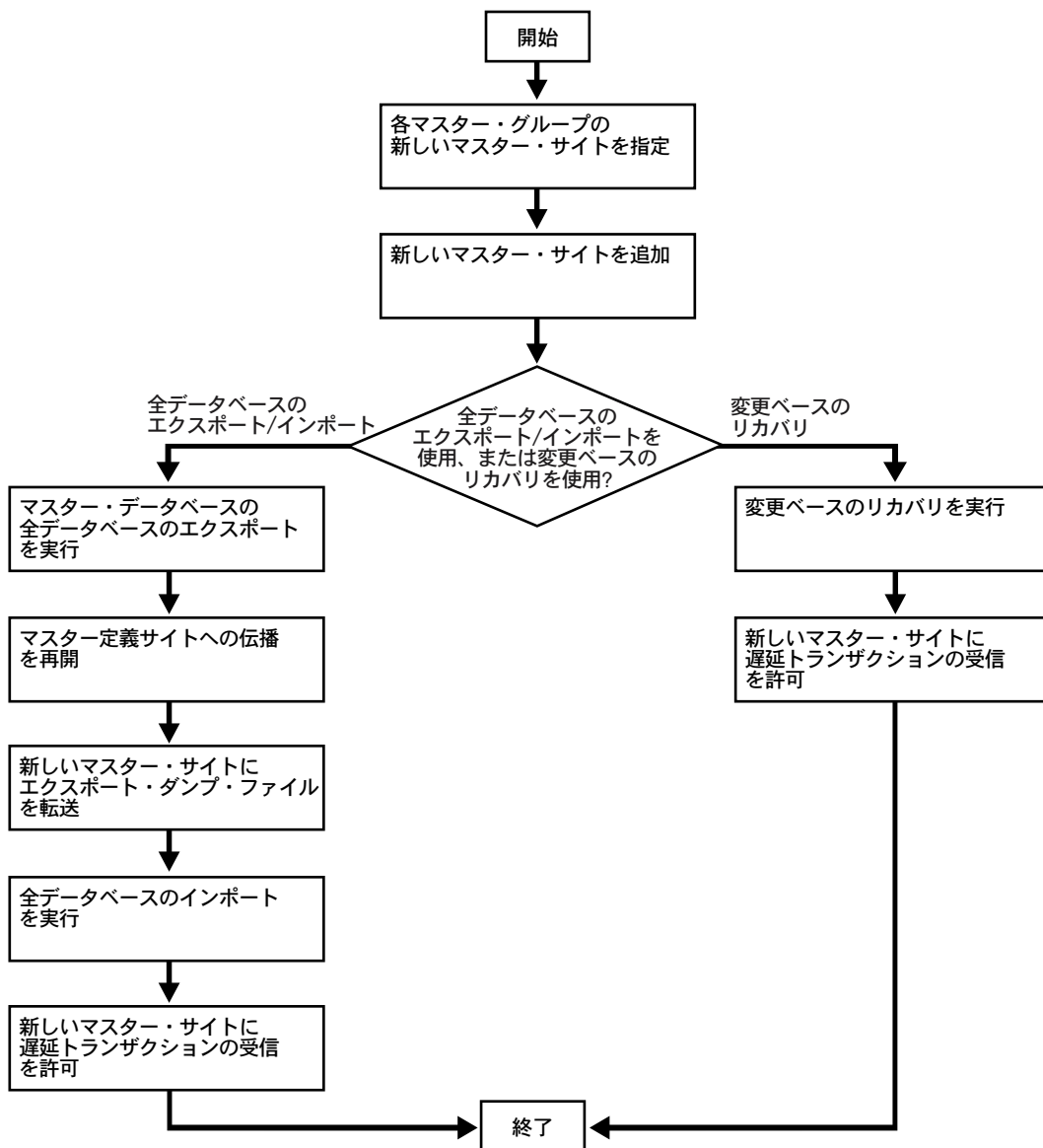
### 関連項目：

- 新しいマスター・サイトをマルチマスター・レプリケーション用に設定する場合の詳細は、2-4 ページの「[マスター・サイトの設定](#)」を参照してください。
- トレース・ファイルおよびアラート・ログの詳細は、『Oracle9i データベース管理者ガイド』を参照してください。
- 取消し領域の管理の詳細は、『Oracle9i データベース管理者ガイド』を参照してください。

## 全データベースのエクスポートまたはインポートの使用、あるいは変更ベースのリカバリの使用

**図 7-2** では、全データベースのエクスポートまたはインポート、あるいは変更ベースのリカバリを使用して、マスター・グループを静止することなく新しいマスター・サイトをマスター・グループに追加する主な手順について説明しています。次のスクリプト例では、新しいマスター・サイト orc4.world および orc5.world を hr\_repg マスター・グループに追加しています。この例では、orc4.world は全データベースのエクスポートまたはインポートを使用して追加され、orc5.world は変更ベースのリカバリを使用して追加されています。

図 7-2 全データベースのエクスポートまたはインポートの使用、あるいは変更ベースのリカバリの使用



これらのアクションを実行するには、次の要件を満たす必要があります。

**実行者:** レプリケーション管理者

**実行場所:**

- 各新しいマスター・サイトで手順 1
- マスター定義サイトで手順 2 - 6b
- 手順 6c ではサイト間のファイル転送が必要
- 各新しいマスター・サイトで手順 6d - 9

**レプリケーションの状態:** 実行中（静止中ではない）

次の手順に従い、全データベースのエクスポートまたはインポート、あるいは変更ベースのリカバリを使用して、サイトをマスター・グループに追加します。

---

---

**注意:** このドキュメントをオンラインで参照している場合は、次の「BEGINNING OF SCRIPT」の行から 7-18 ページの「END OF SCRIPT」の行までのテキストをテキスト・エディタにコピーして編集し、使用している環境に適したスクリプトを作成します。

---

---

```
/***** BEGINNING OF SCRIPT *****/
```

**手順 1 全データベースのエクスポートまたはインポートを使用している場合は、マスター・グループへ追加するデータベースを作成します。**

変更ベースのリカバリを使用している場合、この手順は必要ありません。

**関連項目:** データベース作成の詳細は『Oracle9i データベース管理者ガイド』を参照してください。

```
*/
```

```
SET ECHO ON
```

```
SPOOL add_masters_full.out
```

```
PAUSE Press <RETURN> when the databases for the new master sites are created.
```

```
/*
```

**手順 2 新しい各マスター・サイトをレプリケーション・サイトとして設定します。次の構成を行う必要があることに注意してください。**

- 新しい各マスター・サイトのレプリケーション管理者
- 既存の各マスター・サイトから新しい各マスター・サイトへのスケジュール・リンク
- 新しい各マスター・サイトから既存の各マスター・サイトへのスケジュール・リンク
- 新しい各マスター・サイトでのスケジュール・ページ

\*/

PAUSE Press <RETURN> to continue the new master sites have been setup and the required scheduled links have been created.

/\*

**関連項目：**

- スケジュール・リンクの詳細は、『Oracle9i アドバンスド・レプリケーション』を参照してください。
- 2-4 ページ「[マスター・サイトの設定](#)」
- 2-16 ページ「[マスター・サイト間のスケジュール・リンクの作成](#)」

**手順 3 レプリケーション管理者としてマスター定義サイトに接続します。**

\*/

CONNECT repadmin/repadmin@orc1.world

/\*

**手順 4 各マスター・グループの新しいマスター・サイトを指定します。**

新しいマスター・サイトを指定する前に、リンクが存在していない場合は既存のマスター・サイトと各新しいマスター・サイト間で必要なスケジュール・リンクを作成します。

**関連項目：**

- スケジュール・リンクの詳細は、『Oracle9i アドバンスド・レプリケーション』を参照してください。
- これらの例は、2-16 ページの「[マスター・サイト間のスケジュール・リンクの作成](#)」を参照してください。

```
*/  
  
BEGIN  
    DBMS_REPCAT.SPECIFY_NEW_MASTERS (  
        gname => 'HR_REPG',  
        master_list => 'orc4.world,orc5.world');  
END;  
/  
  
/*
```

他の SQL\*Plus セッションで次のデータ・ディクショナリ・ビューに問い合わせることで、拡張プロセスの追跡を開始できます。

- DBA\_REPSITES\_NEW
- DBA\_REPEXTENSIONS

```
*/  
  
PAUSE Press <RETURN> when you have completed the these steps.  
  
/*
```

### 手順 5 新しいマスター・サイトを追加します。

次のプロシージャを実行する前に、新しいマスター・サイトで実行中のバックグラウンド・ジョブの数が適切であることを確認します。全データベースのエクスポートまたはインポートを使用している場合は、このプロシージャを実行する前に、エクスポートのロールバック・セグメントまたは UNDO 表領域に十分な領域があることを確認します。

#### 関連項目：

- レプリケーション環境の JOB\_QUEUE\_PROCESSES 初期化パラメータ・プロパティの設定の詳細は、『Oracle9i アドバンスド・レプリケーション』を参照してください。
- 取消し領域の管理の詳細は、『Oracle9i データベース管理者ガイド』を参照してください。

```
*/  
  
VARIABLE masterdef_flashback_scn NUMBER;  
VARIABLE extension_id VARCHAR2(32);  
BEGIN  
    DBMS_REPCAT.ADD_NEW_MASTERS (  
        export_required => true,  
        available_master_list => NULL,  
        masterdef_flashback_scn => :masterdef_flashback_scn,  
        extension_id => :extension_id,
```

```

break_trans_to_masterdef => false,
break_trans_to_new_masters => false,
percentage_for_catchup_mdef => 80,
cycle_seconds_mdef => 60,
percentage_for_catchup_new => 80,
cycle_seconds_new => 60);

```

```
END;
```

```
/
```

```
/*
```

masterdef\_flashback\_scn および extension\_id の値は変数に保存され、後のプロセスで使用します。これらの値を参照する場合は、DBA\_REPSITES\_NEW および DBA\_REPEXTENSIONS データ・ディクショナリ・ビューへの問合せを行います。

```
*/
```

PAUSE Press <RETURN> when you have completed the these steps.

```
/*
```

SPECIFY\_NEW\_MASTERS および ADD\_NEW\_MASTERS プロシージャで、特定のマスター・サイトに対して行った変更を取り消す場合は、DBMS\_REPCAT.UNDO\_ADD\_NEW\_MASTERS\_REQUEST プロシージャを使用します。

orc4.world は全データベースのエクスポートまたはインポートを使用して追加されたため、export\_required パラメータには、TRUE が指定されます。orc5.world は変更ベースのリカバリを使用していますが、TRUE の設定は誤りではありません。これは、少なくとも 1 つの新しいマスター・サイトが、エクスポートまたはインポートを使用しているためです。

このプロシージャを実行した後、他の SQL\*Plus セッションで DBA\_REPCATLOG データ・ディクショナリ・ビューに問合せを行うことで、進行状況を監視します。このビューに新しいマスター・サイトの追加情報がなくなるまでは、手順 6 には進まないでください。

DBA\_REPCATLOG には他の操作による無関係の情報が含まれていないことを前提に、次の文を入力します。

```
SELECT COUNT(*) FROM DBA_REPCATLOG;
```

この文から 0（ゼロ）が戻されれば、すべてのプロセスは完了です。

```
*/
```

PAUSE Press <RETURN> to continue when DBA\_REPCATLOG is empty.

```
/*
```

**手順 6 全データベースのエクスポートまたはインポートを使用して追加されたマスター・サイトの次のサブステップを実行します。変更ベースのリカバリを使用して追加されたマスター・サイトでは、このサブステップは必要ないので、7-16 ページの手順 7 に進みます。**

- a. マスター定義データベースの全データベースのエクスポートを実行します。  
FLASHBACK\_SCN エクスポート・パラメータの手順 5 の  
masterdef\_flashback\_scn パラメータから戻されるシステム変更番号 (SCN)  
を使用します。

DBA\_REPEXTENSIONS データ・ディクショナリ・ビューへ FLASHBACK\_SCN 値の  
問合せを行えます。

```
SELECT FLASHBACK_SCN FROM DBA_REPEXTENSIONS;
```

この例で、orc4.world は全データベースのエクスポートまたはインポートを使用  
しています。このため、今後のステップで orc4.world へのインポートが行えるよう  
に、マスター定義データベースの全データベースのエクスポートを実行します。一  
方、orc5.world データベースは変更ベースのリカバリを使用しています。このた  
め、orc5.world のみの追加では、エクスポートは必要とされません。

エクスポート文の例を次に示します。

```
exp system/manager FILE=fulldb_orc1.dmp FULL=y DIRECT=n  
GRANTS=y ROWS=y COMPRESS=y STATISTICS=compute LOG=exp_orc1.log  
FLASHBACK_SCN=124723
```

---

---

**注意：** また、データベースのエクスポートを実行するには、コマンドラ  
イン・プロンプトで exp system/manager と入力し、以降のプロンプト  
に答える方法もあります。

---

---

エクスポート・ユーティリティを実行する場合の考慮事項は、次のとおりです。

- DBA ロールまたは EXP\_FULL\_DATABASE ロールを所有するユーザーのみが全  
データベース・モードでのエクスポートを実行できます。
- エクスポートを実行する前に、UNDO\_RETENTION 初期化パラメータが正しく  
セットされていることを確認します。
- CONSISTENT エクスポート・パラメータは使用しないでください。

#### 関連項目：

- 全データベースのエクスポートの詳細は『Oracle9i データベース・  
ユーティリティ』を参照してください。
- 取消し領域の管理およびこのパラメータの設定の詳細は『Oracle9i  
データベース管理者ガイド』を参照してください。



```
*/
```

```
PAUSE Press <RETURN> to continue when the export is complete.
```

```
/*
```

- b.** マスター定義サイトへの伝播を再開します。

次のプロシージャを実行することで、マスター・サイトの拡張されたマスター・グループおよび影響されていないマスター・グループに対するエクスポートが正常に行われ、伝播が有効化されたことが示されます。

```
*/
```

```
BEGIN
```

```
    DBMS_REPCAT.RESUME_PROPAGATION_TO_MDEF (  
        extension_id => :extension_id);
```

```
END;
```

```
/
```

```
/*
```

DBA\_REPSITES\_NEW データ・ディクショナリ・ビューへ問合せを行うことで、extension\_id を検索できます。

- c.** エクスポート・ダンプ・ファイルを新しいマスター・サイトに転送します。

全データベースのエクスポートまたはインポートで追加された他の新しいマスター・サイトへは、FTP などの方法により、エクスポート・ダンプ・ファイルが転送されます。次の手順で説明するインポートを実行する場合、新しいサイトで、このエクスポート・ダンプ・ファイルが必要となります。

```
*/
```

```
PAUSE Press <RETURN> to continue after transferring the dump file.
```

```
/*
```

- d.** 新しいマスター・サイトの JOB\_QUEUE\_PROCESSES 初期化パラメータを 0（ゼロ）にセットします。

```
*/
```

```
PAUSE Press <RETURN> to continue after JOB_QUEUE_PROCESSES is set to zero at  
each new master site.
```

```
/*
```

**手順 7 新しいマスター・サイトでインポートまたは変更ベースのリカバリを実行します。**

- 全データベースのエクスポートまたはインポートを使用している場合は、全データベースのエクスポートまたはインポートで追加された新しいマスター・サイトの手順 6a でエクスポートしたデータベースの全データベースのインポートを実行します。

インポート文の例を次に示します。

```
imp system/manager FILE=fulldb_orcl.dmp FULL=y BUFFER=30720  
IGNORE=y GRANTS=y ROWS=y DESTROY=y COMMIT=y LOG=import.log
```

---

**注意：** また、データベースのインポートを実行するには、コマンドライン・プロンプトで `imp system/manager` と入力し、以降のプロンプトに答える方法もあります。

---

DBA ロールまたは `IMP_FULL_DATABASE` ロールを所有するユーザーのみが全データベース・モードでのインポートを実行できます。

**関連項目：** 全データベースのインポートの詳細は、『Oracle9i データベース・ユーティリティ』を参照してください

\*/

PAUSE Press <RETURN> to continue when the import is complete.

/\*

- 変更ベースのリカバリを使用している場合は、手順 5 の `masterdef flashback_scn` パラメータから戻されたシステム変更番号 (SCN) を使用して変更ベースのリカバリを実行します。DBA\_REPEXTENSIONS データ・ディクショナリ・ビューへ `masterdef flashback_scn` 値の問合せを行えます。

次の方法で変更ベースのリカバリを実行できます。

- \* SQL\*Plus の RECOVER コマンドを使用します。指示の詳細は、『Oracle9i ユーザー管理バックアップおよびリカバリ・ガイド』を参照してください。
- \* Recovery Manager の DUPLICATE コマンドを使用します。指示の詳細は、『Oracle9i Recovery Manager ユーザーズ・ガイド』を参照してください。

変更ベースのリカバリを実行するサイトに接続します。

```
*/
```

```
CONNECT repadmin/repadmin@orc5.world
```

```
PAUSE Press <RETURN> to continue when the change-based recovery is complete.  
You can use a separate terminal window to perform the change-based recovery.
```

```
/*
```

**手順 8 次の手順に従って、新しいサイトをマルチマスター・レプリケーション用に構成します。**

- a. 新しいマスター・サイトにレプリケート・スキーマとして存在するデータ・ファイルなどの、データベース構造を確認します。この例では、レプリケート・スキーマは hr です。
- b. 新しいマスター・サイトにグローバル名をセットします。新しいマスター・サイトのグローバル名は、手順 4 で実行した SPECIFY\_NEW\_MASTERS プロシージャに指定したグローバル名と一致する必要があります。DBA\_REPSITES\_NEW データ・ディクショナリ・ビューの DBLINK 列への問合せを行い、新しいマスター・サイトのグローバル名を確認できます。

次の例に示すように、ALTER DATABASE 文を使用してグローバル名をセットできます。

```
ALTER DATABASE RENAME GLOBAL_NAME TO orc4.world;
```

- c. 新しいマスター・サイトと既存のマスター・サイト間に、マスター定義サイトなどの、適切なスケジュール・リンクを作成します。

**関連項目：** 詳細は、2-16 ページの「[マスター・サイト間のスケジュール・リンクの作成](#)」を参照してください。

```
*/
```

```
PAUSE Press <RETURN> when you have completed the these steps.
```

```
/*
```

### 手順9 新しいマスター・サイトで遅延トランザクションを受信します。

次のプロシージャでは、新しく準備されたマスター・サイトや既存のマスター・サイトから起動マスター・サイトへの遅延トランザクションの伝播が可能です。このプロシージャでは、起動マスター・サイトから新しく準備されたマスター・サイトや既存のマスター・サイトへの遅延トランザクションの伝播も可能です。

---

---

**注意：** 新しいマスター・サイトのインスタンス化（エクスポートまたはインポート、あるいは変更ベースのリカバリ）が完了するまでは、このプロシージャを起動しないでください。

プロシージャから完了通知が戻るまでは、データ操作言語（DML）文を新しいマスター・サイトの拡張マスター・グループのオブジェクトに適用しないでください。これは、これらの DML 文が、レプリケートされないためです。

---

---

```
*/

CONNECT repadmin/repadmin@orc4.world

BEGIN
    DBMS_REPCAT.PREPARE_INSTANTIATED_MASTER (
        extension_id => :extension_id);
END;
/

CONNECT repadmin/repadmin@orc5.world

BEGIN
    DBMS_REPCAT.PREPARE_INSTANTIATED_MASTER (
        extension_id => :extension_id);
END;
/

SET ECHO OFF

SPOOL OFF

/*
```

---

---

**注意：** DBA\_REPSITES\_NEW データ・ディクショナリ・ビューへ問合せを行うことで、extension\_idを検索できます。

---

---

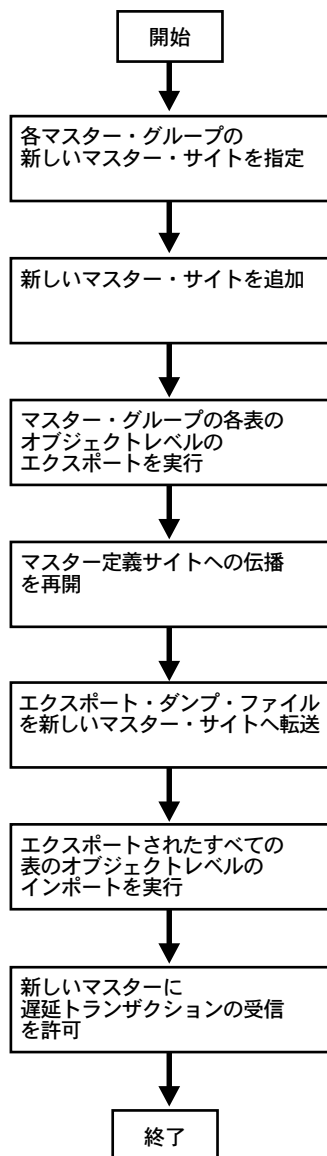
```
***** END OF SCRIPT *****/
```

## オブジェクトレベルのエクスポート/インポートの使用

図 7-3 では、オブジェクトレベルのエクスポートまたはインポートを使用して、マスター・グループを静止することなく新しいマスター・サイトをマスター・グループに追加する主な手順について説明しています。次のプロシージャ例では、新しいマスター・サイト `orc4.world` および `orc5.world` を `hr_repg` マスター・グループに追加しています。オブジェクトレベルのエクスポートまたはインポートには、マスター・グループの表のエクスポートとインポートが含まれます。表のエクスポートとインポートでは、索引などの、その他の依存データベース・オブジェクトも同様にエクスポート、インポートされます。

2 つのマスター・グループにまたがる整合性制約がある場合は、一方のマスター・グループに子表（子マスター・グループ）、他方のマスター・グループに親表（親マスター・グループ）を用意します。この場合、オラクル社は新しいマスター・サイトを両方のマスター・グループに同時に追加することをお勧めします。しかし、同時に追加することができない場合は、子マスター・グループを静止してから新しいマスター・サイトを追加します。ここでは、子表には外部キーが含まれているため、子表は親表の値に依存します。子マスター・グループを静止しないで追加を行うと、マスター・サイトの追加時に競合が発生します。親マスター・グループへのマスター・サイトの追加は静止なしでも行えます。

図 7-3 オブジェクトレベルのエクスポート/インポートの使用



これらのアクションを実行するには、次の要件を満たす必要があります。

**実行者:** レプリケーション管理者

**実行場所:**

- マスター定義サイトで手順 1-8
- 手順 9 ではサイト間のファイル転送が必要
- 各新しいマスター・サイトで手順 10-11

**レプリケーションの状態:** 実行中（静止中ではない）

次の手順に従って、オブジェクトレベルのエクスポートまたはインポートを使用してサイトをマスター・グループに追加します。

---



---

**注意:** このドキュメントをオンラインで参照している場合は、次の「BEGINNING OF SCRIPT」の行から 7-29 ページの「END OF SCRIPT」の行までのテキストをテキスト・エディタにコピーして編集し、使用している環境に適したスクリプトを作成します。

---



---

```
/***** BEGINNING OF SCRIPT *****/
```

**手順 1 新しいマスター・サイトにエクスポート・スキーマのユーザーが存在しない場合は、ここで作成します。この例では、レプリケート・スキーマは `hr` です。このスキーマは、通常、Oracle とともにインストールされるサンプル・スキーマであるため、新しいマスター・サイトに存在します。**

**関連項目:** サンプル・スキーマの概要とそのインストール方法の詳細は、『Oracle9i サンプル・スキーマ』を参照してください。

```
*/
```

```
SET ECHO ON
```

```
SPOOL add_masters_object.out
```

```
PAUSE Press <RETURN> to continue when the users are created at the new master sites.
```

```
/*
```

**手順 2 マスター・グループ内の表のいずれかに循環依存がある場合は、これらの表を新しいマスター・サイトで事前に作成しておきます。循環依存を持つ表を事前に作成しないと、後でエラーが発生します。循環依存がなければ、このステップは必要ないので手順 3 に進みます。**

hr スキーマの一部の表には、循環依存が含まれています。そのため、この例では、hr スキーマ内の表を新しい各マスター・サイトで事前に作成する必要があります。また、hr スキーマの表は、通常は Oracle のインストール中に作成されるため、すでに新しいマスター・サイトに存在する可能性があります。

表の事前作成が必要な場合は、インポート前に新しいマスター・サイトで、これらの表に対する参照整合性制約を無効化します。参照整合性制約が有効化されていると、既存の表にデータをインポートするときにエラーの原因になることがあります。この例では、新しいマスター・サイト `orc4.world` で、hr スキーマに事前作成する表に対する参照整合性制約を無効化します。

また、新しいマスター・サイトで事前作成する表には、データを含めないでください。この例では、データが含まれないように、新しいマスター・サイト `orc4.world` で hr スキーマ内の表を切り捨てます。

### 関連項目：

- 循環依存の詳細は、7-4 ページの「[新しいマスター・サイトの追加](#)」の「注意」を参照してください。
- 既存の表にデータをインポートする方法の詳細は、『Oracle9i データベース・ユーティリティ』を参照してください。

\*/

```
PAUSE Press <RETURN> to continue when the tables are precreated at the new master
sites, if table precreation is required. After the tables are precreated, the
following statements disable the referential integrity constraints in the hr schema
and truncate the tables in the hr schema at the new site.
```

```
CONNECT hr/hr@orc4.world

ALTER TABLE hr.countries
  DISABLE CONSTRAINT countr_reg_fk;

ALTER TABLE hr.departments
  DISABLE CONSTRAINT dept_mgr_fk
  DISABLE CONSTRAINT dept_loc_fk;
```



```
ALTER TABLE hr.employees
  DISABLE CONSTRAINT emp_dept_fk
  DISABLE CONSTRAINT emp_job_fk
  DISABLE CONSTRAINT emp_manager_fk;
```

```
ALTER TABLE hr.job_history
  DISABLE CONSTRAINT jhist_job_fk
  DISABLE CONSTRAINT jhist_emp_fk
  DISABLE CONSTRAINT jhist_dept_fk;
```

```
ALTER TABLE hr.locations
  DISABLE CONSTRAINT loc_c_id_fk;
```

```
TRUNCATE TABLE hr.countries;
TRUNCATE TABLE hr.departments;
TRUNCATE TABLE hr.employees;
TRUNCATE TABLE hr.jobs;
TRUNCATE TABLE hr.job_history;
TRUNCATE TABLE hr.locations;
TRUNCATE TABLE hr.regions;
```

/\*

**手順 3 新しい各マスター・サイトをレプリケーション・サイトとして設定します。次の構成を行う必要があることに注意してください。**

- 新しい各マスター・サイトのレプリケーション管理者
- 既存の各マスター・サイトから新しい各マスター・サイトへのスケジュール・リンク
- 新しい各マスター・サイトから既存の各マスター・サイトへのスケジュール・リンク
- 新しい各マスター・サイトでのスケジュール・ページ

\*/

PAUSE Press <RETURN> to continue the new master sites have been setup and the required scheduled links have been created.

/\*

### 関連項目：

- スケジュール・リンクの詳細は、『Oracle9i アドバンスド・レプリケーション』を参照してください。
- 2-4 ページ「[マスター・サイトの設定](#)」
- 2-16 ページ「[マスター・サイト間のスケジュール・リンクの作成](#)」

### 手順 4 レプリケーション管理者としてマスター定義サイトに接続します。

\*/

```
CONNECT repadmin/repadmin@orc1.world
```

/\*

### 手順 5 各マスター・グループの新しいマスター・サイトを指定します。

\*/

```
BEGIN
```

```
  DBMS_REPCAT.SPECIFY_NEW_MASTERS (  
    gname => 'hr_repg',  
    master_list => 'orc4.world,orc5.world');
```

```
END;
```

```
/
```

/\*

他の SQL\*Plus セッションで次のデータ・ディクショナリ・ビューに問い合わせることで、拡張プロセスの追跡を開始できます。

- DBA\_REPSITES\_NEW
- DBA\_REPEXTENSIONS

### 手順 6 新しいマスター・サイトを追加します。

次のプロシージャを実行する前に、新しいマスター・サイトで実行中のバックグラウンド・ジョブの数が適切であることを確認します。このプロシージャを実行する前に、エクスポートのロールバック・セグメントまたは UNDO 表領域に十分な領域があることも確認します。

**関連項目：**

- レプリケーション環境の JOB\_QUEUE\_PROCESSES 初期化パラメータ・プロパティの設定の詳細は、『Oracle9i アドバンスド・レプリケーション』を参照してください。
- 取消し領域の管理の詳細は、『Oracle9i データベース管理者ガイド』を参照してください。

```

*/

VARIABLE masterdef_flashback_scn NUMBER;
VARIABLE extension_id VARCHAR2(32);
BEGIN
    DBMS_REPCAT.ADD_NEW_MASTERS (
        export_required => true,
        available_master_list => 'orc4.world,orc5.world',
        masterdef_flashback_scn => :masterdef_flashback_scn,
        extension_id => :extension_id,
        break_trans_to_masterdef => false,
        break_trans_to_new_masters => false,
        percentage_for_catchup_mdef => 80,
        cycle_seconds_mdef => 60,
        percentage_for_catchup_new => 80,
        cycle_seconds_new => 60);
END;
/

/*

```

available\_master\_list パラメータで指定したサイトは、手順 5 の SPECIFY\_NEW\_MASTERS プロシージャに指定したサイトと同一の必要があります。

masterdef\_flashback\_scn および extension\_id の値は変数に保存され、後のプロセスで使用します。これらの値を参照する場合は、DBA\_REPSITES\_NEW および DBA\_REPEXTENSIONS データ・ディクショナリ・ビューへの問合せも行えます。

SPECIFY\_NEW\_MASTERS および ADD\_NEW\_MASTERS プロシージャで、特定のマスター・サイトに対して行った変更を取り消す場合は、UNDO\_ADD\_NEW\_MASTERS\_REQUEST プロシージャを使用します。

このプロシージャを実行した後、他の SQL\*Plus セッションで DBA\_REPCATLOG データ・ディクショナリ・ビューに問合せを行うことで、進行状況を監視します。このビューに新しいマスター・サイトの追加情報がなくなるまでは、手順 7 には進まないでください。DBA\_REPCATLOG には他の操作による無関係の情報が含まれていないことを前提に、次の文を入力します。

```
SELECT COUNT(*) FROM DBA_REPCATLOG;
```

この文から 0 (ゼロ) が戻されれば、すべてのプロセスは完了です。

```
*/
```

```
PAUSE Press <RETURN> to continue when DBA_REPCATLOG is empty.
```

```
/*
```

### 手順 7 マスター定義データベースでオブジェクトレベルの表のエクスポートを実行します。

マスター定義データベースで、新しいマスター・サイトに作成するマスター・グループ内の各マスター表のオブジェクトレベルのエクスポートを実行します。オブジェクトレベルのエクスポートには、表モード、ユーザー・モードまたは表領域モードで実行するエクスポートがあります。

FLASHBACK\_SCN エクスポート・パラメータの手順 6 の masterdef\_flashback\_scn パラメータから戻されるシステム変更番号 (SCN) を使用します。DBA\_REPEXTENSIONS データ・ディクショナリ・ビューへ FLASHBACK\_SCN 値の問合せを行えます。

```
SELECT FLASHBACK_SCN FROM DBA_REPEXTENSIONS;
```

ユーザー・モードのすべての hr スキーマをエクスポートするオブジェクトレベルのエクスポートの例を次に示します。

```
exp system/manager FILE=hr_schema.dmp OWNER=hr DIRECT=n GRANTS=y  
ROWS=y COMPRESS=y INDEXES=y CONSTRAINTS=y STATISTICS=compute  
FLASHBACK_SCN=3456871
```

---

---

**注意：** また、データベースのエクスポートを実行するには、コマンドライン・プロンプトで exp system/manager と入力し、以降のプロンプトに答える方法もあります。

---

---

表のエクスポートでは、索引が自動的にエクスポートされます。

エクスポート・ユーティリティを実行する場合の考慮事項は、次のとおりです。

- エクスポートを実行する前に、UNDO\_RETENTION 初期化パラメータが正しくセットされていることを確認します。
- CONSISTENT エクスポート・パラメータは使用しないでください。

**関連項目：**

- オブジェクトレベルのデータベースのエクスポートの詳細は、『Oracle9i データベース・ユーティリティ』を参照してください。
- 取消し領域の管理および UNDO\_RETENTION 初期化パラメータの設定の詳細は、『Oracle9i データベース管理者ガイド』を参照してください。

```
*/
```

```
PAUSE Press <RETURN> to continue when the export is complete.
```

```
/*
```

**手順 8 マスター定義サイトへの伝播を再開します。**

次のプロシージャを実行することで、マスター・サイトの拡張されたマスター・グループおよび影響されていないマスター・グループに対するエクスポートが正常に行われ、伝播が有効化されたことが示されます。

```
*/
```

```
BEGIN
```

```
    DBMS_REPCAT.RESUME_PROPAGATION_TO_MDEF (
        extension_id => :extension_id);
```

```
END;
```

```
/
```

```
/*
```

DBA\_REPSITES\_NEW データ・ディクショナリ・ビューへ問合せを行うことで、extension\_id を検索できます。

**手順 9 エクスポート・ダンプ・ファイルを新しいマスター・サイトに転送します。**

オブジェクトレベルのエクスポートまたはインポートで追加された他の新しいマスター・サイトへは、FTP などの方法により、エクスポート・ダンプ・ファイルが転送されます。次の手順で説明するインポートを実行する場合、各新しいサイトで、これらのエクスポート・ダンプ・ファイルが必要となります。

```
*/
```

```
PAUSE Press <RETURN> to continue when the export dump files have been
transferred to the new master sites that are being added with object-level
export/import.
```

```
/*
```

**手順 10 手順 7 でエクスポートした各オブジェクトの各新しいマスター・サイトでオブジェクトレベルのインポートを実行します。**

すべての hr スキーマをインポートするオブジェクトレベルのインポートの例を次に示します。

```
imp system/manager FILE=hr_schema.dmp FROMUSER=hr BUFFER=30720 IGNORE=y
GRANTS=y ROWS=y DESTROY=y COMMIT=y
```

---

**注意：** また、データベースのエクスポートとインポートを実行するには、コマンドライン・プロンプトで `exp system/manager` と入力し、以降のプロンプトに答える方法もあります。

---

表に基づいて作成された索引など、他のオブジェクトは自動的にインポートされます。

**関連項目：** オブジェクトレベルのインポートの詳細は、『Oracle9i データベース・ユーティリティ』を参照してください。

オブジェクトレベルのインポートを実行するサイトに接続して、各サイトでインポートを実行します。

\*/

```
CONNECT repadmin/repadmin@orc4.world
```

```
PAUSE Press <RETURN> to continue when the imports are complete at this site. You can
use a separate terminal window to perform the object-level imports.
```

```
CONNECT repadmin/repadmin@orc5.world
```

```
PAUSE Press <RETURN> to continue when the imports are complete at this site. You can
use a separate terminal window to perform the object-level imports.
```

/\*

**手順 11 新しいマスター・サイトで遅延トランザクションを受信します。**

次のプロシージャでは、新しく準備されたマスター・サイトや既存のマスター・サイトから起動マスター・サイトへの遅延トランザクションの伝播が可能です。このプロシージャでは、起動マスター・サイトから新しく準備されたマスター・サイトや既存のマスター・サイトへの遅延トランザクションの伝播も可能です。

---

---

**注意：** 新しいマスター・サイトに対するオブジェクトレベルのエクスポートまたはインポートが完了するまでは、このプロシージャは起動しないでください。

プロシージャから完了通知が戻るまでは、データ操作言語（DML）文を新しいマスター・サイトの拡張マスター・グループのオブジェクトに適用しないでください。これは、これらの DML 文が、レプリケートされないためです。

---

---

```
*/

CONNECT repadmin/repadmin@orc4.world

BEGIN
    DBMS_REPCAT.PREPARE_INSTANTIATED_MASTER (
        extension_id => :extension_id);
END;
/

CONNECT repadmin/repadmin@orc5.world

BEGIN
    DBMS_REPCAT.PREPARE_INSTANTIATED_MASTER (
        extension_id => :extension_id);
END;
/

SET ECHO OFF

SPOOL OFF

/*
```

---

---

**注意：** DBA\_REPSITES\_NEW データ・ディクショナリ・ビューへ問合せを行うことで、extension\_id を検索できます。

---

---

```
***** END OF SCRIPT *****/
```

## 静止中のマスター・グループに新しいマスター・サイトを追加

次に示す方法で、新しいマスター・サイトを静止中のマスター・グループに追加できます。

- [ADD\\_MASTER\\_DATABASE プロシージャを使用して新しいマスター・サイトを追加](#)
- [エクスポート / インポートを使用するオフライン・インスタンスエーションで新しいマスター・サイトを追加](#)

通常は、マスター・グループが小規模の場合または新しいマスター・サイトでレプリケーション表を事前に作成してデータをロードする場合にのみ `ADD_MASTER_DATABASE` プロシージャを使用します。これ以外の場合は、`ADD_MASTER_DATABASE` プロシージャは使用しません。このプロシージャでは、ネットワーク上のすべてのマスター・グループがコピーされるためです。大規模なマスター・グループでは、新しいマスター・サイトのマスター・グループにオブジェクトを事前に作成するか、オフライン・インスタンスエーションを使用します。

### **ADD\_MASTER\_DATABASE プロシージャを使用して新しいマスター・サイトを追加**

`ADD_MASTER_DATABASE` プロシージャを使用して、静止中の既存のマスター・グループにマスター・サイトを追加できます。このプロシージャを実行すると、既存のマスター・オブジェクトが新しいサイトにレプリケートされます。互換性のレベルが 9.0.0 より低いマスター・サイトがある場合は、次のプロシージャを使用する必要があります。すなわち、マスター・グループを静止して、新しいマスター・サイトに拡張する必要があります。データベースの互換性のレベルは、`COMPATIBLE` 初期化パラメータで管理します。

これらのアクションを実行するには、次の要件を満たす必要があります。

**実行者:** レプリケーション管理者

**実行場所:** マスター定義サイト

**レプリケーションの状態:** 静止中

次の手順に従って、`ADD_MASTER_DATABASE` プロシージャを使用してサイトをマスター・グループに追加します。

---

---

**注意:** このドキュメントをオンラインで参照している場合は、次の「BEGINNING OF SCRIPT」の行から 7-33 ページの「END OF SCRIPT」の行までのテキストをテキスト・エディタにコピーして編集し、使用している環境に適したスクリプトを作成します。

---

---



```
/***** BEGINNING OF SCRIPT *****/
```

### 手順 1 新しいマスター・サイトを設定します。

新しいマスター・サイトを追加する前に、適切なスキーマおよびデータベース・リンクが作成されていることを確認します。新しいマスター・サイトから既存の各マスター・サイトへのデータベース・リンクの作成を確認します。既存の各マスター・サイトから新しいマスター・サイトへのデータベース・リンクも作成します。データベース・リンクの作成を終えた後、新しいデータベース・リンクごとにスケジュール・リンクも定義します。

#### 関連項目：

- 2-4 ページ「マスター・サイトの設定」
- 2-16 ページ「マスター・サイト間のスケジュール・リンクの作成」

```
*/
```

```
SET ECHO ON
```

```
SPOOL add_masters_quiesced.out
```

```
PAUSE Press <RETURN> to the new master site has been set up.
```

```
/*
```

### 手順 2 レプリケーション管理者としてマスター定義サイトに接続します。

```
*/
```

```
CONNECT repadmin/repadmin@orc1.world
```

```
/*
```

### 手順 3 レプリケーションの状態が通常の場合は、静止中に変更します。

```
*/
```

```
BEGIN
```

```
  DBMS_REPCAT.SUSPEND_MASTER_ACTIVITY (  
    gname => 'hr_repg');
```

```
END;
```

```
/
```

```
/*
```

### 手順 4 ADD\_MASTER\_DATABASE プロシージャを使用して新しいマスター・サイトを追加します。

この例では、新しいマスター・サイトにレプリケート・オブジェクトが存在しないことを前提としています。このため、マスター定義サイトのレプリケート・オブジェクトの行を新しいマスター・サイトにコピーする場合は `copy_rows` パラメータを `TRUE` にセットします。また、Advanced Replication が新しいサイトにレプリケート・オブジェクトを作成するためには、`use_existing_objects` パラメータを `FALSE` にセットします。レプリケート・オブジェクトが新しいサイトにすでに存在して、データが含まれていない場合は、`use_existing_objects` を `TRUE` にセットします。

```
*/  
  
BEGIN  
    DBMS_REPCAT.ADD_MASTER_DATABASE (  
        gname => 'hr_repg',  
        master => 'orc4.world',  
        use_existing_objects => FALSE,  
        copy_rows => TRUE,  
        propagation_mode => 'ASYNCHRONOUS');
```

```
END;
```

```
/
```

```
/*
```

DBA\_REPCATLOG ビューが空になるまで待機します。このビューには、正常な実行後には消去される一時的な情報が含まれます。他の SQL\*Plus セッションで次の `SELECT` 文を実行して、DBA\_REPCATLOG ビューを監視します。

```
SELECT COUNT(*) FROM DBA_REPCATLOG WHERE GNAME = 'HR_REPG';
```

この文から 0（ゼロ）が戻されれば、すべてのプロセスは完了です。

```
*/
```

```
PAUSE Press <RETURN> to continue when DBA_REPCATLOG is empty.
```

```
/*
```

**手順 5 RESUME\_MASTER\_ACTIVITY プロシージャを使用してレプリケーション・アクティビティを再開します。**

```

*/

BEGIN
    DBMS_REPCAT.RESUME_MASTER_ACTIVITY (
        gname => 'hr_repg');
END;
/

SET ECHO OFF

SPOOL OFF

/***** END OF SCRIPT *****/

```

**エクスポート/インポートを使用するオフライン・インスタンスーションで新しいマスター・サイトを追加**

確立されたレプリケーション環境が拡張されると、ADD\_MASTER\_DATABASE プロシージャを使用して新しいマスター・サイトをレプリケーション環境に追加した場合に、ネットワークの通信量が増大します。これは、ネットワークを通じて、表またはマテリアライズド・ビューの内容全体が新しいレプリケート・サイトに伝播されるためです。

このようなネットワークの通信量を最少にするには、レプリケーション環境を拡張する場合にオフライン・インスタンスーション・プロシージャを使用します。オフライン・インスタンスーションでは、Oracle のエクスポート・ユーティリティおよびインポート・ユーティリティを利用できます。これによって、エクスポート・ファイルを作成し、CD-ROM やテープなどの記憶メディアを使用して新しいサイトにデータを転送できます。

次のスクリプト例は、マスター・サイトのオフライン・インスタンスーションを行う方法を示します。このスクリプトを実行すると、既存のマスター・グループに新しいマスター・サイトを追加する通常の方法と比べ、ネットワークの通信量をかなり絞り込むことができます。

これらのアクションを実行するには、次の要件を満たす必要があります。

**実行者:** レプリケーション管理者

**実行場所:** マスター定義サイトと新しいマスター・サイト

**レプリケーションの状態:** 静止中と部分的な再開

次の手順に従って、オフライン・インスタンスーションを使用してサイトをマスター・グループに追加します。

---

---

**注意：** このドキュメントをオンラインで参照している場合は、次の「BEGINNING OF SCRIPT」の行から 7-39 ページの「END OF SCRIPT」の行までのテキストをテキスト・エディタにコピーして編集し、使用している環境に適したスクリプトを作成します。

---

---

```
/***** BEGINNING OF SCRIPT *****/
```

### 手順 1 新しいマスター・サイトを設定します。

新しいマスター・サイトのオフライン・インスタンスーションを実行する前に、適切なスキーマおよびデータベース・リンクが作成されていることを確認します。新しいマスター・サイトから既存の各マスター・サイトへのデータベース・リンクの作成を確認します。既存の各マスター・サイトから新しいマスター・サイトへのデータベース・リンクも作成します。データベース・リンクの作成を終えた後、新しいデータベース・リンクごとにスケジュール・リンクも定義します。

#### 関連項目：

- 2-4 ページ「[マスター・サイトの設定](#)」
- 2-16 ページ「[マスター・サイト間のスケジュール・リンクの作成](#)」

```
*/
```

```
SET ECHO ON
```

```
SPOOL add_masters_instant.out
```

```
PAUSE Press <RETURN> to the new master site has been set up.
```

```
/*
```

### 手順 2 レプリケーション管理者としてマスター定義サイトに接続します。

```
*/
```

```
CONNECT repadmin/repadmin@orcl.world
```

```
/*
```

**手順 3 マスター・アクティビティを一時停止します。**

マスター・データのエクスポートおよびオフライン・インスタンス化・プロセスの開始の前に、既存のマスター・サイトのマスター・アクティビティを一時停止する必要があります。

```
*/  
  
BEGIN  
    DBMS_REPCAT.SUSPEND_MASTER_ACTIVITY (  
        gname => 'hr_repg');  
END;  
/  
  
/*
```

**手順 4 個別の SQL\*Plus セッションにはペンディング・トランザクションがないことを検証します。**

ここでは、未解決の遅延トランザクションのプッシュ、エラー・トランザクションの解決、および管理トランザクションのプッシュが含まれます。このステップは、既存の各マスター・サイトで実施する必要があります。

エラー・トランザクション・キューをチェックします。

```
SELECT * FROM DEFERROR;
```

遅延トランザクションがエラー・キューに入力された場合、エラーを解決するために手動で遅延トランザクションを再実行します。次に例を示します。

```
BEGIN  
    DBMS_DEFER_SYS.EXECUTE_ERROR (  
        deferred_tran_id => '128323',  
        destination => 'orc1.world');  
END;  
/
```

未処理の管理要求をチェックします。

```
SELECT * FROM DBA_REPCATLOG;
```

未処理の管理要求が残っている場合は、これらのトランザクションを手動でプッシュすることも、自動的に実行されるまで待機することも可能です。複数の手順を含む管理操作も存在するため、DBMS\_REPCAT.DO\_DEFERRED\_REPCAT\_ADMIN プロシージャを数回実行します。次に例を示します。

```
BEGIN
    DBMS_REPCAT.DO_DEFERRED_REPCAT_ADMIN (
        gname => 'hr_repg',
        all_sites => TRUE);
END;
/

*/

PAUSE Press <RETURN> to continue when you have verified that there are no pending
requests.

/*
```

### 手順 5 オフライン・インスタンスエーション・プロシージャを開始します。

```
*/

BEGIN
    DBMS_OFFLINE_OG.BEGIN_INSTANTIATION (
        gname => 'hr_repg',
        new_site => 'orc4.world');
END;
/

/*

DBA_REPCATLOG ビューが空になるまで待機します。このビューには、正常な実行後には消
去される一時的な情報が含まれます。他の SQL*Plus セッションで次の SELECT 文を実行し
て、DBA_REPCATLOG ビューを監視します。

SELECT * FROM DBA_REPCATLOG WHERE GNAME = 'HR_REPG';

*/

PAUSE Press <RETURN> to continue when DBA_REPCATLOG is empty.

/*
```

**手順 6 個別の端末ウィンドウで、hr/hr として接続してエクスポートします。**

Oracle のエクスポート・ユーティリティを使用して、新しいマスター・サイトに転送するエクスポート・ファイルを生成します。エクスポート・ファイルには新しいマスター・サイトで追加されるレプリケート・オブジェクトが含まれます。

**関連項目：** 追加情報は、『Oracle9i データベース・ユーティリティ』を参照してください。

hr スキーマのエクスポート・コマンドの例に示します。

```
exp hr/hr@orc1.world
```

---

---

**注意：** また、データベースのエクスポートを実行するには、コマンドライン・プロンプトで `exp system/manager` と入力し、以降のプロンプトに答える方法もあります。

---

---

```
*/
```

```
PAUSE Press <RETURN> to continue when the export is complete.
```

```
/*
```

**手順 7 レプリケーション・アクティビティを部分的に再開します。**

オフライン・インスタンスエーション・プロセスの完了には時間がかかるため、エクスポートの終了後、DBMS\_OFFLINE\_OG パッケージ内の RESUME\_SUBSET\_OF\_MASTERS プロシージャを実行することにより、残りのマスター・サイト（新しいマスター・サイトは除きます）のレプリケーション・アクティビティを再開できます。次の例では、新しいマスター・サイトの `orc4.world` を除くすべてのマスター・サイトでレプリケーション・アクティビティが再開されます。

```
*/
```

```
BEGIN
```

```
  DBMS_OFFLINE_OG.RESUME_SUBSET_OF_MASTERS (
    gname => 'hr_repg',
    new_site => 'orc4.world');
```

```
END;
```

```
/
```

```
/*
```

**手順 8 レプリケーション管理者として新しいマスター・サイトに接続します。**

\*/

```
CONNECT repadmin/repadmin@orc4.world
```

/\*

**手順 9 新しいマスター・サイトを準備します。**

マスター定義サイトから新しいマスター・サイトへエクスポート・ファイルを転送した後、新しいサイトを準備してエクスポート・ファイルにデータをインポートします。新しいマスター・サイトで次のプロシージャを必ず実行します。

\*/

```
BEGIN
```

```
    DBMS_OFFLINE_OG.BEGIN_LOAD (  
        gname => 'hr_repg',  
        new_site => 'orc4.world');
```

```
END;
```

```
/
```

/\*

**手順 10 個別の端末ウィンドウで、エクスポート・ファイルからデータをインポートします。**

以前に生成したエクスポート・ファイルのインポート後、マスター定義サイトから新しいマスター・サイトへデータを転送します。

hr スキーマのインポート・コマンドの例を次に示します。

```
imp hr/hr@orc4.world FULL=y IGNORE=y
```

---

---

**注意：** また、データベースのインポートを実行するには、コマンドライン・プロンプトで `imp system/manager` と入力し、以降のプロンプトに答える方法もあります。

---

---

\*/

```
PAUSE Press <RETURN> to continue when the import is complete.
```

/\*



**手順 11 新しいマスター・サイトのロード・プロセスを完了します。**

エクスポート・ファイルをインポートすれば、新しいマスター・サイトのオフライン・インスタンス化・プロセスを完了できます。DBMS\_OFFLINE\_OG.END\_LOAD プロシージャを実行して、通常のレプリケーション・アクティビティ用の新しいサイトを準備します。

```
*/

BEGIN
    DBMS_OFFLINE_OG.END_LOAD (
        gname => 'hr_repg',
        new_site => 'orc4.world');
END;
/

/*
```

**手順 12 レプリケーション管理者としてマスター定義サイトに接続します。**

```
*/

CONNECT repadmin/repadmin@orc1.world

/*
```

**手順 13 インスタンス化処理を完了します。**

新しいマスター・サイトの手順を完了すれば、オフライン・インスタンス化・プロセスを完了できます。DBMS\_OFFLINE\_OG パッケージの END\_INSTANTIATION プロシージャを実行することで、プロセスを完了して、すべてのマスター・サイトにおける通常のレプリケーション・アクティビティを再開します。マスター定義サイトで次のプロシージャを必ず実行します。

```
*/

BEGIN
    DBMS_OFFLINE_OG.END_INSTANTIATION (
        gname => 'hr_repg',
        new_site => 'orc4.world');
END;
/

SET ECHO OFF

SPOOL OFF

/***** END OF SCRIPT *****/
```

## マスター・グループからマスター・サイトの削除

マスター・グループからマスター・サイトを削除する場合は、`REMOVE_MASTER_DATABASES` プロシージャを使用して 1 つ以上のマスター・サイトを削除します。

これらのアクションを実行するには、次の要件を満たす必要があります。

**実行者:** レプリケーション管理者

**実行場所:** マスター定義サイト

**レプリケーションの状態:** 静止中

次に示す手順に従って、マスター・サイトを削除します。

---

---

**注意:** このドキュメントをオンラインで参照している場合は、次の「BEGINNING OF SCRIPT」の行から 7-41 ページの「END OF SCRIPT」の行までのテキストをテキスト・エディタにコピーして編集し、使用している環境に適したスクリプトを作成します。

---

---

```
/***** BEGINNING OF SCRIPT *****/
```

**手順 1 レプリケーション管理者としてマスター定義サイトに接続します。**

```
*/
```

```
SET ECHO ON
```

```
SPOOL remove_masters.out
```

```
CONNECT repadmin/repadmin@orcl.world
```

```
/*
```

**手順 2 マスター・グループのレプリケーションの状態が通常の場合は、静止中に変更してください。**

```
*/
```

```
BEGIN
```

```
    DBMS_REPCAT.SUSPEND_MASTER_ACTIVITY (  
        gname => 'hr_repg');
```

```
END;
```

```
/
```

```
/*
```

**手順 3 REMOVE\_MASTER\_DATABASES プロシージャを使用してマスター・サイトを削除します。**

\*/

BEGIN

```
DBMS_REPCAT.REMOVE_MASTER_DATABASES (
    gname => 'hr_repg',
    master_list => 'orc4.world');
```

END;

/

/\*

DBA\_REPCATLOG ビューが空になるまで待機します。他の SQL\*Plus セッションで次の SELECT 文を実行して、DBA\_REPCATLOG ビューを監視します。

```
SELECT * FROM DBA_REPCATLOG WHERE GNAME = 'HR_REPG';
```

\*/

PAUSE Press <RETURN> to continue when DBA\_REPCATLOG is empty for the master group.

/\*

**手順 4 マスター・グループのマスター・アクティビティを再開します。**

\*/

BEGIN

```
DBMS_REPCAT.RESUME_MASTER_ACTIVITY (
    gname => 'hr_repg');
```

END;

/

SET ECHO OFF

SPOOL OFF

/\*\*\*\*\* END OF SCRIPT \*\*\*\*\*/

## 使用不可能なマスター・サイトの削除

マスター・グループから削除するサイトはアクセス可能である必要はありません。システムまたはネットワークの障害により、長期間、マスター・サイトが使用できない場合は、マスター・グループからマスター・サイトを削除できます。

しかし、サイトが使用不可能であるため、多くの場合、マスター・グループのレプリケーション・アクティビティを一時停止することはできません。この場合、DBMS\_REPCAT パッケージの REMOVE\_MASTER\_DATABASES プロシージャを使用することで、マスター・グループが静止中でない場合でも、マスター・グループからマスター・サイトを削除できます。

この場合の対処を次に示します。

- 遅延トランザクション・キューのクリーン
- 非一貫性データの削除

具体的には、マスター・グループのレプリケーション・アクティビティを次に一時停止するときは、使用不可能なマスター・サイトが削除された後すぐに、次の手順を完了する必要があります。

**手順 1 マスター・グループのレプリケーション・アクティビティを一時停止します。詳細は、20-106 ページの「[SUSPEND\\_MASTER\\_ACTIVITY プロシージャ](#)」を参照してください。**

**手順 2 遅延トランザクションの接続先が削除済のマスター・サイトである場合は、そのマスター・サイトからすべての遅延トランザクションを削除します。詳細は、14-7 ページの「[DELETE\\_TRAN プロシージャ](#)」を参照してください。**

**手順 3 削除済のマスター・サイトから、すべての遅延トランザクションを削除します。詳細は、14-7 ページの「[DELETE\\_TRAN プロシージャ](#)」を参照してください。**

**手順 4 残りの各マスター・サイトで、すべてのエラー・トランザクションを再実行または削除します。エラー・トランザクションの再実行の詳細は、9-18 ページの「[エラー・キューの管理](#)」を、エラー・トランザクションの削除の詳細は、14-7 ページの「[DELETE\\_TRAN プロシージャ](#)」をそれぞれ参照してください。**

**手順 5 遅延トランザクションまたはエラー・トランザクションが、現存の各マスターに存在しないことを確認してください。現存のマスターから 1 つ以上の遅延トランザクションが削除できない場合は、マスター・サイトで DBMS\_DEFER\_SYS.DELETE\_TRAN プロシージャを実行します。**

**手順 6 レプリケート・データがすべて一貫したデータであることを確認します。違いの判別と訂正の詳細は、[第 18 章「DBMS\\_RECTIFIER\\_DIFF」](#)を参照してください。**

手順 7 マスター・グループのレプリケーション・アクティビティを再開します。詳細は、20-98 ページの「[RESUME\\_MASTER\\_ACTIVITY プロシージャ](#)」を参照してください。

**注意：** マスター・グループから使用不可能なマスター・サイトを削除した後は、削除したサイトからマスター・グループを削除してクリーン・アップを終了します。

# データ・ディクショナリ・ビュー内のコメント・フィールドの更新

DBMS\_REPCAT パッケージには、レプリケーションに関連する様々なデータ・ディクショナリ・ビュー内のコメント情報を更新できるプロシージャがいくつかあります。表 7-1 に、各ビュー用にコールするプロシージャの一覧を示します。

表 7-1 アドバンスド・レプリケーション機能のビュー内のコメントの更新

ビュー	DBMS_REPCAT プロシージャ	パラメータ情報の参照先
DBA_REPGROUP	COMMENT_ON_REPGROUP ( gname                IN VARCHAR2, comment              IN VARCHAR2)	20-41 ページ「 <a href="#">COMMENT_ON_REPGROUP プロシージャ</a> 」
DBA_REPOBJECT	COMMENT_ON_REPOBJECT ( sname                IN VARCHAR2, oname                IN VARCHAR2, type                  IN VARCHAR2, comment              IN VARCHAR2)	20-42 ページ「 <a href="#">COMMENT_ON_REPOBJECT プロシージャ</a> 」
DBA_REPSITES	COMMENT_ON_REPSITES ( gname                IN VARCHAR2, master               IN VARCHAR, comment              IN VARCHAR2)	20-43 ページ「 <a href="#">COMMENT_ON_REPSITES プロシージャ</a> 」
DBA_REPCOLUMN_GROUP	COMMENT_ON_COLUMN_GROUP ( sname                IN VARCHAR2, oname                IN VARCHAR2, column_group          IN VARCHAR2, comment              IN VARCHAR2)	20-38 ページ「 <a href="#">COMMENT_ON_COLUMN_GROUP プロシージャ</a> 」
DBA_REPPRIORITY_GROUP	COMMENT_ON_PRIORITY_GROUP ( gname                IN VARCHAR2, pgroup               IN VARCHAR2, comment              IN VARCHAR2)	20-40 ページ「 <a href="#">COMMENT_ON_PRIORITY_GROUP/COMMENT_ON_SITE_PRIORITY プロシージャ</a> 」

表 7-1 アドバンスト・レプリケーション機能のビュー内のコメントの更新（続き）

ビュー	DBMS_REPCAT プロシージャ	パラメータ情報の参照先
DBA_REPPRIORITY_GROUP (サイト優先順位グループ)	COMMENT_ON_SITE_PRIORITY( sname                IN VARCHAR2, name                IN VARCHAR2, comment              IN VARCHAR2)	20-40 ページ「 <a href="#">COMMENT_ON_PRIORITY_GROUP/COMMENT_ON_SITE_PRIORITY</a> プロシージャ」
DBA_REPRESOLUTION (一意性競合)	COMMENT_ON_UNIQUE_RESOLUTION( sname                IN VARCHAR2, oname                IN VARCHAR2, constraint_name      IN VARCHAR2, sequence_no          IN NUMBER, comment              IN VARCHAR2)	COMMENT_ON_UNIQUE_RESOLUTION プロシージャのパラメータは、20-45 ページの「 <a href="#">COMMENT_ON_conflicttype_RESOLUTION</a> プロシージャ」を参照してください。
DBA_REPRESOLUTION (更新競合)	COMMENT_ON_UPDATE_RESOLUTION( sname                IN VARCHAR2, oname                IN VARCHAR2, column_group         IN VARCHAR2, sequence_no          IN NUMBER, comment              IN VARCHAR2)	COMMENT_ON_UNIQUE_RESOLUTION プロシージャのパラメータは、20-45 ページの「 <a href="#">COMMENT_ON_conflicttype_RESOLUTION</a> プロシージャ」を参照してください。
DBA_REPRESOLUTION (削除競合)	COMMENT_ON_DELETE_RESOLUTION( sname                IN VARCHAR2, oname                IN VARCHAR2, sequence_no          IN NUMBER, comment              IN VARCHAR2)	COMMENT_ON_UNIQUE_RESOLUTION プロシージャのパラメータは、20-45 ページの「 <a href="#">COMMENT_ON_conflicttype_RESOLUTION</a> プロシージャ」を参照してください。

## プロシージャ・レプリケーションの使用

プロシージャ・レプリケーションは、レプリケーション環境内で逐次実行され、多くの行を操作する大規模なバッチ指向の操作に有効です。

これを応用したふさわしい例がパージ操作です。これは、アーカイブ操作とも呼ばれ、空き時間に不定期に実行（たとえば、四半期に1回）して、旧データまたはオンライン・データベースから論理的に削除されたデータを削除します。プロシージャ・レプリケーションを使用して削除された行をパージする例は、『Oracle9i アドバンスド・レプリケーション』の第5章「削除の競合の回避」の項、「競合解消の概要およびアーキテクチャ」で説明しています。

## プロシージャ・レプリケーションの制限事項

レプリケート・プロシージャのパラメータはすべて、IN パラメータです。OUT および IN/OUT モードはサポートされていません。これらのパラメータでは次のデータ型がサポートされます。

- VARCHAR2
- NVARCHAR2
- NUMBER
- DATE
- RAW
- ROWID
- CHAR
- NCHAR
- バイナリ・ラージ・オブジェクト（Binary Large Object: BLOB）（BLOB）
- キャラクタ・ラージ・オブジェクト（Character Large Object: CLOB）（CLOB）
- 各国語キャラクタ・ラージ・オブジェクト（National Character Large Object: NCLOB）（NCLOB）
- ユーザー定義データ型

Oracle では、レプリケート・プロシージャが生成した更新の競合が検出されません。レプリケート・プロシージャ自らが、競合の検出と解消を行う必要があります。独自の競合解消ルーチンの表現は困難なため、競合の可能性を単純に回避することが最良です。

次のガイドラインに従うことで、プロシージャ・レプリケーションを使用するすべてのサイトで、表の一貫性を維持することができます。

- 遅延プロシージャ本体の行レベルのレプリケーションを無効化する必要があります。  
7-43 ページの「[データ・ディクショナリ・ビュー内のコメント・フィールドの更新](#)」を参照してください。
- 7-48 ページの「[トランザクションのシリアル化](#)」で説明されているように、一度に実行できるレプリケート・プロシージャは1つのみです。
- 遅延トランザクションの伝播は逐次行います。スケジュール・リンクのガイドラインの詳細は、『[Oracle9i アドバンスド・レプリケーション](#)』を参照してください。
- レプリケート・プロシージャはパッケージ化が必要です。ただし、パッケージにファンクションを含めることはできません。スタンドアロン遅延プロシージャおよび、スタンドアロンまたはパッケージ化遅延ファンクションは現在サポートされていません。
- 遅延プロシージャは所有データをローカルでのみ参照します。
- このプロシージャは、ローカルで生成されたフィールド、値または環境に依存する SQL ファンクションを使用しません。たとえば、このプロシージャは `SYSDATE` をコールしません。
- データの所有権は静的に分割されます。すなわち、行の所有権はサイト間で変更されません。
- マスター・サイトに複数のマスター・グループを所有していて、1つ以上のマスター・グループが静止中の場合は、マスター・サイトのマスター・グループに対するプロシージャ・レプリケーションは実行できません。あるマスター・グループのプロシージャは他のマスター・グループのオブジェクトを更新できるため、この制限が規定されています。マスター・サイトのすべてのマスター・グループがデータを正常にレプリケートした場合にのみ、プロシージャ・レプリケーションを実行できます（すなわち、静止中のマスター・グループがない場合）。

たとえば、マスター・サイト `db1` のマスター・グループ `A` に `sal_raise` という名前のプロシージャを所有している場合で、マスター・サイト `db1` のマスター・グループ `B` が静止中の場合は、マスター・グループ `A` がレプリケーションを正常に終了しても、`sal_raise` プロシージャを実行できません。

- プロシージャ・レプリケーションを使用する場合は、プロシージャ・コールのみがマスター・レプリケーション・サイトに伝播されます。プロシージャ・コールは、マテリアライズド・ビュー・サイトに伝播されません。しかし、マテリアライズド・ビュー・サイトではプロシージャ・レプリケーションを開始できます。この場合、プロシージャ・コールはレプリケーション環境のすべてのマスター・サイトに伝播されますが、マテリアライズド・ビュー・サイトには伝播されません。他のマテリアライズド・ビュー・サイトは、マテリアライズド・ビューをリフレッシュして、マスター・サイトの変更を取り込む必要があります。



たとえば、レプリケーション環境に `msite1` および `msite2` という名前の 2 つのマスター・サイトと `mview1` および `mview2` という名前の 2 つのマテリアライズド・ビュー・サイトが含まれているものと仮定します。`mview1` でプロシージャ・レプリケーションが開始されると、`mview1` でこのプロシージャが実行されます。また、プロシージャ・コールが 2 つのマスター・サイト、`msite1` および `msite2` に伝播され、プロシージャが実行されます。しかし、プロシージャ・コールは `mview2` に伝播されません。このため、次のリフレッシュの間に、`mview2` はマスター・サイトのプロシージャが行ったすべての変更をプルします。

## ユーザー定義型とプロシージャ・レプリケーション

プロシージャ・レプリケーションを使用する場合、プロシージャが参照するユーザー定義型およびユーザー定義のオブジェクトは次の条件を満たす必要があります。

- オブジェクト型に関しては、すべてのレプリケーション・サイトは、オブジェクト型の属性の順序に従う必要があります。オブジェクト型を作成する場合は、属性の順序を設定します。次のオブジェクト型を考慮します。

```
CREATE TYPE cust_address_typ AS OBJECT
(
  street_address  VARCHAR2(40),
  postal_code     VARCHAR2(10),
  city            VARCHAR2(30),
  state_province  VARCHAR2(10),
  country_id      CHAR(2)
);
/
```

すべてのレプリケーション・サイトにおいて、`street_address` が第 1 属性、`postal_code` が第 2 属性、`city` が第 3 属性であることなどです。

- Oracle オブジェクトに関しては、すべてのレプリケーション・サイトは同一のオブジェクト識別子 (OID)、スキーマ所有者および各レプリケート・オブジェクト型のタイプ名を所有する必要があります。

分散スキーマ管理を常に使用して、オブジェクト型、列オブジェクトを持つ表およびオブジェクト表などのレプリケート・オブジェクトを作成または修正することで、これらの条件を満たすことができます。オブジェクト型の作成や修正に分散スキーマ管理を使用しない場合は、レプリケーション・エラーが発生することがあります。

**関連項目：** レプリケーション・サイトにおける型の取決めの詳細は、『Oracle9i アドバンスド・レプリケーション』を参照してください。

## トランザクションのシリアル化

逐次実行ではデータの一貫性が維持できます。レプリケーション機能では、一度に1つのレプリケート・トランザクションを伝播および実行します。たとえば、ローカル・データの更新を行う2つのプロシージャ、AとBがあるとします。この条件で次のアクションを順番に実行するとします。

**手順1 AとBをローカルで実行します。**

**手順2 別のノード上にあるAとBの他のレプリカの実行要求をキューします。**

**手順3 コミットします。**

別のノードにあるAとBのレプリカは、起点となるサイトでコミットされた順番通りに、逐次実行されます。AとBが起点となるサイトで同時に実行されると、ローカルとリモートとでは結果が異なることになります。起点となるサイトでAとBを逐次実行した場合は、すべてのサイトで同一の結果が得られます。トランザクションを逐次伝播することで、AとBはターゲット・サイトで必ず順番に実行されます。

一方、シリアル化を確実にを行うために、プロシージャへの書込みは慎重に行います。たとえば、パラレル伝播を使用している場合は、起点となるサイトおよびターゲット・サイトにおけるシリアル化を確実にを行うための問合せには、`SELECT... FOR UPDATE` を使用します。

## レプリケート・プロシージャのサポートの生成

プロシージャの開始時には行レベルのレプリケーション・サポートを無効化し、終了時にはサポートを再び有効化する必要があります。この操作により、プロシージャを実行したことで発生する更新が他のサイトに伝播されることがなくなります。行レベルのレプリケーションの有効化および無効化は、次のプロシージャをコールすることで行います。

- `DBMS_REPUTIL.REPLICATION_ON`
- `DBMS_REPUTIL.REPLICATION_OFF`

### 関連項目：

- 9-6 ページ [「レプリケーションの使用禁止」](#)
- 24-3 ページ [「REPLICATION\\_ON プロシージャ」](#)
- 24-3 ページ [「REPLICATION\\_OFF プロシージャ」](#)

レプリケート・パッケージのレプリケーション・サポートを生成すると、Oracle では、レプリケーション・プロパゲータのスキーマに `Wrapper` パッケージが作成されます。

---

**注意：** 現行のプロパゲータを登録解除すると、プロパゲータのスキーマにある既存の生成済ラッパーのすべてが削除されます。新しいプロパゲータを登録した後は、ラップされたストアド・プロシージャのレプリケーション・サポートを再生成します。

---

Wrapper パッケージの名前はオリジナル・パッケージと同じですが、このプロシージャのレプリケーション・サポートを生成したときに指定した文字列の接頭辞が付けられています。接頭辞を指定しないと、Oracle では、デフォルトの接頭辞、`defer_` が使用されます。ラッパー・プロシージャは、`call_local` および `call_remote` の 2 つの追加パラメータ以外にオリジナルと同じパラメータを所有しています。これら 2 つの CHAR パラメータで、プロシージャが実行されている場所を特定します。`call_local` が 'Y' の場合は、ローカルでプロシージャが実行されます。`call_remote` が 'Y' の場合は、レプリケーション環境のすべてのマスター・サイトでプロシージャが実行されます。

同期式で変更を伝播する場合、または非同期式で変更を伝播する場合に、これらのプロシージャに対するコールが遅延トランザクション・キューに追加された場合は、リモート・プロシージャが直接コールされます。デフォルトでは、`call_local` は 'N' で、`call_remote` は 'Y' です。

Oracle では、パッケージのレプリケーション・サポートが 2 つのフェーズで生成されます。フェーズ 1 では、全サイトのパッケージ仕様部が作成されます。フェーズ 2 では、全サイトのパッケージ本体が生成されます。同期レプリケーションをサポートするには、これら 2 つのフェーズが必要です。

たとえば、1 つの引数、`email` を取得するプロシージャ `new_dept` を含む `emp_mgmt` パッケージの作成を想定します。このパッケージをシステム内のすべてのマスター・サイトにレプリケートする場合、レプリケーション管理ツールを使用してパッケージをマスター・グループに追加して、オブジェクトのレプリケーション・サポートを生成します。これらの手順を完了すれば、アプリケーションは、次に示すレプリケート・パッケージのプロシージャをコールできます。

```
BEGIN
defer_emp_mgmt.new_dept( email      => 'jones',
                        call_local  => 'Y',
                        call_remote => 'Y');

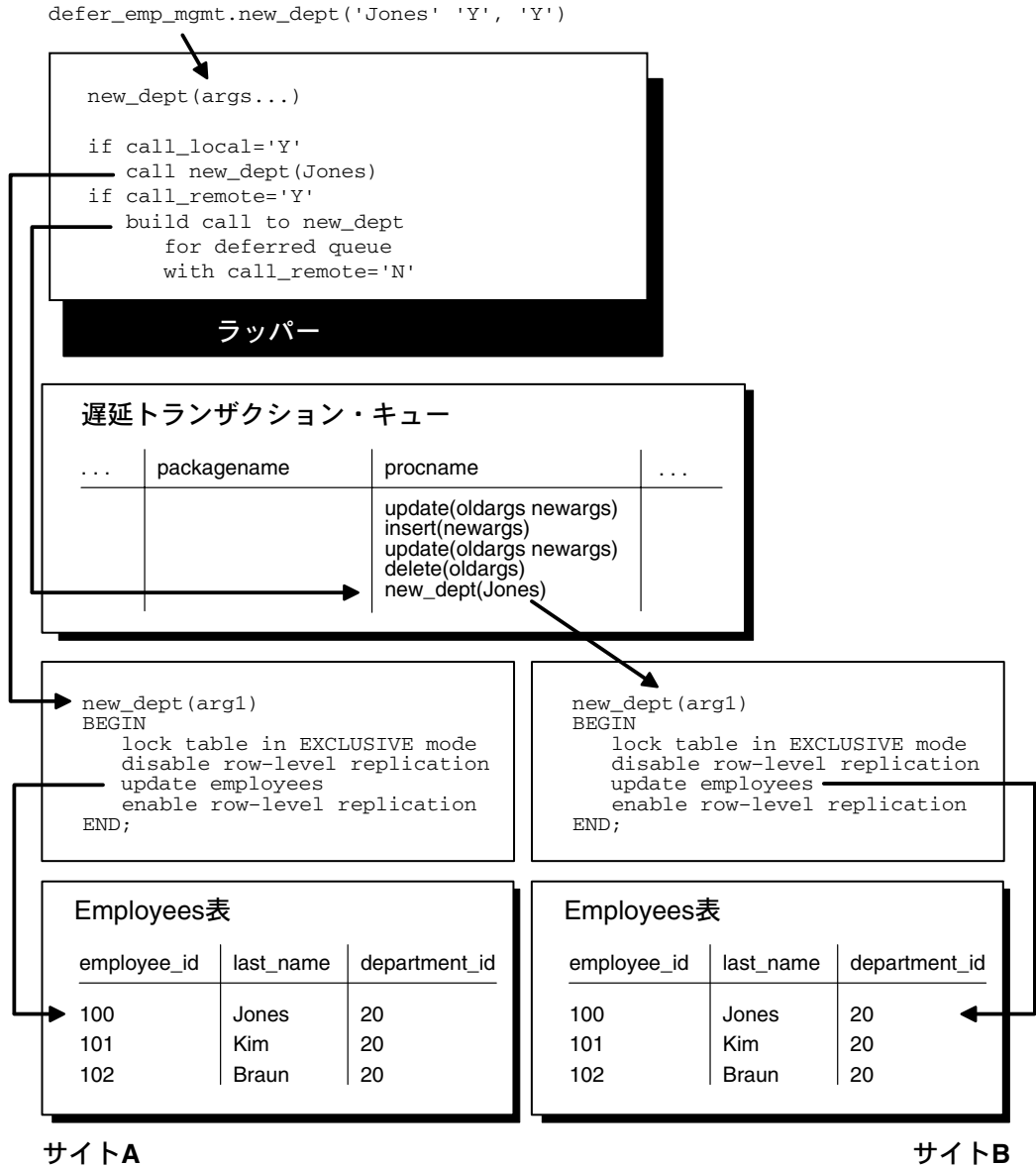
END;
/
```

**関連項目：** レプリケーション管理ツールを使用したマスター・グループおよびレプリケート・オブジェクトの管理の詳細は、レプリケーション管理ツールのオンライン・ヘルプを参照してください。

図 7-4 で説明されているように、ラッパー・プロシージャの論理により、ローカル・サイトおよびすべてのリモート・サイトでのプロシージャのコールが確実に行えます。ラッパー・プロシージャの論理により、レプリケート・プロシージャがリモート・サイトでコールされ、`call_remote` が `FALSE` の場合、プロシージャが先に伝播されることも回避できます。

データ所有権の静的分割を伴う複合レプリケーション環境で操作を行っている場合（すなわち、行レベルのレプリケーションを妨げない場合）は、行レベル・レプリケーションおよびプロシージャ・レプリケーションの両方が同じ非同期キューを使用するため、Advanced Replication では、リモート・ノードでの操作順序が維持されます。

図 7-4 非同期プロシージャ・レプリケーション





---

# マテリアライズド・ビュー・レプリケーション 環境の管理

マテリアライズド・ビュー・レプリケーションは、ユーザーおよびフロントエンド・アプリケーションの必要を満たすと同時に、セキュリティ構成の要件を満たす、データ・セットの構築に柔軟性を提供します。この章では、レプリケーション・マネージメント API を使用したマテリアライズド・ビュー・サイトの管理方法を説明します。

この章では、次の項目を説明します。

- [マテリアライズド・ビューのリフレッシュ](#)
- [マテリアライズド・ビュー・グループのマスター・サイトの変更](#)
- [マテリアライズド・ビュー・グループおよびオブジェクトの削除](#)
- [マテリアライズド・ビュー・ログの管理](#)
- [エクスポート / インポートによるマテリアライズド・ビュー・サイトのオフライン・インスタンス化](#)
- [マテリアライズド・ビュー・グループに対するグループ所有者の使用](#)

## マテリアライズド・ビューのリフレッシュ

マテリアライズド・ビューのリフレッシュにより、マテリアライズド・ビューのマスターのデータとマテリアライズド・ビューのデータが同期化されます。リフレッシュ・グループのすべてのマテリアライズド・ビューを一度にリフレッシュすることも、マテリアライズド・ビューを個別にリフレッシュすることもできます。マテリアライズド・ビュー・サイトで複数のマテリアライズド・ビューに依存するアプリケーションがある場合、アプリケーションで使用されるすべてのマテリアライズド・ビューでデータ変換に整合性が保たれるよう、リフレッシュ・グループの使用をお勧めします。

次の例では、hr\_refg リフレッシュ・グループがリフレッシュされます。

```
EXECUTE DBMS_REFRESH.REFRESH ('hr_refg');
```

次の例では、hr.departments\_mv マテリアライズド・ビューがリフレッシュされます。

```
BEGIN
    DBMS_MVIEW.REFRESH (
        list    => 'hr.departments_mv',
        method => '?');
END;
/
```

## マテリアライズド・ビュー・グループのマスター・サイトの変更

レベル1のマテリアライズド・ビュー・サイトのマテリアライズド・ビュー・グループのマスター・サイトを他のマスター・サイトに変更するには、次の例に示すように、DBMS\_REPCAT パッケージの SWITCH\_MVIEW\_MASTER プロシージャをコールします。

```
BEGIN
    DBMS_REPCAT.SWITCH_MVIEW_MASTER (
        gname => 'hr_repg',
        master => 'orc3.world');
END;
/
```

ここでは、hr\_repg レプリケーション・グループのマスター・サイトが orc3.world マスター・サイトに変更されます。このプロシージャは、マスター・サイトを変更するマテリアライズド・ビュー・サイトでコールする必要があります。また、新しいデータベースとして、レプリケーション環境のマスター・サイトを指定する必要があります。このプロシージャをコールすると、新しいマスターを使用して、ローカル・マテリアライズド・ビュー・グループのマテリアライズド・ビューごとに完全リフレッシュが実行されます。SWITCH\_MVIEW\_MASTER プロシージャを実行する前に、必ず新しいマスター・サイトを使用するためのマテリアライズド・ビュー・サイトを設定してください。



マスター・サイトが変更されたマテリアライズド・ビューの古いマスター・サイトにある SYS.SLOG\$ 表のエントリは削除されません。このため、古いマスター・サイトにある、更新可能なマテリアライズド・ビュー・ログ (MLOG\$ 表) は際限なく増大する可能性があります。これを回避するには、DBMS\_MVIEW.PURGE\_LOG をコールしてパージします。

---

**注意：** 他のマテリアライズド・ビューをベースとするマテリアライズド・ビュー（レベル 2 以上のマテリアライズド・ビュー）のマスターは変更できません。このようなマテリアライズド・ビューのベースを別のマスターにするには、マテリアライズド・ビューを削除して再作成する必要があります。

---

**関連項目：** 2-20 ページ「マテリアライズド・ビュー・サイトの設定」

## マテリアライズド・ビュー・グループおよびオブジェクトの削除

マテリアライズド・ビュー・サイトのレプリケーション・アクティビティを削除する必要性は、様々な場合に生じます。たとえば、データ要件が変更されたり、従業員が退職した場合などです。いずれの場合も、データベース管理者がターゲット・マテリアライズド・ビュー・サイトに対するレプリケーション・サポートを削除する必要があります。

この項では、次の項について説明します。

- [配置テンプレートで作成されたマテリアライズド・ビュー・グループの削除](#)
- [手動で作成されたマテリアライズド・ビュー・グループまたはオブジェクトの削除](#)
- [マスター・サイトまたはマスター・マテリアライズド・ビュー・サイトのクリーン・アップ](#)

## 配置テンプレートで作成されたマテリアライズド・ビュー・グループの削除

配置テンプレートで作成されたマテリアライズド・ビュー・グループの場合は、リモート・マテリアライズド・ビュー・サイトで削除する前に、このマテリアライズド・ビュー・グループのターゲット・マスター・サイトで DROP\_SITE\_INSTANTIATION プロシージャを実行する必要があります。このプロシージャは、マテリアライズド・ビュー・グループに関連するメタデータを削除します。また、そのサイトに関する配置テンプレートのデータも削除します。

DROP\_SITE\_INSTANTIATION プロシージャにはパブリック・バージョンとプライベート・バージョンがあります。パブリック・バージョンでは、マテリアライズド・ビュー・グループの所有者がマテリアライズド・ビュー・サイトを削除できます。これに対して、プライベート・バージョンでは、マテリアライズド・ビュー・グループの所有者のかわりにレプリケーション管理者が、マテリアライズド・ビュー・サイトを削除します。

## パブリックの DROP\_SITE\_INSTANTIATION の使用

これらのアクションを実行するには、次の要件を満たす必要があります。

**実行者：**

- マスター・サイトのマテリアライズド・ビュー・グループ所有者
- マテリアライズド・ビュー・サイトのマテリアライズド・ビュー管理者

**実行場所：**

- ターゲット・マテリアライズド・ビュー・サイト用マスター・サイト
- マテリアライズド・ビュー・サイト

**レプリケーションの状態：**通常

次に示す手順に従って、配置テンプレートで作成されたマテリアライズド・ビュー・グループを削除します。

---

---

**注意：** このドキュメントをオンラインで参照している場合は、次の「BEGINNING OF SCRIPT」の行から 8-6 ページの「END OF SCRIPT」の行までのテキストをテキスト・エディタにコピーして編集し、使用している環境に適したスクリプトを作成します。

---

---

```
/***** BEGINNING OF SCRIPT *****/
```

**手順 1 マテリアライズド・ビュー・グループ所有者としてマスター・サイトに接続します。**

```
*/
```

```
SET ECHO ON
```

```
SPOOL drop_mv_group_public.out
```

```
CONNECT hr/hr@orc3.world
```

```
/*
```

**手順 2 インスタンス化されたマテリアライズド・ビュー・サイトをマスター・サイトから削除します。**

```
*/  
  
BEGIN  
    DBMS_REPCAT_INSTANTIATE.DROP_SITE_INSTANTIATION(  
        refresh_template_name => 'hr_refg_dt',  
        site_name => 'orc3.world');  
END;  
/  
  
/*
```

**手順 3 マテリアライズド・ビュー管理者としてリモート・マテリアライズド・ビュー・サイトに接続します。**

```
*/  
  
CONNECT mviewadmin/mviewadmin@mv4.world  
  
/*
```

リモート・マテリアライズド・ビュー・サイトに接続できない場合は、ターゲット・マテリアライズド・ビュー・グループはリフレッシュできませんが、既存のデータはまだマテリアライズド・ビュー・サイトに残っています。

**手順 4 マテリアライズド・ビュー・グループを削除します。**

```
*/  
  
BEGIN  
    DBMS_REPCAT.DROP_MVIEW_REPGROUP (  
        gname => 'hr_repg',  
        drop_contents => TRUE);  
END;  
/  
  
/*
```

マテリアライズド・ビュー・グループの内容をマテリアライズド・ビュー・データベースから物理的に削除する場合は、`drop_contents` パラメータを `TRUE` に指定してください。

#### 手順 5 リフレッシュ・グループを削除します。

```
*/  
  
BEGIN  
    DBMS_REFRESH.DESTROY (  
        name => 'hr_refg');  
END;  
/  
  
SET ECHO OFF  
  
SPOOL OFF  
  
/***** END OF SCRIPT *****/
```

### プライベートの DROP\_SITE\_INSTANTIATION の使用

次のスナップショットをマテリアライズド・ビュー・グループの所有者のかわりに、レプリケーション管理者が実行します。これらのアクションを実行するには、次の要件を満たす必要があります。

#### 実行者：

- マスター・サイトのレプリケーション管理者
- マテリアライズド・ビュー・サイトのマテリアライズド・ビュー管理者

#### 実行場所：

- ターゲット・マテリアライズド・ビュー・サイト用マスター・サイト
- マテリアライズド・ビュー・サイト

#### レプリケーションの状態：通常

次に示す手順に従って、配置テンプレートで作成されたマテリアライズド・ビュー・グループを削除します。

---

---

**注意：** このドキュメントをオンラインで参照している場合は、次の「BEGINNING OF SCRIPT」の行から 8-8 ページの「END OF SCRIPT」の行までのテキストをテキスト・エディタにコピーして編集し、使用している環境に適したスクリプトを作成します。

---

---

```
/***** BEGINNING OF SCRIPT *****/
```

### 手順1 レプリケーション管理者としてマスター・サイトに接続します。

```
*/
```

```
SET ECHO ON
```

```
SPOOL drop_mv_group_private.out
```

```
CONNECT repadmin/repadmin@orc1.world
```

```
/*
```

### 手順2 インスタンス化されたマテリアライズド・ビュー・サイトをマスター・サイトから削除します。

```
*/
```

```
BEGIN
```

```
  DBMS_REPCAT_RGT.DROP_SITE_INSTANTIATION (  
    refresh_template_name => 'hr_refg_dt',  
    user_name => 'hr',  
    site_name => 'orc3.world');
```

```
END;
```

```
/
```

```
/*
```

### 手順3 マテリアライズド・ビュー管理者としてリモート・マテリアライズド・ビュー・サイトに接続します。

```
*/
```

```
CONNECT mviewadmin/mviewadmin@mv4.world
```

```
/*
```

リモート・マテリアライズド・ビュー・サイトに接続できない場合は、ターゲット・マテリアライズド・ビュー・グループはリフレッシュできませんが、既存のデータはまだマテリアライズド・ビュー・サイトに残っています。

**手順 4 マテリアライズド・ビュー・グループを削除します。**

```
*/  
  
BEGIN  
    DBMS_REPCAT.DROP_MVIEW_REPGROUP (  
        gname => 'hr_repg',  
        drop_contents => TRUE  
        gowner => 'hr');  
END;  
/  
  
/*
```

マテリアライズド・ビュー・グループの内容をマテリアライズド・ビュー・データベースから物理的に削除する場合は、drop\_contents パラメータを TRUE に指定してください。

**手順 5 リフレッシュ・グループを削除します。**

```
*/  
  
BEGIN  
    DBMS_REFRESH.DESTROY (  
        name => 'hr_refg');  
END;  
/  
  
SET ECHO OFF  
  
SPOOL OFF  
  
/***** END OF SCRIPT *****/
```

## 手動で作成されたマテリアライズド・ビュー・グループまたはオブジェクトの削除

マテリアライズド・ビュー・サイトのレプリケーション・サポートを削除する場合の最も安全な方法は、マテリアライズド・ビュー・サイトのレプリケート・オブジェクトまたはグループを物理的に削除することです。次の2つの項では、マテリアライズド・ビュー・グループに接続した状態で、これらのオブジェクトとグループを削除する方法を示します。

ここに示すプロシージャは、マテリアライズド・ビューが、ターゲット・マスター・サイトまたはマスター・マテリアライズド・ビュー・サイトに接続した状態で実行する必要があります。接続していると、マスター・サイトまたはマスター・マテリアライズド・ビュー・サイトにあるすべての関連メタデータを、確実に削除できるためです。マスター・サイトまたはマスター・マテリアライズド・ビュー・サイトに接続できない場合は、8-11 ページの「[マスター・サイトまたはマスター・マテリアライズド・ビュー・サイトのクリーン・アップ](#)」に示すスクリプトを実行して、関連メタデータを手動で削除します。

### 手動で作成されたマテリアライズド・ビュー・グループの削除

マテリアライズド・ビュー・グループをマテリアライズド・ビュー・サイトから削除することが必要になった場合は、DROP\_MVIEW\_REPGROUP プロシージャを使用してマテリアライズド・ビュー・グループを削除します。このプロシージャを実行してターゲット・マスター・サイトまたはマスター・マテリアライズド・ビュー・サイトに接続すると、マスター・サイトまたはマスター・マテリアライズド・ビュー・サイトにあるターゲット・マテリアライズド・ビュー・グループのメタデータが削除されます。接続できない場合は、8-11 ページの「[マスター・サイトまたはマスター・マテリアライズド・ビュー・サイトのクリーン・アップ](#)」を参照してください。

これらのアクションを実行するには、次の要件を満たす必要があります。

**実行者:** マテリアライズド・ビュー管理者

**実行場所:** リモート・マテリアライズド・ビュー・サイト

**レプリケーションの状態:** 該当せず

次に示す手順に従って、マテリアライズド・ビュー・サイトにあるマテリアライズド・ビュー・グループを削除します。

**手順 1 マテリアライズド・ビュー管理者としてマテリアライズド・ビュー・サイトに接続します。**

```
CONNECT mviewadmin/mviewadmin@mv1.world
```

**手順 2 マテリアライズド・ビュー・グループを削除します。**

```
BEGIN
    DBMS_REPCAT.DROP_MVIEW_REPGROUP (
        gname => 'hr_repg',
        drop_contents => TRUE);
END;
/
```

マテリアライズド・ビュー・グループの内容をマテリアライズド・ビュー・データベースから物理的に削除する場合は、`drop_contents` パラメータを `TRUE` に指定してください。

### マテリアライズド・ビュー・サイトのオブジェクトの削除

個別のマテリアライズド・ビューをマテリアライズド・ビュー・サイトから削除することが必要になった場合は、`DROP_MVIEW_REPOBJECT` プロシージャ API を使用してマテリアライズド・ビューを削除します。このプロシージャを実行してターゲット・マスター・サイトまたはマスター・マテリアライズド・ビュー・サイトに接続すると、マスター・サイトまたはマスター・マテリアライズド・ビュー・サイトにあるターゲット・マテリアライズド・ビューのメタデータが削除されます。接続できない場合は、8-11 ページの「[マスター・サイトまたはマスター・マテリアライズド・ビュー・サイトのクリーン・アップ](#)」を参照してください。

これらのアクションを実行するには、次の要件を満たす必要があります。

**実行者:** マテリアライズド・ビュー管理者

**実行場所:** リモート・マテリアライズド・ビュー・サイト

**レプリケーションの状態:** 該当せず



次に示す手順に従って、マテリアライズド・ビュー・サイトにある個別のマテリアライズド・ビューを削除します。

#### 手順 1 マテリアライズド・ビュー管理者としてマテリアライズド・ビュー・サイトに接続します。

```
CONNECT mvviewadmin/mvviewadmin@mv1.world
```

#### 手順 2 マテリアライズド・ビューを削除します。

```
BEGIN
    DBMS_REPCAT.DROP_MVIEW_REPOBJECT (
        sname => 'hr',
        oname => 'employees_mv1',
        type => 'SNAPSHOT',
        drop_objects => TRUE);
END;
/
```

マテリアライズド・ビューの内容をマテリアライズド・ビュー・データベースから物理的に削除する場合は、`drop_contents` パラメータを `TRUE` に指定してください。

## マスター・サイトまたはマスター・マテリアライズド・ビュー・サイトのクリーン・アップ

ターゲット・マスター・サイトまたはターゲット・マテリアライズド・ビュー・サイトに接続中に、マテリアライズド・ビュー・グループまたはマテリアライズド・ビュー・オブジェクトを削除できない場合は、マスター・サイトまたはマスター・マテリアライズド・ビュー・サイトにある関連メタデータを手動で削除する必要があります。メタデータをクリーン・アップすると、マスター表またはマスター・マテリアライズド・ビューに加えられた変更をマテリアライズド・ビュー・ログに保持する必要がなくなります。次の項では、マテリアライズド・ビュー・グループまたはオブジェクトを削除した後、マスター・サイトまたはマテリアライズド・ビュー・サイトをクリーン・アップする方法を説明します。

### マテリアライズド・ビュー・グループの削除後のクリーン・アップ

8-9 ページの「[手動で作成されたマテリアライズド・ビュー・グループの削除](#)」に示す手順を実行してもマスター・サイトまたはマスター・マテリアライズド・ビュー・サイトに接続されない場合は、次に示す手順を実行して、ターゲット・マスター・サイトまたはマスター・マテリアライズド・ビュー・サイトをクリーン・アップすることをお勧めします。

これらのアクションを実行するには、次の要件を満たす必要があります。

**実行者:** レプリケーション管理者

**実行場所:** ターゲット・マテリアライズド・ビュー・サイト用のマスター・サイトまたはマスター・マテリアライズド・ビュー・サイト

**レプリケーションの状態:** 通常

マテリアライズド・ビュー・グループを削除した後、次に示す手順を実行して、マスター・サイトまたはマスター・マテリアライズド・ビュー・サイトをクリーン・アップします。

---

---

**注意:** このドキュメントをオンラインで参照している場合は、次の「BEGINNING OF SCRIPT」の行から 8-14 ページの「END OF SCRIPT」の行までのテキストをテキスト・エディタにコピーして編集し、使用している環境に適したスクリプトを作成します。

---

---

```
/***** BEGINNING OF SCRIPT *****/
```

**手順 1 レプリケーション管理者としてマスター・サイトまたはマスター・マテリアライズド・ビュー・サイトに接続します。**

```
*/
```

```
SET ECHO ON
```

```
SPOOL cleanup_master1.out
```

```
CONNECT repadmin/repadmin@orc1.world
```

```
/*
```

**手順 2 マテリアライズド・ビュー・グループを登録解除します。**

```
*/
```

```
BEGIN
```

```
  DBMS_REPCAT.UNREGISTER_MVIEW_REPGROUP (  
    gname => 'hr_repg',  
    mviewsite => 'mv1.world');
```

```
END;
```

```
/
```

```
/*
```

**手順 3** ターゲット・マテリアライズド・ビューとしてマークされていたエントリのマテリアライズド・ビュー・ログをパージします。手順 2 で登録解除したマテリアライズド・ビュー・グループにあった各マテリアライズド・ビューに対し、PURGE\_MVIEW\_FROM\_LOG プロシージャを実行します。

---

**注意：** なんらかの理由でマテリアライズド・ビュー・グループの登録解除が失敗した場合も、この手順は実行する必要があります。

---

```
*/

BEGIN
    DBMS_MVIEW.PURGE_MVIEW_FROM_LOG (
        mviewowner => 'hr',
        mviewname => 'countries_mv1',
        mviewsite => 'mv1.world');
END;
/

BEGIN
    DBMS_MVIEW.PURGE_MVIEW_FROM_LOG (
        mviewowner => 'hr',
        mviewname => 'departments_mv1',
        mviewsite => 'mv1.world');
END;
/

BEGIN
    DBMS_MVIEW.PURGE_MVIEW_FROM_LOG (
        mviewowner => 'hr',
        mviewname => 'employees_mv1',
        mviewsite => 'mv1.world');
END;
/

BEGIN
    DBMS_MVIEW.PURGE_MVIEW_FROM_LOG (
        mviewowner => 'hr',
        mviewname => 'jobs_mv1',
        mviewsite => 'mv1.world');
END;
/
```

```
BEGIN
    DBMS_MVIEW.PURGE_MVIEW_FROM_LOG (
        mviewowner => 'hr',
        mviewname => 'job_history_mv1',
        mviewsite => 'mv1.world');
END;
/

BEGIN
    DBMS_MVIEW.PURGE_MVIEW_FROM_LOG (
        mviewowner => 'hr',
        mviewname => 'locations_mv1',
        mviewsite => 'mv1.world');
END;
/

BEGIN
    DBMS_MVIEW.PURGE_MVIEW_FROM_LOG (
        mviewowner => 'hr',
        mviewname => 'regions_mv1',
        mviewsite => 'mv1.world');
END;
/

SET ECHO OFF

SPOOL OFF

/***** END OF SCRIPT *****/
```

### 個別のマテリアライズド・ビュー・サポートのクリーン・アップ

8-10 ページの「[マテリアライズド・ビュー・サイトのオブジェクトの削除](#)」に示す手順を実行してもマスター・サイトまたはマスター・マテリアライズド・ビュー・サイトに接続されない場合は、次に示す手順を実行して、ターゲット・マスター・サイトまたはマスター・マテリアライズド・ビュー・サイトをクリーン・アップすることをお勧めします。

これらのアクションを実行するには、次の要件を満たす必要があります。

**実行者:** レプリケーション管理者

**実行場所:** ターゲット・マテリアライズド・ビュー・サイト用のマスター・サイトまたはマスター・マテリアライズド・ビュー・サイト

**レプリケーションの状態:** 通常

個別のマテリアライズド・ビューを削除した後、次に示す手順を実行して、マスター・サイトまたはマスター・マテリアライズド・ビュー・サイトをクリーン・アップします。

---

---

**注意：** このドキュメントをオンラインで参照している場合は、次の「BEGINNING OF SCRIPT」の行から 8-16 ページの「END OF SCRIPT」の行までのテキストをテキスト・エディタにコピーして編集し、使用している環境に適したスクリプトを作成します。

---

---

```
/****** BEGINNING OF SCRIPT *****
```

**手順 1 レプリケーション管理者としてマスター・サイトまたはマスター・マテリアライズド・ビュー・サイトに接続します。**

```
*/
```

```
SET ECHO ON
```

```
SPOOL cleanup_master2.out
```

```
CONNECT repadmin/repadmin@orc1.world
```

```
/*
```

**手順 2 マテリアライズド・ビューを登録解除します。**

```
*/
```

```
BEGIN
```

```
  DBMS_MVIEW.UNREGISTER_MVIEW (  
    mviewowner => 'hr',  
    mviewname => 'employees_mv1',  
    mviewsite => 'mv1.world');
```

```
END;
```

```
/
```

```
/*
```

**手順 3 ターゲット・マテリアライズド・ビューとしてマークされていたエントリの、関連するマテリアライズド・ビュー・ログをページします。**

---

---

**注意：** なんらかの理由でマテリアライズド・ビューの登録解除が失敗した場合も、この手順は実行する必要があります。

---

---

```
*/

BEGIN
    DBMS_MVIEW.PURGE_MVIEW_FROM_LOG (
        mviewowner => 'hr',
        mviewname => 'employees_mv1',
        mviewsite => 'mv1.world');
END;
/

SET ECHO OFF

SPOOL OFF

/***** END OF SCRIPT *****/
```

## マテリアライズド・ビュー・ログの管理

次の項では、マテリアライズド・ビュー・ログの管理方法を説明します。

- [マテリアライズド・ビュー・ログの変更](#)
- [マテリアライズド・ビュー・ログ領域の管理](#)
- [マテリアライズド・ビュー・ログが含まれるマスター表の再編成](#)
- [マテリアライズド・ビュー・ログの削除](#)

## マテリアライズド・ビュー・ログの変更

マテリアライズド・ビュー・ログを作成すると、記憶領域パラメータおよび対応するマテリアライズド・ビューに対するサポートを変更できます。次の項では、マテリアライズド・ビュー・ログの変更方法を説明します。次のユーザーのみが、マテリアライズド・ビュー・ログを変更できます。

- マスター表またはマスター・マテリアライズド・ビューの所有者
- マスター表またはマスター・マテリアライズド・ビューの `SELECT` 権限、および `MLOG$_master_name` の `ALTER` 権限を持つユーザー。*master\_name* とは、マテリアライズド・ビュー・ログ用マスターの名前です。たとえば、マスター表が `employees` の場合は、マテリアライズド・ビュー・ログ表名は `MLOG$_employees` です。

## マテリアライズド・ビュー・ログ記憶域パラメータの変更

マテリアライズド・ビュー・ログの記憶域パラメータを変更するには、ALTER MATERIALIZED VIEW LOG 文を使用します。たとえば、次の文では、hr スキーマ内の employees 表のマテリアライズド・ビュー・ログが変更されます。

```
ALTER MATERIALIZED VIEW LOG ON hr.employees
  PCTFREE 25
  PCTUSED 40;
```

## 列の追加のためのマテリアライズド・ビュー・ログの変更

マテリアライズド・ビュー・ログに新しい列を追加するには、SQL 文 ALTER MATERIALIZED VIEW LOG を使用します。たとえば、次の文では、sales スキーマ内の customers 表のマテリアライズド・ビュー・ログが変更されます。

```
ALTER MATERIALIZED VIEW LOG ON hr.employees
  ADD (department_id);
```

**関連項目：** マテリアライズド・ビュー・ログへの列の追加の詳細は、『Oracle9i アドバンスド・レプリケーション』を参照してください。

## マテリアライズド・ビュー・ログ領域の管理

マテリアライズド・ビューのリフレッシュ時にマテリアライズド・ビュー・ログ内のどの行が使用されたかが自動的に追跡され、ログが際限なく増大しないよう、不要な行がログからページされます。複数の単純マテリアライズド・ビューでは同じマテリアライズド・ビュー・ログを使用できるため、あるマテリアライズド・ビューのリフレッシュにすでに使用された行が、まだ別のマテリアライズド・ビューのリフレッシュに必要な場合があります。このため、すべてのマテリアライズド・ビューで使用されるまで行はログから削除されません。

たとえば、マスター・サイトの customers 表に対して 2 つのマテリアライズド・ビューが作成されたとします。customers マテリアライズド・ビューは、spdb1 データベースでリフレッシュされます。この場合、マスター表および関連するマテリアライズド・ビュー・ログを管理するサーバーでは、このマテリアライズド・ビューのリフレッシュ時に使用されたマテリアライズド・ビュー・ログの行は、spdb2 データベースの customers マテリアライズド・ビューがこの行を使用してリフレッシュされるまで、ページされません。

マテリアライズド・ビュー・ログから行をページする前に、すべての依存マテリアライズド・ビューがリフレッシュされるのを待つ必要があるため、複数のマテリアライズド・ビューが同じマスター表またはマスター・マテリアライズド・ビューをベースにしている場合、マテリアライズド・ビュー・ログが際限なく増大する可能性があります。このような状況は、たとえば 1 つのマスター表またはマスター・マテリアライズド・ビューが複数のマテリアライズド・ビューのベースとなっており、かつ次の条件のいずれかに該当する場合に発生することがあります。

- あるマテリアライズド・ビューが自動リフレッシュ用に構成されておらず、手動でのリフレッシュが長期間実行されていない場合
- あるマテリアライズド・ビューのリフレッシュ間隔が非常に長い場合（たとえば1年ごと）
- ネットワーク障害によって、マスター表またはマスター・マテリアライズド・ビューをベースとする1つ以上のマテリアライズド・ビューの自動リフレッシュが実行できない場合
- ネットワークまたはサイト障害によって、マテリアライズド・ビューが削除されたことをマスター表またはマスター・マテリアライズド・ビューが認識できない場合

---

**注意：** マテリアライズド・ビューで削除された変更がリフレッシュされる前にマテリアライズド・ビュー・ログのページまたは TRUNCATE を行った場合は、マテリアライズド・ビューの完全リフレッシュを実行する必要があります。

---

### マテリアライズド・ビュー・ログからの行のページ

マテリアライズド・ビュー・ログが使用するデータベース領域を最小にするため、マテリアライズド・ビュー・ログは常にできるかぎり小さくなるようにしてください。マテリアライズド・ビュー・ログから行を削除して、新しいログ・レコード用に領域を解放するには、次のアクションのいずれかを実行します。

- ログに対応付けられたマテリアライズド・ビューをリフレッシュします。これによって、マテリアライズド・ビュー・ログから行がページされます。
- 最も古い  $n$  回目のリフレッシュが行われたマテリアライズド・ビューのみに必要な行を削除することにより、ログのレコードを手動でページします。

マテリアライズド・ビュー・ログから行を手動でページするには、そのログを含むデータベースで、DBMS\_MVIEW パッケージの PURGE\_LOG プロシージャを実行します。たとえば、最も前にリフレッシュされたマテリアライズド・ビューのみに必要なエントリを customers 表のマテリアライズド・ビュー・ログからページするには、次のプロシージャを実行します。

```
BEGIN
    DBMS_MVIEW.PURGE_LOG (
        master => 'hr.employees',
        num    => 1,
        flag   => 'DELETE');
END;
/
```

マテリアライズド・ビュー・ログの所有者または DBMS\_MVIEW パッケージの EXECUTE 権限を持つユーザーのみが、PURGE\_LOG プロシージャを実行してマテリアライズド・ビュー・ログから行をページできます。



## マテリアライズド・ビュー・ログの切捨て

マテリアライズド・ビュー・ログが増大し、多数のエクステンツが割り当てられた場合、その行のログをパージしても、ログに割り当てられた領域の量は減少しません。このような場合は、マテリアライズド・ビュー・ログを切り捨てる必要があります。マテリアライズド・ビュー・ログの所有者または DELETE ANY TABLE システム権限を持つユーザーのみが、マテリアライズド・ビュー・ログを切り捨てることができます。

マテリアライズド・ビュー・ログを切り捨てることによりマテリアライズド・ビュー・ログに割り当てられた領域を減少させるには、次に示す手順を実行します。

**手順 1** マスター表またはマスター・マテリアライズド・ビューに対する排他ロックを取得して、後続のプロセス中に更新が発生しないようにします。たとえば、次のような文を実行します。

```
LOCK TABLE hr.employees IN EXCLUSIVE MODE;
```

**手順 2** 2 番目のデータベース・セッションで、マテリアライズド・ビュー・ログ (MLOG\$ 実表) の行を一時表にコピーします。たとえば、次のような文を実行します。

```
CREATE TABLE hr.templlog AS SELECT * FROM hr.MLOG$_employees;
```

**手順 3** 2 番目のセッションで、SQL 文 TRUNCATE を使用してログを切り捨てます。たとえば、次のような文を実行します。

```
TRUNCATE hr.MLOG$_employees;
```

**手順 4** 2 番目のセッションで、元の行を再挿入します。この作業によって、依存マテリアライズド・ビューの完全リフレッシュを実行する手間が省けます。たとえば、次のような文を実行します。

```
INSERT INTO hr.MLOG$_employees SELECT * FROM hr.templlog;
```

```
DROP TABLE hr.templlog;
```

**手順 5** 最初のセッションで、ロールバックを実行して、マスター表またはマスター・マテリアライズド・ビューの排他ロックを解除します。

```
ROLLBACK;
```

---

**注意：** 新しい位置に行をコピーしてからログを切り捨てるまでの間に、マスター表またはマスター・マテリアライズド・ビューに対して行われた変更は、完全リフレッシュを実行するまで表示されません。

---

## マテリアライズド・ビュー・ログが含まれるマスター表の再編成

パフォーマンスを改善し、ディスクの使用率を最適化するために、マスター表を定期的に再編成できます。この項では、マスターの再編成方法および関連するマテリアライズド・ビューの高速リフレッシュ機能の維持方法を説明します。

---

---

**注意：** これらの項では表のオンライン再定義は説明しません。マテリアライズド・ビュー・ログ、マスター・マテリアライズド・ビューまたはマテリアライズド・ビューが含まれるマスター表のオンライン再定義はできません。オンライン再定義は、マテリアライズド・ビュー・ログを含まないマスター表でのみ可能です。表のオンライン再定義の詳細は、『Oracle9i データベース管理者ガイド』を参照してください。

---

---

### 再編成通知

表を再編成するときに、マテリアライズド・ビュー・ログの ROWID 情報を無効にする必要があります。表の再編成は、再編成によって表が切り捨てられた場合のみ自動的に検出されます。

表が切り捨てられない場合は、表の再編成を通知する必要があります。表の再編成をサポートするため、DBMS\_MVIEW パッケージの BEGIN\_TABLE\_REORGANIZATION プロシージャと END\_TABLE\_REORGANIZATION プロシージャによって、表の再編成が通知されます。これらのプロシージャは、クリーン・アップ操作を実行し、高速リフレッシュ・メカニズムが必要とするログの整合性とトリガーを検証し、表のマテリアライズド・ビュー・ログ内の ROWID 情報を無効にします。入力値は、再編成されるマスターの名前と所有者です。出力はありません。

**関連項目：** 8-22 ページ「[表 employees の再編成の方法 2](#)」

### マスターの切捨て

表が切り捨てられると、そのマテリアライズド・ビュー・ログも切り捨てられます。ただし、主キー・マテリアライズド・ビューの場合、マテリアライズド・ビュー・ログを切り捨てずに、高速リフレッシュを継続できます。マテリアライズド・ビュー・ログに格納されている情報は保持されますが、マスターが切り捨てられるとマテリアライズド・ビュー・ログは ROWID に関して無効になります。マテリアライズド・ビュー・ログの ROWID 情報は新たに作成されたことになり、ROWID マテリアライズド・ビューで高速リフレッシュ用に使えません。

PRESERVE MATERIALIZED VIEW LOG オプションはデフォルトです。このため、PRESERVE MATERIALIZED VIEW LOG オプションを指定した場合はオプションを指定しなかった場合は、マスターのマテリアライズド・ビュー・ログ内の情報は保持されますが、現行の ROWID マテリアライズド・ビューでログを使用して高速リフレッシュを行うには、完全リフレッシュを実行する必要があります。

---

**注意：** 以前に高速リフレッシュが可能だったマテリアライズド・ビューのリフレッシュを有効にするには、8-21 ページの「データベース表の再編成方法」のガイドラインに従ってください。

---

PURGE MATERIALIZED VIEW LOG オプションが指定された場合は、マテリアライズド・ビュー・ログがマスターとともにパージされます。

**例** 次の2つの文のいずれも、orders という名前のマスター表が切り捨てられた場合にマテリアライズド・ビュー・ログ情報を保持します。

```
TRUNCATE TABLE hr.employees PRESERVE MATERIALIZED VIEW LOG;  
TRUNCATE TABLE hr.employees;
```

次の文は、マスター表とともにマテリアライズド・ビュー・ログを切り捨てます。

```
TRUNCATE TABLE hr.employees PURGE MATERIALIZED VIEW LOG;
```

## データベース表の再編成方法

高速リフレッシュ機能を保持したまま表を再編成する方法が4つ提供されています。これらの方法については、次の項で説明します。他の再編成方法では、後続の高速リフレッシュを使用可能にするために、最初に完全リフレッシュを実行する必要があります。

---

**注意：** マスターの再編成中はダイレクト・ローダーを使用しないでください。ダイレクト・ローダーは列の再順序付けの原因となり、副問合せおよびLOB マテリアライズド・ビューで使用されるログ情報を無効にすることがあります。

---

**表 employees の再編成の方法 1** 次の手順に従います。

**手順 1** 表 `employees` に対して `DBMS_MVIEW.BEGIN_TABLE_REORGANIZATION` をコールします。

**手順 2** 表名 `employees` を `employees_old` に変更します。

**手順 3** 表 `employees` を `SELECT * FROM employees_old` として作成します。

**手順 4** 新しい表 `employees` に対して `DBMS_MVIEW.END_TABLE_REORGANIZATION` をコールします。

---

**注意：** 表名を変更すると、対応付けられた PL/SQL のトリガーも表の新しい名前に変更されます。

---

`BEGIN_TABLE_REORGANIZATION` のコールと `END_TABLE_REORGANIZATION` のコールの間に、再編成された表に対してトランザクションが発行されないようにしてください。

**表 employees の再編成の方法 2** 次の手順に従います。

**手順 1** 表 `employees` に対して `DBMS_MVIEW.BEGIN_TABLE_REORGANIZATION` をコールします。

**手順 2** 表 `employees` をエクスポートします。

**手順 3** 表 `employees` を `PRESERVE MATERIALIZED VIEW LOG` オプションで切り捨てます。

**手順 4** 従来型パスを使用して表 `employees` をインポートします。

**手順 5** 新しい表 `employees` に対して `DBMS_MVIEW.END_TABLE_REORGANIZATION` をコールします。

---

**注意：** 再編成中にマスターを切り捨てるときは、切り捨てる表の DDL の `PRESERVE MATERIALIZED VIEW LOG` 句を使用する必要があります。

---

BEGIN\_TABLE\_REORGANIZATION のコールと END\_TABLE\_REORGANIZATION のコールの間に、再編成された表に対してトランザクションが発行されないようにしてください。

**表 employees の再編成の方法 3** 次の手順に従います。

**手順 1** 表 employees に対して DBMS\_MVIEW.BEGIN\_TABLE\_REORGANIZATION をコールします。

**手順 2** 表 employees をエクスポートします。

**手順 3** 表名 employees を employees\_old に変更します。

**手順 4** 従来型パスを使用して表 employees をインポートします。

**手順 5** 新しい表 employees に対して DBMS\_MVIEW.END\_TABLE\_REORGANIZATION をコールします。

---

**注意：** 表名を変更すると、対応付けられた PL/SQL のトリガーも表の新しい名前に変更されます。

---

BEGIN\_TABLE\_REORGANIZATION のコールと END\_TABLE\_REORGANIZATION のコールの間に、再編成された表に対してトランザクションが発行されないようにしてください。

**表 employees の再編成の方法 4** 次の手順に従います。

**手順 1** 表 employees に対して DBMS\_MVIEW.BEGIN\_TABLE\_REORGANIZATION をコールします。

**手順 2** 表 employees の内容を選択してフラット・ファイルに格納します。

**手順 3** 表名 employees を employees\_old に変更します。

**手順 4** 表 employees\_old と同じ shape で表 employees を作成します。

**手順 5** 従来型パスを使用して SQL\*Loader を実行します。

**手順 6** 新しい表 employees に対して DBMS\_MVIEW.END\_TABLE\_REORGANIZATION をコールします。

---

**注意：** 表名を変更すると、対応付けられた PL/SQL のトリガーも表の新しい名前に変更されます。

---

BEGIN `TABLE_REORGANIZATION` のコールと END `TABLE_REORGANIZATION` のコールの間に、再編成された表に対してトランザクションが発行されないようにしてください。

**関連項目：**

- 15-3 ページ [「BEGIN\\_TABLE\\_REORGANIZATION プロシージャ」](#)
- 15-4 ページ [「END\\_TABLE\\_REORGANIZATION プロシージャ」](#)

## マテリアライズド・ビュー・ログの削除

マテリアライズド・ビュー・ログは、マスターや既存のマテリアライズド・ビューに関係なく削除できます。たとえば、次の条件のいずれかに該当する場合は、マテリアライズド・ビュー・ログを削除できます。

- マスターのすべてのマテリアライズド・ビューが削除された場合
- マスターのすべてのマテリアライズド・ビューで、高速リフレッシュではなく、完全リフレッシュを使用する場合
- マスターが、高速リフレッシュを必要とするマテリアライズド・ビューをサポートしない場合

ここでマスターとは、マスター表またはマスター・マテリアライズド・ビューです。マテリアライズド・ビュー・ログを削除するには、**SQL\*Plus** で `DROP MATERIALIZED VIEW LOG` 文を実行します。たとえば、次の文では、`sales` スキーマ内の `customers` という名前の表のマテリアライズド・ビュー・ログが削除されます。

```
DROP MATERIALIZED VIEW LOG ON hr.employees;
```

マスターの所有者または `DROP ANY TABLE` システム権限を持つユーザーのみが、マテリアライズド・ビュー・ログを削除できます。

## エクスポート/インポートによるマテリアライズド・ビュー・サイトのオフライン・インスタンス化

確立されたレプリケーション環境が拡張されると、新しいマテリアライズド・ビュー・サイトをレプリケーション環境に追加した場合に、ネットワークの通信量が増大します。これは、ネットワークを通じて、表またはマテリアライズド・ビューの内容全体が新しいレプリケート・サイトに伝播されるためです。

このようなネットワークの通信量を最少にするには、レプリケーション環境を拡張する場合にオフライン・インスタンス化・プロセスを使用します。オフライン・インスタンス化では、Oracle のエクスポート・ユーティリティおよびインポート・ユーティリティを利用できます。これによって、エクスポート・ファイルを作成し、CD-ROM やテープなどの記憶メディアを使用して新しいサイトにデータを転送できます。

マスター・サイトのオフライン・インスタンス化を行うのと同じ理由から、オフライン・インスタンス化・プロセスを使用して、新しいマテリアライズド・ビュー・グループをマテリアライズド・ビュー・サイトに作成する場合があります。マテリアライズド・ビュー・サイトではモデム接続を使用したラップトップがターゲット・コンピュータである場合が多いため、場合によってはこの方が効果的です。

次のスクリプトでは、新しいマテリアライズド・ビュー・グループに対するオフライン・インスタンス化が実行されます。これらのアクションを実行するには、次の要件を満たす必要があります。

### 実行者:

- マスター・サイトのレプリケーション管理者
- 新しいマテリアライズド・ビュー・サイトのマテリアライズド・ビュー管理者

### 実行場所:

- ターゲット・マテリアライズド・ビュー・サイト用マスター・サイト
- 新しいマテリアライズド・ビュー・サイト

### レプリケーションの状態: 通常

次に示す手順に従って、エクスポートまたはインポートを使用したマテリアライズド・ビュー・サイトのオフライン・インスタンス化を実行します。

---

**注意:** このドキュメントをオンラインで参照している場合は、次の「BEGINNING OF SCRIPT」の行から 8-35 ページの「END OF SCRIPT」の行までのテキストをテキスト・エディタにコピーして編集し、使用している環境に適したスクリプトを作成します。

---

```
/***** BEGINNING OF SCRIPT *****/
```

**手順 1 独立した端末ウィンドウで、新しいマテリアライズド・ビュー・サイトを設定します。**

マテリアライズド・ビューのオフライン・インスタンス化を実行する前に、適切なスキーマおよびデータベース・リンクが作成されていることを確認してください。

**関連項目：** 2-20 ページ [「マテリアライズド・ビュー・サイトの設定」](#)

```
*/
```

```
SET ECHO ON
```

```
SPOOL offline_instant_mv.out
```

```
PAUSE Press <RETURN> to continue the new materialized view site is set up.
```

```
/*
```

**手順 2 レプリケーション管理者としてマスター・サイトに接続します。**

```
*/
```

```
CONNECT repadmin/repadmin@orc1.world
```

```
/*
```

**手順 3 必要なマテリアライズド・ビュー・ログを作成します。**

ターゲット・マスター表に対するマテリアライズド・ビュー・ログがまだ存在していない場合は、ターゲット・マスター・サイトで作成します。

```
*/
```

```
CREATE MATERIALIZED VIEW LOG ON hr.countries;  
CREATE MATERIALIZED VIEW LOG ON hr.departments;  
CREATE MATERIALIZED VIEW LOG ON hr.employees;  
CREATE MATERIALIZED VIEW LOG ON hr.jobs;  
CREATE MATERIALIZED VIEW LOG ON hr.job_history;  
CREATE MATERIALIZED VIEW LOG ON hr.locations;  
CREATE MATERIALIZED VIEW LOG ON hr.regions;
```

```
/*
```



**手順 4 マスター・サイトで一時マテリアライズド・ビューを作成します。このマテリアライズド・ビューには、エクスポート・ファイルを使用して新しいマテリアライズド・ビュー・サイトに転送するデータが含まれます。**

---

---

**注意：** 第 6 章「競合解消の構成」に示す競合解消のいずれかを追加した場合は、表に列を追加できます。この追加の列は、次に示す SELECT 文に必ず含めてください。更新可能なマテリアライズド・ビューでは、マスター表のすべての列を明示的に選択する必要があります。このため、SELECT \* 文は使用しないでください。

---

---

\*/

```
CREATE MATERIALIZED VIEW hr.countries_mv
REFRESH FAST WITH PRIMARY KEY FOR UPDATE AS SELECT
country_id, country_name, region_id
FROM hr.countries;

CREATE MATERIALIZED VIEW hr.departments_mv
REFRESH FAST WITH PRIMARY KEY FOR UPDATE AS SELECT
department_id, department_name, manager_id, location_id
FROM hr.departments;

CREATE MATERIALIZED VIEW hr.employees_mv
REFRESH FAST WITH PRIMARY KEY FOR UPDATE AS SELECT
employee_id, first_name, last_name, email, phone_number,
hire_date, job_id, salary, commission_pct, manager_id,
department_id
FROM hr.employees;

CREATE MATERIALIZED VIEW hr.jobs_mv
REFRESH FAST WITH PRIMARY KEY FOR UPDATE AS SELECT
job_id, job_title, min_salary, max_salary
FROM hr.jobs;

CREATE MATERIALIZED VIEW hr.job_history_mv
REFRESH FAST WITH PRIMARY KEY FOR UPDATE AS SELECT
employee_id, start_date, end_date, job_id, department_id
FROM hr.job_history;

CREATE MATERIALIZED VIEW hr.locations_mv
REFRESH FAST WITH PRIMARY KEY FOR UPDATE AS SELECT
location_id, street_address, postal_code, city,
state_province, country_id
FROM hr.locations;
```

```
CREATE MATERIALIZED VIEW hr.regions_mv
  REFRESH FAST WITH PRIMARY KEY FOR UPDATE AS SELECT
    region_id, region_name
  FROM hr.regions;
```

```
/*
```

**手順 5 独立した端末ウィンドウで、マテリアライズド・ビューの所有者として接続し、手順 4 で作成した一時マテリアライズド・ビューをエクスポートします。**

Oracle Export ユーティリティを使用し、新しいマテリアライズド・ビュー・サイトに転送するエクスポート・ファイルを生成します。エクスポート・ファイルには、一時マテリアライズド・ビューの実表が含まれます。

---

---

**注意：** 次の例は、Oracle8i 以上のデータベースのみに適用されます。  
Oracle8i より前のバージョンのデータベースの実表には、SNAP\$ 接頭辞が  
付加されます (SNAP\$\_employees\_mv など)。

---

---

**関連項目：** エクスポートに関する追加情報は、『Oracle9i データベース・ユーティリティ』を参照してください。

hr スキーマに対するエクスポート・コマンドの例を次に示します。

```
exp hr/hr@orcl.world TABLES='countries_mv','departments_mv','employees_mv',
'jobs_mv','job_history_mv','locations_mv','regions_mv'
```

```
*/
```

PAUSE Press <RETURN> to continue when the export is complete.

```
/*
```

**手順 6 SYSTEM ユーザーとして新しいマテリアライズド・ビュー・サイトに接続します。**

```
*/
```

```
CONNECT system/manager@mview.world
```

```
/*
```

**手順7 マテリアライズド・ビュー・サイトに必要なスキーマおよびデータベース・リンクが存在しない場合は、これらを作成します。**

マテリアライズド・ビューのオフライン・インスタンス化を実行する前に、新しいマテリアライズド・ビュー・サイトのマテリアライズド・ビューを含むスキーマと、マテリアライズド・ビュー・サイトからマスター・サイトへのデータベース・リンクを作成します。マテリアライズド・ビューは、マスター・サイトのマスター・オブジェクトを含むスキーマと同じスキーマに存在する必要があります。

```

*/

CREATE TABLESPACE demo_mview
  DATAFILE 'demo_mview.dbf' SIZE 10M AUTOEXTEND ON
  EXTENT MANAGEMENT LOCAL AUTOALLOCATE;

CREATE TEMPORARY TABLESPACE temp_mview
  TEMPFILE 'temp_mview.dbf' SIZE 5M AUTOEXTEND ON;

CREATE USER hr IDENTIFIED BY hr;

ALTER USER hr DEFAULT TABLESPACE demo_mview
  QUOTA UNLIMITED ON demo_mview;

ALTER USER hr TEMPORARY TABLESPACE temp_mview;

GRANT
  CREATE SESSION,
  CREATE TABLE,
  CREATE PROCEDURE,
  CREATE SEQUENCE,
  CREATE TRIGGER,
  CREATE VIEW,
  CREATE SYNONYM,
  ALTER SESSION,
  CREATE MATERIALIZED VIEW,
  ALTER ANY MATERIALIZED VIEW,
  CREATE DATABASE LINK
TO hr;

CONNECT hr/hr@mview.world

CREATE DATABASE LINK orcl.world CONNECT TO hr IDENTIFIED by hr;

/*

```

**手順 8 マテリアライズド・ビュー管理者として新しいマテリアライズド・ビュー・サイトに接続します。**

\*/

```
CONNECT mviewadmin/mviewadmin@mview.world
```

/\*

**手順 9 空のマテリアライズド・ビュー・グループを作成します。**

新しいマテリアライズド・ビュー・サイトで DBMS\_REPCAT.CREATE\_MVIEW\_REPGROUP プロシージャを実行し、マテリアライズド・ビューを追加するための空のマテリアライズド・ビュー・グループを作成します。

\*/

```
BEGIN
```

```
    DBMS_REPCAT.CREATE_MVIEW_REPGROUP (
        gname => 'hr_repg',
        master => 'orcl.world',
        propagation_mode => 'ASYNCHRONOUS');
```

```
END;
```

```
/
```

/\*

**手順 10 オフライン・インスタンス化用のマテリアライズド・ビュー・サイトを準備します。**

DBMS\_OFFLINE\_SNAPSHOT.BEGIN\_LOAD プロシージャは、新しいマテリアライズド・ビューに必要なサポート・メカニズムを作成します。またこの手順では、前の手順で作成したマテリアライズド・ビュー・グループに新しいマテリアライズド・ビューが追加されます。インポートする各マテリアライズド・ビューに、BEGIN\_LOAD プロシージャを必ず実行してください。

\*/

```
BEGIN
```

```
    DBMS_OFFLINE_SNAPSHOT.BEGIN_LOAD (
        gname => 'hr_repg',
        sname => 'hr',
        master_site => 'orcl.world',
        snapshot_oname => 'countries_mv');
```

```
END;
```

```
/
```

```
BEGIN
    DBMS_OFFLINE_SNAPSHOT.BEGIN_LOAD (
        gname => 'hr_repg',
        sname => 'hr',
        master_site => 'orcl.world',
        snapshot_onsame => 'departments_mv');
END;
/

BEGIN
    DBMS_OFFLINE_SNAPSHOT.BEGIN_LOAD (
        gname => 'hr_repg',
        sname => 'hr',
        master_site => 'orcl.world',
        snapshot_onsame => 'employees_mv');
END;
/

BEGIN
    DBMS_OFFLINE_SNAPSHOT.BEGIN_LOAD (
        gname => 'hr_repg',
        sname => 'hr',
        master_site => 'orcl.world',
        snapshot_onsame => 'jobs_mv');
END;
/

BEGIN
    DBMS_OFFLINE_SNAPSHOT.BEGIN_LOAD (
        gname => 'hr_repg',
        sname => 'hr',
        master_site => 'orcl.world',
        snapshot_onsame => 'job_history_mv');
END;
/

BEGIN
    DBMS_OFFLINE_SNAPSHOT.BEGIN_LOAD (
        gname => 'hr_repg',
        sname => 'hr',
        master_site => 'orcl.world',
        snapshot_onsame => 'locations_mv');
END;
/
```

```
BEGIN
    DBMS_OFFLINE_SNAPSHOT.BEGIN_LOAD (
        gname => 'hr_repg',
        sname => 'hr',
        master_site => 'orcl.world',
        snapshot_onsame => 'regions_mv');
END;
/

/*
```

**手順 11 独立した端末ウィンドウで、マテリアライズド・ビューの所有者として接続し、新しいマテリアライズド・ビュー・サイトでインポートします。**

Oracle Import ユーティリティを使用して、手順 5 でエクスポートしたファイルをインポートします。必ず、データをエクスポートしたユーザーと同一のユーザーとしてデータをインポートします。次の例の場合、このユーザーは hr です。

```
*/

imp hr/hr@mview.world FULL=y IGNORE=y

PAUSE Press <RETURN> to continue when the import is complete.

/*
```

**手順 12 オフライン・インスタンス化を実行します。**

DBMS\_OFFLINE\_SNAPSHOT.END\_LOAD プロシージャを実行し、インポートされたマテリアライズド・ビューのオフライン・インスタンス化を完了します。

```
*/

BEGIN
    DBMS_OFFLINE_SNAPSHOT.END_LOAD (
        gname => 'hr_repg',
        sname => 'hr',
        snapshot_onsame => 'countries_mv');
END;
/

BEGIN
    DBMS_OFFLINE_SNAPSHOT.END_LOAD (
        gname => 'hr_repg',
        sname => 'hr',
        snapshot_onsame => 'departments_mv');
END;
/
```

```
BEGIN
    DBMS_OFFLINE_SNAPSHOT.END_LOAD (
        gname => 'hr_repg',
        sname => 'hr',
        snapshot_onsame => 'employees_mv');
END;
/

BEGIN
    DBMS_OFFLINE_SNAPSHOT.END_LOAD (
        gname => 'hr_repg',
        sname => 'hr',
        snapshot_onsame => 'jobs_mv');
END;
/

BEGIN
    DBMS_OFFLINE_SNAPSHOT.END_LOAD (
        gname => 'hr_repg',
        sname => 'hr',
        snapshot_onsame => 'job_history_mv');
END;
/

BEGIN
    DBMS_OFFLINE_SNAPSHOT.END_LOAD (
        gname => 'hr_repg',
        sname => 'hr',
        snapshot_onsame => 'locations_mv');
END;
/

BEGIN
    DBMS_OFFLINE_SNAPSHOT.END_LOAD (
        gname => 'hr_repg',
        sname => 'hr',
        snapshot_onsame => 'regions_mv');
END;
/

/*
```

**手順 13 マテリアライズド・ビュー・サイトでマテリアライズド・ビューの所有者として接続します。**

\*/

```
CONNECT hr/hr@mview.world
```

/\*

**手順 14 マテリアライズド・ビューをリフレッシュして、マスター・サイトで登録します。**

新しいマテリアライズド・ビュー・サイトにあるマテリアライズド・ビューをリフレッシュすると、マスター表からの最新の変更が取得される以外に、ターゲット・マスター・サイトにあるオフライン・インスタンスエーションを行ったマテリアライズド・ビューが登録されます。

\*/

```
BEGIN
    DBMS_MVIEW.REFRESH ('countries_mv');
END;
/
```

```
BEGIN
    DBMS_MVIEW.REFRESH ('departments_mv');
END;
/
```

```
BEGIN
    DBMS_MVIEW.REFRESH ('employees_mv');
END;
/
```

```
BEGIN
    DBMS_MVIEW.REFRESH ('jobs_mv');
END;
/
```

```
BEGIN
    DBMS_MVIEW.REFRESH ('job_history_mv');
END;
/
```

```
BEGIN
    DBMS_MVIEW.REFRESH ('locations_mv');
END;
/
```



```
BEGIN
    DBMS_MVIEW.REFRESH ('regions_mv');
END;
/
```

```
/*
```

**手順 15 レプリケーション管理者としてマスター・サイトに接続します。**

```
*/
```

```
CONNECT repadmin/repadmin@orc1.world
```

```
/*
```

**手順 16 マスター・サイトにある、手順 4 で作成した一時マテリアライズド・ビューを削除します。**

```
*/
```

```
DROP MATERIALIZED VIEW hr.countries_mv;
DROP MATERIALIZED VIEW hr.departments_mv;
DROP MATERIALIZED VIEW hr.employees_mv;
DROP MATERIALIZED VIEW hr.jobs_mv;
DROP MATERIALIZED VIEW hr.job_history_mv;
DROP MATERIALIZED VIEW hr.locations_mv;
DROP MATERIALIZED VIEW hr.regions_mv;
```

```
SET ECHO OFF
```

```
SPOOL OFF
```

```
/****** END OF SCRIPT *****/
```

## マテリアライズド・ビュー・グループに対するグループ所有者の使用

新しいマテリアライズド・ビュー・グループとその関連オブジェクトを定義する場合は、グループの所有者を指定します。所有者を指定しておくで、同じレプリケーション・グループに基づいて、1つのマテリアライズド・ビュー・サイトに複数のマテリアライズド・ビュー・グループを作成できます。マテリアライズド・ビュー・サイトでは、グループ所有者を指定することにより、マスター・サイトまたはマスター・マテリアライズド・ビュー・サイトの同じレプリケーション・グループをベースとする、複数のマテリアライズド・ビュー・グループを作成できます。複数のグループを作成するには、マテリアライズド・ビュー・サイトで、異なるスキーマに基づくマテリアライズド・ビュー・グループを作成します。

次に示す手順に従って、グループ所有者を使用します。

---

---

**注意：** このドキュメントをオンラインで参照している場合は、次の「BEGINNING OF SCRIPT」の行から 8-42 ページの「END OF SCRIPT」の行までのテキストをテキスト・エディタにコピーして編集し、使用している環境に適したスクリプトを作成します。

---

---

**関連項目：** グループ所有者の使用の詳細、および複数のデータ・セットを使用するメリットの詳細は、『Oracle9i アドバンスド・レプリケーション』を参照してください。

/\*\*\*\*\* BEGINNING OF SCRIPT \*\*\*\*\*/

**手順 1 マテリアライズド・ビュー管理者としてマテリアライズド・ビュー・サイトに接続します。**

\*/

SET ECHO ON

SPOOL mv\_group\_owner.out

CONNECT mviewadmin/mviewadmin@mv1.world

/\*

**手順 2 CREATE\_MVIEW\_REPGROUP プロシーダを使用して、マテリアライズド・ビュー・グループをグループ所有者 (gowner) bob で作成します。**

gname パラメータで指定するレプリケーション・グループは、ターゲット・マスター・サイトまたはマスター・マテリアライズド・ビュー・サイトでレプリケートするレプリケーション・グループの名前と一致する必要があります。gowner パラメータにより、追加の識別子を指定できます。これにより、同じマテリアライズド・ビュー・サイトの同じレプリケーション・グループをベースとする、複数のマテリアライズド・ビュー・グループを作成できます。

この例では、グループ所有者 bob および jane 用にマテリアライズド・ビュー・グループが作成され、どちらのマテリアライズド・ビュー・グループも同じレプリケーション・グループをベースとします。

```

*/

BEGIN
    DBMS_REPCAT.CREATE_MVIEW_REPGROUP (
        gname => 'hr_repg',
        master => 'orcl.world',
        propagation_mode => 'ASYNCHRONOUS',
        gowner => 'bob');
END;
/

BEGIN
    DBMS_REPCAT.CREATE_MVIEW_REPGROUP (
        gname => 'hr_repg',
        master => 'orcl.world',
        propagation_mode => 'ASYNCHRONOUS',
        gowner => 'jane');
END;
/

/*

```

**手順 3 bob が所有するマテリアライズド・ビューを作成します。**

マテリアライズド・ビュー・オブジェクトを作成するときに使用される gowner 値は、前のプロシーダでマテリアライズド・ビュー・グループを作成したときに指定した gowner 値と一致する必要があります。マテリアライズド・ビュー・グループを作成した後、bob と jane が所有する hr\_repg マテリアライズド・ビュー・グループの、同じマスターをベースとするマテリアライズド・ビューを作成できます。

---

**注意：** 各オブジェクトは一意の名前を持つ必要があります。ある `gowner` を使用して複数のマテリアライズド・ビュー・グループを使用するときは、重複するオブジェクト名は問題になる可能性があります。オブジェクトのネーミングの競合を回避するため、次のプロシージャに示すように、作成するオブジェクト名の末尾に `gowner` 値を追加しても構いません（`CREATE MATERIALIZED VIEW hr.countries_bob` など）。このようなネーミング・メソッドにより、競合する名前を持つオブジェクトの作成を回避できます。

---

マテリアライズド・ビューを作成するときは、必ず、マテリアライズド・ビューに対する問合せの中で表の所有者のスキーマ名を指定します。次の例では、各問合せにおける表の所有者として `hr` が指定されています。

```
*/

CREATE MATERIALIZED VIEW hr.countries_bob
  REFRESH FAST WITH PRIMARY KEY FOR UPDATE
  AS SELECT * FROM hr.countries@orc1.world;

CREATE MATERIALIZED VIEW hr.departments_bob
  REFRESH FAST WITH PRIMARY KEY FOR UPDATE
  AS SELECT * FROM hr.departments@orc1.world;

CREATE MATERIALIZED VIEW hr.employees_bob
  REFRESH FAST WITH PRIMARY KEY FOR UPDATE
  AS SELECT * FROM hr.employees@orc1.world;

CREATE MATERIALIZED VIEW hr.jobs_bob
  REFRESH FAST WITH PRIMARY KEY FOR UPDATE
  AS SELECT * FROM hr.jobs@orc1.world;

CREATE MATERIALIZED VIEW hr.job_history_bob
  REFRESH FAST WITH PRIMARY KEY FOR UPDATE
  AS SELECT * FROM hr.job_history@orc1.world;

CREATE MATERIALIZED VIEW hr.locations_bob
  REFRESH FAST WITH PRIMARY KEY FOR UPDATE
  AS SELECT * FROM hr.locations@orc1.world;

CREATE MATERIALIZED VIEW hr.regions_bob
  REFRESH FAST WITH PRIMARY KEY FOR UPDATE
  AS SELECT * FROM hr.regions@orc1.world;

/*
```

**手順 4** jane が所有するマテリアライズド・ビューを作成します。

```
*/  
  
CREATE MATERIALIZED VIEW hr.departments_jane  
  REFRESH FAST WITH PRIMARY KEY FOR UPDATE  
  AS SELECT * FROM hr.departments@orcl.world;  
  
CREATE MATERIALIZED VIEW hr.employees_jane  
  REFRESH FAST WITH PRIMARY KEY FOR UPDATE  
  AS SELECT * FROM hr.employees@orcl.world;  
  
/*
```

**手順 5** bob が所有するマテリアライズド・ビューをマテリアライズド・ビュー・グループに追加します。

```
*/  
  
BEGIN  
  DBMS_REPCAT.CREATE_MVIEW_REPOBJECT (  
    gname => 'hr_repg',  
    sname => 'hr',  
    oname => 'countries_bob',  
    type => 'SNAPSHOT',  
    min_communication => TRUE,  
    gowner => 'bob');  
END;  
/  
  
BEGIN  
  DBMS_REPCAT.CREATE_MVIEW_REPOBJECT (  
    gname => 'hr_repg',  
    sname => 'hr',  
    oname => 'departments_bob',  
    type => 'SNAPSHOT',  
    min_communication => TRUE,  
    gowner => 'bob');  
END;  
/
```

```
BEGIN
    DBMS_REPCAT.CREATE_MVIEW_REPOBJECT (
        gname => 'hr_repg',
        sname => 'hr',
        oname => 'employees_bob',
        type => 'SNAPSHOT',
        min_communication => TRUE,
        gowner => 'bob');
END;
/

BEGIN
    DBMS_REPCAT.CREATE_MVIEW_REPOBJECT (
        gname => 'hr_repg',
        sname => 'hr',
        oname => 'jobs_bob',
        type => 'SNAPSHOT',
        min_communication => TRUE,
        gowner => 'bob');
END;
/

BEGIN
    DBMS_REPCAT.CREATE_MVIEW_REPOBJECT (
        gname => 'hr_repg',
        sname => 'hr',
        oname => 'job_history_bob',
        type => 'SNAPSHOT',
        min_communication => TRUE,
        gowner => 'bob');
END;
/

BEGIN
    DBMS_REPCAT.CREATE_MVIEW_REPOBJECT (
        gname => 'hr_repg',
        sname => 'hr',
        oname => 'locations_bob',
        type => 'SNAPSHOT',
        min_communication => TRUE,
        gowner => 'bob');
END;
/
```

```
BEGIN
  DBMS_REPCAT.CREATE_MVIEW_REPOBJECT (
    gname => 'hr_repg',
    sname => 'hr',
    oname => 'regions_bob',
    type => 'SNAPSHOT',
    min_communication => TRUE,
    gowner => 'bob');
END;
/

/*
```

**手順 6** jane が所有するマテリアライズド・ビューをマテリアライズド・ビュー・グループに追加します。

```
*/

BEGIN
  DBMS_REPCAT.CREATE_MVIEW_REPOBJECT (
    gname => 'hr_repg',
    sname => 'hr',
    oname => 'departments_jane',
    type => 'SNAPSHOT',
    min_communication => TRUE,
    gowner => 'jane');
END;
/

BEGIN
  DBMS_REPCAT.CREATE_MVIEW_REPOBJECT (
    gname => 'hr_repg',
    sname => 'hr',
    oname => 'employees_jane',
    type => 'SNAPSHOT',
    min_communication => TRUE,
    gowner => 'jane');
END;
/

SET ECHO OFF

SPOOL OFF
```

**手順 7 マテリアライズド・ビューをリフレッシュ・グループに追加します。**

**関連項目：** マテリアライズド・ビューのリフレッシュ・グループへの追加の詳細は、[第 5 章「マテリアライズド・ビュー・グループの作成」](#)（手順 6）を参照してください。

/\*\*\*\*\* END OF SCRIPT \*\*\*\*\*/



---

# レプリケーション・オブジェクトおよびキューの管理

この章では、レプリケーション・マネージメント API を使用した、レプリケーション環境でのレプリケーション・オブジェクトおよびキューの管理方法を説明します。この章では、次の項目を説明します。

- レプリケート・オブジェクトの変更
- 変更をレプリケートしない表の変更
- レプリケート表での LONG 列の LOB 列への変換
- レプリケート表間の違いの判別
- 遅延トランザクション・キューの管理
- エラー・キューの管理

## レプリケート・オブジェクトの変更

データベースを変更する場合、レプリケート・オブジェクトの特性も変更する必要があります。DDL を直接実行して、レプリケート・オブジェクトを変更しないでください。直接実行すると、レプリケーション環境に障害が発生する原因となります。

### 静止中のマスター・グループにあるレプリケート・オブジェクトの変更

DBMS\_REPCAT パッケージの ALTER\_MASTER\_REPOBJECT プロシージャを使用して、静止中のマスター・グループにあるレプリケート・オブジェクトの特性を変更します。その場合、次のスクリプト例に示すように、必要な DDL をプロシージャ・コールに含めます (ddl\_text パラメータを参照してください)。

互換性のレベルが 9.0.0 より低いマスター・サイトがある場合は、次のプロシージャを使用する必要があります。すなわち、マスター・グループを静止してレプリケート・オブジェクトを変更する必要があります。データベースの互換性のレベルは、COMPATIBLE 初期化パラメータで管理します。

これらのアクションを実行するには、次の要件を満たす必要があります。

**実行者:** レプリケーション管理者

**実行場所:** マスター定義サイト

**レプリケーションの状態:** 静止中

次に示す手順に従って、静止中のマスター・グループにあるレプリケート・オブジェクトを変更します。

---

---

#### 注意:

- 単一マスター環境のマスター・サイトで Oracle8i リリース 8.1.7 以上を実行しており、かつレプリケート・オブジェクトに安全な変更を行う場合は、マスター・グループを静止する必要はありません。静止が不要場合の詳細は、20-27 ページの「ALTER\_MASTER\_REPOBJECT プロシージャ」を参照してください。
  - このドキュメントをオンラインで参照している場合は、次の「BEGINNING OF SCRIPT」の行から 9-5 ページの「END OF SCRIPT」の行までのテキストをテキスト・エディタにコピーして編集し、使用している環境に適したスクリプトを作成します。
- 
-

```
/***** BEGINNING OF SCRIPT *****/
```

**手順 1 レプリケーション管理者としてマスター定義サイトに接続します。**

```
*/
```

```
SET ECHO ON
```

```
SPOOL alter_rep_object.out
```

```
CONNECT repadmin/repadmin@orc1.world
```

```
/*
```

**手順 2 必要な場合は、マスター・グループを静止します。静止が不要な場合の詳細は、20-27 ページの「ALTER\_MASTER\_REPOBJECT プロシージャ」を参照してください。**

```
*/
```

```
BEGIN
```

```
    DBMS_REPCAT.SUSPEND_MASTER_ACTIVITY (  
        gname => 'hr_repg');
```

```
END;
```

```
/
```

```
/*
```

**手順 3 別の SQL\*Plus セッションで静止するマスター・グループのステータスをチェックし、グループのステータスが QUIESCED になるまで進行を中断します。**

ステータスをチェックするには、次の問合せを実行します。

```
SELECT GNAME, STATUS FROM DBA_REPGROUP;
```

```
*/
```

```
PAUSE Press <RETURN> to continue when the master group's status is QUIESCED.
```

```
/*
```

**手順 4 レプリケート・オブジェクトを変更します。**

```
*/  
  
BEGIN  
    DBMS_REPCAT.ALTER_MASTER_REPOBJECT (  
        sname => 'hr',  
        oname => 'employees',  
        type => 'TABLE',  
        ddl_text => 'ALTER TABLE hr.employees ADD (timestamp DATE)';  
END;  
/  
  
/*
```

**手順 5 変更されたオブジェクトに対するレプリケーション・サポートを再生成します。**

```
*/  
  
BEGIN  
    DBMS_REPCAT.GENERATE_REPLICATION_SUPPORT (  
        sname => 'hr',  
        oname => 'employees',  
        type => 'TABLE',  
        min_communication => TRUE);  
END;  
/  
  
/*
```

**手順 6 別の SQL\*Plus セッションで DBA\_REPCATLOG が空かどうかをチェックし、空になるまで進行を中断します。**

他の SQL\*Plus セッションで次の SELECT 文を実行して、DBA\_REPCATLOG ビューを監視します。

```
SELECT * FROM DBA_REPCATLOG WHERE GNAME = 'HR_REPG';  
  
*/  
  
PAUSE Press <RETURN> to continue when DBA_REPCATLOG is empty.  
  
/*
```

**手順7 レプリケーション・アクティビティを再開します。**

```

*/

BEGIN
    DBMS_REPCAT.RESUME_MASTER_ACTIVITY (
        GNAME => 'hr_repg');
END;
/

SET ECHO OFF

SPOOL OFF

/***** END OF SCRIPT *****/

```

## 変更をレプリケートしない表の変更

レプリケート・オブジェクトを変更する必要があるが、この変更をレプリケーション環境の他のサイトにレプリケートしない場合があります。たとえば、次の状況でレプリケーションを使用禁止にする必要があります。

- プロシージャ・レプリケーションを使用して変更を伝播するときは、必ずプロシージャの開始時に行レベル・レプリケーションを使用禁止にします。
- トリガー・アクションを複数回レプリケートするのを回避するため、レプリケート表で定義されたトリガー中のレプリケーションを使用禁止にする必要がある場合があります。9-7 ページの「[レプリケート・トリガーの起動を1回のみにする](#)」を参照してください。
- 競合を手動で解決する場合、表の他のコピーにこの変更をレプリケートしない場合があります。

これが必要になるのは、たとえば、エラー・トランザクションを実行したときに競合するレプリケート更新が成功するように、1つのサイトのレコードの状態を訂正する必要がある場合です。または、トランザクションが接続先サイトで適用できなかったため、起点サイトでのトランザクションの効果を取り消すために、レプリケートされない変更を使用する場合があります。この例では、レプリケーション・マネージメント・ツールを使用して、競合するトランザクションを接続先サイトから削除できます。

変更をレプリケートせずに表を変更するには、DBMS\_REPUTIL パッケージの REPLICATION\_ON プロシージャおよび REPLICATION\_OFF プロシージャを使用します。これらのプロシージャは引数を取らず、生成されたレプリケーション・トリガーでフラグとして使用されます。

---

**注意：** レプリケーションを使用可能または使用禁止にするには、DBMS\_REPUTIL パッケージの EXECUTE 権限を持っている必要があります。

---

## レプリケーションの使用禁止

DBMS\_REPUTIL.REPLICATION\_OFF プロシージャは、カレント・セッションに関する内部レプリケーション変数の状態を FALSE に設定します。すべてのレプリケート・トリガーが、トランザクションをキューに入れる前にこの変数の状態をチェックするため、行レベル・レプリケーションを使用してレプリケート表を変更しても、遅延トランザクションはキューに入れられません。

---

**注意：** レプリケーションのオンまたはオフの切り替えは、カレント・セッションにのみ適用されます。すなわち、現在同じサーバーに接続している他のユーザーは、遅延トランザクション・キューでコミットされる変更の入力を制限されません。

---

プロシージャ・レプリケーションを使用している場合は、次の例に示すようにプロシージャの開始時に REPLICATION\_OFF をコールします。これにより、レプリケーション機能により行レベル・レプリケーションが使用されて変更が伝播されることがなくなります。

```
CREATE OR REPLACE PACKAGE update AS
  PROCEDURE update_emp(adjustment IN NUMBER);
END;
/

CREATE OR REPLACE PACKAGE BODY update AS
  PROCEDURE update_emp(adjustment IN NUMBER) IS
  BEGIN
    --turn off row-level replication for set update
    DBMS_REPUTIL.REPLICATION_OFF;
    UPDATE emp . . .;
    --re-enable replication
    DBMS_REPUTIL.REPLICATION_ON;
  EXCEPTION WHEN OTHERS THEN
    . . .
    DBMS_REPUTIL.REPLICATION_ON;
  END;
END;
/
```

## レプリケーション機能の再有効化

競合を解消した後、またはレプリケート・プロシージャの終了時に、必ず DBMS\_REPUTIL.REPLICATION\_ON をコールし、レプリケート表またはマテリアライズド・ビューに対する変更の通常のレプリケーションを再開してください。このプロシージャは引数を取りません。REPLICATION\_ON のコールにより、内部レプリケーション変数が TRUE に設定されます。

## レプリケート・トリガーの起動を 1 回のみにする

レプリケート表でレプリケート・トリガーを定義済の場合、変更を行うごとにトリガーが 1 回のみ起動するようにする必要があります。通常は、トリガーを起動するのは変更が最初に行われたときのみで、変更がリモート・サイトにレプリケートされたときはリモート・トリガーを起動しません。

トリガーの開始時に DBMS\_REPUTIL.FROM\_REMOTE パッケージ変数の値をチェックする必要があります。この変数の値が FALSE の場合のみ、トリガーで表が更新される必要があります。

または、トリガーを起動する行以外の行を変更する場合は、トリガーの開始時にレプリケーションを使用禁止にし、トリガーの終了時に再び使用可能にできます。この方法により、元の変更のみがリモート・サイトにレプリケートされます。これによりレプリケート・トリガーが各リモート・サイトで起動します。レプリケート・トリガーで実行される更新は、他のサイトにプッシュされません。

このアプローチでは、競合解消はコールされません。このため、トリガーにより生じる変更がデータの整合性に影響しないようにする必要があります。

## レプリケート表での LONG 列の LOB 列への変換

LOB 列はレプリケートできますが、LONG 列はレプリケートできません。LONG 列から CLOB 列へのデータ型の変換、および LONG\_RAW 列から BLOB 列へのデータ型の変換が可能です。

LONG 列を LOB 列に変換すると、このような列のデータは変換後にレプリケートされるため、ネットワーク帯域幅の要件が増加する可能性があります。この項のプロシージャを実行する前に、十分なネットワーク帯域幅があることを確認してください。

**関連項目：** アプリケーションの詳細および LONG から LOB への変換の詳細は、『Oracle9i アプリケーション開発者ガイドラージ・オブジェクト』を参照してください。

次に示す手順に従って、レプリケート表で LONG 列を LOB 列に変換します。

**手順 1 LONG 列のデータがすべてのレプリケーション・サイトで整合性があることを確認します。**

LONG 列を含む表がマスター表として構成されている場合は、LONG 列のデータへの変更はレプリケートされません。このため、LONG 列のデータがすべてのレプリケーション・サイトで一致しない場合があります。変換前に、LONG 列のデータがすべてのマスター・サイトで一致することを確認する必要があります。

**手順 2 レプリケーション管理者としてマスター定義サイトに接続します。次に例を示します。**

```
CONNECT repadmin/repadmin@orc1.world
```

**手順 3 レプリケーションの状態が通常の場合は、静止中に変更します。次に例を示します。**

```
BEGIN
    DBMS_REPCAT.SUSPEND_MASTER_ACTIVITY (
        gname => 'sales_mg');
END;
/
```

**手順 4 DBMS\_REPCAT パッケージの ALTER\_MASTER\_REPOBJECT プロシージャを実行し、LONG 列を LOB 列に変換します。次に例を示します。**

```
BEGIN
    DBMS_REPCAT.ALTER_MASTER_REPOBJECT (
        sname => 'staff',
        oname => 'positions',
        type => 'TABLE',
        ddl_text => 'ALTER TABLE positions MODIFY (job_desc CLOB)');
END;
/
```

類似した ALTER TABLE 文を使用して、LONG\_RAW 列を BLOB 列に変換できます。

**手順 5 変更されたマスター表に対するレプリケーション・サポートを再生成します。次に例を示します。**

```
BEGIN
    DBMS_REPCAT.GENERATE_REPLICATION_SUPPORT (
        sname => 'staff',
        oname => 'positions',
        type => 'TABLE',
        min_communication => TRUE);
END;
/
```

**手順 6 レプリケーションを再開します。次に例を示します。**

```
BEGIN
    DBMS_REPCAT.RESUME_MASTER_ACTIVITY (
        GNAME => 'sales_mg');
END;
/
```

**手順 7 マテリアライズド・ビューがいずれかのマスター・サイトの変更された表をベースにしている場合は、このマテリアライズド・ビューを再作成します。**



## レプリケート表間の違いの判別

レプリケート表間に違いが生じる場合があります。レプリケーション環境を管理するとき、2つのレプリケート表の内容が同じであるかどうかを定期的にチェックできます。両方のサイトが Oracle リリース 7.3 以上のときは、DBMS\_RECTIFIER\_DIFF パッケージ内の次のプロシージャを使用すると、2つの表の違いを識別でき、また任意で調整もできます。

### DIFFERENCES プロシージャの使用

DIFFERENCES プロシージャは、1つの表の2つのレプリカを比較し、第1のレプリカにあるが第2のレプリカにはない行、および第2のレプリカにあるが第1のレプリカにはない行をすべて判別します。このプロシージャの出力は、ユーザーが作成した2つの表に格納されます。第1の表には欠落行の値が格納され、第2の表にはそれぞれの欠落行を含むサイトを示す情報が格納されます。

### RECTIFY プロシージャの使用

RECTIFY プロシージャは、DIFFERENCES プロシージャで生成された情報を使用して、2つの表を調整します。第1の表にはあるが第2の表にはない行はすべて、第2の表に挿入されます。第2の表にはあるが第1の表にはない行はすべて、第2の表から削除されます。

レプリケート表のコピーをすべて同一にリストアするには、次の手順に従います。

**手順 1 表のコピーを1つ選択して参照表にします。他のすべての表のレプリカを必要に応じて更新するときに、このコピーを使用します。**

**手順 2 違いをチェックする必要があるのが、表の中のすべての行および列か、またはサブセットのみかを判別します。**

たとえば、前回の違いのチェック以降に更新されていない行は、チェックしなくてもよい場合があります。すべての列をチェックする必要はありませんが、列のリストには、表の主キーを構成する（または代替識別キーとして指定した）列がすべて含まれている必要があります。

**手順 3 表の中のどの列をチェックするかを決定した後は、比較の結果を保持するための表を2つ作成します。**

まず、比較する列のデータを保持する表を作成する必要があります。たとえば、employees 表の employee\_id 列、salary 列および department\_id 列を比較する場合、次のような CREATE 文を実行する必要があります。

```
CREATE TABLE hr.missing_rows_data (  
  employee_id    NUMBER(6),  
  salary         NUMBER(8,2),  
  department_id  NUMBER(4));
```

次に、行が検出された位置を示す表も作成する必要があります。この表には、次の例に示すデータ型の列を3つ入れる必要があります。

```
CREATE TABLE hr.missing_rows_location (
  present      VARCHAR2(128),
  absent       VARCHAR2(128),
  r_id         ROWID);
```

**手順 4 比較対象の表があるレプリケーション・グループのレプリケーション・アクティビティを中断します。グループのレプリケーション・アクティビティの中断は必須ではありませんが、最初に表を静止せずに調整すると、データに不整合が生じることがあります。**

```
BEGIN
  DBMS_REPCAT.SUSPEND_MASTER_ACTIVITY (
    gname => 'hr_repg');
END;
/
```

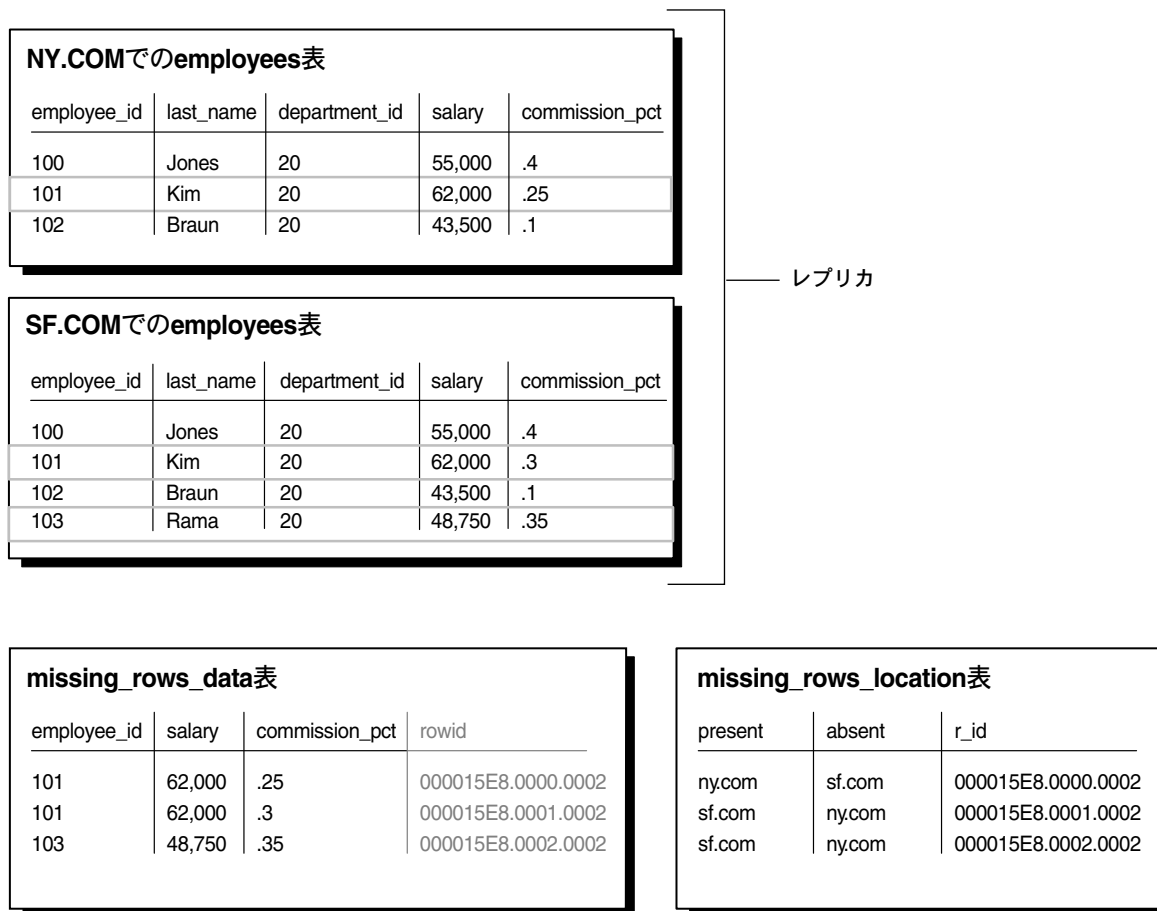
**手順 5 参照表があるサイトで、DBMS\_RECTIFIER\_DIFF パッケージの DIFFERENCES プロシージャをコールします。**

たとえば、ニューヨークのサイトとサンフランシスコのサイトの employees 表を比較する場合、次のようなプロシージャをコールします。

```
BEGIN
  DBMS_RECTIFIER_DIFF.DIFFERENCES (
    sname1          => 'hr',
    oname1          => 'employees',
    reference_site  => 'ny.world',
    sname2          => 'hr',
    oname2          => 'employees',
    comparison_site => 'mv4.world',
    where_clause    => '',
    column_list     => 'employee_id,salary,department_id',
    missing_rows_sname => 'hr',
    missing_rows_oname1 => 'missing_rows_data',
    missing_rows_oname2 => 'missing_rows_location',
    missing_rows_site  => 'ny.world',
    max_missing     => 500,
    commit_rows     => 50);
END;
/
```

図 9-1 に、employee 表の2つのレプリカと、これらのレプリカに対して DIFFERENCES プロシージャを実行した場合に出力される欠落行表の例を示します。

図 9-1 レプリカ間の違いの判別



2つの欠落行表が、ROWID および r\_id 列によって対応付けられている点に注目してください。

**手順 6 次の例に示すように、DBMS\_RECTIFIER\_DIFF パッケージの RECTIFY プロシージャをコールして、参照サイトの表に等しくなるように比較サイトの表を調整します。**

```
BEGIN
    DBMS_RECTIFIER_DIFF.RECTIFY (
        sname1          => 'hr',
        oname1          => 'employees',
        reference_site  => 'ny.world',
        sname2          => 'hr',
        oname2          => 'employees',
        comparison_site => 'mv4.world',
        column_list     => 'employee_id,salary,department_id',
        missing_rows_sname => 'hr',
        missing_rows_oname1 => 'missing_rows_data',
        missing_rows_oname2 => 'missing_rows_location',
        missing_rows_site  => 'ny.world',
        commit_rows     => 50);
END;
/
```

必要な挿入および削除を実行している間、その変更を伝播しないよう、RECTIFY プロシージャは、比較サイトのレプリケーションを一時的に使用禁止にします。RECTIFY は、最初に必要な DELETE 操作をすべて実行し、次に INSERT 操作をすべて実行します。これによって、PRIMARY KEY 制約の違反は発生しなくなります。

RECTIFY プロシージャが正常に実行されると、欠落行表は空になります。

---

---

**注意：** 比較表にその他の制約がある場合は、RECTIFY のコール時にそれらに違反しないようにする必要があります。欠落行表の情報を使用して、表を直接更新することが必要な場合もあります。その場合は、必ず欠落行表から該当の行を DELETE してください。

---

---

**手順 7 レプリケート表の残りのコピーに対して、手順 5 および 6 を繰り返します。このプロシージャの完了時にすべてのコピーが確実に同一になるように、必ず毎回同じ参照表を使用してください。**

**手順 8 マスター・グループのレプリケーション・アクティビティを再開します。**

```
BEGIN
    DBMS_REPCAT.RESUME_MASTER_ACTIVITY (
        gname => 'hr_repg');
END;
/
```

## 遅延トランザクション・キューの管理

通常、Advanced Replication は、遅延トランザクション・キューのプッシュおよびパージを自動的に行うように構成されます。ただし、遅延トランザクション・キューのプッシュやパージを手動で行う必要があることもあります。遅延トランザクション・キューをプッシュするプロセスは、マスター・サイトでもマテリアライズド・ビュー・サイトでも同じです。

### 遅延トランザクション・キューのプッシュ

マスター・サイトは、遅延トランザクション・キューのプッシュを設定された間隔で自動的に行うように構成されます。マテリアライズド・ビュー・サイトでは、マテリアライズド・ビューのリフレッシュ時に、遅延トランザクション・キューのトランザクションを自動的に伝播させない場合は、次の手順を実行して、更新可能なマテリアライズド・ビューに加えられた変更をそのマスター表またはマスター・マテリアライズド・ビューに伝播します。

この例では、マテリアライズド・ビュー・サイトでの遅延トランザクション・キューのプッシュを示しますが、そのプロセスはマスター・サイトでもマテリアライズド・ビュー・サイトでも同じです。

**実行者:** マテリアライズド・ビュー管理者

**実行場所:** マテリアライズド・ビュー・サイト

次の手順に従います。

**手順 1** マテリアライズド・ビュー管理者としてマテリアライズド・ビュー・サイトに接続します。

```
CONNECT mviewadmin/mviewadmin@mv1.world
```

**手順 2** 次の **SELECT** 文を実行して、遅延トランザクションとその接続先を表示します。遅延トランザクション・キューの伝播は、トランザクションの接続先をベースにします。それぞれ固有の接続先と、その接続先に保留されているトランザクションの数が表示されます。

```
SELECT DISTINCT(dblink), COUNT(deferred_tran_id)
FROM deftrandest GROUP BY dblink;
```

**手順 3 遅延トランザクションの接続先としてリストされている各サイトに、DBMS\_DEFER\_SYS.PUSH ファンクションを実行します。**

```
DECLARE
    temp INTEGER;
BEGIN
    temp := DBMS_DEFER_SYS.PUSH (
        destination => 'orcl.world',
        stop_on_error => FALSE,
        delay_seconds => 0,
        parallelism => 0);
END;
/
```

手順 2 で実行した SELECT 文で戻された各接続先に、PUSH プロシージャを実行します。

## 遅延トランザクション・キューのページ

遅延トランザクション・キュー内の正常に伝播されたトランザクションを、定期的に自動的にページするようにシステムが設定されていない場合は、次の手順を実行して、手動でページする必要があります。

この例では、マテリアライズド・ビュー・サイトでの遅延トランザクション・キューのページを示しますが、そのプロセスはマスター・サイトでもマテリアライズド・ビュー・サイトでも同じです。

**実行者:** マテリアライズド・ビュー管理者

**実行場所:** マテリアライズド・ビュー・サイト

次の手順に従います。

**手順 1 マテリアライズド・ビュー管理者としてマテリアライズド・ビュー・サイトに接続します。**

```
CONNECT mviewadmin/mviewadmin@mv1.world
```

**手順 2 遅延トランザクション・キューをページします。**

```
DECLARE
    temp INTEGER;
BEGIN
    temp := DBMS_DEFER_SYS.PURGE (
        purge_method => dbms_defer_sys.purge_method_quick);
END;
/
```

---

**注意:** `purge_method_quick` パラメータを使用する場合、正常にプッシュされた遅延トランザクションおよび遅延プロシージャ・コールが、ページされるまで、それぞれ `DEFTRAN` および `DEFCALL` データ・ディクショナリ・ビューに予想以上に長時間保持されることがあります。詳細は、14-14 ページの `DBMS_DEFER_SYS.PURGE` に関する「[使用上の注意](#)」を参照してください。

---

## AnyData 型を使用した遅延コールの引数の値の判定

レプリケート表で列オブジェクト、コレクションまたは REF を使用している場合、`DBMS_DEFER_QUERY` パッケージの `GET_AnyData_ARG` ファンクションを使用して、これらのユーザー定義型の 1 つを含む遅延コールの引数の値を判定できます。

次の例は、`GET_AnyData_ARG` ファンクションの使用方法を示します。この例では、`oe` サンプル・スキーマで次のユーザー定義型を使用します。

```
CREATE TYPE phone_list_typ AS VARRAY(5) OF VARCHAR2(25);
/

CREATE TYPE warehouse_typ AS OBJECT
(warehouse_id      NUMBER(3),
 warehouse_name    VARCHAR2(35),
 location_id       NUMBER(4)
);
/

CREATE TYPE inventory_typ AS OBJECT
(product_id        NUMBER(6),
 warehouse         warehouse_typ,
 quantity_on_hand  NUMBER(8)
);
/

CREATE TYPE inventory_list_typ AS TABLE OF inventory_typ;
/
```

次のプロシージャは、遅延トランザクション・キューに格納されたコールのコレクション、オブジェクトおよび REF インスタンスの引数値を取得します。このプロシージャは、コール番号およびトランザクション ID が使用可能であることを前提としています。

プロシージャを作成するユーザーは、DBMS\_DEFER\_QUERY パッケージの EXECUTE 権限を持っていることと、CREATE PROCEDURE 権限を持っている必要があります。この例では、oe サンプル・スキーマを使用します。このため、この例を実行するには、oe ユーザーにこれらの権限を与える必要があります。

```
CONNECT system/manager as sysdba
```

```
GRANT EXECUTE ON DBMS_DEFER_QUERY TO oe;
```

```
GRANT CREATE PROCEDURE TO oe;
```

```
CONNECT oe/oe@orcl.world
```

```
CREATE OR REPLACE PROCEDURE get_userdef_arg AS
  call_no      NUMBER := 0;
  txn_id       VARCHAR2(128) := 'xx.xx.xx';
  anydata_val  Sys.AnyData;
  t            SYS.AnyType;
  data_pl      phone_list_typ;      -- varray
  data_ntt     inventory_list_typ;  -- nested table type
  data_p       warehouse_typ;       -- object type
  refl         REF inventory_typ;    -- REF type
  rval         PLS_INTEGER;          -- return value
  tc           PLS_INTEGER;          -- return value
  prec         PLS_INTEGER;          -- precision
  scale        PLS_INTEGER;          -- scale
  len          PLS_INTEGER;          -- length
  csid         PLS_INTEGER;          -- character set id
  csfrm        PLS_INTEGER;          -- character set form
  cnt          PLS_INTEGER;          -- count of varray elements or number of
                                     -- object attributes
  sname        VARCHAR2(35);         -- schema name
  type_name    VARCHAR2(35);         -- type name
  version      VARCHAR2(35);
BEGIN
  FOR i IN 1 .. 5 LOOP
    anydata_val := DBMS_DEFER_QUERY.GET_AnyData_ARG(call_no, i, txn_id);
    -- Get the type information, including type name.
    tc := anydata_val.GetType(t);
    tc := t.GetInfo(prec, scale, len, csid, csfrm, sname, type_name,
                   version, cnt);
    -- Based on the type name, convert the anydata value to the appropriate
    -- user-defined types.
    IF type_name = 'PHONE_LIST_TYP' THEN
```



```
-- The anydata_val contains phone_list_typ varray instance.
rval := anydata_val.GetCollection(data_pl);
-- Do something with data_pl.
ELSIF type_name = 'INVENTORY_LIST_TYP' THEN
-- anydata_val contains inventory_list_typ nested table instance.
rval := anydata_val.GetCollection(data_ntt);
-- Do something with data_ntt.
ELSIF type_name = 'WAREHOUSE_TYP' THEN
-- The anydata_val contains warehouse_typ object instance.
rval := anydata_val.GetObject(data_p);
-- Do something with data_p.
ELSIF type_name = 'INVENTORY_TYP' THEN
-- The anydata_val contains a reference to inventory_typ object instance.
rval := anydata_val.GetRef(ref1);
-- Do something with ref1.
END IF;
END LOOP;
END;
/
```

**関連項目：**

- 13-7 ページ [「GET\\_datatype\\_ARG ファンクション」](#)
- AnyData データ型の詳細は、『Oracle9i SQL リファレンス』および『Oracle9i アプリケーション開発者ガイド - オブジェクト・リレーションアル機能』を参照してください。

## エラー・キューの管理

レプリケーション環境の管理者は、エラー・キューを定期的に監視し、ターゲット・マスター・サイトに正常に適用されなかった遅延トランザクションがあるかどうかを確認します。

エラー・キューを調べるには、ターゲット・マスター・サイトに接続した状態で、次の SELECT 文を（レプリケーション管理者として）実行します。

```
SELECT * FROM deferror;
```

エラー・キューにエラーがあった場合は、エラー条件を解決し、遅延トランザクションを再実行します。遅延トランザクションを再実行するには、2通りの方法があります。つまり、遅延トランザクションを受信したユーザーのセキュリティ・コンテキストで再実行するか、または別のセキュリティ・コンテキストで再実行します。

---

---

**注意：** エラー・トランザクションが複数あり、正しい順序で再実行されたことを確認する場合は、次の項のプロシージャで `deferred_tran_id` パラメータに `NULL` を指定できます。`NULL` を指定しない場合、個々のトランザクションが不正な順序で再実行され、競合の原因となる場合があります。

---

---

## 受信者としてのエラー・トランザクションの再実行

次に示すプロシージャは、遅延トランザクションを受信したユーザーのセキュリティ・コンテキストで、指定された遅延トランザクションを再実行します。このプロシージャは、エラー条件が解決した後で実行します。

これらのアクションを実行するには、次の要件を満たす必要があります。

**実行者：**レプリケーション管理者

**実行場所：**エラーの発生したサイト

**レプリケーションの状態：**通常

次の手順に従います。

**手順 1 レプリケーション管理者としてマスター・サイトに接続します。**

```
CONNECT repadmin/repadmin@orc2.world
```

**手順 2 エラー・トランザクションを再実行します。**

```
BEGIN
    DBMS_DEFER_SYS.EXECUTE_ERROR (
        deferred_tran_id => '1.12.2904',
        destination => 'orc2.world');
END;
/
```

## 代替ユーザーとしてのエラー・トランザクションの再実行

次に示すプロシージャは、カレント接続ユーザーのセキュリティ・コンテキストで、指定された遅延トランザクションを再実行します。このプロシージャは、エラー条件が解決した後で実行します。

これらのアクションを実行するには、次の要件を満たす必要があります。

**実行者:** 接続ユーザー

**実行場所:** エラーの発生したサイト

**レプリケーションの状態:** 通常

次の手順に従います。

**手順 1 代替ユーザーとしてマスター・サイトに接続します。**

```
CONNECT hr/hr@orc2.world
```

**手順 2 エラー・トランザクションを再実行します。**

```
BEGIN
    DBMS_DEFER_SYS.EXECUTE_ERROR_AS_USER (
        deferred_tran_id => '1.12.2904',
        destination => 'orc2.world');
END;
/
```



---

## レプリケーション環境の監視

この章では、データ・ディクショナリを使用してレプリケーション環境を監視する方法を説明します。この章では、次の項目を説明します。

- マスター・レプリケーション環境の監視
- マテリアライズド・ビュー・サイトの監視
- 管理要求の監視
- 遅延トランザクション・キューの監視
- エラー・キューの監視
- レプリケーション環境のパフォーマンスの監視
- レプリケーション環境のパフォーマンスの監視

---

**注意：** Oracle Enterprise Manager のレプリケーション・マネージメント・ツールも、レプリケーション環境を監視するための優れた手段です。この章の問合せで得られる情報のほとんどは、レプリケーション・マネージメント・ツールで利用可能なレポートで検索できます。詳細は、レプリケーション・マネージメント・ツールのオンライン・ヘルプを参照してください。

---

## マスター・レプリケーション環境の監視

この項では、マスター・レプリケーション環境の情報を表示するために実行できる問合せを示します。レプリケーション環境には、マルチマスター環境、マスター・マテリアライズド・ビュー環境、および複数のマスター・サイトとマテリアライズド・ビューを含むハイブリッド環境があります。

### マスター・サイトの監視

この項では、マスター・サイトの情報を表示するために実行できる問合せを示します。

#### マスター・サイトの一般情報のリスト表示

この項の問合せを実行することにより、マスター・サイトに関する次の一般情報を検索できます。

- 管理要求の数。
- 管理要求エラーの数。
- 伝播されていない遅延トランザクションと接続先のペアの数。各遅延トランザクションは伝播される接続先を複数持つ場合があります、また各接続先は、1つの遅延トランザクションと接続先のペアです。

たとえば、遅延トランザクションが10あり、各トランザクションが3つのサイトに伝播される必要がある場合、この問合せで戻される遅延トランザクションのペアは30あります。しばらくして、最初の遅延トランザクションが3つの接続先サイトのうちの2つに伝播された場合、まだ10の遅延トランザクションがありますが、遅延トランザクションのペアは2つ減り、この問合せで戻される伝播されていない遅延トランザクションのペアは28になります。この場合、最初の遅延トランザクションに残されているトランザクションのペアは1つのみです。

- 遅延トランザクション・エラー（エラー・トランザクション）の数。
- まだキューにある正常に伝播されたトランザクションの数。このトランザクションはキューからパージする必要があります。

次の問合せを実行して、カレント・マスター・サイトに関するこの情報をリストします。

```
COLUMN GLOBAL_NAME HEADING 'Database' FORMAT A25
COLUMN ADMIN_REQUESTS HEADING 'Admin|Requests' FORMAT 9999
COLUMN STATUS HEADING 'Admin|Errors' FORMAT 9999
COLUMN TRAN HEADING 'Def|Trans|Pairs' FORMAT 9999
COLUMN ERRORS HEADING 'Def|Trans|Errors' FORMAT 9999
COLUMN COMPLETE HEADING 'Propagated|Trans' FORMAT 9999

SELECT G.GLOBAL_NAME, D.ADMIN_REQUESTS, E.STATUS, DT.TRAN, DE.ERRORS, C.COMPLETE
FROM (SELECT GLOBAL_NAME FROM GLOBAL_NAME) G,
     (SELECT COUNT(ID) ADMIN_REQUESTS FROM DBA_REPCATLOG) D,
     (SELECT COUNT(STATUS) STATUS FROM DBA_REPCATLOG WHERE STATUS = 'ERROR') E,
     (SELECT COUNT(*) TRAN FROM DEFTRANDEST) DT,
     (SELECT COUNT(*) ERRORS FROM DEFERROR) DE,
     (SELECT COUNT(A.DEFERRED_TRAN_ID) COMPLETE FROM DEFTRAN A
WHERE A.DEFERRED_TRAN_ID NOT IN (
     SELECT B.DEFERRED_TRAN_ID FROM DEFTRANDEST B)) C;
```

出力は次のようになります。

Database	Admin		Def		Propagated
	Requests	Errors	Trans Pairs	Trans Errors	
-----	-----	-----	-----	-----	-----
mv4.world	5	0	37	0	53

**注意：** 遅延トランザクション・キューに多数のトランザクションがある場合、この問合せは高コストになる場合があります。

## マスター・グループの監視

この項では、レプリケーション・サイトにあるマスター・グループの情報を表示するために実行できる問合せを示します。

### マスター・グループに関するマスター・サイトのリスト表示

次の問合せを実行して、レプリケーション・サイトの各マスター・グループのマスター・サイトをリストし、各マスター・グループのマスター定義サイトを指定します。

```
COLUMN GNAME HEADING 'Master Group' FORMAT A20
COLUMN DBLINK HEADING 'Sites' FORMAT A25
COLUMN MASTERDEF HEADING 'Master|Definition|Site?' FORMAT A10

SELECT GNAME, DBLINK, MASTERDEF
      FROM DBA_REPSITES
     WHERE MASTER = 'Y'
    AND GNAME NOT IN (SELECT GNAME FROM DBA_REPSITES WHERE SNAPMASTER = 'Y')
    ORDER BY GNAME;
```

SELECT 文の副問合せは、マテリアライズド・ビュー・グループが出力されないようにします。出力は次のようになります。

Master_Group	Sites	Master Definition Site?
-----	-----	-----
HR_RG	mv4.world	Y
HR_RG	NY.WORLD	N

このリストは、mv4.world が hr\_rg マスター・グループのマスター定義サイトであることを示します。

### マスター・グループの一般情報のリスト表示

この項の問合せを使用して、マスター・サイトにあるマスター・グループに関する次の一般情報をリストできます。

- 各マスター・グループの名前。
- 伝播されていない遅延トランザクションと接続先のペアの数。各遅延トランザクションは伝播される接続先を複数持つ場合があります、また各接続先は、1つの遅延トランザクションと接続先のペアです。

たとえば、遅延トランザクションが 10 あり、各トランザクションが 3 つのサイトに伝播される必要がある場合、この問合せで戻される遅延トランザクションのペアは 30 あります。しばらくして、最初の遅延トランザクションが 3 つの接続先サイトのうちの 2 つに伝播された場合、まだ 10 の遅延トランザクションがありますが、遅延トランザクションのペアは 2 つ減り、この問合せで戻される伝播されていない遅延トランザクションのペアは 28 になり



ます。この場合、最初の遅延トランザクションに残されているトランザクションのペアは1つのみです。

- 各マスター・グループの遅延トランザクション・エラー（エラー・トランザクション）の数。
- 各マスター・グループの管理要求の数。
- 各マスター・グループの管理要求エラーの数。

次の問合せを実行し、この情報をリストします。

```
COLUMN GNAME HEADING 'Master Group' FORMAT A15
COLUMN deftran HEADING 'Number of|Deferred|Transaction|Pairs' FORMAT 9999
COLUMN deftranerror HEADING 'Number of|Deferred|Transaction|Errors' FORMAT 9999
COLUMN adminreq HEADING 'Number of|Administrative|Requests' FORMAT 9999
COLUMN adminreqerror HEADING 'Number of|Administrative|Request|Errors'
COLUMN adminreqerror FORMAT 9999
```

```
SELECT G.GNAME,
       NVL(T.CNT1, 0) deftran,
       NVL(IE.CNT2, 0) deftranerror,
       NVL(A.CNT3, 0) adminreq,
       NVL(B.CNT4, 0) adminreqerror
FROM
  (SELECT DISTINCT GNAME FROM DBA_REPGROUP WHERE MASTER='Y') G,
  (SELECT DISTINCT RO.GNAME, COUNT(DISTINCT D.DEFERRED_TRAN_ID) CNT1
   FROM DBA_REPOBJECT RO, DEFCALL D, DEFTRANDEST TD
   WHERE RO.SNAME = D.SCHEMANAME
   AND RO.ONAME = D.PACKAGENAME
   AND RO.TYPE IN ('TABLE', 'PACKAGE', 'SNAPSHOT')
   AND TD.DEFERRED_TRAN_ID = D.DEFERRED_TRAN_ID
   GROUP BY RO.GNAME ) T,
  (SELECT DISTINCT RO.GNAME, COUNT(DISTINCT E.DEFERRED_TRAN_ID) CNT2
   FROM DBA_REPOBJECT RO, DEFCALL D, DEFERROR E
   WHERE RO.SNAME = D.SCHEMANAME
   AND RO.ONAME = D.PACKAGENAME
   AND RO.TYPE IN ('TABLE', 'PACKAGE', 'SNAPSHOT')
   AND E.DEFERRED_TRAN_ID = D.DEFERRED_TRAN_ID
   AND E.CALLNO = D.CALLNO
   GROUP BY RO.GNAME ) IE,
  (SELECT GNAME, COUNT(*) CNT3 FROM DBA_REPCATLOG GROUP BY GNAME) A,
  (SELECT GNAME, COUNT(*) CNT4 FROM DBA_REPCATLOG
   WHERE STATUS = 'ERROR'
   GROUP BY GNAME) B WHERE G.GNAME = IE.GNAME (+)
   AND G.GNAME = T.GNAME (+)
   AND G.GNAME = A.GNAME (+)
   AND G.GNAME = B.GNAME (+) ORDER BY G.GNAME;
```

出力は次のようになります。

Master Group	Number of Deferred Transaction Pairs	Number of Deferred Transaction Errors	Number of Administrative Requests	Number of Administrative Request Errors
HR_RG	54	0	0	0
OE_RG	33	1	5	0

**注意：** 伝播されるのを待っているトランザクションが多数ある場合、この問合せは高コストになる場合があります。

マスターの監視

マスターには、マスター・サイトとマスター・マテリアライズド・ビュー・サイトがあります。この項では、マスターの情報を表示するために実行できる問合せを示します。

マスターでのマテリアライズド・ビューの情報のリスト表示

マスターをベースとするマテリアライズド・ビュー・サイトがある場合は、この項の問合せを使用して、マスターに関する次の情報をリストできます。

- マスターにあるレプリケーション・グループの数。レプリケーション・グループには、マスター・グループとマテリアライズド・ビュー・グループがあります。
- マスターにあるレプリケーション・グループをベースとする、登録済のマテリアライズド・ビュー・グループの数。
- マスターにあるオブジェクトをベースとする、登録済のマテリアライズド・ビューの数。オブジェクトには、マスター表とマスター・マテリアライズド・ビューがあります。
- マスターにあるマテリアライズド・ビュー・ログの数。
- マスターにある配置テンプレートの数。

次の問合せを実行し、この情報をリストします。

```

COLUMN repgroup HEADING 'Number of|Replication|Groups' FORMAT 9999
COLUMN mvgroup HEADING 'Number of|Registered|MV Groups' FORMAT 9999
COLUMN mv HEADING 'Number of|Registered MVs' FORMAT 9999
COLUMN mvlog HEADING 'Number of|MV Logs' FORMAT 9999
COLUMN template HEADING 'Number of|Templates' FORMAT 9999

SELECT A.REPGROUP repgroup,
       B.MVGROUP mvgroup,
       C.MV mv,
       D.MVLOG mvlog,
       E.TEMPLATE template
FROM (SELECT COUNT(G.GNAME) REPGROUP
      FROM DBA_REPGROUP G, DBA_REPSITES S
      WHERE G.MASTER = 'Y'
      AND S.MASTER = 'Y'
      AND G.GNAME = S.GNAME
      AND S.MY_DBLINK = 'Y') A,
     (SELECT COUNT(*) MVGROUP
      FROM DBA_REGISTERED_MVIEW_GROUPS) B,
     (SELECT COUNT(*) MV
      FROM DBA_REGISTERED_MVIEWS) C,
     (SELECT COUNT(*) MVLOG
      FROM (SELECT 1 FROM DBA_MVIEW_LOGS
            GROUP BY LOG_OWNER, LOG_TABLE)) D,
     (SELECT COUNT(*) TEMPLATE FROM DBA_REPCAT_REFRESH_TEMPLATES) E;

```

出力は次のようになります。

Number of Replication Groups	Number of Registered MV Groups	Number of Registered MVs	Number of MV Logs	Number of Templates
1	5	27	6	3

## マスターにあるマテリアライズド・ビュー・ログの情報のリスト表示

マテリアライズド・ビュー・ログにより、マスターをベースとするマテリアライズド・ビューを高速リフレッシュできます。マスターには、マスター表とマスター・マテリアライズド・ビューがあります。マスターをベースとするマテリアライズド・ビュー・ログがある場合は、この項の問合せを使用して、このログに関する次の情報をリストできます。

- マテリアライズド・ビュー・ログ・データが格納される各ログ表の名前
- 各マテリアライズド・ビュー・ログの所有者
- 各マテリアライズド・ビュー・ログがベースとするマスター
- マテリアライズド・ビュー・ログが行 ID マテリアライズド・ビュー・ログかどうか
- マテリアライズド・ビュー・ログが主キー・マテリアライズド・ビュー・ログかどうか
- マテリアライズド・ビュー・ログがオブジェクト ID マテリアライズド・ビュー・ログかどうか
- マテリアライズド・ビュー・ログにフィルタ列があるかどうか

次の問合せを実行し、この情報をリストします。

```
COLUMN LOG_TABLE HEADING 'Log Table' FORMAT A20
COLUMN LOG_OWNER HEADING 'Log|Owner' FORMAT A5
COLUMN MASTER HEADING 'Master' FORMAT A15
COLUMN ROWIDS HEADING 'Row|ID?' FORMAT A3
COLUMN PRIMARY_KEY HEADING 'Primary|Key?' FORMAT A7
COLUMN OBJECT_ID HEADING 'Object|ID?' FORMAT A6
COLUMN FILTER_COLUMNS HEADING 'Filter|Columns?' FORMAT A8

SELECT DISTINCT LOG_TABLE,
               LOG_OWNER,
               MASTER,
               ROWIDS,
               PRIMARY_KEY,
               OBJECT_ID,
               FILTER_COLUMNS
FROM DBA_MVIEW_LOGS
ORDER BY 1;
```

出力は次のようになります。

Log Table	Log		Row ID?	Primary Key?	Object ID?	Filter Columns?
	Owner	Master				
MLOG\$_COUNTRIES	HR	COUNTRIES	NO	YES	NO	NO
MLOG\$_DEPARTMENTS	HR	DEPARTMENTS	NO	YES	NO	NO
MLOG\$_EMPLOYEES	HR	EMPLOYEES	NO	YES	NO	NO
MLOG\$_JOBS	HR	JOBS	NO	YES	NO	NO
MLOG\$_JOB_HISTORY	HR	JOB_HISTORY	NO	YES	NO	NO
MLOG\$_LOCATIONS	HR	LOCATIONS	NO	YES	NO	NO

**関連項目：** マテリアライズド・ビュー・ログの詳細は、『Oracle9i アドバンスド・レプリケーション』を参照してください。

## マテリアライズド・ビュー・ログを使用するマテリアライズド・ビューのリスト表示

あるマテリアライズド・ビュー・ログを複数のマテリアライズド・ビューが使用する場合があります。マスターをベースとするマテリアライズド・ビュー・ログがある場合は、この項の問合せを使用して、各ログを使用するマテリアライズド・ビューに関する次の情報をリストできます。

- マテリアライズド・ビュー・ログ・データが格納される各ログ表の名前
- 各マテリアライズド・ビュー・ログの所有者
- 各マテリアライズド・ビュー・ログがベースとするマスター
- マテリアライズド・ビュー・ログを使用する各マテリアライズド・ビューの、マテリアライズド・ビュー識別番号
- マテリアライズド・ビュー・ログを使用する各マテリアライズド・ビューの名前

次の問合せを実行し、この情報をリストします。

```

COLUMN LOG_TABLE HEADING 'Mview|Log Table' FORMAT A20
COLUMN LOG_OWNER HEADING 'Mview|Log Owner' FORMAT A10
COLUMN MASTER HEADING 'Master' FORMAT A20
COLUMN MVIEW_ID HEADING 'Mview|ID' FORMAT 9999
COLUMN NAME HEADING 'Mview Name' FORMAT A20

SELECT L.LOG_TABLE, L.LOG_OWNER, B.MASTER, B.MVIEW_ID, R.NAME
FROM ALL_MVIEW_LOGS L, ALL_BASE_TABLE_MVIEWS B, ALL_REGISTERED_MVIEWS R
WHERE B.MVIEW_ID = R.MVIEW_ID
AND B.OWNER = L.LOG_OWNER
AND B.MASTER = L.MASTER;
```

出力は次のようになります。

Mview			Mview
Log Table	Owner	Master	ID Mview Name
-----	-----	-----	-----
MLOG\$_COUNTRIES	HR	COUNTRIES	24 COUNTRIES_MVIEW1
MLOG\$_COUNTRIES	HR	COUNTRIES	31 COUNTRIES_MVIEW2
MLOG\$_DEPARTMENTS	HR	DEPARTMENTS	19 DEPARTMENTS_MVIEW1
MLOG\$_DEPARTMENTS	HR	DEPARTMENTS	64 DEPARTMENTS_MVIEW2
MLOG\$_DEPARTMENTS	HR	DEPARTMENTS	15 DEPARTMENTS_MVIEW3

マスターにある配置テンプレートの情報のリスト表示

配置テンプレートにより、複数のマテリアライズド・ビュー環境を素早く作成できます。また、変数を使用して、各マテリアライズド・ビュー環境を個別の必要に応じてカスタマイズすることもできます。この項の問合せを使用して、マスターにある配置テンプレートに関する次の情報をリストできます。

- 各配置テンプレートの名前
- 各配置テンプレートの所有者
- 配置テンプレートがパブリックかどうか
- 各配置テンプレートをベースとする、インスタンス化されたマテリアライズド・ビュー・サイトの数
- 各配置テンプレートに対応付けられたコメント

次の問合せを実行し、この情報をリストします。

```
COLUMN REFRESH_TEMPLATE_NAME HEADING 'Template|Name' FORMAT A10
COLUMN OWNER HEADING 'Owner' FORMAT A10
COLUMN PUBLIC_TEMPLATE HEADING 'Public?' FORMAT A7
COLUMN INSTANTIATED HEADING 'Number of|Instantiated|Sites' FORMAT 9999
COLUMN TEMPLATE_COMMENT HEADING 'Comment' FORMAT A35

SELECT DISTINCT RT.REFRESH_TEMPLATE_NAME,
                OWNER,
                PUBLIC_TEMPLATE,
                RS.INSTANTIATED,
                RT.TEMPLATE_COMMENT
FROM DBA_REPCAT_REFRESH_TEMPLATES RT,
     (SELECT Y.REFRESH_TEMPLATE_NAME, COUNT(X.STATUS) INSTANTIATED
      FROM DBA_REPCAT_TEMPLATE_SITES X, DBA_REPCAT_REFRESH_TEMPLATES Y
      WHERE X.REFRESH_TEMPLATE_NAME(+) = Y.REFRESH_TEMPLATE_NAME
      GROUP BY Y.REFRESH_TEMPLATE_NAME) RS
WHERE RT.REFRESH_TEMPLATE_NAME(+) = RS.REFRESH_TEMPLATE_NAME
ORDER BY 1;
```

出力は次のようになります。

Template Name	Owner	Public?	Number of Instantiated Sites	Comment
-----	-----	-----	-----	-----
HR_REFG_DT	HR	N	2	Human Resources Deployment Template

Public? 列の N は、配置テンプレートがプライベートであることを意味します。したがって、権限を与えられたユーザーのみがこのテンプレートをインスタンス化できます。この列の Y は、配置テンプレートがパブリックであることを意味します。パブリック配置テンプレートはすべてのユーザーがインスタンス化できます。

## マテリアライズド・ビュー・サイトの監視

この項では、マテリアライズド・ビュー・サイトの情報を表示するために実行できる問合せを示します。

### マテリアライズド・ビュー・サイトの一般情報のリスト表示

この項の問合せを使用して、カレント・マテリアライズド・ビュー・サイトに関する次の一般情報をリストできます。

- サイトにあるマテリアライズド・ビュー・グループの数
- サイトにあるマテリアライズド・ビューの数
- サイトにあるリフレッシュ・グループの数

次の問合せを実行し、この情報をリストします。

```

COLUMN MVGROUP HEADING 'Number of|Materialized|View Groups' FORMAT 9999
COLUMN MV HEADING 'Number of|Materialized|Views' FORMAT 9999
COLUMN RGROUP HEADING 'Number of|Refresh Groups' FORMAT 9999

SELECT A.MVGROUP, B.MV, C.RGROUP
FROM
  (SELECT COUNT(S.GNAME) MVGROUP
   FROM DBA_REPSITES S
   WHERE S.SNAPMASTER = 'Y') A,
  (SELECT COUNT(*) MV
   FROM DBA_MVIEWS) B,
  (SELECT COUNT(*) RGROUP
   FROM DBA_REFRESH) C;
```

出力は次のようになります。

Number of Materialized View Groups	Number of Materialized Views	Number of Refresh Groups
5	25	5

マテリアライズド・ビュー・グループの一般情報のリスト表示

この項の問合せを使用して、カレント・マテリアライズド・ビュー・サイトにあるマテリアライズド・ビュー・グループに関する次の一般情報をリストできます。

- 各マテリアライズド・ビュー・グループの名前
- 各マテリアライズド・ビュー・グループのマスター
- マテリアライズド・ビュー・グループのマスターへの伝播の方法が非同期か同期か
- 各マテリアライズド・ビュー・グループに対応付けられたコメント

次の問合せを実行し、この情報をリストします。

```
COLUMN GNAME HEADING 'Group Name' FORMAT A10
COLUMN DBLINK HEADING 'Master' FORMAT A25
COLUMN Propagation HEADING 'Propagation|Method' FORMAT A12
COLUMN SCHEMA_COMMENT HEADING 'Comment' FORMAT A30

SELECT S.GNAME,
       S.DBLINK,
       DECODE(S.PROP_UPDATES,
              0, 'ASYNCHRONOUS',
              1, 'SYNCHRONOUS') Propagation,
       G.SCHEMA_COMMENT
FROM DBA_REPSITES S, DBA_REPGROUP G
WHERE S.GNAME = G.GNAME
AND S.SNAPMASTER = 'Y';
```

出力は次のようになります。

Group Name	Master	Propagation Method	Comment
HR_RG	mv4.world	ASYNCHRONOUS	Human Resources Group



## マテリアライズド・ビューの情報のリスト表示

この項では、レプリケーション・サイトにあるマテリアライズド・ビューの情報を表示するために実行できる問合せを示します。

### マテリアライズド・ビューのマスター情報のリスト表示

次の問合せは、レプリケーション・サイトにある各マテリアライズド・ビューのマスターを示し、かつマテリアライズド・ビューの高速リフレッシュが可能かどうかを示します。

```
COLUMN MVIEW_NAME HEADING 'Materialized|View Name' FORMAT A15
COLUMN OWNER HEADING 'Owner' FORMAT A10
COLUMN MASTER_LINK HEADING 'Master Link' FORMAT A30
COLUMN Fast_Refresh HEADING 'Fast|Refreshable?' FORMAT A16

SELECT MVIEW_NAME,
       OWNER,
       MASTER_LINK,
       DECODE (FAST_REFRESHABLE,
               'NO', 'NO',
               'DML', 'YES',
               'DIRLOAD', 'DIRECT LOAD ONLY',
               'DIRLOAD_DML', 'YES',
               'DIRLOAD_LIMITEDDML', 'LIMITED') Fast_Refresh
FROM DBA_MVIEWS;
```

出力は次のようになります。

Materialized View Name	Owner	Master Link	Fast Refreshable?
DEPARTMENTS_MV	HR	@mv4.world	YES
EMPLOYEES_MV	HR	@mv4.world	YES
JOBS_MV	HR	@mv4.world	YES
JOB_HISTORY_MV	HR	@mv4.world	YES
LOCATIONS_MV	HR	@mv4.world	YES

マテリアライズド・ビューのプロパティのリスト表示

この項の問合せを使用して、カレント・レプリケーション・サイトにあるマテリアライズド・ビューに関する次の情報をリストできます。

- 各マテリアライズド・ビューの名前
- 各マテリアライズド・ビューの所有者
- 各マテリアライズド・ビューが使用するリフレッシュ方法が COMPLETE、FORCE、FAST または NEVER のいずれであるか
- マテリアライズド・ビューが更新可能かどうか
- 各マテリアライズド・ビューがリフレッシュされた最後の日付

次の問合せを実行し、この情報をリストします。

```
COLUMN MVIEW_NAME HEADING 'Materialized|View Name' FORMAT A15
COLUMN OWNER HEADING 'Owner' FORMAT A10
COLUMN REFRESH_METHOD HEADING 'Refresh|Method' FORMAT A10
COLUMN UPDATABLE HEADING 'Updatable?' FORMAT A10
COLUMN LAST_REFRESH_DATE HEADING 'Last|Refresh|Date'
COLUMN LAST_REFRESH_TYPE HEADING 'Last|Refresh|Type' FORMAT A15

SELECT MVIEW_NAME,
       OWNER,
       REFRESH_METHOD,
       UPDATABLE,
       LAST_REFRESH_DATE,
       LAST_REFRESH_TYPE
FROM DBA_MVIEWS;
```

出力は次のようになります。

Materialized View Name	Owner	Refresh Method	Updatable?	Last Refresh Date	Last Refresh Type
DEPARTMENTS_MV	HR	FORCE	Y	22-JAN-01	FAST
EMPLOYEES_MV	HR	FAST	Y	22-JAN-01	COMPLETE
JOBS_MV	HR	COMPLETE	Y	22-JAN-01	COMPLETE
JOB_HISTORY_MV	HR	FAST	Y	22-JAN-01	FAST
LOCATIONS_MV	HR	FAST	Y	22-JAN-01	FAST

## マテリアライズド・ビュー・サイトにあるリフレッシュ・グループの情報のリスト表示

マテリアライズド・ビュー・サイトにある各リフレッシュ・グループには、設定された間隔でリフレッシュ・グループのマテリアライズド・ビューをリフレッシュする、リフレッシュ・ジョブが対応付けられています。DBA\_REFRESH データ・ディクショナリ・ビューの間合せにより、マテリアライズド・ビュー・サイトにあるリフレッシュ・ジョブに関する次の情報をリストできます。

- リフレッシュ・グループの名前。
- リフレッシュ・グループの所有者。
- リフレッシュ・ジョブが中断されているかどうか。
- 次回リフレッシュ・ジョブが実行される日付および時刻。
- リフレッシュ・ジョブの現在の間隔設定。間隔設定では、ジョブの開始と同じジョブの次の開始の間の、時間の長さを指定します。

次の問合せでこの情報が表示されます。

```

COLUMN RNAME HEADING 'Refresh|Group|Name' FORMAT A10
COLUMN ROWNER HEADING 'Refresh|Group|Owner' FORMAT A10
COLUMN BROKEN HEADING 'Broken?' FORMAT A7
COLUMN next_refresh HEADING 'Next Refresh'
COLUMN INTERVAL HEADING 'Interval' FORMAT A20

SELECT RNAME,
       ROWNER,
       BROKEN,
       TO_CHAR(NEXT_DATE, 'DD-MON-YYYY HH:MI:SS AM') next_refresh,
       INTERVAL
FROM DBA_REFRESH
ORDER BY 1;

```

出力は次のようになります。

Refresh Group Name	Refresh Group Owner	Broken?	Next Refresh	Interval
HR_REFG	MVIEWADMIN	N	01-JAN-4000 12:00:00 AM	SYSDATE + 1/24

Broken? 列の N は、ジョブが中断されていないことを意味します。したがって、リフレッシュ・ジョブは次回開始時刻に実行されます。この列の Y は、ジョブが中断されていることを意味します。

## マテリアライズド・ビュー・サイトにある各リフレッシュ・ジョブのジョブ ID の判定

この項の問合せを使用して、マテリアライズド・ビュー・サイトにあるリフレッシュ・ジョブに関する次の情報をリストできます。

- 各リフレッシュ・ジョブのジョブ識別番号。DBMS\_JOBS パッケージで作成された各ジョブには、一意の識別番号が割り当てられます。
- 権限スキーマ。権限スキーマとは、ジョブに適用されるデフォルトの権限を所有するスキーマです。
- 各リフレッシュ・ジョブを所有するスキーマ。通常は、マテリアライズド・ビュー管理者がリフレッシュ・ジョブを所有します。マテリアライズド・ビュー管理者の共通ユーザー名は mviewadmin です。
- ジョブがリフレッシュするリフレッシュ・グループの名前。
- リフレッシュ・ジョブのステータスが通常か中断か。

次の問合せでこの情報が表示されます。

```
COLUMN JOB HEADING 'Job ID' FORMAT 999999
COLUMN PRIV_USER HEADING 'Privilege|Schema' FORMAT A10
COLUMN RNAME HEADING 'Refresh|Group|Name' FORMAT A10
COLUMN ROWNER HEADING 'Refresh|Group|Owner' FORMAT A10
COLUMN BROKEN HEADING 'Broken?' FORMAT A7

SELECT J.JOB,
       J.PRIV_USER,
       R.ROWNER,
       R.RNAME,
       J.BROKEN
FROM DBA_REFRESH R, DBA_JOBS J
WHERE R.JOB = J.JOB
ORDER BY 1;
```

出力は次のようになります。

		Refresh	Refresh	Broken?
		Group	Group	
Job ID	Schema	Owner	Name	
-----				
21	MVIEWADMIN	MVIEWADMIN	HR_REFG	N

Broken? 列の N は、ジョブが中断されていないことを意味します。したがって、ジョブは次回開始時刻に実行されます。この列の Y は、ジョブが中断されていることを意味します。

## 現在リフレッシュしているマテリアライズド・ビューの判定

次の問合せは、現在リフレッシュしているマテリアライズド・ビューを示します。

```
COLUMN SID HEADING 'Session|Identifier' FORMAT 9999
COLUMN SERIAL# HEADING 'Serial|Number' FORMAT 999999
COLUMN CURRMVOWNER HEADING 'Owner' FORMAT A15
COLUMN CURRMVNAME HEADING 'Materialized|View' FORMAT A25

SELECT * FROM V$MVREFRESH;
```

出力は次のようになります。

Session Identifier	Serial Number	Owner	Materialized View
19	233	HR	COUNTRIES_MV
5	647	HR	EMPLOYEES_MV

**注意：** マテリアライズド・ビューの遅延トランザクションがマスターにプッシュされているときは、V\$MVREFRESH 動的パフォーマンス・ビューには、更新可能なマテリアライズド・ビューの情報は含まれません。

## 管理要求の監視

この項では、マスター・サイトにある管理要求の情報を表示するために実行できる問合せを示します。

### 管理要求の一般情報のリスト表示

この項の問合せを使用して、マスター・サイトにある管理要求に関する次の一般情報をリストできます。

- 各管理要求の識別番号
- 各管理要求が要求するアクション
- 各要求のステータス
- 要求が実行されるマスター・サイト

次の問合せでこの情報が表示されます。

```
COLUMN ID HEADING 'Admin|Request|ID' FORMAT 999999
COLUMN REQUEST HEADING 'Request' FORMAT A25
COLUMN STATUS HEADING 'Status' FORMAT A15
COLUMN MASTER HEADING 'Master|Site' FORMAT A25

SELECT ID, REQUEST, STATUS, MASTER FROM DBA_REPCATLOG;
```

出力は次のようになります。

Admin		Master	
Request		Site	
ID	Request	Status	Site
-----			
44	RESUME_MASTER_ACTIVITY	AWAIT_CALLBACK	NY.WORLD

管理要求エラーの原因の判定

管理要求エラーの原因を、エラー・メッセージを表示して判定できます。次の問合せは、エラーとなった各管理要求に関するエラー・メッセージを表示します。

```
COLUMN ID HEADING 'Admin|Request|ID' FORMAT 999999
COLUMN REQUEST HEADING 'Request' FORMAT A30
COLUMN ERRNUM HEADING 'Error|Number' FORMAT 999999
COLUMN MESSAGE HEADING 'Error|Message' FORMAT A32

SELECT ID, REQUEST, ERRNUM, MESSAGE
FROM DBA_REPCATLOG WHERE STATUS = 'ERROR';
```

出力は次のようになります。

Admin		Error	Error
Request		Number	Message
ID	Request		
-----			
70	CREATE_MASTER_REPOBJECT	-2292	ORA-02292: 整合性制約 (HR.DEPT_LOC_FK) に違反しました - 子レコードがあります ORA-02266: 表には使用可能な外部キーによって参照される一意キー / 主キーが含まれています。
71	GENERATE_INTERNAL_PKG_SUPPORT	-23308	ORA-23308: オブジェクト HR.LOCATIONS が存在しないか、無効です。

管理要求を実行するジョブの一般情報のリスト表示

各マスター・グループには、管理要求を実行する do\_deferred\_repcat\_admin ジョブが対応付けられています。DBA\_JOBS データ・ディクショナリ・ビューの間合せにより、レプリケーション・サイトにあるこのジョブに関する次の情報をリストできます。

- 各 do\_deferred\_repcat\_admin ジョブのジョブ識別番号。DBMS\_JOBS パッケージで作成された各ジョブには、一意の識別番号が割り当てられます。
- 権限スキーマ。権限スキーマとは、ジョブに適用されるデフォルトの権限を所有するスキーマです。
- 各 do\_deferred\_repcat\_admin ジョブのステータスが通常か中断か。
- 次回各 do\_deferred\_repcat\_admin ジョブが実行される日付および時刻。
- 各 do\_deferred\_repcat\_admin ジョブの現在の間隔設定。間隔設定では、ジョブの開始と同じジョブの次の開始の間の、時間の長さを指定します。

次の間合せでこの情報が表示されます。

```
COLUMN JOB HEADING 'Job ID' FORMAT 999999
COLUMN PRIV_USER HEADING 'Privilege|Schema' FORMAT A10
COLUMN BROKEN HEADING 'Broken?' FORMAT A7
COLUMN next_start HEADING 'Next Start'
COLUMN INTERVAL HEADING 'Interval' FORMAT A20

SELECT JOB,
        PRIV_USER,
        BROKEN,
        TO_CHAR(NEXT_DATE,'DD-MON-YYYY HH:MI:SS AM') next_start,
        INTERVAL
FROM DBA_JOBS
WHERE WHAT LIKE '%dbms_repcat.do_deferred_repcat_admin%'
ORDER BY 1;
```

出力は次のようになります。

Privilege				
Job ID	Schema	Broken?	Next Start	Interval
3	REPADMIN	N	02-FEB-2001 04:34:36 PM	SYSDATE + (1/144)

Broken? 列の N は、ジョブが中断されていないことを意味します。したがって、ジョブは次回開始時刻に実行されます。この列の Y は、ジョブが中断されていることを意味します。

## 各 do\_deferred\_repcat\_admin ジョブの定義のチェック

DBA\_JOBS データ・ディクショナリ・ビューの問合せにより、レプリケーション・サイトにある各 do\_deferred\_repcat\_admin ジョブの定義を表示できます。次の問合せで定義が表示されます。

```
COLUMN JOB HEADING 'Job ID' FORMAT 999999
COLUMN WHAT HEADING 'Definitions of Admin Req Jobs' FORMAT A70
```

```
SELECT JOB, WHAT
FROM DBA_JOBS
WHERE WHAT LIKE '%dbms_repcat.do_deferred_repcat_admin%'
ORDER BY 1;
```

出力は次のようになります。

```
Job ID Definitions of Admin Req Jobs
-----
321 dbms_repcat.do_deferred_repcat_admin('"HR_RG"', FALSE);
342 dbms_repcat.do_deferred_repcat_admin('"OE_RG"', FALSE);
```

## 遅延トランザクション・キューの監視

この項では、レプリケーション・サイトにある遅延トランザクション・キューの情報を表示するために実行できる問合せを示します。

### トランザクション伝播の監視

この項では、遅延トランザクション・キューにあるトランザクションの伝播の情報を表示するために実行できる問合せを示します。

#### 各接続先マスター・サイトの遅延トランザクションの数のリスト表示

この項の問合せを実行することにより、各接続先マスター・サイトの伝播されていない遅延トランザクションの数がわかります。この問合せは、カレント・マスター・サイトが遅延トランザクションを伝播している各マスター・サイトを示し、かつ各接続先サイトに伝播される遅延トランザクションの数を示します。

次の問合せを実行し、遅延トランザクションおよびエラー・トランザクションの数を確認します。

```
COLUMN DEST HEADING 'Destination' FORMAT A45
COLUMN TRANS HEADING 'Def Trans' FORMAT 9999
```

```
SELECT DBLINK DEST, COUNT(*) TRANS
FROM DEFTRANDEST D
GROUP BY DBLINK;
```



出力は次のようになります。

Destination	Def Trans
-----	-----
NY.WORLD	27
mv4.world	44

**注意：** 伝播されるのを待っているトランザクションが多数ある場合、この問合せは高コストになる場合があります。

## レプリケーション・サイトにあるプッシュ・ジョブの一般情報のリスト表示

レプリケーション・サイトにある各スケジュール・リンクには、遅延トランザクション・キューにある遅延トランザクションを接続先サイトに伝播する、プッシュ・ジョブが対応付けられています。この項の問合せを使用して、レプリケーション・サイトにあるプッシュ・ジョブに関する次の情報をリストできます。

- 各プッシュ・ジョブのジョブ識別番号。DBMS\_JOBS パッケージで作成された各ジョブには、一意の識別番号が割り当てられます。
- 権限スキーマ。権限スキーマとは、ジョブに適用されるデフォルトの権限を所有するスキーマです。
- 遅延トランザクションがプッシュされる接続先サイト。
- プッシュ・ジョブのステータスが通常か中断か。

次の問合せでこの情報が表示されます。

```
COLUMN JOB HEADING 'Job ID' FORMAT 999999
COLUMN PRIV_USER HEADING 'Privilege|Schema' FORMAT A10
COLUMN DBLINK HEADING 'Destination' FORMAT A40
COLUMN BROKEN HEADING 'Broken?' FORMAT A7

SELECT J.JOB,
       J.PRIV_USER,
       S.DBLINK,
       J.BROKEN
FROM DEFSCHEDULE S, DBA_JOBS J
WHERE S.DBLINK != (SELECT GLOBAL_NAME FROM GLOBAL_NAME)
AND S.JOB = J.JOB
ORDER BY 1;
```

出力は次のようになります。

Privilege		Destination	Broken?
Job ID	Schema		
2	REPADMIN	NY.WORLD	N

Broken? 列の N は、ジョブが中断されていないことを意味します。したがって、ジョブは次回開始時刻に実行されます。この列の Y は、ジョブが中断されていることを意味します。

プッシュ・ジョブの次回開始時刻および間隔の判定

レプリケーション・サイトにある各スケジュール・リンクには、遅延トランザクション・キューにある遅延トランザクションを接続先サイトに伝播する、プッシュ・ジョブが対応付けられています。DEFSCHEDULE および DBA\_JOBS データ・ディクショナリ・ビューの問合せにより、レプリケーション・サイトにあるプッシュ・ジョブに関する次の情報をリストできます。

- 各プッシュ・ジョブのジョブ識別番号。DBMS\_JOBS パッケージで作成された各ジョブには、一意の識別番号が割り当てられます。
- 遅延トランザクションがプッシュされる接続先サイト。
- 次回プッシュ・ジョブが実行される日付および時刻。
- プッシュ・ジョブの現在の間隔設定。間隔設定では、ジョブの開始と同じジョブの次の開始の間の、時間の長さを指定します。

次の問合せでこの情報が表示されます。

```
COLUMN JOB HEADING 'Job ID' FORMAT 999999
COLUMN DBLINK HEADING 'Destination' FORMAT A22
COLUMN next_start HEADING 'Next Start'
COLUMN INTERVAL HEADING 'Interval' FORMAT A25

SELECT JOB,
       DBLINK,
       TO_CHAR(NEXT_DATE, 'DD-MON-YYYY HH:MI:SS AM') next_start,
       INTERVAL
FROM DEFSCHEDULE
WHERE DBLINK != (SELECT GLOBAL_NAME FROM GLOBAL_NAME)
AND JOB IS NOT NULL
ORDER BY 1;
```

出力は次のようになります。

Job ID	Destination	Next Start	Interval
2	NY.WORLD	02-FEB-2001 04:44:39 PM	SYSDATE + 10 / (24 * 60)

### 伝播のためにキューに入れられているトランザクションの合計数の判定

次の問合せを実行して、遅延トランザクション・キューで伝播されるのを待っているトランザクションの合計数を表示します。

```
SELECT COUNT(DISTINCT DEFERRED_TRAN_ID) "Transactions Queued"
FROM DEFTRANDEST;
```

出力は次のようになります。

Transactions Queued
37

**注意：** 伝播されるのを待っているトランザクションが多数ある場合、この問合せは高コストになる場合があります。

## 正常に伝播されたトランザクションのページの監視

この項では、遅延トランザクション・キューから正常に伝播されたトランザクションの、ページの情報を表示するために実行できる問合せを示します。

### ページ・ジョブの一般情報のリスト表示

レプリケーション・サイトの標準設定の途中でページ・ジョブを構成して、正常に伝播されたトランザクションを遅延トランザクション・キューから削除します。DBA\_JOBS データ・ディクショナリ・ビューの問合せにより、レプリケーション・サイトにあるページ・ジョブに関する次の情報をリストできます。

- ページ・ジョブのジョブ識別番号。DBMS\_JOBS パッケージで作成された各ジョブには、一意の識別番号が割り当てられます。
- 権限スキーマ。権限スキーマとは、ジョブに適用されるデフォルトの権限を所有するスキーマです。
- ジョブのステータスが通常か中断か。
- 次回ページ・ジョブが実行される日付および時刻。
- ページ・ジョブの現在の間隔設定。間隔設定では、ジョブの開始と同じジョブの次の開始の間の、時間の長さを指定します。

次の問合せでこの情報が表示されます。

```
COLUMN JOB HEADING 'Job ID' FORMAT 999999
COLUMN PRIV_USER HEADING 'Privilege|Schema' FORMAT A10
COLUMN BROKEN HEADING 'Broken?' FORMAT A7
COLUMN next_start HEADING 'Next Start'
COLUMN INTERVAL HEADING 'Interval' FORMAT A25

SELECT JOB,
       PRIV_USER,
       BROKEN,
       TO_CHAR(NEXT_DATE, 'DD-MON-YYYY HH:MI:SS AM') next_start,
       INTERVAL
FROM DBA_JOBS
WHERE WHAT LIKE '%dbms_defer_sys.purge%'
ORDER BY 1;
```

出力は次のようになります。

Job ID	Privilege Schema	Broken?	Next Start	Interval
1	REPADMIN	N	02-FEB-2001 05:06:43 PM	SYSDATE + 1/24

Broken? 列の N は、ジョブが中断されていないことを意味します。したがって、ジョブは次回開始時刻に実行されます。この列の Y は、ジョブが中断されていることを意味します。

ページ・ジョブの定義のチェック

DBA\_JOBS データ・ディクショナリ・ビューの問合せにより、レプリケーション・サイトにあるページ・ジョブの定義を表示できます。次の問合せで定義が表示されます。

```
SELECT WHAT "Definition of the Purge Job"
FROM DBA_JOBS
WHERE WHAT LIKE '%dbms_defer_sys.purge%' ORDER BY 1;
```

出力は次のようになります。

```
Definition of the Purge Job
-----
declare rc binary_integer; begin rc := sys.dbms_defer_sys.purge( delay_seconds=>
0); end;
```

## 最後の削除以降の時間の長さの判定

次の問合せは、正常に伝播されたトランザクションが遅延トランザクション・キューからページされて以降の、時間の長さの合計を分単位で示します。

```
SELECT ((SYSDATE - LAST_PURGE_TIME) / 60) "Minutes Since Last Purge"
      FROM V$REPLQUEUE;
```

出力は次のようになります。

```
Minutes Since Last Purge
-----
                13.43333
```

## ページされたトランザクションの合計数の判定

次の問合せは、インスタンスが最後に開始されて以降遅延トランザクション・キューからページされた、正常に伝播されたトランザクションの合計数を示します。

```
SELECT TXNS_PURGED "Transactions Purged"
      FROM V$REPLQUEUE;
```

出力は次のようになります。

```
Transactions Purged
-----
                6541
```

## エラー・キューの監視

この項では、レプリケーション・サイトにあるエラー・キューの情報を表示するために実行できる問合せを示します。エラー・キューには接続先サイトでエラーになった遅延トランザクションが含まれます。このエラー・トランザクションは接続先サイトでエラー・キューに入れられます。

## レプリケーション・サイトにあるエラー・トランザクションの一般情報のリスト表示

次の問合せは、レプリケーション・サイトにあるエラー・トランザクションの一般情報をリストします。

```
COLUMN DEFERRED_TRAN_ID HEADING 'Deferred|Transaction|ID' FORMAT A11
COLUMN ORIGIN_TRAN_DB HEADING 'Origin|Database' FORMAT A15
COLUMN DESTINATION HEADING 'Destination|Database' FORMAT A15
COLUMN TIME_OF_ERROR HEADING 'Time of|Error' FORMAT A22
COLUMN ERROR_NUMBER HEADING 'Oracle|Error|Number' FORMAT 999999

SELECT DEFERRED_TRAN_ID,
       ORIGIN_TRAN_DB,
       DESTINATION,
       TO_CHAR(START_TIME, 'DD-Mon-YYYY hh24:mi:ss') TIME_OF_ERROR,
       ERROR_NUMBER
FROM DEFERROR ORDER BY START_TIME;
```

出力は次のようになります。

Deferred Transaction ID	Origin Database	Destination Database	Time of Error	Oracle Error Number
1.8.2470	mv4.world	NY.WORLD	25-Jan-2001 17:11:17	1403

遅延トランザクション ID と接続先データベースを使用して、エラー原因となったトランザクションの再実行またはエラーの削除ができます。

たとえば、前の例のトランザクションを再実行するには、次のように入力します。

```
EXECUTE DBMS_DEFER_SYS.EXECUTE_ERROR('1.8.2470', 'NY.WORLD');
```

前の例のエラーを削除するには、次のように入力します。

```
EXECUTE DBMS_DEFER_SYS.DELETE_ERROR('1.8.2470', 'NY.WORLD');
```

通常は、エラーを手動で解決した場合のみ、エラーを削除します。

## エラー・トランザクションの割合の判定

トランザクションをリモート・マスター・サイトに伝播するとき、正常に伝播されて適用されるトランザクションがある一方、リモート・マスター・サイトでエラーになるトランザクションがある場合があります。エラーになるトランザクションはエラー・トランザクションと呼ばれます。

次の問合せを実行して、リモート・マスター・サイト mv4.world への伝播から生じたエラー・トランザクションの割合を表示します。

```
SELECT DECODE(TOTAL_TXN_COUNT, 0, 'No Transactions',
              (TOTAL_ERROR_COUNT/TOTAL_TXN_COUNT)*100) "ERROR PERCENTAGE"
       FROM DEFSCHEDULE
       WHERE DBLINK = 'mv4.world';
```

出力は次のようになります。

```
Error Percentage
-----
          3.265
```

---

---

**注意：** この問合せが 'No transactions' を戻す場合は、統計が最後に消去されて以降、指定されたリモート・サイトに伝播されたトランザクションはありません。

---

---

## 各起点マスター・サイトからのエラー・トランザクションの数のリスト表示

この項の問合せを実行することにより、各起点マスター・サイトによるプッシュから生じたトランザクション・エラーの数を検索できます。

次の問合せを実行し、遅延トランザクションおよびエラー・トランザクションの数を確認します。

```
COLUMN SOURCE HEADING 'Origin' FORMAT A45
COLUMN ERRORS HEADING 'Def Trans Errors' FORMAT 9999

SELECT E.ORIGIN_TRAN_DB SOURCE, COUNT(*) ERRORS
       FROM DEFERROR E
       GROUP BY E.ORIGIN_TRAN_DB;
```

出力は次のようになります。

Origin	Def Trans Errors
-----	-----
NY.WORLD	1
mv4.world	3

## レプリケーション・サイトのエラー・トランザクションに関するエラー・メッセージのリスト表示

次の問合せは、レプリケーション・サイトにあるエラー・トランザクションに関するエラー・メッセージをリストします。

```
COLUMN DEFERRED_TRAN_ID HEADING 'Deferred|Transaction|ID' FORMAT A11
COLUMN ERROR_MSG HEADING 'Error Messages' FORMAT A68
```

```
SELECT DEFERRED_TRAN_ID, ERROR_MSG
FROM DEFERROR;
```

出力は次のようになります。

Deferred Transaction ID	Error Messages
-----	-----
1.8.2470	ORA-01403: データが見つかりません。

## レプリケーション・サイトのエラー操作の判定

次の問合せは、レプリケーション・サイトでエラー原因となった各コールに関して試みられた操作のタイプをリストします。

```
COLUMN CALLNO HEADING 'Call|Number' FORMAT 9999
COLUMN DEFERRED_TRAN_ID HEADING 'Deferred|Transaction|ID' FORMAT A11
COLUMN PACKAGENAME HEADING 'Package|Name' FORMAT A20
COLUMN PROCNAME HEADING 'Operation' FORMAT A15
COLUMN ORIGIN_TRAN_DB HEADING 'Origin|Database' FORMAT A15
```

```
SELECT /*+ ORDERED */
      C.CALLNO,
      C.DEFERRED_TRAN_ID,
      C.PACKAGENAME,
      C.PROCNAME, E.ORIGIN_TRAN_DB
FROM DEFERROR E, DEFCALL C
WHERE C.DEFERRED_TRAN_ID = E.DEFERRED_TRAN_ID
AND C.CALLNO = E.CALLNO
ORDER BY E.START_TIME;
```

出力は次のようになります。

Deferred Call Number	Transaction ID	Package Name	Operation	Origin Database
-----	-----	-----	-----	-----
0 1.8.2470		EMPLOYEES\$SRP	REP_UPDATE	mv4.world



## レプリケーション環境のパフォーマンスの監視

この項では、マスター・レプリケーション環境のパフォーマンスを監視するために実行できる問合せを示します。

### レプリケーション・トランザクションの行変更の平均数の追跡

次の問合せは、インスタンス起動以降のレプリケーション・トランザクションの行変更の平均数を示します。

```
SELECT DECODE(TXNS_ENQUEUED, 0, 'No Transactions Enqueued',  
              (CALLS_ENQUEUED / TXNS_ENQUEUED)) "Average Number of Row Changes"  
FROM V$REPLQUEUE;
```

出力は次のようになります。

```
Average Number of Row Changes  
-----  
                             56.16
```

---

**注意：** この問合せが 'No Transactions Enqueued' を戻す場合は、インスタンスの開始以降エンキューされたトランザクションはありません。

---

### 遅延トランザクション・キューに入るトランザクションの率の追跡

次の問合せは、インスタンス起動以降現在のサイトで遅延トランザクション・キューに入るトランザクションの、1秒当たりの平均数を示します。

```
SELECT (R.TXNS_ENQUEUED / ((SYSDATE - I.STARTUP_TIME)*24*60*60)) "Average TPS"  
FROM V$REPLQUEUE R, V$INSTANCE I;
```

出力は次のようになります。

```
Average TPS  
-----  
          150
```

# トランザクションの伝播のために作成されるネットワークの平均通信量の判定

遅延トランザクションの伝播は、ネットワークに一定の通信量を生じさせます。ここでトランザクションにより生じるネットワークの通信量とは、送受信されるバイト数と、トランザクションの伝播に必要なネットワーク・ラウンドトリップの数です。

ラウンドトリップとは、連続して受信される1つ以上のメッセージが後に続く、連続して送信される1つ以上のメッセージです。たとえば、次の場合はどちらも1つのラウンドトリップです。

- サイト A がサイト B に1つのメッセージを送信し、次にサイト B がサイト A に1つのメッセージを送信します。
- サイト A がサイト B に20のメッセージを送信し、次にサイト B がサイト A に20つのメッセージを送信します。

これは、ラウンドトリップの数とはサイト間の往復通信の数であり、ラウンドトリップの数を評価するときは、メッセージの数は無関係であることを示しています。

次の問合せは、トランザクションを mv4.world リモート・マスター・サイトに伝播するときに生じるネットワークの平均通信量を示します。

```
SELECT
  DECODE(TOTAL_TXN_COUNT, 0, 'No Transactions',
    ((TOTAL_BYTES_SENT + TOTAL_BYTES_RECEIVED) / TOTAL_TXN_COUNT)) "Average Bytes",
  DECODE(TOTAL_TXN_COUNT, 0, 'No Transactions',
    (TOTAL_ROUND_TRIPS / TOTAL_TXN_COUNT)) "Average Round Trips"
FROM DEFSCHEDULE WHERE DBLINK = 'mv4.world';
```

出力は次のようになります。

Average Bytes	Average Round Trips
-----	
69621.5	5

**注意：** この問合せが両方の列で 'No transactions' を戻す場合は、統計が最後に消去されて以降、指定されたリモート・サイトに伝播されたトランザクションはありません。

## リモート・サイトでトランザクションを適用するための平均時間の判定

平均待機時間とは、現在のサイトでのトランザクションの最初のコールと、リモート・サイトでトランザクションが適用されたことの確認の間の、平均秒数です。最初のコールは、トランザクションがコミットされるのではなく、ユーザーがデータ操作言語（DML）の最初の変更を行うときに開始されます。

次の問合せは、トランザクションをリモート・マスター・サイト mv4.world で適用するための平均待機時間を示します。

```
SELECT AVG_LATENCY "Average Latency"
FROM DEFSCHEDULE
WHERE DBLINK='mv4.world';
```

出力は次のようになります。

```
Average Latency
-----
                25.5
```

## パラレル伝播ジョブがスリープに費やす時間の割合の判定

パラレル伝播コーディネータがアクティブでないことを、スリープしているといいます。DBMS\_DEFER\_SYS.PUSH プロシージャの delay\_seconds パラメータを使用して、伝播コーディネータがスリープする時間を制御します。

次の問合せは、トランザクションを mv4.world リモート・マスター・サイトに伝播するとき、パラレル伝播コーディネータがスリープに費やす時間の割合を示します。

```
SELECT DECODE(AVG_THROUGHPUT, 0, NULL,
              ((TOTAL_SLEEP_TIME / (TOTAL_TXN_COUNT / AVG_THROUGHPUT)) * 100))
       "Percent Sleep Time"
FROM DEFSCHEDULE WHERE DBLINK = 'mv4.world';
```

出力は次のようになります。

```
Percent Sleep Time
-----
                2
```

当然このケースでは、パラレル伝播コーディネータは 98% の時間アクティブです。

---

**注意：** この問合せが NULL を戻す場合は、統計が最後に消去されて以降、または最後のデータベース起動以降、指定されたリモート・サイトに伝播されたトランザクションはありません。

---

## DEFSCHEDULE ビューのリモート・マスター・サイトに関する統計の消去

特定のリモート・マスター・サイトに関する DEFSCHEDULE ビューの伝播統計を消去するには、DBMS\_DEFER\_SYS パッケージの CLEAR\_PROP\_STATISTICS プロシージャを使用します。たとえば、mv4.world リモート・マスター・サイトに関する伝播統計を消去するには、次のプロシージャを実行します。

```
BEGIN
  DBMS_DEFER_SYS.CLEAR_PROP_STATISTICS (
    dblink => 'mv4.world');
END;
/
```

## V\$REPLPROP を使用した遅延トランザクションの平行伝播の監視

V\$REPLPROP 動的パフォーマンス・ビューは、現在の平行伝播セッションの情報を提供します。

---

---

**注意：** V\$REPLPROP 動的パフォーマンス・ビューは、遅延トランザクションの平行伝播を使用している場合にのみ関係します。シリアル伝播を使用している場合は、このビューは空です。

---

---

### 遅延トランザクションを伝播するデータベースの判定

次の問合せを実行して、現在平行伝播を使用して遅延トランザクションを伝播している各データベースの、データベース・リンクをリストします。

```
SELECT DBLINK "Database Link"
FROM V$REPLPROP
WHERE NAME LIKE '%Coordinator%';
```

出力は次のようになります。

```
Database Link
-----
mv4.world
NY.WORLD
HK.WORLD
```

## 現在リモート・マスターに伝播されているトランザクションの判定

現在パラレル伝播を使用して特定のリモート・マスター・サイトに伝播されているトランザクションの情報に関しては、次の情報をリストできます。

- 各トランザクションのトランザクション識別番号。
- 各トランザクションのコールの数。
- 各トランザクションの処理済のコールの割合。この列の数値は、トランザクションのコールが処理されるに従って増加します。数値が 100 になったときは、すべてのコールが処理済です。

次の問合せでこの情報が表示されます。

```
SELECT /*+ ORDERED */ P.XID "Tran Being Propagated",
      (MAX(C.CALLNO) + 1) "Number of Calls in Tran",
      (P.SEQUENCE/MAX(C.CALLNO) + 1) * 100 "% Processed Calls"
FROM V$REPLPROP P, DEFCALL C
WHERE P.NAME LIKE '%SLAVE%'
AND P.DBLINK = 'mv4.world'
AND C.DEFERRED_TRAN_ID = P.XID
GROUP BY P.XID, P.SEQUENCE;
```

出力は次のようになります。

Tran Being Propagated	Number of Calls in Tran	% Processed Calls
1.11.4264	43357	78
1.15.4256	23554	49

既存のトランザクションがプッシュされて新しいトランザクションが処理されると同時に、トランザクション識別番号は変更される必要があります。この問合せは、レプリケーション環境に次の条件のいずれかが該当する場合に特に役立ちます。

- 多数のトランザクションを定期的にプッシュする場合。
- 非常に大きなトランザクションがいくつかある場合。
- 非同期伝播を使用した連続的なプッシュのシミュレーションを行う場合。

レプリケーション環境に最初の 2 つの条件が該当する場合は、この問合せを実行して、スレーブ・プロセスがトランザクションをプッシュしているかどうかをチェックできます。このタイプの環境では、トランザクションをプッシュしていないときはスレーブ・プロセスは存在しません。

連続的な送信のシミュレーションを行っているレプリケーション環境では、遅延トランザクション・キューにプッシュすべきトランザクションがあるときは必ずスレーブ・プロセスが存在します。プッシュすべきトランザクションが存在しないときは、スレーブ・プロセスが存在しない場合があります。このため、プッシュすべきトランザクションがあるときは、こ

の問合せを実行して、スレーブ・プロセスが存在してトランザクションを処理しているかどうかを確認できます。

**関連項目：** レプリケーション環境での連続的なプッシュのスケジューリングの詳細は、『Oracle9i アドバンスド・レプリケーション』を参照してください。

# 第 III 部

---

## レプリケーション・マネージメント API パッケージ・リファレンス

第 III 部には、レプリケーション・マネージメント API に関する次のリファレンス情報が含まれます。

- 各パッケージのプロシージャおよびファンクション
- パッケージ化されたプロシージャやファンクションのパラメータ
- プロシージャやファンクションで発生する可能性のある例外

---

**注意：** 第 III 部の章で説明する PL/SQL プロシージャおよびファンクションの一部は、オーバーロードされています。すなわち、1つのパッケージ内に、同一の名前のプロシージャまたはファンクションが複数ありますが、それらの仮パラメータの個数、順序またはデータ型のグループは異なります。オーバーロードされているプロシージャまたはファンクションは、その旨を記述します。オーバーロードおよび PL/SQL の一般情報の詳細は、『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。

---

## PL/SQL パッケージ

Oracle のレプリケーション・マネージメント API には次の PL/SQL パッケージがあります。

- [DBMS\\_DEFER](#)
- [DBMS\\_DEFER\\_QUERY](#)
- [DBMS\\_DEFER\\_SYS](#)
- [DBMS\\_MVIEW](#)
- [DBMS\\_OFFLINE\\_OG](#)
- [DBMS\\_OFFLINE\\_SNAPSHOT](#)
- [DBMS\\_RECTIFIER\\_DIFF](#)
- [DBMS\\_REFRESH](#)
- [DBMS\\_REPCAT](#)
- [DBMS\\_REPCAT\\_ADMIN](#)
- [DBMS\\_REPCAT\\_INSTANTIATE](#)
- [DBMS\\_REPCAT\\_RGT](#)
- [DBMS\\_REPUTIL](#)



---

## レプリケーション・マネージメント API リファレンスの概要

Advanced Replication をインストールすると、レプリケーション・マネージメント Application Program Interface (API) もインストールされます。**レプリケーション・マネージメント API** は、管理者が各サイトのレプリケーション機能を設定および管理するための、PL/SQL パッケージのコレクションです。また、Oracle Enterprise Manager のレプリケーション・マネージメント・ツールでも、作業の実行に各サイトのレプリケーション・マネージメント API のプロシージャやファンクションが使用されます。

この章では、次の項目を説明します。

- [Oracle レプリケーション・マネージメント API の使用例](#)
- [レプリケーション・マネージメント API を使用する際の考慮事項](#)
- [レプリケーション・マネージメント・ツールとレプリケーション・マネージメント API](#)
- [日時データ型および期間データ型の略称](#)

---

**注意：** 第 III 部の章で説明する PL/SQL プロシージャおよびファンクションの一部は、オーバーロードされています。すなわち、1つのパッケージ内に、同一の名前のプロシージャまたはファンクションが複数ありますが、それらの仮パラメータの個数、順序またはデータ型のグループは異なります。オーバーロードされているプロシージャまたはファンクションは、その旨を記述します。オーバーロードおよび PL/SQL の一般情報の詳細は、『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。

---

## Oracle レプリケーション・マネージメント API の使用例

Oracle のレプリケーション・マネージメント API を使用するには、SQL\*Plus や Enterprise Manager SQL Worksheet などの問合せツールを使用して、プロシージャまたはファンクションをコールします。たとえば、次に示す DBMS\_REPCAT.CREATE\_MASTER\_REPOBJECT プロシージャをコールすると、hr\_repg レプリケーション・グループ内に新しいレプリケート表 hr.employees が作成されます。

```
BEGIN
  DBMS_REPCAT.CREATE_MASTER_REPOBJECT (
    gname => 'hr_repg',
    type => 'TABLE',
    oname => 'employees',
    sname => 'hr',
    use_existing_object => TRUE,
    copy_rows => FALSE);
END;
/
```

レプリケーション・マネージメント API ファンクションをコールするには、ファンクションの戻り値を受け取る環境を用意する必要があります。たとえば、次に示す無名 PL/SQL ブロックは、DBMS\_DEFER\_SYS.DISABLED ファンクションを IF 文でコールします。

```
BEGIN
  IF DBMS_DEFER_SYS.DISABLED ('inst2') THEN
    DBMS_OUTPUT.PUT_LINE('Propagation to INST2 is disabled.');
```

ELSE

```
    DBMS_OUTPUT.PUT_LINE('Propagation to INST2 is enabled.');
```

END IF;

```
END;
/
```

## レプリケーション・マネージメント API を使用する際の考慮事項

レプリケーション・マネージメント API のプロシージャおよびファンクションの多くには、いくつかの重要な考慮事項があります。次に例を示します。

- プロシージャやファンクションのいくつかは、マルチマスター構成のマスター定義サイトからのみコールします。
- マスター・グループに対して特定の管理操作を行う場合は、レプリケーション・マネージメント API のプロシージャやファンクションをコールする前に、そのグループのレプリケーション・アクティビティを中断する必要があります。
- Oracle レプリケーション・マネージメント API のプロシージャやファンクションをコールする順序は大変重要です。レプリケーション・マネージメント・コールを発行する方法の詳細は、次の項を参照してください。

## レプリケーション・マネージメント・ツールとレプリケーション・マネージメント API

レプリケーション・マネージメント・ツールでは、ほとんどのファンクションの実行に、レプリケーション・マネージメント API が使用されます。レプリケーション・マネージメント・ツールには次に示す機能が備わっているため、レプリケーション・マネージメント API コールを個別に発行するより便利です。

- API コールのパラメータの入力および調整を行う GUI インタフェースを用意
- 大量の関連 API コールを、適切に順序付け
- API コールで戻された出力を、メッセージ・ボックスとエラー・ファイルに表示

レプリケーション・マネージメント・ツールのスクリプト機能を使用すると、Oracle レプリケーション・マネージメント API の使用方法を簡単に理解できます。レプリケーション・マネージメント・ツールで管理セッションを開始するときに、スクリプトをオンにします。作業を終了した後、スクリプトをオフにしてスクリプト・ファイルを確認します。スクリプト・ファイルには、セッション中に発行されたすべてのレプリケーション・マネージメント API コールが保存されます。レプリケーション・マネージメント・ツールのスクリプト機能の詳細は、レプリケーション・マネージメント・ツールのヘルプを参照してください。

## 日時データ型および期間データ型の略称

日時データ型および期間データ型の多くの名前は、レプリケーション・マネージメント API のプロシージャおよびファンクションとともに使用するには長すぎます。このため、これらのデータ型に関してはフルネームのかわりに略称を使用する必要があります。各データ型およびその略称を次の表に示します。DATE および TIMESTAMP データ型には略称は必要ありません。

データ型	略称
TIMESTAMP WITH TIME ZONE	TSTZ
TIMESTAMP LOCAL TIME ZONE	TSLTZ
INTERVAL YEAR TO MONTH	IYM
INTERVAL DAY TO SECOND	IDS

たとえば、DBMS\_DEFER\_QUERY.GET\_datatype\_ARG ファンクションを使用して遅延コールの TIMESTAMP LOCAL TIME ZONE 引数を判定する場合は、データ型のかわりに TSLTZ を使用します。したがって、DBMS\_DEFER\_QUERY.GET\_TSLTZ\_ARG ファンクションを実行します。

DBMS\_DEFER は、レプリケート・トランザクション遅延リモート・プロシージャ・コール機能のユーザー・インタフェースです。レプリケート・アプリケーションは、このインタフェースのコールを使用して、リモート・ノードでの後のトランザクション実行のためにプロシージャ・コールをキューに入れます。

通常こうしたプロシージャは、行トリガーの後、またはアプリケーション固有の更新プロシージャの後でコールされます。

この章では、次の項目を説明します。

- [DBMS\\_DEFER サブプログラムの概要](#)

## DBMS\_DEFER サブプログラムの概要

表 12-1 DBMS\_DEFER パッケージのサブプログラム

サブプログラム	説明
12-2 ページ「CALL プロシージャ」	リモート・プロシージャへの遅延コールを作成します。
12-4 ページ「COMMIT_WORK プロシージャ」	正しく定義された遅延リモート・プロシージャ・コールのチェック後に、トランザクションのコミットを実行します。
12-5 ページ「datatype_ARG プロシージャ」	遅延リモート・プロシージャ・コールに渡されるデータを提供します。
12-7 ページ「TRANSACTION プロシージャ」	新しい遅延トランザクションの開始を示します。

### CALL プロシージャ

このプロシージャは、リモート・プロシージャへの遅延コールを作成します。

### 構文

```
DBMS_DEFER.CALL (
    schema_name      IN  VARCHAR2,
    package_name     IN  VARCHAR2,
    proc_name        IN  VARCHAR2,
    arg_count         IN  NATURAL,
    { nodes           IN  node_list_t
    | group_name      IN  VARCHAR2 := ''});
```

**注意：** このプロシージャはオーバーロードされています。nodes パラメータと group\_name パラメータは、同時には指定できません。

## パラメータ

表 12-2 CALL プロシージャのパラメータ

パラメータ	説明
schema_name	ストアド・プロシージャが置かれたスキーマの名前です。
package_name	ストアド・プロシージャを含むパッケージの名前です。ストアド・プロシージャはパッケージの一部になります。スタンドアロン・プロシージャへの遅延コールはサポートされていません。
proc_name	コールを遅らせる、リモート・プロシージャの名前です。
arg_count	プロシージャのパラメータ数です。各パラメータにつき 1 回、DBMS_DEFER.datatype_ARG へのコールが必要です。 <b>注意：</b> パラメータの一部がデフォルトである場合でも、プロシージャのパラメータをすべて数に含める必要があります。
nodes	遅延コールを伝播する、完全修飾されたデータベース名の PL/SQL 索引付き表です。表の索引付けは、1 の位置から始まり、NULL エントリが見つかるまで、または no_data_found 例外が発生するまで続行されます。表内のデータには、大文字と小文字の区別がありません。このパラメータはオプションです。
group_name	内部使用に予約されています。

## 例外

表 12-3 CALL プロシージャの例外

例外	説明
ORA-23304 (malformedcall)	引数の数が正しくありません。
ORA-23319	パラメータ値が無効です
ORA-23352	nodes、または前の DBMS_DEFER.TRANSACTION コールで指定された接続先リストに重複があります。

# COMMIT\_WORK プロシージャ

このプロシージャは、正しく定義された遅延リモート・プロシージャ・コールのチェック後に、トランザクションのコミットを実行します。

## 構文

```
DBMS_DEFER.COMMIT_WORK (  
    commit_work_comment IN VARCHAR2);
```

## パラメータ

表 12-4 COMMIT\_WORK プロシージャのパラメータ

パラメータ	説明
commit_work_comment	SQL の COMMIT COMMENT 文に相当します。

## 例外

表 12-5 COMMIT\_WORK プロシージャの例外

例外	説明
ORA-23304 (malformedcall)	トランザクションの形式が正しくなかったか、またはトランザクションが正常に終了しませんでした。



## ***datatype\_ARG* プロシージャ**

このプロシージャは、遅延リモート・プロシージャ・コールに渡すデータを提供します。遅延リモート・プロシージャに渡す必要があるデータ型に応じて、次に示すプロシージャのいずれかをコールする必要があります。

DBMS\_DEFER.CALL を実行した後、プロシージャの各パラメータを、*datatype\_ARG* プロシージャを使用して指定する必要があります。すなわち、遅延リモート・プロシージャ・コールにはデフォルト・パラメータを使用できません。たとえば、次のようなプロシージャがあります。

```
CREATE OR REPLACE PACKAGE my_pack AS
    PROCEDURE my_proc(a VARCHAR2, b VARCHAR2 DEFAULT 'SALES');
END;
/
```

DBMS\_DEFER.CALL プロシージャを実行するときは、パラメータごとに個別のプロシージャ・コールを *my\_proc* プロシージャに組み込む必要があります。

```
CREATE OR REPLACE PROCEDURE load_def_tx IS
    node DBMS_DEFER.NODE_LIST_T;
BEGIN
    node(1) := 'MYCOMPUTER.WORLD';
    node(2) := NULL;
    DBMS_DEFER.TRANSACTION(node);
    DBMS_DEFER.CALL('PR', 'MY_PACK', 'MY_PROC', 2);
    DBMS_DEFER.VARCHAR2_ARG('TEST');
    DBMS_DEFER.VARCHAR2_ARG('SALES'); -- required, cannot omit to use default
END;
```

---

---

### **注意：**

- AnyData\_ARG プロシージャは、ユーザー定義型のオブジェクト型、コレクションおよび REF をサポートします。AnyData データ型の詳細は、『Oracle9i SQL リファレンス』および『Oracle9i アプリケーション開発者ガイド - オブジェクト・リレーショナル機能』を参照してください。
  - このプロシージャは日時データ型および期間データ型の略称を使用します。たとえば、TIMESTAMP WITH TIME ZONE データ型には TSTZ が使用されます。これらの略称の詳細は、11-4 ページの「[日時データ型および期間データ型の略称](#)」を参照してください。
- 
-

構文

```
DBMS_DEFER.AnyData_ARG      (arg IN SYS.AnyData);
DBMS_DEFER.NUMBER_ARG       (arg IN NUMBER);
DBMS_DEFER.DATE_ARG         (arg IN DATE);
DBMS_DEFER.VARCHAR2_ARG     (arg IN VARCHAR2);
DBMS_DEFER.CHAR_ARG         (arg IN CHAR);
DBMS_DEFER.ROWID_ARG        (arg IN ROWID);
DBMS_DEFER.RAW_ARG          (arg IN RAW);
DBMS_DEFER.BLOB_ARG         (arg IN BLOB);
DBMS_DEFER.CLOB_ARG         (arg IN CLOB);
DBMS_DEFER.NCLOB_ARG        (arg IN NCLOB);
DBMS_DEFER.NCHAR_ARG        (arg IN NCHAR);
DBMS_DEFER.NVARCHAR2_ARG    (arg IN NVARCHAR2);
DBMS_DEFER.ANY_CLOB_ARG     (arg IN CLOB);
DBMS_DEFER.ANY_VARCHAR2_ARG (arg IN VARCHAR2);
DBMS_DEFER.ANY_CHAR_ARG     (arg IN CHAR);
DBMS_DEFER.IDS_ARG          (arg IN DSINTERVAL_UNCONSTRAINED);
DBMS_DEFER.IYM_ARG          (arg IN YMINTERVAL_UNCONSTRAINED);
DBMS_DEFER.TIMESTAMP_ARG    (arg IN TIMESTAMP_UNCONSTRAINED);
DBMS_DEFER.TSLTZ_ARG        (arg IN TIMESTAMP_LTZ_UNCONSTRAINED);
DBMS_DEFER.TSTZ_ARG         (arg IN TIMESTAMP_TZ_UNCONSTRAINED);
```

パラメータ

表 12-6 datatype\_ARG プロシージャのパラメータ

パラメータ	説明
arg	以前コールを遅延したリモート・プロシージャに渡すパラメータ値です。

例外

表 12-7 datatype\_ARG プロシージャの例外

例外	説明
ORA-23323	引数値が長すぎます。

# TRANSACTION プロシージャ

このプロシージャは、新しい遅延トランザクションの開始を示します。このコールを省略した場合、DBMS\_DEFER.CALL への最初のコールが、新しいトランザクションの開始と見なされます。

## 構文

```
DBMS_DEFER.TRANSACTION (  
    nodes IN    node_list_t);
```

**注意：** このプロシージャはオーバーロードされています。入力パラメータがない場合の動作は、入力パラメータがある場合の動作と似ています。ただし、前者では、ノード・パラメータのノードではなく、DEFDEFAULTDEST ビューの nodes を使用します。

## パラメータ

表 12-8 TRANSACTION プロシージャのパラメータ

パラメータ	説明
nodes	トランザクションの遅延コールを伝播する完全修飾されたデータベース名の、PL/SQL 索引付き表です。表の索引付けは、1 の位置から始まり、NULL エントリが見つかるまで、または no_data_found 例外が発生するまで続行されます。表内のデータには、大文字と小文字の区別がありません。

## 例外

表 12-9 TRANSACTION プロシージャの例外

例外	説明
ORA-23304 (malformedcall)	前のトランザクションの形式が正しくなかったか、または前のトランザクションが正常に終了しませんでした。
ORA-23319	パラメータ値が無効です
ORA-23352	ノード・リストに重複が存在するとき、DBMS_DEFER.CALL によって発生します。



---

## DBMS\_DEFER\_QUERY

DBMS\_DEFER\_QUERY は、ビューを通じて公開されていない遅延トランザクション・キュー・データの間合せを可能にします。

この章では、次の項目を説明します。

- [DBMS\\_DEFER\\_QUERY サブプログラムの概要](#)

## DBMS\_DEFER\_QUERY サブプログラムの概要

表 13-1 DBMS\_DEFER\_QUERY パッケージのサブプログラム

サブプログラム	説明
13-2 ページ「 <a href="#">GET_ARG_FORM</a> ファンクション」	遅延コールの引数の形式を判定します。
13-4 ページ「 <a href="#">GET_ARG_TYPE</a> ファンクション」	遅延コールの引数のタイプを判定します。
13-6 ページ「 <a href="#">GET_CALL_ARGS</a> プロシージャ」	指定されたコールの様々な引数のテキスト・バージョンを戻します。
13-7 ページ「 <a href="#">GET_datatype_ARG</a> ファンクション」	遅延コールの引数値を判定します。
13-9 ページ「 <a href="#">GET_OBJECT_NULL_VECTOR_ARG</a> ファンクション」	列オブジェクトのタイプ情報を戻します。

### GET\_ARG\_FORM ファンクション

このファンクションは、遅延コール・パラメータのキャラクタ・セット・フォームを戻します。

**関連項目：** レプリケーション・マネージメント・ツールでの遅延トランザクションおよびエラー・トランザクションの表示の詳細は、レプリケーション・マネージメント・ツールのオンライン・ヘルプを参照してください。

### 構文

```
DBMS_DEFER_QUERY.GET_ARG_FORM (  
  callno          IN  NUMBER,  
  arg_no          IN  NUMBER,  
  deferred_tran_id IN  VARCHAR2)  
RETURN NUMBER;
```

## パラメータ

表 13-2 GET\_ARG\_FORM ファンクションのパラメータ

パラメータ	説明
callno	DEFCALL ビューからのコール識別子です。
arg_no	コールの引数リスト内のパラメータの位置です。指定できるパラメータの位置は、1 からコール内のパラメータの数までです。
deferred_tran_id	遅延トランザクションの識別子です。

## 例外

表 13-3 GET\_ARG\_FORM ファンクションの例外

例外	説明
NO_DATA_FOUND	入力パラメータが遅延コールのパラメータに対応しません。

## 戻り値

表 13-4 GET\_ARG\_FORM ファンクションの戻り値

定数戻り値	戻り値	可能なデータ型
DBMS_DEFER_QUERY.ARG_FORM_NONE	0	DATE NUMBER ROWID RAW BLOB User-defined types
DBMS_DEFER_QUERY.ARG_FORM_IMPLICIT	1	CHAR VARCHAR2 CLOB
DBMS_DEFER_QUERY.ARG_FORM_NCHAR	2	NCHAR NVARCHAR2 NCLOB

## GET\_ARG\_TYPE ファンクション

このファンクションは、遅延コールの引数のタイプを判定します。遅延リモート・プロシージャ・コール（Remote Procedure Call: RPC）パラメータのタイプが戻されます。

**関連項目：** レプリケーション・マネージメント・ツールでの遅延トランザクションおよびエラー・トランザクションの表示の詳細は、レプリケーション・マネージメント・ツールのオンライン・ヘルプを参照してください。

### 構文

```
DBMS_DEFER_QUERY.GET_ARG_TYPE (  
  callno           IN   NUMBER,  
  arg_no           IN   NUMBER,  
  deferred_tran_id IN   VARCHAR2)  
RETURN NUMBER;
```

### パラメータ

表 13-5 GET\_ARG\_TYPE ファンクションのパラメータ

パラメータ	説明
callno	遅延リモート・プロシージャ・コールの DEFSCALL ビューから渡される識別番号です。
arg_no	タイプを判定するコールへの引数の数値的な位置です。プロシージャへの最初の引数は、1 の位置にあります。
deferred_tran_id	遅延トランザクションの識別子です。

### 例外

表 13-6 GET\_ARG\_TYPE ファンクションの例外

例外	説明
NO_DATA_FOUND	入力パラメータが遅延コールのパラメータに対応しません。



# 戻り値

表 13-7 GET\_ARG\_TYPE ファンクションの戻り値

定数戻り値	戻り値	対応するデータ型
DBMS_DEFER_QUERY.ARG_TYPE_VARCHAR2	1	VARCHAR2
DBMS_DEFER_QUERY.ARG_TYPE_NUM	2	NUMBER
DBMS_DEFER_QUERY.ARG_TYPE_ROWID	11	ROWID
DBMS_DEFER_QUERY.ARG_TYPE_DATE	12	DATE
DBMS_DEFER_QUERY.ARG_TYPE_RAW	23	RAW
DBMS_DEFER_QUERY.ARG_TYPE_CHAR	96	CHAR
DBMS_DEFER_QUERY.ARG_TYPE_AnyData	109	AnyData
DBMS_DEFER_QUERY.ARG_TYPE_CLOB	112	CLOB
DBMS_DEFER_QUERY.ARG_TYPE_BLOB	113	BLOB
DBMS_DEFER_QUERY.ARG_TYPE_BFIL	114	BFILE
DBMS_DEFER_QUERY.ARG_TYPE_OBJECT_NULL_VECTOR	121	OBJECT_NULL_VECTOR
DBMS_DEFER_QUERY.ARG_TYPE_TIMESTAMP	180	TIMESTAMP
DBMS_DEFER_QUERY.ARG_TYPE_TSTZ	181	TSTZ
DBMS_DEFER_QUERY.ARG_TYPE_IYM	182	IYM
DBMS_DEFER_QUERY.ARG_TYPE_IDS	183	IDS
DBMS_DEFER_QUERY.ARG_TYPE_TSLTZ	231	TSLTZ

## 注意：

- AnyData データ型は、ユーザー定義型のオブジェクト型、コレクションおよび REF をサポートします。AnyData データ型の詳細は、『Oracle9i SQL リファレンス』および『Oracle9i アプリケーション開発者ガイド - オブジェクト・リレーショナル機能』を参照してください。
- このファンクションは日時データ型および期間データ型の略称を使用します。たとえば、TIMESTAMP WITH TIME ZONE データ型には TSTZ が使用されます。これらの略称の詳細は、11-4 ページの「[日時データ型および期間データ型の略称](#)」を参照してください。

## GET\_CALL\_ARGS プロシージャ

このプロシージャは、指定されたコールの様々な引数のテキスト・バージョンを戻します。  
テキスト・バージョンは最初の 2000 バイトに制限されます。

**関連項目：**

- 13-7 ページ「[GET\\_datatype\\_ARG ファンクション](#)」
- AnyData データ型の詳細は、『Oracle9i SQL リファレンス』および『Oracle9i アプリケーション開発者ガイド - オブジェクト・リレーションアル機能』を参照してください。

### 構文

```
DBMS_DEFER_QUERY.GET_CALL_ARGS (  
    callno      IN  NUMBER,  
    startarg    IN  NUMBER := 1,  
    argcnt      IN  NUMBER,  
    argsize     IN  NUMBER,  
    tran_id     IN  VARCHAR2,  
    date_fmt    IN  VARCHAR2,  
    types       OUT TYPE_ARY,  
    forms       OUT TYPE_ARY,  
    vals        OUT VAL_ARY);
```

### パラメータ

表 13-8 GET\_CALL\_ARGS プロシージャのパラメータ

パラメータ	説明
callno	遅延リモート・プロシージャ・コール (RPC) の DEFCALL ビューから渡される識別番号です。
startarg	記述される最初の引数の数値的な位置です。
argcnt	コール内の引数の数です。
argsize	戻される引数の最大サイズです。
tran_id	遅延トランザクションの識別子です。
date_fmt	戻される日付の形式です。
types	引数のタイプを含む配列です。
forms	引数のキャラクタ・セット形式を含む配列です。
vals	テキスト形式の引数値を含む配列です。

例外

表 13-9 GET\_CALL\_ARGS プロシージャの例外

例外	説明
NO_DATA_FOUND	入力パラメータが遅延コールのパラメータに対応しません。

GET\_datatype\_ARG ファンクション

このファンクションは、遅延コールの引数値を判定します。

AnyData 型は、ユーザー定義型のオブジェクト型、コレクションおよび REF をサポートします。このファンクションがサポートする型の中には、DBMS\_DEFER パッケージの AnyData\_ARG プロシージャがエンキューできないものもあります。

型引数に戻されるテキストには、型所有者、型名、型バージョン、長さ、精度、スケール、キャラクタ・セット識別子、キャラクタ・セット・フォームおよびコレクションの要素数またはオブジェクト型の属性数の値があります。これらの値はコロン (:) で区切られます。

関連項目：

- 12-5 ページ「[datatype\\_ARG プロシージャ](#)」
- レプリケーション・マネージメント・ツールでの遅延トランザクションおよびエラー・トランザクションの表示の詳細は、レプリケーション・マネージメント・ツールのオンライン・ヘルプを参照してください。
- AnyData データ型の詳細は、『Oracle9i SQL リファレンス』および『Oracle9i アプリケーション開発者ガイド - オブジェクト・リレーショナル機能』を参照してください。
- このファンクションは日時データ型および期間データ型の略称を使用します。たとえば、TIMESTAMP WITH TIME ZONE データ型には TSTZ が使用されます。これらの略称の詳細は、11-4 ページの「[日時データ型および期間データ型の略称](#)」を参照してください。

構文

取得する引数値のタイプに応じて、該当するファンクションの構文は次のとおりです。これらのファンクションでは、指定された引数の値が戻されます。

```
DBMS_DEFER_QUERY.GET_datatype_ARG (  
    callno           IN    NUMBER,  
    arg_no           IN    NUMBER,  
    deferred_tran_id IN    VARCHAR2 DEFAULT NULL)  
RETURN datatype;
```

*datatype* は次のとおりです。

```
{ AnyData  
| NUMBER  
| VARCHAR2  
| CHAR  
| DATE  
| RAW  
| ROWID  
| BLOB  
| CLOB  
| NCLOB  
| NCHAR  
| NVARCHAR2  
| IDS  
| IYM  
| TIMESTAMP  
| TSLTZ  
| TSTZ }
```

パラメータ

表 13-10 GET\_datatype\_ARG ファンクションのパラメータ

パラメータ	説明
callno	遅延リモート・プロシージャ・コールの DEFCALL ビューから渡される識別番号です。
arg_no	値を判定するコールへの引数の数値的な位置です。プロシージャへの最初の引数は、1 の位置にあります。
deferred_tran_id	遅延トランザクションの識別子です。GET_ARG_TYPE ファンクションに渡された、直前のトランザクション識別子が設定されます。デフォルトでは NULL です。

## 例外

表 13-11 GET\_datatype\_ARG ファンクションの例外

例外	説明
NO_DATA_FOUND	入力パラメータが遅延コールのパラメータに対応しません。
ORA-26564	入力パラメータが、指定された型でも AnyData 型がサポートする型でもありません。

## GET\_OBJECT\_NULL\_VECTOR\_ARG ファンクション

このファンクションは、型所有者、名前およびハッシュコードを含む列オブジェクトのタイプ情報を戻します。

## 構文

```
DBMS_DEFER_QUERY.GET_OBJECT_NULL_VECTOR_ARG (  
    callno           IN    NUMBER,  
    arg_no           IN    NUMBER,  
    deferred_tran_id IN    VARCHAR2)  
RETURN SYSTEM.REPCAT$_OBJECT_NULL_VECTOR;
```

## パラメータ

表 13-12 GET\_OBJECT\_NULL\_VECTOR\_ARG ファンクション・パラメータ

パラメータ	説明
callno	DEFCALL ビューからのコール識別子です。
arg_no	コールの引数リスト内のパラメータの位置です。指定できるパラメータの位置は、1 からコール内のパラメータの数までです。
deferred_tran_id	遅延トランザクションの識別子です。

## 例外

表 13-13 GET\_OBJECT\_NULL\_VECTOR\_ARG ファンクションの例外

例外	説明
NO_DATA_FOUND	入力パラメータが遅延コールのパラメータに対応しません。
ORA-26564	パラメータが object_null_vector 型ではありません。

戻り値

表 13-14 GET\_OBJECT\_NULL\_VECTOR\_ARG ファンクションの戻り値

戻り値	型定義
SYSTEM.REPCAT\$_OBJECT_NULL_VECTOR <small>型</small>	<pre>CREATE TYPE SYSTEM.REPCAT\$_OBJECT_NULL_VECTOR AS OBJECT (     type_owner      VARCHAR2(30),     type_name       VARCHAR2(30),     type_hashcode   RAW(17),     null_vector     RAW(2000));</pre>

---

## DBMS\_DEFER\_SYS

DBMS\_DEFER\_SYS プロシージャは、デフォルト・レプリケーション・ノード・リストを管理します。このパッケージは、レプリケート・トランザクション遅延リモート・プロシージャ・コール機能のシステム管理者インタフェースです。管理者およびレプリケーション・デーモンは、この機能を使用して、リモート・ノードへのキューに入れられているトランザクションを実行できます。また管理者は、リモート・コールの接続先のノードを制御できます。

この章では、次の項目を説明します。

- [DBMS\\_DEFER\\_SYS サブプログラムの概要](#)

## DBMS\_DEFER\_SYS サブプログラムの概要

表 14-1 DBMS\_DEFER\_SYS パッケージのサブプログラム

サブプログラム	説明
14-4 ページ「ADD_DEFAULT_DEST プロシージャ」	接続先データベースを DEFDEFAULTDEST ビューに追加します。
14-5 ページ「CLEAR_PROP_STATISTICS プロシージャ」	DEFSCHEDULE データ・ディクショナリ・ビューの伝播統計を消去します。
14-6 ページ「DELETE_DEFAULT_DEST プロシージャ」	接続先データベースを DEFDEFAULTDEST ビューから削除します。
14-6 ページ「DELETE_DEF_DESTINATION プロシージャ」	接続先データベースを DEFSCHEDULE ビューから削除します。
14-7 ページ「DELETE_ERROR プロシージャ」	トランザクションを DEFERROR ビューから削除します。
14-7 ページ「DELETE_TRAN プロシージャ」	トランザクションを DEFTRANDEST ビューから削除します。
14-8 ページ「DISABLED ファンクション」	カレント・サイトから指定されたサイトへの、遅延トランザクション・キューの伝播が有効になっているかどうかを判定します。
14-9 ページ EXCLUDE_PUSH ファンクション	遅延トランザクション PUSH を防ぐ排他ロックを取得します。
14-10 ページ「EXECUTE_ERROR プロシージャ」	正常に終了しなかった遅延トランザクションを、トランザクションの元の受信者のセキュリティ・コンテキストで再実行します。
14-11 ページ「EXECUTE_ERROR_AS_USER プロシージャ」	正常に終了しなかった遅延トランザクションを、このプロシージャを実行するユーザーのセキュリティ・コンテキストで再実行します。
14-12 ページ「PURGE ファンクション」	カレント・マスター・サイトまたはマテリアライズド・ビュー・サイトの遅延トランザクション・キューから、プッシュされたトランザクションをパージします。
14-14 ページ「PUSH ファンクション」	カレント・マスター・サイトまたはマテリアライズド・ビュー・サイトの遅延リモート・プロシージャ・コール・キューを、強制的にリモート・サイトへプッシュします。
14-17 ページ「REGISTER_PROPAGATOR プロシージャ」	指定されたユーザーを、ローカル・データベースのプロパゲータとして登録します。



表 14-1 DBMS\_DEFER\_SYS パッケージのサブプログラム（続き）

サブプログラム	説明
14-18 ページ「 <a href="#">SCHEDULE_PURGE</a> プロシージャ」	カレント・マスター・サイトまたはマテリアライズド・ビュー・サイトの遅延トランザクション・キューから、プッシュされたトランザクションをパージするジョブをスケジュールします。
14-20 ページ「 <a href="#">SCHEDULE_PUSH</a> プロシージャ」	遅延トランザクション・キューをリモート・サイトにプッシュするジョブをスケジュールします。
14-22 ページ「 <a href="#">SET_DISABLED</a> プロシージャ」	カレント・サイトから、指定された接続先サイトへの遅延トランザクション・キューの伝播を、使用禁止または使用可能にします。
14-24 ページ「 <a href="#">UNREGISTER_PROPAGATOR</a> プロシージャ」	プロパゲータとしてのユーザーの登録を、ローカル・データベースから削除します。
14-25 ページ「 <a href="#">UNSCHEDULE_PURGE</a> プロシージャ」	マスター・サイトまたはマテリアライズド・ビュー・サイトの遅延トランザクション・キューからの、プッシュされたトランザクションの自動パージを中止します。
14-25 ページ「 <a href="#">UNSCHEDULE_PUSH</a> プロシージャ」	マスター・サイトまたはマテリアライズド・ビュー・サイトからリモート・サイトへの、遅延トランザクション・キューの自動プッシュを中止します。

## ADD\_DEFAULT\_DEST プロシージャ

このプロシージャは、接続先データベースを DEFDEFAULTDEST データ・ディクショナリ・ビューに追加します。

### 構文

```
DBMS_DEFER_SYS.ADD_DEFAULT_DEST (  
    dblink    IN    VARCHAR2);
```

### パラメータ

表 14-2 ADD\_DEFAULT\_DEST プロシージャのパラメータ

パラメータ	説明
dblink	DEFDEFAULTDEST ビューに追加するノードの、完全修飾されたデータベース名です。

### 例外

表 14-3 ADD\_DEFAULT\_DEST プロシージャの例外

例外	説明
ORA-23352	指定された dblink は、デフォルトのリストにすでに存在しています。

## CLEAR\_PROP\_STATISTICS プロシージャ

このプロシージャは、DEFSCHEDULE データ・ディクショナリ・ビューの伝播統計を消去します。このプロシージャが正常に実行されると、このビューのすべての統計は 0（ゼロ）に戻され、新たに統計収集が開始されます。

特に、このプロシージャは DEFSCHEDULE データ・ディクショナリ・ビューの次の列からの統計を消去します。

- TOTAL\_TXN\_COUNT
- AVG\_THROUGHPUT
- AVG\_LATENCY
- TOTAL\_BYTES\_SENT
- TOTAL\_BYTES\_RECEIVED
- TOTAL\_ROUND\_TRIPS
- TOTAL\_ADMIN\_COUNT
- TOTAL\_ERROR\_COUNT
- TOTAL\_SLEEP\_TIME

### 構文

```
DBMS_DEFER_SYS.CLEAR_PROP_STATISTICS (  
    dblink    IN    VARCHAR2);
```

### パラメータ

表 14-4 CLEAR\_PROP\_STATISTICS プロシージャのパラメータ

パラメータ	説明
dblink	統計を消去するノードの、完全修飾されたデータベース名です。消去される統計は、現在のノードから dblink に指定されたノードへの遅延トランザクションの伝播に関する統計です。

## DELETE\_DEFAULT\_DEST プロシージャ

このプロシージャは、接続先データベースを DEFDEFAULTDEST ビューから削除します。

### 構文

```
DBMS_DEFER_SYS.DELETE_DEFAULT_DEST (  
    dblink      IN    VARCHAR2);
```

### パラメータ

表 14-5 DELETE\_DEFAULT\_DEST プロシージャのパラメータ

パラメータ	説明
dblink	DEFDEFAULTDEST ビューから削除するノードの、完全修飾されたデータベース名です。ビューでこの dblink がみつからない場合、削除作業は実行されません。

## DELETE\_DEF\_DESTINATION プロシージャ

このプロシージャは、接続先データベースを DEFSCHEDULE ビューから削除します。

### 構文

```
DBMS_DEFER_SYS.DELETE_DEF_DESTINATION (  
    destination  IN    VARCHAR2,  
    force        IN    BOOLEAN := false);
```

### パラメータ

表 14-6 DELETE\_DEF\_DESTINATION プロシージャのパラメータ

パラメータ	説明
destination	DEFSCHEDULE ビューから削除する接続先の完全修飾されたデータベース名です。ビューでこの接続先がみつからない場合、削除作業は実行されません。
force	値が TRUE に設定されていると、すべての安全チェックが無視され、接続先が削除されます。

## DELETE\_ERROR プロシージャ

このプロシージャは、DEFERROR ビューからトランザクションを削除します。

### 構文

```
DBMS_DEFER_SYS.DELETE_ERROR(  
    deferred_tran_id    IN    VARCHAR2,  
    destination         IN    VARCHAR2);
```

### パラメータ

**表 14-7 DELETE\_ERROR プロシージャのパラメータ**

パラメータ	説明
deferred_tran_id	DEFERROR ビューから削除する遅延トランザクションの、DEFERROR ビューから渡される識別番号です。このパラメータが NULL の場合、他のパラメータの要件に一致するすべてのトランザクションが削除されます。
destination	トランザクションが初めにキューに入れられたデータベースの DEFERROR ビューから渡される、完全修飾されたデータベース名です。このパラメータが NULL の場合、他のパラメータの要件に一致するすべてのトランザクションが、DEFERROR ビューから削除されます。

## DELETE\_TRAN プロシージャ

このプロシージャは、DEFTRANDEST ビューからトランザクションを削除します。他の DEFTRANDEST または DEFERROR エントリがない場合、トランザクションは DEFTRAN ビューと DEFCALL ビューからも削除されます。

### 構文

```
DBMS_DEFER_SYS.DELETE_TRAN (  
    deferred_tran_id    IN    VARCHAR2,  
    destination         IN    VARCHAR2);
```

パラメータ

表 14-8 DELETE\_TRAN プロシージャのパラメータ

パラメータ	説明
deferred_tran_id	削除する遅延トランザクションの DEFTRAN ビューから渡される識別番号です。値が NULL の場合、他のパラメータの要件に一致するすべてのトランザクションが削除されます。
destination	トランザクションが初めにキューに入れられたデータベースの DEFTRANDEST ビューから渡される、完全修飾されたデータベース名です。値が NULL の場合、他のパラメータの要件に一致するすべてのトランザクションが削除されます。

DISABLED ファンクション

このファンクションは、カレント・サイトから指定されたサイトへの、遅延トランザクション・キューの伝播が有効になっているかどうかを判定します。指定された接続先に対して遅延リモート・プロシージャ・コール (Remote Procedure Call: RPC) のキューが使用禁止になっている場合、DISABLED ファンクションによって TRUE が戻されます。

構文

```
DBMS_DEFER_SYS.DISABLED (  
    destination IN VARCHAR2)  
RETURN BOOLEAN;
```

パラメータ

表 14-9 DISABLED ファンクションのパラメータ

パラメータ	説明
destination	伝播ステータスをチェックするノードの、完全修飾されたデータベース名です。

戻り値

表 14-10 DISABLED ファンクションの戻り値

値	説明
true	カレント・サイトからこのサイトへの伝播は使用禁止になっています。
false	カレント・サイトからこのサイトへの伝播は有効になっています。

## 例外

表 14-11 DISABLED ファンクションの例外

例外	説明
NO_DATA_FOUND	指定された destination が DEFSCHEDULE ビューにありません。

## EXCLUDE\_PUSH ファンクション

このファンクションは、遅延トランザクション PUSH を妨ぐ排他ロックを取得します（シリアルまたはパラレル）。このファンクションは、ロックの取得時にコミットを実行します。ロックは `RELEASE_ON_COMMIT => true` で取得されるため、次のコミット後に、遅延トランザクション・キューのプッシュを再開できます。

## 構文

```
DBMS_DEFER_SYS.EXCLUDE_PUSH (  
    timeout IN INTEGER)  
RETURN INTEGER;
```

## パラメータ

表 14-12 EXCLUDE\_PUSH ファンクションのパラメータ

パラメータ	説明
timeout	タイムアウトを秒で設定します。エラーが発生したか、または PUSH が行われているために、設定された期間内にロックが取得されない場合は、値 1 が戻されます。タイムアウトの値を <code>DBMS_LOCK.MAXWAIT</code> に設定すると、無期限に待機します。

## 戻り値

表 14-13 EXCLUDE\_PUSH ファンクションの戻り値

値	説明
0	ロックは正常に取得されました。
1	タイムアウトにより、ロックは取得されませんでした。
2	デッドロックにより、ロックは取得されませんでした。
4	ロックはすでに取得されています。

## EXECUTE\_ERROR プロシージャ

このプロシージャは、正常に終了しなかった遅延トランザクションを、トランザクションの元の受信者のセキュリティ・コンテキストで再実行します。

### 構文

```
DBMS_DEFER_SYS.EXECUTE_ERROR (  
    deferred_tran_id IN   VARCHAR2,  
    destination      IN   VARCHAR2);
```

### パラメータ

表 14-14 EXECUTE\_ERROR プロシージャのパラメータ

パラメータ	説明
deferred_tran_id	再実行する遅延トランザクションの DEFERROR ビューから渡される識別番号です。これが NULL の場合、destination へのキューに入れているすべてのトランザクションが再実行されます。
destination	トランザクションが初めにキューに入れられたデータベースの DEFERROR ビューから渡される、完全修飾されたデータベース名です。NULL は指定できません。指定したデータベース名が完全修飾されていない場合や無効の場合、エラーは発生しません。

### 例外

表 14-15 EXECUTE\_ERROR プロシージャの例外

例外	説明
ORA-24275	NULL パラメータと NULL 以外のパラメータの無効な組合せが使用されています。
badparam	パラメータ値が欠落しているか、または無効です（たとえば、destination が NULL）。
missinguser	無効なユーザーです。



## EXECUTE\_ERROR\_AS\_USER プロシージャ

このプロシージャは、正常に終了しなかった遅延トランザクションを再実行します。各トランザクションは、接続しているユーザーのセキュリティ・コンテキストで実行されます。

### 構文

```
DBMS_DEFER_SYS.EXECUTE_ERROR_AS_USER (  
    deferred_tran_id IN    VARCHAR2,  
    destination      IN    VARCHAR2);
```

### パラメータ

表 14-16 EXECUTE\_ERROR\_AS\_USER プロシージャのパラメータ

パラメータ	説明
deferred_tran_id	再実行する遅延トランザクションの DEFERROR ビューから渡される識別番号です。これが NULL の場合、destination へのキューに入れられているすべてのトランザクションが再実行されます。
destination	トランザクションが初めにキューに入れられたデータベースの DEFERROR ビューから渡される、完全修飾されたデータベース名です。NULL は指定できません。

### 例外

表 14-17 EXECUTE\_ERROR\_AS\_USER プロシージャの例外

例外	説明
ORA-24275	NULL パラメータと NULL 以外のパラメータの無効な組合せが使用されています。
badparam	パラメータ値が欠落しているか、または無効です（たとえば、destination が NULL）。
missinguser	無効なユーザーです。

## PURGE ファンクション

このファンクションは、カレント・マスター・サイトまたはマテリアライズド・ビュー・サイトの遅延トランザクション・キューから、プッシュされたトランザクションをパージします。

### 構文

```
DBMS_DEFER_SYS.PURGE (  
    purge_method          IN  BINARY_INTEGER := purge_method_quick,  
    rollback_segment      IN  VARCHAR2       := NULL,  
    startup_seconds       IN  BINARY_INTEGER := 0,  
    execution_seconds     IN  BINARY_INTEGER := seconds_infinity,  
    delay_seconds         IN  BINARY_INTEGER := 0,  
    transaction_count     IN  BINARY_INTEGER := transactions_infinity,  
    write_trace           IN  BOOLEAN         := NULL);  
RETURN BINARY_INTEGER;
```

### パラメータ

表 14-18 PURGE ファンクションのパラメータ

パラメータ	説明
purge_method	遅延トランザクション・キューのパージ方法を制御します。 purge_method_quick はコストを低く抑えられますが、 purge_method_precise の方が精度が高くなります。  purge_method_quick を使用するには、次のように指定します。 dbms_defer_sys.purge_method_quick  purge_method_precise を使用するには、次のように指定します。 dbms_defer_sys.purge_method_precise  purge_method_quick を使用する場合、正常にプッシュされた遅延 トランザクションおよび遅延プロシージャ・コールがパージされるま で、それぞれ DEFTRAN および DEFCALL データ・ディクショナリ・ ビューに保持されることがあります。詳細は、14-14 ページの「 <a href="#">使用 上の注意</a> 」を参照してください。
rollback_segment	パージに使用するロールバック・セグメントの名前、またはデフォル トの NULL です。
startup_seconds	同じ遅延トランザクション・キューの前のパージを待つ最大秒数で す。
execution_seconds	値が > 0 の場合、指定された実秒数後に、パージが完全に停止しま す。

表 14-18 PURGE ファンクションのパラメータ (続き)

パラメータ	説明
delay_seconds	delay_seconds の間、遅延トランザクション・キューにパージするトランザクションがなくなると、パージが完全に停止します。
transaction_count	値が >0 の場合は、transaction_count の数だけトランザクションがパージされると、完全にシャットダウンされます。
write_trace	値が TRUE に設定されている場合、PURGE ファンクションから戻された結果値が、サーバーのトレース・ファイルに記録されます。値が FALSE に設定されている場合は、結果値が記録されません。

## 戻り値

表 14-19 PURGE ファンクションの戻り値

値	説明
result_ok	OK です。delay_seconds の経過後に終了しました。
result_startup_seconds	スタート時にロック・タイムアウトにより終了しました。
result_execution_seconds	execution_seconds の超過により終了しました。
result_transaction_count	transaction_count の超過により終了しました。
result_errors	エラー発生後に終了しました。
result_split_del_order_limit	排他モードでエンキューの取得の失敗後に終了しました。このリターン・コードが戻された場合は、パージを再試行してください。問題が解決しない場合は、オラクル社カスタマ・サポート・センターまでお問い合わせください。
result_purge_disabled	静止せずに新しいマスター・サイトを追加するときの同期のため、キューのパージが内部で使用禁止になっています。

## 例外

表 14-20 PURGE ファンクションの例外

例外	説明
argoutofrange	パラメータ値が有効範囲外です。
executiondisabled	パージの実行が使用禁止になっています。
defererror	内部エラーが発生しました。

## 使用上の注意

DBMS\_DEFER\_SYS.PURGE ファンクションの `purge_method` パラメータに `purge_method_quick` を指定すると、遅延トランザクションおよび遅延プロシージャ・コールが正常にプッシュされた後、それぞれ DEFCALL および DEFTRAN データ・ディクショナリ・ビューに保持されることがあります。これは、複数のデータベース・リンクがあるレプリケーション環境で、1つのデータベース・リンクのみに対してプッシュが実行された場合に発生します。

遅延トランザクションおよび遅延プロシージャ・コールをパージするには、次のいずれかを実行します。

- `purge_method` パラメータに、`purge_method_quick` ではなく `purge_method_precise` を指定します。`purge_method_precise` を使用するとコストは増大しますが、遅延トランザクションおよびプロシージャ・コールは、正常にプッシュされた後で確実にパージされます。
- `purge_method` パラメータに `purge_method_quick` を指定して、遅延トランザクションをすべてのデータベース・リンクにプッシュします。遅延トランザクションおよび遅延プロシージャ・コールは、最後のデータベース・リンクに正常にプッシュされた時点でパージされます。

## PUSH ファンクション

このファンクションは、カレント・マスター・サイトまたはマテリアライズド・ビュー・サイトの遅延リモート・プロシージャ・コール (RPC) ・キューを、強制的にリモート・サイトへプッシュ (伝播) します。シリアル伝播またはパラレル伝播のいずれかが使用されます。

## 構文

```
DBMS_DEFER_SYS.PUSH (
    destination          IN  VARCHAR2,
    parallelism          IN  BINARY_INTEGER := 0,
    heap_size            IN  BINARY_INTEGER := 0,
    stop_on_error        IN  BOOLEAN        := false,
    write_trace          IN  BOOLEAN        := false,
    startup_seconds      IN  BINARY_INTEGER := 0,
    execution_seconds    IN  BINARY_INTEGER := seconds_infinity,
    delay_seconds        IN  BINARY_INTEGER := 0,
    transaction_count    IN  BINARY_INTEGER := transactions_infinity,
    delivery_order_limit IN  NUMBER          := delivery_order_infinity)
RETURN BINARY_INTEGER;
```

## パラメータ

表 14-21 PUSH ファンクションのパラメータ

パラメータ	説明
destination	変更を転送するマスター・サイトまたはマテリアライズド・ビュー・サイトの、完全修飾されたデータベース名です。
parallelism	0（ゼロ）はシリアル伝播を指定します。  $n > 1$ は、 $n$ 個の平行処理の平行伝播を指定します。  1 は、平行処理を 1 つのみ使用する平行伝播を指定します。
heap_size	平行伝播スケジュールのために同時に検査されるトランザクションの最大数です。最適なパフォーマンスを実現するデフォルト値が、自動的に計算されます。  <b>注意：</b> オラクル社カスタマ・サポート・センターから指示された場合を除き、パラメータを設定しないでください。
stop_on_error	値が FALSE の場合（デフォルト）、競合などのエラーが発生しても処理が続行されます。値が TRUE の場合、接続先サイトでのエラーの発生が最初に指摘されたときに伝播を中止します。
write_trace	値が TRUE の場合、ファンクションから戻された結果値が、サーバーのトレース・ファイルに記録されます。値が FALSE に設定されている場合は、結果値が記録されません。
startup_seconds	同じ接続先への前のプッシュを待つ最大秒数です。
execution_seconds	値が $> 0$ の場合は、指定された実秒数後に、プッシュが完全に停止します。transaction_count と execution_seconds が 0（ゼロ）の場合（デフォルト）、キューにトランザクションがなくなるまで、トランザクションが実行されます。  execution_seconds パラメータは、操作が開始可能な期間のみを制御します。リモート・サイトのトランザクションに必要な時間は含まれません。したがって、execution_seconds パラメータは、リモート・サイトへのトランザクションの伝播を正確に制御して停止するためのものではありません。正確に制御する必要がある場合は、transaction_count または delivery_order パラメータを使用します。
delay_seconds	キューが空でも、指定された秒数が経過するまでプッシュは終了しません。タイトなループから PUSH がコールされるとき、実行オーバーヘッドを縮小するのに便利です。
transaction_count	値が $> 0$ の場合は、トランザクションの最大数のプッシュ後に停止します。transaction_count と execution_seconds が 0（ゼロ）の場合（デフォルト）、キューにプッシュする必要のあるトランザクションがなくなるまで、トランザクションが実行されます。

表 14-21 PUSH ファンクションのパラメータ（続き）

パラメータ	説明
delivery_order_limit	delivery_order >= delivery_order_limit のトランザクションのプッシュ前に、実行が完全に停止します。

戻り値

表 14-22 PUSH ファンクションの戻り値

値	説明
result_ok	OK です。delay_seconds の経過後に終了しました。
result_startup_seconds	スタート時にロック・タイムアウトにより終了しました。
result_execution_seconds	execution_seconds の超過により終了しました。
result_transaction_count	transaction_count の超過により終了しました。
result_delivery_order_limit	delivery_order_limit の超過により終了しました。
result_errors	エラー発生後に終了しました。
result_push_disabled	PUSH が内部で使用禁止になりました。通常はこの戻り値は、マスター・グループを静止せずにマスター・グループに新しいマスター・サイトを追加するとき、伝播同期のため、接続先への伝播が内部で使用禁止に設定されたことを意味します。伝播は後で自動的に使用可能になります。
result_split_del_order_limit	排他モードでエンキューの取得の失敗後に終了しました。このリターン・コードが戻された場合は、プッシュを再試行してください。問題が解決しない場合は、オラクル社カスタマ・サポート・センターまでお問い合わせください。

## 例外

**表 14-23 PUSH ファンクションの例外**

例外	説明
incompleteparallellpush	シリアル伝播を実行するには、パラレル伝播を完全にシャットダウンする必要があります。
executiondisabled	遅延リモート・プロシージャ・コール（RPC）の実行が、接続先で使用禁止になっています。
crt_err_err	DEFERROR にエントリを作成中、エラーが発生しました。
deferred_rpc_quiesce	レプリケーション・グループのレプリケーション・アクティビティが中断されました。
commfailure	遅延リモート・プロシージャ・コール（RPC）中に通信障害が発生しました。
missingpropagator	プロパゲータが存在しません。

## REGISTER\_PROPAGATOR プロシージャ

このプロシージャは、指定されたユーザーをローカル・データベースのプロパゲータとして登録します。また、指定されたユーザーに（ユーザーがラッパーを作成できるように）次の権限を付与します。

- CREATE SESSION
- CREATE PROCEDURE
- CREATE DATABASE LINK
- EXECUTE ANY PROCEDURE

## 構文

```
DBMS_DEFER_SYS.REGISTER_PROPAGATOR (  
    username IN VARCHAR2);
```

## パラメータ

**表 14-24 REGISTER\_PROPAGATOR プロシージャのパラメータ**

パラメータ	説明
username	ユーザー名です。

例外

表 14-25 REGISTER\_PROPAGATOR プロシージャの例外

例外	説明
missinguser	指定されたユーザーは存在しません。
alreadypropagator	指定されたユーザーはすでにプロパゲータとして登録されています。
duplicatepropagator	他のプロパゲータがすでに登録されています。

SCHEDULE\_PURGE プロシージャ

このプロシージャは、カレント・マスター・サイトまたはマテリアライズド・ビュー・サイトの遅延トランザクション・キューから、プッシュされたトランザクションのページ・ジョブをスケジュールします。1つのページ・ジョブをスケジュールする必要があります。

**関連項目：** このプロシージャを使用した、遅延トランザクション・キューの連続的または定期的なページのスケジュールの詳細は、『Oracle9i アドバンスド・レプリケーション』を参照してください。

構文

```
DBMS_DEFER_SYS.SCHEDULE_PURGE (  
  interval          IN  VARCHAR2,  
  next_date         IN  DATE,  
  reset             IN  BOOLEAN      := NULL,  
  purge_method      IN  BINARY_INTEGER := NULL,  
  rollback_segment  IN  VARCHAR2     := NULL,  
  startup_seconds   IN  BINARY_INTEGER := NULL,  
  execution_seconds IN  BINARY_INTEGER := NULL,  
  delay_seconds     IN  BINARY_INTEGER := NULL,  
  transaction_count IN  BINARY_INTEGER := NULL,  
  write_trace       IN  BOOLEAN      := NULL);
```



## パラメータ

**表 14-26 SCHEDULE\_PURGE プロシージャのパラメータ**

パラメータ	説明
interval	次のバージの時間を計算するファンクションを指定できます。値は DEFSCCHEDULE ビューの interval フィールドに格納され、このビューの next_date フィールドの計算に使用します。このパラメータのデフォルト値 NULL を使用した場合、このフィールドの値は変更されません。このフィールドに既存の値が設定されていない場合は、NULL で作成されます。このフィールドに値を設定しない場合は、next_date に値を設定する必要があります。
next_date	サイトのキューから、プッシュされたトランザクションをバージする時間を指定できます。値は DEFSCCHEDULE ビューの next_date フィールドに格納されます。このパラメータのデフォルト値 NULL を使用した場合、このフィールドの値は変更されません。このフィールドに既存の値が入っていない場合は、NULL で作成されます。このフィールドに値を設定しない場合は、interval に値を設定する必要があります。
reset	値を TRUE に設定すると、LAST_TXN_COUNT、LAST_ERROR および LAST_MSG が NULL にリセットされます。
purge_method	<p>遅延トランザクション・キューのバージ方法を制御します。purge_method_quick はコストを低く抑えられますが、purge_method_precise の方が精度が高くなります。</p> <p>purge_method_quick を使用するには、次のように指定します。</p> <pre>dbms_defer_sys.purge_method_quick</pre> <p>purge_method_precise を使用するには、次のように指定します。</p> <pre>dbms_defer_sys.purge_method_precise</pre> <p>purge_method_quick を使用する場合、正常にプッシュされた遅延トランザクションおよび遅延プロシージャ・コールがバージされるまで、それぞれ DEFTRAN および DEFCALL データ・ディクショナリ・ビューに保持されることがあります。詳細は、14-14 ページの「<a href="#">使用上の注意</a>」を参照してください。これらの使用法は DBMS_DEFER_SYS.PURGE ファンクションに関するものですが、DBMS_DEFER_SYS.SCHEDULE_PURGE プロシージャにも適用可能です。</p>
rollback_segment	バージに使用するロールバック・セグメントの名前、またはデフォルトの NULL です。
startup_seconds	同じ遅延トランザクション・キューの前のバージを待つ最大秒数です。

表 14-26 SCHEDULE\_PURGE プロシージャのパラメータ（続き）

パラメータ	説明
execution_seconds	値が > 0 の場合、指定された実秒数後に、パージが完全に停止します。
delay_seconds	delay_seconds の間、遅延トランザクション・キューにパージするトランザクションがなくなると、パージが完全に停止します。
transaction_count	値が > 0 の場合は、transaction_count の数だけトランザクションがパージされると、完全にシャットダウンされます。
write_trace	値が TRUE に設定されている場合、PURGE ファンクションから戻された結果値が、サーバーのトレース・ファイルに記録されます。

SCHEDULE\_PUSH プロシージャ

このプロシージャは、遅延トランザクション・キューをリモート・サイトにプッシュするジョブをスケジュールします。このプロシージャでは COMMIT が実行されます。

**関連項目：** このプロシージャを使用した、遅延トランザクション・キューの連続的または定期的なプッシュのスケジュールの詳細は、『Oracle9i アドバンスト・レプリケーション』を参照してください。

構文

```
DBMS_DEFER_SYS.SCHEDULE_PUSH (
    destination      IN  VARCHAR2,
    interval         IN  VARCHAR2,
    next_date        IN  DATE,
    reset            IN  BOOLEAN      := false,
    parallelism      IN  BINARY_INTEGER := NULL,
    heap_size        IN  BINARY_INTEGER := NULL,
    stop_on_error    IN  BOOLEAN      := NULL,
    write_trace      IN  BOOLEAN      := NULL,
    startup_seconds  IN  BINARY_INTEGER := NULL,
    execution_seconds IN  BINARY_INTEGER := NULL,
    delay_seconds    IN  BINARY_INTEGER := NULL,
    transaction_count IN  BINARY_INTEGER := NULL);
```

## パラメータ

表 14-27 SCHEDULE\_PUSH プロシージャのパラメータ

パラメータ	説明
destination	変更を転送するマスター・サイトまたはマテリアライズド・ビュー・サイトの、完全修飾されたデータベース名です。
interval	次のプッシュの時間を計算するファンクションを指定できます。値は DEFSCHEDULE ビューの interval フィールドに格納され、このビューの next_date フィールドの計算に使用します。このパラメータのデフォルト値 NULL を使用した場合、このフィールドの値は変更されません。このフィールドに既存の値が設定されていない場合は、NULL で作成されます。このフィールドに値を設定しない場合は、next_date に値を設定する必要があります。
next_date	遅延トランザクションをリモート・サイトにプッシュする時間を指定できます。値は DEFSCHEDULE ビューの next_date フィールドに格納されます。このパラメータのデフォルト値 NULL を使用した場合、このフィールドの値は変更されません。このフィールドに既存の値が設定されていない場合は、NULL で作成されます。このフィールドに値を設定しない場合は、interval に値を設定する必要があります。
reset	値を TRUE に設定すると、LAST_TXN_COUNT、LST_ERROR および LAST_MSG が NULL にリセットされます。
parallelism	0 (ゼロ) はシリアル伝播を指定します。 $n > 1$ は、 $n$ 個の平行処理の平行伝播を指定します。 1 は、平行処理を 1 つのみ使用する平行伝播を指定します。
heap_size	平行伝播スケジュールのために同時に検査されるトランザクションの最大数です。最適なパフォーマンスを実現するデフォルト値が、自動的に計算されます。  <b>注意：</b> オラクル社カスタマ・サポート・センターから指示された場合を除き、パラメータを設定しないでください。
stop_on_error	値が FALSE の場合 (デフォルト)、競合などのエラーが発生しても処理が続行されます。値が TRUE の場合、接続先サイトでのエラーの発生が最初に指摘されたときに伝播を中止します。
write_trace	値が TRUE の場合、ファンクションから戻された結果値が、サーバーのトレース・ファイルに記録されます。
startup_seconds	同じ接続先への前のプッシュを待つ最大秒数です。
execution_seconds	値が $> 0$ の場合、指定された実秒数後に、実行が完全に停止します。transaction_count と execution_seconds が 0 (ゼロ) の場合 (デフォルト)、キューにトランザクションがなくなるまで、トランザクションが実行されます。

表 14-27 SCHEDULE\_PUSH プロシージャのパラメータ（続き）

パラメータ	説明
delay_seconds	キューが空でも、指定された秒数が経過するまでプッシュは終了しません。タイトなループから PUSH がコールされるとき、実行オーバーヘッドを縮小するのに便利です。
transaction_count	値が > 0 の場合は、トランザクションの最大数のプッシュ後に停止します。transaction_count と execution_seconds が 0（ゼロ）の場合（デフォルト）、キューにプッシュする必要のあるトランザクションがなくなるまで、トランザクションが実行されます。

SET\_DISABLED プロシージャ

カレント・サイトから指定された接続先サイトへの、遅延トランザクション・キューの伝播が使用禁止あるいは使用可能にされます。disabled パラメータが TRUE の場合、指定された接続先への伝播が使用禁止になり、PUSH が呼び出されても遅延リモート・プロシージャ・コール（RPC）キューがプッシュされなくなります。SET\_DISABLED は、指定された接続先へキューをすでにプッシュしているセッションには適用されますが、DBMS\_DEFER でキューに追加されているセッションには適用されません。

disabled パラメータが FALSE の場合、指定された接続先への伝播が有効になり、キューはプッシュされませんが、PUSH の呼び出しによる指定された接続先へのキューのプッシュが可能になります。disabled パラメータが TRUE か FALSE かに関係なく、設定が他のセッションにも適用されるようにするには、COMMIT が必要です。

構文

```
DBMS_DEFER_SYS.SET_DISABLED (  
  destination    IN    VARCHAR2,  
  disabled       IN    BOOLEAN := true,  
  catchup       IN    RAW := '00',  
  override      IN    BOOLEAN := false);
```

パラメータ

表 14-28 SET\_DISABLED プロシージャのパラメータ

パラメータ	説明
destination	伝播ステータスを変更するノードの、完全修飾されたデータベース名です。
disabled	デフォルトでは、このパラメータによって、カレント・サイトから指定された接続先への遅延トランザクション・キューの伝播が使用禁止になります。値を FALSE に設定すると、伝播が有効になります。
catchup	マスター・グループを静止せずに新しいマスター・サイトをマスター・グループに追加するための、拡張識別子です。新しいマスター・サイトが接続先です。DEFSCHEDULE データ・ディクショナリ・ビューの間合せを行い、既存の拡張識別子を検索します。
override	<p>デフォルトの FALSE 設定では、disabled パラメータが FALSE に設定されており、Oracle 内部で伝播が使用禁止になった場合、cantsetdisabled 例外が呼び出されます。</p> <p>TRUE 設定では、同期のために内部で伝播の使用禁止状態が設定されたかどうかは無視され、常に disabled パラメータの指定どおりの状態の設定が行われます。</p> <p><b>注意：</b>オラクル社カスタマ・サポート・センターから指示された場合を除き、このパラメータを設定しないでください。</p>

例外

表 14-29 SET\_DISABLED プロシージャの例外

例外	説明
NO_DATA_FOUND	指定された destination の DEFSCHEDULE ビューには、エントリが見つかりませんでした。
cantsetdisabled	マスター・グループを静止せずにマスター・グループに新しいマスター・サイトを追加している間に、このサイトの使用禁止ステータスは、同期をとるために内部で設定されます。このプロシージャを呼び出す前に、静止しないで新しいマスター・サイトの追加が完了したことを確認してください。

## UNREGISTER\_PROPAGATOR プロシージャ

プロパゲータとしてのユーザーの登録を、ローカル・データベースから削除します。このプロシージャでは、次の処理が行われます。

- 指定されたプロパゲータが DEFPROPAGATOR から削除されます。
- REGISTER\_PROPAGATOR によってユーザーに付与された権限が取り消されます。別個に付与された同一の権限も取り消されます。
- 指定されたプロパゲータのスキーマ内に生成されたラッパーが削除されます。削除されたラッパーには、レプリケーション・カタログで削除済のマークが付けられます。

### 構文

```
DBMS_DEFER_SYS.UNREGISTER_PROPAGATOR (  
    username IN VARCHAR2  
    timeout   IN  INTEGER DEFAULT DBMS_LOCK.MAXWAIT);
```

### パラメータ

表 14-30 UNREGISTER\_PROPAGATOR プロシージャのパラメータ

パラメータ	説明
username	プロパゲータ・ユーザーの名前です。
timeout	タイムアウトを秒で設定します。プロパゲータが稼働中の場合、プロシージャはタイムアウトで指定された時間が経過するまで待機します。デフォルトは DBMS_LOCK.MAXWAIT です。

### 例外

表 14-31 UNREGISTER\_PROPAGATOR プロシージャの例外

パラメータ	説明
missingpropagator	指定されたユーザーはプロパゲータではありません。
propagator_inuse	プロパゲータが稼働中のため、登録を削除できません。後で再度実行してください。

## UNSCHEDULE\_PURGE プロシージャ

このプロシージャは、マスター・サイトまたはマテリアライズド・ビュー・サイトの遅延トランザクション・キューからの、プッシュされたトランザクションの自動パージを中止します。

### 構文

```
DBMS_DEFER_SYS.UNSCHEDULE_PURGE();
```

### パラメータ

なし

## UNSCHEDULE\_PUSH プロシージャ

このプロシージャは、マスター・サイトまたはマテリアライズド・ビュー・サイトからリモート・サイトへの、遅延トランザクション・キューの自動プッシュを中止します。

### 構文

```
DBMS_DEFER_SYS.UNSCHEDULE_PUSH (  
    dblink      IN    VARCHAR2);
```

### パラメータ

表 14-32 UNSCHEDULE\_PUSH プロシージャのパラメータ

パラメータ	説明
dblink	遅延リモート・プロシージャ・コールの定期的な実行スケジュールを変更するデータベースの、完全修飾されたパス名です。

表 14-33 UNSCHEDULE\_PUSH プロシージャの例外

例外	説明
NO_DATA_FOUND	指定された dblink の DEFSCHEDULE ビューには、エントリが見つかりませんでした。





---

## DBMS\_MVIEW

DBMS\_MVIEW により、クエリー・リライト機能をはじめとする、マテリアライズド・ビューおよび今後作成されるマテリアライズド・ビューの機能が理解できるようになります。また、同じリフレッシュ・グループまたはパージ・ログの一部ではないマテリアライズド・ビューのリフレッシュも可能にします。

この章では、次の項目を説明します。

- **DBMS\_MVIEW サブプログラムの概要**

---

**注意：** DBMS\_SNAPSHOT は DBMS\_MVIEW のシノニムです。

---

**関連項目：**

- レプリケーション環境でのマテリアライズド・ビューの使用の詳細は、『Oracle9i アドバンスド・レプリケーション』を参照してください。
- データ・ウェアハウス環境でのマテリアライズド・ビューの使用の詳細は、『Oracle9i データ・ウェアハウス・ガイド』を参照してください。

# DBMS\_MVIEW サブプログラムの概要

表 15-1 DBMS\_MVIEW パッケージのサブプログラム

サブプログラム	説明
15-3 ページ「 <a href="#">BEGIN_TABLE_REORGANIZATION</a> プロシージャ」	リフレッシュに必要なマテリアライズド・ビュー・データの保存処理を実行します。
15-4 ページ「 <a href="#">END_TABLE_REORGANIZATION</a> プロシージャ」	マスター表のマテリアライズド・ビュー・データが有効であり、マスター表が適切な状態にあるようにします。
15-4 ページ「 <a href="#">EXPLAIN_MVIEW</a> プロシージャ」	マテリアライズド・ビューまたは今後作成されるマテリアライズド・ビューで可能なことを説明します。
15-6 ページ「 <a href="#">EXPLAIN_REWRITE</a> プロシージャ」	問合せがリライトに失敗した理由を説明します。
15-7 ページ「 <a href="#">I_AM_A_REFRESH</a> ファンクション」	I_AM_REFRESH パッケージの状態の値を戻します。
15-7 ページ「 <a href="#">PMARKER</a> ファンクション」	ROWID からのパーティション・マーカを戻します。このファンクションはパーティション・チェンジ・トラッキング (Partition Change Tracking: PCT) 用に使用されます。
15-8 ページ「 <a href="#">PURGE_DIRECT_LOAD_LOG</a> プロシージャ」	マテリアライズド・ビューで不要になった行を、ダイレクト・ローダー・ログからパージします (データ・ウェアハウスで使用)。
15-8 ページ「 <a href="#">PURGE_LOG</a> プロシージャ」	マテリアライズド・ビュー・ログから行をパージします。
15-10 ページ「 <a href="#">PURGE_MVIEW_FROM_LOG</a> プロシージャ」	マテリアライズド・ビュー・ログから行をパージします。
15-12 ページ「 <a href="#">REFRESH</a> プロシージャ」	同じリフレッシュ・グループのメンバーではない 1 つ以上のマテリアライズド・ビューを、同じようにリフレッシュします。
15-14 ページ「 <a href="#">REFRESH_ALL_MVIEWS</a> プロシージャ」	マスター表またはマスター・マテリアライズド・ビューへの変更を反映しないマテリアライズド・ビューを、すべてリフレッシュします。
15-16 ページ「 <a href="#">REFRESH_DEPENDENT</a> プロシージャ」	指定されたマスター表またはマスター・マテリアライズド・ビューに依存する表ベースのマテリアライズド・ビューのすべて、またはマスター表またはマスター・マテリアライズド・ビューのリストを、リフレッシュします。
15-18 ページ「 <a href="#">REGISTER_MVIEW</a> プロシージャ」	マテリアライズド・ビューの個別管理を有効にします。

表 15-1 DBMS\_MVIEW パッケージのサブプログラム（続き）

サブプログラム	説明
15-20 ページ「 <a href="#">UNREGISTER_MVIEW プロシージャ</a> 」	マテリアライズド・ビューの個別管理を有効にします。マスター・サイトまたはマスター・マテリアライズド・ビュー・サイトでコールされ、マテリアライズド・ビューを登録解除します。

## BEGIN\_TABLE\_REORGANIZATION プロシージャ

このプロシージャは、リフレッシュに必要なマテリアライズド・ビュー・データの保存処理を実行します。このプロシージャは、マスター表が再編成される前にコールする必要があります。

**関連項目：** 8-20 ページ「[マテリアライズド・ビュー・ログが含まれるマスター表の再編成](#)」

### 構文

```
DBMS_MVIEW.BEGIN_TABLE_REORGANIZATION (  
    tabowner    IN    VARCHAR2,  
    tabname     IN    VARCHAR2);
```

### パラメータ

表 15-2 BEGIN\_TABLE\_REORGANIZATION プロシージャのパラメータ

パラメータ	説明
tabowner	再編成される表の所有者です。
tabname	再編成される表の名前です。

## END\_TABLE\_REORGANIZATION プロシージャ

このプロシージャは、マスター表のマテリアライズド・ビュー・データが有効であり、マスター表が適切な状態にあるようにします。このプロシージャは、マスター表が再編成された後にコールする必要があります。

**関連項目：** 8-20 ページ「マテリアライズド・ビュー・ログが含まれるマスター表の再編成」

### 構文

```
DBMS_MVIEW.END_TABLE_REORGANIZATION (  
    tabowner      IN   VARCHAR2,  
    tabname       IN   VARCHAR2);
```

### パラメータ

表 15-3 END\_TABLE\_REORGANIZATION プロシージャのパラメータ

パラメータ	説明
tabowner	再編成される表の所有者です。
tabname	再編成される表の名前です。

## EXPLAIN\_MVIEW プロシージャ

このプロシージャにより、マテリアライズド・ビューまたは今後作成されるマテリアライズド・ビューで可能な内容が確認できます。たとえば、マテリアライズド・ビューが高速リフレッシュ可能かどうかの判断、特定のマテリアライズド・ビューで実行可能なクエリー・リライトのタイプの判断などが可能です。

このプロシージャの使用は簡単です。単に DBMS\_MVIEW.EXPLAIN\_MVIEW をコールし、パラメータとして既存のマテリアライズド・ビューのスキーマおよびマテリアライズド・ビュー名を渡します。または、今後作成されるマテリアライズド・ビューの SELECT 文字列を指定することもできます。この場合マテリアライズド・ビューまたは今後作成されるマテリアライズド・ビューが分析され、結果は MV\_CAPABILITIES\_TABLE と呼ばれる表（デフォルト）または MSG\_ARRAY と呼ばれる配列に書き込まれます。

出力先を VARRAY に指定した場合を除き、EXPLAIN\_MVIEW をコールする前に utlxmv.sql スクリプトを実行する必要がある点に注意してください。このスクリプトは admin ディレクトリにあります。このスクリプトは、現行のスキーマに MV\_CAPABILITIES\_TABLE という名前の表を作成します。

# 構文

DBMS\_MVIEW パッケージで開発者用に作成された次の PL/SQL 宣言は、これらのパラメータの順序およびデータ型を示します。これらは、表または VARRAY に出力され、既存のマテリアライズド・ビュー、および今後作成されるマテリアライズド・ビューの理解を容易にします。

既存の、または今後作成されるマテリアライズド・ビューの説明を MV\_CAPABILITIES\_TABLE に出力するには、次のようにします。

```
DBMS_MVIEW.EXPLAIN_MVIEW (
  mv          IN VARCHAR2,
  statement_id IN VARCHAR2:= NULL);
```

既存の、または今後作成されるマテリアライズド・ビューの説明を VARRAY に出力するには、次のようにします。

```
DBMS_MVIEW.EXPLAIN_MVIEW (
  mv          IN VARCHAR2,
  msg_array   OUT SYS.ExplainMVArrayType);
```

# パラメータ

表 15-4 EXPLAIN\_MVIEW プロシージャのパラメータ

パラメータ	説明
mv	既存のマテリアライズド・ビューの名前（オプションで、ピリオド (.) で区切られた所有者名で修飾）、または今後作成されるマテリアライズド・ビューの SELECT 文です。
statement_id	出力行を EXPLAIN_MVIEW の特定の起動に対応付ける、クライアントが入力する一意の識別子です。
msg_array	出力を受け取る PL/SQL VARRAY です。このパラメータを使用して、EXPLAIN_MVIEW の出力を MV_CAPABILITIES_TABLE ではなく PL/SQL VARRAY に出力します。

## EXPLAIN\_REWRITE プロシージャ

このプロシージャにより、クエリー・リライトに失敗した理由、またはクエリー・リライトに成功した場合は、使用されるマテリアライズド・ビューを確認できるようになります。このプロシージャからの結果を使用することにより、可能な場合、クエリー・リライトに必要な適切な処置を取ることができます。EXPLAIN\_REWRITE 文で指定される問合せは、実際には実行されません。

表への出力を取得するには、EXPLAIN\_REWRITE をコールする前に admin/utlxlrw.sql スクリプトを実行する必要があります。このスクリプトは、現行のスキーマに REWRITE\_TABLE という名前の表を作成します。

### 構文

EXPLAIN\_REWRITE からの出力は、2 つの方法で取得できます。最初の方法は表を使用し、2 番目の方法は VARRAY を作成します。次に示すのは、出力表を使用するための基本構文です。

```
DBMS_MVIEW.EXPLAIN_REWRITE (  
    query           IN VARCHAR2,  
    mv              IN VARCHAR2,  
    statement_id    IN VARCHAR2;
```

EXPLAIN\_REWRITE の出力を表ではなく VARRAY に出力する場合は、次のようにプロシージャをコールする必要があります。

```
DBMS_MVIEW.EXPLAIN_REWRITE (  
    query           IN VARCHAR2(2000),  
    mv              IN VARCHAR2(30),  
    msg_array       IN OUT SYS.RewriteArrayType);
```

### パラメータ

表 15-5 EXPLAIN\_REWRITE プロシージャのパラメータ

パラメータ	説明
query	説明される SQL SELECT 文です。
mv	SCHEMA.MV 形式の、既存のマテリアライズド・ビューの完全修飾名です。
statement_id	出力メッセージを識別するための、クライアントが入力する一意の識別子です。
msg_array	出力を受け取る PL/SQL VARRAY です。このパラメータを使用して、EXPLAIN_REWRITE の出力を PL/SQL VARRAY に出力します。

## I\_AM\_A\_REFRESH ファンクション

このファンクションは、I\_AM\_REFRESH パッケージのステータス値を戻します。戻り値の TRUE は、マテリアライズド・ビューに対するすべてのローカル・レプリケーション・トリガーが、このセッションで使用禁止にされていることを示します。これは、各レプリケーション・トリガーが最初にこの状態を確認するためです。戻り値の FALSE は、これらのトリガーが使用可能になっていることを示します。

### 構文

```
DBMS_MVIEW.I_AM_A_REFRESH()  
RETURN BOOLEAN;
```

### パラメータ

なし

## PMARKER ファンクション

このファンクションは ROWID からのパーティション・マーカを戻します。このファンクションはパーティション・チェンジ・トラッキング (PCT) 用に使用されます。

### 構文

```
DBMS_MVIEW.PMARKER(rid IN ROWID)  
RETURN NUMBER;
```

### パラメータ

表 15-6 PMARKER プロシージャのパラメータ

パラメータ	説明
rid	マスター表の行エントリの ROWID です。

## PURGE\_DIRECT\_LOAD\_LOG プロシージャ

このプロシージャは、マテリアライズド・ビューで不要になったエントリを、ダイレクト・ローダー・ログから削除します。このプロシージャは、通常、Oracle のデータ・ウェアハウス・テクノロジーを使用する環境で実行されます。

**関連項目：** 詳細は、『Oracle9i データ・ウェアハウス・ガイド』を参照してください。

### 構文

```
DBMS_MVIEW.PURGE_DIRECT_LOAD_LOG();
```

### パラメータ

なし

## PURGE\_LOG プロシージャ

このプロシージャは、マテリアライズド・ビュー・ログから行をパージします。

### 構文

```
DBMS_MVIEW.PURGE_LOG (  
    master      IN   VARCHAR2,  
    num         IN   BINARY_INTEGER := 1,  
    flag        IN   VARCHAR2      := 'NOP');
```



パラメータ

表 15-7 PURGE\_LOG プロシージャのパラメータ

パラメータ	説明
master	マスター表またはマスター・マテリアライズド・ビューの名前です。
num	<p>マテリアライズド・ビュー・ログから行が削除される、最後にリフレッシュされたマテリアライズド・ビューの数です。たとえば、次の文では、最後にリフレッシュされたマテリアライズド・ビューを 2 つ指定し、そのリフレッシュに必要な行をログから削除します。</p> <pre>DBMS_MVIEW.PURGE_LOG('master_table', 2);</pre> <p>マテリアライズド・ビュー・ログのすべての行を削除するには、次の例で示すように、マテリアライズド・ビューの数に高い数値を指定します。</p> <pre>DBMS_MVIEW.PURGE_LOG('master_table', 9999);</pre> <p>master_table をベースとするマテリアライズド・ビューの数が 9999 より小さい場合、この文によって、master_table に対応するマテリアライズド・ビュー・ログが完全にパージされます。マテリアライズド・ビュー・ログから行がパージされている単純マテリアライズド・ビューは、次のリフレッシュ時に完全リフレッシュする必要があります。</p>
flag	<p>少なくとも 1 つのマテリアライズド・ビューのマテリアライズド・ビュー・ログから行が必ず削除されるように、delete を指定します。このパラメータは、パラメータ num の設定をオーバーライドできます。たとえば、次の文では、最後にリフレッシュされたマテリアライズド・ビューに依存行を持つマテリアライズド・ビュー・ログから、行が削除されます。</p> <pre>DBMS_MVIEW.PURGE_LOG('master_table', 1, 'delete');</pre>

## PURGE\_MVIEW\_FROM\_LOG プロシージャ

このプロシージャはマスター・サイトまたはマスター・マテリアライズド・ビュー・サイトでコールされ、マテリアライズド・ビューのリフレッシュに関連するデータ・ディクショナリ表の行を削除します。このデータ・ディクショナリ表は、`mview_id` で、または `mviewowner`、`mviewname` および `mviewsite` の組合せで識別される、指定されたマテリアライズド・ビューのマスターに保持されます。指定されたマテリアライズド・ビューが、いずれかのマスター表またはマスター・マテリアライズド・ビューで一番最初にリフレッシュされたマテリアライズド・ビューの場合は、マテリアライズド・ビュー・ログもページされます。このプロシージャでは、マテリアライズド・ビューの登録解除は行われません。

1 つのマテリアライズド・ビュー・ログのページでエラーが発生した場合、それ以前に正常に行われたマテリアライズド・ビュー・ログのページはロールバックされません。これは、マテリアライズド・ビュー・ログのサイズを最小に保つためです。エラーが発生した場合は、すべてのマテリアライズド・ビュー・ログがページされるまで、このプロシージャを再実行できます。

### 構文

```
DBMS_MVIEW.PURGE_MVIEW_FROM_LOG (  
    mview_id      IN    BINARY_INTEGER |  
    mviewowner    IN    VARCHAR2,  
    mviewname     IN    VARCHAR2,  
    mviewsite     IN    VARCHAR2);
```

---

**注意：** このプロシージャはオーバーロードされています。`mview_id` パラメータは、残りの `mviewowner`、`mviewname` および `mviewsite` の 3 つのパラメータと同時に指定できません。

---

パラメータ

表 15-8 PURGE\_MVIEW\_FROM\_LOG プロシージャのパラメータ

パラメータ	説明
mview_id	<p>ターゲット・マテリアライズド・ビューの識別に基づいてこのプロシージャを実行する場合は、mview_id パラメータを使用してマテリアライズド・ビュー識別を指定します。マテリアライズド・ビュー ID を確認するには、マテリアライズド・ビュー・ログ・サイトで DBA_BASE_TABLE_MVIEWS ビューの問合せを行います。</p> <p>ターゲット・マテリアライズド・ビューが、登録済マテリアライズド・ビューのリスト (DBA_REGISTERED_MVIEWS) にない場合は、マテリアライズド・ビュー識別に基づいてこのプロシージャを実行すると便利です。</p>
mviewowner	<p>mview_id を指定しない場合は、mviewowner パラメータを使用して、ターゲット・マテリアライズド・ビューの所有者を入力してください。マテリアライズド・ビューの所有者を確認するには、マテリアライズド・ビュー・ログ・サイトで DBA_REGISTERED_MVIEWS ビューの問合せを行います。</p>
mviewname	<p>mview_id を指定しない場合は、mviewname パラメータを使用して、ターゲット・マテリアライズド・ビューの名前を入力してください。マテリアライズド・ビューの名前を確認するには、マテリアライズド・ビュー・ログ・サイトで DBA_REGISTERED_MVIEWS ビューの問合せを行います。</p>
mviewsite	<p>mview_id を指定しない場合は、mviewsite パラメータを使用して、ターゲット・マテリアライズド・ビューのサイトを入力してください。マテリアライズド・ビューのサイトを確認するには、マテリアライズド・ビュー・ログ・サイトで DBA_REGISTERED_MVIEWS ビューの問合せを行います。</p>

# REFRESH プロシージャ

このプロシージャは、リストで指定されたマテリアライズド・ビューをリフレッシュします。

## 構文

```
DBMS_MVIEW.REFRESH (  
  { list                IN      VARCHAR2,  
    | tab                IN OUT DBMS_UTILITY.UNCL_ARRAY, }  
  method                IN      VARCHAR2      := NULL,  
  rollback_seg          IN      VARCHAR2      := NULL,  
  push_deferred_rpc      IN      BOOLEAN       := true,  
  refresh_after_errors  IN      BOOLEAN       := false,  
  purge_option           IN      BINARY_INTEGER := 1,  
  parallelism           IN      BINARY_INTEGER := 0,  
  heap_size             IN      BINARY_INTEGER := 0,  
  atomic_refresh        IN      BOOLEAN       := true);
```

**注意：** このプロシージャはオーバーロードされています。list パラメータと tab パラメータは、同時には指定できません。

## パラメータ

表 15-9 REFRESH プロシージャのパラメータ

パラメータ	説明
list   tab	<p>リフレッシュするマテリアライズド・ビューの、カンマ区切りのリストです。(シノニムはサポートされていません。) これらのマテリアライズド・ビューは、異なるスキーマに置くことができ、また異なるマスター表またはマスター・マテリアライズド・ビューを持つことができます。ただし、すべてのマテリアライズド・ビューが、ローカル・データベース内に存在する必要があります。</p> <p>カンマ区切りのリストのかわりに、DBMS_UTILITY.UNCL_ARRAY タイプの PL/SQL 表を使用できます。この表の各要素は、マテリアライズド・ビューの名前になります。</p>

表 15-9 REFRESH プロシージャのパラメータ (続き)

パラメータ	説明
method	<p>指定されたマテリアライズド・ビューのリフレッシュ方法を示す、リフレッシュ方法の文字列です。f は高速リフレッシュ、? は強制リフレッシュ、C または c は完全リフレッシュ、A または a は常時リフレッシュを示します。A と C は等価です。</p> <p>リフレッシュ方法を指定していないマテリアライズド・ビューがあるとき、つまりリフレッシュ方法の数を上回る数のマテリアライズド・ビューを指定した場合、そのマテリアライズド・ビューには、デフォルトのリフレッシュ方法が適用されます。例として SQL*Plus 内の EXECUTE 文を次に示します。</p> <pre>DBMS_MVIEW.REFRESH ('countries_mv,regions_mv,hr.employees_mv','cf');</pre> <p>この文は、countries_mv マテリアライズド・ビューの完全リフレッシュ、regions_mv マテリアライズド・ビューの高速リフレッシュ、および hr.employees マテリアライズド・ビューのデフォルト・リフレッシュを実行します。</p>
rollback_seg	マテリアライズド・ビューのリフレッシュ時に使用する、マテリアライズド・ビュー・サイト・ロールバック・セグメントの名前です。
push_deferred_rpc	更新可能なマテリアライズド・ビューのみで使用します。マテリアライズド・ビューをリフレッシュする前に、マテリアライズド・ビューから対応付けられたマスター表またはマスター・マテリアライズド・ビューに変更をプッシュする場合は、このパラメータを TRUE に設定します。FALSE に設定されている場合は、変更が一時的に失われたように見えることがあります。
refresh_after_errors	このパラメータが TRUE の場合、マテリアライズド・ビューのマスター表またはマスター・マテリアライズド・ビューの DEFERROR ビューに解消されていない競合が残っていても、更新可能なマテリアライズド・ビューのリフレッシュが続行されます。このパラメータが TRUE で atomic_refresh が FALSE の場合、このプロシージャで 1 つのマテリアライズド・ビューのリフレッシュが失敗しても、他のマテリアライズド・ビューのリフレッシュが続行されます。
purge_option	<p>パラレル伝播メカニズムを使用している場合、つまり parallelism が 1 以上に設定されている場合、0 はページなし、1 はレイジー・ページ、2 はアグレッシブ・ページになります。ほとんどの場合、レイジー・ページが最適な設定になります。複数マスター・レプリケーション・グループが異なるターゲット・サイトにプッシュされ、1 つ以上のレプリケーション・グループに対する更新が不定期に行われ不定期にプッシュされる場合は、アグレッシブ・ページを設定してキューを少なくします。すべてのレプリケーション・グループが不定期に更新およびプッシュされる場合は、0 に設定して、随時 PUSH を実行します。PUSH を実行するときは、2 に設定してキューを削減します。</p>

表 15-9 REFRESH プロシージャのパラメータ（続き）

パラメータ	説明
parallelism	0（ゼロ）はシリアル伝播を指定します。  $n > 1$ は、 $n$ 個の平行処理の平行伝播を指定します。  1 は、平行処理を 1 つのみ使用する平行伝播を指定します。
heap_size	平行伝播スケジュールのために同時に検査されるトランザクションの最大数です。最適なパフォーマンスを実現するデフォルト値が、自動的に計算されます。  <b>注意：</b> オラクル社カスタマ・サポート・センターから指示された場合を除き、このパラメータを設定しないでください。
atomic_refresh	このパラメータが TRUE の場合、リストで指定されたすべてのマテリアライズド・ビューは、1 つのトランザクションでリフレッシュされます。リフレッシュされたマテリアライズド・ビューはすべて、時間内のある 1 点で更新されます。いずれかのマテリアライズド・ビューのリフレッシュに失敗すると、すべてのマテリアライズド・ビューが更新されません。  このパラメータが FALSE の場合、各マテリアライズド・ビューはそれぞれ別個のトランザクションでリフレッシュされます。ジョブ・キュー・プロセスの数を 1 以上に設定する必要があります。

REFRESH\_ALL\_MVIEWS プロシージャ

このプロシージャは、次のプロパティを持つすべてのマテリアライズド・ビューをリフレッシュします。

- マテリアライズド・ビューが依存するマスター表またはマスター・マテリアライズド・ビューへの最新の変更以降、マテリアライズド・ビューがリフレッシュされていない
- マテリアライズド・ビューと、マテリアライズド・ビューが依存するマスター表またはマスター・マテリアライズド・ビューのすべてがローカルにある
- マテリアライズド・ビューが DBA\_MVIEWS ビューにある

このプロシージャは、データ・ウェアハウスでの使用を目的としています。

構文

```
DBMS_MVIEW.REFRESH_ALL_MVIEWS (  
  number_of_failures OUT BINARY_INTEGER,  
  method            IN  VARCHAR2         := NULL,  
  rollback_seg      IN  VARCHAR2         := NULL,  
  refresh_after_errors IN BOOLEAN        := false,  
  atomic_refresh     IN  BOOLEAN          := true);
```

パラメータ

表 15-10 REFRESH\_ALL\_MVIEWS プロシージャのパラメータ

パラメータ	説明
number_of_failures	処理中に発生した失敗の数が戻されます。
method	各マテリアライズド・ビューに対して実行するリフレッシュのタイプを示す、単一のリフレッシュ方法です。F または f は高速リフレッシュ、? は強制リフレッシュ、C または c は完全リフレッシュ、A または a は常時リフレッシュを示します。A と C は等価です。方法を指定しない場合、マテリアライズド・ビューはデフォルトのリフレッシュ方法でリフレッシュされます。
rollback_seg	マテリアライズド・ビューのリフレッシュ時に使用する、マテリアライズド・ビュー・サイト・ロールバック・セグメントの名前です。
refresh_after_errors	このパラメータが TRUE の場合、マテリアライズド・ビューのマスター表またはマスター・マテリアライズド・ビューの DEFERROR ビューに解消されていない競合が残っていても、更新可能なマテリアライズド・ビューのリフレッシュが続行されます。このパラメータが TRUE で atomic_refresh が FALSE の場合、このプロシージャで 1 つのマテリアライズド・ビューのリフレッシュが失敗しても、他のマテリアライズド・ビューのリフレッシュが続行されます。
atomic_refresh	<p>このパラメータが TRUE の場合、リフレッシュされるマテリアライズド・ビューは、1 つのトランザクションでリフレッシュされます。リフレッシュされたマテリアライズド・ビューはすべて、時間内のある 1 点で更新されます。いずれかのマテリアライズド・ビューのリフレッシュに失敗すると、すべてのマテリアライズド・ビューが更新されません。</p> <p>このパラメータが FALSE の場合、リフレッシュされる各マテリアライズド・ビューは、それぞれ別個のトランザクションでリフレッシュされます。ジョブ・キュー・プロセスの数を 1 以上に設定する必要があります。</p>

## REFRESH\_DEPENDENT プロシージャ

このプロシージャは、次のプロパティを持つすべてのマテリアライズド・ビューをリフレッシュします。

- マテリアライズド・ビューが、指定されたマスターリストのマスター表またはマスター・マテリアライズド・ビューに依存している
- マテリアライズド・ビューが依存するマスター表またはマスター・マテリアライズド・ビューへの最新の変更以降、マテリアライズド・ビューがリフレッシュされていない
- マテリアライズド・ビューと、マテリアライズド・ビューが依存するマスター表またはマスター・マテリアライズド・ビューのすべてがローカルにある
- マテリアライズド・ビューが DBA\_MVIEWS ビューにある

このプロシージャは、データ・ウェアハウスでの使用を目的としています。

### 構文

```
DBMS_MVIEW.REFRESH_DEPENDENT (  
  number_of_failures    OUT    BINARY_INTEGER,  
  { list                IN      VARCHAR2,  
    | tab               IN OUT DBMS_UTILITY.UNCL_ARRAY, }  
  method               IN      VARCHAR2      := NULL,  
  rollback_seg         IN      VARCHAR2      := NULL,  
  refresh_after_errors IN      BOOLEAN       := false,  
  atomic_refresh       IN      BOOLEAN       := true);
```

---

---

**注意：** このプロシージャはオーバーロードされています。list パラメータと tab パラメータは、同時には指定できません。

---

---



パラメータ

表 15-11 REFRESH\_DEPENDENT プロシージャのパラメータ

パラメータ	説明
number_of_failures	処理中に発生した失敗の数が戻されます。
list   tab	<p>マテリアライズド・ビューが依存できるマスター表またはマスター・マテリアライズド・ビューの、カンマ区切りのリストです。(シノニムはサポートされていません。) 表とその表に依存するマテリアライズド・ビューが、異なるスキーマに置かれていても問題はありません。ただし、すべての表およびマテリアライズド・ビューが、ローカル・データベース内に存在する必要があります。</p> <p>カンマ区切りのリストのかわりに、DBMS_UTILITY.UNCL_ARRAY タイプの PL/SQL 表を使用できます。この表の各要素は、表の名前になります。</p>
method	<p>依存マテリアライズド・ビューのリフレッシュ方法を示す、リフレッシュ方法の文字列です。特定の表に依存するすべてのマテリアライズド・ビューが、その表に対応付けられたリフレッシュ方法でリフレッシュされます。F または f は高速リフレッシュ、? は強制リフレッシュ、C または c は完全リフレッシュ、A または a は常時リフレッシュを示します。A と C は等価です。</p> <p>リフレッシュ方法を指定していない表があるとき、つまりリフレッシュ方法の数を上回る数の表を指定した場合、その表に依存するすべてのマテリアライズド・ビューには、デフォルトのリフレッシュ方法が適用されます。例として SQL*Plus 内の EXECUTE 文を次に示します。</p> <pre>DBMS_MVIEW.REFRESH_DEPENDENT ('employees,departments,hr.regions','cf');</pre> <p>この文は、employees 表に依存するマテリアライズド・ビューの完全リフレッシュ、departments 表に依存するマテリアライズド・ビューの高速リフレッシュ、および hr.regions 表に依存するマテリアライズド・ビューのデフォルト・リフレッシュを実行します。</p>
rollback_seg	マテリアライズド・ビューのリフレッシュ時に使用する、マテリアライズド・ビュー・サイト・ロールバック・セグメントの名前です。

表 15-11 REFRESH\_DEPENDENT プロシージャのパラメータ（続き）

パラメータ	説明
refresh_after_errors	このパラメータが TRUE の場合、マテリアライズド・ビューのマスター表またはマスター・マテリアライズド・ビューの DEFERROR ビューに解消されていない競合が残っていても、更新可能なマテリアライズド・ビューのリフレッシュが続行されます。このパラメータが TRUE で atomic_refresh が FALSE の場合、このプロシージャで 1 つのマテリアライズド・ビューのリフレッシュが失敗しても、他のマテリアライズド・ビューのリフレッシュが続行されます。
atomic_refresh	<p>このパラメータが TRUE の場合、リフレッシュされるマテリアライズド・ビューは、1 つのトランザクションでリフレッシュされます。リフレッシュされたマテリアライズド・ビューはすべて、時間内のある 1 点で更新されます。いずれかのマテリアライズド・ビューのリフレッシュに失敗すると、すべてのマテリアライズド・ビューが更新されません。</p> <p>このパラメータが FALSE の場合、リフレッシュされる各マテリアライズド・ビューは、それぞれ別個のトランザクションでリフレッシュされます。ジョブ・キュー・プロセスの数を 1 以上に設定する必要があります。</p>

REGISTER\_MVIEW プロシージャ

このプロシージャを実行すると、マテリアライズド・ビューの個別管理が有効になります。このプロシージャはマスター・サイトまたはマスター・マテリアライズド・ビュー・サイトでコールされ、マテリアライズド・ビューを登録します。

**注意：** 通常は、マテリアライズド・ビューはマテリアライズド・ビューの作成中に自動的に登録されます。自動登録が失敗した場合、または登録情報が削除された場合に、マテリアライズド・ビューを手動で登録するためにのみ、このプロシージャを実行する必要があります。

構文

```
DBMS_MVIEW.REGISTER_MVIEW (  
  mviewowner  IN   VARCHAR2,  
  mviewname   IN   VARCHAR2,  
  mviewsite   IN   VARCHAR2,  
  mview_id    IN   DATE | BINARY_INTEGER,  
  flag        IN   BINARY_INTEGER,  
  qry_txt     IN   VARCHAR2,  
  rep_type    IN   BINARY_INTEGER := DBMS_MVIEW.REG_UNKNOWN);
```

パラメータ

表 15-12 REGISTER\_MVIEW プロシージャのパラメータ

パラメータ	説明
mviewowner	マテリアライズド・ビューの所有者。
mviewname	マテリアライズド・ビューの名前。
mviewsite	Oracle8 以上のマスター・サイトまたはマスター・マテリアライズド・ビュー・サイトで登録されるマテリアライズド・ビューの、マテリアライズド・ビュー・サイトの名前。二重引用符は使用できません。
mview_id	マテリアライズド・ビューの識別番号です。Oracle8 以上のマテリアライズド・ビューは BINARY_INTEGER として指定します。Oracle8 以上のマスター・サイトまたはマスター・マテリアライズド・ビュー・サイトに登録される、Oracle7 のマテリアライズド・ビューは、DATE として指定します。
flag	<p>登録されるマテリアライズド・ビューのプロパティを記述する定数。割り当てることができる有効な定数を次に示します。</p> <ul style="list-style-type: none"><li>■ ROWID マテリアライズド・ビューには dbms_mview.reg_rowid_mview</li><li>■ 主キー・マテリアライズド・ビューには dbms_mview.reg_primary_key_mview</li><li>■ オブジェクト ID マテリアライズド・ビューには dbms_mview.reg_object_id_mview</li><li>■ 高速リフレッシュ可能なマテリアライズド・ビューには dbms_mview.reg_fast_refreshable_mview</li><li>■ 更新可能マテリアライズド・ビューには dbms_mview.reg_updatable_mview</li></ul> <p>1 つのマテリアライズド・ビューが、これらのプロパティの中の複数を持つ場合があります。この場合、複数のプロパティを指定するためにプラス符号 (+) を使用します。たとえば、主キー・マテリアライズド・ビューが高速リフレッシュ可能な場合は、このパラメータに次のように入力できます。</p> <div>dbms_mview.reg_primary_key_mview + dbms_mview.reg_fast_refreshable_mview</div> <p>マテリアライズド・ビューのプロパティは、ALL_MVIEWS データ・ディクショナリ・ビューを問い合わせることにより判断できます。</p>
qry_txt	マテリアライズド・ビュー定義問合せの最初の 32,000 バイト。

表 15-12 REGISTER\_MVIEW プロシージャのパラメータ（続き）

パラメータ	説明
rep_type	マテリアライズド・ビューのバージョン。割り当てることができる有効な定数を次に示します。 <ul style="list-style-type: none"><li>■ マテリアライズド・ビューが Oracle7 サイトにある場合、dbms_mview.reg_v7_snapshot</li><li>■ マテリアライズド・ビューが Oracle8 以上のサイトにある場合、dbms_mview.reg_v8_snapshot</li><li>■ マテリアライズド・ビューが Oracle7 サイトにあるか Oracle8（以上）サイトにあるかが不明な場合、dbms_mview.reg_unknown(デフォルト)</li></ul>

使用上の注意

このプロシージャは、リモート・マテリアライズド・ビュー・サイトにより、マスター・サイトまたはマスター・マテリアライズド・ビュー・サイトでリモート・プロシージャ・コールを使用してコールされます。同じ mviewowner、mviewname および mviewsite で REGISTER\_MVIEW が複数回コールされた場合は、mview\_id、flag および qry\_txt の最新の値が格納されます。問合せが VARCHAR2 の最大サイズを超えた場合、qry\_txt は問合せの最初の 32000 文字になり、残りの文字は切り捨てられます。手動でコールされたとき、mview\_id の値は、プロシージャをコールする人が、マテリアライズド・ビュー・データ・ディクショナリ・ビューで参照する必要があります。

UNREGISTER\_MVIEW プロシージャ

このプロシージャを実行すると、マテリアライズド・ビューの個別管理が有効になります。このプロシージャはマスター・サイトまたはマスター・マテリアライズド・ビュー・サイトでコールされ、マテリアライズド・ビューを登録解除します。

構文

```
DBMS_MVIEW.UNREGISTER_MVIEW (  
    mviewowner      IN   VARCHAR2,  
    mviewname       IN   VARCHAR2,  
    mviewsite       IN   VARCHAR2);
```

# パラメータ

表 15-13 UNREGISTER\_MVIEW プロシージャのパラメータ

パラメータ	説明
mviewowner	マテリアライズド・ビューの所有者。
mviewname	マテリアライズド・ビューの名前。
mviewsite	マテリアライズド・ビュー・サイトの名前。



---

## DBMS\_OFFLINE\_OG

DBMS\_OFFLINE\_OG パッケージには、マスター・グループのオフライン・インスタンスエーションのためのパブリック API が含まれます。

この章では、次の項目を説明します。

- [DBMS\\_OFFLINE\\_OG サブプログラムの概要](#)

---

**注意：** このプロシージャは、マルチマスター・レプリケーション環境での、マスター表のオフライン・インスタンスエーションに使用します。

これらのプロシージャを、[DBMS\\_OFFLINE\\_SNAPSHOT](#) パッケージのプロシージャ（マテリアライズド・ビューのオフライン・インスタンスエーションに使用）や、[DBMS\\_REPCAT\\_INSTANTIATE](#) パッケージのプロシージャ（配置テンプレートのインスタンス化に使用）と混同しないでください。これらのプロシージャの使用の詳細は、該当するパッケージの説明を参照してください。

---

## DBMS\_OFFLINE\_OG サブプログラムの概要

表 16-1 DBMS\_OFFLINE\_OG パッケージのサブプログラム

サブプログラム	説明
16-2 ページ「 <a href="#">BEGIN_INSTANTIATION</a> プロシージャ」	マスター・グループのオフライン・インスタンスエーションを開始します。
16-4 ページ「 <a href="#">BEGIN_LOAD</a> プロシージャ」	オフライン・インスタンスエーションの一環として行われる新しいマスター・サイトへのデータのインポート中、トリガーを使用禁止にします。
16-5 ページ「 <a href="#">END_INSTANTIATION</a> プロシージャ」	マスター・グループのオフライン・インスタンスエーションを終了します。
16-7 ページ「 <a href="#">END_LOAD</a> プロシージャ」	オフライン・インスタンスエーションの一環として行われる新しいマスター・サイトへのデータのインポート後、トリガーを再び有効にします。
16-8 ページ「 <a href="#">RESUME_SUBSET_OF_MASTERS</a> プロシージャ」	マスター・グループのオフライン・インスタンスエーションが行われている間、新しいサイトを除くすべての既存のサイトで、レプリケーション・アクティビティを再開します。

### BEGIN\_INSTANTIATION プロシージャ

このプロシージャは、マスター・グループのオフライン・インスタンスエーションを開始します。このプロシージャは、マスター定義サイトからコールする必要があります。

**注意：** このプロシージャは、マルチマスター・レプリケーション環境での、マスター表のオフライン・インスタンスエーションに使用します。

このプロシージャを、[DBMS\\_OFFLINE\\_SNAPSHOT](#) パッケージのプロシージャ（マテリアライズド・ビューのオフライン・インスタンスエーションに使用）や、[DBMS\\_REPCAT\\_INSTANTIATE](#) パッケージのプロシージャ（配置テンプレートのインスタンス化に使用）と混同しないでください。これらのプロシージャの使用の詳細は、該当するパッケージの説明を参照してください。

**関連項目：** マスター・サイトのオフライン・インスタンスエーションによる、マスター・グループへの新しいマスター・サイトの追加の詳細は、7-33 ページの「[エクスポート / インポートを使用するオフライン・インスタンスエーションで新しいマスター・サイトを追加](#)」を参照してください。



## 構文

```
DBMS_OFFLINE_OG.BEGIN_INSTANTIATION (
    gname      IN   VARCHAR2,
    new_site   IN   VARCHAR2
    fname      IN   VARCHAR2);
```

## パラメータ

**表 16-2 BEGIN\_INSTANTIATION プロシージャのパラメータ**

パラメータ	説明
gname	新しいサイトにレプリケートするレプリケーション・グループの名前です。
new_site	レプリケーション・グループをレプリケートする新しいサイトの、完全修飾されたデータベース名です。
fname	このパラメータは内部で使います。  <b>注意：</b> オラクル社カスタマ・サポート・センターから指示された場合を除き、このパラメータを設定しないでください。

## 例外

**表 16-3 BEGIN\_INSTANTIATION プロシージャの例外**

例外	説明
badargument	レプリケーション・グループまたは新しいマスター・サイトの名前が、NULL または空の文字列です。
dbms_repcat.nonmasterdef	このプロシージャは、マスター定義サイトからコールする必要があります。
sitealreadyexists	指定されたサイトは、すでにこのレプリケーション・グループのマスター・サイトになっています。
wrongstate	マスター定義サイトのステータスは、静止中である必要があります。
dbms_repcat.missingrepgroup	gname は、マスター・グループとしては存在していません。
dbms_repcat.missing_flavor	この例外が発生した場合は、オラクル社カスタマ・サポート・センターまでお問い合わせください。

## BEGIN\_LOAD プロシージャ

このプロシージャは、オフライン・インスタンスエーションの一環として行われる新しいマスター・サイトへのデータのインポート中、トリガーを使用禁止にします。このプロシージャは、新しいマスター・サイトからコールする必要があります。

**注意：** このプロシージャは、マルチマスター・レプリケーション環境での、マスター表のオフライン・インスタンスエーションに使用します。

このプロシージャを、[DBMS\\_OFFLINE\\_SNAPSHOT](#) パッケージのプロシージャ（マテリアライズド・ビューのオフライン・インスタンスエーションに使用）や、[DBMS\\_REPCAT\\_INSTANTIATE](#) パッケージのプロシージャ（配置テンプレートのインスタンス化に使用）と混同しないでください。これらのプロシージャの使用の詳細は、該当するパッケージの説明を参照してください。

**関連項目：** マスター・サイトのオフライン・インスタンスエーションによる、マスター・グループへの新しいマスター・サイトの追加の詳細は、7-33 ページの「[エクスポート / インポートを使用するオフライン・インスタンスエーションで新しいマスター・サイトを追加](#)」を参照してください。

### 構文

```
DBMS_OFFLINE_OG.BEGIN_LOAD (
  gname      IN   VARCHAR2,
  new_site   IN   VARCHAR2);
```

### パラメータ

表 16-4 BEGIN\_LOAD プロシージャのパラメータ

パラメータ	説明
gname	メンバーをインポートするレプリケーション・グループの名前です。
new_site	レプリケーション・グループ・メンバーをインポートする新しいサイトの、完全修飾されたデータベース名です。

## 例外

**表 16-5 BEGIN\_LOAD プロシージャの例外**

例外	説明
badargument	レプリケーション・グループまたは新しいマスター・サイトの名前が、NULL または空の文字列です。
wrongsite	このプロシージャは、新しいマスター・サイトからコールする必要があります。
unknownsite	指定されたサイトは、レプリケーション・グループで認識されません。
wrongstate	新しいマスター・サイトのステータスは、静止中である必要があります。
dbms_repcat.missingrepgroup	gname は、マスター・グループとしては存在していません。

## END\_INSTANTIATION プロシージャ

このプロシージャは、マスター・グループのオフライン・インスタンスエーションを終了します。このプロシージャは、マスター定義サイトからコールする必要があります。

---

**注意：** このプロシージャは、マルチマスター・レプリケーション環境での、マスター表のオフライン・インスタンスエーションに使用します。

このプロシージャを、[DBMS\\_OFFLINE\\_SNAPSHOT](#) パッケージのプロシージャ（マテリアライズド・ビューのオフライン・インスタンスエーションに使用）や、[DBMS\\_REPCAT\\_INSTANTIATE](#) パッケージのプロシージャ（配置テンプレートのインスタンス化に使用）と混同しないでください。これらのプロシージャの使用の詳細は、該当するパッケージの説明を参照してください。

---

**関連項目：** マスター・サイトのオフライン・インスタンスエーションによる、マスター・グループへの新しいマスター・サイトの追加の詳細は、7-33 ページの「[エクスポート / インポートを使用するオフライン・インスタンスエーションで新しいマスター・サイトを追加](#)」を参照してください。

構文

```
DBMS_OFFLINE_OG.END_INSTANTIATION (  
  gname      IN  VARCHAR2,  
  new_site   IN  VARCHAR2);
```

パラメータ

表 16-6 END\_INSTANTIATION プロシージャのパラメータ

パラメータ	説明
gname	新しいサイトにレプリケートするレプリケーション・グループの名前です。
new_site	レプリケーション・グループをレプリケートする新しいサイトの、完全修飾されたデータベース名です。

例外

表 16-7 END\_INSTANTIATION プロシージャの例外

例外	説明
badargument	レプリケーション・グループまたは新しいマスター・サイトの名前が、NULL または空の文字列です。
dbms_repcat.nonmasterdef	このプロシージャは、マスター定義サイトからコールする必要があります。
unknownsite	指定されたサイトは、レプリケーション・グループで認識されません。
wrongstate	マスター定義サイトのステータスは、静止中である必要があります。
dbms_repcat.missingrepgroup	gname は、マスター・グループとしては存在していません。

## END\_LOAD プロシージャ

このプロシージャは、オフライン・インスタンスエーションの一環として行われる新しいマスター・サイトへのデータのインポート後、トリガーを再び有効にします。このプロシージャは、新しいマスター・サイトからコールする必要があります。

**注意：** このプロシージャは、マルチマスター・レプリケーション環境での、マスター表のオフライン・インスタンスエーションに使用します。

このプロシージャを、[DBMS\\_OFFLINE\\_SNAPSHOT](#) パッケージのプロシージャ（マテリアライズド・ビューのオフライン・インスタンスエーションに使用）や、[DBMS\\_REPCAT\\_INSTANTIATE](#) パッケージのプロシージャ（配置テンプレートのインスタンス化に使用）と混同しないでください。これらのプロシージャの使用の詳細は、該当するパッケージの説明を参照してください。

**関連項目：** マスター・サイトのオフライン・インスタンスエーションによる、マスター・グループへの新しいマスター・サイトの追加の詳細は、7-33 ページの「[エクスポート / インポートを使用するオフライン・インスタンスエーションで新しいマスター・サイトを追加](#)」を参照してください。

## 構文

```
DBMS_OFFLINE_OG.END_LOAD (  
    gname      IN   VARCHAR2,  
    new_site   IN   VARCHAR2  
    fname      IN   VARCHAR2);
```

## パラメータ

表 16-8 END\_LOAD プロシージャのパラメータ

パラメータ	説明
gname	メンバーのインポートを終了したレプリケーション・グループの名前です。
new_site	レプリケーション・グループ・メンバーをインポートした新しいサイトの、完全修飾されたデータベース名です。
fname	このパラメータは内部で使います。 <b>注意：</b> オラクル社カスタマ・サポート・センターから指示された場合を除き、このパラメータを設定しないでください。

例外

表 16-9 END\_LOAD プロシージャの例外

例外	説明
badargument	レプリケーション・グループまたは新しいマスター・サイトの名前が、NULL または空の文字列です。
wrongsite	このプロシージャは、新しいマスター・サイトからコールする必要があります。
unknownsite	指定されたサイトは、レプリケーション・グループで認識されません。
wrongstate	新しいマスター・サイトのステータスは、静止中である必要があります。
dbms_repcat.missingrepgroup	gname は、マスター・グループとしては存在していません。
dbms_repcat.flavor_noobject	この例外が発生した場合は、オラクル社カスタマ・サポート・センターまでお問い合わせください。
dbms_repcat.flavor_contains	この例外が発生した場合は、オラクル社カスタマ・サポート・センターまでお問い合わせください。

RESUME\_SUBSET\_OF\_MASTERS プロシージャ

マスター・サイトのオフライン・インスタンス化によりマスター・グループに新しいマスター・サイトを追加するとき、オフライン・インスタンス化・プロセスの完了に時間がかかる場合があります。このプロシージャは、マスター・グループのオフライン・インスタンス化の間、新しいサイトを除くすべての既存のサイトで、レプリケーション・アクティビティを再開します。このプロシージャは、通常 DBMS\_OFFLINE\_OG.BEGIN\_INSTANTIATION プロシージャを実行した後に実行します。このプロシージャは、マスター定義サイトからコールする必要があります。

**注意：** このプロシージャは、マルチマスター・レプリケーション環境での、マスター表のオフライン・インスタンス化に使用します。

このプロシージャを、[DBMS\\_OFFLINE\\_SNAPSHOT](#) パッケージのプロシージャ（マテリアライズド・ビューのオフライン・インスタンス化に使用）や、[DBMS\\_REPCAT\\_INSTANTIATE](#) パッケージのプロシージャ（配置テンプレートのインスタンス化に使用）と混同しないでください。これらのプロシージャの使用の詳細は、該当するパッケージの説明を参照してください。

**関連項目：** マスター・サイトのオフライン・インスタンスーションによる、マスター・グループへの新しいマスター・サイトの追加の詳細は、7-33 ページの「[エクスポート / インポートを使用するオフライン・インスタンスーションで新しいマスター・サイトを追加](#)」を参照してください。

構文

```
DBMS_OFFLINE_OG.RESUME_SUBSET_OF_MASTERS (  
    gname      IN  VARCHAR2,  
    new_site   IN  VARCHAR2  
    override   IN  BOOLEAN := false);
```

パラメータ

表 16-10 RESUME\_SUBSET\_OF\_MASTERS プロシージャのパラメータ

パラメータ	説明
gname	新しいサイトにレプリケートするレプリケーション・グループの名前です。
new_site	レプリケーション・グループをレプリケートする新しいサイトの、完全修飾されたデータベース名です。
override	値が TRUE の場合は、保留になっている RepCat 管理要求が無視され、各マスターで通常のレプリケーション・アクティビティがリストアされます。override パラメータは、緊急の場合のみ TRUE に設定します。  値が FALSE の場合は、各マスターで通常のレプリケーション・アクティビティがリストアされます。ただし、そのマスターで gname に対する RepCat 管理要求が保留されていない場合に限られます。

例外

表 16-11 RESUME\_SUBSET\_OF\_MASTERS プロシージャの例外

例外	説明
badargument	レプリケーション・グループまたは新しいマスター・サイトの名前が、NULL または空の文字列です。
dbms_repcat.nonmasterdef	このプロシージャは、マスター定義サイトからコールする必要があります。
unknownsite	指定されたサイトは、レプリケーション・グループで認識されません。
wrongstate	マスター定義サイトのステータスは、静止中である必要があります。
dbms_repcat.missingrepgroup	gname は、マスター・グループとしては存在していません。



---

## DBMS\_OFFLINE\_SNAPSHOT

---

DBMS\_OFFLINE\_SNAPSHOT パッケージには、マテリアライズド・ビューのオフライン・インスタンス化のためのパブリック API が含まれます。

この章では、次の項目を説明します。

- [DBMS\\_OFFLINE\\_SNAPSHOT サブプログラムの概要](#)

---

**注意：** これらのプロシージャは、マテリアライズド・ビューのオフライン・インスタンス化に使用します。

これらのプロシージャを、[DBMS\\_OFFLINE\\_OG](#) パッケージのプロシージャ（マスター表のオフライン・インスタンス化に使用）や、[DBMS\\_REPCAT\\_INSTANTIATE](#) パッケージのプロシージャ（配置テンプレートのインスタンス化に使用）と混同しないでください。これらのプロシージャの使用方法の詳細は、該当するパッケージの説明を参照してください。

---

## DBMS\_OFFLINE\_SNAPSHOT サブプログラムの概要

表 17-1 DBMS\_OFFLINE\_SNAPSHOT パッケージのサブプログラム

サブプログラム	説明
17-2 ページ「 <a href="#">BEGIN_LOAD プロシージャ</a> 」	オフライン・インスタンスエーションの一環として行われる新しいマテリアライズド・ビューのインポート用に、マテリアライズド・ビュー・サイトを用意します。
17-4 ページ「 <a href="#">END_LOAD プロシージャ</a> 」	マテリアライズド・ビューのオフライン・インスタンスエーションを終了します。

### BEGIN\_LOAD プロシージャ

このプロシージャは、オフライン・インスタンスエーションの一環として行われる新しいマテリアライズド・ビューのインポート用に、マテリアライズド・ビュー・サイトを用意します。このプロシージャは、新しいマテリアライズド・ビューのマテリアライズド・ビュー・サイトからコールする必要があります。

**注意：** このプロシージャは、マテリアライズド・ビューのオフライン・インスタンスエーションに使用します。

これらのプロシージャを、[DBMS\\_OFFLINE\\_OG](#) パッケージのプロシージャ（マスター表のオフライン・インスタンスエーションに使用）や、[DBMS\\_REPCAT\\_INSTANTIATE](#) パッケージのプロシージャ（配置テンプレートのインスタンス化に使用）と混同しないでください。これらのプロシージャの使用の詳細は、該当するパッケージの説明を参照してください。

**関連項目：** エクスポートまたはインポートを使用したオフライン・インスタンスエーションによる、新しいマテリアライズド・ビュー・サイトの追加の詳細は、8-25 ページの「[エクスポート / インポートによるマテリアライズド・ビュー・サイトのオフライン・インスタンスエーション](#)」を参照してください。

### 構文

```
DBMS_OFFLINE_SNAPSHOT.BEGIN_LOAD (  
  gname          IN   VARCHAR2,  
  sname          IN   VARCHAR2,  
  master_site    IN   VARCHAR2,  
  snapshot_otype IN   VARCHAR2,  
  storage_c      IN   VARCHAR2 := '',  
  comment        IN   VARCHAR2 := '',  
  min_communication IN BOOLEAN := true);
```

## パラメータ

表 17-2 BEGIN\_LOAD プロシージャのパラメータ

パラメータ	説明
gname	オフライン・インスタンスエーションで作成するマテリアライズド・ビューの、レプリケーション・グループの名前です。
sname	新しいマテリアライズド・ビューのスキーマの名前です。
master_site	マテリアライズド・ビューのマスター・サイトの、完全修飾されたデータベース名です。
snapshot_ename	マスター・サイトで作成される一時マテリアライズド・ビューの名前です。
storage_c	マテリアライズド・ビュー・サイトでの新しいマテリアライズド・ビューの作成に使用する記憶域オプションです。
comment	ユーザーのコメントです。
min_communication	値が TRUE の場合、UPDATE 文で列が変更されるときにかぎり、更新トリガーで列の新しい値が送られます。また、列がキー列、または変更された列グループ内の列のときにかぎり、更新トリガーで列の古い値が送られます。

## 例外

表 17-3 BEGIN\_LOAD プロシージャの例外

例外	説明
badargument	レプリケーション・グループ、スキーマ、マスター・サイトまたはマテリアライズド・ビューの名前が、NULL または空の文字列です。
dbms_repcat.missingrepgroup	gname は、レプリケーション・グループとしては存在していません。
missingremotemview	指定されたマスター・サイトで指定されたマテリアライズド・ビューが見つかりませんでした。
dbms_repcat.missingschema	指定されたスキーマは存在しません。
mviewtabmismatch	マスターのマテリアライズド・ビューの実表名とマテリアライズド・ビューが一致しません。

## END\_LOAD プロシージャ

このプロシージャは、マテリアライズド・ビューのオフライン・インスタンス化を完了します。このプロシージャは、新しいマテリアライズド・ビューのマテリアライズド・ビュー・サイトからコールする必要があります。

**注意：** このプロシージャは、マテリアライズド・ビューのオフライン・インスタンス化に使用します。

これらのプロシージャを、[DBMS\\_OFFLINE\\_OG](#) パッケージのプロシージャ（マスター表のオフライン・インスタンス化に使用）や、[DBMS\\_REPCAT\\_INSTANTIATE](#) パッケージのプロシージャ（配置テンプレートのインスタンス化に使用）と混同しないでください。これらのプロシージャの使用の詳細は、該当するパッケージの説明を参照してください。

**関連項目：** エクスポートまたはインポートを使用したオフライン・インスタンス化による、新しいマテリアライズド・ビュー・サイトの追加の詳細は、8-25 ページの「[エクスポート / インポートによるマテリアライズド・ビュー・サイトのオフライン・インスタンス化](#)」を参照してください。

### 構文

```
DBMS_OFFLINE_SNAPSHOT.END_LOAD (  
  gname          IN  VARCHAR2,  
  sname          IN  VARCHAR2,  
  snapshot_ename IN  VARCHAR2);
```

### パラメータ

表 17-4 END\_LOAD プロシージャのパラメータ

パラメータ	説明
gname	オフライン・インスタンス化で作成するマテリアライズド・ビューの、レプリケーション・グループの名前です。
sname	新しいマテリアライズド・ビューのスキーマの名前です。
snapshot_ename	マテリアライズド・ビューの名前です。

例外

表 17-5 END\_LOAD プロシージャの例外

例外	説明
badargument	レプリケーション・グループ、スキーマまたはマテリアライズド・ビューの名前が、NULL または空の文字列です。
dbms_repcat.missingrepgroup	gname は、レプリケーション・グループとしては存在していません。
dbms_repcat.nonmview	このプロシージャは、マテリアライズド・ビュー・サイトからコールする必要があります。



---

## DBMS\_RECTIFIER\_DIFF

DBMS\_RECTIFIER\_DIFF パッケージには、2つのレプリケーション・サイトの間の、データの非一貫性の検出および解決に使用される API が含まれます。

この章では、次の項目を説明します。

- [DBMS\\_RECTIFIER\\_DIFF サブプログラムの概要](#)

## DBMS\_RECTIFIER\_DIFF サブプログラムの概要

表 18-1 DBMS\_RECTIFIER\_DIFF パッケージのサブプログラム

サブプログラム	説明
18-2 ページ 「DIFFERENCES プロ シージャ」	2 つの表の相違点を判定します。
18-5 ページ「RECTIFY プロシージャ」	2 つの表の相違点を解決します。

### DIFFERENCES プロシージャ

このプロシージャは、2 つの表の相違点を判定します。このプロシージャは、ネストした表の記憶表を受け入れます。

**注意：** このプロシージャは、LOB 列やユーザー定義型をベースとする列には使用できません。

### 構文

```
DBMS_RECTIFIER_DIFF.DIFFERENCES (  
  sname1          IN  VARCHAR2,  
  oname1          IN  VARCHAR2,  
  reference_site  IN  VARCHAR2 := '',  
  sname2          IN  VARCHAR2,  
  oname2          IN  VARCHAR2,  
  comparison_site IN  VARCHAR2 := '',  
  where_clause    IN  VARCHAR2 := '',  
  { column_list   IN  VARCHAR2 := '',  
    | array_columns IN  dbms_utility.name_array, }  
  missing_rows_sname IN  VARCHAR2,  
  missing_rows_oname1 IN  VARCHAR2,  
  missing_rows_oname2 IN  VARCHAR2,  
  missing_rows_site IN  VARCHAR2 := '',  
  max_missing      IN  INTEGER,  
  commit_rows      IN  INTEGER := 500);
```

**注意：** このプロシージャはオーバーロードされています。column\_list パラメータと array\_columns パラメータは、同時には指定できません。



## パラメータ

**表 18-2 DIFFERENCES プロシージャのパラメータ**

パラメータ	説明
sname1	reference_site のスキーマの名前です。
oname1	reference_site の表の名前です。
reference_site	参照データベース・サイトの名前です。デフォルトの NULL は、カレント・サイトを示します。
sname2	comparison_site のスキーマの名前です。
oname2	comparison_site の表の名前です。
comparison_site	比較データベース・サイトの名前です。デフォルトの NULL は、カレント・サイトを示します。
where_clause	この句を満たす行のみが比較されます。デフォルトの NULL の場合は、すべての行が比較されます。
column_list	2 つの表で比較される 1 つ以上の列名の、カンマ区切りのリストです。カンマの前後には空白を入れません。デフォルトの NULL の場合は、すべての列が比較されます。
array_columns	2 つの表で比較される列名の、PL/SQL 索引付き表です。索引付けは 1 の位置から始まり、配列の最後の要素は NULL になります。1 の位置が NULL の場合は、すべての列が使用されます。
missing_rows_sname	欠落行のある表が含まれるスキーマの名前です。
missing_rows_oname1	comparison_site の表から欠落している reference_site の表の行、および reference_site の表から欠落している comparison_site サイトの行の情報が格納される、missing_rows_site にある既存の表の名前です。
missing_rows_oname2	欠落行の情報が格納される missing_rows_site にある既存の名前です。この表には 3 つの列があります。R_ID 列は missing_rows_oname1 表の行の ROWID を示し、PRESENT 列は行が存在するサイトの名前を示し、ABSENT 列は行が欠落しているサイトの名前を示します。
missing_rows_site	missing_rows_oname1 表と missing_rows_oname2 表があるサイトの名前です。デフォルトの NULL は、2 つの表がカレント・サイトにあることを示します。

表 18-2 DIFFERENCES プロシージャのパラメータ (続き)

パラメータ	説明
max_missing	missing_rows_onsame 表に挿入される行の最大数を示す整数値です。max_missing を超える数の行が欠落している場合は、指定された最大数の行が missing_rows_onsame に挿入され、それを超える行は欠落しているかどうかを判定せずに、ルーチンが通常どおり終了します。フラグメント間の違いが大きく、欠落行表へのエントリが多くなりすぎると、ルーチンを続行しても無駄になります。このような場合、このパラメータを使用すると便利です。max_missing の値が 1 より小さい場合、または NULL の場合は、例外 badnumber が発生します。
commit_rows	COMMIT が行われる前に、参照表や比較表に挿入される、または参照表や比較表から削除される行の最大数です。デフォルトでは、500 行が挿入または削除された後、COMMIT が行われます。空の文字列 ( '' ) や NULL の場合は、1 つの表に対してすべての行が挿入または削除された後に、COMMIT が行われます。

例外

表 18-3 DIFFERENCES プロシージャの例外

例外	説明
nosuchsite	データベース・サイトが見つかりませんでした。
badnumber	commit_rows パラメータが 1 未満です。
missingprimarykey	列のリストには、主キー (または等価の SET_COLUMNS) が含まれる必要があります。
badname	表またはスキーマの名前が、NULL または空の文字列です。
cannotbenull	パラメータは NULL に設定できません。
notshapeequivalent	比較される表の形式が一致しません。形式とは、列数、列名、列のデータ型を意味します。
unknowncolumn	列が存在しません。
unsupportedtype	タイプがサポートされていません。
dbms_repcat.commfailure	リモート・サイトにアクセスできません。
dbms_repcat.missingobject	表が存在しません。

## 制限事項

欠落行表に一意キーまたは主キーの制約がある場合、エラー ORA-00001（一意制約に反しています）が発生します。

## RECTIFY プロシージャ

このプロシージャは、2 つの表の相違点を解決します。このプロシージャは、ネストした表の記憶表を受け入れます。

---

---

**注意：** このプロシージャは、LOB 列やユーザー定義型をベースとする列には使用できません。

---

---

## 構文

```
DBMS_RECTIFIER_DIFF.RECTIFY (  
    sname1             IN  VARCHAR2,  
    oname1             IN  VARCHAR2,  
    reference_site     IN  VARCHAR2 := '',  
    sname2             IN  VARCHAR2,  
    oname2             IN  VARCHAR2,  
    comparison_site    IN  VARCHAR2 := '',  
    { column_list      IN  VARCHAR2 := '',  
      | array_columns  IN  dbms_utility.name_array, }  
    missing_rows_sname IN  VARCHAR2,  
    missing_rows_oname1 IN VARCHAR2,  
    missing_rows_oname2 IN VARCHAR2,  
    missing_rows_site  IN  VARCHAR2 := '',  
    commit_rows        IN  INTEGER := 500);
```

---

---

**注意：** このプロシージャはオーバーロードされています。column\_list パラメータと array\_columns パラメータは、同時には指定できません。

---

---

パラメータ

表 18-4 RECTIFY プロシージャのパラメータ

パラメータ	説明
sname1	reference_site のスキーマの名前です。
oname1	reference_site の表の名前です。
reference_site	参照データベース・サイトの名前です。デフォルトの NULL は、カレント・サイトを示します。
sname2	comparison_site のスキーマの名前です。
oname2	comparison_site の表の名前です。
comparison_site	比較データベース・サイトの名前です。デフォルトの NULL は、カレント・サイトを示します。
column_list	2 つの表で比較される 1 つ以上の列名の、カンマ区切りのリストです。カンマの前後には空白を入れません。デフォルトの NULL の場合は、すべての列が比較されます。
array_columns	2 つの表で比較される列名の、PL/SQL 索引付き表です。索引付けは 1 の位置から始まり、配列の最後の要素は NULL になります。1 の位置が NULL の場合は、すべての列が使用されます。
missing_rows_sname	欠落行のある表が含まれるスキーマの名前です。
missing_rows_oname1	comparison_site の表から欠落している reference_site の表の行、および reference_site の表から欠落している comparison_site の行の情報が格納される、missing_rows_site の表の名前です。
missing_rows_oname2	欠落行の情報が格納される missing_rows_site の表の名前です。この表には 3 つの列があります。それぞれの列には、missing_rows_oname1 表の行の ROWID、行が存在するサイトの名前、および行が欠落しているサイトの名前が格納されます。
missing_rows_site	missing_rows_oname1 表と missing_rows_oname2 表があるサイトの名前です。デフォルトの NULL は、2 つの表がカレント・サイトにあることを示します。
commit_rows	COMMIT が行われる前に、参照表や比較表に挿入される、または参照表や比較表から削除される行の最大数です。デフォルトでは、500 行が挿入または削除された後、COMMIT が行われます。空の文字列（' '）や NULL の場合は、1 つの表に対してすべての行が挿入または削除された後に、COMMIT が行われます。

例外

表 18-5 RECTIFY プロシージャの例外

例外	説明
<code>nosuchsite</code>	データベース・サイトが見つかりませんでした。
<code>badnumber</code>	<code>commit_rows</code> パラメータが 1 未満です。
<code>badname</code>	表またはスキーマの名前が、NULL または空の文字列です。
<code>dbms_repcat.commfailure</code>	リモート・サイトにアクセスできません。
<code>dbms_repcat.missingobject</code>	表が存在しません。



---

## DBMS\_REFRESH

DBMS\_REFRESH により、リフレッシュを一括して行えるマテリアライズド・ビューのグループを作成し、ある一時点において、整合性のある変換が行えるようになります。

この章では、次の項目を説明します。

- [DBMS\\_REFRESH サブプログラムの概要](#)

## DBMS\_REFRESH サブプログラムの概要

表 19-1 DBMS\_REFRESH パッケージのサブプログラム

サブプログラム	説明
19-2 ページ「 <a href="#">ADD プロシージャ</a> 」	マテリアライズド・ビューをリフレッシュ・グループに追加します。
19-3 ページ「 <a href="#">CHANGE プロシージャ</a> 」	リフレッシュ・グループのリフレッシュ間隔を変更します。
19-6 ページ「 <a href="#">DESTROY プロシージャ</a> 」	リフレッシュ・グループからすべてのマテリアライズド・ビューを削除し、リフレッシュ・グループを削除します。
19-6 ページ「 <a href="#">MAKE プロシージャ</a> 」	リフレッシュ・グループ・メンバー、およびこのグループ・メンバーをリフレッシュする間隔を指定します。
19-9 ページ「 <a href="#">REFRESH プロシージャ</a> 」	リフレッシュ・グループを手動でリフレッシュします。
19-10 ページ「 <a href="#">SUBTRACT プロシージャ</a> 」	マテリアライズド・ビューをリフレッシュ・グループから削除します。

### ADD プロシージャ

このプロシージャは、マテリアライズド・ビューをリフレッシュ・グループに追加します。

**関連項目：** 詳細は、5-11 ページの手順 6「リフレッシュ・グループへのオブジェクトの追加」および『Oracle9i アドバンスド・レプリケーション』を参照してください。

### 構文

```
DBMS_REFRESH.ADD (  
    name      IN VARCHAR2,  
    { list    IN VARCHAR2,  
      | tab   IN DBMS_UTILITY.UNCL_ARRAY, }  
    lax       IN BOOLEAN := false);
```

**注意：** このプロシージャはオーバーロードされています。list パラメータと tab パラメータは、同時には指定できません。



## パラメータ

表 19-2 ADD プロシージャのパラメータ

パラメータ	説明
name	メンバーを追加するリフレッシュ・グループの名前です。
list	リフレッシュ・グループに追加するマテリアライズド・ビューの、カンマ区切りのリストです。(シノニムはサポートされていません。)
tab	カンマ区切りのリストのかわりに、DBMS_UTILITY.UNCL_ARRAY タイプの PL/SQL 索引付き表を使用できます。この表の各要素は、マテリアライズド・ビューの名前になります。最初のマテリアライズド・ビューは 1 の位置に置かれます。最後の位置は NULL になります。
lax	マテリアライズド・ビューは、一度に 1 つのリフレッシュ・グループのみに所属できます。マテリアライズド・ビューを他のグループに移動する場合は、lax フラグを TRUE に設定する必要があります。マテリアライズド・ビューは自動的に移動され、移動先のグループのリフレッシュ間隔が適用されます。このフラグが FALSE に設定されている場合は、ADD のコールでエラー・メッセージが生成されます。

## CHANGE プロシージャ

このプロシージャは、リフレッシュ・グループのリフレッシュ間隔を変更します。

**関連項目：** リフレッシュ・グループの詳細は、『Oracle9i アドバンスト・レプリケーション』を参照してください。

## 構文

```
DBMS_REFRESH.CHANGE (
    name                IN VARCHAR2,
    next_date            IN DATE          := NULL,
    interval             IN VARCHAR2      := NULL,
    implicit_destroy     IN BOOLEAN       := NULL,
    rollback_seg         IN VARCHAR2      := NULL,
    push_deferred_rpc    IN BOOLEAN       := NULL,
    refresh_after_errors IN BOOLEAN       := NULL,
    purge_option         IN BINARY_INTEGER := NULL,
    parallelism          IN BINARY_INTEGER := NULL,
    heap_size            IN BINARY_INTEGER := NULL);
```

パラメータ

表 19-3 CHANGE プロシージャのパラメータ

パラメータ	説明
name	リフレッシュ間隔を変更するリフレッシュ・グループの名前です。
next_date	次にリフレッシュを行う日付です。デフォルトでは、この日付は変更されません。
interval	このファンクションを使用して、リフレッシュ・グループ内のマテリアライズド・ビューを次にリフレッシュするタイミングが計算されます。 <b>interval</b> は、リフレッシュの直前に評価されます。したがって、リフレッシュにかかる時間より長い間隔を選択する必要があります。デフォルトでは、 <b>interval</b> は変更されません。
implicit_destroy	<b>implicit_destroy</b> フラグの値をリセットできます。このフラグが設定されている場合、グループにメンバーが存在しなくなると、そのグループが自動的に削除されます。デフォルトでは、このフラグは変更されません。
rollback_seg	使用するロールバック・セグメントを変更できます。デフォルトでは、ロールバック・セグメントは変更されません。このパラメータをリセットしてデフォルトのロールバック・セグメントを使用するには、引用符を含めて <b>NULL</b> を指定します。引用符を含めずに <b>NULL</b> を指定すると、現在使用しているロールバック・セグメントが変更されません。
push_deferred_rpc	更新可能なマテリアライズド・ビューのみで使います。マテリアライズド・ビューをリフレッシュする前に、マテリアライズド・ビューから対応付けられたマスター表またはマスター・マテリアライズド・ビューに変更をプッシュする場合は、このパラメータを <b>TRUE</b> に設定します。 <b>FALSE</b> に設定されている場合は、変更が一時的に失われたように見えることがあります。デフォルトでは、このフラグは変更されません。
refresh_after_errors	更新可能なマテリアライズド・ビューのみで使います。マテリアライズド・ビューのマスター表またはマスター・マテリアライズド・ビューの <b>DEFERROR</b> ビューに、未解決の競合がログインされていてリフレッシュを行う場合は、このパラメータを <b>TRUE</b> に設定します。デフォルトでは、このフラグは変更されません。

表 19-3 CHANGE プロシージャのパラメータ（続き）

パラメータ	説明
purge_option	<p>パラレル伝播メカニズムを使用している場合（すなわち parallelism が 1 以上に設定されている場合）、このパラメータは次のようになります。</p> <ul style="list-style-type: none"><li>■ 0 = パージなし</li><li>■ 1 = レイジー・パージ（デフォルト）</li><li>■ 2 = アグレッシブ・パージ</li></ul> <p>ほとんどの場合、レイジー・パージが最適な設定になります。複数マスター・レプリケーション・グループが異なるターゲット・サイトにプッシュされ、1 つ以上のレプリケーション・グループに対する更新が不定期に行われるか不定期にプッシュされる場合は、アグレッシブ・パージに設定してキューを少なくします。すべてのレプリケーション・グループが不定期に更新およびプッシュされる場合は、パージなしに設定して、随時 PUSH を実行します。PUSH を実行するときは、アグレッシブ・パージに設定してキューを削減します。</p>
parallelism	<p>0（ゼロ）はシリアル伝播を指定します。</p> <p><math>n &gt; 1</math> は、<math>n</math> 個のパラレル処理のパラレル伝播を指定します。</p> <p>1 は、パラレル処理を 1 つのみ使用するパラレル伝播を指定します。</p>
heap_size	<p>パラレル伝播スケジュールのために同時に検査されるトランザクションの最大数です。最適なパフォーマンスを実現するデフォルト値が、自動的に計算されます。</p> <p><b>注意：</b>オラクル社カスタマ・サポート・センターから指示された場合を除き、このパラメータを設定しないでください。</p>

## DESTROY プロシージャ

このプロシージャは、リフレッシュ・グループからすべてのマテリアライズド・ビューを削除して、リフレッシュ・グループを削除します。

**関連項目：** リフレッシュ・グループの詳細は、『Oracle9i アドバンスト・レプリケーション』を参照してください。

### 構文

```
DBMS_REFRESH.DESTROY (  
    name    IN    VARCHAR2);
```

### パラメータ

表 19-4 DESTROY プロシージャのパラメータ

パラメータ	説明
name	破棄するリフレッシュ・グループの名前です。

## MAKE プロシージャ

このプロシージャは、リフレッシュ・グループ・メンバー、およびこのグループ・メンバーをリフレッシュする間隔を指定します。

**関連項目：** 詳細は、5-7 ページの手順 4「リフレッシュ・グループの作成」および『Oracle9i アドバンスト・レプリケーション』を参照してください。

### 構文

```
DBMS_REFRESH.MAKE (  
    name                IN          VARCHAR2  
    { list               IN          VARCHAR2,  
      | tab              IN          DBMS_UTILITY.UNCL_ARRAY, }  
    next_date           IN          DATE,  
    interval            IN          VARCHAR2,  
    implicit_destroy    IN          BOOLEAN      := false,  
    lax                 IN          BOOLEAN      := false,  
    job                 IN          BINARY_INTEGER := 0,  
    rollback_seg        IN          VARCHAR2     := NULL,  
    push_deferred_rpc    IN          BOOLEAN      := true,  
    refresh_after_errors IN          BOOLEAN      := false  
    purge_option         IN          BINARY_INTEGER := NULL,  
    parallelism         IN          BINARY_INTEGER := NULL,  
    heap_size           IN          BINARY_INTEGER := NULL);
```

**注意：** このプロシージャはオーバーロードされています。list パラメータと tab パラメータは、同時には指定できません。

パラメータ

表 19-5 MAKE プロシージャのパラメータ

パラメータ	説明
name	リフレッシュ・グループの識別に使用される、一意の名前です。リフレッシュ・グループは、表と同じ命名規則に従って名前付けをします。
list	リフレッシュするマテリアライズド・ビューの、カンマ区切りのリストです。(シノニムはサポートされていません。) これらのマテリアライズド・ビューは、異なるスキーマに置くことができ、また異なるマスター表またはマスター・マテリアライズド・ビューを持つことができます。ただし、すべてのマテリアライズド・ビューが、カレント・データベース内に存在する必要があります。
tab	カンマ区切りのリストのかわりに、DBMS_UTILITY.UNCL_ARRAY データ型を使用した PL/SQL 索引付き表で、リフレッシュするマテリアライズド・ビューの名前を指定できます。表にマテリアライズド・ビュー名が <i>n</i> 個ある場合、最初のマテリアライズド・ビューは 1 の位置に置かれ、 <i>n</i> + 1 の位置は NULL に設定されます。
next_date	次にリフレッシュを行う日付です。
interval	このファンクションを使用して、グループ内のマテリアライズド・ビューを次にリフレッシュするタイミングが計算されます。このフィールドは、next_date 値とともに使用します。  たとえば、interval に NEXT_DAY(SYSDATE+1, "MONDAY") を指定した場合、next_date が月曜と評価されると、毎週月曜日にマテリアライズド・ビューがリフレッシュされます。interval は、リフレッシュの直前に評価されます。したがって、リフレッシュにかかる時間より長い間隔を選択する必要があります。
implicit_destroy	リフレッシュ・グループにメンバーが存在しなくなったとき、リフレッシュ・グループを自動的に削除するには、このパラメータを TRUE に設定します。このフラグは、SUBTRACT プロシージャがコールされたときのみチェックされます。つまり、このフラグが設定されていても、空のリフレッシュ・グループを作成できます。

表 19-5 MAKE プロシージャのパラメータ（続き）

パラメータ	説明
lax	マテリアライズド・ビューは、一度に1つのリフレッシュ・グループのみに所属できます。マテリアライズド・ビューを既存のグループから新しいリフレッシュ・グループに移動する場合は、このパラメータを TRUE に設定する必要があります。マテリアライズド・ビューは自動的に移動され、移動先のグループのリフレッシュ間隔が適用されます。FALSE に設定されている場合は、MAKE のコールでエラー・メッセージが生成されます。
job	インポート・ユーティリティで必要です。デフォルトの 0（ゼロ）を使用してください。
rollback_seg	マテリアライズド・ビューのリフレッシュ時に使用するロールバック・セグメントの名前です。デフォルトの NULL の場合、デフォルトのロールバック・セグメントが使用されます。
push_deferred_rpc	更新可能なマテリアライズド・ビューのみで使します。マテリアライズド・ビューをリフレッシュする前に、対応付けられたマスター表またはマスター・マテリアライズド・ビューに、マテリアライズド・ビューから変更をプッシュする場合は、デフォルト値 TRUE を使します。FALSE に設定されている場合は、変更が一時的に失われたように見えることがあります。
refresh_after_errors	更新可能なマテリアライズド・ビューのみで使します。マテリアライズド・ビューのマスター表またはマスター・マテリアライズド・ビューの DEFERROR ビューに、未解決の競合がログインされていてもリフレッシュを行う場合は、このパラメータを 0（ゼロ）に設定します。
purge_option	<p>パラレル伝播メカニズムを使用している場合（すなわち parallelism が 1 以上に設定されている場合）、0 はページなし、1 はレイジー・ページ（デフォルト）、2 はアグレッシブ・ページになります。ほとんどの場合、レイジー・ページが最適な設定になります。</p> <p>複数マスター・レプリケーション・グループが異なるターゲット・サイトにプッシュされ、1 つ以上のレプリケーション・グループに対する更新が不定期に行われるか不定期にプッシュされる場合は、アグレッシブ・ページに設定してキューを少なくします。すべてのレプリケーション・グループが不定期に更新およびプッシュされる場合は、ページなしに設定して、随時 PUSH を実行します。PUSH を実行するときは、アグレッシブ・ページに設定してキューを削減します。</p>
parallelism	<p>0（ゼロ）はシリアル伝播を指定します。</p> <p><math>n &gt; 1</math> は、<math>n</math> 個のパラレル処理のパラレル伝播を指定します。</p> <p>1 は、パラレル処理を 1 つのみ使用するパラレル伝播を指定します。</p>

表 19-5 MAKE プロシージャのパラメータ（続き）

パラメータ	説明
heap_size	パラレル伝播スケジュールのために同時に検査されるトランザクションの最大数です。最適なパフォーマンスを実現するデフォルト値が、自動的に計算されます。  <b>注意：</b> オラクル社カスタマ・サポート・センターから指示された場合を除き、このパラメータを設定しないでください。

## REFRESH プロシージャ

このプロシージャは、リフレッシュ・グループを手動でリフレッシュします。

**関連項目：** リフレッシュ・グループの詳細は、『Oracle9i アドバンスト・レプリケーション』を参照してください。

## 構文

```
DBMS_REFRESH.REFRESH (  
    name    IN    VARCHAR2);
```

## パラメータ

表 19-6 REFRESH プロシージャのパラメータ

パラメータ	説明
name	手動でリフレッシュするリフレッシュ・グループの名前です。

## SUBTRACT プロシージャ

このプロシージャは、マテリアライズド・ビューをリフレッシュ・グループから削除します。

**関連項目：** リフレッシュ・グループの詳細は、『Oracle9i アドバンスト・レプリケーション』を参照してください。

### 構文

```
DBMS_REFRESH.SUBTRACT (  
    name          IN    VARCHAR2,  
    { list        IN    VARCHAR2,  
      | tab       IN    DBMS_UTILITY.UNCL_ARRAY, }  
    lax           IN    BOOLEAN := false);
```

**注意：** このプロシージャはオーバーロードされています。list パラメータと tab パラメータは、同時には指定できません。

### パラメータ

表 19-7 SUBTRACT プロシージャのパラメータ

パラメータ	説明
name	メンバーを削除するリフレッシュ・グループの名前です。
list	リフレッシュ・グループから削除するマテリアライズド・ビューの、カンマ区切りのリストです。(シノニムはサポートされていません。) これらのマテリアライズド・ビューは、異なるスキーマに置くことができ、また異なるマスター表またはマスター・マテリアライズド・ビューを持つことができます。ただし、すべてのマテリアライズド・ビューが、カレント・データベース内に存在する必要があります。
tab	カンマ区切りのリストのかわりに、DBMS_UTILITY.UNCL_ARRAY データ型を使用した PL/SQL 索引付き表で、リフレッシュするマテリアライズド・ビューの名前を指定できます。表にマテリアライズド・ビュー名が <i>n</i> 個ある場合、最初のマテリアライズド・ビューは 1 の位置に置かれ、 <i>n</i> + 1 の位置は NULL に設定されます。
lax	リフレッシュ・グループ・メンバーではないマテリアライズド・ビューの削除を試みたときに、エラー・メッセージが生成されるようにするには、このパラメータを FALSE に設定します。



DBMS\_REPCAT は、レプリケーション・カタログと環境を管理および更新するルーチンを提供します。

この章では、次の項目を説明します。

- [DBMS\\_REPCAT サブプログラムの概要](#)

## DBMS\_REPCAT サブプログラムの概要

表 20-1 DBMS\_REPCAT パッケージのサブプログラム

サブプログラム	説明
<a href="#">20-7 ページ「ADD_GROUPED_COLUMN プロシージャ」</a>	既存の列グループにメンバーを追加します。
<a href="#">20-8 ページ「ADD_MASTER_DATABASE プロシージャ」</a>	レプリケーション環境に他のマスター・サイトを追加します。
<a href="#">20-10 ページ「ADD_NEW_MASTERS プロシージャ」</a>	DBA_REPSITES_NEW データ・ディクショナリ・ビューのマスター・サイトを、使用可能な全マスター・サイトのレプリケーション・カタログに追加します。
<a href="#">20-16 ページ「ADD_PRIORITY_datatype プロシージャ」</a>	優先グループにメンバーを追加します。
<a href="#">20-17 ページ「ADD_SITE_PRIORITY_SITE プロシージャ」</a>	サイト優先グループに新しいサイトを追加します。
<a href="#">20-18 ページ「ADD_conflicttype_RESOLUTION プロシージャ」</a>	更新、削除または一意性競合のための、解消方法を指定します。
<a href="#">20-23 ページ「ALTER_CATCHUP_PARAMETERS プロシージャ」</a>	DBA_REPEXTENSIONS データ・ディクショナリ・ビューに保存されたパラメータの値を変更します。
<a href="#">20-25 ページ「ALTER_MASTER_PROPAGATION プロシージャ」</a>	指定されたマスター・サイトの、指定されたレプリケーション・グループの伝播方法を変更します。
<a href="#">20-27 ページ「ALTER_MASTER_REPOBJECT プロシージャ」</a>	レプリケーション環境のオブジェクトを変更します。
<a href="#">20-30 ページ「ALTER_MVIEW_PROPAGATION プロシージャ」</a>	カレント・マテリアライズド・ビュー・サイトの、指定されたレプリケーション・グループの伝播方法を変更します。
<a href="#">20-31 ページ「ALTER_PRIORITY プロシージャ」</a>	指定された優先グループ・メンバーに対応付けられた優先順位レベルを変更します。
<a href="#">20-33 ページ「ALTER_PRIORITY_datatype プロシージャ」</a>	優先グループ・メンバーの値を変更します。
<a href="#">20-34 ページ「ALTER_SITE_PRIORITY プロシージャ」</a>	指定されたサイトに対応付けられた優先順位レベルを変更します。
<a href="#">20-36 ページ「ALTER_SITE_PRIORITY_SITE プロシージャ」</a>	指定された優先順位レベルに対応付けられたサイトを変更します。

表 20-1 DBMS\_REPCAT パッケージのサブプログラム（続き）

サブプログラム	説明
20-37 ページ「 <a href="#">CANCEL_STATISTICS</a> プロシージャ」	正常に実行された、表の更新、一意性および削除の競合解消の統計の収集を中止します。
20-38 ページ「 <a href="#">COMMENT_ON_COLUMN_GROUP</a> プロシージャ」	列グループの <code>ALL_REPCOLUMN_GROUP</code> ビュー内のコメント・フィールドを更新します。
20-45 ページ「 <a href="#">COMMENT_ON_conflicttype_RESOLUTION</a> プロシージャ」	マテリアライズド・ビュー・サイトの <code>ALL_REPGROUP</code> ビュー内の <code>SCHEMA_COMMENT</code> フィールドを更新します。
20-40 ページ「 <a href="#">COMMENT_ON_PRIORITY_GROUP/COMMENT_ON_SITE_PRIORITY</a> プロシージャ」	(サイト) 優先グループの <code>ALL_REPPRIORITY_GROUP</code> ビュー内のコメント・フィールドを更新します。
20-41 ページ「 <a href="#">COMMENT_ON_REPGROUP</a> プロシージャ」	マスター・グループの <code>ALL_REPGROUP</code> ビュー内のコメント・フィールドを更新します。
20-42 ページ「 <a href="#">COMMENT_ON_REPOBJECT</a> プロシージャ」	レプリケート・オブジェクトの <code>ALL_REPOBJECT</code> ビュー内のコメント・フィールドを更新します。
20-43 ページ「 <a href="#">COMMENT_ON_REPSITES</a> プロシージャ」	レプリケート・サイトの <code>ALL_REPSITE</code> ビュー内のコメント・フィールドを更新します。
20-45 ページ「 <a href="#">COMMENT_ON_conflicttype_RESOLUTION</a> プロシージャ」	競合解消ルーチンの <code>ALL_REPRESOLUTION</code> ビュー内のコメント・フィールドを更新します。
20-47 ページ「 <a href="#">COMPARE_OLD_VALUES</a> プロシージャ」	更新および削除のために、各マスター・サイトの古い列値をレプリケート表の非キー列と比較するかどうかを指定します。
20-49 ページ「 <a href="#">CREATE_MASTER_REPGROUP</a> プロシージャ」	空の静止したマスター・グループを新規に作成します。
20-50 ページ「 <a href="#">CREATE_MASTER_REPOBJECT</a> プロシージャ」	オブジェクトがレプリケート・オブジェクトであると指定します。
20-54 ページ「 <a href="#">CREATE_MVIEW_REPGROUP</a> プロシージャ」	ローカル・データベースに、空のマテリアライズド・ビュー・グループを新規に作成します。
20-55 ページ「 <a href="#">CREATE_MVIEW_REPOBJECT</a> プロシージャ」	マテリアライズド・ビュー・グループにレプリケート・オブジェクトを追加します。
20-59 ページ「 <a href="#">DEFINE_COLUMN_GROUP</a> プロシージャ」	空の列グループを作成します。
20-60 ページ「 <a href="#">DEFINE_PRIORITY_GROUP</a> プロシージャ」	マスター・グループの新しい優先グループを作成します。

表 20-1 DBMS\_REPCAT パッケージのサブプログラム（続き）

サブプログラム	説明
20-61 ページ「 <a href="#">DEFINE_SITE_PRIORITY プロシージャ</a> 」	マスター・グループの新しいサイト優先グループを作成します。
20-62 ページ「 <a href="#">DO_DEFERRED_REPCAT_ADMIN プロシージャ</a> 」	カレント・マスター・サイトの指定されたマスター・グループ、またはすべてのマスター・サイトで、ローカルな遅延管理プロシージャを実行します。
20-63 ページ「 <a href="#">DROP_COLUMN_GROUP プロシージャ</a> 」	列グループを削除します。
20-64 ページ「 <a href="#">DROP_GROUPED_COLUMN プロシージャ</a> 」	列グループからメンバーを削除します。
20-66 ページ「 <a href="#">DROP_MASTER_REPGROUP プロシージャ</a> 」	カレント・サイトからマスター・グループを削除します。
20-67 ページ「 <a href="#">DROP_MASTER_REPOBJECT プロシージャ</a> 」	マスター・グループからレプリケート・オブジェクトを削除します。
20-71 ページ「 <a href="#">DROP_PRIORITY プロシージャ</a> 」	マスター・グループからレプリケート・オブジェクトを削除します。
20-69 ページ「 <a href="#">DROP_MVIEW_REPGROUP プロシージャ</a> 」	レプリケーション環境からマテリアライズド・ビュー・サイトを削除します。
20-70 ページ「 <a href="#">DROP_MVIEW_REPOBJECT プロシージャ</a> 」	マテリアライズド・ビュー・サイトからレプリケート・オブジェクトを削除します。
20-71 ページ「 <a href="#">DROP_PRIORITY プロシージャ</a> 」	優先順位レベルに基づいて、優先グループ・メンバーを削除します。
20-72 ページ「 <a href="#">DROP_PRIORITY_GROUP プロシージャ</a> 」	指定されたマスター・グループの優先グループを削除します。
20-73 ページ「 <a href="#">DROP_PRIORITY_datatype プロシージャ</a> 」	値に基づいて、優先グループ・メンバーを削除します。
20-75 ページ「 <a href="#">DROP_SITE_PRIORITY プロシージャ</a> 」	指定されたマスター・グループのサイト優先グループを削除します。
20-76 ページ「 <a href="#">DROP_SITE_PRIORITY_SITE プロシージャ</a> 」	サイト優先グループから、指定された名前のサイトを削除します。
20-77 ページ「 <a href="#">DROP_conflicttype_RESOLUTION プロシージャ</a> 」	更新、削除または一意性競合解消メソッドを削除します。
20-79 ページ「 <a href="#">EXECUTE_DDL プロシージャ</a> 」	各マスター・サイトで実行される DDL を渡します。

表 20-1 DBMS\_REPCAT パッケージのサブプログラム（続き）

サブプログラム	説明
20-80 ページ「 <a href="#">GENERATE_MVIEW_SUPPORT</a> プロシージャ」	トリガーをアクティブ化して、更新可能なマテリアライズド・ビューのレプリケーションやプロシージャ・レプリケーションのサポートに必要なパッケージを生成します。
20-82 ページ「 <a href="#">GENERATE_REPLICATION_SUPPORT</a> プロシージャ」	指定したオブジェクトのレプリケーションのサポートに必要なトリガー、パッケージおよびプロシージャを生成します。
20-84 ページ「 <a href="#">MAKE_COLUMN_GROUP</a> プロシージャ」	1 つ以上のメンバーを持つ新しい列グループを作成します。
20-85 ページ「 <a href="#">PREPARE_INSTANTIATED_MASTER</a> プロシージャ」	マスター・グループに追加するデータベースのグローバル名を変更します。
20-87 ページ「 <a href="#">PURGE_MASTER_LOG</a> プロシージャ」	指定された ID 番号、ソースまたはマスター・グループに対応付けられた、DBA_REPCATLOG 内のローカル・メッセージを削除します。
20-88 ページ「 <a href="#">PURGE_STATISTICS</a> プロシージャ」	ALL_REPRESOLUTION_STATISTICS ビューから情報を削除します。
20-89 ページ「 <a href="#">REFRESH_MVIEW_REPGROUP</a> プロシージャ」	マテリアライズド・ビュー・グループを、対応付けられたマスター・サイトまたはマスター・マテリアライズド・ビュー・サイトからの最新データでリフレッシュします。
20-91 ページ「 <a href="#">REGISTER_MVIEW_REPGROUP</a> プロシージャ」	DBA_REGISTERED_MVIEW_GROUPS でのマテリアライズド・ビューの挿入、変更または削除を行って、マスター・サイトまたはマスター・マテリアライズド・ビュー・サイトでのマテリアライズド・ビューの管理を容易にします。
20-92 ページ「 <a href="#">REGISTER_STATISTICS</a> プロシージャ」	正常に実行された、表の更新、削除および一意性の競合解消の統計を収集します。
20-93 ページ「 <a href="#">RELOCATE_MASTERDEF</a> プロシージャ」	マスター定義サイトを、レプリケーション環境の他のマスター・サイトに変更します。
20-94 ページ「 <a href="#">REMOVE_MASTER_DATABASES</a> プロシージャ」	レプリケーション環境から、1 つ以上のマスター・データベースを削除します。
20-96 ページ「 <a href="#">RENAME_SHADOW_COLUMN_GROUP</a> プロシージャ」	レプリケート表のシャドウ列グループの名前を変更して、名前付き列グループにします。

表 20-1 DBMS\_REPCAT パッケージのサブプログラム（続き）

サブプログラム	説明
20-97 ページ「 <a href="#">REPCAT_IMPORT_CHECK プロシージャ</a> 」	レプリケート・オブジェクト、またはアドバンスド・レプリケーション機能で使用されたオブジェクトのエクスポートまたはインポート後に、マスター・グループのオブジェクトが適切なオブジェクト識別子およびステータス値を持つようにします。
20-98 ページ「 <a href="#">RESUME_MASTER_ACTIVITY プロシージャ</a> 」	レプリケーション環境を静止した後で、通常のレプリケーション・アクティビティを再開します。
20-99 ページ「 <a href="#">RESUME_PROPAGATION_TO_MDEF プロシージャ</a> 」	エクスポートが終了し、マスター・サイトに存在する拡張レプリケーション・グループおよび影響を受けないレプリケーション・グループの両方の伝播を有効にできるようになったことを示します。
20-100 ページ「 <a href="#">SEND_OLD_VALUES プロシージャ</a> 」	更新および削除のために、レプリケート表のキー列ではない古い列値を送信するかどうかを指定します。
20-102 ページ「 <a href="#">SET_COLUMNS プロシージャ</a> 」	主キーのかわりに代替列または列グループを使用して、行レベル・レプリケーションを使用する際に比較する表中の列を決定するよう指定します。
20-104 ページ「 <a href="#">SPECIFY_NEW_MASTERS プロシージャ</a> 」	既存のレプリケーション・グループに追加したいマスター・サイトを、グループを静止せずに指定します。
20-106 ページ「 <a href="#">SUSPEND_MASTER_ACTIVITY プロシージャ</a> 」	マスター・グループのレプリケーション・アクティビティを一時停止します。
20-107 ページ「 <a href="#">SWITCH_MVIEW_MASTER プロシージャ</a> 」	マテリアライズド・ビュー・グループのマスター・サイトを他のマスター・サイトに変更します。
20-109 ページ「 <a href="#">UNDO_ADD_NEW_MASTERS_REQUEST プロシージャ</a> 」	指定した <code>extension_id</code> の、 <code>SPECIFY_NEW_MASTERS</code> プロシージャおよび <code>ADD_NEW_MASTERS</code> プロシージャによるすべての変更を元に戻します。
20-111 ページ「 <a href="#">UNREGISTER_MVIEW_REPGROUP プロシージャ</a> 」	<code>DBA_REGISTERED_MVIEW_GROUPS</code> でのマテリアライズド・ビューの挿入、変更または削除を行って、マスター・サイトおよびマスター・マテリアライズド・ビュー・サイトでのマテリアライズド・ビューの管理を容易にします。
20-111 ページ「 <a href="#">VALIDATE ファンクション</a> 」	マルチマスター・レプリケーション環境のキー条件が正確であるかどうかを確認します。
20-114 ページ「 <a href="#">WAIT_MASTER_LOG プロシージャ</a> 」	マスター・サイトに非同期で伝播された変更の適用状況を判定します。

## ADD\_GROUPED\_COLUMN プロシージャ

このプロシージャは、既存の列グループにメンバーを追加します。このプロシージャは、マスター定義サイトからコールする必要があります。

### 構文

```
DBMS_REPCAT.ADD_GROUPED_COLUMN (  
    sname                IN    VARCHAR2,  
    oname                IN    VARCHAR2,  
    column_group         IN    VARCHAR2,  
    list_of_column_names IN    VARCHAR2 | DBMS_REPCAT.VARCHAR2s);
```

### パラメータ

表 20-2 ADD\_GROUPED\_COLUMN プロシージャのパラメータ

パラメータ	説明
sname	レプリケート表が置かれたスキーマの名前です。
oname	列グループが対応付けられているレプリケート表の名前です。表はネストした表の記憶表を利用できます。
column_group	メンバーを追加する列グループの名前です。
list_of_column_names	指定された列グループに追加する列の名前です。列名は、カンマ区切りのリストと PL/SQL 索引付き表のいずれでも指定できます。 PL/SQL 索引付き表のタイプは DBMS_REPCAT.VARCHAR2 にする必要があります。表のすべての列を含む列グループを作成するには、単一の値 '*' を使用します。  列オブジェクトは指定できますが、列オブジェクトの属性は指定できません。  表がオブジェクトである場合、SYS_NC_OID\$ を指定して列グループにオブジェクト識別子列を追加できます。この列は各行オブジェクトのオブジェクト識別子を追跡します。  表がネストした表の記憶表である場合は、NESTED_TABLE_ID を指定して、ネストした表の各行の識別子を追跡する列を追加できます。

表 20-3 ADD\_GROUPED\_COLUMN プロシージャの例外

例外	説明
nonmasterdef	起動サイトはマスター定義サイトではありません。
missingobject	指定された表は存在しません。
missinggroup	指定された列グループは存在しません。
missingcolumn	指定された列は、指定された表に存在しません。
duplicatecolumn	指定された列は、すでに他の列グループのメンバーになっています。
missingschema	指定されたスキーマは存在しません。
notquiesced	指定された表が所属するレプリケーション・グループは静止していません。

ADD\_MASTER\_DATABASE プロシージャ

このプロシージャは、レプリケーション環境に他のマスター・サイトを追加します。このプロシージャは、既存のマスター・サイトで、すべてのトリガーとそれらに対応付けられたパッケージを再生成します。このプロシージャは、マスター定義サイトからコールする必要があります。

構文

```
DBMS_REPCAT.ADD_MASTER_DATABASE (  
  gname          IN   VARCHAR2,  
  master         IN   VARCHAR2,  
  use_existing_objects IN  BOOLEAN := true,  
  copy_rows      IN   BOOLEAN := true,  
  comment        IN   VARCHAR2 := '',  
  propagation_mode IN  VARCHAR2 := 'ASYNCHRONOUS',  
  fname         IN   VARCHAR2 := NULL);
```



# パラメータ

表 20-4 ADD\_MASTER\_DATABASE プロシージャのパラメータ

パラメータ	説明
gname	レプリケートされるレプリケーション・グループの名前です。このレプリケーション・グループは、マスター定義サイトに存在している必要があります。
master	新しいマスター・データベースの、完全修飾されたデータベース名です。
use_existing_objects	新しいマスター・サイトのスキーマにすでに存在する同じタイプで同じ shape のオブジェクトを再利用する場合は、TRUE を指定します。
copy_rows	新しいマスター・サイトの表の初期の内容を、マスター定義サイトの表の内容と一致させる場合は、TRUE を指定します。
comment	このコメントは DBA_REPSITES ビューの MASTER_COMMENT フィールドに追加されます。
propagation_mode	新しいマスター・データベースに変更を送る、または新しいマスター・データベースから変更を受け取る方法です。使用できる値は synchronous および asynchronous です。
fname	このパラメータは内部で使します。  <b>注意：</b> オラクル社カスタマ・サポート・センターから指示された場合を除き、このパラメータを設定しないでください。

# 例外

表 20-5 ADD\_MASTER\_DATABASE プロシージャの例外

例外	説明
nonmasterdef	起動サイトはマスター定義サイトではありません。
notquiesced	マスター・グループのレプリケーションは中断されていません。
missingrepgroup	レプリケーション・グループは、指定されたデータベース・サイトに存在しません。
commfailure	新しいマスターにアクセスできません。
typefailure	指定された伝播モードは正しくありません。
notcompat	互換モードは 7.3.0.0 以上である必要があります。
duplrepgrp	マスター・サイトはすでに存在しています。

## ADD\_NEW\_MASTERS プロシージャ

このプロシージャは、DBA\_REPSITES\_NEW のデータ・ディクショナリ・ビューのマスター・サイトを、SPECIFY\_NEW\_MASTERS プロシージャの実行時に指定されたマスター・グループに追加します。これらの新しいマスター・サイトに関する情報は、使用可能な全マスター・サイトのレプリケーション・カタログに追加されます。

この時点では、オブジェクトレベルのエクスポートまたはインポートでインスタンス化された全マスター・サイトがアクセス可能である必要があります。新しいレプリケーション・グループは、静止中状態で追加されます。全データベースのエクスポートやインポートを通して、または変更ベースのリカバリを通してインスタンス化されたマスター・サイトは、アクセス可能である必要はありません。

このプロシージャは、SPECIFY\_NEW\_MASTERS プロシージャを実行した後に実行します。

**注意：** このプロシージャの実行後は、新しいマスター・サイトが追加されるまで遅延トランザクション・キューの伝播を無効または有効にしないでください。DBA\_REPEXTENSIONS データ・ディクショナリ・ビューは、伝播を無効または有効にする前に消去しておく必要があります。レプリケーション管理ツール、または DBMS\_DEFER\_SYS パッケージの SET DISABLED プロシージャを使用して、伝播を無効または有効にできます。

**関連項目：**

- 20-104 ページ「SPECIFY\_NEW\_MASTERS プロシージャ」
- マスター・グループへのマスター・サイトの追加の詳細は、7-4 ページの「新しいマスター・サイトの追加」を参照してください。

## 構文

```
DBMS_REPCAT.ADD_NEW_MASTERS (  
    export_required                IN      BOOLEAN,  
    { available_master_list       IN      VARCHAR2,  
      | available_master_table   IN      DBMS_UTILITY.DBLINK_ARRAY, }  
    masterdef_flashback_scn       OUT     NUMBER,  
    extension_id                  OUT     RAW,  
    break_trans_to_masterdef      IN      BOOLEAN := false,  
    break_trans_to_new_masters   IN      BOOLEAN := false,  
    percentage_for_catchup_mdef  IN      BINARY_INTEGER := 100,  
    cycle_seconds_mdef           IN      BINARY_INTEGER := 60,  
    percentage_for_catchup_new   IN      BINARY_INTEGER := 100,  
    cycle_seconds_new            IN      BINARY_INTEGER := 60);
```

**注意：** このプロシージャはオーバーロードされています。  
available\_master\_list パラメータと available\_master\_table  
パラメータは、同時には指定できません。

パラメータ

表 20-6 ADD\_NEW\_MASTERS プロシージャのパラメータ

パラメータ	説明
export_required	新しいマスター・サイトのうち少なくとも1つにオブジェクトレベルまたは全データベースのエクスポートが必要な場合、TRUE に設定します。新しいマスター・サイトすべてに変更ベースのリカバリを使用している場合、FALSE に設定します。
available_master_list	オブジェクトレベルのエクスポートまたはインポートを使用してインスタンス化される新しいマスター・サイトの、カンマ区切りのリストです。リストされたサイトは、SPECIFY_NEW_MASTERS プロシージャで指定されたサイトと一致する必要があります。既存のマスター・サイトは含めず、新しいマスター・サイトのみをリストします。各サイト名の間に空白は入れないでください。  すべてのマスターが全データベースのエクスポートやインポートまたは変更ベースのリカバリを使用してインスタンス化される場合は、NULL を指定します。
available_master_table	オブジェクトレベルのエクスポートまたはインポートを使用してインスタンス化される新しいマスター・サイトをリストした表です。表中のサイトは SPECIFY_NEW_MASTERS プロシージャで指定されたサイトと一致する必要があります。全データベースのエクスポートやインポートまたは変更ベースのリカバリを使用してインスタンス化されるマスターは、指定しないでください。  オブジェクトレベルのエクスポートまたはインポートを使用してインスタンス化されるマスター・サイトをリストする表には、拡張されるマスター・グループにおける新しいマスター・サイトのみをリストします。拡張されるマスター・グループの既存のマスター・サイトはリストしないでください。最初のマスター・サイトは 1 の位置、2 番目のマスターは 2 の位置と、順に配置します。

表 20-6 ADD\_NEW\_MASTERS プロシージャのパラメータ (続き)

パラメータ	説明
masterdef_flashback_scn	この OUT パラメータは、エクスポートまたは変更ベースのリカバリの最中に使用する必要のあるシステム変更番号 (SCN) を戻します。エクスポートを実行する際は、このパラメータに戻された値を FLASHBACK_SCN エクスポート・パラメータに使用します。flashback_scn の値は、DBA_REPEXTENSIONS データ・ディクショナリ・ビューを問い合せて検索できます。
extension_id	この OUT パラメータは現在保留になっている要求の識別子を戻して、静止せずにマスター・データベースを追加します。extension_id は、DBA_REPSITES_NEW データ・ディクショナリ・ビューおよび DBA_REPEXTENSIONS データ・ディクショナリ・ビューを問い合せて検索できます。
break_trans_to_masterdef	<p>このパラメータは export_required が TRUE に設定されている場合のみ有効です。</p> <p>break_trans_to_masterdef が TRUE に設定されている場合、既存のマスターは遅延トランザクションを、マスター・サイトを追加していないレプリケーション・グループのマスター定義サイトに伝播し続ける場合があります。マスター・サイトを追加するレプリケーション・グループの遅延トランザクションは、エクスポートが完了するまで伝播させることはできません。</p> <p>各遅延トランザクションは、1 つ以上のリモート・プロシージャ・コール (RPC) からなっています。この値が FALSE に設定されており、影響を受けないマスター・グループおよび拡張されているマスター・グループの両方のオブジェクトを参照するトランザクションが発生した場合、トランザクションは 2 つに分割されて、2 つの別個のトランザクションとして異なる時間に接続先に送られる場合があります。このようなトランザクションを分割トランザクションといいます。分割トランザクションが可能な場合は、この動作によって破られる可能性のある整合性制約を、新しいマスター・サイトが追加されるまで無効にしておく必要があります。</p> <p>break_trans_to_masterdef が FALSE に設定されている場合、既存のマスターは遅延トランザクションをマスター定義サイトに伝播することはできません。</p>

表 20-6 ADD\_NEW\_MASTERS プロシージャのパラメータ (続き)

パラメータ	説明
break_trans_to_new_masters	<p>break_trans_to_new_masters が TRUE に設定されている場合、既存のマスターは遅延トランザクションを、マスター・サイトを追加していないレプリケーション・グループの新しいマスター・サイトに伝播し続ける場合があります。</p> <p>各遅延トランザクションは、1 つ以上のリモート・プロシージャ・コール (RPC) からなっています。これが TRUE に設定されており、影響を受けないマスター・グループおよび拡張されているマスター・グループの両方のオブジェクトを参照するトランザクションが発生した場合、トランザクションは 2 つに分割されて、2 つの別個のトランザクションとして異なる時間に接続先に送られる場合があります。このようなトランザクションを分割トランザクションといいます。分割トランザクションが可能な場合は、この動作によって破られる可能性のある整合性制約を、新しいマスター・サイトが追加されるまで無効にしておく必要があります。</p> <p>break_trans_to_new_masters が FALSE に設定されている場合、遅延トランザクション・キューの新しいマスターへの伝播は無効になります。</p>
percentage_for_catchup_mdef	<p>このパラメータは export_required および break_trans_to_masterdef がともに TRUE に設定されている場合のみ有効です。</p> <p>マスター定義サイトへの伝播に追いつくために使用する必要のある、伝播リソースのパーセンテージです。10 の倍数で、かつ 0 以上 100 以下の数値にする必要があります。</p>
cycle_seconds_mdef	<p>このパラメータは percentage_for_catchup_mdef がともに有効で、10 以上 90 以下の値に設定されている場合に有効です。この場合、masterdef への伝播は、拡張されていないレプリケーション・グループと拡張されているレプリケーション・グループに対して、各サイクル中にプッシュが行われるたびに、交互に実行されます。このパラメータはサイクルの長さを秒単位で示します。</p>
percentage_for_catchup_new	<p>このパラメータは break_trans_to_new_masters が TRUE に設定されている場合のみ有効です。</p> <p>新しいマスター・サイトへの伝播に追いつくために使用する必要のある、伝播リソースのパーセンテージです。10 の倍数で、かつ 0 以上 100 以下の数値にする必要があります。</p>

表 20-6 ADD\_NEW\_MASTERS プロシージャのパラメータ（続き）

パラメータ	説明
cycle_seconds_new	このパラメータは <code>percentage_for_catchup_new</code> がとも に有効で、10 以上 90 以下の値に設定されている場合に有 効です。この場合、新しいマスター・サイトへの伝播は、拡 張されていないレプリケーション・グループと拡張されて いるレプリケーション・グループの間で交互に切り替わり ます。1 サイクルごとに各グループに 1 回データがプッシュ されます。このパラメータはサイクルの長さを秒単位で示 します。

例外

表 20-7 ADD\_NEW\_MASTERS プロシージャの例外

例外	説明
nonmasterdef	起動サイトはマスター定義サイトではありません。
typefailure	パラメータの 1 つに指定されたパラメータ値が不適正です。
novalidextreq	有効な拡張要求がありません。 <code>extension_id</code> は使用できません。
nonewsites	指定された拡張要求に追加する新しいマスター・サイトがありませ ん。
notanewsites	拡張要求のための新しいサイトではありません。 <code>SPECIFY_NEW_MASTERS</code> プロシージャを実行した際に指定されな かったサイトが指定されました。
dbnotcompatible	機能がデータベースのバージョンと互換性がありません。データベー スの互換性のレベルは、すべて 9.0.1 以上である必要があります。

## 使用上の注意

変更ベースのリカバリまたはデータベース全体のエクスポートやインポートを使用してインスタンス化される新しいマスター・サイトには、次の条件が適用されます。

- 新しいマスター・サイトは既存のレプリケーション・グループを持つことはできません。
- マスター定義サイトはマテリアライズド・ビュー・グループを持つことはできません。
- マスター定義サイトはすべてのマスター・グループについて同じである必要があります。1 つ以上のマスター・グループが異なるマスター定義サイトを持つ場合は、変更ベースのリカバリやデータベース全体のエクスポートまたはインポートは使用しないでください。その場合、かわりにオブジェクトレベルのエクスポートまたはインポートを使用します。
- 新しいマスター・サイトは、拡張プロセスの完了時には、マスター定義サイトのすべてのレプリケーション・グループを含んでいる必要があります。すなわち、新しいマスター・サイトにマスター定義サイトのマスター・グループのサブセットを追加することはできません。すべてのグループを追加する必要があります。

---

---

**注意：** 変更ベースのリカバリを使用するには、既存のマスター・サイトおよび新しいマスター・サイトを同じオペレーティング・システム上で実行する必要があります。ただしオペレーティング・システムのリリースは同じでなくても構いません。

---

---

オブジェクトレベルのエクスポートまたはインポートの際は、インポートの前に拡張グループの DBA\_REPCATLOG データ・ディクショナリ・ビューの全要求がエラーなしに処理されたことを確認してください。

## ADD\_PRIORITY\_datatype プロシージャ

このプロシージャは、優先グループにメンバーを追加します。このプロシージャは、マスター定義サイトからコールする必要があります。コールするプロシージャは、`priority` 列のデータ型によって異なります。`priority` 列の指定するすべての値について、このプロシージャを1回ずつコールする必要があります。

**関連項目：** 競合解消メソッドの詳細は、[第 6 章「競合解消の構成」](#)および『[Oracle9i アドバンスド・レプリケーション](#)』を参照してください。

### 構文

```
DBMS_REPCAT.ADD_PRIORITY_datatype (  
    gname             IN   VARCHAR2,  
    pgroup            IN   VARCHAR2,  
    value              IN   datatype,  
    priority           IN   NUMBER);
```

`datatype` は次のとおりです。

```
{ NUMBER  
| VARCHAR2  
| CHAR  
| DATE  
| RAW  
| NCHAR  
| NVARCHAR2 }
```

### パラメータ

表 20-8 ADD\_PRIORITY\_datatype プロシージャのパラメータ

パラメータ	説明
gname	優先グループを作成するマスター・グループの名前です。
pgroup	優先グループの名前です。
value	優先グループ・メンバーの値です。この優先グループを使用する表に対応付けられた <code>priority</code> 列の値の1つになります。
priority	この値の優先順位です。数値が高いほど、優先順位が高くなります。



例外

表 20-9 ADD\_PRIORITY\_datatype プロシージャの例外

例外	説明
nonmasterdef	起動サイトはマスター定義サイトではありません。
duplicatevalue	指定された値は、優先グループにすでに存在しています。
duplicatepriority	指定された優先順位は、優先グループにすでに存在しています。
missingregroup	指定されたマスター・グループは存在しません。
missingprioritygroup	指定された優先グループは存在しません。
typefailure	指定された値のデータ型は、この優先グループには不適切です。
notquiesced	指定されたマスター・グループは静止していません。

ADD\_SITE\_PRIORITY\_SITE プロシージャ

このプロシージャは、サイト優先グループに新しいサイトを追加します。このプロシージャは、マスター定義サイトからコールする必要があります。

**関連項目：** 競合解消メソッドの詳細は、第 6 章「競合解消の構成」および『Oracle9i アドバンスト・レプリケーション』を参照してください。

構文

```
DBMS_REPCAT.ADD_SITE_PRIORITY_SITE (  
  gname          IN   VARCHAR2,  
  name           IN   VARCHAR2  
  site           IN   VARCHAR2,  
  priority       IN   NUMBER);
```

パラメータ

表 20-10 ADD\_SITE\_PRIORITY\_SITE プロシージャのパラメータ

パラメータ	説明
gname	グループにサイトを追加するマスター・グループの名前です。
name	メンバーを追加するサイト優先グループの名前です。
site	追加するサイトのグローバル・データベース名です。
priority	追加するサイトの優先順位レベルです。数値が高いほど、優先順位レベルが高くなります。

例外

表 20-11 ADD\_SITE\_PRIORITY\_SITE プロシージャの例外

例外	説明
nonmasterdef	起動サイトはマスター定義サイトではありません。
missingreggroup	指定されたマスター・グループは存在しません。
missingpriority	指定されたサイト優先グループは存在しません。
duplicatepriority	指定された優先順位レベルは、グループの他のサイト用にすでに存在しています。
duplicatevalue	指定されたサイトは、サイト優先グループにすでに存在しています。
notquiesced	マスター・グループは静止していません。

ADD\_conflicttype\_RESOLUTION プロシージャ

これらのプロシージャは、更新、削除または一意性競合を解消する方法を指定します。これらのプロシージャは、マスター定義サイトからコールする必要があります。コールするプロシージャは、ルーチンが解消する競合のタイプによって異なります。

表 20-12 ADD\_conflicttype\_RESOLUTION プロシージャ

競合のタイプ	プロシージャ名
update	ADD_UPDATE_RESOLUTION
uniqueness	ADD_UNIQUE_RESOLUTION
delete	ADD_DELETE_RESOLUTION

**関連項目：** 更新の競合解消メソッドの指定、一意性競合解消メソッドの選択、削除の競合解消メソッドの割当ての詳細は、[第 6 章「競合解消の構成」](#) および『Oracle9i アドバンスト・レプリケーション』を参照してください。

## 構文

```

DBMS_REPCAT.ADD_UPDATE_RESOLUTION (
    sname           IN   VARCHAR2,
    oname           IN   VARCHAR2,
    column_group    IN   VARCHAR2,
    sequence_no     IN   NUMBER,
    method          IN   VARCHAR2,
    parameter_column_name IN VARCHAR2
                                | DBMS_REPCAT.VARCHAR2s
                                | DBMS_UTILITY.LNAME_ARRAY,
    priority_group   IN   VARCHAR2      := NULL,
    function_name    IN   VARCHAR2      := NULL,
    comment          IN   VARCHAR2      := NULL);

DBMS_REPCAT.ADD_DELETE_RESOLUTION (
    sname           IN   VARCHAR2,
    oname           IN   VARCHAR2,
    sequence_no     IN   NUMBER,
    parameter_column_name IN VARCHAR2 | DBMS_REPCAT.VARCHAR2s,
    function_name    IN   VARCHAR2,
    comment          IN   VARCHAR2      := NULL
    method           IN   VARCHAR2      := 'USER FUNCTION');

DBMS_REPCAT.ADD_UNIQUE_RESOLUTION (
    sname           IN   VARCHAR2,
    oname           IN   VARCHAR2,
    constraint_name  IN   VARCHAR2,
    sequence_no     IN   NUMBER,
    method          IN   VARCHAR2,
    parameter_column_name IN VARCHAR2
                                | DBMS_REPCAT.VARCHAR2s
                                | DBMS_UTILITY.LNAME_ARRAY,
    function_name    IN   VARCHAR2      := NULL,
    comment          IN   VARCHAR2      := NULL);

```

パラメータ

表 20-13 ADD\_conflicttype\_RESOLUTION プロシージャのパラメータ

パラメータ	説明
sname	レプリケートされる表が含まれるスキーマの名前です。
oname	競合解消ルーチンを追加する表の名前です。表はネストした表の記憶表を利用できます。
column_group	競合解消ルーチンを追加する列グループの名前です。列グループは、更新の競合解消ルーチンのみに必要です。
constraint_name	競合解消ルーチンを追加する一意制約または一意索引の名前です。一意索引の名前が対応付けられた一意制約の名前と異なる場合は、一意索引の名前を使用します。制約名は、一意性競合解消ルーチンのみに必要です。
sequence_no	指定された競合解消メソッドを適用する順序です。
method	<p>作成する競合解消ルーチンのタイプです。アドバンスト・レプリケーションで提供される標準のルーチンの名前を指定できます。また、独自のルーチンを記述した場合は、ユーザー・ファンクションを選択して、独自の方法の名前を <code>function_name</code> パラメータとして指定します。</p> <p>このリリースでサポートされている更新の競合のための標準メソッドは次のとおりです。</p> <ul style="list-style-type: none"><li>■ <code>minimum</code></li><li>■ <code>maximum</code></li><li>■ <code>latest timestamp</code></li><li>■ <code>earliest timestamp</code></li><li>■ <code>additive、average</code></li><li>■ <code>priority group</code></li><li>■ <code>site priority</code></li><li>■ <code>overwrite</code></li><li>■ <code>discard</code></li></ul> <p>このリリースでサポートされている一意性競合のための標準メソッドは、<code>append site name</code>、<code>append sequence</code> および <code>discard</code> です。削除の競合については、組込み（Oracle 提供）方法はありません。</p>

表 20-13 ADD\_conflicttype\_RESOLUTION プロシージャのパラメータ (続き)

パラメータ	説明
parameter_column_name	<p>競合の解消に使用する列の名前です。標準メソッドは、1つの列で操作されます。たとえば、列グループに latest timestamp メソッドを使用する場合は、タイムスタンプ値が含まれる列の名前を、パラメータとして渡す必要があります。user function を使用する場合は、任意の数の列を使用して競合を解消できます。</p> <p>更新の競合または一意性競合については、このパラメータは列名のカンマ区切りのリスト、または DBMS_REPCAT.VARCHAR2S および DBMS_UTILITY.LNAME_ARRAY タイプの PL/SQL 索引付き表を受け付けます。30 バイト以上の列名がある場合は、DBMS_UTILITY.LNAME_ARRAY を使用します。これは列オブジェクトの属性を指定した際に起こる場合があります。</p> <p>削除の競合については、このパラメータは列名のカンマ区切りのリスト、または DBMS_REPCAT.VARCHAR2S タイプの PL/SQL 索引付き表を受け付けます。</p> <p>単一の値 '*' の場合は、競合の解消に表（更新の競合の場合は列グループ）内のすべての列が使用されます。'*' を指定した場合は、列はアルファベット順にファンクションへ渡されます。</p> <p>このパラメータには、LOB 列は指定できません。</p> <p><b>関連項目：</b>列オブジェクトを使用する場合は、20-23 ページの「<a href="#">使用上の注意</a>」を参照してください。</p>
priority_group	<p>更新の競合解消メソッドの priority_group または site priority を使用する場合は、作成した優先グループの名前を渡す必要があります。</p> <p>詳細は、<a href="#">第 6 章「競合解消の構成」</a> および『Oracle9i アドバンスド・レプリケーション』を参照してください。その他のメソッドを使用する場合は、このパラメータのデフォルト値 NULL を使用できます。このパラメータは、更新の競合のみに適用されます。</p>
function_name	<p>user function メソッドを選択した場合、または削除の競合解消ルーチンを追加する場合は、記述した競合解消ルーチンの名前を渡す必要があります。標準メソッドを使用する場合は、このパラメータのデフォルト値 NULL を使用できます。</p>
comment	<p>このユーザー・コメントは、DBA_REPRESOLUTION ビューに追加されます。</p>

例外

表 20-14 ADD\_conflicttype\_RESOLUTION プロシージャの例外

例外	説明
nonmasterdef	起動サイトはマスター定義サイトではありません。
missingobject	指定されたオブジェクトは、行レベル・レプリケーションを使用する指定されたスキーマに表として存在しません。
missingschema	指定されたスキーマは存在しません。
missingcolumn	parameter_column_name パラメータの一部として指定された列は存在しません。
missinggroup	指定された列グループは存在しません。
missingprioritygroup	指定された優先グループは、表に存在しません。
invalidmethod	指定された解決メソッドは認識されません。
invalidparameter	parameter_column_name パラメータに指定された列数は無効です。(標準のルーチンでは、1 つの列名のみが受け付けられます。)
missingfunction	指定されたユーザー・ファンクションは存在しません。
missingconstraint	一意性競合に指定された制約は存在しません。
notquiesced	指定された表が所属するレプリケーション・グループは静止していません。
duplicateresolution	指定された競合解消メソッドはすでに登録されています。
duplicatesequenece	指定された順序番号は指定されたオブジェクトにすでに割り当てられています。
invalidprioritygroup	指定された優先グループは存在しません。
paramtype	優先グループに割り当てられたタイプと異なります。

## 使用上の注意

列オブジェクトを使用する場合は、`parameter_column_name` パラメータに列オブジェクトの属性を指定できるかどうかは、競合解消メソッドが組込み（Oracle 提供）かユーザー作成かによって異なります。

- 組込みの競合解消メソッドを使用する場合は、このパラメータにオブジェクトの属性を指定できます。たとえば、`cust_address` という列オブジェクトに `street_address` という属性がある場合、このパラメータに `cust_address.street_address` を指定できます。
- 組込みの競合解消メソッドを使用する場合は、列オブジェクトの LOB 属性、列オブジェクトのコレクションまたはコレクション属性、REF、および列オブジェクト全体といったタイプの列は、このパラメータには指定できません。
- ユーザー作成の競合解消メソッドを使用する場合は、列オブジェクト全体を指定する必要があります。列オブジェクトの属性は指定できません。たとえば、`cust_address` という列オブジェクトに `street_address` という属性が（複数の属性の1つとして）ある場合、このパラメータには `cust_address` のみを指定できます。

## ALTER\_CATCHUP\_PARAMETERS プロシージャ

このプロシージャは DBA\_REPEXTENSIONS のデータ・ディクショナリ・ビューに保存されたパラメータの値を変更します。

- `percentage_for_catchup_mdef`
- `cycle_seconds_mdef`
- `percentage_for_catchup_new`
- `cycle_seconds_new`

これらのパラメータはもともと、`ADD_NEW_MASTERS` プロシージャで設定されたものです。これらのパラメータに指定する新しい値は、マスター・グループに新しいマスター・サイトを追加するプロセスの残りの手順において使用されます。この変更はそれが実行されるサイトのみで行われます。そのためすべてのサイトでパラメータを変更する場合は、マスター定義サイトを含む各マスター・サイトで変更を行う必要があります。

### 関連項目：

- 20-10 ページ [「ADD\\_NEW\\_MASTERS プロシージャ」](#)
- マスター・グループへのマスター・サイトの追加の詳細は、7-4 ページの [「新しいマスター・サイトの追加」](#) を参照してください。

構文

```
DBMS_REPCAT.ALTER_CATCHUP_PARAMETERS (
    extension_id          IN      RAW,
    percentage_for_catchup_mdef IN  BINARY_INTEGER := NULL,
    cycle_seconds_mdef    IN  BINARY_INTEGER := NULL,
    percentage_for_catchup_new IN  BINARY_INTEGER := NULL,
    cycle_seconds_new     IN  BINARY_INTEGER := NULL);
```

パラメータ

表 20-15 ALTER\_CATCHUP\_PARAMETERS プロシージャのパラメータ

パラメータ	説明
extension_id	静止せずにマスター・データベースを追加するための、現在保留になっている要求の識別子です。extension_id は、DBA_REPSITES_NEW データ・ディクショナリ・ビューおよび DBA_REPEXTENSIONS データ・ディクショナリ・ビューを問い合わせて検索できます。
percentage_for_catchup_mdef	マスター定義サイトへの伝播に追いつくために使用する必要のある、伝播リソースのパーセンテージです。10 の倍数で、かつ 0 以上 100 以下の数値にする必要があります。
cycle_seconds_mdef	このパラメータは percentage_for_catchup_mdef がともに有効で、10 以上 90 以下の値に設定されている場合に有効です。この場合、masterdef への伝播は、拡張されていないレプリケーション・グループと拡張されているレプリケーション・グループに対して、各サイクル中にプッシュが行われるたびに、交互に実行されます。このパラメータはサイクルの長さを秒単位で示します。
percentage_for_catchup_new	新しいマスター・サイトへの伝播に追いつくために使用する必要のある、伝播リソースのパーセンテージです。10 の倍数で、かつ 0 以上 100 以下の数値にする必要があります。
cycle_seconds_new	このパラメータは percentage_for_catchup_new がともに有効で、10 以上 90 以下の値に設定されている場合に有効です。この場合、masterdef への伝播は、拡張されていないレプリケーション・グループと拡張されているレプリケーション・グループに対して、各サイクル中にプッシュが行われるたびに、交互に実行されます。このパラメータはサイクルの長さを秒単位で示します。



例外

表 20-16 ALTER\_CATCHUP\_PARAMETERS プロシージャの例外

例外	説明
typfailure	パラメータの 1 つに指定されたパラメータ値が不適正です。
dbnotcompatible	機能がデータベースのバージョンと互換性がありません。データベースの互換性のレベルは、すべて 9.0.1 以上である必要があります。

ALTER\_MASTER\_PROPAGATION プロシージャ

このプロシージャは、指定されたマスター・サイトにある指定されたレプリケーション・グループの伝播方法を変更します。レプリケーション・グループは静止する必要があります。このプロシージャは、マスター定義サイトからコールする必要があります。マスターが dblink\_list または dblink\_table に出現した場合、ALTER\_MASTER\_PROPAGATION ではそのデータベース・リンクが無視されます。伝播モードをマスターからマスターに変更できません。

構文

```
DBMS_REPCAT.ALTER_MASTER_PROPAGATION (  
  gname          IN   VARCHAR2,  
  master         IN   VARCHAR2,  
  { dblink_list  IN   VARCHAR2,  
    | dblink_table IN   DBMS_UTILITY.DBLINK_ARRAY, }  
  propagation_mode IN  VARCHAR2 : = 'ASYNCHRONOUS',  
  comment        IN   VARCHAR2 := '' );
```

**注意：** このプロシージャはオーバーロードされています。dblink\_list パラメータと dblink\_table パラメータは、同時には指定できません。

パラメータ

表 20-17 ALTER\_MASTER\_PROPAGATION プロシージャのパラメータ

パラメータ	説明
gname	伝播モードの変更先のレプリケーション・グループの名前です。
master	伝播モードを変更するマスター・サイトの名前です。
dblink_list	伝播方法を変更するデータベース・リンクの、カンマ区切りのリストです。値が NULL に設定されている場合、変更されているマスター・サイト以外のすべてのマスター・サイトが使用されます。
dblink_table	伝播を変更するデータベース・リンクの、1 の位置から索引付けされている PL/SQL 索引付き表です。
propagation_mode	指定されたマスター・サイトで変更された内容を、データベース・リンクのリストで識別されるサイトへ伝播する方法を指定します。適切な値は synchronous および asynchronous です。
comment	このコメントは、DBA_REPPROP ビューに追加されます。

例外

表 20-18 ALTER\_MASTER\_PROPAGATION プロシージャの例外

例外	説明
nonmasterdef	起動サイトはマスター定義サイトではありません。
notquiesced	起動サイトは静止していません。
typefailure	指定された伝播モードは認識されませんでした。
nonmaster	データベース・リンクのリストに、マスター・サイト以外のサイトが含まれています。

## ALTER\_MASTER\_REPOBJECT プロシージャ

- このプロシージャは、レプリケーション環境のオブジェクトを変更します。このプロシージャは、マスター定義サイトからコールする必要があります。
- このプロシージャでは、次の条件のいずれかに該当する場合、オブジェクトのマスター・グループを静止させる必要があります。
- マルチマスター・レプリケーション環境の表を変更する場合。
  - 単一のマスター・レプリケーション環境で、safe\_table\_change パラメータが FALSE に設定された表を変更する場合。
- このプロシージャを使用すると、マスター・グループを静止させることなく、表以外のオブジェクトを変更できます。

### 構文

```
DBMS_REPCAT.ALTER_MASTER_REPOBJECT (  
    sname          IN    VARCHAR2,  
    oname          IN    VARCHAR2,  
    type           IN    VARCHAR2,  
    ddl_text       IN    VARCHAR2,  
    comment        IN    VARCHAR2    := '',  
    retry          IN    BOOLEAN      := false  
    safe_table_change IN    BOOLEAN      := false);
```

### パラメータ

表 20-19 ALTER\_MASTER\_REPOBJECT プロシージャのパラメータ

パラメータ	説明
sname	変更するオブジェクトが含まれるスキーマの名前です。
oname	変更するオブジェクトの名前です。オブジェクトには、ネストした表の記憶表は指定できません。

表 20-19 ALTER\_MASTER\_REPOBJECT プロシージャのパラメータ（続き）

パラメータ	説明
type	変更するオブジェクトのタイプです。サポートされているタイプは次のとおりです。 <div><div>FUNCTION</div><div>SYNONYM</div><div>INDEX</div><div>TABLE</div><div>INDEXTYPE</div><div>TRIGGER</div><div>OPERATOR</div><div>TYPE</div><div>PACKAGE</div><div>TYPE BODY</div><div>PACKAGE BODY</div><div>VIEW</div><div>PROCEDURE</div></div>
ddl_text	オブジェクトの変更に使用する DDL テキストです。DDL は、適用の前に解析されることはありません。このため、変更されるオブジェクトについての適切なスキーマおよびオブジェクト名が、DDL テキストで提供されるようにする必要があります。  スキーマの指定なしで DDL が渡された場合、レプリケーション管理者のスキーマがデフォルトのスキーマになります。レプリケーション管理者のスキーマ以外のスキーマを使用する場合は、必ずスキーマを指定してください。
comment	値が NULL に設定されていない場合、このコメントは DBA_REPOBJECT ビューの COMMENT フィールドに追加されます。
retry	値が TRUE に設定されている場合、ALTER_MASTER_REPOBJECT では、オブジェクト・ステータスが VALID でないマスターでのみ、オブジェクトが変更されます。

表 20-19 ALTER\_MASTER\_REPOBJECT プロシージャのパラメータ (続き)

パラメータ	説明
safe_table_change	<p>表の変更が安全である場合は、TRUE を指定します。表の変更が安全でない場合は、FALSE を指定します。</p> <p>安全な変更は、単一のマスター・レプリケーション環境のマスター表に、その表を含むマスター・グループを静止させずに行うことができます。安全でない変更を行うには、マスター・グループを静止させる必要があります。</p> <p>このパラメータは、単一のマスター・レプリケーション環境の表のみに指定します。このパラメータはマルチマスター・レプリケーション環境では、また指定したオブジェクトが表でない場合には無視されます。マルチマスター・レプリケーション環境では、マスター・グループを静止させて表の ALTER_MASTER_REPOBJECT プロシージャを実行する必要があります。</p> <p>安全な変更は次のとおりです。</p> <ul style="list-style-type: none"><li>■ 記憶域情報および拡張情報の変更</li><li>■ 既存の列の拡大。たとえば VARCHAR2 (20) 列を VARCHAR2 (50) 列にすること</li><li>■ 非主キー制約の追加</li><li>■ 非主キー制約の変更</li><li>■ 非主キー制約の有効化と無効化</li></ul> <p>安全でない変更は次のとおりです。</p> <ul style="list-style-type: none"><li>■ キーの列の追加または削除による主キーの変更</li><li>■ 列の追加または削除</li><li>■ 既存の列の縮小。たとえば VARCHAR2 (50) 列を VARCHAR2 (20) 列にすること</li><li>■ 主キー制約の無効化</li><li>■ 既存の列のデータ型の変更</li><li>■ 既存の列の削除</li></ul> <p>変更が安全かどうかが明確でない場合は、ALTER_MASTER_REPOBJECT プロシージャの実行前にマスター・グループを静止させます。</p>

例外

表 20-20 ALTER\_MASTER\_REPOBJECT プロシージャの例外

例外	説明
nonmasterdef	起動サイトはマスター定義サイトではありません。
notquiesced	対応付けられたレプリケーション・グループは中断されていません。
missingobject	sname および oname で識別されるオブジェクトは存在しません。
typefailure	指定されたタイプ・パラメータはサポートされていません。
ddlfailure	マスター定義サイトの DDL が失敗しました。
commfailure	少なくとも 1 つのマスター・サイトにアクセスできません。

ALTER\_MVIEW\_PROPAGATION プロシージャ

このプロシージャはカレント・マテリアライズド・ビュー・サイトの、指定されたレプリケーション・グループの伝播方法を変更します。このプロシージャは、マテリアライズド・ビュー・サイトの遅延トランザクション・キューをプッシュし、マテリアライズド・ビューの実表をロックし、トリガーおよびトリガーに対応付けられたパッケージを再生成します。このプロシージャは、マテリアライズド・ビュー・サイトからコールする必要があります。

構文

```
DBMS_REPCAT.ALTER_MVIEW_PROPAGATION (  
  gname                IN  VARCHAR2,  
  propagation_mode     IN  VARCHAR2,  
  comment              IN  VARCHAR2  := '',  
  gowner               IN  VARCHAR2  := 'PUBLIC');
```

パラメータ

表 20-21 ALTER\_MVIEW\_PROPAGATION プロシージャのパラメータ

パラメータ	説明
gname	伝播方法の変更先のレプリケーション・グループの名前です。
propagation_mode	カレント・マテリアライズド・ビュー・サイトからの変更が、対応付けられたマスター・サイトまたはマスター・マテリアライズド・ビュー・サイトに伝播される様式です。適切な値は synchronous および asynchronous です。
comment	このコメントは、DBA_REPPROP ビューに追加されます。
gowner	マテリアライズド・ビュー・グループの所有者です。

例外

表 20-22 ALTER\_MVIEW\_PROPAGATION プロシージャの例外

例外	説明
missingrepgroup	指定されたレプリケーション・グループは存在しません。
typefailure	伝播モードが正しく指定されませんでした。
nonmview	カレント・サイトは、指定されたレプリケーション・グループのマテリアライズド・ビュー・サイトではありません。
commfailure	マスター・サイトまたはマスター・マテリアライズド・ビュー・サイトにコンタクトできません。
notcompat	互換モードは 7.3.0.0 以上である必要があります。
failaltermviewrop	マテリアライズド・ビュー・グループの伝播は、マテリアライズド・ビュー・サイトを共有するマスター・サイトまたはマスター・マテリアライズド・ビュー・サイトが同一の、他のマテリアライズド・ビュー・グループが存在しない場合のみ変更できます。

ALTER\_PRIORITY プロシージャ

このプロシージャは、指定された優先グループ・メンバーに対応付けられた優先順位レベルを変更します。このプロシージャは、マスター定義サイトからコールする必要があります。

**関連項目：** 競合解消メソッドの詳細は、[第 6 章「競合解消の構成」](#)および『Oracle9i アドバンスド・レプリケーション』を参照してください。

構文

```
DBMS_REPCAT.ALTER_PRIORITY (  
  gname          IN   VARCHAR2,  
  pgroup         IN   VARCHAR2,  
  old_priority    IN   NUMBER,  
  new_priority    IN   NUMBER);
```

パラメータ

表 20-23 ALTER\_PRIORITY プロシージャのパラメータ

パラメータ	説明
gname	優先グループの対応付けられたマスター・グループです。
pgroup	変更する優先順位が含まれる優先グループの名前です。
old_priority	優先グループ・メンバーの現行の優先順位レベルです。
new_priority	優先グループ・メンバーに割り当てる新しい優先順位レベルです。

例外

表 20-24 ALTER\_PRIORITY プロシージャの例外

例外	説明
nonmasterdef	起動サイトはマスター定義サイトではありません。
duplicatepriority	新しい優先順位レベルは、優先グループにすでに存在しています。
missingrepgroup	指定されたマスター・グループは存在しません。
missingvalue	DBMS_REPCAT.ADD_PRIORITY_datatype へのコールで値が登録されませんでした。
missingprioritygroup	指定された優先グループは存在しません。
notquiesced	指定されたマスター・グループは静止していません。



# ALTER\_PRIORITY\_datatype プロシージャ

このプロシージャは、優先グループ・メンバーの値を変更します。このプロシージャは、マスター定義サイトからコールする必要があります。コールするプロシージャは、priority 列のデータ型によって異なります。

**関連項目：** 競合解消メソッドの詳細は、第 6 章「競合解消の構成」および『Oracle9i アドバンスド・レプリケーション』を参照してください。

## 構文

```
DBMS_REPCAT.ALTER_PRIORITY_datatype (
    gname          IN   VARCHAR2,
    pgroup         IN   VARCHAR2,
    old_value      IN   datatype,
    new_value      IN   datatype);
```

datatype は次のとおりです。

```
{ NUMBER
| VARCHAR2
| CHAR
| DATE
| RAW
| NCHAR
| NVARCHAR2 }
```

## パラメータ

表 20-25 ALTER\_PRIORITY\_datatype プロシージャのパラメータ

パラメータ	説明
gname	優先グループの対応付けられたマスター・グループです。
pgroup	変更する値が含まれる優先グループの名前です。
old_value	優先グループ・メンバーのカレント値です。
new_value	優先グループ・メンバーに割り当てる新しい値です。

例外

表 20-26 ALTER\_PRIORITY\_datatype プロシージャの例外

例外	説明
nonmasterdef	起動サイトはマスター定義サイトではありません。
duplicatevalue	新しい値は、優先グループにすでに存在しています。
missingrepgroup	指定されたマスター・グループは存在しません。
missingprioritygroup	指定された優先グループは存在しません。
missingvalue	古い値は存在しません。
paramtype	新しい値のデータ型は、この優先グループには不適切です。
typefailure	指定された値のデータ型は、この優先グループには不適切です。
notquiesced	指定されたマスター・グループは静止していません。

ALTER\_SITE\_PRIORITY プロシージャ

このプロシージャは、指定されたサイトに対応付けられた優先順位レベルを変更します。このプロシージャは、マスター定義サイトからコールする必要があります。

**関連項目：** 競合解消メソッドの詳細は、[第 6 章「競合解消の構成」](#)および『Oracle9i アドバンスド・レプリケーション』を参照してください。

構文

```
DBMS_REPCAT.ALTER_SITE_PRIORITY (  
  gname          IN   VARCHAR2,  
  name           IN   VARCHAR2,  
  old_priority   IN   NUMBER,  
  new_priority   IN   NUMBER);
```

## パラメータ

表 20-27 ALTER\_SITE\_PRIORITY プロシージャのパラメータ

パラメータ	説明
gname	サイト優先グループの対応付けられたマスター・グループです。
name	メンバーを変更するサイト優先グループの名前です。
old_priority	優先順位レベルを変更するサイトの、現行の優先順位レベルです。
new_priority	サイトの新しい優先順位レベルです。数値が高いほど、優先順位レベルが高くなります。

## 例外

表 20-28 ALTER\_SITE\_PRIORITY プロシージャの例外

例外	説明
nonmasterdef	起動サイトはマスター定義サイトではありません。
missingrepgroup	指定されたマスター・グループは存在しません。
missingpriority	古い優先順位レベルは、どのグループ・メンバーにも対応付けられていません。
duplicatepriority	新しい優先順位レベルは、グループの他のサイト用にすでに存在しています。
missingvalue	古い値は存在しません。
paramtype	新しい値のデータ型は、この優先グループには不適切です。
notquiesced	マスター・グループは静止していません。

## ALTER\_SITE\_PRIORITY\_SITE プロシージャ

このプロシージャは、指定された優先順位レベルに対応付けられたサイトを変更します。このプロシージャは、マスター定義サイトからコールする必要があります。

**関連項目：** 競合解消メソッドの詳細は、[第 6 章「競合解消の構成」](#)および『[Oracle9i アドバンスト・レプリケーション](#)』を参照してください。

### 構文

```
DBMS_REPCAT.ALTER_SITE_PRIORITY_SITE (  
    gname      IN   VARCHAR2,  
    name       IN   VARCHAR2,  
    old_site   IN   VARCHAR2,  
    new_site   IN   VARCHAR2);
```

### パラメータ

表 20-29 ALTER\_SITE\_PRIORITY\_SITE プロシージャのパラメータ

パラメータ	説明
gname	サイト優先グループの対応付けられたマスター・グループです。
name	メンバーを変更するサイト優先グループの名前です。
old_site	優先順位レベルとの対応付けを解除するサイトの、カレント・グローバル・データベース名です。
new_site	カレント優先順位レベルと対応付ける、新しいグローバル・データベース名です。

### 例外

表 20-30 ALTER\_SITE\_PRIORITY\_SITE プロシージャの例外

例外	説明
nonmasterdef	起動サイトはマスター定義サイトではありません。
missingrepgroup	指定されたマスター・グループは存在しません。
missingpriority	指定されたサイト優先グループは存在しません。
missingvalue	古いサイトはグループ・メンバーではありません。
notquiesced	マスター・グループは静止していません。

## CANCEL\_STATISTICS プロシージャ

このプロシージャは、正常に実行された表の更新、一意性および削除の競合解消の統計の収集を中止します。

### 構文

```
DBMS_REPCAT.CANCEL_STATISTICS (  
    sname      IN   VARCHAR2,  
    oname      IN   VARCHAR2);
```

### パラメータ

表 20-31 CANCEL\_STATISTICS プロシージャのパラメータ

パラメータ	説明
sname	表が置かれたスキーマの名前です。
oname	競合解消の統計を収集しない表の名前です。

### 例外

表 20-32 CANCEL\_STATISTICS プロシージャの例外

例外	説明
missingschema	指定されたスキーマは存在しません。
missingobject	指定された表は存在しません。
statnotreg	指定された表は現在、統計の収集用に登録されていません。

## COMMENT\_ON\_COLUMN\_GROUP プロシージャ

このプロシージャは、列グループの DBA\_REPCOLUMN\_GROUP ビュー内のコメント・フィールドを更新します。このコメントは、DBMS\_REPCAT.GENERATE\_REPLICATION\_SUPPORT への次のコールまで、どのマスター・サイトでも追加されません。

### 構文

```
DBMS_REPCAT.COMMENT_ON_COLUMN_GROUP (  
    sname          IN   VARCHAR2,  
    oname          IN   VARCHAR2,  
    column_group   IN   VARCHAR2,  
    comment        IN   VARCHAR2);
```

### パラメータ

表 20-33 COMMENT\_ON\_COLUMN\_GROUP プロシージャのパラメータ

パラメータ	説明
sname	オブジェクトが置かれたスキーマの名前です。
oname	列グループが対応付けられているレプリケート表の名前です。
column_group	列グループの名前です。
comment	DBA_REPCOLUMN_GROUP ビューの GROUP_COMMENT フィールドに含める、更新されたコメントのテキストです。

### 例外

表 20-34 COMMENT\_ON\_COLUMN\_GROUP プロシージャの例外

例外	説明
nonmasterdef	起動サイトはマスター定義サイトではありません。
missinggroup	指定された列グループは存在しません。
missingobj	オブジェクトがありません。

# COMMENT\_ON\_MVIEW\_REPSITES プロシージャ

このプロシージャは、指定されたマテリアライズド・ビュー・グループの DBA\_REPGROUP データ・ディクショナリ・ビュー内の SCHEMA\_COMMENT フィールドを更新します。グループ名は、ローカルでレプリケート・マテリアライズド・ビュー・グループとして登録する必要があります。このプロシージャは、マテリアライズド・ビュー・サイトで実行する必要があります。

## 構文

```
DBMS_REPCAT.COMMENT_ON_MVIEW_REPSITES (
    gowner      IN   VARCHAR2,
    gname       IN   VARCHAR2,
    comment     IN   VARCHAR2);
```

## パラメータ

表 20-35 COMMENT\_ON\_MVIEW\_REPSITES プロシージャのパラメータ

パラメータ	説明
gowner	マテリアライズド・ビュー・グループの所有者です。
gname	マテリアライズド・ビュー・グループの名前です。
comment	DBA_REPGROUP ビューの SCHEMA_COMMENT フィールドに含める、更新されたコメントです。

## 例外

表 20-36 COMMENT\_ON\_MVIEW\_REPSITES プロシージャの例外

パラメータ	説明
missingrepgroup	マテリアライズド・ビュー・グループは存在しません。
nonmview	接続サイトはマテリアライズド・ビュー・サイトではありません。

## COMMENT\_ON\_PRIORITY\_GROUP/COMMENT\_ON\_SITE\_PRIORITY プロシージャ

COMMENT\_ON\_PRIORITY\_GROUP では、優先グループの DBA\_REPPRIORITY\_GROUP ビューのコメント・フィールドが更新されます。このコメントは、GENERATE\_REPLICATION\_SUPPORT への次のコールまで、どのマスター・サイトでも追加されません。

COMMENT\_ON\_SITE\_PRIORITY では、サイト優先グループの DBA\_REPPRIORITY\_GROUP ビューのコメント・フィールドが更新されます。このプロシージャは COMMENT\_ON\_COLUMN\_GROUP プロシージャのラッパーで、便宜的に提供されるものです。このプロシージャは、マスター定義サイトからコールする必要があります。

### 構文

```
DBMS_REPCAT.COMMENT_ON_PRIORITY_GROUP (  
  gname      IN   VARCHAR2,  
  pgroup     IN   VARCHAR2,  
  comment    IN   VARCHAR2);  
  
DBMS_REPCAT.COMMENT_ON_SITE_PRIORITY (  
  gname      IN   VARCHAR2,  
  name       IN   VARCHAR2,  
  comment    IN   VARCHAR2);
```

### パラメータ

表 20-37 COMMENT\_ON\_PRIORITY\_GROUP/COMMENT\_ON\_SITE\_PRIORITY のパラメータ

パラメータ	説明
gname	マスター・グループの名前です。
pgroup/name	優先グループまたはサイト優先グループの名前です。
comment	DBA_REPPRIORITY_GROUP ビューの PRIORITY_COMMENT フィールドに含める、更新されたコメントのテキストです。



# 例外

**表 20-38 COMMENT\_ON\_PRIORITY\_GROUP/COMMENT\_ON\_SITE\_PRIORITY の例外**

例外	説明
nonmasterdef	起動サイトはマスター定義サイトではありません。
missingrepgroup	指定されたマスター・グループは存在しません。
missingprioritygroup	指定された優先グループは存在しません。

# COMMENT\_ON\_REPGROUP プロシージャ

このプロシージャは、マスター・グループの DBA\_REPGROUP ビュー内のコメント・フィールドを更新します。このプロシージャは、マスター定義サイトからコールする必要があります。

# 構文

```
DBMS_REPCAT.COMMENT_ON_REPGROUP (
    gname      IN   VARCHAR2,
    comment    IN   VARCHAR2);
```

# パラメータ

**表 20-39 COMMENT\_ON\_REPGROUP プロシージャのパラメータ**

パラメータ	説明
gname	コメントを付けるレプリケーション・グループの名前です。
comment	DBA_REPGROUP ビューの SCHEMA_COMMENT フィールドに含める、更新されたコメントです。

# 例外

**表 20-40 COMMENT\_ON\_REPGROUP プロシージャの例外**

例外	説明
nonmasterdef	起動サイトはマスター定義サイトではありません。
commfailure	少なくとも 1 つのマスター・サイトにアクセスできません。

## COMMENT\_ON\_REPOBJECT プロシージャ

このプロシージャは、マスター・グループ内のレプリケート・オブジェクトの DBA\_REPOBJECT ビュー内のコメント・フィールドを更新します。このプロシージャは、マスター定義サイトからコールする必要があります。

### 構文

```
DBMS_REPCAT.COMMENT_ON_REPOBJECT (  
    sname      IN   VARCHAR2,  
    oname      IN   VARCHAR2,  
    type       IN   VARCHAR2,  
    comment    IN   VARCHAR2);
```

### パラメータ

表 20-41 COMMENT\_ON\_REPOBJECT プロシージャのパラメータ

パラメータ	説明
sname	オブジェクトが置かれたスキーマの名前です。
oname	コメントを付けるオブジェクトの名前です。オブジェクトには、ネストした表の記憶表は指定できません。
type	オブジェクトのタイプです。サポートされているタイプは次のとおりです。 <div><div>FUNCTION</div><div>INDEX</div><div>INDEXTYPE</div><div>OPERATOR</div><div>PACKAGE</div><div>PACKAGE BODY</div><div>PROCEDURE</div><div>SYNONYM</div><div>TABLE</div><div>TRIGGER</div><div>TYPE</div><div>TYPE BODY</div><div>VIEW</div></div>
comment	DBA_REPOBJECT ビューの OBJECT_COMMENT フィールドに含める、更新されたコメントのテキストです。

例外

表 20-42 COMMENT\_ON\_REPOBJECT プロシージャの例外

例外	説明
nonmasterdef	起動サイトはマスター定義サイトではありません。
missingobject	指定されたオブジェクトは存在しません。
typefailure	指定されたタイプ・パラメータはサポートされていません。
commfailure	少なくとも 1 つのマスター・サイトにアクセスできません。

COMMENT\_ON\_REPSITES プロシージャ

レプリケーション・グループがマスター・グループの場合、このプロシージャは、マスター・サイトの DBA\_REPSITES ビュー内の MASTER\_COMMENT フィールドを更新します。レプリケーション・グループがマテリアライズド・ビュー・グループの場合、このプロシージャは、マテリアライズド・ビュー・サイトの DBA\_REPGROUP ビュー内の SCHEMA\_COMMENT フィールドを更新します。

このプロシージャは、マスター・サイトでもマテリアライズド・ビュー・サイトでも実行できます。マテリアライズド・ビュー・サイトでこのプロシージャを実行する場合は、マテリアライズド・ビュー・グループの所有者が PUBLIC である必要があります。

**関連項目：** マテリアライズド・ビュー・グループの所有者が PUBLIC でない場合にマテリアライズド・ビュー・サイトの DBA\_REPGROUP ビュー内の SCHEMA\_COMMENT フィールドにコメントを配置する方法の詳細は、20-45 ページの「[COMMENT\\_ON\\_conflicttype\\_RESOLUTION プロシージャ](#)」を参照してください。

構文

```
DBMS_REPCAT.COMMENT_ON_REPSITES (  
  gname      IN   VARCHAR2,  
  [ master   IN   VARCHAR,]  
  comment    IN   VARCHAR2);
```

パラメータ

表 20-43 COMMENT\_ON\_REPSITES プロシージャのパラメータ

パラメータ	説明
gname	レプリケーション・グループの名前です。1 つのデータベースが複数のレプリケーション環境でマスター・サイトになっている場合、このパラメータによって混乱が回避されます。
master	コメントを付けるマスター・サイトの、完全修飾されたデータベース名です。マスター・サイトでこのプロシージャを実行する場合は、このパラメータが必要です。マテリアライズド・ビュー・サイトでコメントを更新するには、このパラメータを省略します。このパラメータはオプションです。
comment	適切なディクショナリ・ビューのコメント・フィールドに含める、更新されたコメントのテキストです。サイトがマスター・サイトの場合、DBA_REPSITES ビューの MASTER_COMMENT フィールドが更新されます。サイトがマテリアライズド・ビュー・サイトの場合、DBA_REPGROUP ビューの SCHEMA_COMMENT フィールドが更新されます。

例外

表 20-44 COMMENT\_ON\_REPSITES プロシージャの例外

例外	説明
nonmasterdef	起動サイトはマスター定義サイトではありません。
nonmaster	起動サイトはマスター・サイトではありません。
commfailure	少なくとも 1 つのマスター・サイトにアクセスできません。
missingrepgroup	レプリケーション・グループは存在しません。
commfailure	1 つ以上のマスター・サイトにアクセスできません。
corrupt	レプリケーション・カタログ・ビューに不整合があります。

# COMMENT\_ON\_conflicttype\_RESOLUTION プロシージャ

このプロシージャは、競合解消ルーチンの DBA\_REPRESOLUTION ビュー内の RESOLUTION\_COMMENT フィールドを更新します。コールするプロシージャは、ルーチンが解消する競合のタイプによって異なります。これらのプロシージャは、マスター定義サイトからコールする必要があります。

表 20-45 COMMENT\_ON\_conflicttype\_RESOLUTION プロシージャ

競合のタイプ	プロシージャ名
update	COMMENT_ON_UPDATE_RESOLUTION
uniqueness	COMMENT_ON_UNIQUE_RESOLUTION
delete	COMMENT_ON_DELETE_RESOLUTION

コメントは、GENERATE\_REPLICATION\_SUPPORT への次のコールまで、どのマスター・サイトでも追加されません。

## 構文

```

DBMS_REPCAT.COMMENT_ON_UPDATE_RESOLUTION (
    sname          IN   VARCHAR2,
    oname          IN   VARCHAR2,
    column_group   IN   VARCHAR2,
    sequence_no    IN   NUMBER,
    comment        IN   VARCHAR2);

DBMS_REPCAT.COMMENT_ON_UNIQUE_RESOLUTION (
    sname          IN   VARCHAR2,
    oname          IN   VARCHAR2,
    constraint_name IN   VARCHAR2,
    sequence_no    IN   NUMBER,
    comment        IN   VARCHAR2);

DBMS_REPCAT.COMMENT_ON_DELETE_RESOLUTION (
    sname          IN   VARCHAR2,
    oname          IN   VARCHAR2,
    sequence_no    IN   NUMBER,
    comment        IN   VARCHAR2);

```

パラメータ

表 20-46 COMMENT\_ON\_conflicttype\_RESOLUTION プロシージャのパラメータ

パラメータ	説明
sname	スキーマの名前です。
oname	競合解消ルーチンが対応付けられているレプリケート表の名前です。
column_group	更新の競合解消ルーチンが対応付けられている列グループの名前です。
constraint_name	一意性競合解消ルーチンが対応付けられている一意制約の名前です。
sequence_no	競合解消プロシージャの順序番号です。
comment	DBA_REPRESOLUTION ビューの RESOLUTION_COMMENT フィールドに含める、更新されたコメントのテキストです。

例外

表 20-47 COMMENT\_ON\_conflicttype\_RESOLUTION プロシージャの例外

例外	説明
nonmasterdef	起動サイトはマスター定義サイトではありません。
missingobject	指定されたオブジェクトは存在しません。
missingresolution	指定された競合解消ルーチンは登録されていません。

## COMPARE\_OLD\_VALUES プロシージャ

このプロシージャは、遅延トランザクションの伝播中に、更新および削除のために各マスター・サイトの古い列値をレプリケート表の非キー列に比較するかどうかを指定します。デフォルトでは、すべての列の古い値が比較されます。マスター定義サイトで DBMS\_REPCAT.COMPARE\_OLD\_VALUES を呼び出すことにより、すべてのマスター・サイトおよびマテリアライズド・ビュー・サイトで、この動作を変更できます。

ユーザー定義型を使用する場合は、列オブジェクトのリーフ属性や、または列オブジェクト全体を指定できます。たとえば、`cust_address` という列オブジェクトに `street_address` という属性がある場合、`column_list` パラメータに、または `column_table` パラメータの一部として、`cust_address.street_address` を指定できます。または `cust_address` のみを指定することもできます。

競合検出のための等価比較を実行する場合、オブジェクトは次のいずれかの条件に該当する場合のみ等価として扱われます。

- 両方のオブジェクトがアトミック NULL の場合（オブジェクト全体は NULL）
- オブジェクト内で、対応する属性すべてが等価な場合

これらの条件をふまえて、一方のオブジェクトがアトミック NULL で他方がそうでない場合、それらのオブジェクトは等価とは扱われません。等価比較を実行する際には、MAP メソッドおよび ORDER メソッドは考慮されません。

## 構文

```
DBMS_REPCAT.COMPARE_OLD_VALUES (
    sname          IN  VARCHAR2,
    oname          IN  VARCHAR2,
    { column_list  IN  VARCHAR2,
    | column_table IN  DBMS_UTILITY.VARCHAR2s | DBMS_UTILITY.LNAME_ARRAY, }
    operation      IN  VARCHAR2 := 'UPDATE',
    compare        IN  BOOLEAN := true );
```

---

**注意：** このプロシージャはオーバーロードされています。 `column_list` パラメータと `column_table` パラメータは、同時には指定できません。

---

パラメータ

表 20-48 COMPARE\_OLD\_VALUES プロシージャのパラメータ

パラメータ	説明
sname	表が置かれたスキーマの名前です。
oname	レプリケート表の名前です。表はネストした表の記憶表を利用できません。
column_list	表の列のカンマ区切りのリストです。各エントリの間にはスペースを挿入しません。
column_table	リストのかわりに、DBMS_REPCAT.VARCHAR2S または DBMS_UTILITY.LNAME_ARRAY の PL/SQL 索引付き表を使用して列名を指定できます。最初の列名は 1 の位置、2 番目の列名は 2 の位置と、順に配置します。  30 バイト以上の列名がある場合は、DBMS_UTILITY.LNAME_ARRAY を使用します。これは列オブジェクトの属性を指定した際に起こる場合があります。
operation	可能な値は、update、delete、または更新と削除を意味するアスタリスク・ワイルドカード '*' です。
compare	値が TRUE に設定されている場合、指定された列の古い値は送信時に比較されます。値が FALSE に設定されている場合、指定された列の古い値は送信時に比較されません。指定されていない列や操作には影響がありません。表の min_communication が TRUE になると、マスター定義サイトで、指定された変更が有効になります。 min_communication TRUE で、マスター・サイトやマテリアライズド・ビュー・サイトで次にレプリケーション・サポートが生成されると、変更が有効になります。

**注意：** operation パラメータにより、行が削除または更新された際に非キー列の古い値を比較するかどうかを決定できます。古い値を比較しない場合、古い値は更新または削除を行った時点でのターゲット・サイドの列のカレント値と見なされます。

Oracle のデフォルトの動作を変更する前に、削減されたデータ伝播の詳細を、COMPARE\_OLD\_VALUES プロシージャを使用して『Oracle9i アドバンスド・レプリケーション』で参照してください。



例外

表 20-49 COMPARE\_OLD\_VALUES プロシージャの例外

例外	説明
nonmasterdef	起動サイトはマスター定義サイトではありません。
missingobject	指定されたオブジェクトは、行レベル・レプリケーションの情報が必要な指定されたスキーマ内の表として存在しません。
missingcolumn	少なくとも 1 つの列が、表に存在しません。
notquiesced	マスター・グループは静止していません。
typefailure	無効な操作が指定されました。
keysendcomp	指定された列は、表中のキー列です。
dbnotcompatible	機能がデータベースのバージョンと互換性がありません。この例外は一般的に、列オブジェクトの属性を比較しようとした際に発生します。この場合、データベースの互換性のレベルは、すべて 9.0.1 以上である必要があります。

CREATE\_MASTER\_REPGROUP プロシージャ

このプロシージャは、新しい空の静止したマスター・グループを作成します。

構文

```
DBMS_REPCAT.CREATE_MASTER_REPGROUP (  
  gname          IN   VARCHAR2,  
  group_comment  IN   VARCHAR2      := '',  
  master_comment IN   VARCHAR2      := ''),  
  qualifier      IN   VARCHAR2      := '');
```

パラメータ

表 20-50 CREATE\_MASTER\_REPGROUP プロシージャのパラメータ

パラメータ	説明
gname	作成するマスター・グループの名前です。
group_comment	このコメントは、DBA_REPGROUP ビューに追加されます。
master_comment	このコメントは、DBA_REPSITES ビューに追加されます。

表 20-50 CREATE\_MASTER\_REPGROUP プロシージャのパラメータ（続き）

パラメータ	説明
qualifier	マスター・グループの接続修飾子です。@ 符号を必ず使用してください。接続修飾子の詳細は、『Oracle9i アドバンスト・レプリケーション』および『Oracle9i データベース管理者ガイド』を参照してください。

例外

表 20-51 CREATE\_MASTER\_REPGROUP プロシージャの例外

例外	説明
duplicaterepgroup	マスター・グループはすでに存在しています。
norepopt	アドバンスト・レプリケーション・オプションがインストールされていません。
missingrepgroup	マスター・グループの名前が指定されませんでした。
qualifiertoolong	接続修飾子が長すぎます。

CREATE\_MASTER\_REPOBJECT プロシージャ

このプロシージャは、オブジェクトをマスター・グループに追加してレプリケート・オブジェクトにします。このプロシージャはユーザー定義型のオブジェクト識別子および全レプリケーション・サイトのオブジェクト表を保存します。

クラスタ化表のレプリケーションがサポートされていますが、クラスタ化表の場合は、use\_existing\_object パラメータを FALSE に設定できません。すなわち、CREATE\_MASTER\_REPOBJECT プロシージャを実行する前に、マスター・グループに関係する全マスター・サイトでクラスタ化表を作成する必要があります。ただし、これらの表に表データを入れる必要はありません。このため、クラスタ化表の copy\_rows パラメータを TRUE に設定できます。

構文

```
DBMS_REPCAT.CREATE_MASTER_REPOBJECT (  
  sname          IN   VARCHAR2,  
  oname          IN   VARCHAR2,  
  type           IN   VARCHAR2,  
  use_existing_object IN  BOOLEAN      := true,  
  ddl_text       IN   VARCHAR2      := NULL,  
  comment        IN   VARCHAR2      := '',  
  retry          IN   BOOLEAN      := false,  
  copy_rows      IN   BOOLEAN      := true,  
  gname          IN   VARCHAR2      := '');
```

## パラメータ

このプロシージャのパラメータを次の表に示します。

表 20-52 CREATE\_MASTER\_REPOBJECT プロシージャのパラメータ

パラメータ	説明														
sname	レプリケートするオブジェクトが置かれたスキーマの名前です。														
oname	レプリケートするオブジェクトの名前です。ddl_text が NULL の場合、このオブジェクトは指定されたスキーマにすでに存在しています。一意性を確実にするため、表の名前は最長で 27 バイト、パッケージの名前は 24 バイト以内にします。オブジェクトには、ネストした表の記憶表は指定できません。														
type	レプリケートするオブジェクトのタイプです。サポートされているタイプは次のとおりです。 <table><tr><td>FUNCTION</td><td>SYNONYM</td></tr><tr><td>INDEX</td><td>TABLE</td></tr><tr><td>INDEXTYPE</td><td>TRIGGER</td></tr><tr><td>OPERATOR</td><td>TYPE</td></tr><tr><td>PACKAGE</td><td>TYPE BODY</td></tr><tr><td>PACKAGE BODY</td><td>VIEW</td></tr><tr><td>PROCEDURE</td><td></td></tr></table>	FUNCTION	SYNONYM	INDEX	TABLE	INDEXTYPE	TRIGGER	OPERATOR	TYPE	PACKAGE	TYPE BODY	PACKAGE BODY	VIEW	PROCEDURE	
FUNCTION	SYNONYM														
INDEX	TABLE														
INDEXTYPE	TRIGGER														
OPERATOR	TYPE														
PACKAGE	TYPE BODY														
PACKAGE BODY	VIEW														
PROCEDURE															
use_existing_object	カレント・マスター・サイトで、同じタイプで同じ shape のオブジェクトを再利用する場合は、TRUE を指定します。詳細は、 <a href="#">表 20-54</a> を参照してください。  <b>注意：</b> クラスタ化表の場合、このパラメータは TRUE に設定する必要があります。														
ddl_text	オブジェクトがマスター定義サイトに存在していない場合は、このオブジェクトの作成に必要な DDL テキストを渡す必要があります。PL/SQL パッケージ、パッケージ本体、プロシージャおよびファンクションには、セミコロン ( ; ) が必要です。SQL 文の最後はセミコロン ( ; ) ではありません。DDL は、適用の前に解析されることはありません。このため、DDL テキストには、作成されるオブジェクトについての適切なスキーマおよびオブジェクト名が含まれるようにする必要があります。  スキーマ (sname パラメータ) を指定せずに DDL が渡された場合、レプリケーション管理者のスキーマがデフォルトのスキーマになります。レプリケーション管理者のスキーマ以外のスキーマを使用する場合は、必ずスキーマを指定してください。  <b>注意：</b> ユーザー定義型やオブジェクト表の追加には、ddl_text パラメータは使用しないでください。かわりにまずオブジェクトを作成して、そのオブジェクトを追加します。														

表 20-52 CREATE\_MASTER\_REPOBJECT プロシージャのパラメータ (続き)

パラメータ	説明
comment	このコメントは、DBA_REPOBJECT ビューの OBJECT_COMMENT フィールドに追加されます。
retry	前に作成できなかったオブジェクトの作成を再実行する場合は、TRUE を指定します。エラーが一時的あるいは調整済の場合、または以前にリソースが不十分であった場合にはこれを使用します。TRUE を指定すると、オブジェクト・ステータスが VALID でないマスター・サイトでのみ、オブジェクトが作成されます。
copy_rows	新しいレプリケート・オブジェクトの初期の内容を、マスター定義サイトのオブジェクトの内容と一致させる場合は、TRUE を指定します。詳細は、表 20-54 を参照してください。
gname	レプリケート・オブジェクトを作成するレプリケーション・グループの名前です。デフォルトのレプリケーション・グループ名の指定がない場合、スキーマ名が使用されます。また、プロシージャを正常に終了するために、スキーマと同じ名前のレプリケーション・グループが存在している必要があります。

例外

表 20-53 CREATE\_MASTER\_REPOBJECT プロシージャの例外

例外	説明
nonmasterdef	起動サイトはマスター定義サイトではありません。
notquiesced	マスター・グループは静止していません。
duplicateobject	指定されたオブジェクトは、マスター・グループにすでに存在しています。また、retry は FALSE です。名前が重複している場合も、この例外が発生します。
missingobject	sname および oname で識別されるオブジェクトは存在しません。また、適切な DDL が渡されていません。
typefailure	指定されたタイプのオブジェクトはレプリケートできません。
ddlfailure	マスター定義サイトの DDL が失敗しました。
commfailure	少なくとも 1 つのマスター・サイトにアクセスできません。
notcompat	7.3 以降のリリースとの互換性モードでないリモート・マスターがあります。

オブジェクト作成

表 20-54 マスター・サイトでのオブジェクト作成

既存の オブジェクトの 有無	COPY_ROWS	USE_EXISTING_ OBJECTS	結果
有	true	true	オブジェクトが一致しない場合、 duplicatedobject メッセージが生成 されます。表の場合は、マスター定義サ イトのデータを使用します。
有	false	true	オブジェクトが一致しない場合、 duplicatedobject メッセージが生成 されます。表の場合、データベース管理 者は、内容を同一にする必要がありま す。
有	true/false	false	duplicatedobject メッセージが生成 されます。
無	true	true/false	オブジェクトが作成されます。マスター 定義サイトのデータを使用して、表が設 定されます。
無	false	true/false	オブジェクトが作成されます。データ ベース管理者は、表を設定し、すべての サイトで表の整合性を保つようにする必 要があります。

## CREATE\_MVIEW\_REPGROUP プロシージャ

このプロシージャは、ローカル・データベースに、新しい空のマテリアライズド・ビュー・グループを作成します。CREATE\_MVIEW\_REPGROUP では REGISTER\_MIEW\_REPGROUP が自動的にコールされますが、登録中にエラーが発生しても無視されます。

### 構文

```
DBMS_REPCAT.CREATE_MVIEW_REPGROUP (
  gname          IN   VARCHAR2,
  master         IN   VARCHAR2,
  comment        IN   VARCHAR2      := '',
  propagation_mode IN VARCHAR2      := 'ASYNCHRONOUS',
  fname         IN   VARCHAR2      := NULL
  gowner        IN   VARCHAR2      := 'PUBLIC');
```

### パラメータ

表 20-55 CREATE\_MVIEW\_REPGROUP プロシージャのパラメータ

パラメータ	説明
gname	レプリケーション・グループの名前です。このグループは、指定されたマスター・サイトまたはマスター・マテリアライズド・ビュー・サイトに存在する必要があります。
master	マスター・サイトまたはマスター・マテリアライズド・ビュー・サイトとして使用する、レプリケーション環境のデータベースの完全修飾されたデータベース名です。必要な場合は接続修飾子も指定できます。接続修飾子の使用方法の詳細は、『Oracle9i アドバンスド・レプリケーション』および『Oracle9i データベース管理者ガイド』を参照してください。
comment	このコメントは、DBA_REPGROUP ビューに追加されます。
propagation_mode	レプリケーション・グループ内の、すべての更新可能なマテリアライズド・ビューのための伝播方法です。使用できる値は synchronous および asynchronous です。
fname	このパラメータは内部で使います。 <b>注意：</b> オラクル社カスタマ・サポート・センターから指示された場合を除き、このパラメータを設定しないでください。
gowner	マテリアライズド・ビュー・グループの所有者です。

例外

表 20-56 CREATE\_MVIEW\_REPGROUP プロシージャの例外

例外	説明
duplicaterepgroup	レプリケーション・グループは、起動サイトにすでに存在しています。
nonmaster	データベースはマスター・サイトまたはマスター・マテリアライズド・ビュー・サイトではありません。
commfailure	指定されたデータベースにはアクセスできません。
norepopt	アドバンスド・レプリケーション・オプションがインストールされていません。
typfailure	伝播モードが正しく指定されませんでした。
missingrepgroup	レプリケーション・グループはマスター・サイトに存在しません。
invalidqualifier	マスター・サイトまたはマスター・マテリアライズド・ビュー・サイトに指定された接続修飾子は、レプリケーション・グループには使用できません。
alreadymastered	ローカル・サイトに、同じグループ名で、マスター・サイトまたはマスター・マテリアライズド・ビュー・サイトが異なる別のマテリアライズド・ビュー・グループがあります。

CREATE\_MVIEW\_REPOBJECT プロシージャ

このプロシージャはマテリアライズド・ビュー・グループにレプリケート・オブジェクトを追加します。

構文

```
DBMS_REPCAT.CREATE_MVIEW_REPOBJECT (  
    sname          IN   VARCHAR2,  
    oname          IN   VARCHAR2,  
    type           IN   VARCHAR2,  
    ddl_text       IN   VARCHAR2  := '',  
    comment        IN   VARCHAR2  := '',  
    gname          IN   VARCHAR2  := '',  
    gen_objs_owner IN   VARCHAR2  := '',  
    min_communication IN   BOOLEAN  := true,  
    generate_80_compatible IN   BOOLEAN  := true,  
    gowner         IN   VARCHAR2  := 'PUBLIC');
```

パラメータ

表 20-57 CREATE\_MVIEW\_REPOBJECT プロシージャのパラメータ

パラメータ	説明
sname	オブジェクトが置かれたスキーマの名前です。スキーマは、このマテリアライズド・ビューのベースとなるマスター表またはマスター・マテリアライズド・ビューを所有するスキーマと同じものである必要があります。
oname	レプリケート・マテリアライズド・ビュー・グループに追加するオブジェクトの名前です。
type	レプリケートするオブジェクトのタイプです。サポートされているタイプは次のとおりです。 <div><div>FUNCTION</div><div>INDEX</div><div>INDEXTYPE</div><div>OPERATOR</div><div>PACKAGE</div><div>PACKAGE BODY</div><div>PROCEDURE</div><div>SNAPSHOT</div><div>SYNONYM</div><div>TRIGGER</div><div>TYPE</div><div>TYPE BODY</div><div>VIEW</div></div>
ddl_text	<p>タイプ SNAPSHOT のオブジェクトの場合、オブジェクトを作成するために必要な DDL です。他のタイプの場合は、次のデフォルトを使用します。</p> <p>'' (空の文字列)</p> <p>同じ名前のマテリアライズド・ビューがすでに存在している場合、DDL は無視され、既存のマテリアライズド・ビューがレプリケート・オブジェクトとして登録されます。マテリアライズド・ビュー用のマスター表またはマスター・マテリアライズド・ビューが、このスキーマに指定されているマスターのレプリケーション・グループに存在しない場合は、missingobject エラーが発生します。</p> <p>スキーマを指定せずに DDL が渡された場合、レプリケーション管理者のスキーマがデフォルトのスキーマになります。レプリケーション管理者のスキーマ以外のスキーマを使用する場合は、必ずスキーマを指定してください。</p> <p>オブジェクトが SNAPSHOT タイプでない場合は、マテリアライズド・ビュー・サイトはマスター・サイトまたはマスター・マテリアライズド・ビュー・サイトに接続して DDL テキストをプルダウンし、オブジェクトを作成します。オブジェクト型が TYPE または TYPE BODY の場合は、マテリアライズド・ビュー・サイトにおけるオブジェクトのオブジェクト識別子 (OID) はマスター・サイトまたはマスター・マテリアライズド・ビュー・サイトにおける OID と同じになります。</p>



表 20-57 CREATE\_MVIEW\_REPOBJECT プロシージャのパラメータ (続き)

パラメータ	説明
comment	このコメントは、DBA_REPOBJECT ビューの OBJECT_COMMENT フィールドに追加されます。
gname	オブジェクトを追加するレプリケート・マテリアライズド・ビュー・グループの名前です。デフォルトのグループ名の指定がない場合、スキーマ名が使用されます。また、プロシージャを正常に終了するために、スキーマと同じ名前のマテリアライズド・ビュー・グループが存在している必要があります。
gen_objs_owner	トランザクションの所有者として割り当てるユーザーの名前です。
min_communication	マテリアライズド・ビューのマスター・サイトで Oracle7 リリース 7.3 が実行されている場合は、FALSE に設定します。新旧の値の伝播を最小にするには、TRUE に設定します。デフォルトは TRUE です。競合解消メソッドの詳細は、 <a href="#">第 6 章「競合解消の構成」</a> および『Oracle9i アドバンスド・レプリケーション』を参照してください。
generate_80_compatible	マテリアライズド・ビューのマスター・サイトで Oracle8i リリース 8.1.5 より前の Oracle サーバーが実行されている場合は、TRUE に設定します。マテリアライズド・ビューのマスター・サイトまたはマスター・マテリアライズド・ビュー・サイトで Oracle8i リリース 8.1.5 以上が実行されている場合は、FALSE に設定します。
gowner	マテリアライズド・ビュー・グループの所有者です。

例外

表 20-58 CREATE\_MVIEW\_REPOBJECT プロシージャの例外

例外	説明
nonmview	起動サイトはマテリアライズド・ビュー・サイトではありません。
nonmaster	マスターがマスター・サイトまたはマスター・マテリアライズド・ビュー・サイトではなくなっています。
missingobject	指定されたオブジェクトは、マスターのレプリケーション・グループに存在しません。
duplicateobject	指定されたオブジェクトは、異なる shape ですすでに存在しています。
typefailure	許可されないタイプです。
ddlfailure	DDL が失敗しました。
commfailure	マスター・サイトまたはマスター・マテリアライズド・ビュー・サイトにアクセスできません。
missingschema	スキーマは、データベース・スキーマとして存在していません。
badmviewddl	DDL は実行されましたが、マテリアライズド・ビューが存在しません。
onlyonemview	マスター表のマテリアライズド・ビュー、またはマスター・マテリアライズド・ビューを、1 つのみ作成できます。
badmviewname	マテリアライズド・ビューの実表がマスター表またはマスター・マテリアライズド・ビューと異なります。
missingrepgroup	マスターのレプリケーション・グループは存在しません。

## DEFINE\_COLUMN\_GROUP プロシージャ

このプロシージャは、空の列グループを作成します。このプロシージャは、マスター定義サイトからコールする必要があります。

**関連項目：** 競合解消メソッドの詳細は、[第 6 章「競合解消の構成」](#)および『[Oracle9i アドバンスド・レプリケーション](#)』を参照してください。

### 構文

```
DBMS_REPCAT.DEFINE_COLUMN_GROUP (  
    sname          IN   VARCHAR2,  
    oname          IN   VARCHAR2,  
    column_group   IN   VARCHAR2,  
    comment        IN   VARCHAR2 := NULL);
```

### パラメータ

表 20-59 DEFINE\_COLUMN\_GROUP プロシージャのパラメータ

パラメータ	説明
sname	レプリケート表が置かれたスキーマの名前です。
oname	列グループを作成するレプリケート表の名前です。
column_group	作成する列グループの名前です。
comment	このユーザー・コメントは、DBA_REPCOLUMN_GROUP ビューに表示されます。

### 例外

表 20-60 DEFINE\_COLUMN\_GROUP プロシージャの例外

例外	説明
nonmasterdef	起動サイトはマスター定義サイトではありません。
missingobject	指定された表は存在しません。
duplicategroup	指定された列グループはこの表にすでに存在しています。
notquiesced	指定された表が所属するレプリケーション・グループは静止していません。

## DEFINE\_PRIORITY\_GROUP プロシージャ

このプロシージャは、マスター・グループの新しい優先グループを作成します。このプロシージャは、マスター定義サイトからコールする必要があります。

**関連項目：** 競合解消メソッドの詳細は、[第 6 章「競合解消の構成」](#)および『[Oracle9i アドバンスト・レプリケーション](#)』を参照してください。

### 構文

```
DBMS_REPCAT.DEFINE_PRIORITY_GROUP (  
  gname          IN   VARCHAR2,  
  pgroup         IN   VARCHAR2,  
  datatype       IN   VARCHAR2,  
  fixed_length  IN   INTEGER := NULL,  
  comment        IN   VARCHAR2 := NULL);
```

### パラメータ

表 20-61 DEFINE\_PRIORITY\_GROUP プロシージャのパラメータ

パラメータ	説明
gname	優先グループを作成するマスター・グループの名前です。
pgroup	作成する優先グループの名前です。
datatype	優先グループ・メンバーのデータ型です。サポートされているデータ型は、CHAR、VARCHAR2、NUMBER、DATE、RAW、NCHAR および NVARCHAR2 です。
fixed_length	CHAR データ型の列の長さを指定する必要があります。他のすべてのデータ型では、デフォルトの NULL を使用できます。
comment	このユーザー・コメントは、DBA_REPPRIORITY ビューに追加されます。

# 例外

表 20-62 DEFINE\_PRIORITY\_GROUP プロシージャの例外

例外	説明
nonmasterdef	起動サイトはマスター定義サイトではありません。
missingrepgroup	指定されたマスター・グループは存在しません。
duplicateprioritygroup	指定された優先グループは、マスター・グループにすでに存在しています。
typefailure	指定されたデータ型はサポートされていません。
notquiesced	マスター・グループは静止していません。

# DEFINE\_SITE\_PRIORITY プロシージャ

このプロシージャは、マスター・グループの新しいサイト優先グループを作成します。このプロシージャは、マスター定義サイトからコールする必要があります。

**関連項目：** 競合解消メソッドの詳細は、第 6 章「競合解消の構成」および『Oracle9i アドバンスト・レプリケーション』を参照してください。

# 構文

```
DBMS_REPCAT.DEFINE_SITE_PRIORITY (
    gname          IN   VARCHAR2,
    name           IN   VARCHAR2,
    comment        IN   VARCHAR2 := NULL);
```

# パラメータ

表 20-63 DEFINE\_SITE\_PRIORITY プロシージャのパラメータ

パラメータ	説明
gname	サイト優先グループを作成するマスター・グループの名前です。
name	作成するサイト優先グループの名前です。
comment	このユーザー・コメントは、DBA_REPPRIORITY ビューに追加されます。

例外

表 20-64 DEFINE\_SITE\_PRIORITY プロシージャの例外

例外	説明
nonmasterdef	起動サイトはマスター定義サイトではありません。
missingrepgroup	指定されたマスター・グループは存在しません。
duplicate prioritygroup	指定されたサイト優先グループは、マスター・グループにすでに存在しています。
notquiesced	マスター・グループは静止していません。

DO\_DEFERRED\_REPCAT\_ADMIN プロシージャ

このプロシージャは、カレント・マスター・サイトの指定されたマスター・グループ、またはすべてのマスター・サイトに対して、ローカルな遅延管理プロシージャを実行します。すべてのマスター・サイトに対する遅延管理プロシージャは、ジョブ・キューを利用して実行されます。

DO\_DEFERRED\_REPCAT\_ADMIN では、DO\_DEFERRED\_REPCAT\_ADMIN をコールした接続ユーザが発行した管理要求のみが実行されます。他のユーザが発行した要求は無視されます。

構文

```
DBMS_REPCAT.DO_DEFERRED_REPCAT_ADMIN (  
  gname          IN   VARCHAR2,  
  all_sites      IN   BOOLEAN := false);
```

パラメータ

表 20-65 DO\_DEFERRED\_REPCAT\_ADMIN プロシージャのパラメータ

パラメータ	説明
gname	マスター・グループの名前です。
all_sites	値が TRUE の場合は、ジョブを使用して、各マスター・サイトでローカルな管理プロシージャが実行されます。

# 例外

表 20-66 DO\_DEFERRED\_REPCAT\_ADMIN プロシージャの例外

例外	説明
nonmaster	起動サイトはマスター・サイトではありません。
commfailure	少なくとも1つのマスター・サイトにアクセスできません。 all_sites は TRUE に設定されています。

# DROP\_COLUMN\_GROUP プロシージャ

このプロシージャは、列グループを削除します。このプロシージャは、マスター定義サイトからコールする必要があります。

**関連項目：** 競合解消メソッドの詳細は、[第 6 章「競合解消の構成」](#)および『Oracle9i アドバンスト・レプリケーション』を参照してください。

# 構文

```
DBMS_REPCAT.DROP_COLUMN_GROUP (
  sname          IN   VARCHAR2,
  oname          IN   VARCHAR2,
  column_group IN   VARCHAR2);
```

# パラメータ

表 20-67 DROP\_COLUMN\_GROUP プロシージャのパラメータ

パラメータ	説明
sname	レプリケート表が置かれたスキーマの名前です。
oname	列グループを削除するレプリケート表の名前です。
column_group	削除対象の列グループの名前です。

例外

表 20-68 DROP\_COLUMN\_GROUP プロシージャの例外

例外	説明
nonmasterdef	起動サイトはマスター定義サイトではありません。
referenced	指定された列グループは、競合の検出および解消に使用されています。
missingobject	指定された表は存在しません。
missinggroup	指定された列グループは存在しません。
notquiesced	表が所属するマスター・グループは静止していません。

DROP\_GROUPED\_COLUMN プロシージャ

このプロシージャは、列グループからメンバーを削除します。このプロシージャは、マスター定義サイトからコールする必要があります。

**関連項目：** 競合解消メソッドの詳細は、[第 6 章「競合解消の構成」](#)および『Oracle9i アドバンスト・レプリケーション』を参照してください。

構文

```
DBMS_REPCAT.DROP_GROUPED_COLUMN (  
  sname          IN   VARCHAR2 ,  
  oname          IN   VARCHAR2 ,  
  column_group   IN   VARCHAR2 ,  
  list_of_column_names IN VARCHAR2 | DBMS_REPCAT.VARCHAR2s);
```



## パラメータ

表 20-69 DROP\_GROUPED\_COLUMN プロシージャのパラメータ

パラメータ	説明
sname	レプリケート表が置かれたスキーマの名前です。
oname	列グループが置かれたレプリケート表の名前です。表はネストした表の記憶表を利用できます。
column_group	メンバーを削除する列グループの名前です。
list_of_column_names	<p>指定された列グループから削除する列の名前です。列名は、カンマ区切りのリストと PL/SQL 索引付き表のいずれでも指定できます。PL/SQL 索引付き表のタイプは DBMS_REPCAT.VARCHAR2 にする必要があります。</p> <p>列オブジェクトは指定できますが、列オブジェクトの属性は指定できません。</p> <p>表がオブジェクトである場合、SYS_NC_OID\$ を指定して列グループにオブジェクト識別子列を追加できます。この列は各行オブジェクトのオブジェクト識別子を追跡します。</p> <p>表がネストした表の記憶表である場合は、NESTED_TABLE_ID を指定して、ネストした表の各行の識別子を追跡する列を追加できます。</p>

## 例外

表 20-70 DROP\_GROUPED\_COLUMN プロシージャの例外

例外	説明
nonmasterdef	起動サイトはマスター定義サイトではありません。
missingobject	指定された表は存在しません。
notquiesced	表が所属するマスター・グループは静止していません。

## DROP\_MASTER\_REPGROUP プロシージャ

このプロシージャは、カレント・サイトからマスター・グループを削除します。マスター・グループを、マスター定義サイトを含むすべてのマスター・サイトから削除するには、このプロシージャをマスター定義サイトでコールして、all\_sites を TRUE に設定します。

### 構文

```
DBMS_REPCAT.DROP_MASTER_REPGROUP (  
    gname                IN VARCHAR2,  
    drop_contents        IN BOOLEAN    := false,  
    all_sites            IN BOOLEAN    := false);
```

### パラメータ

表 20-71 DROP\_MASTER\_REPGROUP プロシージャのパラメータ

パラメータ	説明
gname	カレント・マスター・サイトから削除するマスター・グループの名前です。
drop_contents	デフォルトでは、マスター・サイトのレプリケーション・グループを削除しても、すべてのオブジェクトはデータベースに残ります。ただし、これらのオブジェクトはレプリケートされなくなります。すなわち、レプリケーション・グループ内のレプリケート・オブジェクトは、他のマスター・サイトとの間で変更をやり取りしなくなります。値を TRUE に設定すると、マスター・グループ内のレプリケート・オブジェクトが、対応付けられたスキーマから削除されます。
all_sites	値が TRUE に設定され、起動サイトがマスター定義サイトの場合、このプロシージャにより要求がすべてのマスターに同期的にマルチキャストされます。この場合、マスター定義サイトではプロシージャが即座に実行されますが、他のマスター・サイトでは実行が遅れることがあります。

例外

表 20-72 DROP\_MASTER\_REPGROUP プロシージャの例外

例外	説明
nonmaster	起動サイトはマスター・サイトではありません。
nonmasterdef	起動サイトはマスター定義サイトではありません。all_sites は TRUE に設定されています。
commfailure	少なくとも 1 つのマスター・サイトにアクセスできません。all_sites は TRUE に設定されています。
fullqueue	遅延リモート・プロシージャ・コール (RPC) キューには、マスター・グループへのエントリがあります。
masternotremoved	マスターはマスター定義サイトを認識しません。all_sites は TRUE に設定されています。

DROP\_MASTER\_REPOBJECT プロシージャ

このプロシージャは、マスター・グループからレプリケート・オブジェクトを削除します。  
このプロシージャは、マスター定義サイトからコールする必要があります。

構文

```
DBMS_REPCAT.DROP_MASTER_REPOBJECT (  
  sname          IN   VARCHAR2,  
  oname          IN   VARCHAR2,  
  type           IN   VARCHAR2,  
  drop_objects   IN   BOOLEAN      := false);
```

パラメータ

表 20-73 DROP\_MASTER\_REOBJECT プロシージャのパラメータ

パラメータ	説明
sname	オブジェクトが置かれたスキーマの名前です。
oname	マスター・グループから削除するオブジェクトの名前です。オブジェクトには、ネストした表の記憶表は指定できません。
type	削除対象のオブジェクトのタイプです。サポートされているタイプは次のとおりです。 <div><div>FUNCTION</div><div>INDEX</div><div>INDEXTYPE</div><div>OPERATOR</div><div>PACKAGE</div><div>PACKAGE BODY</div><div>PROCEDURE</div><div>SYNONYM</div><div>TABLE</div><div>TRIGGER</div><div>TYPE</div><div>TYPE BODY</div><div>VIEW</div></div>
drop_objects	デフォルトでは、オブジェクトはスキーマには残りますが、マスター・グループからは削除されます。すなわち、オブジェクトに対する変更が、他のマスター・サイトやマテリアライズド・ビュー・サイトにレプリケートされなくなります。レプリケーション環境からオブジェクトを完全に削除するには、このパラメータを TRUE に設定します。パラメータを TRUE に設定すると、オブジェクトは各マスター・サイトのデータベースから削除されます。

例外

表 20-74 DROP\_MASTER\_REOBJECT プロシージャの例外

例外	説明
nonmasterdef	起動サイトはマスター定義サイトではありません。
missingobject	指定されたオブジェクトは存在しません。
typefailure	指定されたタイプ・パラメータはサポートされていません。
commfailure	少なくとも 1 つのマスター・サイトにアクセスできません。

# DROP\_MVIEW\_REPGROUP プロシージャ

このプロシージャは、レプリケーション環境からマテリアライズド・ビュー・サイトを削除します。DROP\_MVIEW\_REPGROUP は、マスター・サイトまたはマスター・マテリアライズド・ビュー・サイトで自動的に UNREGISTER\_MVIEW\_REPGROUP をコールしてマテリアライズド・ビューの登録を解除しますが、登録解除の際に起こったエラーはすべて無視します。DROP\_MVIEW\_REPGROUP が正常に終了しなかった場合は、マスター・サイトまたはマスター・マテリアライズド・ビュー・サイトに接続して UNREGISTER\_MVIEW\_REPGROUP を実行します。

## 構文

```
DBMS_REPCAT.DROP_MVIEW_REPGROUP (
    gname          IN   VARCHAR2,
    drop_contents  IN   BOOLEAN   := false,
    gowner         IN   VARCHAR2  := 'PUBLIC');
```

## パラメータ

表 20-75 DROP\_MVIEW\_REPGROUP プロシージャのパラメータ

パラメータ	説明
gname	現行のマテリアライズド・ビュー・サイトから削除するレプリケーション・グループの名前です。トリガーやパッケージなど、レプリケーションのサポート用に生成されたすべてのオブジェクトが削除されます。
drop_contents	デフォルトでは、マテリアライズド・ビュー・サイトのレプリケーション・グループを削除しても、すべてのオブジェクトは対応付けられたスキーマに残ります。ただし、これらのオブジェクトはレプリケートされなくなります。値を TRUE に設定すると、レプリケーション・グループ内のレプリケート・オブジェクトが、対応付けられたスキーマから削除されます。
gowner	マテリアライズド・ビュー・グループの所有者です。

## 例外

表 20-76 DROP\_MVIEW\_REPGROUP プロシージャの例外

例外	説明
nonmview	起動サイトはマテリアライズド・ビュー・サイトではありません。
missingrepgroup	指定されたレプリケーション・グループは存在しません。

## DROP\_MVIEW\_REPOBJECT プロシージャ

このプロシージャは、マテリアライズド・ビュー・サイトからレプリケート・オブジェクトを削除します。

### 構文

```
DBMS_REPCAT.DROP_MVIEW_REPOBJECT (  
    sname          IN   VARCHAR2,  
    oname          IN   VARCHAR2,  
    type           IN   VARCHAR2,  
    drop_objects   IN   BOOLEAN := false);
```

### パラメータ

表 20-77 DROP\_MVIEW\_REPOBJECT プロシージャのパラメータ

パラメータ	説明
sname	オブジェクトが置かれたスキーマの名前です。
oname	レプリケーション・グループから削除するオブジェクトの名前です。
type	削除対象のオブジェクトのタイプです。サポートされているタイプは次のとおりです。 <div><div>FUNCTION</div><div>INDEX</div><div>INDEXTYPE</div><div>OPERATOR</div><div>PACKAGE</div><div>PACKAGE BODY</div><div>PROCEDURE</div><div>SNAPSHOT</div><div>SYNONYM</div><div>TRIGGER</div><div>TYPE</div><div>TYPE BODY</div><div>VIEW</div></div>
drop_objects	デフォルトでは、オブジェクトは対応付けられたスキーマには残りますが、対応付けられたレプリケーション・グループからは削除されます。カレント・マテリアライズド・ビュー・サイトのスキーマからオブジェクトを完全に削除するには、このパラメータを TRUE に設定します。パラメータを TRUE に設定すると、オブジェクトはマテリアライズド・ビュー・サイトのデータベースから削除されます。

## 例外

**表 20-78 DROP\_MVIEW\_REPOBJECT プロシージャの例外**

例外	説明
nonmview	起動サイトはマテリアライズド・ビュー・サイトではありません。
missingobject	指定されたオブジェクトは存在しません。
typefailure	指定されたタイプ・パラメータはサポートされていません。

## DROP\_PRIORITY プロシージャ

このプロシージャは、優先順位レベルに基づいて、優先グループのメンバーを削除します。このプロシージャは、マスター定義サイトからコールする必要があります。

**関連項目：** 競合解消メソッドの詳細は、第 6 章「競合解消の構成」および『Oracle9i アドバンスド・レプリケーション』を参照してください。

## 構文

```
DBMS_REPCAT.DROP_PRIORITY(  
    gname          IN   VARCHAR2,  
    pgroup         IN   VARCHAR2,  
    priority_num   IN   NUMBER);
```

## パラメータ

**表 20-79 DROP\_PRIORITY プロシージャのパラメータ**

パラメータ	説明
gname	優先グループの対応付けられたマスター・グループです。
pgroup	削除対象のメンバーが含まれる優先グループの名前です。
priority_num	グループから削除する優先グループ・メンバーの優先順位レベルです。

例外

表 20-80 DROP\_PRIORITY プロシージャの例外

例外	説明
nonmasterdef	起動サイトはマスター定義サイトではありません。
missingrepgroup	指定されたマスター・グループは存在しません。
missingprioritygroup	指定された優先グループは存在しません。
notquiesced	マスター・グループは静止していません。

DROP\_PRIORITY\_GROUP プロシージャ

このプロシージャは、指定されたマスター・グループの優先グループを削除します。このプロシージャは、マスター定義サイトからコールする必要があります。

**関連項目：** 競合解消メソッドの詳細は、第 6 章「競合解消の構成」および『Oracle9i アドバンスド・レプリケーション』を参照してください。

構文

```
DBMS_REPCAT.DROP_PRIORITY_GROUP (  
  gname      IN   VARCHAR2,  
  pgroup     IN   VARCHAR2);
```

パラメータ

表 20-81 DROP\_PRIORITY\_GROUP プロシージャのパラメータ

パラメータ	説明
gname	優先グループの対応付けられたマスター・グループです。
pgroup	削除対象の優先グループの名前です。



例外

表 20-82 DROP\_PRIORITY\_GROUP プロシージャの例外

例外	説明
nonmasterdef	起動サイトはマスター定義サイトではありません。
missingrepgroup	指定されたマスター・グループは存在しません。
referenced	指定された優先グループは、競合解消に使用されています。
notquiesced	指定されたマスター・グループは静止していません。

DROP\_PRIORITY\_datatype プロシージャ

このプロシージャは、値に基づいて、優先グループのメンバーを削除します。このプロシージャは、マスター定義サイトからコールする必要があります。コールするプロシージャは、priority列のデータ型によって異なります。

**関連項目：** 競合解消メソッドの詳細は、第 6 章「競合解消の構成」および『Oracle9i アドバンスト・レプリケーション』を参照してください。

構文

```
DBMS_REPCAT.DROP_PRIORITY_datatype (  
    gname      IN   VARCHAR2,  
    pgroup     IN   VARCHAR2,  
    value      IN   datatype);
```

datatype は次のとおりです。

```
{ NUMBER  
| VARCHAR2  
| CHAR  
| DATE  
| RAW  
| NCHAR  
| NVARCHAR2 }
```

パラメータ

表 20-83 DROP\_PRIORITY\_datatype プロシージャのパラメータ

パラメータ	説明
gname	優先グループの対応付けられたマスター・グループです。
pgroup	削除対象のメンバーが含まれる優先グループの名前です。
value	グループから削除する優先グループ・メンバーの値です。

例外

表 20-84 DROP\_PRIORITY\_datatype プロシージャの例外

例外	説明
nonmasterdef	起動サイトはマスター定義サイトではありません。
missingrepgroup	指定されたマスター・グループは存在しません。
missingprioritygroup	指定された優先グループは存在しません。
paramtype, typefailure	値のデータ型は、この優先グループには不適切です。
notquiesced	指定されたマスター・グループは静止していません。

# DROP\_SITE\_PRIORITY プロシージャ

このプロシージャは、指定されたマスター・グループのサイト優先グループを削除します。  
このプロシージャは、マスター定義サイトからコールする必要があります。

**関連項目：** 競合解消メソッドの詳細は、[第 6 章「競合解消の構成」](#)および『Oracle9i アドバンスド・レプリケーション』を参照してください。

## 構文

```
DBMS_REPCAT.DROP_SITE_PRIORITY (
    gname      IN   VARCHAR2,
    name       IN   VARCHAR2);
```

## パラメータ

表 20-85 DROP\_SITE\_PRIORITY プロシージャのパラメータ

パラメータ	説明
gname	サイト優先グループの対応付けられたマスター・グループです。
name	削除対象のサイト優先グループの名前です。

## 例外

表 20-86 DROP\_SITE\_PRIORITY プロシージャの例外

例外	説明
nonmasterdef	起動サイトはマスター定義サイトではありません。
missingrepgroup	指定されたマスター・グループは存在しません。
referenced	指定されたサイト優先グループは、競合解消に使用されています。
notquiesced	指定されたマスター・グループは静止していません。

## DROP\_SITE\_PRIORITY\_SITE プロシージャ

このプロシージャは、指定された名前のサイトをサイト優先グループから削除します。このプロシージャは、マスター定義サイトからコールする必要があります。

**関連項目：** 競合解消メソッドの詳細は、[第 6 章「競合解消の構成」](#)および『[Oracle9i アドバンスト・レプリケーション](#)』を参照してください。

### 構文

```
DBMS_REPCAT.DROP_SITE_PRIORITY_SITE (  
  gname      IN   VARCHAR2,  
  name       IN   VARCHAR2,  
  site       IN   VARCHAR2);
```

### パラメータ

表 20-87 DROP\_SITE\_PRIORITY\_SITE プロシージャのパラメータ

パラメータ	説明
gname	サイト優先グループの対応付けられたマスター・グループです。
name	メンバーを削除するサイト優先グループの名前です。
site	グループから削除するサイトのグローバル・データベース名です。

### 例外

表 20-88 DROP\_SITE\_PRIORITY\_SITE プロシージャの例外

例外	説明
nonmasterdef	起動サイトはマスター定義サイトではありません。
missingrepgroup	指定されたマスター・グループは存在しません。
missingpriority	指定されたサイト優先グループは存在しません。
notquiesced	指定されたマスター・グループは静止していません。

# DROP\_conflictttype\_RESOLUTION プロシージャ

このプロシージャは、更新、削除または一意性競合解消ルーチンを削除します。これらのプロシージャは、マスター定義サイトからコールする必要があります。コールするプロシージャは、ルーチンが解消する競合のタイプによって異なります。

## 競合解消ルーチン

各競合解消ルーチンのプロシージャ名を次の表に示します。

表 20-89 競合解消ルーチン

ルーチン	プロシージャ名
update	DROP_UPDATE_RESOLUTION
uniqueness	DROP_UNIQUE_RESOLUTION
delete	DROP_DELETE_RESOLUTION

## 構文

```
DBMS_REPCAT.DROP_UPDATE_RESOLUTION (
    sname          IN   VARCHAR2,
    oname          IN   VARCHAR2,
    column_group   IN   VARCHAR2,
    sequence_no    IN   NUMBER);
```

```
DBMS_REPCAT.DROP_DELETE_RESOLUTION (
    sname          IN   VARCHAR2,
    oname          IN   VARCHAR2,
    sequence_no    IN   NUMBER);
```

```
DBMS_REPCAT.DROP_UNIQUE_RESOLUTION (
    sname          IN   VARCHAR2,
    oname          IN   VARCHAR2,
    constraint_name IN   VARCHAR2,
    sequence_no    IN   NUMBER);
```

パラメータ

表 20-90 DROP\_conflictype\_RESOLUTION プロシージャのパラメータ

パラメータ	説明
sname	表が置かれたスキーマの名前です。
oname	競合解消ルーチンを削除する表の名前です。
column_group	更新の競合解消ルーチンを削除する列グループの名前です。
constraint_name	一意性競合解消ルーチンを削除する一意制約の名前です。
sequence_no	削除対象の競合解消メソッドに割り当てられた順序番号です。この番号でルーチンが識別されます。

例外

表 20-91 DROP\_conflictype\_RESOLUTION プロシージャの例外

例外	説明
nonmasterdef	起動サイトはマスター定義サイトではありません。
missingobject	指定されたオブジェクトが指定されたスキーマ内の表として存在しないか、または指定された順序番号の競合解消ルーチンが登録されていません。
notquiesced	マスター・グループは静止していません。

# EXECUTE\_DDL プロシージャ

このプロシージャは、いくつかの、またはすべてのマスター・サイトで実行される DDL を渡します。このプロシージャは、マスター定義サイトからのみコールできます。

## 構文

```
DBMS_REPCAT.EXECUTE_DDL (
  gname          IN   VARCHAR2,
  { master_list  IN   VARCHAR2      := NULL,
    | master_table IN   DBMS_UTILITY.DBLINK_ARRAY, }
  DDL_TEXT       IN   VARCHAR2);
```

**注意：** このプロシージャはオーバーロードされています。master\_list パラメータと master\_table パラメータは、同時には指定できません。

## パラメータ

表 20-92 EXECUTE\_DDL プロシージャのパラメータ

パラメータ	説明
gname	マスター・グループの名前です。
master_list	渡された DDL を実行するマスター・サイトの、カンマ区切りのリストです。各サイト名の間に空白は入れないでください。デフォルトの NULL の場合、マスター定義サイトを含むすべてのサイトで DDL が実行されます。
master_table	渡された DDL を実行するマスター・サイトをリストした表です。最初のマスターは 1 の位置、2 番目のマスターは 2 の位置と、順に配置します。
ddl_text	指定された各マスター・サイトで実行する DDL です。スキーマを指定せずに DDL が渡された場合、レプリケーション管理者のスキーマがデフォルトのスキーマになります。レプリケーション管理者のスキーマ以外のスキーマを使用する場合は、必ずスキーマを指定してください。

例外

表 20-93 EXECUTE\_DDL プロシージャの例外

例外	説明
nonmasterdef	起動サイトはマスター定義サイトではありません。
nonmaster	少なくとも1つのサイトが、マスター・サイトではありません。
ddlfailure	マスター定義サイトの DDL が失敗しました。
commfailure	少なくとも1つのマスター・サイトにアクセスできません。

GENERATE\_MVIEW\_SUPPORT プロシージャ

このプロシージャは、トリガーをアクティブにして、更新可能なマテリアライズド・ビューのレプリケーション、またはプロシージャ・レプリケーションのサポートに必要なパッケージを生成します。このプロシージャは、マテリアライズド・ビュー・サイトからコールする必要があります。

**注意：** CREATE\_MVIEW\_REPOBJECT は自動的に、更新可能なマテリアライズド・ビューにマテリアライズド・ビュー・サポートを生成します。

構文

```
DBMS_REPCAT.GENERATE_MVIEW_SUPPORT (  
  sname          IN VARCHAR2,  
  oname          IN VARCHAR2,  
  type           IN VARCHAR2,  
  gen_objs_owner IN VARCHAR2 := '',  
  min_communication IN BOOLEAN := true,  
  generate_80_compatible IN BOOLEAN := true);
```



# パラメータ

表 20-94 GENERATE\_MVIEW\_SUPPORT プロシージャのパラメータ

パラメータ	説明
sname	オブジェクトが置かれたスキーマの名前です。
oname	サポートを生成するオブジェクトの名前です。
type	オブジェクトのタイプです。サポートされているタイプは、SNAPSHOT、PACKAGE および PACKAGE BODY です。
gen_objs_owner	PACKAGE タイプまたは PACKAGE BODY タイプのオブジェクトの場合、オブジェクトが作成されるスキーマです。値が NULL の場合、オブジェクトは SNAME に作成されます。
min_communication	値が TRUE の場合、UPDATE 文で列が変更されるときにかぎり、更新トリガーで列の新しい値が送られます。列がキー列の場合、または変更された列グループ内の列の場合は、更新トリガーで列の古い値が送信されます。
generate_80_compatible	マテリアライズド・ビューのマスター・サイトで Oracle8i リリース 8.1.5 より前の Oracle Server が実行されている場合は、TRUE に設定します。マテリアライズド・ビューのマスター・サイトまたはマスター・マテリアライズド・ビュー・サイトで Oracle8i リリース 8.1.5 以上が実行されている場合は、FALSE に設定します。

# 例外

表 20-95 GENERATE\_MVIEW\_SUPPORT プロシージャの例外

例外	説明
nonmview	起動サイトはマテリアライズド・ビュー・サイトではありません。
missingobject	指定されたオブジェクトは、行または列レベル・レプリケーションの情報が必要なレプリケート・スキーマ内のマテリアライズド・ビューとして存在しないか、またはラッパーの生成が必要なパッケージ（本体）として存在しません。
typfailure	指定されたタイプ・パラメータはサポートされていません。
missingschema	指定された生成オブジェクトの所有者は存在しません。
missingremoteobject	マスター・サイトまたはマスター・マテリアライズド・ビュー・サイトのオブジェクトは、まだレプリケーション・サポートを生成していません。
commfailure	マスター・サイトまたはマスター・マテリアライズド・ビュー・サイトにアクセスできません。

## GENERATE\_REPLICATION\_SUPPORT プロシージャ

このプロシージャは、指定されたオブジェクトのレプリケーションのサポートに必要なトリガーおよびパッケージを生成します。このプロシージャは、マスター定義サイトからコールする必要があります。

### 構文

```
DBMS_REPCAT.GENERATE_REPLICATION_SUPPORT (
  sname          IN    VARCHAR2,
  oname          IN    VARCHAR2,
  type           IN    VARCHAR2,
  package_prefix IN    VARCHAR2 := NULL,
  procedure_prefix IN  VARCHAR2 := NULL,
  distributed    IN    BOOLEAN  := true,
  gen_objs_owner IN    VARCHAR2 := NULL,
  min_communication IN  BOOLEAN  := true,
  generate_80_compatible IN  BOOLEAN := true);
```

### パラメータ

表 20-96 GENERATE\_REPLICATION\_SUPPORT プロシージャのパラメータ

パラメータ	説明
sname	オブジェクトが置かれたスキーマの名前です。
oname	レプリケーション・サポートを生成するオブジェクトの名前です。
type	オブジェクトのタイプです。サポートされているタイプは、TABLE、PACKAGE および PACKAGE BODY です。
package_prefix	PACKAGE タイプまたは PACKAGE BODY タイプのオブジェクトの場合は、生成されたラッパー・パッケージの名前にこの値が付加されます。デフォルトは DEFER_ です。
procedure_prefix	PACKAGE タイプまたは PACKAGE BODY タイプのオブジェクトの場合は、生成されたラッパー・プロシージャの名前にこの値が付加されます。デフォルトでは接頭辞が付加されません。
distributed	このパラメータは、必ず TRUE に設定します。
gen_objs_owner	PACKAGE タイプまたは PACKAGE BODY タイプのオブジェクトの場合、オブジェクトが作成されるスキーマです。値が NULL の場合、オブジェクトは sname に作成されます。

表 20-96 GENERATE\_REPLICATION\_SUPPORT プロシージャのパラメータ (続き)

パラメータ	説明
min_communication	マスター・サイトのいずれかで Oracle7 リリース 7.3 が実行されている場合は、FALSE に設定します。新旧の値の伝播を最小にするには、TRUE に設定します。デフォルトは TRUE です。詳細は、『Oracle9i アドバンスド・レプリケーション』を参照してください。
generate_80_compatible	マスター・サイトのいずれかで Oracle8i リリース 8.1.5 より前の Oracle サーバーが実行されている場合は、TRUE に設定します。全マスター・サイトで Oracle8i リリース 8.1.5 以上が実行されている場合は、FALSE に設定します。

## 例外

表 20-97 GENERATE\_REPLICATION\_SUPPORT プロシージャの例外

例外	説明
nonmasterdef	起動サイトはマスター定義サイトではありません。
missingobject	指定されたオブジェクトは、行レベル・レプリケーションの情報が必要な指定されたスキーマ内の表として存在しないか、またはラッパーの生成が必要なパッケージ (本体) として存在しません。
typefailure	指定されたタイプ・パラメータはサポートされていません。
notquiesced	レプリケーション・グループは静止していません。
commfailure	少なくとも 1 つのマスター・サイトにアクセスできません。
missingschema	スキーマが存在しません。
dbnotcompatible	リリース 7.3.0.0 との互換性がないマスター・サイトがあります。
notcompat	リリース 7.3.0.0 との互換性がないマスター・サイトがあります。(dbnotcompatible と同様です。)
duplicateobject	オブジェクトはすでに存在しています。

## MAKE\_COLUMN\_GROUP プロシージャ

このプロシージャは、1 つ以上のメンバーを持つ新しい列グループを作成します。このプロシージャは、マスター定義サイトからコールする必要があります。

**関連項目：** 競合解消メソッドの詳細は、[第 6 章「競合解消の構成」](#)および『[Oracle9i アドバンスド・レプリケーション](#)』を参照してください。

### 構文

```
DBMS_REPCAT.MAKE_COLUMN_GROUP (  
    sname                IN    VARCHAR2,  
    oname                IN    VARCHAR2,  
    column_group         IN    VARCHAR2,  
    list_of_column_names IN    VARCHAR2 | DBMS_REPCAT.VARCHAR2s);
```

### パラメータ

表 20-98 MAKE\_COLUMN\_GROUP プロシージャのパラメータ

パラメータ	説明
sname	レプリケート表が置かれたスキーマの名前です。
oname	新しい列グループを作成するレプリケート表の名前です。表はネストした表の記憶表を利用できます。
column_group	作成する列グループに割り当てる名前です。
list_of_column_names	グループ化する列の名前です。列名は、カンマ区切りのリストと PL/SQL 索引付き表のいずれでも指定できます。PL/SQL 索引付き表のタイプは DBMS_REPCAT.VARCHAR2 にする必要があります。表のすべての列を含む列グループを作成するには、単一の値 '*' を使用します。  列オブジェクトは指定できますが、列オブジェクトの属性は指定できません。  表がオブジェクト表である場合、SYS_NC_OID\$ を指定して列グループにオブジェクト識別子列を追加できます。この列は各行オブジェクトのオブジェクト識別子を追跡します。  表がネストした表の記憶表である場合は、NESTED_TABLE_ID を指定して、ネストした表の各行の識別子を追跡する列を追加できます。

例外

表 20-99 MAKE\_COLUMN\_GROUP プロシージャの例外

例外	説明
nonmasterdef	起動サイトはマスター定義サイトではありません。
duplicategroup	指定された列グループはこの表にすでに存在しています。
missingobject	指定された表は存在しません。
missingcolumn	指定された列は、指定された表に存在しません。
duplicatecolumn	指定された列は、すでに他の列グループのメンバーになっています。
notquiesced	マスター・グループは静止していません。

PREPARE\_INSTANTIATED\_MASTER プロシージャ

このプロシージャは、他の準備された新しいマスター・サイトおよび既存のマスター・サイトから起動マスター・サイトへの遅延トランザクションの伝播を有効にします。このプロシージャはまた、起動マスター・サイトから他の準備された新しいマスター・サイトおよび既存のマスター・サイトへの遅延トランザクションの伝播も有効にします。

データベース全体のエクスポートやインポートまたは変更ベースのリカバリを実行した場合は、新しいマスター・サイトにはマスター定義サイトの遅延トランザクション・キューにあった遅延トランザクションがすべて含まれます。これらの遅延トランザクションは新しいマスター・サイトには存在できないため、このプロシージャではデータベース全体のエクスポートやインポートまたは変更ベースのリカバリが使用された場合、遅延トランザクション・キューおよびエラー・キューのすべてのトランザクションを削除します。

オブジェクトレベルのエクスポートまたはインポートの際は、このプロシージャを実行する前に、拡張グループの DBA\_REPCATLOG データ・ディクショナリ・ビューのすべての要求がエラーなしに処理されたことを確認してください。

注意：

- 新しいマスター・サイトのインスタンス化（エクスポートやインポートまたは変更ベースのリカバリ）が完了するまで、このプロシージャをコールしないでください。
- このプロシージャの実行が正常に戻されるまで、データ操作言語（DML）文を新しいマスター・サイトの拡張マスター・グループのオブジェクトに直接使用しないでください。DML 文はレプリケートされない場合があります。
- このプロシージャの実行が正常に戻されるまで、遅延トランザクションの作成に DBMS\_DEFER パッケージは使用しないでください。これらの遅延トランザクションはレプリケートされない場合があります。

**注意：** 変更ベースのリカバリを使用するには、既存のマスター・サイトおよび新しいマスター・サイトを同じオペレーティング・システム上で実行する必要があります。ただしオペレーティング・システムのリリースは同じでなくても構いません。

構文

```
DBMS_REPCAT.PREPARE_INSTANTIATED_MASTER (  
    extension_id          IN          RAW);
```

パラメータ

表 20-100 PREPARE\_INSTANTIATED\_MASTER プロシージャのパラメータ

パラメータ	説明
extension_id	静止せずにマスター・グループにマスター・データベースを追加するための、現在保留になっている要求の識別子です。extension_id は、DBA_REPSITES_NEW データ・ディクショナリ・ビューおよび DBA_REPEXTENSIONS データ・ディクショナリ・ビューを問い合わせで検索できます。

# 例外

**表 20-101 PREPARE\_INSTANTIATED\_MASTER プロシージャの例外**

例外	説明
typfailure	パラメータの 1 つに指定されたパラメータ値が不適正です。
dbnotcompatible	機能がデータベースのバージョンと互換性がありません。データベースの互換性のレベルは、すべて 9.0.1 以上である必要があります。

# PURGE\_MASTER\_LOG プロシージャ

このプロシージャは、指定された ID 番号、ソースまたはマスター・グループに対応付けられた、DBA\_REPCATLOG ビュー内のローカル・メッセージを削除します。

特定のソースから管理要求をすべてパージするには、id パラメータに NULL を指定します。全ソースから管理要求をすべてパージするには、id パラメータと source パラメータの両方に NULL を指定します。

# 構文

```
DBMS_REPCAT.PURGE_MASTER_LOG (
    id      IN   BINARY_INTEGER,
    source  IN   VARCHAR2,
    gname   IN   VARCHAR2);
```

# パラメータ

**表 20-102 PURGE\_MASTER\_LOG プロシージャのパラメータ**

パラメータ	説明
id	DBA_REPCATLOG ビューに表示される要求の ID 番号です。
source	要求が発行されたマスター・サイトです。
gname	要求の対象であるマスター・グループの名前です。

# 例外

**表 20-103 PURGE\_MASTER\_LOG プロシージャの例外**

例外	説明
nonmaster	gname が NULL でなく、起動サイトがマスター・サイトではありません。

## PURGE\_STATISTICS プロシージャ

このプロシージャは、DBA\_REPRESOLUTION\_STATISTICS ビューから情報を削除します。

### 構文

```
DBMS_REPCAT.PURGE_STATISTICS (  
    sname      IN   VARCHAR2,  
    oname      IN   VARCHAR2,  
    start_date IN   DATE,  
    end_date   IN   DATE);
```

### パラメータ

表 20-104 PURGE\_STATISTICS プロシージャのパラメータ

パラメータ	説明
sname	レプリケート表が置かれたスキーマの名前です。
oname	競合解消の統計をパージする表の名前です。
start_date/end_date	統計をパージする日付の範囲です。start_date が NULL の場合は、end_date までのすべての統計がパージされます。end_date が NULL の場合は、start_date 以後のすべての統計がパージされます。

### 例外

表 20-105 PURGE\_STATISTICS プロシージャの例外

例外	説明
missingschema	指定されたスキーマは存在しません。
missingobject	指定された表は存在しません。
statnotreg	統計の収集に表が登録されていません。



## REFRESH\_MVIEW\_REPGROUP プロシージャ

このプロシージャは、マテリアライズド・ビュー・グループを、対応付けられたマスター・サイトまたはマスター・マテリアライズド・ビュー・サイトからの最新データでリフレッシュします。

### 構文

```
DBMS_REPCAT.REFRESH_MVIEW_REPGROUP (  
    gname                IN    VARCHAR2,  
    drop_missing_contents IN    BOOLEAN    := false,  
    refresh_mviews        IN    BOOLEAN    := false,  
    refresh_other_objects IN    BOOLEAN    := false,  
    gowner                IN    VARCHAR2   := 'PUBLIC');
```

### パラメータ

表 20-106 REFRESH\_MVIEW\_REPGROUP プロシージャのパラメータ

パラメータ	説明
gname	レプリケーション・グループの名前です。
drop_missing_contents	オブジェクトはマスター・サイトまたはマスター・マテリアライズド・ビュー・サイトのレプリケーション・グループから削除されても、マテリアライズド・ビュー・サイトのスキーマからは自動的に削除はされません。このようなオブジェクトは単に、レプリケートされなくなります。すなわち、オブジェクトに加えられた変更が、対応付けられたマスター・サイトまたはマスター・マテリアライズド・ビュー・サイトに送信されなくなります。マテリアライズド・ビューは、対応付けられたマスター表またはマスター・マテリアライズド・ビューからリフレッシュし続けることができます。ただし、更新可能なマテリアライズド・ビューに加えられた変更は失われます。オブジェクトをレプリケーション・グループから削除する際は、このパラメータを TRUE に設定すると、スキーマから完全に削除できます。

表 20-106 REFRESH\_MVIEW\_REPGROUP プロシージャのパラメータ（続き）

パラメータ	説明
refresh_mviews	値を TRUE に設定すると、レプリケーション・グループ内のマテリアライズド・ビューの内容がリフレッシュされます。
refresh_other_objects	この値を TRUE に設定すると、レプリケーション・グループ内の非マテリアライズド・ビュー・オブジェクトの内容がリフレッシュされます。非マテリアライズド・ビュー・オブジェクトには、次のようなものがあります。 <ul style="list-style-type: none"><li>■ 表</li><li>■ ビュー</li><li>■ 索引</li><li>■ PL/SQL パッケージおよびパッケージ本体</li><li>■ PL/SQL プロシージャおよびファンクション</li><li>■ トリガー</li><li>■ シノニム</li></ul>
gowner	マテリアライズド・ビュー・グループの所有者です。

例外

表 20-107 REFRESH\_MVIEW\_REPGROUP プロシージャの例外

例外	説明
nonmview	起動サイトはマテリアライズド・ビュー・サイトではありません。
nonmaster	マスターがマスター・サイトまたはマスター・マテリアライズド・ビュー・サイトではなくなっています。
commfailure	マスター・サイトまたはマスター・マテリアライズド・ビュー・サイトにアクセスできません。
missingrepgroup	レプリケーション・グループの名前が指定されていません。

## REGISTER\_MVIEW\_REPGROUP プロシージャ

このプロシージャは DBA\_REGISTERED\_MVIEW\_GROUPS でのマテリアライズド・ビュー・グループの挿入または変更を行って、マスター・サイトまたはマスター・マテリアライズド・ビュー・サイトでのマテリアライズド・ビューの管理を容易にします。

### 構文

```
DBMS_REPCAT.REGISTER_MVIEW_REPGROUP (  
    gname          IN   VARCHAR2,  
    mviewsite      IN   VARCHAR2,  
    comment        IN   VARCHAR2  := NULL,  
    rep_type       IN   NUMBER     := reg_unknown,  
    fname          IN   VARCHAR2  := NULL,  
    gowner         IN   VARCHAR2  := 'PUBLIC');
```

### パラメータ

表 20-108 REGISTER\_MVIEW\_REPGROUP プロシージャのパラメータ

パラメータ	説明
gname	登録するマテリアライズド・ビュー・グループの名前です。
mviewsite	マテリアライズド・ビュー・サイトのグローバル名です。
comment	マテリアライズド・ビュー・サイトのコメント、または既存のコメントの更新です。
rep_type	マテリアライズド・ビュー・グループのバージョンです。割り当てることができる有効な定数を次に示します。 <ul style="list-style-type: none"><li>■ dbms_repcat.reg_unknown (デフォルト)</li><li>■ dbms_repcat.reg_v7_group</li><li>■ dbms_repcat.reg_v8_group</li></ul>
fname	このパラメータは内部で使⽤します。 <b>注意：</b> オラクル社カスタマ・サポート・センターから指示された場合を除き、このパラメータを設定しないでください。
gowner	マテリアライズド・ビュー・グループの所有者です。

例外

表 20-109 REGISTER\_MVIEW\_REPGROUP プロシージャの例外

例外	説明
failregmviewrepgroup	マテリアライズド・ビュー・グループの登録に失敗しました。
missingrepgroup	レプリケーション・グループの名前が指定されていません。
nullsitename	マテリアライズド・ビュー・サイトが指定されていません。
nonmaster	プロシージャはマテリアライズド・ビューのマスター・サイトまたはマスター・マテリアライズド・ビュー・サイトで実行する必要があります。
duplicaterepgroup	レプリケーション・グループはすでに存在しています。

REGISTER\_STATISTICS プロシージャ

このプロシージャは、正常に実行された、表の更新、一意性および削除の競合解消の情報を収集します。

構文

```
DBMS_REPCAT.REGISTER_STATISTICS (  
    sname IN    VARCHAR2,  
    oname IN    VARCHAR2);
```

パラメータ

表 20-110 REGISTER\_STATISTICS プロシージャのパラメータ

パラメータ	説明
sname	表が置かれたスキーマの名前です。
oname	競合解消の統計を収集する表の名前です。

例外

表 20-111 REGISTER\_STATISTICS プロシージャの例外

例外	説明
missingschema	指定されたスキーマは存在しません。
missingobject	指定された表は存在しません。

# RELOCATE\_MASTERDEF プロシージャ

このプロシージャは、マスター定義サイトをレプリケーション環境の他のマスター・サイトに変更します。

RELOCATE\_MASTERDEF をコールするときに、古いマスター定義サイトまたは新しいマスター定義サイトが使用可能である必要はありません。再構成を行うときには、notify\_masters および include\_old\_masterdef を TRUE に設定して、RELOCATE\_MASTERDEF をコールします。

## 構文

```
DBMS_REPCAT.RELOCATE_MASTERDEF (
  gname          IN  VARCHAR2,
  old_masterdef  IN  VARCHAR2,
  new_masterdef  IN  VARCHAR2,
  notify_masters IN  BOOLEAN    := true,
  include_old_masterdef IN  BOOLEAN := true,
  require_flavor_change IN  BOOLEAN := false);
```

## パラメータ

表 20-112 RELOCATE\_MASTERDEF プロシージャのパラメータ

パラメータ	説明
gname	マスター定義を再配置するレプリケーション・グループの名前です。
old_masterdef	現行のマスター定義サイトの、完全修飾されたデータベース名です。
new_masterdef	新しいマスター定義サイトにする既存のマスター・サイトの、完全修飾されたデータベース名です。
notify_masters	TRUE の場合、すべてのマスター（include_old_masterdef が TRUE の場合には old_masterdef も含まれます）に、変更が同期的にマルチキャストされます。いずれかのマスターで変更が適用されなかった場合は、すべてのマスターでその変更がロールバックされます。  マスター定義サイトのみに障害が発生している場合は、notify_masters を TRUE、include_old_masterdef を FALSE に設定して、RELOCATE_MASTERDEF をコールします。複数のマスター・サイトとマスター定義サイトで障害が発生した場合、それぞれの操作マスターで、管理者が notify_masters を FALSE に設定して RELOCATE_MASTERDEF をコールします。

表 20-112 RELOCATE\_MASTERDEF プロシージャのパラメータ（続き）

パラメータ	説明
include_old_masterdef	notify_masters が TRUE で、include_old_masterdef も TRUE の場合は、古いマスター定義サイトにも変更が通知されます。
require_flavor_change	このパラメータは内部で使われます。  <b>注意：</b> オラクル社カスタマ・サポート・センターから指示された場合を除き、このパラメータを設定しないでください。

例外

表 20-113 RELOCATE\_MASTERDEF プロシージャの例外

例外	説明
nonmaster	new_masterdef または起動サイトがマスター・サイトではありません。
nonmasterdef	old_masterdef はマスター定義サイトではありません。
commfailure	少なくとも 1 つのマスター・サイトにアクセスできません。 notify_masters は TRUE に設定されています。

REMOVE\_MASTER\_DATABASES プロシージャ

このプロシージャは、レプリケーション環境から 1 つ以上のマスター・データベースを削除します。このプロシージャは、残りのマスター・サイトで、トリガーとそれらに対応付けられたパッケージを再生成します。このプロシージャは、マスター定義サイトからコールする必要があります。

構文

```
DBMS_REPCAT.REMOVE_MASTER_DATABASES (  
  gname          IN   VARCHAR2,  
  master_list     IN   VARCHAR2 |  
  master_table    IN   DBMS_UTILITY.DBLINK_ARRAY);
```

**注意：** このプロシージャはオーバーロードされています。master\_list パラメータと master\_table パラメータは、同時には指定できません。

## パラメータ

表 20-114 REMOVE\_MASTER\_DATABASES プロシージャのパラメータ

パラメータ	説明
gname	レプリケーション環境に対応付けられたレプリケーション・グループの名前です。1 つのマスター・データベースが複数のレプリケーション環境で使用されている場合、このパラメータによって混乱が避けられます。
master_list	レプリケーション環境から削除する完全修飾されたマスター・データベース名の、カンマ区切りのリストです。リスト内の名前の間にはスペースを挿入しません。
master_table	リストのかわりに、DBMS_UTILITY.DBLINK_ARRAY タイプの PL/SQL 索引付き表でデータベース名を指定できます。

## 例外

表 20-115 REMOVE\_MASTER\_DATABASES プロシージャの例外

例外	説明
nonmasterdef	起動サイトはマスター定義サイトではありません。
nonmaster	指定されたデータベースのうち少なくとも 1 つのデータベースが、マスター・サイトではありません。
reconfigerror	指定されたデータベースの 1 つが、マスター定義サイトです。
commfailure	残りのマスター・サイトのうち、少なくとも 1 つのマスター・サイトにアクセスできません。

## RENAME\_SHADOW\_COLUMN\_GROUP プロシージャ

このプロシージャは、レプリケート表のシャドウ列グループの名前を変更して、名前付きの列グループにします。このプロシージャを実行するためにレプリケート表のマスター・グループを静止させる必要はありません。

### 構文

```
DBMS_REPCAT.RENAME_SHADOW_COLUMN_GROUP (  
    sname                IN VARCHAR2,  
    oname                IN VARCHAR2,  
    new_col_group_name   IN VARCHAR2)
```

### パラメータ

表 20-116 RENAME\_SHADOW\_COLUMN\_GROUP プロシージャのパラメータ

パラメータ	説明
sname	レプリケート表が置かれたスキーマの名前です。
oname	レプリケート表の名前です。
new_col_group_name	新しい列グループの名前です。現在シャドウ・グループにある列は、指定した名前の列グループに配置されます。

### 例外

表 20-117 RENAME\_SHADOW\_COLUMN\_GROUP プロシージャの例外

例外	説明
missmview	指定されたスキーマは存在しません。
nonmasterdef	起動サイトはマスター定義サイトではありません。
missingobject	指定されたオブジェクトは存在しません。
duplicategroup	作成を指定された列グループはすでに存在します。



# REPCAT\_IMPORT\_CHECK プロシージャ

このプロシージャを実行すると、レプリケート・オブジェクト、または Advanced Replication で使用されたオブジェクトのエクスポートまたはインポート後に、マスター・グループのオブジェクトが適切なオブジェクト識別子およびステータス値を持つようになります。

## 構文

```
DBMS_REPCAT.REPCAT_IMPORT_CHECK (
    gname      IN   VARCHAR2,
    master     IN   BOOLEAN,
    gowner     IN   VARCHAR2  := 'PUBLIC');
```

## パラメータ

表 20-118 REPCAT\_IMPORT\_CHECK プロシージャのパラメータ

パラメータ	説明
gname	マスター・グループの名前です。両方のパラメータを省略した場合、カレント・サイトで、すべてのマスター・グループがチェックされます。
master	マスター・サイトをチェックする場合はこれを TRUE に、マテリアライズド・ビュー・サイトをチェックする場合は FALSE に設定します。
gowner	マスター・グループの所有者です。

## 例外

表 20-119 REPCAT\_IMPORT\_CHECK プロシージャの例外

例外	説明
nonmaster	master は TRUE に設定されています。データベースはレプリケーション・グループのマスター・サイトでないか、または望まれるデータベースではありません。
nonmview	master は FALSE に設定されています。データベースはレプリケーション・グループのマテリアライズド・ビュー・サイトではありません。
missingobject	レプリケーション・グループ内の有効なレプリケート・オブジェクトが存在しません。
missingrepgroup	指定されたレプリケート・レプリケーション・グループは存在しません。
missingschema	指定したスキーマが存在しません。

## RESUME\_MASTER\_ACTIVITY プロシージャ

このプロシージャは、レプリケーション環境を静止させた後で、通常のレプリケーション・アクティビティを再開します。

### 構文

```
DBMS_REPCAT.RESUME_MASTER_ACTIVITY (  
  gname          IN  VARCHAR2,  
  override       IN  BOOLEAN := false);
```

### パラメータ

表 20-120 RESUME\_MASTER\_ACTIVITY プロシージャのパラメータ

パラメータ	説明
gname	マスター・グループの名前です。
override	値が TRUE の場合は、保留になっている RepCat 管理要求が無視され、各マスターで通常のレプリケーション・アクティビティがリストアされます。これは緊急の場合のみ使用します。  値が FALSE の場合は、各マスターで通常のレプリケーション・アクティビティがリストアされます。ただし、そのマスターで gname に対する RepCat 管理要求が保留にされていない場合に限られます。

### 例外

表 20-121 RESUME\_MASTER\_ACTIVITY プロシージャの例外

例外	説明
nonmasterdef	起動サイトはマスター定義サイトではありません。
notquiesced	マスター・グループが静止中でないか、または静止していません。
commfailure	少なくとも 1 つのマスター・サイトにアクセスできません。
notallgenerated	レプリケーション・アクティビティを再開する前にレプリケーション・サポートが生成されました。

## RESUME\_PROPAGATION\_TO\_MDEF プロシージャ

新しいマスター・サイトを静止させずにマスター・グループに追加するプロセスにおいて、このプロシージャは、エクスポートが終了し、マスター・サイトに存在する拡張および影響を受けないレプリケーション・グループのマスター定義サイトへの伝播が可能になったことを示します。このプロシージャは、新しいマスター・サイトをマスター・グループに追加するために必要なエクスポートが完了した後の実行します。

**関連項目：** マスター・グループへのマスター・サイトの追加の詳細は、7-4 ページの「[新しいマスター・サイトの追加](#)」を参照してください。

### 構文

```
DBMS_REPCAT.RESUME_PROPAGATION_TO_MDEF (  
    extension_id          IN      RAW);
```

### パラメータ

表 20-122 RESUME\_PROPAGATION\_TO\_MDEF プロシージャのパラメータ

パラメータ	説明
extension_id	静止せずにマスター・グループにマスター・データベースを追加するための、現在保留になっている要求の識別子です。extension_id は、DBA_REPSITES_NEW データ・ディクショナリ・ビューおよび DBA_REPEXTENSIONS データ・ディクショナリ・ビューを問い合わせで検索できます。

### 例外

表 20-123 RESUME\_PROPAGATION\_TO\_MDEF プロシージャの例外

例外	説明
nonmasterdef	起動サイトはマスター定義サイトではありません。
extstinapp	拡張ステータスが不適切です。このプロシージャを実行する際、拡張ステータスは EXPORTING である必要があります。拡張ステータスをチェックするには、DBA_REPEXTENSIONS のデータ・ディクショナリ・ビューを問い合わせます。
dbnotcompatible	機能がデータベースのバージョンと互換性がありません。データベースの互換性のレベルは、すべて 9.0.1 以上である必要があります。

## SEND\_OLD\_VALUES プロシージャ

レプリケート表の各非キー列についての遅延トランザクションの伝播中に、表中の行が更新または削除される際、古い列値を送るというオプションがあります。min\_communication が TRUE に設定されている場合、デフォルトでは次のようになります。

- 削除された行については、すべての列の古い値を送ります。
- 更新された行については、キー列の古い値および列グループ内の修正された列を送ります。

マスター定義サイトで DBMS\_REPCAT.SEND\_OLD\_VALUES を呼び出すことにより、すべてのマスター・サイトおよびマテリアライズド・ビュー・サイトで、この動作を変更できます。次にすべてのマスター・サイトおよび各マテリアライズド・ビュー・サイトで、レプリケーション・サポートを生成します。

ユーザー定義型を使用する場合は、列オブジェクトのリーフ属性や、または列オブジェクト全体を指定できます。たとえば、cust\_address という列オブジェクトに street\_address という属性がある場合、column\_list パラメータに、または column\_table パラメータの一部として、cust\_address.street\_address を指定できます。または cust\_address のみを指定することもできます。

## 構文

```
DBMS_REPCAT.SEND_OLD_VALUES (
    sname          IN  VARCHAR2,
    oname          IN  VARCHAR2,
    { column_list  IN  VARCHAR2,
      | column_table IN  DBMS_UTILITY.VARCHAR2s | DBMS_UTILITY.LNAME_ARRAY, }
    operation      IN  VARCHAR2 := 'UPDATE',
    send           IN  BOOLEAN  := true );
```

---

---

**注意：** このプロシージャはオーバーロードされています。column\_list パラメータと column\_table パラメータは、同時には指定できません。

---

---

パラメータ

表 20-124 SEND\_OLD\_VALUES プロシージャのパラメータ

パラメータ	説明
sname	表が置かれたスキーマの名前です。
oname	レプリケート表の名前です。表はネストした表の記憶表を利用できます。
column_list	表の列のカンマ区切りのリストです。各エントリの間にはスペースを挿入しません。
column_table	リストのかわりに、DBMS_REPCAT.VARCHAR2S または DBMS_UTILITY.LNAME_ARRAY の PL/SQL 索引付き表を使用して列名を指定できます。最初の列名は 1 の位置、2 番目の列名は 2 の位置と、順に配置します。  30 バイト以上の列名がある場合は、DBMS_UTILITY.LNAME_ARRAY を使用します。これは列オブジェクトの属性を指定した際に起こる場合があります。
operation	可能な値は、update、delete、または更新と削除を意味するアスタリスク・ワイルドカード '*' です。
send	値が TRUE の場合は、指定された列の古い値が送られます。値が FALSE の場合は、指定された列の古い値が送られません。指定されていない列や操作には影響がありません。  表の min_communication が TRUE になると、マスター定義サイトで、指定された変更が有効になります。min_communication TRUE で、マスター・サイトやマテリアライズド・ビュー・サイトで次にレプリケーション・サポートが生成されると、変更が有効になります。

**注意：** operation パラメータにより、行が削除または更新された際に非キー列の古い値を送信するかどうかを指定できます。古い値を渡さない場合、古い値のかわりに NULL が渡されます。その場合、古い値は更新または削除を行った時点でのターゲット・サイトの列のカレント値と見なされます。

Oracle のデフォルトの動作を変更する前に、削減されたデータ伝播の詳細を、SEND\_OLD\_VALUES プロシージャを使用して『Oracle9i アドバンスド・レプリケーション』で参照してください。

例外

表 20-125 SEND\_OLD\_VALUES プロシージャの例外

例外	説明
nonmasterdef	起動サイトはマスター定義サイトではありません。
missingobject	指定されたオブジェクトは、行レベル・レプリケーションの情報が必要な指定されたスキーマ内の表として存在しません。
missingcolumn	少なくとも 1 つの列が、表に存在しません。
notquiesced	マスター・グループは静止していません。
typefailure	無効な操作が指定されました。
keysendcomp	指定された列は、表中のキー列です。
dbnotcompatible	機能がデータベースのバージョンと互換性がありません。この例外は一般的に、列オブジェクトの属性を送ろうとした際に発生します。この場合、データベースの互換性のレベルは、すべて 9.0.1 以上である必要があります。

SET\_COLUMNS プロシージャ

このプロシージャを使用すると、主キーのかわりに代替列または列グループを使用して、行レベル・レプリケーションを使用する際に比較する表中の列を決定できます。このプロシージャは、マスター定義サイトからコールする必要があります。

列オブジェクトを使用する際、列オブジェクトの属性が主キーまたは主キーの一部として使用できる場合は、その属性を代替キー列の一部にできます。たとえば、`cust_address` という列オブジェクトに `street_address` という `VARCHAR2` 属性がある場合、`column_list` パラメータに、または `column_table` パラメータの一部として、`cust_address.street_address` を指定できます。ただし、列オブジェクト `cust_address` 全体は指定できません

ネストした表の列の記憶表については、このプロシージャは `NESTED_TABLE_ID` を代替キー列として受け付けます。

オブジェクト表を使用する際は、代替キー列は指定できません。オブジェクト表の `OID` がシステム生成の場合は、オブジェクト表の `OID` 列がオブジェクト表のキーとして使用されます。オブジェクト表の `OID` がユーザー定義の場合は、オブジェクト表の主キーがキーとして使用されます。

次のタイプの列は、代替キー列として指定できません。

- 列オブジェクトの LOB または LOB 属性
- 列オブジェクトのコレクションまたはコレクション属性
- REF
- 列オブジェクト全体

**関連項目：** 主キー列の制限事項の詳細は、『Oracle9i SQL リファレンス』の「constraint\_clause」を参照してください。

構文

```
DBMS_REPCAT.SET_COLUMNS (  
    sname          IN    VARCHAR2,  
    oname          IN    VARCHAR2,  
    { column_list  IN    VARCHAR2  
    | column_table IN    DBMS_UTILITY.NAME_ARRAY | DBMS_UTILITY.LNAME_ARRAY } );
```

**注意：** このプロシージャはオーバーロードされています。column\_list パラメータと column\_table パラメータは、同時には指定できません。

パラメータ

表 20-126 SET\_COLUMNS プロシージャのパラメータ

パラメータ	説明
sname	表が置かれたスキーマの名前です。
oname	表の名前です。
column_list	主キーとして使用する表内の列の、カンマ区切りのリストです。各エントリの間にはスペースを挿入しません。
column_table	リストのかわりに、DBMS_UTILITY.NAME_ARRAY または DBMS_UTILITY.LNAME_ARRAY の PL/SQL 索引付き表を使用して列名を指定できます。最初の列名は 1 の位置、2 番目の列名は 2 の位置と、順に配置します。  30 バイト以上の列名がある場合は、DBMS_UTILITY.LNAME_ARRAY を使用します。これは列オブジェクトの属性を指定した際に起こる場合があります。

例外

表 20-127 SET\_COLUMNS プロシージャの例外

例外	説明
nonmasterdef	起動サイトはマスター定義サイトではありません。
missingobject	指定されたオブジェクトは、行レベル・レプリケーションの情報が必要な指定されたスキーマ内の表として存在しません。
missingcolumn	少なくとも 1 つの列が、表に存在しません。
notquiesced	レプリケーション・グループが静止中でないか、または静止していません。

SPECIFY\_NEW\_MASTERS プロシージャ

このプロシージャは、既存のレプリケーション・グループに追加したいマスター・サイトを、グループを静止させずに指定します。このプロシージャは、指定されたマスター・グループのマスター定義サイトで実行する必要があります。

必要な場合は、このプロシージャはマスター・グループに新しいマスター・サイトを追加するプロセスを追跡する extension\_id を作成します。この extension\_id は、プロセスの様々な段階で実行する他のプロシージャで使用します。extension\_id の情報は、DBA\_REPSITES\_NEW データ・ディクショナリ・ビューおよび DBA\_REPEXTENSIONS データ・ディクショナリ・ビューで表示できます。

このプロシージャは、指定されたレプリケーション・グループの DBA\_REPSITES\_NEW データ・ディクショナリ・ビューに新しいマスター・サイトを追加します。このプロシージャは指定されたレプリケーション・グループに何度でも実行できます。2 度以上実行する場合は、指定されたレプリケーション・グループのローカルな DBA\_REPSITES\_NEW データ・ディクショナリ・ビューのマスターは、master\_list パラメータまたは master\_table パラメータで指定されたマスターに置換されます。

このプロシージャは、ADD\_NEW\_MASTERS プロシージャを実行する前に実行する必要があります。ADD\_NEW\_MASTERS プロシージャを実行するまで、マスター・グループに新しいマスター・サイトは追加されません。

関連項目：

- 20-10 ページ [「ADD\\_NEW\\_MASTERS プロシージャ」](#)
- マスター・グループへのマスター・サイトの追加の詳細は、7-4 ページの [「新しいマスター・サイトの追加」](#) を参照してください。



構文

```
DBMS_REPCAT.SPECIFY_NEW_MASTERS (  
  gname          IN    VARCHAR2,  
  { master_list  IN    VARCHAR2  
  | master_table IN    DBMS_UTILITY.DBLINK_ARRAY});
```

**注意：** このプロシージャはオーバーロードされています。master\_list  
パラメータと master\_table パラメータは、同時には指定できません。

パラメータ

表 20-128 SPECIFY\_NEW\_MASTERS プロシージャのパラメータ

パラメータ	説明
gname	新しいマスター・サイトを追加するマスター・グループの名前です。
master_list	マスター・グループに追加する新しいマスター・サイトの、カンマ区切りのリストです。既存のマスター・サイトは含めず、新しいマスター・サイトのみをリストします。各サイト名の間に空白は入れないでください。  master_list が NULL の場合、指定されたレプリケーション・グループのマスター・サイトはすべて DBA_REPSITES_NEW データ・ディクショナリ・ビューから削除されます。マスター・グループが拡張されていないことを示すには、NULL を指定します。
master_table	マスター・グループに追加する新しいマスター・サイトをリストした表です。表には既存のマスター・サイトは含めず、新しいマスター・サイトのみをリストします。最初のマスター・サイトは 1 の位置、2 番目のマスターは 2 の位置と、順に配置します。  表が空の場合は、指定されたレプリケーション・グループのマスター・サイトはすべて DBA_REPSITES_NEW データ・ディクショナリ・ビューから削除されます。マスター・グループが拡張されていないことを示すには、空の表を使用します。

例外

表 20-129 SPECIFY\_NEW\_MASTERS プロシージャの例外

例外	説明
duplicaterepgroup	追加しようとしているマスター・サイトは、すでにマスター・グループの一部となっています。
nonmasterdef	起動サイトはマスター定義サイトではありません。
propmodenotallowed	同期伝播モードはこの操作では使用できません。使用できるのは非同期伝播モードのみです。
extstinapp	ステータス付きの拡張要求は使用できません。マスター・グループに extension_id はないか、または extension_id ステータスは READY である必要があります。各 extension_id のステータスは、DBA_REPEXTENSIONS データ・ディクショナリ・ビューのマスター・サイトで表示できます。
dbnotcompatible	機能がデータベースのバージョンと互換性がありません。データベースの互換性のレベルは、すべて 9.0.1 以上である必要があります。
notsamecq	マスター・グループが同じ接続修飾子を持っていません。

SUSPEND\_MASTER\_ACTIVITY プロシージャ

このプロシージャは、マスター・グループのレプリケーション・アクティビティを中断します。このプロシージャを使用して、マスター・グループを静止させます。このプロシージャは、マスター定義サイトからコールする必要があります。

構文

```
DBMS_REPCAT.SUSPEND_MASTER_ACTIVITY (  
  gname    IN   VARCHAR2);
```

# パラメータ

表 20-130 SUSPEND\_MASTER\_ACTIVITY プロシージャのパラメータ

パラメータ	説明
gname	アクティビティを中断するマスター・グループの名前です。

# 例外

表 20-131 SUSPEND\_MASTER\_ACTIVITY プロシージャの例外

例外	説明
nonmasterdef	起動サイトはマスター定義サイトではありません。
notnormal	マスター・グループで通常の操作が行われていません。
commfailure	少なくとも 1 つのマスター・サイトにアクセスできません。

# SWITCH\_MVIEW\_MASTER プロシージャ

このプロシージャは、マテリアライズド・ビュー・グループのマスター・サイトを他のマスター・サイトに変更します。このプロシージャは、影響を受けるマテリアライズド・ビューの完全なリフレッシュを行い、必要に応じてトリガーおよびその対応付けられたパッケージを再生成します。このプロシージャでは、マスター・サイトの変更に、キューが古いマスター・サイトにプッシュされることはありません。

マテリアライズド・ビューの min\_communication が TRUE に設定されており、新しいマスター・サイトが Oracle7 のマスター・サイトである場合は、min\_communication を FALSE に設定したマテリアライズド・ビューのレプリケーション・サポートを再生成します。

マテリアライズド・ビューの generate\_80\_compatible が FALSE に設定されており、新しいマスター・サイトが Oracle8i より前のリリース (Oracle7 または Oracle8) である場合は、generate\_80\_compatible を TRUE に設定したマテリアライズド・ビューのレプリケーション・サポートを再生成します。

DBMS\_REPCAT.GENERATE\_MVIEW\_SUPPORT を 1 度コールすれば、マテリアライズド・ビューに両方のパラメータを設定できます。

**注意：** 他のマテリアライズド・ビュー (レベル 2 以上) に基づくマテリアライズド・ビューのマスターを切替えることはできません。このようなマテリアライズド・ビューのベースを別のマスターにするには、マテリアライズド・ビューを削除して再作成する必要があります。

関連項目： [20-80 ページ「GENERATE\\_MVIEW\\_SUPPORT プロシージャ」](#)

構文

```
DBMS_REPCAT.SWITCH_MVIEW_MASTER (  
  gname          IN   VARCHAR2,  
  master         IN   VARCHAR2,  
  gowner         IN   VARCHAR2  := 'PUBLIC');
```

パラメータ

表 20-132 SWITCH\_MVIEW\_MASTER プロシージャのパラメータ

パラメータ	説明
gname	マスター・サイトを変更するマテリアライズド・ビュー・グループの名前です。
master	マテリアライズド・ビュー・グループに使用する新しいマスター・サイトの、完全修飾されたデータベース名です。
gowner	マテリアライズド・ビュー・グループの所有者です。

例外

表 20-133 SWITCH\_MVIEW\_MASTER プロシージャの例外

例外	説明
nonmview	起動サイトはマテリアライズド・ビュー・サイトではありません。
nonmaster	データベースはマスター・サイトではありません。
commfailure	指定されたデータベースにはアクセスできません。
missingrepgroup	マテリアライズド・ビュー・グループは存在しません。
qrytoolong	マテリアライズド・ビュー定義問合せが 32KB を超えています。
alreadymastered	ローカル・サイトに、同じグループ名で、古いマスター・サイトをマスターに持つ別のマテリアライズド・ビュー・グループがあります。

## UNDO\_ADD\_NEW\_MASTERS\_REQUEST プロシージャ

このプロシージャは、指定された `extension_id` について、`SPECIFY_NEW_MASTERS` プロシージャおよび `ADD_NEW_MASTERS` プロシージャによるすべての変更を元に戻します。

このプロシージャは1つのマスター・サイト（マスター定義サイトの場合もある）で実行され、そのマスター・サイトのみに影響を与えます。このプロシージャを、要求の影響を受けた1つのマスター・サイトで実行するには、その要求の影響を受けた、新規および既存のマスター・サイトすべてで実行する必要があります。 `DBA_REPSITES_NEW` データ・ディクショナリ・ビューを問い合わせれば、`extension_id` の影響を受けた新しいマスター・サイトを調べることができます。このデータ・ディクショナリ・ビューには、レプリケーション・グループ名もリストされています。このプロシージャはレプリケーション・グループ内の既存のマスター・サイトすべてで実行する必要があります。

---

**注意：** このプロシージャは、通常はコールしません。このプロシージャは、1つ以上のマスター・サイトで、新しいマスターを静止なしで追加する操作が行えない場合にのみ使用します。このプロシージャは `SPECIFY_NEW_MASTERS` および `ADD_NEW_MASTERS` の各プロシージャを実行した後に、また `RESUME_PROPAGATION_TO_MDEF` および `PREPARE_INSTANTIATED_MASTER` の各プロシージャを実行する前に実行します。

特定の `extension_id` に `RESUME_PROPAGATION_TO_MDEF` または `PREPARE_INSTANTIATED_MASTER` を実行した後に、このプロシージャを実行しないでください。

---

### 関連項目：

- 20-104 ページ [「SPECIFY\\_NEW\\_MASTERS プロシージャ」](#)
- 20-10 ページ [「ADD\\_NEW\\_MASTERS プロシージャ」](#)
- 20-99 ページ [「RESUME\\_PROPAGATION\\_TO\\_MDEF プロシージャ」](#)
- 20-85 ページ [「PREPARE\\_INSTANTIATED\\_MASTER プロシージャ」](#)

構文

```
DBMS_REPCAT.UNDO_ADD_NEW_MASTERS_REQUEST (  
    extension_id      IN RAW,  
    drop_contents     IN BOOLEAN := TRUE);
```

パラメータ

表 20-134 UNDO\_ADD\_NEW\_MASTERS\_REQUEST プロシージャのパラメータ

パラメータ	説明
extension_id	静止せずにマスター・グループにマスター・データベースを追加するための、現在保留になっている要求の識別子です。extension_id は、DBA_REPSITES_NEW データ・ディクショナリ・ビューおよび DBA_REPEXTENSIONS データ・ディクショナリ・ビューを問い合わせで検索できます。
drop_contents	ローカル・サイトで拡張されている新しいレプリケーション・グループ内のオブジェクトの目次を削除するには、デフォルトの TRUE を指定します。目次を保存するには、FALSE を指定します。

例外

表 20-135 UNDO\_ADD\_NEW\_MASTERS\_REQUEST プロシージャの例外

例外	説明
dbnotcompatible	機能がデータベースのバージョンと互換性がありません。データベースの互換性のレベルは、すべて 9.0.1 以上である必要があります。
typefail	指定されたパラメータ値は不適正です。

## UNREGISTER\_MVIEW\_REPGROUP プロシージャ

このプロシージャは DBA\_REGISTERED\_MVIEW\_GROUPS からのマテリアライズド・ビュー・グループの削除を行って、マスター・サイトまたはマスター・マテリアライズド・ビュー・サイトでのマテリアライズド・ビューの管理を容易にします。このプロシージャは、マスター・サイトまたはマスター・マテリアライズド・ビュー・サイトで実行します。

### 構文

```
DBMS_REPCAT.UNREGISTER_MVIEW_REPGROUP (
    gname      IN   VARCHAR2,
    mviewsite  IN   VARCHAR2,
    gowner     IN   VARCHAR2  := 'PUBLIC');
```

### パラメータ

**表 20-136 UNREGISTER\_MVIEW\_REPGROUP プロシージャのパラメータ**

パラメータ	説明
gname	登録解除するマテリアライズド・ビュー・グループの名前です。
mviewsite	マテリアライズド・ビュー・サイトのグローバル名です。
gowner	マテリアライズド・ビュー・グループの所有者です。

## VALIDATE ファンクション

このファンクションは、マルチマスター・レプリケーション環境のキー条件が正確であるかどうかを確認します。

### 構文

```
DBMS_REPCAT.VALIDATE (
    gname      IN   VARCHAR2,
    check_genflags IN BOOLEAN := false,
    check_valid_objs IN BOOLEAN := false,
    check_links_sched IN BOOLEAN := false,
    check_links IN BOOLEAN := false,
    error_table OUT DBMS_REPCAT.VALIDATE_ERR_TABLE)
RETURN BINARY_INTEGER;
```

```
DBMS_REPCAT.VALIDATE (
    gname      IN   VARCHAR2,
    check_genflags IN BOOLEAN := false,
    check_valid_objs IN BOOLEAN := false,
    check_links_sched IN BOOLEAN := false,
    check_links IN BOOLEAN := false,
```

```
error_msg_table      OUT DBMS_UTILITY.UNCL_ARRAY,  
error_num_table      OUT DBMS_UTILITY.NUMBER_ARRAY )  
RETURN BINARY_INTEGER;
```

**注意：** このファンクションはオーバーロードされています。VALIDATE の戻り値は、検出されたエラーの数です。このファンクションの OUT パラメータでは、検出されたエラーが戻されます。20-111 ページの「構文」に示される最初のインタフェース・ファンクションで、error\_table は、レコードの配列で構成されます。それぞれのレコードには、VARCHAR2 と NUMBER があります。文字列フィールドにはエラー・メッセージが、数値フィールドには Oracle エラー番号が入ります。

20-111 ページの「構文」に示される 2 番目のインタフェース・ファンクションも同様ですが、OUT 配列が 2 つあります。VARCHAR2 配列にはエラー・メッセージが、NUMBER 配列にはエラー番号が入ります。

パラメータ

表 20-137 VALIDATE ファンクションのパラメータ

パラメータ	説明
gname	確認するマスター・グループの名前です。
check_genflags	グループ内のすべてのオブジェクトが生成されたかどうかを確認されます。この処理は、マスター定義サイトでのみ行われます。
check_valid_objs	グループ内のオブジェクトの基になるオブジェクトが有効であるかどうかを確認されます。この処理は、マスター定義サイトでのみ行われます。マスター定義サイトでの処理が他のすべてのサイトに波及し、基になるオブジェクトが有効であるかどうかチェックされます。オブジェクトの有効性は、接続ユーザーのスキーマ内で確認されます。
check_links_sched	リンクがスケジュールされているかどうかを確認されます。この処理は各マスター・サイトで実行されます。
check_links	レプリケーションが正しく行われるように、接続ユーザー (repsadmin) およびプロパゲータに正しいリンクが設定されているかどうかを確認されます。リンクがデータベースに存在し、アクセス可能であるかどうかチェックされます。この処理は各マスター・サイトで実行されます。
error_table	検出されたエラーの総数およびメッセージが戻されます。
error_msg_table	検出されたすべてのエラーのメッセージが戻されます。
error_num_table	検出されたエラーの総数が戻されます。



例外

表 20-138 VALIDATE ファンクションの例外

例外	説明
missingdblink	データベース・リンクがレプリケーション・プロパゲータのスキーマに存在しないか、またはスケジュールされていません。データベース・リンクが確実にデータベースに存在し、アクセス可能で、その実行がスケジュールされるようにしてください。
dblinkmismatch	ローカル・ノードのデータベース・リンク名が、リンクがアクセスするデータベースのグローバル名と一致しません。GLOBAL_NAMES 初期化パラメータが TRUE に設定され、リンク名がグローバル名と一致するようにしてください。
dblinkuidmismatch	ローカル・ノードのレプリケーション管理ユーザーのユーザー名と、データベース・リンクに対応するノードのユーザー名が異なります。Advanced Replication では、これらのユーザーの名前が同じことを前提としています。ローカル・ノードのレプリケーション管理ユーザーのユーザー ID と、データベース・リンクに対応するノードのユーザー ID を同じにしてください。
objectnotgenerated	オブジェクトが他のマスター・サイトで生成されていないか、生成がまだ終了していません。マスター定義サイトのオブジェクトに対する GENERATE_REPLICATION_SUPPORT および DO_DEFERRED_REPCAT_ADMIN のコールで、確実にオブジェクトが生成されるようにしてください。
opnotsupported	レプリケーション・グループが Oracle8 より前のノードでレプリケートされる場合は、操作がサポートされません。マスター・グループのすべてのノードが Oracle8 以上を実行するようにしてください。

使用上の注意

VALIDATE の戻り値は、検出されたエラーの数です。このファンクションの OUT パラメータでは、検出されたエラーが戻されます。最初のインタフェース・ファンクション error\_table は、レコードの配列で構成されます。それぞれのレコードには、VARCHAR2 と NUMBER があります。文字列フィールドにはエラー・メッセージが、数値フィールドには Oracle エラー番号が入ります。

2 番目のインタフェースは最初のインタフェースに似ていますが、OUT 配列が 2 つあります。VARCHAR2 配列にはエラー・メッセージが、NUMBER 配列にはエラー番号が入ります。

## WAIT\_MASTER\_LOG プロシージャ

このプロシージャは、マスター・サイトに非同期で伝播された変更が適用されたかどうかを判別します。

### 構文

```
DBMS_REPCAT.WAIT_MASTER_LOG (  
    gname           IN    VARCHAR2,  
    record_count    IN    NATURAL,  
    timeout         IN    NATURAL,  
    true_count      OUT   NATURAL);
```

### パラメータ

表 20-139 WAIT\_MASTER\_LOG プロシージャのパラメータ

パラメータ	説明
gname	マスター・グループの名前です。
record_count	完了していないアクティビティの数がこのしきい値と同じか、このしきい値より少なくなると、プロシージャが終了します。
timeout	プロシージャが終了するまで待つ最大秒数です。
true_count (out parameter)	完了していないアクティビティの数が戻されます。

### 例外

表 20-140 WAIT\_MASTER\_LOG プロシージャの例外

例外	説明
nonmaster	起動サイトはマスター・サイトではありません。

---

## DBMS\_REPCAT\_ADMIN

DBMS\_REPCAT\_ADMIN は、対称型レプリケーション機能で必要となる権限を持つユーザーの作成を可能にします。

この章では、次の項目を説明します。

- [DBMS\\_REPCAT\\_ADMIN サブプログラムの概要](#)

## DBMS\_REPCAT\_ADMIN サブプログラムの概要

表 21-1 DBMS\_REPCAT\_ADMIN パッケージのサブプログラム

サブプログラム	説明
21-3 ページ「GRANT_ADMIN_ANY_SCHEMA プロシージャ」	レプリケーション管理者に、カレント・サイトでのレプリケーション・グループの管理に必要な権限を付与します。
21-4 ページ「GRANT_ADMIN_SCHEMA プロシージャ」	レプリケーション管理者に、カレント・サイトでのスキーマの管理に必要な権限を付与します。
21-4 ページ「REGISTER_USER_REPGROUP プロシージャ」	マスター・サイトまたはマスター・マテリアライズド・ビュー・サイトで、リモート・サイトに対するプロキシ・マテリアライズド・ビュー管理者または受信者権限を割り当てます。
21-6 ページ「REVOKE_ADMIN_ANY_SCHEMA プロシージャ」	GRANT_ADMIN_ANY_SCHEMA プロシージャでレプリケーション管理者に付与された権限やロールを取り消します。
21-7 ページ「REVOKE_ADMIN_SCHEMA プロシージャ」	GRANT_ADMIN_SCHEMA プロシージャでレプリケーション管理者に付与された権限やロールを取り消します。
21-8 ページ「UNREGISTER_USER_REPGROUP プロシージャ」	REGISTER_USER_REPGROUP プロシージャでプロキシ・マテリアライズド・ビュー管理者または受信者に付与された権限やロールを取り消します。

# GRANT\_ADMIN\_ANY\_SCHEMA プロシージャ

このプロシージャは、レプリケーション管理者に、カレント・サイトでのレプリケーション・グループの管理に必要な権限を付与します。

## 構文

```
DBMS_REPCAT_ADMIN.GRANT_ADMIN_ANY_SCHEMA (  
    username IN VARCHAR2);
```

## パラメータ

表 21-2 GRANT\_ADMIN\_ANY\_SCHEMA プロシージャのパラメータ

パラメータ	説明
username	カレント・サイトでのレプリケーション・グループの管理に必要な権限とロールを付与する、レプリケーション管理者の名前です。

## 例外

表 21-3 GRANT\_ADMIN\_ANY\_REPGROUP プロシージャの例外

例外	説明
ORA-01917	指定されたユーザーは存在しません。

## GRANT\_ADMIN\_SCHEMA プロシージャ

このプロシージャは、レプリケーション管理者に、カレント・サイトでのスキーマの管理に必要な権限を付与します。レプリケーション・グループが複数のスキーマにまたがらない場合は、このプロシージャを使用すると便利です。

### 構文

```
DBMS_REPCAT_ADMIN.GRANT_ADMIN_SCHEMA (  
    username IN VARCHAR2);
```

### パラメータ

表 21-4 GRANT\_ADMIN\_REPSchema プロシージャのパラメータ

パラメータ	説明
username	レプリケーション管理者の名前です。このユーザーには、カレント・サイトのレプリケーション・マスター・グループ内での、同じ名前のスキーマの管理に必要な権限とロールが付与されます。

### 例外

表 21-5 GRANT\_ADMIN\_REPSchema プロシージャの例外

例外	説明
ORA-01917	指定されたユーザーは存在しません。

## REGISTER\_USER\_REPGROUP プロシージャ

このプロシージャは、マスター・サイトまたはマスター・マテリアライズド・ビュー・サイトで、リモート・サイトに対するプロキシ・マテリアライズド・ビュー管理者または受信者権限を割り当てます。このプロシージャは、プロキシ・マテリアライズド・ビュー管理者または受信者に、必要な権限のみを付与します。GRANT\_ADMIN\_SCHEMA プロシージャや GRANT\_ADMIN\_ANY\_SCHEMA プロシージャで付与される強力な権限は、このプロシージャでは付与されません。

**関連項目：** トラストッド・セキュリティ・モデルとアントラストッド・セキュリティ・モデルの詳細は、[付録 A「セキュリティ・オプション」](#)を参照してください。

## 構文

```
DBMS_REPCAT_ADMIN.REGISTER_USER_REPGROUP (
  username          IN   VARCHAR2,
  privilege_type    IN   VARCHAR2,
  {list_of_gnames   IN   VARCHAR2 |
  table_of_gnames   IN   DBMS_UTILITY.NAME_ARRAY) };
```

---

**注意：** このプロシージャはオーバーロードされています。  
list\_of\_gnames パラメータと table\_of\_gnames パラメータは、同時  
には指定できません。

---

## パラメータ

**表 21-6 REGISTER\_USER\_REPGROUP プロシージャのパラメータ**

パラメータ	説明
username	プロキシ・マテリアライズド・ビュー管理者または受信者の権限を付与するユーザーの名前です。
privilege_type	割り当てる権限のタイプを指定します。privilege_type の定義には、次の値を使用します。 <ul style="list-style-type: none"> <li>■ 受信者の権限を付与する場合は、receiver を使用します。</li> <li>■ プロキシ・マテリアライズド・ビュー管理者の権限を付与する場合は、proxy_snapadmin を使用します。</li> </ul>
list_of_gnames	ユーザーに付与した受信者権限の適用対象にするレプリケーション・グループの、カンマ区切りのリストです。リスト内のエントリの間にはスペースを挿入しません。list_of_gnames を NULL に設定すると、このプロシージャのコール時にはまだ確認されていないレプリケーション・グループを含め、すべてのレプリケーション・グループが受信者権限の適用対象になります。list_of_gnames を NULL に設定するには、名前表記法を使用する必要があります。リスト内に無効なレプリケーション・グループがあると、リスト全体の登録が失敗します。
table_of_gnames	ユーザーに付与した受信者権限の適用対象にするレプリケーション・グループの、PL/SQL 索引付き表です。PL/SQL 索引付き表のタイプは DBMS_UTILITY.NAME_ARRAY にする必要があります。この表は 1 ベースです（1 の位置から始まり、1 ずつ増加します）。ユーザーをすべてのレプリケーション・グループに登録するには、NULL を使用します。表内に無効なレプリケーション・グループがあると、表全体の登録が失敗します。

例外

表 21-7 REGISTER\_USER\_REPGROUP プロシージャの例外

例外	説明
nonmaster	指定されたレプリケーション・グループが存在しないか、または起動データベースがマスター・サイトあるいはマスター・マテリアライズド・ビュー・サイトではありません。
ORA-01917	指定されたユーザーは存在しません。
typefailure	指定された権限のタイプは正しくありません。

REVOKE\_ADMIN\_ANY\_SCHEMA プロシージャ

このプロシージャは、GRANT\_ADMIN\_ANY\_SCHEMA でレプリケーション管理者に付与された権限やロールを取り消します。

**注意：** GRANT\_ADMIN\_ANY\_SCHEMA 以外で付与された同一の権限やロールも取り消されます。

構文

```
DBMS_REPCAT_ADMIN.REVOKE_ADMIN_ANY_SCHEMA (  
  username IN VARCHAR2);
```

パラメータ

表 21-8 REVOKE\_ADMIN\_ANY\_SCHEMA プロシージャのパラメータ

パラメータ	説明
username	権限を取り消すレプリケーション管理者の名前です。

例外

表 21-9 REVOKE\_ADMIN\_ANY\_SCHEMA プロシージャの例外

例外	説明
ORA-01917	指定されたユーザーは存在しません。



# REVOKE\_ADMIN\_SCHEMA プロシージャ

このプロシージャは、GRANT\_ADMIN\_SCHEMA でレプリケーション管理者に付与された権限やロールを取り消します。

**注意：** GRANT\_ADMIN\_SCHEMA 以外で付与された同一の権限やロールも取り消されます。

## 構文

```
DBMS_REPCAT_ADMIN.REVOKE_ADMIN_SCHEMA (  
    username IN VARCHAR2);
```

## パラメータ

表 21-10 REVOKE\_ADMIN\_SCHEMA プロシージャのパラメータ

パラメータ	説明
username	権限を取り消すレプリケーション管理者の名前です。

## 例外

表 21-11 REVOKE\_ADMIN\_SCHEMA プロシージャの例外

例外	説明
ORA-01917	指定されたユーザーは存在しません。

## UNREGISTER\_USER\_REPGROUP プロシージャ

このプロシージャは、REGISTER\_USER\_REPGROUP プロシージャでプロキシ・マテリアライズド・ビュー管理者または受信者に付与された権限やロールを取り消します。

### 構文

```
DBMS_REPCAT_ADMIN.UNREGISTER_USER_REPGROUP (  
    username          IN   VARCHAR2,  
    privilege_type    IN   VARCHAR2,  
    {list_of_gnames   IN   VARCHAR2 |  
     table_of_gnames  IN   DBMS_UTILITY.NAME_ARRAY});
```

**注意：** このプロシージャはオーバーロードされています。  
list\_of\_gnames パラメータと table\_of\_gnames パラメータは、同時  
には指定できません。

### パラメータ

表 21-12 UNREGISTER\_USER\_REPGROUP プロシージャのパラメータ

パラメータ	説明
username	登録を取り消すユーザーの名前です。
privilege_type	取り消す権限のタイプを指定します。privilege_type の定義には、次の値を使用します。 <ul style="list-style-type: none"><li>■ 受信者の権限を取り消す場合は、receiver を使用します。</li><li>■ プロキシ・マテリアライズド・ビュー管理者の権限を取り消す場合は、proxy_snapadmin を使用します。</li></ul>
list_of_gnames	ユーザーに付与しない受信者権限の適用対象にするレプリケーション・グループの、カンマ区切りのリストです。リスト内のエントリの間にはスペースを挿入しません。list_of_gnames を NULL に設定すると、すべてのレプリケーション・グループに対するユーザーの登録が取り消されます。list_of_gnames を NULL に設定するには、名前表記法を使用する必要があります。リスト内に無効なレプリケーション・グループがあると、リスト全体の登録解除が失敗します。
table_of_gnames	ユーザーに付与した受信者権限の非適用対象にするレプリケーション・グループの、PL/SQL 索引付き表です。PL/SQL 索引付き表のタイプは DBMS_UTILITY.NAME_ARRAY にする必要があります。この表は 1 ベースです（1 の位置から始まり、1 ずつ増加します）。すべてのレプリケーション・グループに対するユーザーの登録を取り消すには、NULL を使用します。表内に無効なレプリケーション・グループがあると、表全体の登録解除が失敗します。

## 例外

**表 21-13 UNREGISTER\_USER\_REPGROUP プロシージャの例外**

例外	説明
nonmaster	指定されたレプリケーション・グループが存在しないか、または起動データベースがマスター・サイトあるいはマスター・マテリアライズド・ビュー・サイトではありません。
ORA-01917	指定されたユーザーは存在しません。
typefailure	指定された権限のタイプは正しくありません。



---

# DBMS\_REPCAT\_INSTANTIATE

DBMS\_REPCAT\_INSTANTIATE パッケージは、配置テンプレートをインスタンス化します。  
この章では、次の項目を説明します。

- [DBMS\\_REPCAT\\_INSTANTIATE サブプログラムの概要](#)

## DBMS\_REPCAT\_INSTANTIATE サブプログラムの概要

表 22-1 DBMS\_REPCAT\_INSTANTIATE パッケージのサブプログラム

サブプログラム	説明
22-2 ページ「 <a href="#">DROP_SITE_INSTANTIATION</a> プロシージャ」	ターゲット・サイトを DBA_REPCAT_TEMPLATE_SITES ビューから削除するパブリック・プロシージャです。
22-3 ページ「 <a href="#">INSTANTIATE_OFFLINE</a> ファンクション」	オフラインでリモート・マテリアライズド・ビュー・サイトにマテリアライズド・ビュー環境を構築する際に使用するスクリプトを、マスター・サイトに生成するパブリック・ファンクションです。
22-5 ページ「 <a href="#">INSTANTIATE_ONLINE</a> ファンクション」	オンラインでリモート・マテリアライズド・ビュー・サイトにマテリアライズド・ビュー環境を構築する際に使用するスクリプトを、マスター・サイトに生成するパブリック・ファンクションです。

### DROP\_SITE\_INSTANTIATION プロシージャ

このプロシージャは、ターゲット・サイトでテンプレートのインスタンス化を削除します。このプロシージャは、マスター・サイトで関連するすべてのメタデータを削除します。また、指定されたサイトでのマテリアライズド・ビューのリフレッシュを使用禁止にします。このプロシージャは、最初にテンプレートをインスタンス化したユーザーとして実行する必要があります。テンプレートをインスタンス化したユーザーを確認するには、[ALL\\_REPCAT\\_TEMPLATE\\_SITES](#) ビューで問合せを行います。

### 構文

```
DBMS_REPCAT_INSTANTIATE.DROP_SITE_INSTANTIATION(  
    refresh_template_name IN VARCHAR2,  
    site_name              IN   VARCHAR2);
```

表 22-2 DROP\_SITE\_INSTANTIATION プロシージャのパラメータ

パラメータ	説明
refresh_template_name	削除対象の配置テンプレートの名前です。
site_name	指定されたテンプレートのインスタンス化を削除するマスター・サイトの位置です。

## INSTANTIATE\_OFFLINE ファンクション

このファンクションは、オフラインでリモート・マテリアライズド・ビュー・サイトにマテリアライズド・ビュー環境を構築する際に使用するファイルを、マスター・サイトに生成します。生成されたファイルは、オフライン・インスタンスエーション・ファイルで、マスター・サイトに長時間接続できないリモート・マテリアライズド・ビュー・サイトで使用します。

リモート・マテリアライズド・ビュー・サイトがラップトップの場合、このプロシーダを使用すると便利です。レプリケーション・マネージメント・ツールのパッケージ・インタフェースを使用して、生成されたファイルやデータを1つのファイルにパッケージ化すると、FTP サイトに転送したり、CD-ROM やフロッピー・ディスクなどにロードできます。

このファンクションで生成されたスクリプトは、USER\_REPCAT\_TEMP\_OUTPUT 一時ビューに格納され、配置テンプレートの配布時に、レプリケーション管理ツールなどの Oracle のツール製品で使用します。このファンクションで戻される数値は、USER\_REPCAT\_TEMP\_OUTPUT ビューから適切な情報を取り出す際に使用します。

このパブリック・ファンクションを実行するユーザーは、指定されたサイトのインスタンス化されたテンプレートの登録ユーザーになります。

---

---

**注意：** このファンクションは、配置テンプレートのオフライン・インスタンスエーションに使用します。

このファンクションを、[DBMS\\_OFFLINE\\_OG](#) パッケージのプロシーダ（マスター表のオフライン・インスタンスエーションに使用）や、[DBMS\\_OFFLINE\\_SNAPSHOT](#) パッケージのプロシーダ（マテリアライズド・ビューのオフライン・インスタンスエーションに使用）と混同しないでください。これらのプロシーダの使用の詳細は、該当するパッケージの説明を参照してください。

---

---

### 関連項目：

- [4-13 ページ「配置テンプレートをパッケージ化してインスタンス化」](#)
- 『Oracle9i アドバンスド・レプリケーション』
- レプリケーション管理ツールのオンライン・ヘルプ

構文

```
DBMS_REPCAT_INSTANTIATE.INSTATIATE_OFFLINE(  
    refresh_template_name    IN    VARCHAR2,  
    site_name                IN    VARCHAR2,  
    runtime_parm_id          IN    NUMBER      := -1e-130,  
    next_date                IN    DATE        := SYSDATE,  
    interval                 IN    VARCHAR2    := 'SYSDATE + 1',  
    use_default_gowner       IN    BOOLEAN    := true)  
return NUMBER;
```

表 22-3 INSTATIATE\_OFFLINE ファンクションのパラメータ

パラメータ	説明
refresh_template_name	インスタンス化される配置テンプレートの名前です。
site_name	配置テンプレートをインスタンス化しているリモート・サイトの名前です。
runtime_parm_id	INSERT_RUNTIME_PARMS プロシージャを使用してランタイム・パラメータ値を定義した場合は、ランタイム・パラメータの作成時に使用した ID を指定します (ID は、GET_RUNTIME_PARM_ID ファンクションで取得済です)。
next_date	リフレッシュ・グループの作成時に使用する、次のリフレッシュ日の値です。
interval	リフレッシュ・グループの作成時に使用する、リフレッシュ間隔です。
use_default_gowner	値が TRUE の場合、作成されたマテリアライズド・ビュー・グループはデフォルトのユーザー PUBLIC が所有します。値が FALSE の場合、作成されたマテリアライズド・ビュー・グループはインスタンス化を実行するユーザーが所有します。

例外

表 22-4 INSTATIATE\_OFFLINE ファンクションの例外

例外	説明
miss_refresh_template	指定された配置テンプレート名が無効か、または存在しません。
dupl_template_site	配置テンプレートは、このマテリアライズド・ビュー・サイトでインスタンス化されています。配置テンプレートは、各マテリアライズド・ビュー・サイトにつき 1 回のみインスタンス化できます。
not_authorized	このユーザーは、配置テンプレートをインスタンス化する許可を受けていません。



戻り値

表 22-5 INSTANTIATE\_OFFLINE ファンクションの戻り値

戻り値	説明
< システム生成番号 >	USER_REPCAT_TEMP_OUTPUT ビューから選択して、生成されたインスタンス化スクリプトを取得するときの、output_id に対するシステム生成番号を示します。

INSTANTIATE\_ONLINE ファンクション

このファンクションは、オンラインでリモート・マテリアライズド・ビュー・サイトにマテリアライズド・ビュー環境を構築する際に使用するスクリプトを、マスター・サイトに生成します。生成されたスクリプトは、マスター・サイトに長時間接続できるリモート・マテリアライズド・ビュー・サイトで使用します。新しいマテリアライズド・ビューに設定されるデータの量によっては、リモート・マテリアライズド・ビュー・サイトでのインスタンス化処理に時間がかかります。

このファンクションで生成されたスクリプトは、USER\_REPCAT\_TEMP\_OUTPUT 一時ビューに格納され、配置テンプレートの配布時に、レプリケーション管理ツールなどの Oracle のツール製品で使います。このファンクションで戻される数値は、USER\_REPCAT\_TEMP\_OUTPUT ビューから適切な情報を取り出す際に使います。

このパブリック・ファンクションを実行するユーザーは、指定されたサイトのインスタンス化されたテンプレートの登録ユーザーになります。

関連項目：

- 4-13 ページ [「配置テンプレートをパッケージ化してインスタンス化」](#)
- 『Oracle9i アドバンスド・レプリケーション』
- レプリケーション管理ツールのオンライン・ヘルプ

構文

```
DBMS_REPCAT_INSTANTIATE.INSTATIATE_ONLINE(  
  refresh_template_name  IN   VARCHAR2,  
  site_name              IN   VARCHAR2,  
  runtime_parm_id        IN   NUMBER      := -1e-130,  
  next_date              IN   DATE        := SYSDATE,  
  interval               IN   VARCHAR2    := 'SYSDATE + 1',  
  use_default_gowner     IN   BOOLEAN     := true)  
return NUMBER;
```

表 22-6 INSTATIATE\_ONLINE ファンクションのパラメータ

パラメータ	説明
refresh_template_name	インスタンス化される配置テンプレートの名前です。
site_name	配置テンプレートをインスタンス化しているリモート・サイトの名前です。
runtime_parm_id	INSERT_RUNTIME_PARMS プロシージャを使用してランタイム・パラメータ値を定義した場合は、ランタイム・パラメータの作成時に使用した ID を指定します (ID は、GET_RUNTIME_PARM_ID ファンクションで取得済です)。
next_date	リフレッシュ・グループの作成時に使用する、次のリフレッシュ日の値を指定します。
interval	リフレッシュ・グループの作成時に使用する、リフレッシュ間隔を指定します。
use_default_gowner	値が TRUE の場合、作成されたマテリアライズド・ビュー・グループはデフォルトのユーザー PUBLIC が所有します。値が FALSE の場合、作成されたマテリアライズド・ビュー・グループはインスタンス化を実行するユーザーが所有します。

例外

表 22-7 INSTANTIATE\_ONLINE ファンクションの例外

例外	説明
miss_refresh_template	指定された配置テンプレート名が無効か、または存在しません。
dupl_template_site	配置テンプレートは、このマテリアライズド・ビュー・サイトでインスタンス化されています。配置テンプレートは、各マテリアライズド・ビュー・サイトにつき 1 回のみインスタンス化できます。
not_authorized	このユーザーは、配置テンプレートをインスタンス化する許可を受けていません。

戻り値

表 22-8 INSTANTIATE\_ONLINE ファンクションの戻り値

戻り値	説明
< システム生成番号 >	USER_REPCAT_TEMP_OUTPUT ビューから選択して、生成されたインスタンス化スクリプトを取得するときの、output_id に対するシステム生成番号を示します。



---

## DBMS\_REPCAT\_RGT

DBMS\_REPCAT\_RGT は、リフレッシュ・グループのテンプレートの管理および定義を制御します。

この章では、次の項目を説明します。

- [DBMS\\_REPCAT\\_RGT サブプログラムの概要](#)

## DBMS\_REPCAT\_RGT サブプログラムの概要

表 23-1 DBMS\_REPCAT\_RGT パッケージのサブプログラム

サブプログラム	説明
23-4 ページ「ALTER_REFRESH_TEMPLATE プロシージャ」	データベース管理者が既存の配置テンプレートを変更します。
23-6 ページ「ALTER_TEMPLATE_OBJECT プロシージャ」	指定された配置テンプレートに追加されたオブジェクトを変更します。
23-8 ページ「ALTER_TEMPLATE_PARM プロシージャ」	データベース管理者が指定された配置テンプレートのパラメータを変更します。
23-10 ページ「ALTER_USER_AUTHORIZATION プロシージャ」	DBA_REPCAT_USER_AUTHORIZATIONS ビューの内容を変更します。
23-12 ページ「ALTER_USER_PARM_VALUE プロシージャ」	特定のユーザーに対して定義されている既存のパラメータ値を変更します。
23-14 ページ「COMPARE_TEMPLATES ファンクション」	データベース管理者が 2 つの配置テンプレートの内容を比較します。
23-15 ページ「COPY_TEMPLATE ファンクション」	データベース管理者が配置テンプレートをコピーします。
23-17 ページ「CREATE_OBJECT_FROM_EXISTING ファンクション」	既存のデータベース・オブジェクトからテンプレート・オブジェクト定義を作成し、ターゲット配置テンプレートに追加します。
23-18 ページ「CREATE_REFRESH_TEMPLATE ファンクション」	データベース管理者がテンプレート名、プライベートまたはパブリックのステータスおよびターゲット・リフレッシュ・グループの定義を行うことができる、配置テンプレートを作成します。
23-20 ページ「CREATE_TEMPLATE_OBJECT ファンクション」	オブジェクト定義をターゲットの配置テンプレート・コンテナに追加します。
23-23 ページ「CREATE_TEMPLATE_PARM ファンクション」	指定された配置テンプレートのパラメータを作成して、リモート・マテリアライズド・ビュー・サイトでカスタム・データ・セットを作成します。
23-25 ページ「CREATE_USER_AUTHORIZATION ファンクション」	特定のユーザーに対して、プライベート配置テンプレートをインスタンス化する許可を与えます。
23-27 ページ「CREATE_USER_PARM_VALUE ファンクション」	特定のユーザーに対して、配置テンプレートのパラメータ値を事前定義します。

表 23-1 DBMS\_REPCAT\_RGT パッケージのサブプログラム (続き)

サブプログラム	説明
23-29 ページ「DELETE_RUNTIME_PARMS プロシージャ」	INSERT_RUNTIME_PARMS プロシージャを使用して定義したランタイム・パラメータ値を削除します。
23-30 ページ「DROP_ALL_OBJECTS プロシージャ」	データベース管理者が、すべてのオブジェクトまたは特定のオブジェクト型を、配置テンプレートから削除します。
23-31 ページ「DROP_ALL_TEMPLATE_PARMS プロシージャ」	データベース管理者が指定された配置テンプレートのテンプレート・パラメータを削除します。
23-32 ページ「DROP_ALL_TEMPLATE_SITES プロシージャ」	DBA_REPCAT_TEMPLATE_SITES ビューからすべてのエントリを削除します。
23-33 ページ「DROP_ALL_TEMPLATES プロシージャ」	プロシージャがコールされたサイトで、すべての配置テンプレートを削除します。
23-33 ページ「DROP_ALL_USER_AUTHORIZATIONS プロシージャ」	データベース管理者が、指定された配置テンプレートに対して与えられた、すべてのユーザー認証を削除します。
23-34 ページ「DROP_ALL_USER_PARM_VALUES プロシージャ」	指定された配置テンプレートのユーザー・パラメータ値を削除します。
23-35 ページ「DROP_REFRESH_TEMPLATE プロシージャ」	配置テンプレートを削除します。
23-36 ページ「DROP_SITE_INSTANTIATION プロシージャ」	DBA_REPCAT_TEMPLATE_SITES ビューからターゲット・サイトを削除します。
23-37 ページ「DROP_TEMPLATE_OBJECT プロシージャ」	指定された配置テンプレートからテンプレート・オブジェクトを削除します。
23-38 ページ「DROP_TEMPLATE_PARM プロシージャ」	DBA_REPCAT_TEMPLATE_PARMS ビューから既存のテンプレート・パラメータを削除します。
23-39 ページ「DROP_USER_AUTHORIZATION プロシージャ」	DBA_REPCAT_USER_AUTHORIZATIONS ビューからユーザー認証エントリを削除します。
23-40 ページ「DROP_USER_PARM_VALUE プロシージャ」	特定の配置テンプレートに対して事前定義されたユーザー・パラメータ値を削除します。
23-41 ページ「GET_RUNTIME_PARM_ID ファンクション」	ランタイム・パラメータ値を定義する際に使用する ID を取得します。
23-41 ページ「INSERT_RUNTIME_PARMS プロシージャ」	テンプレートのインスタンス化の前に、ランタイム・パラメータ値を定義します。

表 23-1 DBMS\_REPCAT\_RGT パッケージのサブプログラム（続き）

サブプログラム	説明
23-43 ページ「 <a href="#">INSTANTIATE_OFFLINE</a> ファンクション」	オフラインでリモート・マテリアライズド・ビュー・サイトにマテリアライズド・ビュー環境を構築する際に使用するスクリプトを、マスター・サイトに生成します。
23-45 ページ「 <a href="#">INSTANTIATE_ONLINE</a> ファンクション」	オンラインでリモート・マテリアライズド・ビュー・サイトにマテリアライズド・ビュー環境を構築する際に使用するスクリプトを、マスター・サイトに生成します。
23-48 ページ「 <a href="#">LOCK_TEMPLATE_EXCLUSIVE</a> プロシージャ」	配置テンプレートが更新または変更されている間、ユーザーによるテンプレートの読取りやインスタンス化を行わないようにします。
23-48 ページ「 <a href="#">LOCK_TEMPLATE_SHARED</a> プロシージャ」	指定された配置テンプレートを読取り専用にします。

## ALTER\_REFRESH\_TEMPLATE プロシージャ

このプロシージャを実行すると、データベース管理者が既存の配置テンプレートを変更できます。変更作業には、新しい配置テンプレート名、新しいリフレッシュ・グループまたは新しい所有者の定義、およびパブリックまたはプライベートへのステータス変更などがあります。

### 構文

```
DBMS_REPCAT_RGT.ALTER_REFRESH_TEMPLATE (  
  refresh_template_name      IN   VARCHAR2,  
  new_owner                  IN   VARCHAR2 := '-',  
  new_refresh_group_name     IN   VARCHAR2 := '-',  
  new_refresh_template_name  IN   VARCHAR2 := '-',  
  new_template_comment      IN   VARCHAR2 := '-',  
  new_public_template        IN   VARCHAR2 := '-',  
  new_last_modified          IN   DATE := to_date('1', 'J'),  
  new_modified_by            IN   NUMBER := -1e-130);
```



## パラメータ

**表 23-2 ALTER\_REFRESH\_TEMPLATE プロシージャのパラメータ**

パラメータ	説明
refresh_template_name	変更される配置テンプレートの名前です。
new_owner	配置テンプレートの新しい所有者の名前です。現行の所有者を維持する場合は、値を指定しないでください。
new_refresh_group_name	必要な場合は、このパラメータを使用して、テンプレート・オブジェクトを追加する新しいリフレッシュ・グループの名前を指定します。現行のリフレッシュ・グループを維持する場合は、値を指定しないでください。
new_refresh_template_name	新しい配置テンプレート名を指定するには、このパラメータを使用します。現行の配置テンプレート名を維持する場合は、値を指定しないでください。
new_template_comment	新しい配置テンプレート・コメントです。現行のテンプレート・コメントを維持する場合は、値を指定しないでください。
new_public_template	配置テンプレートがパブリックであるかプライベートであるかを指定します。受け付けられる値は、Y（パブリック）および N（プライベート）です。カレント値を維持する場合は、値を指定しないでください。
new_last_modified	この配置テンプレートが最後に変更された日付です。値を指定しない場合は、現行の日付が自動的に使用されます。
new_modified_by	この配置テンプレートを最後に変更したユーザーの名前です。値を指定しない場合は、現行のユーザー名が自動的に使用されます。

## 例外

**表 23-3 ALTER\_REFRESH\_TEMPLATE プロシージャの例外**

例外	説明
miss_refresh_template	指定された配置テンプレート名が無効か、または存在しません。
bad_public_template	public_template パラメータが正しく指定されませんでした。public_template パラメータは、パブリック・テンプレートの場合は Y、プライベート・テンプレートの場合は N を指定します。
dupl_refresh_template	指定された名前のテンプレートは、すでに存在しています。「 <a href="#">ALL_REPCAT_REFRESH_TEMPLATES</a> ビュー」を参照してください。

## ALTER\_TEMPLATE\_OBJECT プロシージャ

このプロシージャは、指定された配置テンプレートに追加されたオブジェクトを変更します。一般的な変更処理は、オブジェクト DDL の変更および他の配置テンプレートへのオブジェクトの再割当てです。

配置テンプレートに加えられた変更は、配置テンプレートをインスタンス化する新しいサイトのみに反映されます。テンプレートをすでにインスタンス化したリモート・サイトで変更を適用するには、配置テンプレートを再びインスタンス化する必要があります。

### 構文

```
DBMS_REPCAT_RGT.ALTER_TEMPLATE_OBJECT (
  refresh_template_name      IN   VARCHAR2,
  object_name                 IN   VARCHAR2,
  object_type                 IN   VARCHAR2,
  new_refresh_template_name  IN   VARCHAR2 := '-',
  new_object_name             IN   VARCHAR2 := '-',
  new_object_type             IN   VARCHAR2 := '-',
  new_ddl_text                IN   CLOB    := '-',
  new_master_rollback_seg    IN   VARCHAR2 := '-',
  new_flavor_id               IN   NUMBER  := -1e-130);
```

### パラメータ

表 23-4 ALTER\_TEMPLATE\_OBJECT プロシージャのパラメータ

パラメータ	説明
refresh_template_name	変更するオブジェクトが含まれる配置テンプレート名です。
object_name	変更するテンプレート・オブジェクトの名前です。
object_type	変更するオブジェクトのタイプです。
new_refresh_template_name	このオブジェクトの再割当てを行う新しい配置テンプレートの名前です。現行の配置テンプレートへのオブジェクトの割当てを維持する場合は、値を指定しないでください。
new_object_name	テンプレート・オブジェクトの新しい名前です。現行のオブジェクト名を維持する場合は、値を指定しないでください。
new_object_type	このパラメータを指定すると、オブジェクト型が変更されます。次のオブジェクト型を指定できます。 <div><div>SNAPSHOT</div><div>INDEX</div><div>TABLE</div><div>VIEW</div><div>SYNONYM</div><div>SEQUENCE</div><div>PROCEDURE</div><div>FUNCTION</div><div>PACKAGE</div><div>PACKAGE BODY</div><div>TRIGGER</div><div>DATABASE LINK</div></div>

表 23-4 ALTER\_TEMPLATE\_OBJECT プロシージャのパラメータ (続き)

パラメータ	説明
new_ddl_text	指定されたオブジェクトの新しいオブジェクト DDL です。現行のオブジェクト DDL を維持する場合は、新しい DDL テキストを指定しないでください。
new_master_rollback_seg	指定されたオブジェクトの新しいマスター・ロールバック・セグメントです。現行のロールバック・セグメントを維持する場合は、値を指定しないでください。
new_flavor_id	このパラメータは内部で使います。  <b>注意：</b> オラクル社カスタマ・サポート・センターから指示された場合を除き、このパラメータを設定しないでください。

例外

表 23-5 ALTER\_TEMPLATE\_OBJECT プロシージャの例外

例外	説明
miss_refresh_template	指定された配置テンプレート名が無効か、または存在しません。
miss_flavor_id	この例外が発生した場合は、オラクル社カスタマ・サポート・センターまでお問い合わせください。
bad_object_type	オブジェクト型が正しく指定されませんでした。有効なオブジェクト型は、表 23-4 を参照してください。
miss_template_object	指定されたテンプレート・オブジェクト名が無効か、または存在しません。
dupl_template_object	new_refresh_template_name パラメータで指定された新しいテンプレート名は、すでに存在しています。

## 使用上の注意

ALTER\_TEMPLATE\_OBJECT プロシージャでは CLOB が利用されています。このため、ALTER\_TEMPLATE\_OBJECT プロシージャを実行するときは、DBMS\_LOB パッケージを使用する必要があります。次の例は、ALTER\_TEMPLATE\_OBJECT プロシージャで DBMS\_LOB パッケージを使用する方法を示します。

```
DECLARE
    tempstring VARCHAR2(100);
    templob CLOB;
BEGIN
    DBMS_LOB.CREATETEMPORARY(templob, TRUE, DBMS_LOB.SESSION);
    tempstring := 'CREATE MATERIALIZED VIEW mview_sales AS SELECT *
        FROM sales WHERE salesperson = :salesid and region_id = :region';
    DBMS_LOB.WRITE(templob, length(tempstring), 1, tempstring);
    DBMS_REPCAT_RGT.ALTER_TEMPLATE_OBJECT(
        refresh_template_name => 'rgt_personnel',
        object_name => 'MVIEW_SALES',
        object_type => 'SNAPSHOT',
        new_ddl_text => templob);
    DBMS_LOB.FREETEMPORARY(templob);
END;
/
```

## ALTER\_TEMPLATE\_PARM プロシージャ

このプロシージャを実行すると、データベース管理者が、指定された配置テンプレートのパラメータを変更できます。変更作業には、パラメータの改名や、デフォルト値およびプロンプト文字列の再定義などがあります。

## 構文

```
DBMS_REPCAT_RGT.ALTER_TEMPLATE_PARM (
    refresh_template_name      IN   VARCHAR2,
    parameter_name             IN   VARCHAR2,
    new_refresh_template_name  IN   VARCHAR2 := '-',
    new_parameter_name         IN   VARCHAR2 := '-',
    new_default_parm_value     IN   CLOB := NULL,
    new_prompt_string          IN   VARCHAR2 := '-',
    new_user_override          IN   VARCHAR2 := '-');
```

## パラメータ

**表 23-6 ALTER\_TEMPLATE\_PARM プロシージャのパラメータ**

パラメータ	説明
refresh_template_name	変更するパラメータを持つ配置テンプレートの名前です。
parameter_name	変更するパラメータの名前です。
new_refresh_template_name	指定されたパラメータが新しく割り当てられる配置テンプレートの名前です。あるテンプレートから他のテンプレートにパラメータを移動するときに便利です。現行のテンプレートへの割当てを維持する場合は、値を指定しないでください。
new_parameter_name	テンプレート・パラメータの新しい名前です。現行のパラメータ名を維持する場合は、値を指定しないでください。
new_default_parm_value	指定されたパラメータの新しいデフォルト値です。現行のデフォルト値を維持する場合は、値を指定しないでください。
new_prompt_string	指定されたパラメータの新しいプロンプト・テキストです。現行のプロンプト文字列を維持する場合は、値を指定しないでください。
new_user_override	インスタンス化処理で確認メッセージが表示された場合、ユーザーがデフォルト値を上書きできるかどうかを決定します。このパラメータにユーザー・パラメータ値が定義されていないと、確認メッセージが表示されます。ユーザーがデフォルト値を上書きする場合は Y に、上書きできないようにする場合は N に設定します。

## 例外

**表 23-7 ALTER\_TEMPLATE\_PARM プロシージャの例外**

例外	説明
miss_refresh_template	指定された配置テンプレート名が無効か、または存在しません。
miss_template_parm	指定されたテンプレート・パラメータが無効か、または存在しません。
dupl_template_parm	new_refresh_template_name と new_parameter_name の組合せは、すでに存在しています。

## 使用上の注意

ALTER\_TEMPLATE\_PARM プロシージャでは CLOB が利用されています。このため、ALTER\_TEMPLATE\_PARM プロシージャを実行するときは、DBMS\_LOB パッケージを使用する必要があります。次の例は、ALTER\_TEMPLATE\_PARM プロシージャで DBMS\_LOB パッケージを使用する方法を示します。

```
DECLARE
    tempstring VARCHAR2(100);
    templob CLOB;
BEGIN
    DBMS_LOB.CREATETEMPORARY(templob, TRUE, DBMS_LOB.SESSION);
    tempstring := 'REGION 20';
    DBMS_LOB.WRITE(templob, length(tempstring), 1, tempstring);
    DBMS_REPCAT_RGT.ALTER_TEMPLATE_PARM(
        refresh_template_name => 'rgt_personnel',
        parameter_name => 'region',
        new_default_parm_value => templob);
    DBMS_LOB.FREETEMPORARY(templob);
END;
/
```

## ALTER\_USER\_AUTHORIZATION プロシージャ

このプロシージャは、DBA\_REPCAT\_USER\_AUTHORIZATIONS ビューの内容を変更します。具体的には、ユーザーまたは配置テンプレートの認可の割当てを変更できます。このプロシージャは、たとえばある従業員の職場が変更になり、他の配置テンプレートのマテリアライズド・ビュー環境を必要とする場合などに役立ちます。データベース管理者が単にこの従業員に新しい配置テンプレートを割り当てることによって、このユーザーにターゲット・テンプレートをインスタンス化する許可が与えられます。

## 構文

```
DBMS_REPCAT_RGT.ALTER_USER_AUTHORIZATION (
    user_name           IN   VARCHAR2,
    refresh_template_name IN  VARCHAR2,
    new_user_name       IN   VARCHAR2 := '-',
    new_refresh_template_name IN  VARCHAR2 := '-');
```

## パラメータ

**表 23-8 ALTER\_USER\_AUTHORIZATION プロシージャのパラメータ**

パラメータ	説明
user_name	認可の変更が行われるユーザーの名前です。
refresh_template_name	指定されたユーザーに現在割り当てられている配置テンプレートの名前です。
new_user_name	このテンプレートを認可する新しいユーザーを定義するには、このパラメータを使用します。カレント・ユーザーを維持する場合は、値を指定しないでください。
new_refresh_template_name	指定されたユーザー（既存のユーザーまたは新しいユーザー）が、インスタンス化する許可を持つ配置テンプレートです。カレント配置テンプレートを維持する場合は、値を指定しないでください。

## 例外

**表 23-9 ALTER\_USER\_AUTHORIZATION プロシージャの例外**

例外	説明
miss_user_authorization	指定された user_name 値と refresh_template_name 値の組合せは、DBA_REPCAT_USER_AUTHORIZATIONS ビューに存在しません。
miss_user	new_user_name パラメータまたは user_name パラメータに指定されたユーザー名が無効か、または存在しません。
miss_refresh_template	new_refresh_template パラメータに指定された配置テンプレートが無効か、または存在しません。
dupl_user_authorization	指定されたユーザー名および配置テンプレート名用の行がすでに存在しています。「 <a href="#">ALL_REPCAT_USER_AUTHORIZATIONS</a> ビュー」を参照してください。

## ALTER\_USER\_PARM\_VALUE プロシージャ

このプロシージャは、指定されたユーザーに定義されている既存のパラメータ値を変更します。マテリアライズド・ビュー環境で割当て表が使用されている場合に、このプロシージャを使用すると便利です。ユーザー・パラメータ値を変更すると、リモート・マテリアライズド・ビュー・サイトのデータ・セットを簡単かつ安全に変更できます。

**関連項目：** 割当て表の使用方法の詳細は、『Oracle9i アドバンスト・レプリケーション』を参照してください。

### 構文

```
DBMS_REPCAT_RGT.ALTER_USER_PARM_VALUE(  
    refresh_template_name      IN   VARCHAR2,  
    parameter_name             IN   VARCHAR2,  
    user_name                  IN   VARCHAR2,  
    new_refresh_template_name  IN   VARCHAR2 := '-',  
    new_parameter_name         IN   VARCHAR2 := '-',  
    new_user_name              IN   VARCHAR2 := '-',  
    new_parm_value             IN   CLOB := NULL);
```

### パラメータ

表 23-10 ALTER\_USER\_PARM\_VALUE プロシージャのパラメータ

パラメータ	説明
refresh_template_name	変更するユーザー・パラメータ値を持つ配置テンプレートの名前です。
parameter_name	変更するパラメータの名前です。
user_name	パラメータ値が変更されるユーザーの名前です。
new_refresh_template_name	指定されたユーザー・パラメータ値が新しく割り当てられる配置テンプレートの名前です。他のテンプレートに対してユーザーを認可するときに便利です。現行のテンプレートへの割当てを維持する場合は、値を指定しないでください。
new_parameter_name	新しいテンプレート・パラメータの名前です。既存のパラメータに定義されたユーザー値を維持する場合は、値を指定しないでください。
new_user_name	このパラメータ値を割り当てる新しいユーザー名です。カレント・ユーザーへ割り当てられたパラメータ値を維持する場合は、値を指定しないでください。
new_parm_value	指定されたユーザー・パラメータの新しいパラメータ値です。現行のパラメータ値を維持する場合は、値を指定しないでください。



例外

表 23-11 ALTER\_USER\_PARM\_VALUE プロシージャの例外

例外	説明
miss_refresh_template	指定された配置テンプレート名が無効か、または存在しません。
miss_template_parm	指定されたテンプレート・パラメータが無効か、または存在しません。
miss_user	user_name パラメータまたは new_user_name パラメータに指定されたユーザー名が無効か、または存在しません。
miss_user_parm_values	指定されたユーザー・パラメータ値は存在しません。
dupl_user_parm_values	指定された新しいユーザー・パラメータはすでに存在しています。

使用上の注意

ALTER\_USER\_PARM\_VALUE プロシージャでは CLOB が利用されています。このため、ALTER\_USER\_PARM\_VALUE プロシージャを実行するときは、DBMS\_LOB パッケージを使用する必要があります。次の例は、ALTER\_USER\_PARM\_VALUE プロシージャで DBMS\_LOB パッケージを使用する方法を示します。

```
DECLARE
    tempstring VARCHAR2(100);
    templob CLOB;
BEGIN
    DBMS_LOB.CREATETEMPORARY(templob, TRUE, DBMS_LOB.SESSION);
    tempstring := 'REGION 20';
    DBMS_LOB.WRITE(templob, length(tempstring), 1, tempstring);
    DBMS_REPCAT_RGT.ALTER_USER_PARM_VALUE(
        refresh_template_name => 'rgt_personnel',
        parameter_name => 'region',
        user_name => 'BOB',
        new_parm_value => templob);
    DBMS_LOB.FREETEMPORARY(templob);
END;
```

## COMPARE\_TEMPLATES ファンクション

このファンクションを実行すると、データベース管理者が 2 つの配置テンプレートの内容を比較できます。2 つの配置テンプレートの内容の相違点は、USER\_REPCAT\_TEMP\_OUTPUT 一時ビューに格納されます。

COMPARE\_TEMPLATES ファンクションでは、USER\_REPCAT\_TEMP\_OUTPUT 一時ビューの問合せの際に WHERE 句で指定する数値が戻されます。たとえば、COMPARE\_TEMPLATES プロシージャで 10 が戻された場合は、次の SELECT 文を実行して、指定された 2 つのテンプレートのすべての相違点を確認します。2 つのテンプレートの内容が同一の場合は、SELECT 文で行が戻されません。

```
SELECT TEXT FROM USER_REPCAT_TEMP_OUTPUT
WHERE OUTPUT_ID = 10 ORDER BY LINE;
```

接続が解除されるか、またはロールバックが実行されると、USER\_REPCAT\_TEMP\_OUTPUT 一時ビューの内容が失われます。

### 構文

```
DBMS_REPCAT_RGT.COMPARE_TEMPLATES (
  source_template_name  IN  VARCHAR2,
  compare_template_name IN  VARCHAR2)
return NUMBER;
```

### パラメータ

表 23-12 COMPARE\_TEMPLATES ファンクションのパラメータ

パラメータ	説明
source_template_name	比較する最初の配置テンプレートの名前です。
compare_template_name	比較する 2 番目の配置テンプレートの名前です。

### 例外

表 23-13 COMPARE\_TEMPLATES ファンクションの例外

例外	説明
miss_refresh_template	比較する配置テンプレート名が無効か、または存在しません。

戻り値

表 23-14 COMPARE\_TEMPLATES ファンクションの戻り値

戻り値	説明
< システム生成番号 >	USER_REPCAT_TEMP_OUTPUT 一時ビューから選択して、比較するテンプレート間の違いを表示するときに、output_id 値に戻される番号を示します。

COPY\_TEMPLATE ファンクション

このファンクションでは配置テンプレートをコピーできます。これは新しい配置テンプレートが既存の配置テンプレート内のオブジェクトの多くを使用する場合に便利です。このファンクションは、配置テンプレート、テンプレート・オブジェクト、テンプレート・パラメータおよびユーザー・パラメータ値をコピーします。データベース管理者は、このファンクションを使用して、テンプレートのユーザー認証をコピーできます。このファンクションで戻される数値は、配置テンプレートの管理用に、Oracle 内部で使います。

**注意：** DBA\_REPCAT\_TEMPLATE\_SITES ビュー内の値はコピーされません。

このファンクションを使用すると、データベース管理者が配置テンプレートを他のマスター・サイトにコピーできます。この機能は、配置テンプレートの配布や、複数のサイトでのネットワーク負荷の分割に役立ちます。

構文

```
DBMS_REPCAT_RGT.COPY_TEMPLATE (
  old_refresh_template_name    IN   VARCHAR2,
  new_refresh_template_name    IN   VARCHAR2,
  copy_user_authorizations     IN   VARCHAR2,
  dblink                       IN   VARCHAR2 := NULL)
return NUMBER;
```

パラメータ

表 23-15 COPY\_TEMPLATE ファンクションのパラメータ

パラメータ	説明
old_refresh_template_name	コピーされる配置テンプレートの名前です。
new_refresh_template_name	新しい配置テンプレートの名前です。
copy_user_authorizations	元のテンプレートに対する認可を、新しい配置テンプレートにコピーするかどうかを指定します。このパラメータに受け付けられる値は、Y、N および NULL です。  <b>注意：</b> すべてのユーザーが、ターゲット・データベースに存在する必要があります。
dblink	配置テンプレートのコピー元を定義します。配置テンプレートを他のマスター・サイトに配布するときに便利です。コピー元を指定しない場合、配置テンプレートはローカル・マスター・サイトからコピーされます。

例外

表 23-16 COPY\_TEMPLATE ファンクションの例外

例外	説明
miss_refresh_template	コピーする配置テンプレート名が無効か、または存在しません。
dupl_refresh_template	指定された新しいリフレッシュ・テンプレートの名前は、すでに存在しています。
bad_copy_auth	copy_user_authorization パラメータに指定された値は無効です。受け付けられる値は、Y、N および NULL です。

戻り値

表 23-17 COPY\_TEMPLATE ファンクションの戻り値

戻り値	説明
< システム生成番号 >	Oracle 内部で使用するシステム生成番号です。

## CREATE\_OBJECT\_FROM\_EXISTING ファンクション

このファンクションは、既存のデータベース・オブジェクトからテンプレート・オブジェクト定義を作成して、ターゲット配置テンプレートに追加します。ターゲット配置テンプレートがリモート・マテリアライズド・ビュー・サイトでインスタンス化されるときに、元のデータベース・オブジェクトを作成した DDL が実行されます。既存のトリガーやプロシージャをテンプレートに追加する場合、このファンクションを使用すると便利です。このファンクションで戻される数値は、配置テンプレートの管理用に、Oracle 内部で使います。

### 構文

```
DBMS_REPCAT_RGT.CREATE_OBJECT_FROM_EXISTING(  
    refresh_template_name IN VARCHAR2,  
    object_name           IN VARCHAR2,  
    sname                 IN VARCHAR2,  
    oname                 IN VARCHAR2,  
    otype                 IN VARCHAR2)  
return NUMBER;
```

### パラメータ

表 23-18 CREATE\_OBJECT\_FROM\_EXISTING ファンクションのパラメータ

パラメータ	説明
refresh_template_name	このオブジェクトが追加される配置テンプレートの名前です。
object_name	オプションで、配置テンプレートに追加する既存のオブジェクトの、新しい名前を指定します（既存のオブジェクトに新しい名前を定義できます）。
sname	作成するテンプレート・オブジェクトの基になるオブジェクトを含むスキーマです。
oname	作成するテンプレート・オブジェクトの基になるオブジェクトの名前です。
otype	テンプレートに追加するデータベース・オブジェクトのタイプ（PROCEDURE、TRIGGER など）です。次のオブジェクト型を指定できます（DATABASE LINK、MATERIALIZED VIEW および SNAPSHOT は、このファンクションに有効なオブジェクト型ではありません）。 <div><div>SEQUENCE</div><div>INDEX</div><div>TABLE</div><div>VIEW</div><div>SYNONYM</div><div>PROCEDURE</div><div>FUNCTION</div><div>PACKAGE</div><div>PACKAGE BODY</div><div>TRIGGER</div></div>

例外

表 23-19 CREATE\_OBJECT\_FROM\_EXISTING ファンクションの例外

例外	説明
miss_refresh_template	指定されたリフレッシュ・テンプレート名が無効か、または存在しません。既存の配置テンプレートを確認するには、DBA_REPCAT_REFRESH_TEMPLATES ビューの問合せを行います。
bad_object_type	オブジェクト型が正しく指定されませんでした。
dupl_template_object	指定された配置テンプレートには、同名で同タイプのオブジェクトがすでに追加されています。
objectmissing	指定されたオブジェクトは存在しません。

戻り値

表 23-20 CREATE\_OBJECT\_FROM\_EXISTING ファンクションの戻り値

戻り値	説明
< システム生成番号 >	Oracle 内部で使用するシステム生成番号です。

CREATE\_REFRESH\_TEMPLATE ファンクション

このファンクションは、テンプレート名、プライベートまたはパブリックのステータスおよびターゲット・リフレッシュ・グループの定義を行う、配置テンプレートを作成します。テンプレート・オブジェクト、ユーザー認証またはテンプレート・パラメータを作成するたびに、このファンクションで作成される配置テンプレートを参照します。このファンクションは、DBA\_REPCAT\_REFRESH\_TEMPLATES ビューに行を追加します。このファンクションで戻される数値は、配置テンプレートの管理用に、Oracle 内部で使用します。

構文

```
DBMS_REPCAT_RGT.CREATE_REFRESH_TEMPLATE (  
  owner                IN   VARCHAR2,  
  refresh_group_name   IN   VARCHAR2,  
  refresh_template_name IN   VARCHAR2,  
  template_comment     IN   VARCHAR2 := NULL,  
  public_template      IN   VARCHAR2 := NULL,  
  last_modified        IN   DATE := SYSDATE,  
  modified_by          IN   VARCHAR2 := USER,  
  creation_date        IN   DATE := SYSDATE,  
  created_by           IN   VARCHAR2 := USER)  
return NUMBER;
```

## パラメータ

**表 23-21 CREATE\_REFRESH\_TEMPLATE ファンクションのパラメータ**

パラメータ	説明
owner	このパラメータで、配置テンプレートの所有者のユーザー名を指定します。所有者を指定しない場合、テンプレートを作成したユーザーの名前が自動的に使用されます。
refresh_group_name	このテンプレートのインスタンス化する時に作成される、リフレッシュ・グループの名前です。このテンプレートで作成されるすべてのオブジェクトは、指定されたリフレッシュ・グループに割り当てられます。
refresh_template_name	作成する配置テンプレートの名前です。この配置テンプレートを使用するすべてのアクティビティで、この名前が参照されます。
template_comment	このパラメータで定義されるユーザー・コメントは、DBA_REPCAT_REFRESH_TEMPLATES ビューにリストされます。
public_template	配置テンプレートがパブリックであるかプライベートであるかを指定します。受け付けられる値は、Y（パブリック）および N（プライベート）です。
last_modified	この配置テンプレートが最後に変更された日付です。値を指定しない場合は、現行の日付が自動的に使用されます。
modified_by	この配置テンプレートを最後に変更したユーザーの名前です。値を指定しない場合は、現行のユーザー名が自動的に使用されます。
creation_date	この配置テンプレートが作成された日付です。値を指定しない場合は、現行の日付が自動的に使用されます。
created_by	この配置テンプレートを作成したユーザーの名前です。値を指定しない場合は、現行のユーザー名が自動的に使用されます。

## 例外

**表 23-22 CREATE\_REFRESH\_TEMPLATE ファンクションの例外**

例外	説明
dupl_refresh_template	指定された名前のテンプレートは、すでに存在しています。既存のテンプレートを確認するには、 <a href="#">ALL_REPCAT_REFRESH_TEMPLATES</a> ビューを参照してください。
bad_public_template	public_template パラメータが正しく指定されませんでした。public_template パラメータは、パブリック・テンプレートの場合は Y、プライベート・テンプレートの場合は N を指定します。

戻り値

表 23-23 CREATE\_REFRESH\_TEMPLATE ファンクションの戻り値

戻り値	説明
< システム生成番号 >	Oracle 内部で使用するシステム生成番号です。

CREATE\_TEMPLATE\_OBJECT ファンクション

このファンクションは、オブジェクト定義をターゲットの配置テンプレート・コンテナに追加します。ターゲット配置テンプレートがリモート・マテリアライズド・ビュー・サイトでインスタンス化されるときに、指定されたオブジェクト DDL が実行されます。マテリアライズド・ビュー以外にも、このファンクションを使用して、表、プロシージャおよび他のオブジェクトをテンプレートに追加できます。このファンクションで戻される数値は、配置テンプレートの管理用に、Oracle 内部で使われます。

構文

```
DBMS_REPCAT_RGT.CREATE_TEMPLATE_OBJECT (  
  refresh_template_name IN VARCHAR2,  
  object_name           IN VARCHAR2,  
  object_type           IN VARCHAR2,  
  ddl_text              IN CLOB,  
  master_rollback_seg   IN VARCHAR2 := NULL,  
  flavor_id             IN NUMBER := -1e-130)  
return NUMBER;
```



パラメータ

表 23-24 CREATE\_TEMPLATE\_OBJECT ファンクションのパラメータ

パラメータ	説明
refresh_template_name	このオブジェクトが追加される配置テンプレートの名前です。
object_name	作成するテンプレート・オブジェクトの名前です。
object_type	テンプレートに追加するデータベース・オブジェクトのタイプ (SNAPSHOT、TRIGGER、PROCEDURE など) です。次のオブジェクト型を指定できます。 <div><div>SNAPSHOT</div><div>INDEX</div><div>TABLE</div><div>VIEW</div><div>SYNONYM</div><div>SEQUENCE</div><div>PROCEDURE</div><div>FUNCTION</div><div>PACKAGE</div><div>PACKAGE BODY</div><div>TRIGGER</div><div>DATABASE LINK</div></div>
ddl_text	テンプレートに追加するオブジェクトを作成する DDL です。DDL は、必ずセミコロンで終了するようにします。テンプレート・オブジェクト用にテンプレート・パラメータを作成する場合は、コロン (:) を使用できます。詳細は、 <a href="#">第 4 章「配置テンプレートの作成」</a> を参照してください。  CREATE MATERIALIZED VIEW 文でマテリアライズド・ビュー (スナップショット) を追加する場合は、マテリアライズド・ビュー問合せでマスター表の所有者のスキーマ名を指定してください。
master_rollback_seg	定義されたオブジェクト DDL をリモート・マテリアライズド・ビュー・サイトで実行する際に使用する、ロールバック・セグメントの名前を指定します。
flavor_id	このパラメータは内部で使います。  <b>注意：</b> オラクル社カスタマ・サポート・センターから指示された場合を除き、このパラメータを設定しないでください。

例外

表 23-25 CREATE\_TEMPLATE\_OBJECT ファンクションの例外

例外	説明
miss_refresh_template	指定されたリフレッシュ・テンプレート名が無効か、または存在しません。既存の配置テンプレートを確認するには、DBA_REPCAT_REFRESH_TEMPLATES ビューの問合せを行います。
bad_object_type	オブジェクト型が正しく指定されませんでした。有効なオブジェクト型は、表 23-24 を参照してください。
dupl_template_object	指定された配置テンプレートには、同名で同タイプのオブジェクトがすでに追加されています。

戻り値

表 23-26 CREATE\_TEMPLATE\_OBJECT ファンクションの戻り値

戻り値	説明
< システム生成番号 >	Oracle 内部で使用されるシステム生成番号です。

使用上の注意

CREATE\_TEMPLATE\_OBJECT ファンクションでは、CLOB が利用されています。このため、CREATE\_TEMPLATE\_OBJECT ファンクションを実行するときは、DBMS\_LOB パッケージを使用する必要があります。次の例は、CREATE\_TEMPLATE\_OBJECT ファンクションで DBMS\_LOB パッケージを使用する方法を示します。

```
DECLARE
    tempstring VARCHAR2(100);
    templob CLOB;
    a NUMBER;
BEGIN
    DBMS_LOB.CREATETEMPORARY(templob, TRUE, DBMS_LOB.SESSION);
    tempstring := 'CREATE MATERIALIZED VIEW mview_sales AS SELECT *
        FROM sales WHERE salesperson = :salesid';
    DBMS_LOB.WRITE(templob, length(tempstring), 1, tempstring);
    a := DBMS_REPCAT_RGT.CREATE_TEMPLATE_OBJECT(
        refresh_template_name => 'rgt_personnel',
        object_name => 'mview_sales',
        object_type => 'SNAPSHOT',
        ddl_text => templob,
        master_rollback_seg => 'RBS');
```

```
DBMS_LOB.FREETEMPORARY(templob);  
END;  
/
```

## CREATE\_TEMPLATE\_PARM ファンクション

このファンクションを実行すると、指定された配置テンプレートのパラメータが作成され、リモート・マテリアライズド・ビュー・サイトでカスタム・データ・セットを作成できます。このファンクションは、データベース管理者が、テンプレート・オブジェクトを追加する前にテンプレート変数のセットを定義する場合のみに必要です。

CREATE\_TEMPLATE\_OBJECT ファンクションを使用してオブジェクトがテンプレートに追加されると、オブジェクト DDL のすべての変数が自動的に DBA\_REPCAT\_TEMPLATE\_PARS ビューに追加されます。

データベース管理者は、通常、ALTER\_TEMPLATE\_PARM プロシージャを使用して、デフォルトのパラメータ値やプロンプト文字列を修正します。詳細は、23-8 ページの「[ALTER\\_TEMPLATE\\_PARM プロシージャ](#)」を参照してください。このファンクションで戻される数値は、配置テンプレートの管理用に、Oracle 内部で使用します。

## 構文

```
DBMS_REPCAT_RGT.CREATE_TEMPLATE_PARM (  
    refresh_template_name  IN   VARCHAR2,  
    parameter_name         IN   VARCHAR2,  
    default_parm_value     IN   CLOB := NULL,  
    prompt_string          IN   VARCHAR2 := NULL,  
    user_override          IN   VARCHAR2 := NULL)  
return NUMBER;
```

パラメータ

表 23-27 CREATE\_TEMPLATE\_PARM ファンクションのパラメータ

パラメータ	説明
refresh_template_name	パラメータを作成する配置テンプレートの名前です。
parameter_name	作成するパラメータの名前です。
default_parm_value	パラメータのデフォルト値は、このパラメータで定義します。ユーザー・パラメータ値やランタイム・パラメータ値がない場合、インスタンス化処理でこのデフォルト値が使用されます。
prompt_string	インスタンス化処理で、このテンプレート・パラメータに表示されるプロンプト・テキストです。
user_override	インスタンス化処理で確認メッセージが表示された場合、ユーザーがデフォルト値を上書きできるかどうかを決定します。このパラメータにユーザー・パラメータ値が定義されていないと、確認メッセージが表示されます。ユーザーがデフォルト値を上書きする場合は Y に、上書きできないようにする場合は N に設定します。

例外

表 23-28 CREATE\_TEMPLATE\_PARM ファンクションの例外

例外	説明
miss_refresh_template	指定されたリフレッシュ・テンプレート名が無効か、または存在しません。
dupl_template_parm	指定された配置テンプレートには、同名のパラメータがすでに定義されています。

戻り値

表 23-29 CREATE\_TEMPLATE\_PARM ファンクションの戻り値

戻り値	説明
< システム生成番号 >	Oracle 内部で使用されるシステム生成番号です。

## 使用上の注意

CREATE\_TEMPLATE\_PARM ファンクションでは CLOB が利用されています。このため、CREATE\_TEMPLATE\_PARM ファンクションを実行するときは、DBMS\_LOB パッケージを使用する必要があります。次の例は、CREATE\_TEMPLATE\_PARM ファンクションで DBMS\_LOB パッケージを使用する方法を示します。

```
DECLARE
    tempstring VARCHAR2(100);
    templob CLOB;
    a NUMBER;
BEGIN
    DBMS_LOB.CREATETEMPORARY(templob, TRUE, DBMS_LOB.SESSION);
    tempstring := 'REGION 20';
    DBMS_LOB.WRITE(templob, length(tempstring), 1, tempstring);
    a := DBMS_REPCAT_RGT.CREATE_TEMPLATE_PARM(
        refresh_template_name => 'rgt_personnel',
        parameter_name => 'region',
        default_parm_value => templob,
        prompt_string => 'Enter your region ID:',
        user_override => 'Y');
    DBMS_LOB.FREETEMPORARY(templob);
END;
/
```

## CREATE\_USER\_AUTHORIZATION ファンクション

このファンクションは、特定のユーザーに対して、プライベート配置テンプレートをインスタンス化する許可を与えます。プライベート配置テンプレートの使用を許可されていないユーザーは、プライベート・テンプレートをインスタンス化できません。このファンクションは、DBA\_REPCAT\_USER\_AUTHORIZATIONS ビューに行を追加します。

ユーザーに許可を与える前に、配置テンプレートをインスタンス化するマスター・サイトに、そのユーザーが存在するかどうかを確認します。このファンクションで戻される数値は、配置テンプレートの管理用に、Oracle 内部で使用します。

## 構文

```
DBMS_REPCAT_RGT.CREATE_USER_AUTHORIZATION (
    user_name          IN   VARCHAR2,
    refresh_template_name IN VARCHAR2)
return NUMBER;
```

パラメータ

表 23-30 CREATE\_USER\_AUTHORIZATION ファンクションのパラメータ

パラメータ	説明
user_name	指定されたテンプレートをインスタンス化する許可を持つユーザーの名前です。複数のユーザーを設定するときは、各ユーザー名をカンマで区切ります ('john, mike, bob' など)。
refresh_template_name	指定されたユーザーにインスタンス化する許可を与えるテンプレートの名前です。

例外

表 23-31 CREATE\_USER\_AUTHORIZATION ファンクションの例外

例外	説明
miss_user	指定されたユーザー名が無効か、または存在しません。
miss_refresh_template	指定されたリフレッシュ・テンプレート名が無効か、または存在しません。
dupl_user_authorization	指定されたユーザーおよび配置テンプレートに対しては、すでに認可が作成されています。テンプレートの認可のリストを確認するには、 <a href="#">ALL_REPCAT_USER_AUTHORIZATIONS</a> ビューを参照してください。

戻り値

表 23-32 CREATE\_USER\_AUTHORIZATION ファンクションの戻り値

戻り値	説明
< システム生成番号 >	Oracle 内部で使用されるシステム生成番号です。

## CREATE\_USER\_PARM\_VALUE ファンクション

このファンクションは、特定のユーザーに対して、配置テンプレートのパラメータ値を事前定義します。たとえば、ユーザー 33456 に対して、領域パラメータを `west` に事前定義する場合、このファンクションを使用します。

このファンクションで指定された値は、テンプレート・パラメータに指定されたデフォルト値に優先します。このファンクションで戻される数値は、配置テンプレートの管理用に、Oracle 内部で使用します。

### 構文

```
DBMS_REPCAT_RGT.CREATE_USER_PARM_VALUE (  
    refresh_template_name    IN    VARCHAR2,  
    parameter_name           IN    VARCHAR2,  
    user_name                IN    VARCHAR2,  
    parm_value               IN    CLOB := NULL)  
return NUMBER;
```

### パラメータ

表 23-33 CREATE\_USER\_PARM\_VALUE ファンクションのパラメータ

パラメータ	説明
refresh_template_name	ユーザー・パラメータ値を作成するパラメータを含む配置テンプレートの名前を指定します。
parameter_name	ユーザー・パラメータ値を定義するテンプレート・パラメータの名前です。
user_name	ユーザー・パラメータ値を事前定義する対象のユーザーの名前を指定します。
parm_value	指定されたユーザーが開始したインスタンス化処理で使用する、事前定義パラメータ値です。

例外

表 23-34 CREATE\_USER\_PARM\_VALUE ファンクションの例外

例外	説明
miss_refresh_template	指定された配置テンプレート名が無効か、または存在しません。
dupl_user_parm_values	指定されたユーザー、パラメータおよび配置テンプレートのパラメータ値は、すでに定義されています。既存のユーザー・パラメータ値を確認するには、DBA_REPCAT_USER_PARM_VALUES ビューの問合せを行います。
miss_template_parm	指定された配置テンプレート・パラメータ名が無効か、または存在しません。
miss_user	指定されたユーザー名が無効か、または存在しません。

戻り値

表 23-35 CREATE\_USER\_PARM\_VALUE ファンクションの戻り値

戻り値	説明
<システム生成番号>	Oracle 内部で使用されるシステム生成番号です。

使用上の注意

CREATE\_USER\_PARM\_VALUE ファンクションでは CLOB が利用されています。このため、このファンクションを実行するときは、DBMS\_LOB パッケージを使用する必要があります。次の例は、CREATE\_USER\_PARM\_VALUE ファンクションで DBMS\_LOB パッケージを使用する方法を示します。

```
DECLARE
    tempstring VARCHAR2(100);
    templob CLOB;
    a NUMBER;
BEGIN
    DBMS_LOB.CREATETEMPORARY(templob, TRUE, DBMS_LOB.SESSION);
    tempstring := 'REGION 20';
    DBMS_LOB.WRITE(templob, length(tempstring), 1, tempstring);
    a := DBMS_REPCAT_RGT.CREATE_USER_PARM_VALUE(
        refresh_template_name => 'rgt_personnel',
        parameter_name => 'region',
        user_name => 'BOB',
        user_parm_value => templob);
    DBMS_LOB.FREETEMPORARY(templob);
END;
/
```



## DELETE\_RUNTIME\_PARMS プロシージャ

INSERT\_RUNTIME\_PARMS プロシージャを使用して定義したランタイム・パラメータ値を削除するには、配置テンプレートをインスタンス化する前に、このプロシージャを実行します。

### 構文

```
DBMS_REPCAT_RGT.DELETE_RUNTIME_PARMS(  
    runtime_parm_id    IN    NUMBER,  
    parameter_name     IN    VARCHAR2);
```

### パラメータ

表 23-36 DELETE\_RUNTIME\_PARMS プロシージャのパラメータ

パラメータ	説明
runtime_parm_id	以前にランタイム・パラメータ値を代入した ID を指定します。この値は、GET_RUNTIME_PARM_ID ファンクションで取得済です。
parameter_name	削除対象のパラメータ値の名前を指定します。配置テンプレート・パラメータを確認するには、DBA_REPCAT_TEMPLATE_PARMS ビューの問合せを行います。

### 例外

表 23-37 DELETE\_RUNTIME\_PARMS プロシージャの例外

例外	説明
miss_template_parm	指定された配置テンプレート・パラメータ名が無効か、または存在しません。

## DROP\_ALL\_OBJECTS プロシージャ

このプロシージャを実行すると、データベース管理者が、すべてのオブジェクトまたは特定のオブジェクト型を、配置テンプレートから削除できます。

**注意：** このプロシージャを実行すると、元に戻せません。実行するときには十分注意してください。

### 構文

```
DBMS_REPCAT_RGT.DROP_ALL_OBJECTS (  
    refresh_template_name    IN    VARCHAR2,  
    object_type               IN    VARCHAR2 := NULL);
```

### パラメータ

表 23-38 DROP\_ALL\_OBJECTS プロシージャのパラメータ

パラメータ	説明
refresh_template_name	削除対象のオブジェクトを持つ配置テンプレートの名前です。
object_type	値を NULL に設定すると、テンプレート内のすべてのオブジェクトが削除されます。オブジェクト型を指定すると、そのタイプのオブジェクトのみが削除されます。次のオブジェクト型を指定できます。 <div><div>SNAPSHOT</div><div>PROCEDURE</div><div>INDEX</div><div>FUNCTION</div><div>TABLE</div><div>PACKAGE</div><div>VIEW</div><div>PACKAGE BODY</div><div>SYNONYM</div><div>TRIGGER</div><div>SEQUENCE</div><div>DATABASE LINK</div></div>

### 例外

表 23-39 DROP\_ALL\_OBJECTS プロシージャの例外

例外	説明
miss_refresh_template	指定された配置テンプレート名が無効か、または存在しません。
bad_object_type	オブジェクト型が正しく指定されませんでした。有効なオブジェクト型は、 <a href="#">表 23-38</a> を参照してください。

## DROP\_ALL\_TEMPLATE\_PARMS プロシージャ

このプロシージャを使用すると、指定された配置テンプレートのテンプレート・パラメータを削除できます。このプロシージャを使用して、テンプレート・オブジェクトが参照しないすべてのパラメータを削除するか、パラメータを参照するすべてのオブジェクトを、パラメータそのものとともにテンプレートから削除できます。

**注意：** このプロシージャを実行すると、元に戻せません。実行するときには十分注意してください。

### 構文

```
DBMS_REPCAT_RGT.DROP_ALL_TEMPLATE_PARMS (  
    refresh_template_name    IN    VARCHAR2,  
    drop_objects              IN    VARCHAR2 := n);
```

### パラメータ

表 23-40 DROP\_ALL\_TEMPLATE\_PARMS プロシージャのパラメータ

パラメータ	説明
refresh_template_name	削除対象のパラメータおよびオブジェクトがある配置テンプレートの名前です。
drop_objects	値を指定しない場合のデフォルトは N になり、テンプレート・オブジェクトが参照しないすべてのパラメータが削除されます。  Y を指定した場合は、テンプレート・パラメータを参照するすべてのオブジェクト、およびテンプレート・パラメータそのものが削除されます。オブジェクトはデータベースからではなく、テンプレートから削除されます。

### 例外

表 23-41 DROP\_ALL\_TEMPLATE\_PARMS プロシージャの例外

例外	説明
miss_refresh_template	指定された配置テンプレート名が無効か、または存在しません。

## DROP\_ALL\_TEMPLATE\_SITES プロシージャ

このプロシージャは、特定の配置テンプレートをインスタンス化したサイトの記録が保存される DBA\_REPCAT\_TEMPLATE\_SITES ビューから、すべてのエントリを削除します。

**注意：** このプロシージャを実行すると、元に戻せません。実行するときは十分注意してください。

### 構文

```
DBMS_REPCAT_RGT.DROP_ALL_TEMPLATE_SITES (  
    refresh_template_name  IN    VARCHAR2);
```

### パラメータ

表 23-42 DROP\_ALL\_TEMPLATE\_SITES プロシージャのパラメータ

パラメータ	説明
refresh_template_name	削除対象のサイトを持つ配置テンプレートの名前です。

### 例外

表 23-43 DROP\_ALL\_TEMPLATE\_SITES プロシージャの例外

例外	説明
miss_refresh_template	指定された配置テンプレート名が無効か、または存在しません。

## DROP\_ALL\_TEMPLATES プロシージャ

このプロシージャは、プロシージャがコールされたサイトで、すべての配置テンプレートを削除します。

**注意：** このプロシージャを実行すると、元に戻せません。実行するときには十分注意してください。

### 構文

```
DBMS_REPCAT_RGT.DROP_ALL_TEMPLATES;
```

### パラメータ

なし

## DROP\_ALL\_USER\_AUTHORIZATIONS プロシージャ

このプロシージャを実行すると、データベース管理者が、指定された配置テンプレートに対して与えられたすべてのユーザー認証を削除できます。このプロシージャを実行すると、DBA\_REPCAT\_USER\_AUTHORIZATIONS ビューから行が削除されます。

このプロシージャは、プライベート・テンプレートがパブリック・テンプレートに変換され、ユーザー認証がなくなっただ後で実装されることもあります。

### 構文

```
DBMS_REPCAT_RGT.DROP_ALL_USER_AUTHORIZATIONS (  
    refresh_template_name IN VARCHAR2);
```

### パラメータ

表 23-44 DROP\_ALL\_USER\_AUTHORIZATIONS プロシージャのパラメータ

パラメータ	説明
refresh_template_name	削除対象のユーザー認証を含む配置テンプレートの名前です。

### 例外

表 23-45 DROP\_ALL\_USER\_AUTHORIZATIONS プロシージャの例外

例外	説明
miss_refresh_template	指定された配置テンプレート名が無効か、または存在しません。

## DROP\_ALL\_USER\_PARM\_VALUES プロシージャ

このプロシージャは、指定された配置テンプレートのユーザー・パラメータ値を削除します。このプロシージャを使用すると、削除対象のユーザー・パラメータ値のセットを簡単に定義できます。

次の表に、パラメータとそれを定義した結果を示します。

パラメータ	パラメータを定義した結果
refresh_template_name	指定された配置テンプレートの、すべてのユーザー・パラメータが削除されます。
refresh_template_name および user_name	指定された配置テンプレートの、すべての指定されたユーザー・パラメータが削除されます。
refresh_template_name および parameter_name	指定された配置テンプレート・パラメータの、すべてのユーザー・パラメータ値が削除されます。
refresh_template_name、parameter_name および user_name:	指定された配置テンプレート・パラメータの、指定されたユーザーの値が削除されます (drop_user_parm の処理に相当します)。

### 構文

```
DBMS_REPCAT_RGT.DROP_ALL_USER_PARM_VALUES (  
    refresh_template_name  IN   VARCHAR2,  
    user_name              IN   VARCHAR2,  
    parameter_name         IN   VARCHAR2);
```

### パラメータ

表 23-46 DROP\_ALL\_USER\_PARM\_VALUES プロシージャのパラメータ

パラメータ	説明
refresh_template_name	削除対象のパラメータ値を持つ配置テンプレートの名前です。
user_name	パラメータ値が削除されるユーザーの名前です。
parameter_name	削除対象の値を持つテンプレート・パラメータです。

## 例外

**表 23-47 DROP\_ALL\_USER\_PARMS プロシージャの例外**

例外	説明
miss_refresh_template	指定された配置テンプレート名が無効か、または存在しません。
miss_user	指定されたユーザー名が無効か、または存在しません。
miss_user_parm_values	配置テンプレート、ユーザーおよびパラメータの組合せは、DBA_REPCAT_USER_PARM_VALUES ビューに存在しません。

## DROP\_REFRESH\_TEMPLATE プロシージャ

このプロシージャは、配置テンプレートを削除します。このプロシージャで配置テンプレートを削除すると、すべての関連するテンプレート・パラメータ、ユーザー認証、テンプレート・オブジェクトおよびユーザー・パラメータも削除されます。ただし、テンプレート・サイトは削除されません。

## 構文

```
DBMS_REPCAT_RGT.DROP_REFRESH_TEMPLATE (  
    refresh_template_name IN VARCHAR2);
```

## パラメータ

**表 23-48 DROP\_REFRESH\_TEMPLATE プロシージャのパラメータ**

パラメータ	説明
refresh_template_name	削除対象の配置テンプレートの名前です。

## 例外

**表 23-49 DROP\_REFRESH\_TEMPLATE プロシージャの例外**

例外	説明
miss_refresh_template	指定された配置テンプレート名が無効か、または存在しません。配置テンプレートを確認するには、DBA_REPCAT_REFRESH_TEMPLATES ビューの問合せを行います。

## DROP\_SITE\_INSTANTIATION プロシージャ

このプロシージャは、ターゲット・サイトでテンプレートのインスタンス化を削除します。このプロシージャは、マスター・サイトで関連するすべてのメタデータを削除します。また、指定されたサイトでのマテリアライズド・ビューのリフレッシュを使用禁止にします。

### 構文

```
DBMS_REPCAT_RGT.DROP_SITE_INSTANTIATION (  
    refresh_template_name  IN   VARCHAR2,  
    user_name              IN   VARCHAR2,  
    site_name              IN   VARCHAR2);
```

表 23-50 DROP\_SITE\_INSTANTIATION プロシージャのパラメータ

パラメータ	説明
refresh_template_name	削除対象のサイトを含むテンプレートの名前です。
user_name	リモート・マテリアライズド・ビュー・サイトでテンプレートを最初にインスタンス化したユーザーの名前です。テンプレートをインスタンス化したユーザーを確認するには、ALL_REPCAT_TEMPLATE_SITES ビューの問合せを行います。詳細は、25-11 ページの「ALL_REPCAT_TEMPLATE_SITES ビュー」を参照してください。
site_name	削除対象のテンプレート・サイトです。

### 例外

表 23-51 DROP\_SITE\_INSTANTIATION プロシージャの例外

例外	説明
miss_refresh_template	指定された配置テンプレート名が無効か、または存在しません。
miss_user	指定されたユーザー名は存在しません。
miss_template_site	ユーザーおよびサイト用に、配置テンプレートがインスタンス化されていません。



## DROP\_TEMPLATE\_OBJECT プロシージャ

このプロシージャは、指定された配置テンプレートからテンプレート・オブジェクトを削除します。たとえば、データベース管理者が古いマテリアライズド・ビューを配置テンプレートから削除するときに、このプロシージャを使用できます。配置テンプレートに加えられた変更は、配置テンプレートをインスタンス化する新しいサイトに反映されます。テンプレートをすでにインスタンス化したリモート・サイトで変更を適用するには、配置テンプレートを再びインスタンス化する必要があります。

### 構文

```
DBMS_REPCAT_RGT.DROP_TEMPLATE_OBJECT (  
    refresh_template_name  IN   VARCHAR2,  
    object_name            IN   VARCHAR2,  
    object_type            IN   VARCHAR2);
```

### パラメータ

表 23-52 DROP\_TEMPLATE\_OBJECT プロシージャのパラメータ

パラメータ	説明
refresh_template_name	オブジェクトが削除される配置テンプレートの名前です。
object_name	削除対象のテンプレート・オブジェクトの名前です。
object_type	削除対象のオブジェクトのタイプです。次のオブジェクト型を指定できます。 <div><div>SNAPSHOT INDEX TABLE VIEW SYNONYM SEQUENCE</div><div>PROCEDURE FUNCTION PACKAGE PACKAGE BODY TRIGGER DATABASE LINK</div></div>

### 例外

表 23-53 DROP\_TEMPLATE\_OBJECT プロシージャの例外

例外	説明
miss_refresh_template	指定された配置テンプレート名が無効か、または存在しません。
miss_template_object	指定されたテンプレート・オブジェクトが無効か、または存在しません。配置テンプレート・オブジェクトを確認するには、DBA_REPCAT_TEMPLATE_OBJECTS ビューの問合せを行います。

## DROP\_TEMPLATE\_PARM プロシージャ

このプロシージャは、DBA\_REPCAT\_TEMPLATE\_PARMS ビューから既存のテンプレート・パラメータを削除します。テンプレート・オブジェクトを削除して、特定のパラメータが不要になった場合、このプロシージャを使用すると便利です。

### 構文

```
DBMS_REPCAT_RGT.DROP_TEMPLATE_PARM (  
    refresh_template_name  IN   VARCHAR2,  
    parameter_name         IN   VARCHAR2);
```

### パラメータ

表 23-54 DROP\_TEMPLATE\_PARM プロシージャのパラメータ

パラメータ	説明
refresh_template_name	削除対象のパラメータを持つ配置テンプレート名です。
parameter_name	削除対象のパラメータの名前です。

### 例外

表 23-55 DROP\_TEMPLATE\_PARM プロシージャの例外

例外	説明
miss_refresh_template	指定された配置テンプレート名が無効か、または存在しません。
miss_template_parm	指定されたパラメータ名が無効か、または存在しません。テンプレート・パラメータを確認するには、DBA_REPCAT_TEMPLATE_PARMS ビューの問合せを行います。

## DROP\_USER\_AUTHORIZATION プロシージャ

このプロシージャは、DBA\_REPCAT\_USER\_AUTHORIZATIONS ビューからユーザー認証エントリを削除します。ユーザーのテンプレート認可を削除するときに、このプロシージャを使用します。認可が削除されたユーザーは、ターゲット配置テンプレートをインスタンス化できなくなります。

**関連項目：** 23-33 ページ「[DROP\\_ALL\\_USER\\_AUTHORIZATIONS プロシージャ](#)」

### 構文

```
DBMS_REPCAT_RGT.DROP_USER_AUTHORIZATION (  
    refresh_template_name  IN   VARCHAR2,  
    user_name              IN   VARCHAR2);
```

### パラメータ

表 23-56 DROP\_USER\_AUTHORIZATION プロシージャのパラメータ

パラメータ	説明
refresh_template_name	ユーザー認証が削除される配置テンプレートの名前です。
user_name	認可が削除されるユーザーの名前です。

### 例外

表 23-57 DROP\_USER\_AUTHORIZATION プロシージャの例外

例外	説明
miss_user	指定されたユーザー名が無効か、または存在しません。
miss_user_authorization	指定されたユーザーと配置テンプレートの組合せは存在しません。ユーザーや配置テンプレートに対する認可を確認するには、DBA_REPCAT_USER_AUTHORIZATIONS ビューの問合せを行います。
miss_refresh_template	指定された配置テンプレート名が無効か、または存在しません。

## DROP\_USER\_PARM\_VALUE プロシージャ

このプロシージャは、特定の配置テンプレートに対して事前定義されたユーザー・パラメータ値を削除します。このプロシージャは、ユーザーのテンプレート認可が削除された後で実行されることがよくあります。

### 構文

```
DBMS_REPCAT_RGT.DROP_USER_PARM_VALUE (  
    refresh_template_name    IN    VARCHAR2,  
    parameter_name          IN    VARCHAR2,  
    user_name                IN    VARCHAR2);
```

### パラメータ

表 23-58 DROP\_USER\_PARM\_VALUE プロシージャのパラメータ

パラメータ	説明
refresh_template_name	削除対象のパラメータ値を持つ配置テンプレート名です。
parameter_name	削除対象の事前定義値を持つパラメータの名前です。
user_name	パラメータ値が削除されるユーザーの名前です。

### 例外

表 23-59 DROP\_USER\_PARM\_VALUE プロシージャの例外

例外	説明
miss_refresh_template	指定された配置テンプレート名が無効か、または存在しません。
miss_user	指定されたユーザー名が無効か、または存在しません。
miss_user_parm_values	配置テンプレート、ユーザーおよびパラメータの組合せは、DBA_REPCAT_USER_PARM_VALUES ビューに存在しません。

## GET\_RUNTIME\_PARM\_ID ファンクション

このファンクションは、ランタイム・パラメータ値を定義する際に使用する ID を取得します。すべてのランタイム・パラメータ値はこの ID に割り当てられ、インスタンス化処理にも使用されます。

### 構文

```
DBMS_REPCAT_RGT.GET_RUNTIME_PARM_ID
RETURN NUMBER;
```

### パラメータ

なし

### 戻り値

**表 23-60 GET\_RUNTIME\_PARM\_ID ファンクションの戻り値**

戻り値	対応するデータ型
< システム生成番号 >	ランタイム・パラメータ値はシステム生成番号に割り当てられ、インスタンス化処理でも使用されます。

## INSERT\_RUNTIME\_PARMS プロシージャ

このプロシージャは、テンプレートのインスタンス化の前に、ランタイム・パラメータ値を定義します。ユーザー・パラメータ値が定義されていない場合で、デフォルトのパラメータ値を使用しないとき、このプロシージャでパラメータ値を定義します。

このプロシージャを使用する前に、必ず GET\_RUNTIME\_PARM\_ID ファンクションを実行して、ランタイム・パラメータを挿入する際に使用するパラメータ ID を取得します。この ID は、ランタイム・パラメータ値の定義および配置テンプレートのインスタンス化に使用します。

### 構文

```
DBMS_REPCAT_RGT.INSERT_RUNTIME_PARMS (
    runtime_parm_id    IN    NUMBER,
    parameter_name     IN    VARCHAR2,
    parameter_value    IN    CLOB);
```

パラメータ

表 23-61 INSERT\_RUNTIME\_PARMS プロシージャのパラメータ

パラメータ	説明
runtime_parm_id	GET_RUNTIME_PARM_ID ファンクションで取得した ID です。この ID は、配置テンプレートをインスタンス化する際にも使用します。1 つの配置テンプレートのすべてのパラメータ値には、必ず同じ ID を使用してください。
parameter_name	ランタイム・パラメータ値を定義するテンプレート・パラメータの名前です。テンプレート・パラメータを確認するには、DBA_REPCAT_TEMPLATE_PARMS ビューの問合せを行います。
parameter_value	配置テンプレートのインスタンス化処理で使用する、ランタイム・パラメータ値です。

例外

表 23-62 INSERT\_RUNTIME\_PARMS プロシージャの例外

例外	説明
miss_refresh_template	指定された配置テンプレート名が無効か、または存在しません。
miss_user	指定されたユーザー名が無効か、または存在しません。
miss_user_parm_values	配置テンプレート、ユーザーおよびパラメータの組合せは、DBA_REPCAT_USER_PARM_VALUES ビューに存在しません。

使用上の注意

このプロシージャでは CLOB が利用されています。このため、INSERT\_RUNTIME\_PARMS プロシージャを実行するときは、DBMS\_LOB パッケージを使用する必要があります。次の例は、INSERT\_RUNTIME\_PARMS プロシージャで DBMS\_LOB パッケージを使用する方法を示します。

```
DECLARE
    tempstring VARCHAR2(100);
    templob CLOB;
BEGIN
    DBMS_LOB.CREATETEMPORARY(templob, TRUE, DBMS_LOB.SESSION);
    tempstring := 'REGION 20';
    DBMS_LOB.WRITE(templob, length(tempstring), 1, tempstring);
    DBMS_REPCAT_RGT.INSERT_RUNTIME_PARMS(
        runtime_parm_id => 20,
        parameter_name => 'region',
        parameter_value => templob);
```

```
DBMS_LOB.FREETEMPORARY(templob);  
END;  
/
```

## INstantiate\_Offline ファンクション

このファンクションは、マテリアライズド・ビュー・サイトがマスターに接続していない（マテリアライズド・ビュー・サイトがオフラインの）状態で、リモート・マテリアライズド・ビュー・サイトにマテリアライズド・ビュー環境を構築する際に使用するスクリプトをマスター・サイトに生成します。生成されたスクリプトは、マスター・サイトに長時間接続できないリモート・マテリアライズド・ビュー・サイトで使用します。新しいマテリアライズド・ビューに設定されるデータの量によっては、リモート・マテリアライズド・ビュー・サイトでのインスタンス化処理に時間がかかります。このファンクションは、ユーザーのインスタンス化ごとに個別に実行する必要があります。

このファンクションで生成されたスクリプトは、USER\_REPCAT\_TEMP\_OUTPUT 一時ビューに格納され、配置テンプレートの配布時に、レプリケーション管理ツールなどの Oracle ツール製品で使います。このファンクションで戻される数値は、USER\_REPCAT\_TEMP\_OUTPUT 一時ビューから適切な情報を取り出す際に使います。

---

**注意：** このファンクションは、配置テンプレートのオフライン・インスタンス化に使用します。また、このファンクションは、他のユーザーのためにインスタンス化を行うレプリケーション管理者が使用します。独自のインスタンス化を行うユーザーは、パブリックの INstantiate\_Offline ファンクションを使用します。詳細は、23-43 ページの「[INstantiate\\_Offline 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を、[DBMS\\_OFFLINE\\_OG](#) パッケージのプロシージャ（マスター表のオフライン・インスタンス化に使用）や、[DBMS\\_OFFLINE\\_SNAPSHOT](#) パッケージのプロシージャ（マテリアライズド・ビューのオフライン・インスタンス化に使用）と混同しないでください。これらのプロシージャの使用の詳細は、該当するパッケージの説明を参照してください。

---

構文

```
DBMS_REPCAT_RGT.INSTANTIATE_OFFLINE(  
  refresh_template_name  IN   VARCHAR2,  
  site_name              IN   VARCHAR2,  
  user_name              IN   VARCHAR2  := NULL,  
  runtime_parm_id        IN   NUMBER    := -1e-130,  
  next_date              IN   DATE      := SYSDATE,  
  interval               IN   VARCHAR2  := 'SYSDATE + 1',  
  use_default_gowner     IN   BOOLEAN   := true)  
return NUMBER;
```

パラメータ

表 23-63 INstantiate\_Offline ファンクションのパラメータ

パラメータ	説明
refresh_template_name	インスタンス化される配置テンプレートの名前です。
site_name	配置テンプレートをインスタンス化している、リモート・サイトの名前です。
user_name	配置テンプレートをインスタンス化中の、許可を受けたユーザーの名前です。
runtime_parm_id	INSERT_RUNTIME_PARMS プロシージャを使用してランタイム・パラメータ値を定義した場合は、ランタイム・パラメータの作成時に使用した ID を指定します (ID は、GET_RUNTIME_PARM_ID ファンクションで取得済です)。
next_date	リフレッシュ・グループの作成時に使用する、次のリフレッシュ日の値を指定します。
interval	リフレッシュ・グループの作成時に使用する、リフレッシュ間隔を指定します。
use_default_gowner	値が TRUE の場合、作成されたマテリアライズド・ビュー・グループはデフォルトのユーザー PUBLIC が所有します。値が FALSE の場合、作成されたマテリアライズド・ビュー・グループはインスタンス化を実行するユーザーが所有します。



例外

表 23-64 INSTANTIATE\_OFFLINE ファンクションの例外

例外	説明
miss_refresh_template	指定された配置テンプレート名が無効か、または存在しません。
miss_user	許可を受けたユーザーの名前が無効か、または存在しません。指定されたユーザーが DBA_REPCAT_USER_AUTHORIZATIONS ビューにリストされているかどうかを確認してください。このビューにリストされていないユーザーは、ターゲット配置テンプレートをインスタンス化する許可を受けていません。

戻り値

表 23-65 INSTANTIATE\_OFFLINE ファンクションの戻り値

戻り値	説明
< システム生成番号 >	USER_REPCAT_TEMP_OUTPUT 一時ビューから選択して、生成されたインスタンス化スクリプトを取得するときの、output_id に対するシステム生成番号を示します。

INSTANTIATE\_ONLINE ファンクション

このファンクションは、マテリアライズド・ビュー・サイトがマスターに接続している（マテリアライズド・ビュー・サイトがオンラインの）状態で、リモート・マテリアライズド・ビュー・サイトにマテリアライズド・ビュー環境を構築する際に使用するスクリプトをマスター・サイトに生成します。生成されたスクリプトは、マスター・サイトに長時間接続できるリモート・マテリアライズド・ビュー・サイトで使用します。新しいマテリアライズド・ビューに設定されるデータの量によっては、リモート・マテリアライズド・ビュー・サイトでのインスタンス化処理に時間がかかります。このファンクションは、ユーザーのインスタンス化ごとに個別に実行する必要があります。

このファンクションで生成されたスクリプトは、USER\_REPCAT\_TEMP\_OUTPUT 一時ビューに格納され、配置テンプレートの配布時に、レプリケーション管理ツールなどの Oracle ツール製品で使用します。このファンクションで戻される数値は、USER\_REPCAT\_TEMP\_OUTPUT 一時ビューから適切な情報を取り出す際に使用します。

**注意：** このファンクションは、他のユーザーのためにインスタンス化を行うレプリケーション管理者が使用します。独自のインスタンス化を行うユーザーは、23-43 ページの「[INSTANTIATE\\_OFFLINE ファンクション](#)」で説明した、パブリックの INSTANTIATE\_OFFLINE ファンクションを使用します。

構文

```
DBMS_REPCAT_RGT.INSTANTIATE_ONLINE(  
  refresh_template_name  IN   VARCHAR2,  
  site_name              IN   VARCHAR2 := NULL,  
  user_name              IN   VARCHAR2 := NULL,  
  runtime_parm_id        IN   NUMBER   := -1e-130,  
  next_date              IN   DATE     := SYSDATE,  
  interval               IN   VARCHAR2 := 'SYSDATE + 1',  
  use_default_gowner     IN   BOOLEAN  := true)  
return NUMBER;
```

パラメータ

表 23-66 INstantiate\_Online ファンクションのパラメータ

パラメータ	説明
refresh_template_name	インスタンス化される配置テンプレートの名前です。
site_name	配置テンプレートをインスタンス化している、リモート・サイトの名前です。
user_name	配置テンプレートをインスタンス化中の、許可を受けたユーザーの名前です。
runtime_parm_id	INSERT_RUNTIME_PARMS プロシージャを使用してランタイム・パラメータ値を定義した場合は、ランタイム・パラメータの作成時に使用した ID を指定します (ID は、GET_RUNTIME_PARM_ID ファンクションで取得済です)。
next_date	リフレッシュ・グループの作成時に使用する、次のリフレッシュ日の値を指定します。
interval	リフレッシュ・グループの作成時に使用する、リフレッシュ間隔を指定します。
use_default_gowner	値が TRUE の場合、作成されたマテリアライズド・ビュー・グループはデフォルトのユーザー PUBLIC が所有します。値が FALSE の場合、作成されたマテリアライズド・ビュー・グループはインスタンス化を実行するユーザーが所有します。

例外

表 23-67 INSTANTIATE\_ONLINE ファンクションの例外

例外	説明
miss_refresh_template	指定された配置テンプレート名が無効か、または存在しません。
miss_user	許可を受けたユーザーの名前が無効か、または存在しません。指定されたユーザーが DBA_REPCAT_USER_AUTHORIZATIONS ビューにリストされているかどうかを確認してください。このビューにリストされていないユーザーは、ターゲット配置テンプレートをインスタンス化する許可を受けていません。
bad_parms	一部のテンプレート・パラメータが、定義されたユーザー・パラメータ値やテンプレートのデフォルト値で設定されませんでした。事前定義された値の数がテンプレート・パラメータの数と一致しなかったか、または事前定義された値がターゲット・パラメータでは無効（タイプの不一致）でした。

戻り値

表 23-68 INSTANTIATE\_ONLINE ファンクションの戻り値

戻り値	説明
< システム生成番号 >	USER_REPCAT_TEMP_OUTPUT 一時ビューから選択して、生成されたインスタンス化スクリプトを取得するときの、output_id に対するシステム生成番号を示します。

## LOCK\_TEMPLATE\_EXCLUSIVE プロシージャ

LOCK\_TEMPLATE\_EXCLUSIVE プロシージャを使用して、配置テンプレートの更新または変更が行われている間、テンプレートの読取りやインスタンス化ができないようにします。

ROLLBACK または COMMIT が実行されると、ロックが解除されます。

---

---

**注意：** 配置テンプレートに変更を加える前に、このプロシージャを実行する必要があります。

---

---

### 構文

```
DBMS_REPCAT_RGT.LOCK_TEMPLATE_EXCLUSIVE();
```

### パラメータ

なし

## LOCK\_TEMPLATE\_SHARED プロシージャ

指定された配置テンプレートを読取り専用にするには、LOCK\_TEMPLATE\_SHARED プロシージャを使用します。テンプレートをインスタンス化する前に、このプロシージャをコールする必要があります。このプロシージャを実行すると、配置テンプレートがインスタンス化されている間、他のユーザーはそのテンプレートを変更できません。

ROLLBACK または COMMIT が実行されると、ロックが解除されます。

### 構文

```
DBMS_REPCAT_RGT.LOCK_TEMPLATE_SHARED();
```

### パラメータ

なし

DBMS\_REPUTIL には表複製のためのシャドウ表、トリガー、およびパッケージを生成するサブプログラムや、スタンドアロン・プロシージャの起動およびパッケージ化されたプロシージャの起動をレプリケートするラッパーを生成するサブプログラムが含まれています。このパッケージを参照するのは、生成されたコードのみです。

この章では、次の項目を説明します。

- [DBMS\\_REPUTIL サブプログラムの概要](#)

## DBMS\_REPUTIL サブプログラムの概要

表 24-1 DBMS\_REPUTIL パッケージのサブプログラム

サブプログラム	説明
24-3 ページ 「REPLICATION_OFF プロシージャ」	レプリケーション環境内の他のサイトに変更内容をレプリケートせずに表を変更するか、またはプロシージャ・レプリケーションを使用する際に行レベル・レプリケーションを使用禁止にします。
24-3 ページ 「REPLICATION_ON プロシージャ」	レプリケーションが一時的に中断された後で、変更内容のレプリケーションを再び有効にします。
24-3 ページ 「REPLICATION_IS_ON ファンクション」	レプリケーションが実行されているかどうかを判定します。
24-4 ページ 「FROM_REMOTE ファンクション」	内部レプリケーション・パッケージ内のプロシージャの開始で TRUE を、これらのプロシージャの終了で FALSE を戻します。
24-4 ページ 「GLOBAL_NAME ファンクション」	ローカル・データベースのグローバル・データベース名を確認します (戻り値がグローバル名です)。
24-5 ページ 「MAKE_INTERNAL_PKG プロシージャ」	内部パッケージとレプリケーション・カタログ内の表を同期させます。 <b>注意：</b> オラクル社カスタマ・サポート・センターから指示された場合を除き、このプロシージャを実行しないでください。
24-6 ページ 「SYNC_UP_REP プロシージャ」	内部トリガーとレプリケーション・カタログ内の表またはマテリアライズド・ビューを同期させます。 <b>注意：</b> オラクル社カスタマ・サポート・センターから指示された場合を除き、このプロシージャを実行しないでください。

## REPLICATION\_OFF プロシージャ

このプロシージャでは、レプリケーション環境内の他のサイトに対する変更をレプリケートせずに表を変更できます。またプロシージャ・レプリケーションを使用する際、行レベルのレプリケーションを使用禁止にします。通常は、このフラグを設定する前に、レプリケーション環境内のすべてのマスター・グループに対するレプリケーション・アクティビティを停止します。

### 構文

```
DBMS_REPUTIL.REPLICATION_OFF();
```

### パラメータ

なし

## REPLICATION\_ON プロシージャ

このプロシージャは、レプリケーションが一時的に中断された後で、変更内容のレプリケーションを再び有効にします。

### 構文

```
DBMS_REPUTIL.REPLICATION_ON();
```

### パラメータ

なし

## REPLICATION\_IS\_ON ファンクション

このファンクションは、レプリケーションが実行されているかどうかを判定します。戻り値が TRUE の場合は、生成されたレプリケーション・トリガーが有効になっています。FALSE の場合は、レプリケーション・グループのカレント・サイトで、レプリケーションが使用禁止になっています。

このファンクションの戻り値は、DBMS\_REPUTIL パッケージの REPLICATION\_ON プロシージャまたは REPLICATION\_OFF プロシージャのコールで設定されます。

### 構文

```
DBMS_REPUTIL.REPLICATION_IS_ON()  
return BOOLEAN;
```

### パラメータ

なし

### FROM\_REMOTE ファンクション

このファンクションは、内部レプリケーション・パッケージ内のプロシージャの開始で **TRUE** を、これらのプロシージャの終了で **FALSE** を戻します。内部パッケージによる更新の結果として発生するトリガーがある場合は、このファンクションをチェックする必要があります。

#### 構文

```
DBMS_REPUTIL.FROM_REMOTE()  
    return BOOLEAN;
```

#### パラメータ

なし

### GLOBAL\_NAME ファンクション

このファンクションは、ローカル・データベースのグローバル・データベース名を判定します（戻り値がグローバル名です）。

#### 構文

```
DBMS_REPUTIL.GLOBAL_NAME()  
    return VARCHAR2;
```

#### パラメータ

なし



## MAKE\_INTERNAL\_PKG プロシージャ

このプロシージャは、内部パッケージの有無を、レプリケーション・カタログ内の表またはマテリアライズド・ビューと同期させます。表にレプリケーション・サポートがある場合は、このプロシージャを実行して内部パッケージを作成します。レプリケーション・サポートがない場合、このプロシージャは関連する内部パッケージを破棄します。このプロシージャは、ネストした表の記憶表は受け付けません。

**注意：** オラクル社カスタマ・サポート・センターから指示された場合を除き、このプロシージャを実行しないでください。

### 構文

```
DBMS_REPUTIL.MAKE_INTERNAL_PKG (  
    canon_sname      IN   VARCHAR2,  
    canon_onsame      IN   VARCHAR2);
```

### パラメータ

表 24-2 MAKE\_INTERNAL\_PKG プロシージャのパラメータ

パラメータ	説明
canon_sname	同期する表が含まれるスキーマです。  このパラメータ値は、正確に定義する必要があります。オブジェクトと同じ大文字と小文字の区別を使用し、二重引用符では囲みません。
canon_onsame	同期する表の名前です。  このパラメータ値は、正確に定義する必要があります。オブジェクトと同じ大文字と小文字の区別を使用し、二重引用符では囲みません。

## SYNC\_UP\_REP プロシージャ

このプロシージャは、内部トリガーの有無を、レプリケーション・カタログ内の表またはマテリアライズド・ビューと同期させます。表またはマテリアライズド・ビューにレプリケーション・サポートがある場合は、このプロシージャを実行して内部レプリケーション・トリガーを作成します。レプリケーション・サポートがない場合、このプロシージャは関連する内部トリガーを破棄します。このプロシージャは、ネストした表の記憶表は受け付けません。

**注意：** オラクル社カスタマ・サポート・センターから指示された場合を除き、このプロシージャを実行しないでください。

### 構文

```
DBMS_REPUTIL.SYNC_UP_REP (  
    canon_sname      IN   VARCHAR2,  
    canon_onsame      IN   VARCHAR2);
```

### パラメータ

表 24-3 SYNC\_UP\_REP プロシージャのパラメータ

パラメータ	説明
canon_sname	同期する表またはマテリアライズド・ビューが含まれるスキーマです。  このパラメータ値は、正確に定義する必要があります。オブジェクトと同じ大文字と小文字の区別を使用し、二重引用符では囲みません。
canon_onsame	同期する表またはマテリアライズド・ビューの名前です。  このパラメータ値は、正確に定義する必要があります。オブジェクトと同じ大文字と小文字の区別を使用し、二重引用符では囲みません。

# 第 IV 部

---

## レプリケーション・データ・ ディクショナリ・リファレンス

第 IV 部では、レプリケーション環境の情報を提供するデータ・ディクショナリ・ビューについて説明します。

第 IV 部は次の章で構成されています。

- [第 25 章「レプリケーション・カタログ・ビュー」](#)
- [第 26 章「レプリケーション動的パフォーマンス・ビュー」](#)
- [第 27 章「遅延トランザクション・ビュー」](#)
- [第 28 章「マテリアライズド・ビューとリフレッシュ・グループ・ビュー」](#)



---

## レプリケーション・カタログ・ビュー

---

レプリケーション機能をサイトにインストールすると、同じサイトにレプリケーション・カタログが自動的にインストールされます。このレプリケーション・カタログには、表とビューが含まれています。この章では、次の項目を説明します。

- [レプリケーション・カタログ・ビューの概要](#)

---

**注意：** レプリケーション・カタログの表は直接変更しないでください。変更が必要な場合は、DBMS\_REPCAT パッケージに用意されているプロシージャを使用してください。

---

**関連項目：** [第 10 章「レプリケーション環境の監視」](#)

## レプリケーション・カタログ・ビューの概要

多くのデータ・ディクショナリ表には、対応するビューが3種類あります。

- `ALL_xxx` ビューには、カレント・ユーザーのスキーマの情報など、カレント・ユーザーからアクセス可能な情報がすべて表示されます。ユーザーが他のスキーマのオブジェクトにアクセスする権限またはロールを付与されている場合には、それらのオブジェクトの情報も表示されます。
- `DBA_xxx` ビューには、データベース全体の関連情報がすべて表示されます。`DBA_xxx` ビューは、管理者のみを対象としています。このビューには、`SELECT ANY TABLE` 権限を持つユーザーのみがアクセスできます。この権限は、Oracle の初期インストール時にデータベース管理者 (DBA) のロールに割り当てられます。
- `USER_xxx` ビューには、カレント・ユーザーのスキーマの情報がすべて表示されます。`USER_xxx` ビューの問合せを行うために特別な権限は必要ありません。

1 つのデータ・ディクショナリ表に対応する `ALL_xxx`、`DBA_xxx` および `USER_xxx` ビューの列は、通常ほぼ同一です。このため、詳細は、`ALL_xxx` ビューの項でまとめて説明します。`DBA_xxx` ビューと `USER_xxx` ビューの詳細は説明しませんが、`ALL_xxx` ビューとの相違点があれば記述しています。

25-3 ページの図 25-1 に示すように、レプリケーション・カタログ・ビューは、レプリケートされるオブジェクトのタイプ、レプリケート先、レプリケーションの実行中にエラーが発生していないかどうかなどを判定するためにマスター・サイトやマテリアライズド・ビュー・サイトで使用されます。25-4 ページの表 25-1 は、すべてのレプリケーション・カタログ・ビューを一覧で示しています。

図 25-1 レプリケーション・カタログ・ビューとレプリケート・オブジェクト

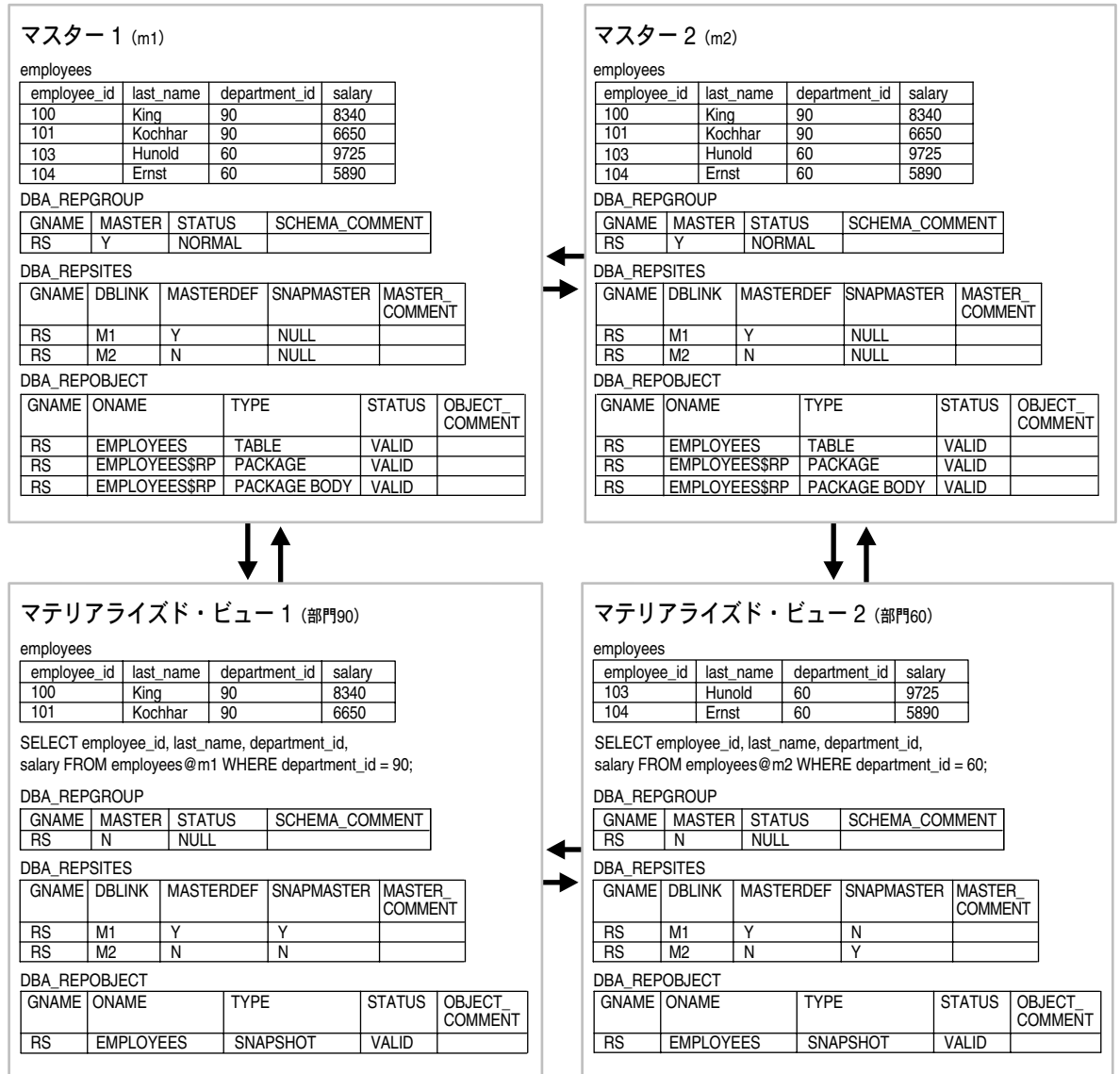


表 25-1 レプリケーション・カタログ・ビュー

ALL_ ビュー	DBA_ ビュー	USER_ ビュー
-	DBA_REGISTERED_MVIEW_GROUPS	-
ALL_REPCAT_REFRESH_TEMPLATES	DBA_REPCAT_REFRESH_TEMPLATES	USER_REPCAT_REFRESH_TEMPLATES
ALL_REPCAT_TEMPLATE_OBJECTS	DBA_REPCAT_TEMPLATE_OBJECTS	USER_REPCAT_TEMPLATE_OBJECTS
ALL_REPCAT_TEMPLATE_PARS	DBA_REPCAT_TEMPLATE_PARS	USER_REPCAT_TEMPLATE_PARS
ALL_REPCAT_TEMPLATE_SITES	DBA_REPCAT_TEMPLATE_SITES	USER_REPCAT_TEMPLATE_SITES
ALL_REPCAT_USER_AUTHORIZATIONS	DBA_REPCAT_USER_AUTHORIZATIONS	USER_REPCAT_USER_AUTHORIZATION
ALL_REPCAT_USER_PARM_VALUES	DBA_REPCAT_USER_PARM_VALUES	USER_REPCAT_USER_PARM_VALUES
ALL_REPCATLOG	DBA_REPCATLOG	USER_REPCATLOG
ALL_REPCOLUMN	DBA_REPCOLUMN	USER_REPCOLUMN
ALL_REPCOLUMN_GROUP	DBA_REPCOLUMN_GROUP	USER_REPCOLUMN_GROUP
ALL_REPCONFLICT	DBA_REPCONFLICT	USER_REPCONFLICT
ALL_REPDDL	DBA_REPDDL	USER_REPDDL
-	DBA_REPEXTENSIONS	-
ALL_REPGENOBJECTS	DBA_REPGENOBJECTS	USER_REPGENOBJECTS
ALL_REPGROUP	DBA_REPGROUP	USER_REPGROUP
ALL_REPGROUP_PRIVILEGES	DBA_REPGROUP_PRIVILEGES	USER_REPGROUP_PRIVILEGES
ALL_REPGROUPED_COLUMN	DBA_REPGROUPED_COLUMN	USER_REPGROUPED_COLUMN
ALL_REPKEY_COLUMNS	DBA_REPKEY_COLUMNS	USER_REPKEY_COLUMNS
ALL_REPOBJECT	DBA_REPOBJECT	USER_REPOBJECT
ALL_REPPARAMETER_COLUMN	DBA_REPPARAMETER_COLUMN	USER_REPPARAMETER_COLUMN
ALL_REPPRIORITY	DBA_REPPRIORITY	USER_REPPRIORITY
ALL_REPPRIORITY_GROUP	DBA_REPPRIORITY_GROUP	USER_REPPRIORITY_GROUP
ALL_REPPROP	DBA_REPPROP	USER_REPPROP
ALL_REPRESOL_STATS_CONTROL	DBA_REPRESOL_STATS_CONTROL	USER_REPRESOL_STATS_CONTROL
ALL_REPRESOLUTION	DBA_REPRESOLUTION	USER_REPRESOLUTION
ALL_REPRESOLUTION_METHOD	DBA_REPRESOLUTION_METHOD	USER_REPRESOLUTION_METHOD



表 25-1 レプリケーション・カタログ・ビュー（続き）

ALL_ ビュー	DBA_ ビュー	USER_ ビュー
<a href="#">ALL_REPRESOLUTION_STATISTICS</a>	<a href="#">DBA_REPRESOLUTION_STATISTICS</a>	<a href="#">USER_REPRESOLUTION_STATISTICS</a>
<a href="#">ALL_REPSITES</a>	<a href="#">DBA_REPSITES</a>	<a href="#">USER_REPSITES</a>
-	<a href="#">DBA_REPSITES_NEW</a>	-

## DBA\_REGISTERED\_MVIEW\_GROUPS

DBA\_REGISTERED\_MVIEW\_GROUPS には、マスター・サイトまたはマスター・マテリアライズド・ビュー・サイトのすべての登録済マテリアライズド・ビュー・グループがリストされます。

列	データ型	NULL	説明
NAME	VARCHAR2 (30)	-	マテリアライズド・ビューのレプリケーション・グループの名前。
MVIEW_SITE	VARCHAR2 (128)	-	マテリアライズド・ビューのレプリケーション・グループのサイト。
GROUP_COMMENT	VARCHAR2 (80)	-	マテリアライズド・ビューのレプリケーション・グループの説明。
VERSION	VARCHAR2 (8)	-	マテリアライズド・ビューのレプリケーション・グループの Oracle バージョン。 <b>注意：</b> Oracle9i のマテリアライズド・ビュー・グループには Oracle8 と表示されます。
FNAME	VARCHAR2 (30)	-	マテリアライズド・ビュー・グループの固有の名前。
OWNER	VARCHAR2 (30)	-	マテリアライズド・ビューのレプリケーション・グループの所有者。

## ALL\_REPCAT\_REFRESH\_TEMPLATES

カレント・ユーザーからアクセス可能な、各配置テンプレートに関するグローバル情報が示されます。テンプレート名、テンプレートの所有者、テンプレート・オブジェクトが属するリフレッシュ・グループ、テンプレートの種類（プライベートまたはパブリック）などの情報があります。

DBA は、テンプレートのコンテナにマテリアライズド・ビュー定義を追加するときに、該当する REFRESH\_TEMPLATE\_NAME を参照します。特定のテンプレートに追加されたマテリアライズド・ビューは、REFRESH\_GROUP\_NAME で指定されたリフレッシュ・グループに追加されます。

パブリックとして作成された配置テンプレートは、マスター・サイトへの接続を許可されているユーザーであれば、どのユーザーも使用できます。一方、プライベートとして作成された配置テンプレートは、ALL\_REPCAT\_USER\_AUTHORIZATIONS ビューにリストされるユーザーのみが使用できます。

### 関連するビュー：

- DBA\_REPCAT\_REFRESH\_TEMPLATES には、データベース内のすべての配置テンプレートが示されます。
- USER\_REPCAT\_REFRESH\_TEMPLATES には、カレント・ユーザーが所有するすべての配置テンプレートが示されます。

列	データ型	NULL	説明
REFRESH_TEMPLATE_NAME	VARCHAR2 (30)	-	配置テンプレートの名前。
OWNER	VARCHAR2 (30)	-	配置テンプレートの所有者。
REFRESH_GROUP_NAME	VARCHAR2 (30)	-	インスタンス化処理中にテンプレート・オブジェクトが追加されるリフレッシュ・グループの名前。
TEMPLATE_COMMENT	VARCHAR2 (2000)	-	ユーザーが入力するコメント。
PUBLIC_TEMPLATE	VARCHAR2 (1)	-	Y の場合、配置テンプレートはパブリックです。 N の場合、配置テンプレートはプライベートです。

## ALL\_REPCAT\_TEMPLATE\_OBJECTS

カレント・ユーザーからアクセス可能な、各配置テンプレートの各オブジェクト定義が示されます。REFRESH\_TEMPLATE\_NAME にターゲット・テンプレートを指定すると、各オブジェクトがテンプレートに追加されます。

DDL\_TEXT には、テンプレートをパラメータ化するための変数を含めることができます。変数を作成するには、変数名の先頭にコロン (:) を入力します (:region など)。テンプレートでパラメータを使用すると、テンプレートのインスタンス化処理を柔軟に行うことができます (マテリアライズド・ビュー・サイト固有のデータ・セットを定義する場合など)。

オブジェクトがテンプレートに追加されると、指定された DDL が検索されます。定義済パラメータが検索されると、それらのパラメータは、自動的に ALL\_REPCAT\_TEMPLATE\_PARMS ビューに追加されます。

### 関連するビュー：

- DBA\_REPCAT\_TEMPLATE\_OBJECTS には、データベース内のすべての配置テンプレートのオブジェクト定義が示されます。
- USER\_REPCAT\_TEMPLATE\_OBJECTS には、カレント・ユーザーが所有する各配置テンプレートのオブジェクト定義が示されます。

列	データ型	NULL	説明
REFRESH_TEMPLATE_NAME	VARCHAR2 (30)	NOT NULL	配置テンプレートの名前。
OBJECT_NAME	VARCHAR2 (30)	NOT NULL	配置テンプレートのオブジェクト名。
OBJECT_TYPE	VARCHAR2 (17)	-	配置テンプレートのオブジェクト型。 FUNCTION                SNAPSHOT INDEX                    SYNONYM INDEXTYPE                TABLE OPERATOR                 TRIGGER PACKAGE                  TYPE PACKAGE BODY             TYPE BODY PROCEDURE                VIEW
DDL_NUM	NUMBER	NOT NULL	複数の DDL 文を使用してオブジェクトを作成する場合に、DDL_TEXT 列に格納されている DDL 文の実行順序を示します。
DDL_TEXT	CLOB (4000)	-	配置テンプレートのオブジェクトを作成するために実行される DDL。
MASTER_ROLLBACK_SEGMENT	VARCHAR2 (30)	-	配置テンプレートのオブジェクトをインスタンス化するとき使用されるロールバック・セグメントの名前。

列	データ型	NULL	説明
DERIVED_FROM_SNAME	VARCHAR2 (30)	-	テンプレート・オブジェクトの基になったオブジェクトを含むスキーマが表示されます（該当する値がある場合）。
DERIVED_FROM_ONAME	VARCHAR2 (30)	-	テンプレート・オブジェクトの基になったオブジェクトの名前が表示されます（該当する値がある場合）。
FLAVOR_ID	NUMBER	-	配置テンプレートのオブジェクトの固有 ID。

DDL\_TEXT 列は CLOB として定義されているため、ALL\_REPCAT\_TEMPLATE\_OBJECTS ビューに対して単純に SELECT 文を実行するとエラーが戻されます。オブジェクト DDL を参照する必要がない場合は、次の SELECT 文を使用してください（DDL\_TEXT パラメータは除外してください）。

```
SELECT REFRESH_TEMPLATE_NAME, OBJECT_NAME, OBJECT_TYPE, MASTER_ROLLBACK_SEG, FLAVOR_ID
FROM DBA_REPCAT_TEMPLATE_OBJECTS;
```

次に、カーソルと DBMS\_LOB パッケージを使用し、ALL\_REPCAT\_TEMPLATE\_OBJECTS ビューの内容をすべて表示するためのスクリプトを示します。DDL\_TEXT 列など、ALL\_REPCAT\_TEMPLATE\_OBJECTS ビューの内容をすべて表示するには、次のスクリプトを使用します。

```
SET SERVEROUTPUT ON

DECLARE
  CURSOR mycursor IS
    SELECT REFRESH_TEMPLATE_NAME, OBJECT_NAME, OBJECT_TYPE, DDL_TEXT,
           MASTER_ROLLBACK_SEG, FLAVOR_ID
    FROM DBA_REPCAT_TEMPLATE_OBJECTS;
  tempstring VARCHAR2(1000);
  len NUMBER;
BEGIN
  FOR myrec IN mycursor LOOP
    len := DBMS_LOB.GETLENGTH(myrec.ddl_text);
    DBMS_LOB.READ(myrec.ddl_text, len, 1, tempstring);
    DBMS_OUTPUT.PUT_LINE(myrec.refresh_template_name||' '||
                          myrec.object_name||' '||myrec.object_type||' '||tempstring||' '||
                          myrec.master_rollback_seg||' '||myrec.flavor_id);
  END LOOP;
END;
/
```

**関連項目：** カーソルの使用方法の詳細は、『Oracle9i アプリケーション開発者ガイド - 基礎編』を参照してください。DBMS\_LOB パッケージと LOB の一般的な使用方法の詳細は、『Oracle9i アプリケーション開発者ガイド ラージ・オブジェクト』を参照してください。

## ALL\_REPCAT\_TEMPLATE\_PARMS

カレント・ユーザーからアクセス可能な、すべてのテンプレートのオブジェクト DDL で定義されたパラメータが含まれます。オブジェクトがテンプレートに追加されると、DDL に変数があるかどうかを検査されます。パラメータが見つかり、自動的にこのビューに追加されます。

このビューでは、デフォルトのパラメータ値およびプロンプト文字列も定義できます。これらの値を定義すると、インスタンス化処理でテンプレートが使用しやすくなります。

**関連項目：** 25-7 ページ「[ALL\\_REPCAT\\_TEMPLATE\\_OBJECTS](#)」

### 関連するビュー：

- DBA\_REPCAT\_TEMPLATE\_PARMS には、データベース内のすべての配置テンプレートのテンプレート・パラメータが表示されます。
- USER\_REPCAT\_TEMPLATE\_PARMS には、カレント・ユーザーが所有するすべての配置テンプレートのテンプレート・パラメータが表示されます。

列	データ型	NULL	説明
REFRESH_TEMPLATE_NAME	VARCHAR2 (30)	NOT NULL	配置テンプレートの名前。
OWNER	VARCHAR2 (30)	NOT NULL	配置テンプレートの所有者。
REFRESH_GROUP_NAME	VARCHAR2 (30)	NOT NULL	インスタンス化処理中にテンプレート・オブジェクトが追加されるリフレッシュ・グループの名前。
TEMPLATE_COMMENTS	VARCHAR2 (2000)	-	ユーザーが入力するコメント。
PUBLIC_TEMPLATE	VARCHAR2 (1)	-	Y の場合、配置テンプレートはパブリックです。 N の場合、配置テンプレートはプライベートです。
PARAMETER_NAME	VARCHAR2 (30)	NOT NULL	パラメータ名。
DEFAULT_PARM_VALUE	CLOB (4000)	-	デフォルトのパラメータ値。
PROMPT_STRING	VARCHAR2 (2000)	-	パラメータのプロンプト文字列。
USER_OVERRIDE	VARCHAR2 (1)	-	Y の場合、ユーザーによるデフォルトのパラメータ値の上書きが可能です。 N の場合、ユーザーによるデフォルトのパラメータ値の上書きが不可能です。

DEFAULT\_PARM\_VALUE 列は CLOB として定義されているため、ALL\_REPCAT\_TEMPLATE\_PARAMS ビューに対して単純に SELECT 文を実行するとエラーが戻されます。デフォルトのパラメータ値を参照する必要がある場合は、次の SELECT 文を使用してください (DEFAULT\_PARM\_VALUE は除外してください)。

```
SELECT REFRESH_TEMPLATE_NAME, OWNER, REFRESH_GROUP_NAME, TEMPLATE_COMMENT,
       PUBLIC_TEMPLATE, PARAMETER_NAME, PROMPT_STRING, USER_OVERRIDE
FROM DBA_REPCAT_TEMPLATE_PARAMS;
```

次に、カーソルと DBMS\_LOB パッケージを使用し、ALL\_REPCAT\_TEMPLATE\_PARAMS ビューの内容をすべて表示するためのスクリプトを示します。DEFAULT\_PARM\_VALUE 列など、ALL\_REPCAT\_TEMPLATE\_PARAMS ビューの内容をすべて表示するには、次のスクリプトを使用します。

```
SET SERVEROUTPUT ON

DECLARE
  CURSOR mycursor IS
    SELECT REFRESH_TEMPLATE_NAME, OWNER, REFRESH_GROUP_NAME,
           TEMPLATE_COMMENT, PUBLIC_TEMPLATE, PARAMETER_NAME, DEFAULT_PARM_VALUE,
           PROMPT_STRING, USER_OVERRIDE
    FROM DBA_REPCAT_TEMPLATE_PARAMS;
  tempstring VARCHAR2(1000);
  len NUMBER;
BEGIN
  FOR myrec IN mycursor LOOP
    len := DBMS_LOB.GETLENGTH(myrec.default_parm_value);
    DBMS_LOB.READ(myrec.default_parm_value, len, 1, tempstring);
    DBMS_OUTPUT.PUT_LINE(myrec.refresh_template_name||' '||
      myrec.owner||' '||myrec.refresh_group_name||' '||
      myrec.template_comment||' '||myrec.public_template||' '||
      myrec.parameter_name||' '||tempstring||' '||myrec.prompt_string||' '||
      myrec.user_override);
  END LOOP;
END;
/
```

**関連項目：** カーソルの使用方法の詳細は、『Oracle9i アプリケーション開発者ガイド - 基礎編』を参照してください。DBMS\_LOB パッケージと LOB の一般的な使用方法の詳細は、『Oracle9i アプリケーション開発者ガイド ラージ・オブジェクト』を参照してください。

## ALL\_REPCAT\_TEMPLATE\_SITES

エンタープライズ・ネットワークにおける、テンプレートのインスタンス化のカレント・ステータス情報が示されます。カレント・ユーザーからアクセス可能な、配置テンプレートのインスタンス化サイトの情報も含まれます。具体的には、DBA は、特定のサイトで実行されるテンプレートのインストールや削除を監視できます。

### 関連するビュー：

- DBA\_REPCAT\_TEMPLATE\_SITES には、データベース内のすべてのテンプレートのリモート・インスタンス化サイトがすべて示されます。
- USER\_REPCAT\_TEMPLATE\_SITES には、カレント・ユーザーが所有するすべてのテンプレートのリモート・インスタンス化サイトが示されます。

列	データ型	NULL	説明
REFRESH_TEMPLATE_NAME	VARCHAR2 (30)	NOT NULL	配置テンプレートの名前。
REFRESH_GROUP_NAME	VARCHAR2 (30)	-	インスタンス化処理中にテンプレート・オブジェクトが追加されるリフレッシュ・グループの名前。
TEMPLATE_OWNER	VARCHAR2 (30)	-	配置テンプレートの所有者とみなされるユーザーの名前。
USER_NAME	VARCHAR2 (30)	NOT NULL	配置テンプレートをインスタンス化したユーザーの名前。
SITE_NAME	VARCHAR2 (128)	-	配置テンプレートのターゲット・マテリアライズド・ビュー・サイト。
REPAPI_SITE_NAME	VARCHAR2 (128)	-	この列は、Oracle の将来のリリースで使用される予定です。
STATUS	VARCHAR2 (10)	-	ターゲット・マテリアライズド・ビュー・サイトにおける配置テンプレートのステータスが表示されます。 0 = インストールされていません。 1 = インストール済です。 -1 = インストールされていますがエラーが発生しています。
INSTANTIATION_DATE	DATE	-	テンプレートがインスタンス化された日付が表示されます。テンプレートがまだインスタンス化されていない場合、このフィールドは NULL になります。

## ALL\_REPCAT\_USER\_AUTHORIZATIONS

カレント・ユーザーからアクセス可能なプライベート配置テンプレートに対する権限を付与されたユーザーが表示されます。このビューにリストされたユーザーは、指定されたテンプレートをインスタンス化できます。このビューに表示されないユーザーは配置テンプレートをインスタンス化できません。

### 関連するビュー：

- DBA\_REPCAT\_USER\_AUTHORIZATIONS には、データベース内のすべてのプライベート配置テンプレートに対する権限を付与されたユーザーがリストされます。
- USER\_REPCAT\_USER\_AUTHORIZATIONS には、カレント・ユーザーが所有するプライベート配置テンプレートに対する権限を付与されたユーザーがリストされます。

列	データ型	NULL	説明
REFRESH_TEMPLATE_NAME	VARCHAR2 (30)	NOT NULL	ユーザーがインスタンス化する権限を付与された配置テンプレートの名前。
OWNER	VARCHAR2 (30)	NOT NULL	配置テンプレートの所有者の名前。
REFRESH_GROUP_NAME	VARCHAR2 (30)	NOT NULL	インスタンス化処理中にテンプレート・オブジェクトが追加されるリフレッシュ・グループの名前。
TEMPLATE_COMMENT	VARCHAR2 (2000)	-	ユーザーが入力するコメント。
PUBLIC_TEMPLATE	VARCHAR2 (1)	-	Y の場合、配置テンプレートはパブリックです。 N の場合、配置テンプレートはプライベートです。
USER_NAME	VARCHAR2 (30)	NOT NULL	配置テンプレートをインスタンス化する権限を付与されたユーザーの名前。



## ALL\_REPCAT\_USER\_PARM\_VALUES

カレント・ユーザーからアクセス可能なすべての配置テンプレートのテンプレート・パラメータが示されます。DBA には、テンプレートを配布してインスタンス化する前に、ユーザー・パラメータの表を作成するオプションの機能があります。指定されたユーザーがテンプレートをインスタンス化する場合は、そのユーザー用に ALL\_REPCAT\_USER\_PARM\_VALUES ビューに格納された値が自動的に使用されます。

### 関連するビュー：

- DBA\_REPCAT\_USER\_PARM\_VALUES には、データベース内のすべての配置テンプレートのテンプレート・パラメータが示されます。
- USER\_REPCAT\_USER\_PARM\_VALUES には、カレント・ユーザーが所有するすべての配置テンプレートのテンプレート・パラメータが示されます。

列	データ型	NULL	説明
REFRESH_TEMPLATE_NAME	VARCHAR2 (30)	NOT NULL	ユーザー・パラメータ値が定義された配置テンプレートの名前。
OWNER	VARCHAR2 (30)	NOT NULL	配置テンプレートの所有者の名前。
REFRESH_GROUP_NAME	VARCHAR2 (30)	NOT NULL	インスタンス化処理中にテンプレート・オブジェクトが追加されるリフレッシュ・グループの名前。
TEMPATE_COMMENT	VARCHAR2 (2000)	-	ユーザーが入力するコメント。
PUBLIC_TEMPLATE	VARCHAR2 (1)	-	Y の場合、配置テンプレートはパブリックです。 N の場合、配置テンプレートはプライベートです。
PARAMETER_NAME	VARCHAR2 (30)	NOT NULL	ユーザー・パラメータ値が定義されたパラメータの名前。
DEFAULT_PARM_VALUE	CLOB (4000)	-	パラメータのデフォルト値。
PROMPT_STRING	VARCHAR2 (2000)	-	パラメータのプロンプト文字列。
PARM_VALUE	CLOB (4000)	-	指定されたユーザーに対して定義されたパラメータ値。
USER_NAME	VARCHAR2 (30)	NOT NULL	指定されたパラメータ値が定義されたユーザーの名前。

DEFAULT\_PARM\_VALUE および PARM\_VALUE は CLOB として定義されているため、ALL\_REPCAT\_USER\_PARM\_VALUES ビューに対して単純に SELECT 文を実行するとエラーが戻されます。デフォルトのパラメータ値やユーザー・パラメータ値を参照する必要がない場合は、次の SELECT 文を使用してください（DEFAULT\_PARM\_VALUE および PARM\_VALUE は除外してください）。

```
SELECT REFRESH_TEMPLATE_NAME, OWNER, REFRESH_GROUP_NAME, TEMPLATE_COMMENT,  
       PUBLIC_TEMPLATE, PARAMETER_NAME, PROMPT_STRING, USER_NAME  
FROM DBA_REPCAT_USER_PARM_VALUES;
```

次に、カーソルと DBMS\_LOB パッケージを使用し、ALL\_REPCAT\_USER\_PARM\_VALUES ビューの内容をすべて表示するためのスクリプトを示します。DEFAULT\_PARM\_VALUE や PARM\_VALUE 列など、ALL\_REPCAT\_TEMPLATE\_PARAMS ビューの内容をすべて表示するには、次のスクリプトを使用します。

```
SET SERVEROUTPUT ON
```

```
DECLARE  
  CURSOR mycursor IS  
    SELECT REFRESH_TEMPLATE_NAME, OWNER, REFRESH_GROUP_NAME,  
           TEMPLATE_COMMENT, PUBLIC_TEMPLATE, PARAMETER_NAME, DEFAULT_PARM_VALUE,  
           PROMPT_STRING, PARM_VALUE, USER_NAME  
    FROM DBA_REPCAT_USER_PARM_VALUES;  
  tempstring VARCHAR2(1000);  
  tempstring2 varchar2(1000);  
  len NUMBER;  
BEGIN  
  FOR myrec IN mycursor LOOP  
    len := DBMS_LOB.GETLENGTH(myrec.default_parm_value);  
    DBMS_LOB.READ(myrec.default_parm_value, len, 1, tempstring);  
    DBMS_OUTPUT.PUT_LINE(myrec.refresh_template_name||' '||  
      myrec.owner||' '||myrec.refresh_group_name||' '||  
      myrec.template_comment||' '||myrec.public_template||' '||  
      myrec.parameter_name||' '||tempstring||' '||myrec.prompt_string||' '||  
      tempstring2||' '||myrec.user_name);  
  END LOOP;  
END;  
/
```

**関連項目：** カーソルの使用方法の詳細は、『Oracle9i アプリケーション開発者ガイド - 基礎編』を参照してください。DBMS\_LOB パッケージと LOB の一般的な使用方法の詳細は、『Oracle9i アプリケーション開発者ガイド ラージ・オブジェクト』を参照してください。

## ALL\_REPCATLOG

各マスター・サイトで生成された非同期管理要求およびエラー・メッセージの暫定的なステータスの情報が示されます。要求の処理中に生成されたすべてのエラー・メッセージは、その要求を作成したマスター・サイトの ALL\_REPCATLOG ビューに転送されます。エラーが発生せずに管理要求が正常に完了した場合は、この要求に関するすべてのトレースは、ALL\_REPCATLOG ビューから削除されます。このビューに含まれる管理要求およびエラー・メッセージは、カレント・ユーザーからアクセス可能です。

### 関連するビュー：

- DBA\_REPCATLOG には、データベース内のすべての非同期管理要求およびエラー・メッセージのステータスが示されます。
- USER\_REPCATLOG には、カレント・ユーザーが所有するすべての非同期管理要求およびエラー・メッセージのステータスが示されます。

列	データ型	NULL	説明
ID	NUMBER	-	順序番号。ID 列と SOURCE 列の値により、1 つの管理要求に関連するすべてのログ・レコード（全マスター・サイトを含む）が識別されます。
SOURCE	VARCHAR2 (128)	-	要求が作成された場所。
USERID	VARCHAR2 (30)	-	要求を作成したユーザー名。
TIMESTAMP	DATE	-	要求が作成された時刻。
ROLE	VARCHAR2 (9)	-	サイトがマスター定義サイト（masterdef）か、マスター・サイト（master）かを示します。
MASTER	VARCHAR2 (128)	-	ROLE 列が 'masterdef' のときにリモート・タスクを実行する場合、どのマスター・サイトがタスクを実行するかを示します。
SNAME	VARCHAR2 (30)	-	レプリケート・オブジェクトのスキーマ名（該当する名前がある場合）。
REQUEST	VARCHAR2 (29)	-	実行された DBMS_REPCAT 管理プロシージャの名前。
ONAME	VARCHAR2 (30)	-	レプリケート・オブジェクトの名前（該当する名前がある場合）。

列	データ型	NULL	説明
TYPE	VARCHAR2 (12)	-	レプリケート・オブジェクトのタイプ。 <div>FUNCTION            SNAPSHOT</div> <div>INDEX                SYNONYM</div> <div>INDEXTYPE            TABLE</div> <div>OPERATOR             TRIGGER</div> <div>PACKAGE              TYPE</div> <div>PACKAGE BODY        TYPE BODY</div> <div>PROCEDURE            VIEW</div>
STATUS	VARCHAR2 (14)	-	管理要求のステータス。ステータスには、READY、DO_CALLBACK、AWAIT_CALLBACK、ERROR があります。
MESSAGE	VARCHAR2 (200)	-	戻されたエラー・メッセージ。
ERRNUM	NUMBER	-	メッセージの Oracle エラー番号。
GNAME	VARCHAR2 (30)	-	レプリケーション・グループの名前。

ALL\_REPCOLUMN

カレント・ユーザーからアクセス可能な表の、レプリケートされた列が表示されます。

表に列オブジェクトが含まれる場合、このビューには、型のプレースホルダと、型属性ごとに 1 行の情報が表示されます。表にネストした表が含まれる場合、このビューには、ネストした表の記憶表が独立した表として表示されます。表がオブジェクト表の場合、このビューには、非表示のオブジェクト識別子列が表示されます。

関連するビュー：

- DBA\_REPCOLUMN には、データベース内のすべての表の、レプリケートされた列が示されます。
- USER\_REPCOLUMN には、カレント・ユーザーが所有するすべての表の、レプリケートされた列が示されます。

列	データ型	NULL	説明
SNAME	VARCHAR2 (30)	NOT NULL	オブジェクトの所有者の名前。
ONAME	VARCHAR2 (30)	NOT NULL	オブジェクトの名前。
TYPE	VARCHAR2 (8)	-	オブジェクトのタイプ。値は、SNAPSHOT または TABLE のいずれかです。
CNAME	VARCHAR2 (4000)	-	レプリケートされた列の名前。
ID	NUMBER	-	レプリケートされた列の ID 番号。
POS	NUMBER	-	レプリケートされた列の順序。
COMPARE_OLD_ON_DELETE	VARCHAR2 (1)	-	レプリケートされた情報の削除時に、列の古い値と比較するかどうかを示します。
COMPARE_OLD_ON_UPDATE	VARCHAR2 (1)	-	レプリケートされた情報の更新時に、列の古い値と比較するかどうかを示します。
SEND_OLD_ON_DELETE	VARCHAR2 (1)	-	レプリケートされた情報の削除時に、列の古い値を送信するかどうかを示します。
SEND_OLD_ON_UPDATE	VARCHAR2 (1)	-	レプリケートされた情報の更新時に、列の古い値を送信するかどうかを示します。
CTYPE	VARCHAR2 (30)	-	列のタイプが表示されます。ユーザー定義型の場合は、ユーザー定義型名が表示されます。
CTYPE_TOID	RAW (16)	-	ユーザー定義型の場合は、その型のオブジェクト識別子 (OID) が表示されます。それ以外の場合、このフィールドは NULL になります。
CTYPE_OWNER	VARCHAR2 (30)	-	ユーザー定義型の場合は、ユーザー定義型の所有者が表示されます。それ以外の場合、このフィールドは NULL になります。
CTYPE_HASHCODE	VARCHAR2 (34)	-	ユーザー定義型の場合は、その型のハッシュコードが表示されます。それ以外の場合、このフィールドは NULL になります。
CTYPE_MOD	VARCHAR2 (3)	-	REF 列の場合は、REF が表示されます。それ以外の場合、このフィールドは NULL になります。
DATA_LENGTH	VARCHAR2 (40)	-	列の長さがバイト単位で表示されます。
DATA_PRECISION	VARCHAR2 (40)	-	列の精度が表示されます。NUMBER 列の場合は 10 進数で表示され、FLOAT 列の場合は 2 進数で表示されます。
DATA_SCALE	VARCHAR2 (40)	-	小数点以下の位の桁数が表示されます。
NULLABLE	VARCHAR2 (1)	-	列に NULL 値を指定できるかどうかを示します。

列	データ型	NULL	説明
CHARACTER_SET_NAME	VARCHAR2 (44)	-	列で使用するキャラクタ・セットの名前が表示されます（該当する値がある場合）。
TOP	VARCHAR2 (30)	-	列オブジェクト内の属性に対する最上位列が表示されます。たとえば、oe.customers 表では、cust_address が列オブジェクトで、その属性の 1 つが street_address です。cust_address は、street_address 属性に対する TOP 列になります。  組込みのデータ型の場合、このフィールドは NULL になります。
CHAR_LENGTH	NUMBER	-	列の長さが文字単位で表示されます。この値は、次のデータ型に対してのみ適用されます。 <ul style="list-style-type: none"><li>■ CHAR</li><li>■ VARCHAR2</li><li>■ NCHAR</li><li>■ NVARCHAR2</li></ul>
CHAR_USED	VARCHAR2 (1)	-	B は、列の長さセマンティクスとして BYTE が使用されていることを示します。C は、列の長さセマンティクスとして CHAR が使用されていることを示します。NULL は、データ型が次のいずれでもないことを示します。 <ul style="list-style-type: none"><li>■ CHAR</li><li>■ VARCHAR2</li><li>■ NCHAR</li><li>■ NVARCHAR2</li></ul>

## ALL\_REPCOLUMN\_GROUP

カレント・ユーザーからアクセス可能な各レプリケート表の列グループが示されます。

### 関連するビュー：

- DBA\_REPCOLUMN\_GROUP には、データベース内のすべての表の列グループが示されます。
- USER\_REPCOLUMN\_GROUP には、カレント・ユーザーが所有するすべての表の列グループが示されます。

列	データ型	NULL	説明
SNAME	VARCHAR2 (30)	NOT NULL	レプリケート表を含むスキーマの名前。
ONAME	VARCHAR2 (30)	NOT NULL	レプリケート表の名前。
GROUP_NAME	VARCHAR2 (30)	NOT NULL	列グループの名前。
GROUP_COMMENT	VARCHAR2 (80)	-	ユーザーが入力するコメント。

---

**注意：** USER\_REPCOLUMN\_GROUP ビューには、SNAME 列が示されません。

---

## ALL\_REPCONFLICT

競合解消メソッドが定義されていて、カレント・ユーザーからアクセス可能な各表の名前と、その方法で解消される競合のタイプが含まれます。

### 関連するビュー：

- DBA\_REPCONFLICT には、競合解消メソッドが定義されているデータベース内のすべての表に対する競合解消メソッドが示されます。
- USER\_REPCONFLICT には、競合解消メソッドが定義されている、カレント・ユーザーが所有するすべての表に対する競合解消メソッドが示されます。

列	データ型	NULL	説明
SNAME	VARCHAR2 (30)	NOT NULL	レプリケート表を含むスキーマの名前。
ONAME	VARCHAR2 (30)	NOT NULL	競合解消メソッドが定義された表の名前。
CONFLICT_TYPE	VARCHAR2 (10)	-	競合解消メソッドを適用する競合のタイプ。削除の競合、一意性競合、更新の競合のいずれかです。
REFERENCE_NAME	VARCHAR2 (30)	NOT NULL	方法を適用するオブジェクト。削除の競合では表の名前です。一意性競合では制約の名前です。更新の競合では列グループの名前です。

**注意：** USER\_REPCONFLICT ビューには、SNAME 列が示されません。

ALL\_REPDDL

カレント・ユーザーからアクセス可能な各レプリケーション・オブジェクトの DDL が示されます。

関連するビュー：

- DBA\_REPDDL には、データベース内の各レプリケート・オブジェクトの DDL が示されます。
- USER\_REPDDL には、カレント・ユーザーが所有する各レプリケート・オブジェクトの DDL が示されます。

列	データ型	NULL	説明
LOG_ID	NUMBER	-	ALL_REPCATLOG レコードの ID 番号。
SOURCE	VARCHAR2 (128)	-	要求の作成元データベースの名前。
ROLE	VARCHAR2 (1)	-	Y の場合、データベースは要求に対するマスター定義サイト (masterdef) です。 N の場合、データベースはマスター・サイトです。
MASTER	VARCHAR2 (128)	-	この要求を処理するデータベースの名前。
LINE	NUMBER (38)	-	1 つの要求に含まれるレコードの順序。
TEXT	VARCHAR2 (2000)	-	引数または DDL テキストの一部分。
DDL_NUM	NUMBER (38)	-	複数の DDL 文を使用する場合に、TEXT 列に格納されている DDL 文の実行順序を示します。



## ALL\_REPGENOBJECTS

カレント・ユーザーからアクセス可能で、レプリケーションのサポート用に生成されたオブジェクトが示されます。

### 関連するビュー：

- DBA\_REPGENOBJECTS には、レプリケーションのサポート用に生成された、データベース内の各オブジェクトが示されます。
- USER\_REPGENOBJECTS には、レプリケーションのサポート用に生成された、カレント・ユーザーが所有する各オブジェクトが示されます。

列	データ型	NULL	説明
SNAME	VARCHAR2 (30)	-	レプリケート・スキーマの名前。
ONAME	VARCHAR2 (30)	-	生成されたオブジェクトの名前。
TYPE	VARCHAR2 (12)	-	生成されたオブジェクトのタイプ。値は、PACKAGE、PACKAGE BODY、TRIGGER または INTERNAL PACKAGE のいずれかです。
BASE_SNAME	VARCHAR2 (30)	-	ベースとなるオブジェクトの所有者。
BASE_ONAME	VARCHAR2 (30)	-	このオブジェクトを生成する原因となったオブジェクト。
BASE_TYPE	VARCHAR2 (12)	-	ベースとなるオブジェクトのタイプ。
PACKAGE_PREFIX	VARCHAR2 (30)	-	パッケージ・ラッパーの接頭辞。
PROCEDURE_PREFIX	VARCHAR2 (30)	-	パッケージ・ラッパーのプロシージャ接頭辞。
DISTRIBUTED	VARCHAR2 (1)	-	この列は廃止されています。
REASON	VARCHAR2 (30)	-	オブジェクトの生成理由。

ALL\_REPGROUP

カレント・ユーザーからアクセス可能なレプリケーション・グループがすべて示されます。  
各レプリケーション・グループのメンバーは、ALL\_REPOBJECT ビューにリストされます。

関連するビュー：

- DBA\_REPGROUP には、レプリケートの対象となる、データベース内のレプリケーション・グループがすべて示されます。
- USER\_REPGROUP には、レプリケートの対象となる、カレント・ユーザーが所有するレプリケーション・グループがすべて示されます。

列	データ型	NULL	説明
SNAME	VARCHAR2 (30)	NOT NULL	レプリケート・スキーマの名前。リリース 7.3 以降では廃止されています。
MASTER	VARCHAR2 (1)	-	Y の場合、カレント・サイトはマスター・サイトです。  N の場合、カレント・サイトはマテリアライズド・ビュー・サイトです。
STATUS	VARCHAR2 (9)	-	マスター・サイトに対してのみ使用されます。ステータスは、normal、quiescing、quiesced のいずれかです。
SCHEMA_COMMENT	VARCHAR2 (80)	-	ユーザーが入力するコメント。
GNAME	VARCHAR2 (30)	NOT NULL	レプリケーション・グループの名前。
FNAME	VARCHAR2 (30)	-	固有の名前。
RPC_PROCESSING_DISABLED	VARCHAR2 (1)	-	N の場合、このサイトでの遅延リモート・プロシージャ・コール (RPC) の受信や適用が可能です。  Y の場合、このサイトでの遅延リモート・プロシージャ・コール (RPC) の受信や適用が不可能です。
OWNER	VARCHAR2 (30)	NOT NULL	レプリケーション・グループの所有者。

## ALL\_REPGROUP\_PRIVILEGES

レプリケーション・グループ内で権限を登録されているユーザーの情報が示されます。カレント・ユーザーからアクセス可能なレプリケーション・グループのみが表示されます。

### 関連するビュー：

- DBA\_REPGROUP\_PRIVILEGES には、データベース内のすべてのレプリケーション・グループ内で権限を登録されているユーザーの情報が示されます。
- USER\_REPGROUP\_PRIVILEGES には、カレント・ユーザーが所有するレプリケーション・グループ内で権限を登録されているユーザーの情報が示されます。

列	データ型	NULL	説明
USERNAME	VARCHAR2 (30)	NOT NULL	ユーザーの名前が表示されます。
GNAME	VARCHAR2 (30)	-	レプリケーション・グループの名前が表示されます。
CREATED	DATE	NOT NULL	レプリケーション・グループの登録日が表示されます。
RECEIVER	VARCHAR2 (1)	-	ユーザーが受信者としての権限を持っているかどうかを示します。
PROXY_SNAPADMIN	VARCHAR2 (1)	-	ユーザーが proxy_snapadmin の権限を持っているかどうかを示します。
OWNER	VARCHAR2 (30)	-	レプリケーション・グループの所有者。

## ALL\_REPGROUPED\_COLUMN

カレント・ユーザーからアクセス可能な各表の列グループを構成する、すべての列が示されます。

### 関連するビュー：

- DBA\_REPGROUPED\_COLUMN には、データベース内の各表の列グループを構成する、すべての列が示されます。
- USER\_REPGROUPED\_COLUMN には、カレント・ユーザーが所有する各表の列グループを構成する、すべての列が示されます。

列	データ型	NULL	説明
SNAME	VARCHAR2 (30)	NOT NULL	レプリケート表を含むスキーマの名前。
ONAME	VARCHAR2 (30)	NOT NULL	レプリケート表の名前。
GROUP_NAME	VARCHAR2 (30)	NOT NULL	列グループの名前。
COLUMN_NAME	VARCHAR2 (30)	NOT NULL	列グループに含まれる列の名前。

**注意：** USER\_REPGROUPED\_COLUMN ビューには、SNAME 列が示されません。

## ALL\_REPKEY\_COLUMNS

カレント・ユーザーからアクセス可能な各表の 1 つ以上のレプリケーション・キー列が示されます。

レプリケーション・キー列は代替列または列グループです。主キーのかわりにレプリケーション・キー列を使用して、行レベル・レプリケーションを使用する際に比較する表中の列を決定します。レプリケーション・キー列を設定するには、DBMS\_REPCAT パッケージの SET\_COLUMNS プロシージャを使用します。

次のタイプの列は、レプリケーション・キー列にすることはできません。

- LOB または列オブジェクトの LOB 属性
- コレクションまたは列オブジェクトのコレクション属性
- REF
- 列オブジェクト全体

関連項目： 20-102 ページ「[SET\\_COLUMNS プロシージャ](#)」

### 関連するビュー：

- DBA\_REPKEY\_COLUMNS には、データベース内の各表の 1 つ以上のレプリケーション・キー列が示されます。
- USER\_REPKEY\_COLUMNS には、カレント・ユーザーが所有する各表の 1 つ以上のレプリケーション・キー列が示されます。

列	データ型	NULL	説明
SNAME	VARCHAR2 (30)	NOT NULL	レプリケート表の所有者。
ONAME	VARCHAR2 (30)	NOT NULL	レプリケート表の名前。
COL	VARCHAR2 (4000)	-	表内の 1 つ以上のレプリケーション・キー列。

## ALL\_REPOBJECT

カレント・ユーザーからアクセス可能な各レプリケーション・グループ内のオブジェクトの情報が示されます。オブジェクトは、1つのレプリケーション・グループのみに属します。レプリケーション・グループは、複数のスキーマにまたがることができます。

### 関連するビュー：

- DBA\_REPOBJECT には、データベースの各レプリケーション・グループ内のオブジェクトの情報が示されます。
- USER\_REPOBJECT には、カレント・ユーザーが所有する各レプリケーション・グループ内のオブジェクトの情報が示されます。

列	データ型	NULL	説明
SNAME	VARCHAR2 (30)	-	レプリケート・オブジェクトを含むスキーマの名前。
ONAME	VARCHAR2 (30)	-	レプリケート・オブジェクトの名前。
TYPE	VARCHAR2 (16)	-	レプリケート・オブジェクトのタイプ。 <div>FUNCTION                      SNAPSHOT</div> <div>INDEX                          SYNONYM</div> <div>INDEXTYPE                      TABLE</div> <div>OPERATOR                       TRIGGER</div> <div>PACKAGE                        TYPE</div> <div>PACKAGE BODY                  TYPE BODY</div> <div>PROCEDURE                      VIEW</div>
STATUS	VARCHAR2 (10)	-	CREATE は、ユーザーまたは Oracle によって生成された DDL が、ローカル・データベースに対して適用され、オブジェクトがローカルに作成されることを示します。ローカル・レプリカが存在する場合、Oracle は、そのレプリカをマスター定義と比較して、両者に整合性があることを確認します (COMPARE)。オブジェクトの作成または比較が正常に実行されると、ステータスが VALID に更新されます。正常に実行されなかった場合、ステータスは ERROR になります。オブジェクトを削除すると、ALL_REPOBJECT ビューから行が削除される前にステータスが DROPPED に更新されます。

列	データ型	NULL	説明
GENERATION_STATUS	VARCHAR2 (9)	-	オブジェクトでレプリケーション・パッケージを生成する必要があるかどうかを示します。
ID	NUMBER	-	ローカル・データベース・オブジェクトの識別子 (該当する値がある場合)。
OBJECT_COMMENT	VARCHAR2 (80)	-	ユーザーが入力するコメント。
GNAME	VARCHAR2 (30)	-	オブジェクトが属するレプリケーション・グループの名前。
MIN_COMMUNICATION	VARCHAR2 (1)	-	Y の場合は、最低限の通信によって更新されます。 N の場合は、更新用の新旧データがすべて送信されます。
REPLICATION_TRIGGER_EXISTS	VARCHAR2 (1)	-	Y の場合、内部レプリケーション・トリガーが存在します。 N の場合、内部レプリケーション・トリガーは存在しません。
INTERNAL_PACKAGE_EXISTS	VARCHAR2 (1)	-	Y の場合、内部パッケージが存在します。 N の場合、内部パッケージは存在しません。
GROUP_OWNER	VARCHAR2 (30)	-	レプリケーション・グループの所有者。
NESTED_TABLE	VARCHAR2 (1)	-	Y の場合、レプリケート・オブジェクトはネストした表の記憶表です。 N の場合、レプリケート・オブジェクトは記憶表ではありません。

ALL\_REPPARAMETER\_COLUMN

ALL\_REPPARAMETER\_COLUMN ビューには、ALL\_REPRESOLUTION ビューに示される情報と、カレント・ユーザーからアクセス可能な各レプリケート表の競合解消に使用される列の情報が示されます。これらの列の値は、list\_of\_column\_names の引数として、DBMS\_REPCAT パッケージの ADD\_conflicttype\_RESOLUTION プロシージャに渡されます。

関連するビュー：

- DBA\_REPPARAMETER\_COLUMN には、データベース内の各レプリケート表の競合解消に使用される列の情報が示されます。
- USER\_REPPARAMETER\_COLUMN ビューには、カレント・ユーザーが所有する各レプリケート表の競合解消に使用される列の情報が示されます。

列	データ型	NULL	説明
SNAME	VARCHAR2 (30)	NOT NULL	レプリケート表を含むスキーマの名前。
ONAME	VARCHAR2 (30)	NOT NULL	レプリケート表の名前。
CONFLICT_TYPE	VARCHAR2 (10)	-	競合解消メソッドを適用する競合のタイプ。削除の競合、一意性競合、更新の競合のいずれかです。
REFERENCE_NAME	VARCHAR2 (30)	NOT NULL	方法を適用するオブジェクト。削除の競合では表の名前です。一意性競合では制約の名前です。更新の競合では列グループの名前です。
SEQUENCE_NO	NUMBER	NOT NULL	競合解消メソッドを適用する順序。1 は、最初に適用されることを示します。
METHOD_NAME	VARCHAR2 (80)	NOT NULL	Oracle によって提供される競合解消メソッドの名前。ユーザー指定の方法を使用する場合、この値は 'user function' になります。
FUNCTION_NAME	VARCHAR2 (92)	NOT NULL	競合解消メソッドが 'user function' の場合は、ユーザーが入力した競合解消メソッドの名前になります。
PRIORITY_GROUP	VARCHAR2 (30)	-	競合解消メソッドが 'priority group' の場合は、優先グループの名前になります。
PARAMETER_TABLE_NAME	VARCHAR2 (30)	NOT NULL	パラメータ列が属する表の名前が表示されます。
PARAMETER_COLUMN_NAME	VARCHAR2 (4000)	-	競合解消メソッドで IN パラメータとして使用される列の名前。
PARAMETER_SEQUENCE_NO	NUMBER	NOT NULL	IN パラメータとして使用される列の順序。



---

**注意：** USER\_REPPARAMETER\_COLUMN ビューには、SNAME 列が示されません。

---

## ALL\_REPPRIORITY

カレント・ユーザーからアクセス可能な各優先グループの各メンバーの値および優先順位レベルが示されます。優先グループには、レプリケーション・グループ内で一意の名前を付ける必要があります。また、指定された優先グループ内のそれぞれの優先順位レベルと値は、そのグループ内で一意であることが必要です。

### 関連するビュー：

- DBA\_REPPRIORITY には、データベース内の各優先グループの各メンバーの値および優先順位レベルが示されます。
- USER\_REPPRIORITY には、カレント・ユーザーが所有する各優先グループの各メンバーの値および優先順位レベルが示されます。

列	データ型	NULL	説明
SNAME	VARCHAR2(30)	NOT NULL	レプリケート・スキーマの名前。リリース 7.3 以降では廃止されています。
PRIORITY_GROUP	VARCHAR2(30)	NOT NULL	優先グループまたはサイト優先グループの名前。
PRIORITY	NUMBER	NOT NULL	メンバーの優先順位レベル。最も大きい数値は、優先順位レベルが最も高いことを示します。
DATA_TYPE	VARCHAR2(9)	-	優先グループの値のデータ型。
FIXED_DATA_LENGTH	NUMBER(38)	-	CHAR データ型の値の最大長。
CHAR_VALUE	CHAR(255)	-	優先グループのメンバーの値 (DATA_TYPE = CHAR の場合)。
VARCHAR2_VALUE	VARCHAR2(4000)	-	優先グループのメンバーの値 (DATA_TYPE = VARCHAR2 の場合)。
NUMBER_VALUE	NUMBER	-	優先グループのメンバーの値 (DATA_TYPE = NUMBER の場合)。
DATE_VALUE	DATE	-	優先グループのメンバーの値 (DATA_TYPE = DATE の場合)。
RAW_VALUE	RAW(2000)	-	優先グループのメンバーの値 (DATA_TYPE = RAW の場合)。
GNAME	VARCHAR2(30)	NOT NULL	レプリケーション・グループの名前。

ALL\_REPPRIORITY\_GROUP

列	データ型	NULL	説明
NCHAR_VALUE	NCHAR(500)	-	優先グループのメンバーの値 (DATA_TYPE = NCHAR の場合)。
NVARCHAR2_VALUE	VARCHAR2(1000)	-	優先グループのメンバーの値 (DATA_TYPE = NVARCHAR2 の場合)。
LARGE_CHAR_VALUE	CHAR(2000)	-	優先グループのメンバーの値 (255 文字以上の空白埋込み文字列の場合)。

**注意：** USER\_REPPRIORITY ビューには、SNAME 列および GNAME 列が示されません。

ALL\_REPPRIORITY\_GROUP

カレント・ユーザーからアクセス可能な各レプリケーション・グループに定義された、優先グループまたはサイト優先グループが示されます。

関連するビュー：

- DBA\_REPPRIORITY\_GROUP には、データベース内の各レプリケーション・グループに定義された、優先グループまたはサイト優先グループが示されます。
- USER\_REPPRIORITY\_GROUP には、カレント・ユーザーが所有する各レプリケーション・グループに定義された、優先グループまたはサイト優先グループが示されます。

列	データ型	NULL	説明
SNAME	VARCHAR2(30)	NOT NULL	レプリケート・スキーマの名前。リリース 7.3 以降では廃止されています。
PRIORITY_GROUP	VARCHAR2(30)	NOT NULL	優先グループまたはサイト優先グループの名前。
DATA_TYPE	VARCHAR2(9)	-	優先グループの各値のデータ型。
FIXED_DATA_LENGTH	NUMBER(38)	-	CHAR データ型の値の最大長。
PRIORITY_COMMENT	VARCHAR2(80)	-	ユーザーが入力するコメント。
GNAME	VARCHAR2(30)	NOT NULL	レプリケーション・グループの名前。

**注意：** USER\_REPPRIORITY ビューには、SNAME 列および GNAME 列が示されません。

## ALL\_REPPROP

各レプリケート・オブジェクトに対する操作を、他のマスター・サイトにある同じオブジェクトに伝播する手法が示されます。操作の伝播は、ストアド・プロシージャやプロシージャ・ラップ・プログラムをコールしたり、表を直接操作することによって実行します。このビューに示されるのは、カレント・ユーザーからアクセス可能なオブジェクトです。

### 関連するビュー：

- DBA\_REPPROP には、各レプリケート・オブジェクトに対する操作を、他のマスター・サイトにある同じオブジェクトに伝播する手法が示されます。このビューに示されるのは、データベース内のすべてのオブジェクトです。
- USER\_REPPROP には、各レプリケート・オブジェクトに対する操作を、他のマスター・サイトにある同じオブジェクトに伝播する手法が示されます。このビューに示されるのは、カレント・ユーザーが所有するオブジェクトです。

列	データ型	NULL	説明
SNAME	VARCHAR2 (30)	NOT NULL	レプリケート・オブジェクトを含むスキーマの名前。
ONAME	VARCHAR2 (30)	NOT NULL	レプリケート・オブジェクトの名前。
TYPE	VARCHAR2 (16)	-	レプリケートされているオブジェクトのタイプ。 FUNCTION                      PROCEDURE INDEXTYPE                      SNAPSHOT OPERATOR                        TABLE PACKAGE                         TYPE PACKAGE BODY                   TYPE BODY
DBLINK	VARCHAR2 (128)	NOT NULL	マスター・サイトのデータベースの完全修飾名。このデータベースに対して、変更が伝播されます。
HOW	VARCHAR2 (13)	-	伝播の実行方法。ローカル・マスター・サイトの場合は 'none' です。それ以外のサイトの場合は 'synchronous' または 'asynchronous' です。
PROPAGATE_COMMENT	VARCHAR2 (80)	-	ユーザーが入力するコメント。

## ALL\_REPRESOL\_STATS\_CONTROL

カレント・ユーザーからアクセス可能なすべてのレプリケート表に対する、競合解消の統計収集が示されます。

### 関連するビュー：

- DBA\_REPRESOL\_STATS\_CONTROL には、データベース内のすべてのレプリケート表に対する、競合解消の統計収集が示されます。
- USER\_REPRESOL\_STATS\_CONTROL には、カレント・ユーザーが所有するすべてのレプリケート表に対する、競合解消の統計収集が示されます。

列	データ型	NULL	説明
SNAME	VARCHAR2 (30)	NOT NULL	表の所有者。
ONAME	VARCHAR2 (30)	NOT NULL	表の名前。
CREATED	DATE	NOT NULL	統計収集の開始を示すタイムスタンプ。
STATUS	VARCHAR2 (9)	-	統計収集のステータス。ACTIVE または CANCELLED になります。
STATUS_UPDATE_DATE	DATE	NOT NULL	前回のステータス更新日を示すタイムスタンプ。
PURGED_DATE	DATE	-	前回の統計データのパージ日を示すタイムスタンプ。
LAST_PURGE_START_DATE	DATE	-	前回の統計情報のパージ期間の開始日。
LAST_PURGE_END_DATE	DATE	-	前回の統計情報のパージ期間の終了日。

**注意：** USER\_REPRESOL\_STATS\_CONTROL ビューには、SNAME 列が示されません。

## ALL\_REPRESOLUTION

カレント・ユーザーからアクセス可能な各レプリケート表での更新、一意性または削除の競合の解消に使用する方法が示されます。この場合の表は、指定されたスキーマに対する行レベル・レプリケーションによってレプリケートされています。

### 関連するビュー：

- DBA\_REPRESOLUTION には、データベース内の各レプリケート表での更新、一意性または削除の競合の解消に使用する方法が示されます。この場合の表は、指定されたスキーマに対する行レベル・レプリケーションによってレプリケートされています。
- USER\_REPRESOLUTION には、カレント・ユーザーが所有する各レプリケート表での更新、一意性または削除の競合の解消に使用する方法が示されます。この場合の表は、行レベル・レプリケーションによってレプリケートされています。

列	データ型	NULL	説明
SNAME	VARCHAR2 (30)	NOT NULL	レプリケート・スキーマの名前。
ONAME	VARCHAR2 (30)	NOT NULL	レプリケート表の名前。
CONFLICT_TYPE	VARCHAR2 (10)	-	競合解消メソッドを適用する競合のタイプ。削除の競合、一意性競合、更新の競合のいずれかです。
REFERENCE_NAME	VARCHAR2 (30)	NOT NULL	方法を適用するオブジェクト。削除の競合では表の名前です。一意性競合では制約の名前です。更新の競合では列グループの名前です。
SEQUENCE_NO	NUMBER	NOT NULL	競合解消メソッドを適用する順序。1 は、最初に適用されることを示します。
METHOD_NAME	VARCHAR2 (80)	NOT NULL	Oracle によって提供される競合解消メソッドの名前。ユーザー指定の方法を使用する場合、この値は 'user function' になります。
FUNCTION_NAME	VARCHAR2 (92)	NOT NULL	競合解消メソッドが 'user function' の場合は、ユーザーが入力した競合解消メソッドの名前になります。
PRIORITY_GROUP	VARCHAR2 (30)	-	競合解消メソッドが 'priority group' の場合は、優先グループの名前になります。
RESOLUTION_COMMENT	VARCHAR2 (80)	-	ユーザーが入力するコメント。

---

**注意：** USER\_REPRESOLUTION ビューには、SNAME 列が示されません。

---

ALL\_REPRESOLUTION\_METHOD

データベース内で使用可能な競合解消メソッドがすべてリストされます。まず、このビューには、Advanced Replication が提供する標準の方法がリストされます。新しいユーザー・ファンクションを作成してデータベース内のオブジェクト用の競合解消メソッドとして追加すると、そのファンクションがこのビューに追加されます。

関連するビュー：

- DBA\_REPRESOLUTION\_METHOD には、データベース内で使用可能な競合解消メソッドがすべてリストされます。
- USER\_REPRESOLUTION\_METHOD には、カレント・ユーザーが所有する使用可能な競合解消メソッドがすべてリストされます。

列	データ型	NULL	説明
CONFLICT_TYPE	VARCHAR2 (10)	-	解決メソッドを適用する競合のタイプ。更新の競合、一意性競合、削除の競合のいずれかです。
METHOD_NAME	VARCHAR2 (80)	NOT NULL	Oracle が提供する競合解消メソッドの名前。またはユーザーが入力した方法の名前。

ALL\_REPRESOLUTION\_STATISTICS

カレント・ユーザーからアクセス可能なすべてのレプリケート表に対して、正常に実行された更新、一意性または削除の競合解消の情報がリストされます。これらの統計情報は、DBMS\_REPCAT.REGISTER\_STATISTICS プロシージャをコールした場合にかぎり、表に対して収集されます。

関連するビュー：

- DBA\_REPRESOLUTION\_STATISTICS には、データベース内のすべてのレプリケート表に対して、正常に実行された更新、一意性または削除の競合解消の情報がリストされます。
- USER\_REPRESOLUTION\_STATISTICS には、カレント・ユーザーが所有するすべてのレプリケート表に対して、正常に実行された更新、一意性または削除の競合解消の情報がリストされます。

列	データ型	NULL	説明
SNAME	VARCHAR2 (30)	NOT NULL	レプリケート・スキーマの名前。
ONAME	VARCHAR2 (30)	NOT NULL	レプリケート表の名前。
CONFLICT_TYPE	VARCHAR2 (10)	-	正常に解消された競合のタイプ。削除の競合、一意性競合、更新の競合のいずれかです。

列	データ型	NULL	説明
REFERENCE_NAME	VARCHAR2 (30)	NOT NULL	競合解消メソッドを適用するオブジェクト。削除の競合では表の名前です。一意性競合では制約の名前です。更新の競合では列グループの名前です。
METHOD_NAME	VARCHAR2 (80)	NOT NULL	Oracle によって提供される競合解消メソッドの名前。ユーザー指定の方法を使用する場合、この値は 'user function' になります。
FUNCTION_NAME	VARCHAR2 (92)	-	競合解消メソッドが 'user function' の場合は、ユーザーが入力した競合解消メソッドの名前になります。
PRIORITY_GROUP	VARCHAR2 (30)	-	競合解消メソッドが 'priority group' の場合は、優先グループの名前になります。
RESOLVED_DATE	DATE	NOT NULL	この行の競合が解消された日付。
PRIMARY_KEY_VALUE	VARCHAR2 (2000)	NOT NULL	行の主キーの連結表現。

---

**注意：** USER\_REPRESOLUTION\_STATISTICS ビューには、SNAME 列が示されません。

---

## ALL\_REPSITES

カレント・ユーザーからアクセス可能な各レプリケーション・グループのメンバーがリストされます。

### 関連するビュー：

- DBA\_REPSITES ビューには、データベース内の各レプリケーション・グループのメンバーがリストされます。
- USER\_REPSITES には、カレント・ユーザーが所有する各レプリケーション・グループのメンバーがリストされます。

列	データ型	NULL	説明
GNAME	VARCHAR2 (30)	NOT NULL	レプリケーション・グループの名前。
DBLINK	VARCHAR2 (128)	NOT NULL	このレプリケーション・グループのマスター・サイトへのデータベース・リンク。
MASTERDEF	VARCHAR2 (1)	-	マスター定義サイトとして指定する DBLINK を示します。
SNAPMASTER	VARCHAR2 (1)	-	リフレッシュ時に使用する DBLINK を示します。マテリアライズド・ビュー・サイトの場合に使用されます。
MASTER_COMMENT	VARCHAR2 (80)	-	ユーザーが入力するコメント。
MASTER	VARCHAR2 (1)	-	Y の場合、サイトはレプリケート・グループのマスター・サイトです。 N の場合、サイトはレプリケート・グループのマスター・サイトではありません。
GROUP_OWNER	VARCHAR2 (30)	NOT NULL	レプリケーション・グループの所有者。

DBA\_REPSITES ビューには、次の列も含まれます。

列	データ型	NULL	説明
PROP_UPDATES	NUMBER	-	伝播方法のエンコーディング（マスター・サイトの場合）。
MY_DBLINK	VARCHAR2 (1)	-	インポート後、問題を検出するために使用されます。Y の場合、DBLINK がグローバル名です。

## DBA\_REPCAT\_REFRESH\_TEMPLATES

DBA\_REPCAT\_REFRESH\_TEMPLATES ビューには、データベース内の各配置テンプレートに関するグローバル情報が示されます。テンプレート名、テンプレートの所有者、テンプレート・オブジェクトが属するリフレッシュ・グループ、テンプレートの種類（プライベートまたはパブリック）などの情報があります。

このビューの列は、ALL\_REPCAT\_REFRESH\_TEMPLATES の列と同じです。このビューおよび列の詳細は、25-6 ページの「[ALL\\_REPCAT\\_REFRESH\\_TEMPLATES](#)」を参照してください。



## DBA\_REPCAT\_TEMPLATE\_OBJECTS

DBA\_REPCAT\_TEMPLATE\_OBJECTS ビューには、データベース内のすべての配置テンプレートの各オブジェクト定義が示されます。REFRESH\_TEMPLATE\_NAME にターゲット・テンプレートを指定すると、各オブジェクトがテンプレートに追加されます。

このビューの列は、ALL\_REPCAT\_TEMPLATE\_OBJECTS の列と同じです。このビューおよび列の詳細は、25-7 ページの「[ALL\\_REPCAT\\_TEMPLATE\\_OBJECTS](#)」を参照してください。

## DBA\_REPCAT\_TEMPLATE\_PARS

データベース内のすべてのテンプレートのオブジェクト DDL で定義されたパラメータは、DBA\_REPCAT\_TEMPLATE\_PARS 表に格納されます。オブジェクトがテンプレートに追加されると、DDL に変数があるかどうかを検査されます。パラメータが見つかったら、自動的にこのビューに追加されます。

このビューの列は、ALL\_REPCAT\_TEMPLATE\_PARS の列と同じです。このビューおよび列の詳細は、25-9 ページの「[ALL\\_REPCAT\\_TEMPLATE\\_PARS](#)」を参照してください。

## DBA\_REPCAT\_TEMPLATE\_SITES

DBA\_REPCAT\_TEMPLATE\_SITES ビューは、エンタープライズ・ネットワークのすべてのサイトにおける、テンプレートのインスタンス化のカレント・ステータス情報を DBA に提供します。このビューには、データベース内のすべての配置テンプレートのインスタンス化サイトの情報が示されます。具体的には、DBA は、特定のサイトで実行されるテンプレートのインストールや削除を監視できます。このビューの列は、25-11 ページの「[ALL\\_REPCAT\\_TEMPLATE\\_SITES](#)」の列と同じです。

## DBA\_REPCAT\_USER\_AUTHORIZATIONS

DBA\_REPCAT\_USER\_AUTHORIZATIONS ビューには、プライベート用に指定された、データベース内のすべてのテンプレートに対する権限を付与されたユーザーがリストされます。このビューにリストされたユーザーは、指定されたテンプレートをインスタンス化できます。このビューにリストされないユーザーはテンプレートをインスタンス化できません。このビューの列は、25-12 ページの「[ALL\\_REPCAT\\_USER\\_AUTHORIZATIONS](#)」の列と同じです。

## DBA\_REPCAT\_USER\_PARM\_VALUES

DBA\_REPCAT\_USER\_PARM\_VALUES ビューには、データベース内のすべての配置テンプレートのテンプレート・パラメータが示されます。DBA には、テンプレートを配布してインスタンス化する前に、ユーザー・パラメータの表を作成するオプションの機能があります。指定されたユーザーがテンプレートをインスタンス化する場合は、そのユーザー用に DBA\_REPCAT\_USER\_PARM\_VALUES 表に格納された値が自動的に使用されます。

このビューの列は、ALL\_REPCAT\_USER\_PARM\_VALUES の列と同じです。このビューおよび列の詳細は、25-13 ページの「[ALL\\_REPCAT\\_USER\\_PARM\\_VALUES](#)」を参照してください。

## DBA\_REPCATLOG

各マスター・サイトの DBA\_REPCATLOG ビューには、非同期管理要求および生成されたエラー・メッセージの暫定的なステータスの情報が示されます。要求の処理中に生成されたすべてのエラー・メッセージは、その要求を作成したマスター・サイトの DBA\_REPCATLOG ビューに転送されます。エラーが発生せずに管理要求が正常に完了した場合は、この要求に関するすべてのトレースは、DBA\_REPCATLOG ビューから削除されます。このビューの列は、25-15 ページの「[ALL\\_REPCATLOG](#)」の列と同じです。

## DBA\_REPCOLUMN

DBA\_REPCOLUMN ビューには、データベース内のすべての表のレプリケートされた列がリストされます。このビューの列は、25-16 ページの「[ALL\\_REPCOLUMN](#)」の列と同じです。

## DBA\_REPCOLUMN\_GROUP

DBA\_REPCOLUMN\_GROUP ビューには、データベース内の各レプリケート表の列グループがリストされます。このビューの列は、25-19 ページの「[ALL\\_REPCOLUMN\\_GROUP](#)」の列と同じです。

## DBA\_REPCONFLICT

DBA\_REPCONFLICT ビューには、競合解消メソッドが定義されている、データベース内の各表の名前と、競合解消メソッドを適用する競合のタイプが表示されます。このビューの列は、25-19 ページの「[ALL\\_REPCONFLICT](#)」の列と同じです。

## DBA\_REPDDL

DBA\_REPDDL ビューには、データベース内のレプリケーション・オブジェクトの DDL が示されます。このビューの列は、25-20 ページの「[ALL\\_REPDDL](#)」の列と同じです。

## DBA\_REPEXTENSIONS

DBA\_REPEXTENSIONS ビューには、マスター・グループを静止させずに、マスター・グループに新しいマスター・サイトを追加しようとしているカレント操作の情報が示されます。

**関連項目：** レプリケーション環境に新しいマスター・サイトを追加するプロシージャの詳細は、7-4 ページの「[マスター・グループを静止することなく、新しいマスター・サイトを追加](#)」を参照してください。

列	データ型	NULL	説明
EXTENSION_ID	RAW (16)	NOT NULL	静止せずにマスター・グループにマスター・データベースを追加する、カレントの保留要求の識別子。
REQUEST	VARCHAR2 (15)	-	拡張要求タイプ。現在可能な値は、 <code>ADD_NEW_MASTERS</code> のみです。この値は、静止せずにマスター・グループに新しいマスター・サイトを追加する要求を示します。
MASTERDEF	VARCHAR2 (128)	-	マスター・サイトが追加されるマスター・グループのマスター定義サイトのグローバル名。
EXPORT_REQUIRED	VARCHAR2 (3)	-	<p><code>YES</code> は、データベース全体または表レベルのエクスポート / インポートを使用して、1 つ以上の新しいマスター・サイトが追加されることを示します。</p> <p><code>NO</code> は、変更ベースのリカバリを使用して、すべての新しいマスター・サイトが追加されることを示します。</p>
REPCATLOG_ID	NUMBER	-	レプリケーション拡張に関連するレプリケーション・カタログ・レコードの識別子。マスター定義サイトは、このレコード上で待機します。この値は、マスター定義サイトでのみ有効です。
EXTENSION_STATUS	VARCHAR2 (13)	-	<p>各レプリケーション拡張のステータス。この値は、マスター定義サイトでのみ有効です。</p> <p>可能な値は次のとおりです。</p> <p><code>READY</code>: 拡張要求が作成され、準備が完了しています。</p> <p><code>STOPPING</code>: マスター・グループへの新しいマスター・サイトの追加が完了しました。マスター定義サイトは、既存のマスター・サイトから新しいマスター・サイトおよびマスター定義サイト自体への伝播を停止しようとしています。</p> <p><code>EXPORTING</code>: 既存のマスター・サイトから新しいマスター・サイトおよびマスター定義サイトへの遅延トランザクションの伝播が停止しました。マスター定義サイトは、エクスポートの完了を待機しています。</p> <p><code>INSTANTIATING</code>: <code>DBMS_REPCAT.RESUME_PROPAGATION_TO_MDEF</code> プロシージャが実行されています (エクスポートが使用された場合)。マスター定義サイトは、新しいマスター・サイトがインスタンス化されるのを待機しています。</p> <p><code>ERROR</code>: 拡張要求の実行時にエラーが発生しました。</p>

列	データ型	NULL	説明
FLASHBACK_SCN	NUMBER	-	新しいマスター・サイトを追加するときに、エクスポートまたは変更ベースのリカバリで使用する必要があるシステム変更番号 (SCN)。新しいマスター・サイトは、リストされた SCN と整合性がある必要があります。
BREAK_TRANS_TO_MASTERDEF	VARCHAR2 (3)	-	<p>この値は、EXPORT_REQUIRED が TRUE の場合のみ有効です。</p> <p>BREAK_TRANS_TO_MASTERDEF が TRUE の場合、既存のマスター・サイトは、マスター・サイトが追加されていないレプリケーション・グループのために、マスター定義サイトへの遅延トランザクションの伝播を継続できます。マスター・サイトが追加されているレプリケーション・グループの遅延トランザクションは、エクスポートが完了するまで伝播できません。</p> <p>BREAK_TRANS_TO_MASTERDEF が FALSE の場合、既存のマスター・サイトは、遅延トランザクションをマスター定義サイトにまったく伝播できません。</p>
BREAK_TRANS_TO_NEW_MASTERS	VARCHAR2 (3)	-	<p>BREAK_TRANS_TO_NEW_MASTERS が TRUE の場合、既存のマスター・サイトは、マスター・サイトが追加されていないレプリケーション・グループのために、新しいマスター・サイトへの遅延トランザクションの伝播を継続できます。</p> <p>BREAK_TRANS_TO_NEW_MASTERS が FALSE の場合、新しいマスター・サイトへの遅延トランザクション・キューの伝播は使用禁止になります。</p>
PERCENTAGE_FOR_CATCHUP_MDEF	NUMBER	-	<p>この値は、BREAK_TRANS_TO_MASTERDEF が TRUE の場合のみ有効です。</p> <p>マスター定義サイトへの伝播に追いつくために使用する必要のある、伝播リソースのパーセンテージです。</p>
CYCLE_SECONDS_MDEF	NUMBER	-	この値は、PERCENTAGE_FOR_CATCHUP_MDEF が有効であり、かつその値が 10 ～ 90 (10 と 90 を含む) の間にある場合に有効です。この場合、マスター定義サイトへの伝播は、拡張されていないレプリケーション・グループと拡張されているレプリケーション・グループの間で交互に切り替わります。1 サイクルごとに各グループに 1 回データがプッシュされます。この値は、1 サイクルの長さを秒単位で示します。

列	データ型	NULL	説明
PERCENTAGE_FOR_CATCHUP_NEW	NUMBER	-	この値は、BREAK_TRANS_TO_NEW_MASTERS が TRUE の場合のみ有効です。  新しいマスター・サイトへの伝播に追いつくために使用する必要のある、伝播リソースのパーセンテージです。
CYCLE_SECONDS_NEW	NUMBER	-	この値は、PERCENTAGE_FOR_CATCHUP_NEW が有効であり、かつその値が 10 ～ 90 (10 と 90 を含む) の間にある場合に有効です。この場合、新しいマスター・サイトへの伝播は、拡張されていないレプリケーション・グループと拡張されているレプリケーション・グループの間で交互に切り替わります。1 サイクルごとに各グループに 1 回データがプッシュされます。この値は、1 サイクルの長さを秒単位で示します。

## DBA\_REPGENOBJECTS

DBA\_REPGENOBJECTS ビューには、レプリケーションのサポート用に生成された、データベース内の各オブジェクトが示されます。このビューの列は、25-21 ページの「[ALL\\_REPGENOBJECTS](#)」の列と同じです。

## DBA\_REPGROUP

DBA\_REPGROUP には、データベース内のすべてのレプリケーション・グループが示されます。各レプリケーション・グループのメンバーは、DBA\_REPOBJECT ビューにリストされます。DBA\_REPGROUP ビューの列は、25-22 ページの「[ALL\\_REPGROUP](#)」の列と同じです。

## DBA\_REPGROUP\_PRIVILEGES

DBA\_REPGROUP\_PRIVILEGES ビューには、レプリケーション・グループ内で権限を登録されているユーザーの情報が示されます。データベース内のすべてのリフレッシュ・グループが表示されます。このビューの列は、25-23 ページの「[ALL\\_REPGROUP\\_PRIVILEGES](#)」の列と同じです。

## DBA\_REPGROUPED\_COLUMN

DBA\_REPGROUPED\_COLUMN ビューには、データベース内の各表の列グループを構成する、すべての列がリストされます。このビューの列は、25-24 ページの「[ALL\\_REPGROUPED\\_COLUMN](#)」の列と同じです。

## DBA\_REPKEY\_COLUMNS

DBA\_REPKEY\_COLUMNS には、データベース内の各表の 1 つ以上のレプリケーション・キー列が示されます。このビューの列は、25-25 ページの「[ALL\\_REPKEY\\_COLUMNS](#)」の列と同じです。

## DBA\_REPOBJECT

DBA\_REPOBJECT には、データベースの各レプリケーション・グループ内のオブジェクトの情報が示されます。オブジェクトは、1 つのレプリケーション・グループのみに属します。レプリケーション・グループは、複数のスキーマにまたがることができます。このビューの列は、25-26 ページの「[ALL\\_REPOBJECT](#)」の列と同じです。

## DBA\_REPPARAMETER\_COLUMN

DBA\_REPPARAMETER\_COLUMN ビューには、DBA\_REPRESOLUTION ビューに示される情報と、データベース内の各レプリケート表の競合解消に使用される列の情報が示されます。これらの列の値は、`list_of_column_names` の引数として、DBMS\_REPCAT パッケージの `ADD_conflicttype_RESOLUTION` プロシージャに渡されます。このビューの列は、25-28 ページの「[ALL\\_REPPARAMETER\\_COLUMN](#)」の列と同じです。

## DBA\_REPPRIORITY

DBA\_REPPRIORITY ビューには、データベース内の各優先グループの各メンバーの値および優先順位レベルが示されます。優先グループには、レプリケーション・グループ内で一意の名前を付ける必要があります。また、指定された優先グループ内のそれぞれの優先順位レベルと値は、そのグループ内で一意であることが必要です。このビューの列は、25-29 ページの「[ALL\\_REPPRIORITY](#)」の列と同じです。

## DBA\_REPPRIORITY\_GROUP

DBA\_REPPRIORITY\_GROUP には、データベース内の各レプリケーション・グループに定義された、優先グループまたはサイト優先グループが示されます。このビューの列は、25-30 ページの「[ALL\\_REPPRIORITY\\_GROUP](#)」の列と同じです。

## DBA\_REPPROP

DBA\_REPPROP ビューには、各レプリケート・オブジェクトに対する操作を、他のマスター・サイトにある同じオブジェクトに伝播する手法が示されます。操作の伝播は、ストアド・プロシージャやプロシージャ・ラップ・プログラムをコールしたり、表を直接操作することによって実行します。このビューに示されるのは、データベース内のすべてのオブジェクトです。このビューの列は、25-31 ページの「[ALL\\_REPPROP](#)」の列と同じです。

## DBA\_REPRESOL\_STATS\_CONTROL

DBA\_REPRESOL\_STATS\_CONTROL ビューには、データベース内のすべてのレプリケート表に対する、競合解消の統計収集が示されます。このビューの列は、25-32 ページの「[ALL\\_REPRESOL\\_STATS\\_CONTROL](#)」の列と同じです。

## DBA\_REPRESOLUTION

DBA\_REPRESOLUTION ビューには、データベース内の各レプリケート表での更新、一意性または削除の競合の解消に使用する方法が示されます。この場合の表は、指定されたスキーマに対する行レベル・レプリケーションによってレプリケートされています。このビューの列は、25-33 ページの「[ALL\\_REPRESOLUTION](#)」の列と同じです。

## DBA\_REPRESOLUTION\_METHOD

DBA\_REPRESOLUTION\_METHOD ビューには、データベース内で使用可能な競合解消メソッドがすべてリストされます。まず、このビューには、アドバンスト・レプリケーション機能で提供する標準の方法がリストされます。新しいユーザー・ファンクションを作成してデータベース内のオブジェクト用の競合解消メソッドとして追加すると、そのファンクションがこのビューに追加されます。このビューの列は、25-34 ページの「[ALL\\_REPRESOLUTION\\_METHOD](#)」の列と同じです。

## DBA\_REPRESOLUTION\_STATISTICS

DBA\_REPRESOLUTION\_STATISTICS ビューには、データベース内のすべてのレプリケート表に対して、正常に実行された更新、一意性または削除の競合解消の情報がリストされます。これらの統計情報は、DBMS\_REPCAT.REGISTER\_STATISTICS プロシージャをコールした場合にかぎり、表に対して収集されます。DBA\_REPRESOLUTION\_STATISTICS ビューの列は、25-34 ページの「[ALL\\_REPRESOLUTION\\_STATISTICS](#)」の列と同じです。

## DBA\_REPSITES

DBA\_REPSITES ビューには、データベース内の各レプリケーション・グループのメンバーがリストされます。

このビューには、ALL\_REPSITES ビューと USER\_REPSITES ビューには含まれていない次の列があります。

列	データ型	NULL	説明
PROP_UPDATES	NUMBER	-	伝播方法のエンコーディング（マスター・サイトの場合）。
MY_DBLINK	VARCHAR2 (1)	-	インポート後、問題を検出するために使用されます。Y の場合、この <code>dblink</code> がグローバル名になります。

前述の列の他に、25-35 ページの [ALL\\_REPSITES](#) と同じ列も表示されます。

## DBA\_REPSITES\_NEW

DBA\_REPSITES\_NEW ビューには、レプリケーション環境への追加を計画している新しいレプリケーション・サイトがリストされます。

**関連項目：** レプリケーション環境に新しいマスター・サイトを追加する  
プロシージャの詳細は、7-4 ページの「[マスター・グループを静止することなく、新しいマスター・サイトを追加](#)」を参照してください。

列	データ型	NULL	説明
EXTENSION_ID	RAW (16)	NOT NULL	静止せずにマスター・グループにマスター・データベースを追加する、カレントの保留要求の識別子。
GOWNER	VARCHAR2 (30)	NOT NULL	マスター・グループを所有するユーザーの名前。
GNAME	VARCHAR2 (30)	NOT NULL	マスター・グループの名前。
DBLINK	VARCHAR2 (128)	NOT NULL	新しいマスター・サイトのデータベース・リンク。
FULL_INSTANTIATION	VARCHAR2 (1)	-	Y は、データベース全体のエクスポート / インポートまたは変更ベースのリカバリを使用して、DBLINK の新しいデータベースが追加されることを示します。  N は、オブジェクト・レベルのエクスポート / インポートを使用して、DBLINK の新しいデータベースが追加されることを示します。



列	データ型	NULL	説明
MASTER_STATUS	VARCHAR2 (13)	-	<p>新しいマスター・サイトのインスタンス化ステータス。この値は、マスター定義サイトでのみ有効です。</p> <p>可能な値は次のとおりです。</p> <p>READY: 新しいマスター・サイトの準備が完了しています。</p> <p>INSTANTIATING: 新しいマスター・サイトのインスタンス化処理が実行されています。</p> <p>INSTANTIATED: 新しいマスター・サイトのインスタンス化が完了し、レプリケーション・アクティビティが準備されています。つまり、DBMS_REPCAT.PREPARE_INSTANTIATED_MASTER プロシージャが実行されています。</p> <p>PREPARED: 新しいマスター・サイトから準備が完了している他のマスター、既存のマスターおよびマスター定義サイトへの遅延トランザクションの伝播が使用可能になりました。これで、新しいマスター・サイトがレプリケーション環境に参加する準備が完了しました。</p>

## USER\_REPCAT\_REFRESH\_TEMPLATES

このビューには、カレント・ユーザーが所有する各配置テンプレートに関するグローバル情報が含まれます。テンプレート名、テンプレートの所有者、テンプレート・オブジェクトが属するリフレッシュ・グループ、テンプレートの種類（プライベートまたはパブリック）などの情報があります。

このビューの列は、ALL\_REPCAT\_REFRESH\_TEMPLATES の列と同じです。このビューおよび列の詳細は、25-6 ページの「[ALL\\_REPCAT\\_REFRESH\\_TEMPLATES](#)」を参照してください。

## USER\_REPCAT\_TEMPLATE\_OBJECTS

USER\_REPCAT\_TEMPLATE\_OBJECTS ビューには、カレント・ユーザーが所有する各配置テンプレートの各オブジェクト定義が示されます。REFRESH\_TEMPLATE\_NAME にターゲット・テンプレートを指定すると、各オブジェクトがテンプレートに追加されます。

このビューの列は、ALL\_REPCAT\_TEMPLATE\_OBJECTS の列と同じです。このビューおよび列の詳細は、25-7 ページの「[ALL\\_REPCAT\\_TEMPLATE\\_OBJECTS](#)」を参照してください。

## USER\_REPCAT\_TEMPLATE\_PARMS

カレント・ユーザーが所有するすべてのテンプレートのオブジェクト DDL で定義されたパラメータは、USER\_REPCAT\_TEMPLATE\_PARMS 表に格納されます。オブジェクトがテンプレートに追加されると、DDL に変数があるかどうかを検査され、パラメータが見つかりと自動的にこのビューに追加されます。

このビューの列は、ALL\_REPCAT\_TEMPLATE\_PARMS の列と同じです。このビューおよび列の詳細は、25-9 ページの「[ALL\\_REPCAT\\_TEMPLATE\\_PARMS](#)」を参照してください。

## USER\_REPCAT\_TEMPLATE\_SITES

USER\_REPCAT\_TEMPLATE\_SITES ビューは、エンタープライズ・ネットワークのサイトにおける、テンプレートのインスタンス化のカレント・ステータス情報をユーザーに提供します。カレント・ユーザーが所有している配置テンプレートのインスタンス化サイトの情報も示されます。具体的には、ユーザーは、特定のサイトで実行されるテンプレートのインストールや削除を監視できます。このビューの列は、25-11 ページの「[ALL\\_REPCAT\\_TEMPLATE\\_SITES](#)」の列と同じです。

## USER\_REPCAT\_USER\_AUTHORIZATION

USER\_REPCAT\_USER\_AUTHORIZATIONS ビューには、プライベート用に指定された、カレント・ユーザーが所有するすべてのテンプレートに対する権限を付与されたユーザーがリストされます。このビューにリストされたユーザーは、指定されたテンプレートをインスタンス化できます。このビューにリストされないユーザーはテンプレートをインスタンス化できません。このビューの列は、25-12 ページの「[ALL\\_REPCAT\\_USER\\_AUTHORIZATIONS](#)」の列と同じです。

## USER\_REPCAT\_USER\_PARM\_VALUES

USER\_REPCAT\_USER\_PARM\_VALUES ビューには、カレント・ユーザーが所有するすべての配置テンプレートのテンプレート・パラメータが示されます。DBA には、テンプレートを配布してインスタンス化する前に、ユーザー・パラメータの表を作成するオプションの機能があります。指定されたユーザーがテンプレートをインスタンス化する場合は、そのユーザー用に USER\_REPCAT\_USER\_PARM\_VALUES ビューに格納された値が自動的に使用されます。

このビューの列は、ALL\_REPCAT\_USER\_PARM\_VALUES の列と同じです。このビューおよび列の詳細は、25-13 ページの「[ALL\\_REPCAT\\_USER\\_PARM\\_VALUES](#)」を参照してください。

## USER\_REPCATLOG

各マスター・サイトの USER\_REPCATLOG ビューには、非同期管理要求および生成されたエラー・メッセージの暫定的なステータスの情報が示されます。要求の処理中に生成されたすべてのエラー・メッセージは、その要求を作成したマスター・サイトの USER\_REPCATLOG ビューに転送されます。エラーが発生せずに管理要求が正常に完了した場合は、この要求に関するすべてのトレースは、USER\_REPCATLOG ビューから削除されます。

このビューに含まれる非同期管理要求およびエラー・メッセージは、カレント・ユーザーが所有しています。このビューの列は、25-15 ページの「[ALL\\_REPCATLOG](#)」の列と同じです。

## USER\_REPCOLUMN

USER\_REPCOLUMN ビューには、カレント・ユーザーが所有するすべての表のレプリケートされた列がリストされます。このビューの列は、25-16 ページの「[ALL\\_REPCOLUMN](#)」の列と同じです。

## USER\_REPCOLUMN\_GROUP

USER\_REPCOLUMN\_GROUP ビューには、カレント・ユーザーが所有する各レプリケート表の列グループがリストされます。このビューの列は、25-19 ページの「[ALL\\_REPCOLUMN\\_GROUP](#)」の列と同じです。

---

---

**注意：** USER\_REPCOLUMN\_GROUP ビューには、SNAME 列が示されません。SNAME 列は、ALL\_REPCOLUMN\_GROUP ビューおよび DBA\_REPCOLUMN\_GROUP ビューで使用可能です。

---

---

## USER\_REPCONFLICT

USER\_REPCONFLICT ビューには、競合解消メソッドが定義されている、カレント・ユーザーが所有する各表の名前と、その方法で解消される競合のタイプが表示されます。このビューの列は、25-19 ページの「[ALL\\_REPCONFLICT](#)」の列と同じです。

---

---

**注意：** USER\_REPCONFLICT ビューには、SNAME 列が示されません。SNAME 列は、ALL\_REPCONFLICT ビューおよび DBA\_REPCONFLICT ビューで使用可能です。

---

---

## USER\_REPDDL

USER\_REPDDL ビューには、カレント・ユーザーが所有する各レプリケーション・オブジェクトの DDL が示されます。このビューの列は、25-20 ページの「[ALL\\_REPDDL](#)」の列と同じです。

## USER\_REPGENOBJECTS

USER\_REPGENOBJECTS ビューには、レプリケーションのサポート用に生成された、カレント・ユーザーが所有する各オブジェクトが示されます。このビューの列は、25-21 ページの「[ALL\\_REPGENOBJECTS](#)」の列と同じです。

## USER\_REPGROUP

USER\_REPGROUP ビューには、カレント・ユーザーが所有するすべてのレプリケーション・グループが示されます。各レプリケーション・グループのメンバーは、USER\_REPOBJECT ビューにリストされます。USER\_REPGROUP ビューの列は、25-22 ページの「[ALL\\_REPGROUP](#)」の列と同じです。

## USER\_REPGROUP\_PRIVILEGES

USER\_REPGROUP\_PRIVILEGES ビューには、レプリケーション・グループ内で権限を登録されているユーザーの情報が示されます。カレント・ユーザーが所有しているレプリケーション・グループのみが表示されます。このビューの列は、25-23 ページの「[ALL\\_REPGROUP\\_PRIVILEGES](#)」の列と同じです。

## USER\_REPGROUPED\_COLUMN

USER\_REPGROUPED\_COLUMN ビューには、各表の列グループを構成するすべての列がリストされます。このビューの列は、25-24 ページの「[ALL\\_REPGROUPED\\_COLUMN](#)」の列と同じです。

---

---

**注意：** USER\_REPGROUPED\_COLUMN ビューには、SNAME 列が示されません。SNAME 列は、ALL\_REPGROUPED\_COLUMN ビューおよび DBA\_REPGROUPED\_COLUMN ビューで使用可能です。

---

---

## USER\_REPKEY\_COLUMNS

USER\_REPKEY\_COLUMNS ビューには、カレント・ユーザーが所有する各表の 1 つ以上のレプリケーション・キー列が示されます。このビューの列は、25-25 ページの「[ALL\\_REPKEY\\_COLUMNS](#)」の列と同じです。

## USER\_REPOBJECT

USER\_REPOBJECT ビューには、カレント・ユーザーが所有する各レプリケーション・グループ内のオブジェクトの情報が示されます。オブジェクトは、1つのレプリケーション・グループのみに属します。レプリケーション・グループは、複数のスキーマにまたがることができます。このビューの列は、25-26 ページの「[ALL\\_REPOBJECT](#)」の列と同じです。

## USER\_REPPARAMETER\_COLUMN

USER\_REPPARAMETER\_COLUMN ビューには、USER\_REPRESOLUTION ビューに示される情報と、カレント・ユーザーが所有する各レプリケート表の競合解消に使用される列の情報が示されます。これらの列の値は、`list_of_column_names` の引数として、DBMS\_REPCAT パッケージの `ADD_conflicttype_RESOLUTION` プロシージャに渡されます。このビューの列は、25-28 ページの「[ALL\\_REPPARAMETER\\_COLUMN](#)」の列と同じです。

---

---

**注意：** USER\_REPPARAMETER\_COLUMN ビューには、SNAME 列が示されません。SNAME 列は、ALL\_REPPARAMETER\_COLUMN ビューおよび DBA\_REPPARAMETER\_COLUMN ビューで使用可能です。

---

---

## USER\_REPPRIORITY

USER\_REPPRIORITY ビューには、カレント・ユーザーが所有する各優先グループの各メンバーの値および優先順位レベルが示されます。優先グループには、レプリケーション・グループ内で一意の名前を付ける必要があります。また、指定された優先グループ内のそれぞれの優先順位レベルと値は、そのグループ内で一意であることが必要です。このビューの列は、25-29 ページの「[ALL\\_REPPRIORITY](#)」の列と同じです。

---

---

**注意：** USER\_REPPRIORITY ビューには、SNAME 列が示されません。SNAME 列は、ALL\_REPPRIORITY ビューおよび DBA\_REPPRIORITY ビューで使用可能です。

---

---

## USER\_REPPRIORITY\_GROUP

USER\_REPPRIORITY\_GROUP ビューには、カレント・ユーザーが所有する各レプリケーション・グループに定義された、優先グループまたはサイト優先グループが示されます。このビューの列は、25-30 ページの「[ALL\\_REPPRIORITY\\_GROUP](#)」の列と同じです。

## USER\_REPPROP

USER\_REPPROP ビューには、各レプリケート・オブジェクトに対する操作を、他のマスター・サイトにある同じオブジェクトに伝播する手法が示されます。操作の伝播は、ストアド・プロシージャやプロシージャ・ラップ・プログラムをコールしたり、表を直接操作することによって実行します。このビューに示されるのは、カレント・ユーザーが所有するオブジェクトです。このビューの列は、25-31 ページの「[ALL\\_REPPROP](#)」の列と同じです。

## USER\_REPRESOL\_STATS\_CONTROL

USER\_REPRESOL\_STATS\_CONTROL ビューには、カレント・ユーザーが所有するすべてのレプリケート表に対する、競合解消の統計収集が示されます。このビューの列は、25-32 ページの「[ALL\\_REPRESOL\\_STATS\\_CONTROL](#)」の列と同じです。

---

---

**注意：** USER\_REPRESOL\_STATS\_CONTROL ビューには、SNAME 列が示されません。SNAME 列は、ALL\_REPRESOL\_STATS\_CONTROL ビューおよび DBA\_REPRESOL\_STATS\_CONTROL ビューで使用可能です。

---

---

## USER\_REPRESOLUTION

USER\_REPRESOLUTION ビューには、カレント・ユーザーが所有する各レプリケート表での更新、一意性または削除の競合の解消に使用する方法が示されます。この場合の表は、指定されたスキーマに対する行レベル・レプリケーションによってレプリケートされています。このビューの列は、25-33 ページの「[ALL\\_REPRESOLUTION](#)」の列と同じです。

---

---

**注意：** USER\_REPREPRESOLUTION ビューには、SNAME 列が示されません。SNAME 列は、ALL\_REPREPRESOLUTION ビューおよび DBA\_REPREPRESOLUTION ビューで使用可能です。

---

---

## USER\_REPRESOLUTION\_METHOD

USER\_REPRESOLUTION\_METHOD ビューには、データベース内で使用可能な競合解消メソッドがすべてリストされます。まず、このビューには、アドバンスド・レプリケーション機能で提供する標準の方法がリストされます。新しいユーザー・ファンクションを作成してデータベース内のオブジェクト用の競合解消メソッドとして追加すると、そのファンクションがこのビューに追加されます。このビューの列は、25-34 ページの「[ALL\\_REPRESOLUTION\\_METHOD](#)」の列と同じです。

## USER\_REPRESOLUTION\_STATISTICS

USER\_REPRESOLUTION\_STATISTICS ビューには、カレント・ユーザーが所有するすべてのレプリケート表に対して、正常に実行された更新、一意性または削除の競合解消の情報がリストされます。これらの統計情報は、DBMS\_REPCAT.REGISTER\_STATISTICS プロシージャをコールした場合にかぎり、表に対して収集されます。

USER\_REPRESOLUTION\_STATISTICS ビューの列は、25-34 ページの「[ALL\\_REPRESOLUTION\\_STATISTICS](#)」の列と同じです。

---

---

**注意：** USER\_REPRESOLUTION\_STATISTICS ビューには、SNAME 列が示されません。SNAME 列は、ALL\_REPRESOLUTION\_STATISTICS ビューおよび DBA\_REPRESOLUTION\_STATISTICS ビューで使用可能です。

---

---

## USER\_REPSITES

USER\_REPSITES ビューには、カレント・ユーザーが所有する各レプリケーション・グループのメンバーがリストされます。このビューの列は、25-35 ページの「[ALL\\_REPSITES](#)」の列と同じです。





---

## レプリケーション動的パフォーマンス・ビュー

すべての Oracle インストールに、V\$ ビューと呼ばれる動的パフォーマンス・ビューが含まれています。この章ではこれについて説明します。このビューは、マスター・サイトおよびマテリアライズド・ビュー・サイトが、現在リフレッシュ中のマテリアライズド・ビューの情報や遅延トランザクション・キューに関する統計情報などを決定する際に使用します。

この章では、次のビューについて説明します。

- [V\\$MVREFRESH](#)
- [V\\$REPLPROP](#)
- [V\\$REPLQUEUE](#)

**関連項目：** [第 10 章「レプリケーション環境の監視」](#)

V\$MVREFRESH

現在リフレッシュ中のマテリアライズド・ビューの情報が含まれます。

列	データ型	説明
SID	NUMBER	セッションの識別子。
SERIAL#	NUMBER	セッションのオブジェクトの識別に使用される、セッションのシリアル番号。セッションが終了して別のセッションが同じセッション ID で開始した場合、セッション・レベルのコマンドが正しいセッション・オブジェクトに適用されることを保証します。
CURRMVOWNER	VARCHAR2 (31)	現在リフレッシュ中のマテリアライズド・ビューの所有者。マテリアライズド・ビューはこのユーザーのスキーマに常駐します。
CURRMVNAME	VARCHAR2 (31)	現在リフレッシュ中のマテリアライズド・ビューの名前。

V\$REPLPROP

現在レプリケーション・サイトで進行中のパラレル伝播の情報が含まれます。このビューは、現在伝播中のトランザクション、各トランザクションで伝播される通話数、およびパラレル伝播のスレーブ・プロセスまたはコーディネータ・プロセスのカレント・アクティビティの決定に使用します。

**注意：** このビューには、カレント・サイトで遅延トランザクションがパラレル伝播を使用してプッシュされているときにのみデータが含まれます。パラレル伝播を使用してプッシュを行うには、parallelism パラメータを DBMS\_DEFER\_SYS.PUSH ファンクションで 1 以上に設定する必要があります。この設定を行わないと、プッシュはシリアル伝播を使用して行われ、プッシュ中にこのビューにはデータは表示されません。

列	データ型	説明
SID	NUMBER	セッションの識別子。
SERIAL#	NUMBER	セッションのシリアル番号。セッションのオブジェクトの識別に使用されます。セッションが終了して別のセッションが同じセッション ID で開始した場合、セッション・レベルのコマンドが正しいセッション・オブジェクトに適用されることを保証します。
NAME	VARCHAR2 (71)	<p>Replication Parallel Prop Slave n はスレーブ・プロセスがアクティブで、待機中、遅延トランザクションのプッシュ中、メタデータのバージ中、またはエラー・トランザクションの作成中のいずれかであることを示します。</p> <p>Replication Parallel Prop Coordinator はコーディネータ・プロセスがアクティブで、待機中、スリープ中、またはスレーブの操作実行をスケジュール中のいずれかであることを示します。</p> <p>Replication Parallel Prop Coordinator は遅延トランザクション・キューからトランザクションを読み取り、Replication Parallel Prop Slaves に割り当てます。その後スレーブがトランザクションを接続先サイトに伝播します。スレーブがプッシュ・セッションでトランザクションをプッシュする際は、スレーブはプッシュするトランザクションがなくなっても、プッシュ・セッションが完了するまでアクティブ状態を続けます。</p>
DBLINK	VARCHAR2 (128)	このレプリケーション・セッションの伝播先のデータベース・リンク。
STATE	VARCHAR2 (12)	<p>WAIT はスレーブ・プロセスまたはコーディネータ・プロセスがイベント（メッセージ）待ち状態であることを示します。</p> <p>SLEEP はコーディネータ・プロセスが、delay_seconds の設定時間の間はスリープ状態であることを示します。 delay_seconds は DBMS_DEFER_SYS パッケージの SCHEDULE_PUSH プロシージャで設定します。</p> <p>PUSH はスレーブ・プロセスが、遅延トランザクション・キューのトランザクションをリモート・サイトにプッシュ中であることを示します。</p> <p>PURGE はスレーブ・プロセスが、リモート・サイトから正常に適用されたトランザクションに関連するメタデータをバージ中であることを示します。</p> <p>CREATE ERROR はスレーブ・プロセスがエラー・トランザクションを作成中であることを示します。この場合、エラーまたは競合はスレーブ・プロセスが遅延トランザクションをリモート・サイトにプッシュする際に発生したものです。</p> <p>SCHEDULE TXN はコーディネータ・プロセスが、トランザクションの適用順序を決定中であり、かつトランザクションを実行するスレーブ・プロセスを割り当て中であることを示します。</p>

列	データ型	説明
XID	VARCHAR2 (22)	セッションがスレーブ・セッションの場合は、スレーブが現在伝播中のトランザクションのトランザクション ID を示します。
SEQUENCE	NUMBER	プロセスがスレーブ・プロセスの場合、現在の操作で伝播される通話があればその順序番号。各トランザクションは 1 つ以上の通話を処理する必要があり、SEQUENCE の値は 0（ゼロ）から始まって各通話が処理されるごとに増加します。つまり SEQUENCE の値は、各トランザクションで現在処理中の通話を示しています。この値はスレーブ・プロセスがトランザクション内の全通話を処理するまで増加し続けます。

V\$REPLQUEUE

レプリケーションの遅延トランザクション・キューに関する統計が含まれます。現行データベース・インスタンスの開始からすべての値が保存されます。

列	データ型	説明
TXNS_ENQUEUED	NUMBER	遅延トランザクション・キューにエンキューされているトランザクションの数。
CALLS_ENQUEUED	NUMBER	遅延トランザクション・キューにエンキューされる通話の数。
TXNS_PURGED	NUMBER	遅延トランザクション・キューからパージされるトランザクションの数。
LAST_ENQUEUE_TIME	DATE	最後のトランザクションが遅延トランザクション・キューにエンキューされた日付。インスタンスの開始以降、遅延トランザクション・キューにトランザクションが 1 つもエンキューされていない場合は、この値は NULL になります。
LAST_PURGE_TIME	DATE	最後のトランザクションが遅延トランザクション・キューからパージされた日付。インスタンスの開始以降、遅延トランザクション・キューからトランザクションが 1 つもパージされていない場合は、この値は NULL になります。

---

## 遅延トランザクション・ビュー

---

Oracle には、遅延トランザクションを管理するためのビューがいくつか用意されています。これらのビューは、トランザクションの接続先、トランザクションを構成する遅延コール、トランザクションの処理中に発生するエラーなど、遅延トランザクションに関する情報を提供します。

この章では、次のビューについて説明します。

- [DEFCALL](#)
- [DEFCALLDEST](#)
- [DEFDEFAULTDEST](#)
- [DEFERRCOUNT](#)
- [DEFERROR](#)
- [DEFLOB](#)
- [DEFPROPAGATOR](#)
- [DEFSCHEDULE](#)
- [DEFTRAN](#)
- [DEFTRANDEST](#)

---

**注意：** 表を直接変更しないでください。変更が必要な場合は、DBMS\_DEFER および DBMS\_DEFER\_SYS パッケージに用意されているプロシージャを使用してください。

---

**関連項目：** [第 10 章「レプリケーション環境の監視」](#)

DEFCALL

すべての遅延リモート・プロシージャ・コールが記録されます。

非同期レプリケーションを使用してキューに配置されたコールについては、3 つ以上連続する NULL を含む列オブジェクトおよびオブジェクト表には NULL 圧縮が使用されます。そのため、このビューには列オブジェクトの属性やオブジェクト表の列が合計より少なく表示される場合があります。たとえば NULL 圧縮のため、8 つの属性を持つ列オブジェクトに 5 つの属性しか表示されない場合があります。

NULL 圧縮はエラー・トランザクションには適用されません。

列	データ型	NULL	説明
CALLNO	NUMBER	-	トランザクション処理中のコールの一意な ID。
DEFERRED_TRAN_ID	VARCHAR2 (30)	-	対応付けられたトランザクションの一意な ID。
SCHEMANAME	VARCHAR2 (30)	-	遅延コールのスキーマ名。
PACKAGENAME	VARCHAR2 (30)	-	遅延コールのパッケージ名。レプリケート表では、表名を示す場合があります。
PROCNAME	VARCHAR2 (30)	-	遅延コールのプロシージャ名。レプリケート表では、操作名を示す場合があります。
ARGCOUNT	NUMBER	-	遅延コールの引数の数。

DEFCALLDEST

各遅延リモート・プロシージャ・コールの接続先がリストされます。

列	データ型	NULL	説明
CALLNO	NUMBER	NOT NULL	トランザクション処理中のコールの一意な ID。
DEFERRED_TRAN_ID	VARCHAR2 (30)	NOT NULL	DEFTRAN ビューの DEFERRED_TRAN_ID 列に対応しています。各遅延トランザクションには、1 つ以上の遅延コールが含まれます。
DBLINK	VARCHAR2 (128)	NOT NULL	接続先データベースの完全修飾名。

# DEFDEFAULTDEST

Advanced Replication を使用せず、遅延トランザクションの接続先や、トランザクション処理中のコールを指定しない場合、Oracle は DEFDEFAULTDEST ビューを使用して、リモート・プロシージャ・コールを遅らせるための接続先データベースを判別します。

列	データ型	NULL	説明
DBLINK	VARCHAR2 (128)	NOT NULL	トランザクションのレプリケート先の、完全修飾されたデータベース名。

# DEFERRCOUNT

接続先に対するエラー・トランザクションの情報が含まれます。

列	データ型	NULL	説明
ERRCOUNT	NUMBER	-	接続先でエラー原因となった既存のトランザクションの数。
DESTINATION	VARCHAR2 (128)	-	接続先を指定するためのデータベース・リンク。

# DEFERROR

適用できなかった各トランザクションの ID が含まれます。この ID を使用して、このトランザクションに対応付けられたキューに入れられたコールを指定できます。これらのコールは、DEFCALL ビューに格納されています。DBMS\_DEFER\_QUERY パッケージのプロシージャを使用して、DEFCALL ビューにリストされているプロシージャに対する引数を指定できます。

列	データ型	NULL	説明
DEFERRED_TRAN_ID	VARCHAR2 (22)	NOT NULL	エラー原因となっているトランザクションの ID。
ORIGIN_TRAN_DB	VARCHAR2 (128)	-	遅延トランザクションを作成したデータベース。
ORIGIN_TRAN_ID	VARCHAR2 (22)	-	トランザクションの元の ID。
CALLNO	NUMBER	-	DEFERRED_TRAN_ID のコールの一意な ID。
DESTINATION	VARCHAR2 (128)	-	接続先を指定するためのデータベース・リンク。
START_TIME	DATE	-	元のトランザクションがエンキューされた時刻。
ERROR_NUMBER	NUMBER	-	Oracle エラー番号。
ERROR_MSG	VARCHAR2 (2000)	-	エラー・メッセージのテキスト。
RECEIVER	VARCHAR2 (30)	-	遅延トランザクションの元の受信者。

# DEFLOB

遅延リモート・プロシージャ・コール（RPC）への LOB パラメータが含まれます。

列	データ型	NULL	説明
ID	RAW (16)	NOT NULL	LOB パラメータの識別子。
DEFERRED_TRAN_ID	VARCHAR2 (22)	-	この LOB パラメータを使用した遅延リモート・プロシージャ・コール（RPC）のトランザクション ID。
BLOB_COL	BLOB (4000)	-	LOB パラメータ（バイナリ）。
CLOB_COL	CLOB (4000)	-	LOB パラメータ（キャラクタ）。
NCLOB_COL	NCLOB (4000)	-	LOB パラメータ（各国語キャラクタ）。

# DEFPROPAGATOR

ローカルなプロパゲータの情報が含まれます。

列	データ型	NULL	説明
USERNAME	VARCHAR2 (30)	NOT NULL	プロパゲータのユーザー名。
USERID	NUMBER	NOT NULL	プロパゲータのユーザー ID。
STATUS	VARCHAR2 (7)	-	プロパゲータのステータス。
CREATED	DATE	NOT NULL	プロパゲータが登録された時刻。



## DEFSCHEDULE

次にスケジュールされているジョブの実行時刻に関する情報、および伝播統計が含まれます。伝播統計は、カレント・サイトから DBLINK 列で指定されたサイトへの遅延トランザクションの伝播に関するものです。

リモート・サイトの伝播統計を消去して最初から始めるには、DBMS\_DEFER\_SYS パッケージの CLEAR\_PROP\_STATISTICS プロシージャを使用します。

**関連項目：** 14-5 ページ「[CLEAR\\_PROP\\_STATISTICS プロシージャ](#)」

列	データ型	NULL	説明
DBLINK	VARCHAR2 (128)	NOT NULL	遅延リモート・プロシージャ・コールを定期的に行うマスター・サイトへの完全修飾パス名。
JOB	NUMBER	-	DBMS_DEFER_SYS.SCHEDULE_PUSH をコールして作成したジョブに割り当てられる番号。ジョブの実行内容を判別するには、USER_JOBS ビューの WHAT 列に問い合わせます。
INTERVAL	VARCHAR2 (200)	-	次回遅延トランザクション・キューをプッシュする時刻を計算するためのファンクション。
NEXT_DATE	DATE	-	次のジョブ実行日。
LAST_DATE	DATE	-	前回キューがプッシュされた日付、または前回この送信先に対してリモート・プロシージャ・コールのプッシュを試みた日付。
DISABLED	CHAR (1)	-	Y の場合、接続先への伝播は無効です。 N の場合、接続先への伝播は有効です。
LAST_TXN_COUNT	NUMBER	-	前回の実行でプッシュしたトランザクションの数。
LAST_ERROR_NUMBER	NUMBER	-	前回のプッシュでの Oracle エラー番号。
LAST_ERROR_MESSAGE	VARCHAR2 (2000)	-	前回のプッシュでのエラー・メッセージ。
CATCHUP	RAW (16)	NOT NULL	マスター・グループを静止せずに追加される新しいマスター・サイトに対応付けられた拡張識別子。マスター・サイトに拡張識別子がない場合、この値は 00 になります。
TOTAL_TXN_COUNT	NUMBER	-	正常終了したトランザクションとエラー・トランザクションの合計数。

列	データ型	NULL	説明
AVG_THROUGHPUT	NUMBER	-	パラレル伝播を使用して 1 秒間に伝播されるトランザクションの平均数。トランザクションには、正常に適用されたトランザクションとリモート・サイトで作成されたエラー・トランザクションの両方が含まれます。伝播コーディネータがアクティブでなかった（スリープ状態だった）間の経過時間も、計算に含まれます。
AVG_LATENCY	NUMBER	-	<p>トランザクションがリモート・サイトで正常に適用された場合、カレント・サイトでのトランザクションの最初のコールからリモート・サイトでトランザクションが適用されたという確認までの平均秒数。最初のコールはトランザクションがコミットされた時点ではなく、ユーザーが最初にデータ操作言語（DML）に変更を加えた時点から開始されます。</p> <p>トランザクションがエラー・トランザクションの場合、カレント・サイトでのトランザクションの最初のコールからリモート・サイトでエラー・トランザクションがコミットされたという確認までの平均秒数。</p>
TOTAL_BYTES_SENT	NUMBER	-	レプリケート・データおよびメタデータを含む、送信されたバイトの合計数。
TOTAL_BYTES_RECEIVED	NUMBER	-	伝播の確認メッセージで受信されたバイトの合計数。
TOTAL_ROUND_TRIPS	NUMBER	-	データをレプリケートする際に完了した、ネットワークのラウンドトリップの合計数。ラウンドトリップとは、連続して受信される 1 つ以上のメッセージが後に続く、連続して送信される 1 つ以上のメッセージです。つまり、サイト A がサイト B に 20 のメッセージを送り、次にサイト B がサイト A に 1 つのメッセージを送ると、それで 1 つのラウンドトリップになります。
TOTAL_ADMIN_COUNT	NUMBER	-	受信サイトで適用されたトランザクション情報のメンテナンスのため送信された管理要求の合計数。受信サイトは、DBLINK 列で指定されたサイトになります。この特別な管理はパラレル伝播でのみ必要です。
TOTAL_ERROR_COUNT	NUMBER	-	リモート・エラーが作成された、競合解消不可能な競合の合計数。
TOTAL_SLEEP_TIME	NUMBER	-	伝播コーディネータがアクティブでなかった（スリープ状態だった）合計秒数。伝播コーディネータがスリープ状態になる時間の長さは、DBMS_DEFER_SYS.PUSH ファンクションの delay_seconds パラメータを使用して制御します。

列	データ型	NULL	説明
DISABLED_INTERNALLY_SET	VARCHAR2 (1)	-	<p>この値は DISABLED が Y の場合のみ使用できます。</p> <p>DISABLED_INTERNALLY_SET が Y の場合は、伝播の同期については Oracle 内部で、マスター・グループを静止せずに新しいマスター・サイトを追加する際は接続先への伝播が使用禁止に設定されています。伝播はのちに自動的に使用可能になります。</p> <p>DISABLED_INTERNALLY_SET が N の場合は、伝播は内部で使用禁止にはされていません。</p>

## DEFTRAN

カレント・サイトで遅延トランザクション・キューにある遅延トランザクションをすべて記録します。

列	データ型	NULL	説明
DEFERRED_TRAN_ID	VARCHAR2 (30)	-	コールをエンキューしたトランザクション ID。
DELIVERY_ORDER	NUMBER	-	キュー内の遅延トランザクションの順序を決定する識別子。識別子は、元のトランザクションのシステム変更番号 (SCN) から派生しています。
DESTINATION_LIST	VARCHAR2 (1)	-	<p>R の場合、接続先は ALL_REPSITES ビューによって決定されます。</p> <p>D の場合、接続先は DEFDEFAULTDEST ビュー、あるいは TRANSACTION プロシージャまたは CALL プロシージャの NODE_LIST 引数によって決定されます。</p>
START_TIME	DATE	-	元のトランザクションがエンキューされた時刻。

DEFTRANDEST

カレント・サイトで遅延トランザクション・キューにある各遅延トランザクションの接続先をリストします。

列	データ型	NULL	説明
DEFERRED_TRAN_ID	VARCHAR2 (30)	NOT NULL	指定されたデータベース・リンクにレプリケートするトランザクションの ID。
DELIVERY_ORDER	NUMBER	-	キュー内の遅延トランザクションの順序を決定する識別子。識別子は、元のトランザクションの SCN から派生しています。
DBLINK	VARCHAR2 (128)	NOT NULL	接続先データベースの完全修飾名。

---

## マテリアライズド・ビューとリフレッシュ・グループ・ビュー

この章では、マテリアライズド・ビューおよびマテリアライズド・ビューのリフレッシュ・グループの情報を提供するデータ・ディクショナリ・ビューをリストします。

---

ALL_ ビュー	DBA_ ビュー	USER_ ビュー
ALL_BASE_TABLE_MVIEWS	DBA_BASE_TABLE_MVIEWS	USER_BASE_TABLE_MVIEWS
-	DBA_MVIEW_LOG_FILTER_COLS	-
ALL_MVIEW_LOGS	DBA_MVIEW_LOGS	USER_MVIEW_LOGS
ALL_MVIEW_REFRESH_TIMES	DBA_MVIEW_REFRESH_TIMES	USER_MVIEW_REFRESH_TIMES
ALL_MVIEWS	DBA_MVIEWS	USER_MVIEWS
-	DBA_RCHILD	-
ALL_REFRESH	DBA_REFRESH	USER_REFRESH
ALL_REFRESH_CHILDREN	DBA_REFRESH_CHILDREN	USER_REFRESH_CHILDREN
ALL_REGISTERED_MVIEWS	DBA_REGISTERED_MVIEWS	USER_REGISTERED_MVIEWS
-	DBA_RGROUP	-

---

**関連項目：** [第 10 章「レプリケーション環境の監視」](#)

# 第 V 部

## 付録

第 V 部は次の付録で構成されています。

- 付録 A「セキュリティ・オプション」
- 付録 B「ユーザー定義の競合解消メソッド」





---

## セキュリティ・オプション

この付録では、次の項目を説明します。

- マルチマスター・レプリケーションのセキュリティ設定
- マテリアライズド・ビュー・レプリケーションのセキュリティ設定

## マルチマスター・レプリケーションのセキュリティ設定

ほとんどの場合、マルチマスター・レプリケーションのセキュリティを設定するときは、レプリケーション管理ツールのセットアップ・ウィザードを使用するのが最も簡単です。ただし、レプリケーション・マネージメント API を使用してこれらの設定を行うことが必要な場合もあります。

レプリケーション環境を構成するには、データベース管理者が DBA 権限で接続して、必要な権限をレプリケーション管理者に付与する必要があります。

最初に、各マスター・サイトで、レプリケーション環境を構成およびメンテナンスする権限と、レプリケートされた変更を伝播および適用する権限を持つユーザー・アカウントを設定します。また、各マスター・サイトでユーザー用のリンクを定義する必要があります。

レプリケーション環境のユーザーのカテゴリとして、レプリケート・オブジェクトにアクセスするエンド・ユーザーに加え、次の 3 つの特別なカテゴリがあります。

- レプリケーション環境の構成およびメンテナンスを行う、レプリケーション管理者
- 遅延トランザクションを伝播する、プロパゲータ
- これらのトランザクションを適用する、リモート・サイトの受信者

通常は、1 人のユーザーが管理者、プロパゲータおよび受信者の役割を果たします。ただし、機能ごとに別々のユーザーを設定することも可能です。1 人のグローバル・レプリケーション管理者のみを設定できれば、レプリケーション・グループが複数のスキーマにまたがっていない場合は、異なるスキーマごとに別々のレプリケーション管理者を設定できます。ただし、各データベースに登録できるプロパゲータは 1 人です。

A-4 ページの表 A-1 に、これらの特別なアカウントに割り当てる必要がある権限を示します。これらのユーザーに必要な権限のほとんどは、レプリケーション・マネージメント API へのコールを通じて付与されます。また、CONNECT や RESOURCE など一部の権限は、直接付与する必要があります。

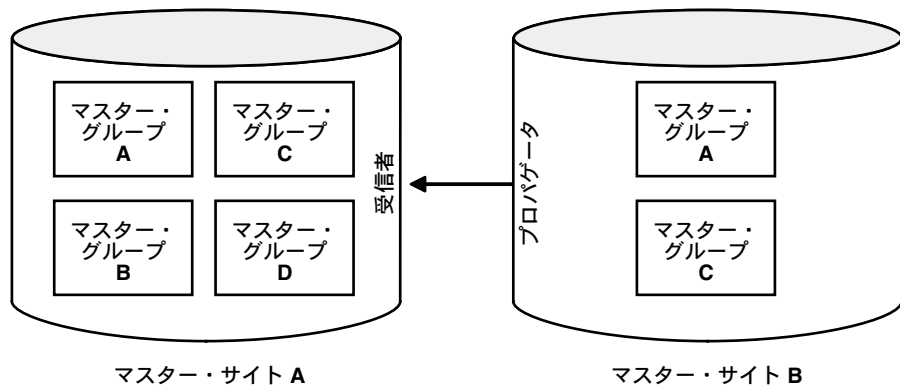
## トラステッド・セキュリティとアントラステッド・セキュリティの比較

ユーザーのカテゴリの他に、セキュリティ・モデルとしてトラステッドとアントラステッドのどちらを実装するかを決定する必要があります。トラステッド・セキュリティ・モデルでは、受信者がすべてのローカル・マスター・グループにアクセスできます。受信者は、リモート・サイトのプロパゲータにかわってローカル・マスター・サイトでデータベース・アクティビティを実行するので、プロパゲータにも、受信者のサイトのすべてのマスター・グループへのアクセス権限があります。1人の受信者をすべての受信トランザクションに使用することに注意してください。

たとえば、[図 A-1](#) の場合を考えます。マスター・サイト B にはマスター・グループ A と C のみが存在しますが、トラステッド・セキュリティ・モデルが使用されているので、プロパゲータは、マスター・サイト A のマスター・グループ A、B、C および D にアクセスできます。その結果、リモート・データベース管理の可動性によりデータベース管理の柔軟性が大幅に向上します。ただし、悪意のあるユーザーがリモート・サイトからマスター・サイトのデータを見たり、破損させたりする可能性も高くなります。

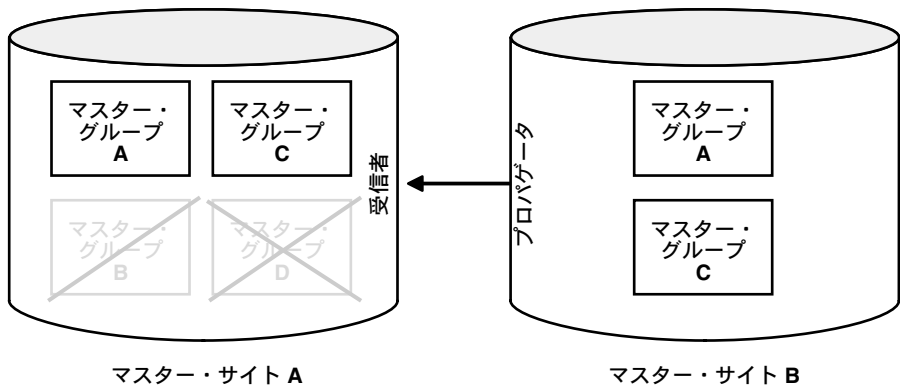
使用するセキュリティ・モデルに関係なく、Oracle では、オブジェクトがレプリケーション環境に追加されると、またはレプリケーション環境から削除されるときに、オブジェクトに適切な権限が自動的に付与されます。

**図 A-1** トラステッド・セキュリティ：マルチマスター・レプリケーション



アントラステッド・セキュリティでは、特定のマスター・グループの使用に必要な権限のみが受信者に割り当てられます。したがって、プロパゲータは、受信者側のローカルなマスター・グループのうち、特定のグループのみにアクセスできます。[図 A-2](#) にアントラステッド・セキュリティ・モデルを示します。マスター・サイト B には、マスター・グループ A と C のみが含まれるため、マスター・サイト A の受信者はマスター・グループ A と C に対する権限のみが付与されています。このため、プロパゲータからマスター・サイト A へのアクセスは制限されます。

図 A-2 アントラステッド・セキュリティ：マルチマスター・レプリケーション



一般にマスター・サイトは信頼できるとみなされるので、トラステッド・セキュリティ・モデルが使用されます。ただし、リモート・マスター・サイトが信頼できない場合は、アントラステッド・モデルを使用して、受信者に割り当てる権限を制限できます。サイトが信頼できないとみなされるのは、たとえばコンサルティング会社が複数の顧客の仕事を担当する場合などです。表 A-1 の受信者の項目に示す適切な API コールを使用して、ユーザーごとに適切な権限を割り当てます。

表 A-1 必要なユーザー・アカウント

ユーザー	権限
グローバル・レプリケーション 管理者	DBMS_REPCAT_ADMIN.GRANT_ADMIN_ANY_SCHEMA
スキーマレベルのレプリケーション 管理者	DBMS_REPCAT_ADMIN.GRANT_ADMIN_SCHEMA
プロパゲータ	DBMS_DEFER_SYS.REGISTER_PROPAGATOR
受信者	詳細は、21-4 ページの「 <a href="#">REGISTER_USER_REPGROUP プロシージャ</a> 」を参照してください。  トラステッド： DBMS_REPCAT_ADMIN.REGISTER_USER_REPGROUP privilege => 'receiver' list_of_gnames => NULL  アントラステッド： DBMS_REPCAT_ADMIN.REGISTER_USER_REPGROUP privilege => 'receiver' list_of_gnames => 'mastergroupname'

これらのアカウントを作成し、適切な権限を割り当てた後、各サイト間でユーザー名とパスワードを含む次のプライベート・データベース・リンクを作成します。

- ローカルのレプリケーション管理者からリモートのレプリケーション管理者へのリンク
- ローカルのプロパゲータからリモートの受信者へのリンク

1つのユーザー・アカウントがレプリケーション管理者、プロパゲータおよび受信者として機能するように指定した場合は、N (N-1) 個のリンクを作成する必要があります。Nはレプリケーション環境内のマスター・サイトの数です。

これらのリンクを作成した後、各ロケーションで DBMS\_DEFER\_SYS.SCHEDULE\_PUSH および DBMS\_DEFER\_SYS.SCHEDULE\_PURGE をコールして、遅延トランザクション・キューを各リモート・ロケーションに伝播する頻度、およびこのキューをパージする頻度を定義する必要があります。DBMS\_DEFER\_SYS.SCHEDULE\_PUSH は、各サイトで、1つのリモート・ロケーションに対し1回ずつ、複数回コールする必要があります。

hq.world と sales.world 間のマルチマスター・レプリケーションを設定するためのスクリプトの例を次に示します。

```

/*--- Create global replication administrator at HQ ---*/
connect system/manager@hq.world
create user repadmin identified by repadmin
execute dbms_repcat_admin.grant_admin_any_schema(username => 'repadmin')

/*--- Create global replication administrator at Sales ---*/
connect system/manager@sales.world
create user repadmin identified by repadmin
execute dbms_repcat_admin.grant_admin_any_schema(username => 'repadmin')

/*--- Create single user to act as both propagator and receiver at HQ ---*/
connect system/manager@hq.world
create user prop_rec identified by prop_rec
/*--- Grant privileges necessary to act as propagator ---*/
execute dbms_defer_sys.register_propagator(username => 'prop_rec')
/*--- Grant privileges necessary to act as receiver ---*/
execute dbms_repcat_admin.register_user_repgroup(
    username => 'prop_rec',
    privilege_type => 'receiver',
    list_of_gnames => NULL)

/*--- Create single user to act as both propagator and receiver at Sales ---*/
connect system/manager@sales.world
create user prop_rec identified by prop_rec
/*--- Grant privileges necessary to act as propagator ---*/execute
dbms_defer_sys.register_propagator(username => 'prop_rec')
/*--- Grant privileges necessary to act as receiver ---*/
execute dbms_repcat_admin.register_user_repgroup(
    username => 'prop_rec',

```

```
        privilege_type => 'receiver',
        list_of_gnames => NULL)

/*--- Create public link from HQ to Sales with necessary USING clause ---*/
connect system/manager@hq.world
create public database link sales.world using sales.world

/*--- Create private repadmin to repadmin link ---*/
connect repadmin/repadmin@hq.world
create database link sales.world connect to repadmin identified by repadmin

/*--- Schedule replication from HQ to Sales ---*/
execute dbms_defer_sys.schedule_push(
    destination => 'sales.world',
    interval => 'sysdate + 1/24',
    next_date => sysdate,
    stop_on_error => FALSE,
    parallelism => 1)

/*--- Schedule purge of def tran queue at HQ ---*/
execute dbms_defer_sys.schedule_purge(
    next_date => sysdate,
    interval => 'sysdate + 1',
    delay_seconds => 0,
    rollback_segment => '')

/*--- Create link from propagator to receiver for scheduled push ---*/
connect prop_rec/prop_rec@hq.world
create database link sales.world connect to prop_rec identified by prop_rec

/*--- Create public link from Sales to HQ with necessary USING clause ---*/
connect system/manager@sales.world
create public database link hq.world using hq.world

/*--- Create private repadmin to repadmin link ---*/
connect repadmin/repadmin@sales.world
create database link hq.world connect to repadmin identified by repadmin

/*--- Schedule replication from Sales to HQ ---*/
execute dbms_defer_sys.schedule_push(
    destination => 'hq.world',
    interval => 'sysdate + 1/24',
    next_date => sysdate,
    stop_on_error => FALSE,
    parallelism => 1)
```

```
/*--- Schedule purge of def tran queue at Sales ---*/
execute dbms_defer_sys.schedule_purge(
    next_date => sysdate,
    interval => 'sysdate + 1',
    delay_seconds => 0,
    rollback_segment => '')

/*--- Create link from propagator to receiver for scheduled push ---*/
connect prop_rec/prop_rec@sales.world
create database link hq.world connect to prop_rec identified by prop_rec
```

## マテリアライズド・ビュー・レプリケーションのセキュリティ設定

ほとんどの場合、マテリアライズド・ビュー・レプリケーションのセキュリティを設定するときは、レプリケーション管理ツールのセットアップ・ウィザードを使用するのが最も簡単です。ただし、一部の特別なケースでは、レプリケーション・マネージメント API を使用してこれらの設定を行う必要があります。レプリケーション環境を構成するには、データベース管理者が DBA 権限で接続して、必要な権限をレプリケーション管理者に付与する必要があります。

最初に、各マテリアライズド・ビュー・サイトで、レプリケーション環境を構成およびメンテナンスする権限と、レプリケートされた変更を伝播する権限を持つユーザー・アカウントを設定します。また、これらのユーザー用に、対応付けられたマスター・サイトまたはマスター・マテリアライズド・ビュー・サイトへのリンクを定義する必要があります。対応付けられたマスター・サイトまたはマスター・マテリアライズド・ビュー・サイトで追加のユーザーを作成するか、ユーザーへ追加の権限を割り当てるが必要な場合もあります。

マテリアライズド・ビュー・サイトのユーザーのカテゴリとして、レプリケート・オブジェクトにアクセスするエンド・ユーザーに加え、次の 3 つの特別なカテゴリがあります。

- レプリケーション環境の構成およびメンテナンスを行う、レプリケーション管理者
- 遅延トランザクションを伝播する、プロパゲータ
- 対応付けられたマスター表またはマスター・マテリアライズド・ビューからマテリアライズド・ビューへ変更をプルダウンする、リフレッシュャ。

通常は、これらの機能を 1 人のユーザーが実行します。ただし、これらの機能をそれぞれ別のユーザーが実行する場合もあります。たとえば、マテリアライズド・ビューは、マテリアライズド・ビュー・サイトの管理者が作成して、他のエンド・ユーザーがリフレッシュします。

表 A-2 に、マテリアライズド・ビュー・サイトの作成とメンテナンスに必要な権限を示します。

表 A-2 マテリアライズド・ビュー・サイトに必要なユーザー・アカウント

ユーザー	権限
マテリアライズド・ビュー・サイト・レプリケーション管理者	DBMS_REPCAT_ADMIN.GRANT_ADMIN_ANY_SCHEMA
プロパゲータ	DBMS_DEFER_SYS.REGISTER_PROPAGATOR
リフレッシャ	CREATE ANY MATERIALIZED VIEW ALTER ANY MATERIALIZED VIEW

マテリアライズド・ビュー・サイトに適切なユーザーを作成することに加え、対応付けられたマスター・サイトまたはマスター・マテリアライズド・ビュー・サイトにも追加ユーザーを作成する必要があります。A-10 ページの表 A-3 に、マスター・サイトまたはマスター・マテリアライズド・ビュー・サイトのユーザーが、新しいマテリアライズド・ビュー・サイトをサポートするために必要な権限を示します。

## トラステッド・セキュリティとアントラステッド・セキュリティの比較

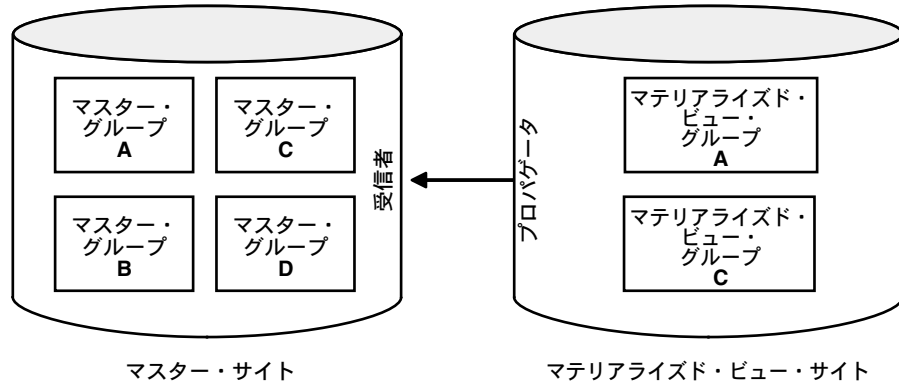
マスター・サイトまたはマスター・マテリアライズド・ビュー・サイトのユーザーのカテゴリの他に、セキュリティ・モデルとしてトラステッドとアントラステッドのどちらを実装するのかを決定する必要があります。トラステッド・セキュリティ・モデルでは、受信者およびプロキシ・マテリアライズド・ビュー管理者がすべてのローカル・レプリケーション・グループにアクセスできます。受信者およびプロキシ・マテリアライズド・ビュー管理者は、プロパゲータおよびマテリアライズド・ビュー管理者のかわりにローカル・マスター・サイトまたはマスター・マテリアライズド・ビュー・サイトのデータベース・アクティビティをリモート・マテリアライズド・ビュー・サイトで実行します。このため、リモート・マテリアライズド・ビュー・サイトのプロパゲータおよびマテリアライズド・ビュー管理者もまた、マスター・サイトまたはマスター・マテリアライズド・ビュー・サイトのすべてのレプリケーション・グループにアクセスできます。1 人の受信者をすべての受信トランザクションに使用することに注意してください。

たとえば、図 A-3 の場合を考えます。マテリアライズド・ビュー・サイトにはマテリアライズド・ビュー・グループ A と C が存在しますが（マスター・サイトのマスター・グループ A と C をベースとしています）、トラステッド・セキュリティ・モデルが使用されているので、プロパゲータとマテリアライズド・ビュー管理者はマスター・サイトのマスター・グループ A、B、C および D にアクセスできます。その結果、データベース管理者は任意のリモート・サイトで管理機能を実行し、変更内容をマスター・サイトに伝播できるため、データベース管理の柔軟性が大幅に向上します。ただし、悪意のあるユーザーがリモート・サイトからマスター・サイトのデータを見たり、破損させたりする可能性も高くなります。

使用するセキュリティ・モデルに関係なく、Oracle では、オブジェクトがレプリケーション環境に追加されるとき、またはレプリケーション環境から削除されるときに、オブジェクトに適切な権限が自動的に付与されます。

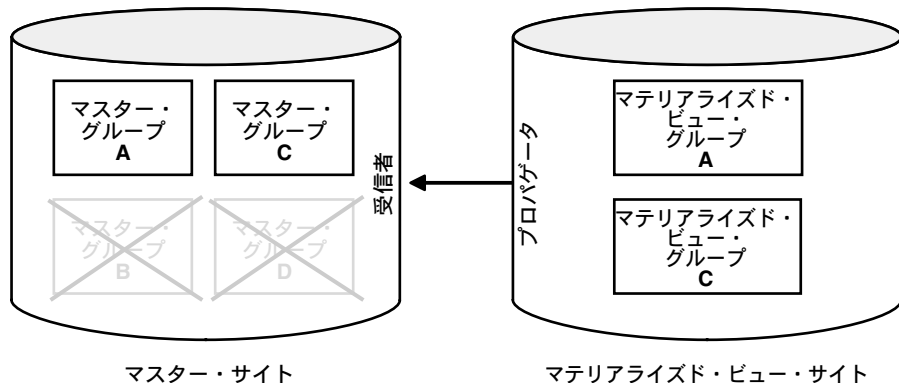


図 A-3 トラストッド・セキュリティ：マテリアライズド・ビュー・レプリケーション



アントラस्टッド・セキュリティでは、特定のレプリケーション・グループの使用に必要な権限のみがプロキシ・マテリアライズド・ビュー管理者と受信者に付与されます。したがって、プロパゲータとマテリアライズド・ビュー管理者は、マスター・サイトのこれらの特定のレプリケーション・グループのみにアクセスできます。図 A-4 に、マテリアライズド・ビュー・レプリケーションのアントラस्टッド・セキュリティ・モデルを示します。マテリアライズド・ビュー・サイトにはマテリアライズド・ビュー・グループ A と C が含まれるので、マスター・グループ A と C のみへのアクセスが必要です。アントラस्टッド・セキュリティを使用すると、マテリアライズド・ビュー・サイトのプロパゲータまたはマテリアライズド・ビュー管理者は、マスター・サイトのマスター・グループ B と D にアクセスできません。

図 A-4 アントラस्टッド・セキュリティ：マテリアライズド・ビュー・レプリケーション



一般にマテリアライズド・ビュー・サイトはセキュリティ侵害を受けやすいので、アントラステッド・セキュリティ・モデルが使用されます。マテリアライズド・ビュー・サイトでトラステッド・セキュリティ・モデルを使用する理由はほとんど考えられません。マテリアライズド・ビュー・サイトではアントラステッド・セキュリティ・モデルを使用することをお勧めします。

トラステッド・セキュリティ・モデルを使用する理由として1つ考えられるのは、マテリアライズド・ビュー・サイトがあらゆる観点（セキュリティ、常時ネットワーク接続、リソース）からマスター・サイトと判断される場合で、データをサブセット化するためにはマテリアライズド・ビューになっている場合です。マルチマスター構成では、行サブセット化と列サブセット化はサポートされていません。

表 A-3 のプロキシ・マテリアライズド・ビュー管理者と受信者の項目に示す適切な API コールを使用して、各ユーザーに適切な権限を割り当てます。

表 A-3 マスター・サイトまたはマスター・マテリアライズド・ビュー・サイトの必須ユーザー・アカウント

ユーザー	権限
プロキシ・マテリアライズド・ビュー・サイト管理者	<p>詳細は、21-4 ページの「<a href="#">REGISTER_USER_REPGROUP プロシージャ</a>」を参照してください。</p> <p>トラステッド:</p> <pre>DBMS_REPCAT_ADMIN.REGISTER_USER_REPGROUP privilege =&gt; 'proxy_snapadmin' list_of_gnames =&gt; NULL</pre> <p>アントラステッド:</p> <pre>DBMS_REPCAT_ADMIN.REGISTER_USER_REPGROUP privilege =&gt; 'proxy_snapadmin' list_of_gnames =&gt; 'mastergroupname'</pre>
受信者	<p>詳細は、21-4 ページの「<a href="#">REGISTER_USER_REPGROUP プロシージャ</a>」を参照してください。</p> <p>トラステッド:</p> <pre>DBMS_REPCAT_ADMIN.REGISTER_USER_REPGROUP privilege =&gt; 'receiver' list_of_gnames =&gt; NULL</pre> <p>アントラステッド:</p> <pre>DBMS_REPCAT_ADMIN.REGISTER_USER_REPGROUP privilege =&gt; 'receiver' list_of_gnames =&gt; 'mastergroupname'</pre>

**表 A-3 マスター・サイトまたはマスター・マテリアライズド・ビュー・サイトの必須ユーザー・アカウント**

ユーザー	権限
プロキシ・リフレッシュャ	<p>トラステッド:</p> <p>CREATE SESSION 権限付与 SELECT ANY TABLE 権限付与</p> <p>アントラステッド:</p> <p>CREATE SESSION 権限付与 必要なマスター表、またはマスター・マテリアライズド・ビュー・ログおよびマテリアライズド・ビュー・ログに SELECT 権限付与</p>

マテリアライズド・ビュー、および対応付けられたマスター・サイトまたはマスター・マテリアライズド・ビュー・サイトにアカウントを作成した後、次に示すプライベート・データベース・リンクを作成します。このリンクは、ユーザー名およびパスワードを含んだ、マテリアライズド・ビュー・サイトからマスター・サイトまたはマスター・マテリアライズド・ビュー・サイトへのリンクです。

- マテリアライズド・ビュー・レプリケーション管理者からプロキシ・マテリアライズド・ビュー・レプリケーション管理者へのリンク
- プロパゲータから受信者へのリンク
- リフレッシュャからプロキシ・リフレッシュャへのリンク
- マテリアライズド・ビュー所有者からマスター・サイトまたはマスター・マテリアライズド・ビュー・サイトへのリフレッシュ用リンク

1つのユーザー・アカウントがマテリアライズド・ビュー管理者、プロパゲータおよびリフレッシュャとして機能するように指定した場合は、これらの機能に対し、1つのマテリアライズド・ビュー・サイトごとにリンクを1つ作成する必要があります。マスター・サイトまたはマスター・マテリアライズド・ビュー・サイトからマテリアライズド・ビューへのリンクは必要ありません。

これらのリンクを作成した後、マテリアライズド・ビュー・サイトで DBMS\_DEFER\_SYS.SCHEDULE\_PUSH および DBMS\_DEFER\_SYS.SCHEDULE\_PURGE をコールして、遅延トランザクション・キューを対応付けられたマスター・サイトまたはマスター・マテリアライズド・ビュー・サイトに伝播する頻度、およびこのキューをバージする頻度を定義する必要があります。対応付けられたマスター・サイトまたはマスター・マテリアライズド・ビュー・サイトから変更を取り込む頻度をスケジュール化する場合は、マテリアライズド・ビュー・サイトで、DMBS\_REFRESH.REFRESH もコールする必要があります。



---

## ユーザー定義の競合解消メソッド

この付録では、ユーザー定義の競合解消メソッドおよびユーザー定義の競合通知方法の作成方法を説明します。この付録では、次の項目を説明します。

- ユーザー定義の競合解消メソッド
- ユーザー定義の競合通知方法
- 競合解消情報の表示

## ユーザー定義の競合解消メソッド

ユーザー独自の競合解消メソッドや通知方法を作成できます。ユーザー定義の競合解消メソッドとは、TRUE または FALSE のいずれかを返す PL/SQL ファンクションです。TRUE は、列グループへの競合する変更すべてが競合解消メソッドによって正常に解消されたことを示します。競合が正常に解消されない場合は、FALSE が返されます。競合解消メソッドから TRUE が返されるか、または使用可能な競合解消メソッドがなくなるまで、方法の評価が順番に続行されます。

競合解消メソッドで例外が呼び出されると、その方法の評価が停止します。それ以降の順序番号でその他の競合解消メソッドが用意されていても、評価は行われません。

## 競合解消メソッドのパラメータ

ユーザー定義の競合解消メソッドに必要なパラメータは、解消する競合のタイプ（一意、更新または削除）およびレプリケートされる表の列によって決定されます。すべての競合解消メソッドは、表の古い列値、新しい列値およびカレント列値の組合せを使用します。

- 古い値は、発信元サイトでの変更前の行の値を表します。
- 新しい値は、発信元サイトでの変更後の行の値を表します。
- カレント値は、受信サイトでの対応する行の値を表します。

---



---

**注意：** Oracle では、比較する行の判別に、主キー、すなわち SET\_COLUMNS で指定されたキーが使用されることに注意してください。

---



---

競合解消機能は、DBMS\_REPCAT.ADD\_conflicttype\_RESOLUTION プロシージャの PARAMETER\_COLUMN\_NAME 引数で指定した列値をパラメータとして受け取ります。列パラメータは、PARAMETER\_COLUMN\_NAME 引数に示される順序で、またこの引数に '\*' を指定した場合はアルファベット昇順で、競合解消メソッドに渡されます。古い列値と新しい列値の両方が（更新の競合の）パラメータとして渡される場合、列の古い値は新しい値の直前に置かれます。

---



---

**注意：**

- ユーザー定義の競合解消メソッドでは、パラメータ列のタイプのチェックは、対応付けられたレプリケート表のレプリケーション・サポートを再生成するまで実行されません。
  - 列オブジェクトの属性は、ユーザー定義の競合解消メソッドの列パラメータとして定義することはできません。
- 
-

## 更新の競合の解消

更新の競合の場合、ユーザー定義ファンクションでは、列グループ内の各列に次の値を使用する必要があります。

- 発信元サイトからの古い列値。このパラメータのモードは IN です。この値は変更できません。
- 発信元サイトからの新しい列値。このパラメータのモードは IN OUT です。そのファンクションで競合を正常に解消できる場合は、必要に応じて新しい列値を変更する必要があります。
- 受信サイトからのカレント列値。このパラメータのモードは IN です。

列値は、古い値、新しい値、カレント値と連続して受信されます。競合解消メソッドに渡す最後の引数は、ブール型フラグである必要があります。このフラグが FALSE の場合、IN OUT パラメータ（すなわち、新しい列）の値をすでに更新したこと、およびカレント列値をこの新しい値に更新する必要があることを示します。このフラグが TRUE の場合、カレント列値を変更できないことを示します。

## 一意性競合の解消

一意性競合は、INSERT または UPDATE の結果として発生する場合があります。ユーザー定義の一意性競合解消メソッドでは、列グループの各列に、発信元サイトからの IN OUT モードの新しい列値を受け入れる必要があります。競合解消メソッドに渡す最後のパラメータは、ブール型フラグである必要があります。

競合解消メソッドで競合が解消された場合、新しい列値を変更して、現在行への挿入または現在行の新しい列値による更新を可能にします。ファンクションにより新しい列値を廃棄する場合はブール型フラグを TRUE に設定し、廃棄しない場合は FALSE に設定します。

競合解消メソッドでは一意性競合におけるデータの収束を保証できないので、ユーザー定義の一意性競合解消メソッドに通知メカニズムを組み込む必要があります。

## 削除の競合の解消

削除の競合は、ローカル・サイトからの削除が正常に実行されても、対応付けられた行が（更新されたなどの理由で）リモート・サイトにない場合に発生します。削除の競合の場合、ファンクションでは、行全体の古い列値を IN OUT モードで受け入れる必要があります。競合解消メソッドに渡す最後のパラメータは、ブール型フラグである必要があります。

競合解消メソッドで競合が解消されると、古い列値が変更され、すべての古い列値に一致する現在行を削除できます。ファンクションによりこれらの列値を廃棄する場合はブール型フラグを TRUE に設定し、廃棄しない場合は FALSE に設定します。

ローカル・サイトで削除を実行し、リモート・サイトで更新を実行した場合、リモート・サイトで削除の競合が検出されますが、ローカル・サイトでは解消不可能な更新の競合が検出されます。このタイプの競合は自動的に処理できません。この競合によって

NO\_DATA\_FOUND 例外が発生し、このトランザクションはエラー・トランザクションとして記録されます。

これらのタイプの更新や削除の競合を適切に処理するメカニズムを設計することは困難です。削除された行を単純にマーク付けして、プロシージャ・レプリケーションを使用してページすることで、これらのタイプの競合を完全に回避するほうがはるかに容易です。

**関連項目：** 6-32 ページの「[削除の競合防止メソッドの作成](#)」

## 複数層マテリアライズド・ビューとユーザー定義の競合解消メソッド

ユーザー定義の競合解消メソッドを複数層マテリアライズド・ビューに使用するときは、これらの方法に関する情報はマスター・マテリアライズド・ビュー・サイトに自動的にプルダウンされます。この情報は、マスター・マテリアライズド・ビュー・サイトのデータ・ディクショナリに格納されます。ただし、ユーザー定義の競合解消メソッドそのものをマスター・サイトからプルダウンすることはできません。このため、マスター・マテリアライズド・ビュー・サイトでこれらの方法を再作成する必要があります。

**関連項目：**

- ユーザー定義の競合解消メソッドの情報を格納するデータ・ディクショナリ・ビューの詳細は、B-11 ページの「[競合解消情報の表示](#)」を参照してください。
- 競合解消と複数層マテリアライズド・ビューの詳細は、『Oracle9i アドバンスト・レプリケーション』を参照してください。

## ユーザー定義の競合解消メソッドの制限事項

次の項では、ユーザー定義の競合解消メソッドの制限事項を説明します。

### ユーザー定義の競合解消メソッドの SQL 文に関する制限事項

ユーザー定義の競合解消メソッドで次の SQL 文を使用しないでください。これらの文を使用すると、予期しない結果が起こる可能性があります。

- データ定義言語 (DDL) 文 (CREATE、ALTER、DROP など)
- トランザクション制御文 (COMMIT、ROLLBACK など)
- セッション制御 (ALTER SESSION など)
- システム制御 (ALTER SYSTEM など)



## ユーザー定義の競合解消メソッドの列サブセット化に関する制限事項

更新可能な複数層マテリアライズド・ビューを作成するときは、列グループの列をサブセット化しないでください。列をサブセット化すると、マスター表またはマスター・マテリアライズド・ビューの列が、これらのマスターをベースとするマテリアライズド・ビューから除外されます。これを行うには、マテリアライズド・ビューの作成中に `SELECT` 文でいくつかの `SELECT` 列を指定します。

競合解消を複数層マテリアライズド・ビューに使用するときは、マテリアライズド・ビュー・サイトでは競合解消メソッドを定義できません。競合解消メソッドは、常にマスター・サイトからプルダウンされます。このため、ユーザー定義の競合解消を持つ列グループの列をサブセット化すると、マスター・マテリアライズド・ビュー・サイトの列グループの列をすべてこの競合解消メソッドで見つけることができなくなります。この場合、競合解消メソッドから次のエラーが戻されます。

ORA-23460 解消方法内の列に値がありません。

たとえば、`hr.employees` 表の `job_id`、`salary` および `commission_pct` 列が、マスター・サイト `hq.world` にある、ユーザー定義の競合解消メソッドを持つ列グループ名 `employees_cg1` の一部である場合を考えます。営業スタッフのプライバシーを保護するため、列サブセット化を使用して `ca.us` オフィスの `salary` および `commission_pct` 列を排除する、レベル 1 の更新可能なマテリアライズド・ビューを作成します。`ca.us` オフィスにこのマテリアライズド・ビューを作成すると、競合解消メソッドが `hq.world` からプルダウンされます。次に、`sf.ca` オフィスに、`ca.us` オフィスのレベル 1 のマテリアライズド・ビューをベースとする更新可能なマテリアライズド・ビューを作成します。

このレプリケーション環境では、`ca.us` オフィスのレベル 1 のマテリアライズド・ビューで `job_id` 値の競合が発生した場合、競合解消メソッドにより `salary` および `commission_pct` 列が見つけられず、前述の ORA-23460 エラーが戻されます。

**関連項目：** 列サブセット化の詳細は、『Oracle9i アドバンスド・レプリケーション』を参照してください。

## ユーザー定義の競合解消メソッドの例

標準的な組込みの最大値および加算による競合解消メソッドを変形した、ユーザー定義方法の例を次に示します。標準方法と異なり、これらのユーザー定義ファンクションでは、競合解消に使用される列内の NULL 値を処理できます。

### 最大値ユーザー・ファンクション

```
-- User function similar to MAXIMUM method.
-- If curr is null or curr < new, then use new values.
-- If new is null or new < curr, then use current values.
-- If both are null, then no resolution.
-- Does not converge with > 2 masters, unless
-- always increasing.

FUNCTION max_null_loses(old          IN    NUMBER,
                        new          IN OUT NUMBER,
                        cur           IN    NUMBER,
                        ignore_discard_flag OUT  BOOLEAN)
RETURN BOOLEAN IS
BEGIN
    IF (new IS NULL AND cur IS NULL) OR new = cur THEN
        RETURN FALSE;
    END IF;
    IF new IS NULL THEN
        ignore_discard_flag := TRUE;
    ELSIF cur IS NULL THEN
        ignore_discard_flag := FALSE;
    ELSIF new < cur THEN
        ignore_discard_flag := TRUE;
    ELSE
        ignore_discard_flag := FALSE;
    END IF;
    RETURN TRUE;
END max_null_loses;
```

## 加算ユーザー・ファンクション

```
-- User function similar to ADDITIVE method.
-- If old is null, then old = 0.
-- If new is null, then new = 0.
-- If curr is null, then curr = 0.
-- new = curr + (new - old) -> just like ADDITIVE method.

FUNCTION additive_nulls(old                IN    NUMBER,
                        new                IN OUT NUMBER,
                        cur                IN    NUMBER,
                        ignore_discard_flag OUT    BOOLEAN)

RETURN BOOLEAN IS
  old_val NUMBER := 0.0;
  new_val NUMBER := 0.0;
  cur_val NUMBER := 0.0;
BEGIN
  IF old IS NOT NULL THEN
    old_val := old;
  END IF;
  IF new IS NOT NULL THEN
    new_val := new;
  END IF;
  IF cur IS NOT NULL THEN
    cur_val := cur;
  END IF;
  new := cur_val + (new_val - old_val);
  ignore_discard_flag := FALSE;
  RETURN TRUE;
END additive_nulls;
```

## ユーザー定義の競合通知方法

競合通知方法は、競合解消のかわりまたは追加として競合を通知する、ユーザー定義ファンクションです。たとえば、独自の競合通知方法を作成して、データベース表に競合情報を記録したり、電子メール・メッセージを送信したり、管理者を呼び出したりできます。競合通知方法を作成した後、それを列グループ（または制約）に特定の順序で割り当て、競合が発生した時点か、後続の競合解消メソッドを試行する前か、競合解消を試行して失敗した後に、通知を受けることができます。

レプリケート表にユーザー定義の競合通知メカニズムを構成するには、次の手順を実行する必要があります。

**手順 1 競合通知ログを作成します。**

**手順 2 パッケージ内にユーザー定義の競合通知方法を作成します。**

次の項で各手順を説明します。

### 競合通知ログの作成

ユーザー定義の競合通知方法を使用するレプリケート表を構成するときは、最初に、競合通知を記録できるデータベース表を作成します。マスター・グループ内の 1 つ以上の表での競合通知を記録するデータベース表を作成できます。

すべてのマスター・サイトで競合通知ログ表を作成するには、レプリケーション実行 DDL 機能を使用します。詳細は、20-79 ページの「[EXECUTE\\_DDL プロシージャ](#)」を参照してください。競合通知表のエントリは競合を検出するサイトに固有のエントリであるため、競合通知表に対するレプリケーション・サポートを生成しないでください。

### 競合通知ログ表の例

次の CREATE TABLE 文では、マスター・グループのいくつかの表から競合通知を記録するのに使用する表を作成します。

```
CREATE TABLE conf_report (
  line          NUMBER(2),      --- used to order message text
  txt           VARCHAR2(80),   --- conflict notification message
  timestamp     DATE,          --- time of conflict
  table_name    VARCHAR2(30),   --- table in which the
                                --- conflict occurred
  table_owner   VARCHAR2(30),   --- owner of the table
  conflict_type VARCHAR2(6)     --- INSERT, DELETE or UNIQUE
);
```

## 競合通知パッケージの作成

競合通知方法を作成するには、PL/SQL パッケージ内で方法を定義し、対応付けられたレプリケート表とともに PL/SQL パッケージをマスター・グループの一部としてレプリケートする必要があります。

競合通知方法では、競合の通知のみ、または競合の通知と解消の両方を実行できます。可能な場合は常に、Oracle の組込み競合解消メソッドのいずれかを使用して競合を解消する必要があります。ユーザー定義の競合通知方法で競合の通知のみを実行する場合は、競合を解消する競合解消メソッドとともに、ユーザー定義の通知方法を列グループ（または制約）に割り当ててください。

---

---

**注意：** レプリケーションの競合が最終的に解消されない場合は、通知表に加えられた更新内容を含め、トランザクション全体がロールバックされます。トランザクションとは無関係に通知が必要な場合は、Oracle DBMS\_PIPE パッケージを使用する通知メカニズムを設計できます。

---

---

## 競合通知パッケージの例

次のパッケージおよびパッケージ本体では、CUSTOMERS 表の一意性競合を定義済の CONF\_REPORT 表に記録することによって、簡単な競合通知を実行します。

---

---

**注意：** この競合通知の例では、競合は解消されません。したがって、競合解消メソッド（廃棄や上書きなど）を用意するか、エラーが解消されずにトランザクションがロールバックされても正しく機能する通知メカニズム（電子メールの利用など）を用意する必要があります。次のユーザー定義の競合通知方法に簡単な変更を加えれば、さらに有効な処置を取ることができます。たとえば、このパッケージでは、通知メッセージを記録でき、また DBMS\_OFFICE ユーティリティ・パッケージを使用して Oracle Office の電子メール・メッセージを管理者に送信することもできます。

---

---

```
CREATE OR REPLACE PACKAGE notify AS
  -- Report uniqueness constraint violations on CUSTOMERS table
  FUNCTION customers_unique_violation (
    first_name      IN OUT VARCHAR2,
    last_name       IN OUT VARCHAR2,
    discard_new_values IN OUT BOOLEAN)
    RETURN BOOLEAN;
END notify;
/

CREATE OR REPLACE PACKAGE BODY notify AS
  -- Define a PL/SQL index-by table to hold the notification message
  TYPE message_table IS TABLE OF VARCHAR2(80) INDEX BY BINARY_INTEGER;
  PROCEDURE report_conflict (
```

```

conflict_report IN MESSAGE_TABLE,
report_length   IN NUMBER,
conflict_time   IN DATE,
conflict_table  IN VARCHAR2,
table_owner     IN VARCHAR2,
conflict_type   IN VARCHAR2) IS
BEGIN
  FOR idx IN 1..report_length LOOP
    BEGIN
      INSERT INTO sales.conf_report
        (line, txt, timestamp, table_name, table_owner, conflict_type)
        VALUES (idx, SUBSTR(conflict_report(idx),1,80), conflict_time,
          conflict_table, table_owner, conflict_type);
      EXCEPTION WHEN others THEN NULL;
    END;
  END LOOP;
END report_conflict;
-- This is the conflict resolution method that is called first when
-- a uniqueness constraint violated is detected in the CUSTOMERS table.
FUNCTION customers_unique_violation (
  first_name IN OUT VARCHAR2,
  last_name IN OUT VARCHAR2,
  discard_new_values IN OUT BOOLEAN)
RETURN BOOLEAN IS
  local_node VARCHAR2(128);
  conf_report MESSAGE_TABLE;
  conf_time DATE := SYSDATE;
BEGIN
  -- Get the global name of the local site
  BEGIN
    SELECT global_name INTO local_node FROM global_name;
    EXCEPTION WHEN others THEN local_node := '?';
  END;
  -- Generate a message for the DBA
  conf_report(1) := 'UNIQUENESS CONFLICT DETECTED IN TABLE CUSTOMERS ON ' ||
    TO_CHAR(conf_time, 'MM-DD-YYYY HH24:MI:SS');
  conf_report(2) := ' AT NODE ' || local_node;
  conf_report(3) := 'ATTEMPTING TO RESOLVE CONFLICT USING ' ||
    'APPEND SEQUENCE METHOD';
  conf_report(4) := 'FIRST NAME: ' || first_name;
  conf_report(5) := 'LAST NAME: ' || last_name;
  conf_report(6) := NULL;
  --- Report the conflict
  report_conflict(conf_report, 5, conf_time, 'CUSTOMERS',
    'OFF_SHORE_ACCOUNTS', 'UNIQUE');
  --- Do not discard the new column values. They are still needed by
  --- other conflict resolution methods.

```

```
discard_new_values := FALSE;
--- Indicate that the conflict was not resolved.
RETURN FALSE;
END customers_unique_violation;
END notify;
/
```

## 競合解消情報の表示

Oracle には、レプリケーション・カタログ (REPCAT) ビューが用意されており、このビューを使用して、レプリケーション環境の各表および列グループで使用している競合解消メソッドを判別できます。各ビューには、`USER_*`、`ALL_*` および `SYS.DBA_*` の 3 つのバージョンがあります。使用可能なビューを次の表に示します。

ビュー	説明
<a href="#">ALL_REPRESOLUTION_METHOD</a>	使用可能な競合解消メソッドがすべて表示されます。
<a href="#">ALL_REPCOLUMN_GROUP</a>	データベースに定義された列グループがすべて表示されます。
<a href="#">ALL_REPGROUPED_COLUMN</a>	データベース内の各列グループの列がすべて表示されます。
<a href="#">ALL_REPPRIORITY_GROUP</a>	データベースに定義された優先グループおよびサイト優先グループがすべて表示されます。
<a href="#">ALL_REPPRIORITY</a>	各優先グループまたはサイト優先グループの値および対応する優先順位レベルが表示されます。
<a href="#">ALL_REPCONFLICT</a>	データベース内の表、列グループおよび一意制約に関して解決メソッドを指定した競合のタイプ（削除、更新または一意性）が表示されます。
<a href="#">ALL_REPRESOLUTION</a>	各オブジェクトで発生する競合解消メソッドに関する、より具体的な情報が表示されます。
<a href="#">ALL_REPPARAMETER_COLUMN</a>	競合解消メソッドが競合の解消に使用する列が表示されます。

**関連項目：** [第 25 章「レプリケーション・カタログ・ビュー」](#)





---

# 索引

## A

ALL\_REPCAT\_REFRESH\_TEMPLATES ビュー, 25-6  
ALL\_REPCAT\_TEMPLATE\_OBJECTS ビュー, 25-7  
ALL\_REPCAT\_TEMPLATE\_PARSMS ビュー, 25-9  
ALL\_REPCAT\_TEMPLATE\_SITES ビュー, 25-11  
ALL\_REPCAT\_USER\_AUTHORIZATIONS ビュー,  
25-12  
ALL\_REPCAT\_USER\_PARM\_VALUES ビュー, 25-13  
ALL\_REPCATLOG ビュー, 25-15  
ALL\_REPCOLUMN\_GROUP ビュー, 25-19  
ALL\_REPCOLUMN ビュー, 25-16  
ALL\_REPCONFLICT ビュー, 25-19  
ALL\_REPDDL ビュー, 25-20  
ALL\_REPGENOBJECTS ビュー, 25-21  
ALL\_REPGROUP\_PRIVILEGES ビュー, 25-23  
ALL\_REPGROUPED\_COLUMN ビュー, 25-24  
ALL\_REPGROUP ビュー, 25-22  
ALL\_REPKEY\_COLUMNS ビュー, 25-25  
ALL\_REPOBJECT ビュー, 25-26  
ALL\_REPPARAMETER\_COLUMN ビュー, 25-28  
ALL\_REPPRIORITY\_GROUP ビュー, 25-30  
ALL\_REPPRIORITY ビュー, 25-29  
ALL\_REPPROP ビュー, 25-31  
ALL\_REPRESOL\_STATS\_CONTROL ビュー, 25-32  
ALL\_REPRESOLUTION\_METHOD ビュー, 25-34  
ALL\_REPRESOLUTION\_STATISTICS ビュー, 25-34  
統計収集, 6-42  
ALL\_REPRESOLUTION ビュー, 25-33  
ALL\_REPSITES ビュー, 25-35  
ALTER MATERIALIZED VIEW LOG 文, 8-17  
AnyData データ型  
レプリケーション, 9-15

## D

DBA\_REGISTERED\_MVIEW\_GROUPS ビュー, 25-5  
DBA\_REPCAT\_REFRESH\_TEMPLATES ビュー, 25-36  
DBA\_REPCAT\_TEMPLATE\_OBJECTS ビュー, 25-37  
DBA\_REPCAT\_TEMPLATE\_PARSMS ビュー, 25-37  
DBA\_REPCAT\_TEMPLATE\_SITES ビュー, 25-37  
DBA\_REPCAT\_USER\_AUTHORIZATIONS ビュー,  
25-37  
DBA\_REPCAT\_USER\_PARM\_VALUES ビュー, 25-37  
DBA\_REPCATLOG ビュー, 25-38  
要求のページ, 20-87  
DBA\_REPCOLUMN\_GROUP ビュー, 25-38  
更新, 7-43, 20-38  
DBA\_REPCOLUMN ビュー, 25-38  
DBA\_REPCONFLICT ビュー, 25-38  
DBA\_REPDDL ビュー, 25-38  
DBA\_REPEXTENSIONS ビュー, 25-38  
DBA\_REPGENOBJECTS ビュー, 25-41  
DBA\_REPGROUP\_PRIVILEGES ビュー, 25-41  
DBA\_REPGROUPED\_COLUMN ビュー, 25-41  
DBA\_REPGROUP ビュー, 25-41  
更新, 7-43, 20-41  
DBA\_REPKEY\_COLUMNS ビュー, 25-42  
DBA\_REPOBJECT ビュー, 25-42  
更新, 7-43, 20-42  
DBA\_REPPARAMETER\_COLUMN ビュー, 25-42  
DBA\_REPPRIORITY\_GROUP ビュー, 25-42  
更新, 7-43, 20-40  
DBA\_REPPRIORITY ビュー, 25-42  
DBA\_REPPROP ビュー, 25-42  
DBA\_REPRESOL\_STATS\_CONTROL ビュー, 25-43  
DBA\_REPRESOLUTION\_METHOD ビュー, 25-43  
DBA\_REPRESOLUTION\_STATISTICS ビュー, 25-43  
ページ, 6-43, 20-88

DBA\_REPRESOLUTION ビュー, 25-43  
     更新, 7-43, 20-45  
 DBA\_REPSITES\_NEW ビュー, 25-44  
 DBA\_REPSITES ビュー, 25-44  
     更新, 7-43, 20-43  
 DBMS\_DEFER\_QUERY パッケージ, 13-1  
     GET\_AnyData\_ARG ファンクション, 9-15, 13-7  
     GET\_ARG\_FORM ファンクション, 13-2  
     GET\_ARG\_TYPE ファンクション, 13-4  
     GET\_BLOB\_ARG ファンクション, 13-7  
     GET\_CALL\_ARGS プロシージャ, 13-6  
     GET\_CHAR\_ARG ファンクション, 13-7  
     GET\_CLOB\_ARG ファンクション, 13-7  
     GET\_datatype\_ARG ファンクション, 13-7  
     GET\_DATE\_ARG ファンクション, 13-7  
     GET\_IDS\_ARG ファンクション, 13-7  
     GET\_IYM\_ARG ファンクション, 13-7  
     GET\_NCHAR\_ARG ファンクション, 13-7  
     GET\_NCLOB\_ARG ファンクション, 13-7  
     GET\_NUMBER\_ARG ファンクション, 13-7  
     GET\_NVARCHAR2\_ARG ファンクション, 13-7  
     GET\_OBJECT\_NULL\_VECTOR\_ARG ファンク  
         ション, 13-9  
     GET\_RAW\_ARG ファンクション, 13-7  
     GET\_ROWID\_ARG ファンクション, 13-7  
     GET\_TIMESTAMP\_ARG ファンクション, 13-7  
     GET\_TSLTZ\_ARG ファンクション, 13-7  
     GET\_TSTZ\_ARG ファンクション, 13-7  
     GET\_VARCHAR2\_ARG ファンクション, 13-7  
 DBMS\_DEFER\_SYS パッケージ  
     ADD\_DEFAULT\_DEST プロシージャ, 14-4  
     CLEAR\_PROP\_STATISTICS プロシージャ, 10-32,  
         14-5  
     DELETE\_DEF\_DESTINATION プロシージャ, 14-6  
     DELETE\_DEFAULT\_DEST プロシージャ, 14-6  
     DELETE\_ERROR プロシージャ, 14-7  
     DELETE\_TRAN プロシージャ, 14-7, 14-8, 14-10  
     DISABLED ファンクション, 14-8  
     EXCLUDE\_PUSH ファンクション, 14-9  
     EXECUTE\_ERROR\_AS\_USER プロシージャ, 9-19,  
         14-11  
     EXECUTE\_ERROR プロシージャ, 7-35, 9-18,  
         14-10  
     PURGE ファンクション, 9-14, 14-12  
     PUSH ファンクション, 9-13, 14-14  
     REGISTER\_PROPAGATOR プロシージャ, 2-5,  
         2-22, 2-29, 14-17  
     SCHEDULE\_EXECUTION プロシージャ, 14-20  
     SCHEDULE\_PURGE プロシージャ, 2-6, 2-25,  
         2-30, 14-18  
     SCHEDULE\_PUSH プロシージャ, 2-17, 2-25,  
         2-31, 14-20  
     SET\_DISABLED プロシージャ, 14-22  
     UNREGISTER\_PROPAGATOR プロシージャ,  
         14-24  
     UNSCHEDULE\_PURGE プロシージャ, 14-25  
     UNSCHEDULE\_PUSH プロシージャ, 14-25  
 DBMS\_DEFER パッケージ, 12-1  
     ANY\_CHAR\_ARG プロシージャ, 12-5  
     ANY\_CLOB\_ARG プロシージャ, 12-5  
     ANY\_VARCHAR2\_ARG プロシージャ, 12-5  
     ANYDATA\_ARG プロシージャ, 12-5  
     BLOB\_ARG プロシージャ, 12-5  
     CALL プロシージャ, 12-2  
     CHAR\_ARG プロシージャ, 12-5  
     CLOB\_ARG プロシージャ, 12-5  
     COMMIT\_WORK プロシージャ, 12-4  
     datatype\_ARG プロシージャ, 12-5  
     DATE\_ARG プロシージャ, 12-5  
     IDS\_ARG プロシージャ, 12-5  
     IYM\_ARG プロシージャ, 12-5  
     NCHAR\_ARG プロシージャ, 12-5  
     NCLOB\_ARG プロシージャ, 12-5  
     NUMBER\_ARG プロシージャ, 12-5  
     NVARCHAR2\_ARG プロシージャ, 12-5  
     RAW\_ARG プロシージャ, 12-5  
     ROWID\_ARG プロシージャ, 12-5  
     TIMESTAMP\_ARG プロシージャ, 12-5  
     TRANSACTION プロシージャ, 12-7  
     TSLTZ\_ARG プロシージャ, 12-5  
     TSTZ\_ARG プロシージャ, 12-5  
     VARCHAR2\_ARG プロシージャ, 12-5  
 DBMS\_MVIEW パッケージ  
     BEGIN\_TABLE\_REORGANIZATION プロシー  
         ジャ, 8-20, 15-3  
     END\_TABLE\_REORGANIZATION プロシージャ,  
         8-20, 15-4  
     EXPLAIN\_MVIEW プロシージャ, 15-4  
     EXPLAIN\_REWRITE プロシージャ, 15-6  
     I\_AM\_A\_REFRESH ファンクション, 15-7  
     PMARKER ファンクション, 15-7  
     PURGE\_DIRECT\_LOAD\_LOG プロシージャ, 15-8  
     PURGE\_LOG プロシージャ, 8-18, 15-8

PURGE\_MVIEW\_FROM\_LOG プロシージャ, 8-11,  
 8-14, 8-16, 8-18, 15-10  
 REFRESH\_ALL\_MVIEWS プロシージャ, 15-14  
 REFRESH\_DEPENDENT プロシージャ, 15-16  
 REFRESH プロシージャ, 8-2, 8-34, 15-12  
 REGISTER\_MVIEW プロシージャ, 15-18  
 UNREGISTER\_MVIEW プロシージャ, 8-14, 15-20  
 DBMS\_OFFLINE\_OG パッケージ  
 BEGIN\_INSTANTIATION プロシージャ, 7-36,  
 16-2  
 BEGIN\_LOAD プロシージャ, 7-38, 16-4  
 END\_INSTANTIATION プロシージャ, 7-39, 16-5  
 END\_LOAD プロシージャ, 7-39, 16-7  
 RESUME\_SUBSET\_OF\_MASTERS プロシージャ,  
 7-37, 16-8  
 DBMS\_OFFLINE\_SNAPSHOT パッケージ  
 BEGIN\_LOAD プロシージャ, 8-30, 17-2  
 END\_LOAD プロシージャ, 8-32, 17-4  
 DBMS\_RECTIFIER\_DIFF パッケージ, 9-9  
 DIFFERENCES プロシージャ, 9-9, 18-2  
 RECTIFY プロシージャ, 9-9, 18-5  
 DBMS\_REFRESH パッケージ  
 ADD プロシージャ, 5-11, 5-18, 19-2  
 CHANGE プロシージャ, 19-3  
 DESTROY プロシージャ, 19-6  
 MAKE プロシージャ, 5-7, 5-15, 19-6  
 REFRESH プロシージャ, 8-2, 19-9  
 SUBTRACT プロシージャ, 19-10  
 DBMS\_REPCAT\_ADMIN パッケージ  
 GRANT\_ADMIN\_ANY\_SCHEMA プロシージャ,  
 2-5, 2-22, 2-28, 21-3  
 GRANT\_ADMIN\_SCHEMA プロシージャ, 21-4  
 REGISTER\_USER\_REPGROUP プロシージャ, 2-6,  
 2-7, 2-11, 2-15, 2-23, 2-26, 21-4  
 REVOKE\_ADMIN\_ANY\_SCHEMA プロシージャ,  
 21-6  
 REVOKE\_ADMIN\_SCHEMA プロシージャ, 21-7  
 UNREGISTER\_USER\_REPGROUP プロシージャ,  
 21-8  
 DBMS\_REPCAT\_INSTANTIATE パッケージ  
 DROP\_SITE\_INSTANTIATION プロシージャ, 8-4,  
 8-6, 22-2  
 INSTANTIATE\_OFFLINE ファンクション, 22-3  
 INSTANTIATE\_ONLINE ファンクション, 22-5  
 DBMS\_REPCAT\_RGT パッケージ  
 ALTER\_REFRESH\_TEMPLATE プロシージャ, 23-4  
 ALTER\_TEMPLATE\_OBJECT プロシージャ, 23-6  
 ALTER\_TEMPLATE\_PARM プロシージャ, 23-8  
 ALTER\_USER\_AUTHORIZATION プロシージャ,  
 23-10  
 ALTER\_USER\_PARM\_VALUE プロシージャ,  
 23-12  
 COMPARE\_TEMPLATES ファンクション, 23-14  
 COPY\_TEMPLATE ファンクション, 23-15  
 CREATE\_OBJECT\_FROM\_EXISTING ファンク  
 ション, 23-17  
 CREATE\_REFRESH\_TEMPLATE ファンクション,  
 23-18  
 CREATE\_REFRESH\_TEMPLATE プロシージャ,  
 4-6  
 CREATE\_TEMPLATE\_OBJECT ファンクション,  
 23-20  
 CREATE\_TEMPLATE\_OBJECT プロシージャ, 4-7  
 CREATE\_TEMPLATE\_PARM ファンクション,  
 23-23  
 CREATE\_USER\_AUTHORIZATION ファンク  
 ション, 23-25  
 CREATE\_USER\_AUTHORIZATION プロシージャ,  
 4-12  
 CREATE\_USER\_PARM\_VALUE ファンクション,  
 23-27  
 DELETE\_RUNTIME\_PARMS プロシージャ, 23-29  
 DROP\_ALL\_OBJECTS プロシージャ, 23-30  
 DROP\_ALL\_TEMPLATE\_PARMS プロシージャ,  
 23-31  
 DROP\_ALL\_TEMPLATE\_SITES プロシージャ,  
 23-32  
 DROP\_ALL\_TEMPLATES プロシージャ, 23-33  
 DROP\_ALL\_USER\_AUTHORIZATIONS プロシー  
 ジャ, 23-33  
 DROP\_ALL\_USER\_PARM\_VALUES プロシージャ,  
 23-34  
 DROP\_REFRESH\_TEMPLATE プロシージャ, 23-35  
 DROP\_SITE\_INSTANTIATION プロシージャ,  
 23-36  
 DROP\_TEMPLATE\_OBJECT プロシージャ, 23-37  
 DROP\_TEMPLATE\_PARM プロシージャ, 23-38  
 DROP\_USER\_AUTHORIZATION プロシージャ,  
 23-39  
 DROP\_USER\_PARM\_VALUE プロシージャ, 23-40  
 GET\_RUNTIME\_PARM\_ID ファンクション, 23-41  
 INSERT\_RUNTIME\_PARMS プロシージャ, 23-41  
 INSTANTIATE\_OFFLINE ファンクション, 23-43  
 INSTANTIATE\_OFFLINE プロシージャ, 4-15

INSTANTIATE\_ONLINE ファンクション, 23-45  
INSTANTIATE\_ONLINE プロシージャ, 4-16  
LOCK\_TEMPLATE\_EXCLUSIVE プロシージャ, 23-48  
LOCK\_TEMPLATE\_SHARED プロシージャ, 23-48  
DBMS\_REPCAT パッケージ  
ADD\_DELETE\_RESOLUTION プロシージャ, 20-18  
ADD\_GROUPED\_COLUMN プロシージャ, 20-7  
ADD\_MASTER\_DATABASE プロシージャ, 3-10, 7-30, 7-32, 20-8  
ADD\_NEW\_MASTERS プロシージャ, 7-12, 7-24, 20-10  
ADD\_PRIORITY\_CHAR プロシージャ, 20-16  
ADD\_PRIORITY\_datatype プロシージャ, 20-16  
ADD\_PRIORITY\_DATE プロシージャ, 20-16  
ADD\_PRIORITY\_NUMBER プロシージャ, 20-16  
ADD\_PRIORITY\_VARCHAR2 プロシージャ, 20-16  
ADD\_SITE\_PRIORITY\_SITE プロシージャ, 6-24, 20-17  
ADD\_UNIQUENESS\_RESOLUTION プロシージャ, 20-18  
ADD\_UPDATE\_RESOLUTION プロシージャ, 6-5, 6-8, 6-12, 6-15, 6-20, 6-25, 20-18  
ALTER\_CATCHUP\_PARAMETERS プロシージャ, 20-23  
ALTER\_MASTER\_PROPAGATION プロシージャ, 20-25  
ALTER\_MASTER\_REPOBJECT プロシージャ, 6-10, 6-22, 6-34, 9-2, 20-27  
ALTER\_MVIEW\_PROPAGATION プロシージャ, 20-30  
ALTER\_PRIORITY\_CHAR プロシージャ, 20-33  
ALTER\_PRIORITY\_datatype プロシージャ, 20-33  
ALTER\_PRIORITY\_DATE プロシージャ, 20-33  
ALTER\_PRIORITY\_NUMBER プロシージャ, 20-33  
ALTER\_PRIORITY\_RAW プロシージャ, 20-33  
ALTER\_PRIORITY プロシージャ, 20-31  
ALTER\_SITE\_PRIORITY\_SITE プロシージャ, 20-36  
ALTER\_SITE\_PRIORITY プロシージャ, 20-34  
CANCEL\_STATISTICS プロシージャ, 6-42, 20-37  
COMMENT\_ON\_COLUMN\_GROUP プロシージャ, 7-43, 20-38  
COMMENT\_ON\_DELETE\_RESOLUTION プロシージャ, 7-43, 20-45  
COMMENT\_ON\_MVIEW\_REPSITES プロシージャ, 20-39

COMMENT\_ON\_PRIORITY\_GROUP プロシージャ, 7-43, 20-40  
COMMENT\_ON\_REPGROUP プロシージャ, 7-43, 20-41  
COMMENT\_ON\_REPOBJECT プロシージャ, 7-43, 20-42  
COMMENT\_ON\_REPSITES プロシージャ, 7-43, 20-43  
COMMENT\_ON\_SITE\_PRIORITY プロシージャ, 20-40  
COMMENT\_ON\_UNIQUE\_RESOLUTION プロシージャ, 7-43, 20-45  
COMMENT\_ON\_UPDATE\_RESOLUTION プロシージャ, 7-43, 20-45  
COMPARE\_OLD\_VALUES プロシージャ, 20-47  
CREATE\_MASTER\_REPGROUP プロシージャ, 3-5, 20-49  
CREATE\_MASTER\_REPOBJECT プロシージャ, 6-11, 6-23, 20-50  
CREATE\_MVIEW\_REPGROUP プロシージャ, 5-7, 5-14, 8-30, 8-36, 8-37, 20-54  
CREATE\_MVIEW\_REPOBJECT プロシージャ, 5-9, 5-10, 5-16, 5-17, 5-18, 8-36, 8-39, 20-55  
DEFINE\_COLUMN\_GROUP プロシージャ, 20-59  
DEFINE\_PRIORITY\_GROUP プロシージャ, 20-60  
DEFINE\_SITE\_PRIORITY プロシージャ, 6-24, 20-61  
DO\_DEFERRED\_REPCAT\_ADMIN プロシージャ, 6-36, 7-36, 20-62  
DROP\_COLUMN\_GROUP プロシージャ, 20-63  
DROP\_DELETE\_RESOLUTION プロシージャ, 20-77  
DROP\_GROUPED\_COLUMN プロシージャ, 20-64  
DROP\_MASTER\_REPGROUP プロシージャ, 20-66  
DROP\_MASTER\_REPOBJECT プロシージャ, 20-67  
DROP\_MVIEW\_REPGROUP プロシージャ, 8-4, 8-6, 8-9, 20-69  
DROP\_MVIEW\_REPOBJECT プロシージャ, 8-10, 20-70  
DROP\_PRIORITY\_CHAR プロシージャ, 20-73  
DROP\_PRIORITY\_datatype プロシージャ, 20-73  
DROP\_PRIORITY\_DATE プロシージャ, 20-73  
DROP\_PRIORITY\_GROUP プロシージャ, 20-72  
DROP\_PRIORITY\_NUMBER プロシージャ, 20-73  
DROP\_PRIORITY\_VARCHAR2 プロシージャ, 20-73  
DROP\_PRIORITY プロシージャ, 20-71

DROP\_SITE\_PRIORITY\_SITE プロシージャ, 20-76  
DROP\_SITE\_PRIORITY プロシージャ, 20-75  
DROP\_UNIQUE\_RESOLUTION プロシージャ,  
20-77  
DROP\_UPDATE\_RESOLUTION プロシージャ,  
20-77  
EXECUTE\_DDL プロシージャ, 20-79  
GENERATE\_MVIEW\_SUPPORT プロシージャ,  
20-80  
GENERATE\_REPLICATION\_SUPPORT プロシ  
ージャ, 3-12, 3-13, 9-4, 20-82  
MAKE\_COLUMN\_GROUP プロシージャ, 6-4,  
6-7, 6-12, 6-15, 6-18, 6-23, 20-84  
PREPARE\_INSTANTIATED\_MASTERS プロシ  
ージャ, 20-85  
PREPARE\_INSTANTIATED\_MASTER プロシ  
ージャ, 7-18, 7-28  
PURGE\_MASTER\_LOG プロシージャ, 20-87  
PURGE\_STATISTICS プロシージャ, 6-43, 20-88  
REFRESH\_MVIEW\_REPGROUP プロシージャ,  
20-89  
REGISTER\_MVIEW\_REPGROUP プロシージャ,  
20-91  
REGISTER\_STATISTICS プロシージャ, 6-42, 20-92  
RELOCATE\_MASTERDEF プロシージャ, 7-2,  
20-93  
REMOVE\_MASTER\_DATABASES プロシージャ,  
20-94  
REMOVE\_MASTER\_DATABASE プロシージャ,  
7-41  
RENAME\_SHADOW\_COLUMN\_GROUP プロシ  
ージャ, 20-96  
REPCAT\_IMPORT\_CHECK プロシージャ, 20-97  
RESUME\_MASTER\_ACTIVITY プロシージャ,  
3-14, 20-98  
RESUME\_PROPAGATION\_TO\_MDEF プロシ  
ージャ, 7-15, 7-27, 20-99  
SEND\_OLD\_VALUES プロシージャ, 20-100  
SET\_COLUMNS プロシージャ, 20-49, 20-102  
SPECIFY\_NEW\_MASTERS プロシージャ, 7-11,  
7-24, 20-104  
SUSPEND\_MASTER\_ACTIVITY プロシージャ,  
20-106  
SWITCH\_MVIEW\_MASTER プロシージャ, 8-2,  
20-107  
UNDO\_ADD\_NEW\_MASTERS\_REQUEST プロ  
シージャ, 20-109

UNREGISTER\_MVIEW\_REPGROUP プロシージャ,  
8-11, 20-111  
VALIDATE プロシージャ, 20-111  
WAIT\_MASTER\_LOG プロシージャ, 20-114  
コメント・プロシージャ, 7-43  
DBMS\_REPUTIL パッケージ  
FROM\_REMOTE ファンクション, 24-4  
GLOBAL\_NAME ファンクション, 24-4  
MAKE\_INTERNAL\_PKG プロシージャ, 24-5  
REPLICATION\_IS\_ON ファンクション, 24-3  
REPLICATION\_OFF プロシージャ, 7-48, 9-6,  
24-3  
REPLICATION\_ON プロシージャ, 7-48, 9-6, 24-3  
SYNC\_UP\_REP プロシージャ, 24-6  
DDL, 「データ定義言語」を参照  
DEFCALLDEST ビュー, 27-2  
DEFCALL ビュー, 27-2  
DEFDEFAULTDEST ビュー, 27-3  
接続先の削除, 14-6  
接続先の追加, 14-4  
DEFERRCOUNT ビュー, 27-3  
DEFERROR ビュー, 9-18, 27-3  
トランザクションの削除, 14-7  
DEFLOB ビュー, 27-4  
DEFPROPAGATOR ビュー, 27-4  
DEFSCHEDULE ビュー, 27-5  
統計の消去, 10-32, 14-5  
DEFTRANDEST ビュー, 27-8  
DEFTRAN ビュー, 27-7  
DROP MATERIALIZED VIEW LOG 文, 8-24

---

## L

LONG 列  
レプリケーション, 9-7

---

## O

Oracle レプリケーション・マネージメント・ツール  
レプリケーションの監視, 10-1

---

## P

PRESERVE MATERIALIZED VIEW LOG オプション  
TRUNCATE TABLE 文, 8-20

## S

---

SYS.ANYDATA, 13-7

## T

---

TRUNCATE TABLE 文  
PRESERVE MATERIALIZED VIEW LOG オプ  
ション, 8-20  
TRUNCATE 文, 8-19

## U

---

USER\_REPCAT\_REFRESH\_TEMPLATES ビュー,  
25-45  
USER\_REPCAT\_TEMP\_OUTPUT ビュー, 4-14  
USER\_REPCAT\_TEMPLATE\_OBJECTS ビュー, 25-45  
USER\_REPCAT\_TEMPLATE\_PARMs ビュー, 25-46  
USER\_REPCAT\_TEMPLATE\_SITES ビュー, 25-46  
USER\_REPCAT\_USER\_AUTHORIZATIONS ビュー,  
25-46  
USER\_REPCAT\_USER\_PARM\_VALUES ビュー,  
25-46  
USER\_REPCATALOG ビュー, 25-47  
USER\_REPCOLUMN\_GROUP ビュー, 25-47  
USER\_REPCOLUMN ビュー, 25-47  
USER\_REPCONFLICT ビュー, 25-47  
USER\_REPDDL ビュー, 25-48  
USER\_REPGENOBJECTS ビュー, 25-48  
USER\_REPGROUP\_PRIVILEGES ビュー, 25-48  
USER\_REPGROUPED\_COLUMN ビュー, 25-48  
USER\_REPGROUP ビュー, 25-48  
USER\_REPKEY\_COLUMNS ビュー, 25-48  
USER\_REPOBJECT ビュー, 25-49  
USER\_REPPARAMETER\_COLUMN ビュー, 25-49  
USER\_REPPRIORITY\_GROUP ビュー, 25-49  
USER\_REPPRIORITY ビュー, 25-49  
USER\_REPPROP ビュー, 25-50  
USER\_REPRESOL\_STATS\_CONTROL ビュー, 25-50  
USER\_REPRESOLUTION\_METHOD ビュー, 25-50  
USER\_REPRESOLUTION\_STATISTICS ビュー, 25-51  
USER\_REPRESOLUTION ビュー, 25-50  
USER\_REPSITES ビュー, 25-51

## V

---

V\$MVREFRESH ビュー, 26-2  
V\$REPLPROP ビュー, 10-32, 26-2  
V\$REPLQUEUE ビュー, 26-4

## い

---

インスタンス化, 4-18  
DROP\_SITE\_INSTANTIATION プロシージャ,  
22-2, 23-36  
オフライン, 4-13  
INSTANTIATE\_OFFLINE ファンクション,  
22-3, 23-43  
オンライン  
INSTANTIATE\_ONLINE ファンクション,  
22-5, 23-45  
スクリプト, 4-16  
リフレッシュ, 4-21  
インポート  
ステータス・チェック, 20-97  
マテリアライズド・ビュー  
オフライン・インスタンス化, 17-2,  
17-4  
レプリケーション・グループ  
オフライン・インスタンス化, 16-4,  
16-7

## え

---

エラー  
エラー・キュー  
DEFERROR ビュー, 9-18  
管理, 9-18  
エラー・トランザクション  
監視, 10-26  
受信者としての再実行, 9-18  
代替ユーザーとしての再実行, 9-19  
複数の再実行, 9-18

## お

---

### オブジェクト

#### 削除

- マテリアライズド・ビュー・サイト, 20-70
- マテリアライズド・ビュー・サイトからの, 8-9

#### 作成

- マスター・グループ, 20-49, 20-50
- マテリアライズド・ビュー・サイト用, 20-55

#### 変更, 20-27

- マテリアライズド・ビュー・サイトへの追加, 20-55
- レプリケーション・サポートの生成, 20-82
- レプリケーションの変更, 9-2

### オフライン・インスタンス化セッション

- INSTANTIATE\_OFFLINE ファンクション, 22-3, 23-43

- マスター・サイトの追加, 7-33
- マテリアライズド・ビュー, 17-2, 17-4
- マテリアライズド・ビュー・サイトの追加, 8-25
- レプリケーション・グループ, 16-2, 16-4, 16-5, 16-7, 16-8

### オンライン・インスタンス化セッション

- INSTANTIATE\_ONLINE ファンクション, 22-5, 23-45

## か

---

### 外部キー制約

- マスター・サイトの追加, 7-4

### 拡張可用性, 6-3, 7-4, 9-2, 20-10, 20-29, 20-85, 20-99, 20-104, 20-109

#### 可用性

- 拡張, 6-3, 7-4, 9-2, 20-10, 20-29, 20-85, 20-99, 20-104, 20-109

#### 管理者

- マテリアライズド・ビュー・サイト用
- 作成, 2-21

#### 管理要求

- 監視, 10-17
- エラー, 10-18
- ジョブ, 10-19

## き

---

### 記憶域パラメータ

- マテリアライズド・ビュー・ログ
- 変更, 8-17

### 競合

#### 回避

- 削除, 6-32
- 動的所有権, 6-37

#### 通知パッケージ

- 作成, B-9
- 例, B-9

#### 通知方法

- ユーザー定義, B-8

#### 通知ログ表

- 作成, B-8
- 例, B-8
- トークン渡し, 6-38
- ワークフロー, 6-37

### 競合解消, 6-1

- DBA\_REPRESOLUTION\_STATISTICS ビュー, 6-43
- 一意性競合, 6-26
- 上書きによるメソッド, 6-3
- 加算によるメソッド, 6-14, 20-18
- 監査, 6-42
- 最小のメソッド, 6-6
- 最大のメソッド, 6-6
- サイトの優先順位によるメソッド, 6-21
- サンプル・トリガー, 6-23

#### 準備, 6-2

#### 情報

- 表示, B-11
- 情報の表示, B-11
- 静止なしの設定, 6-3
- タイムスタンプによるメソッド, 6-9
- サンプル・トリガー, 6-11

#### 統計, 20-37, 20-92

- 収集, 6-42
- 取消し, 6-42
- 表示, 6-42

#### 廃棄によるメソッド, 6-3

#### プロシージャ・レプリケーション, 7-48

#### 平均によるメソッド, 6-14

#### ユーザー定義方法, B-2

- 一意性競合, B-3
- 更新の競合, B-3
- 削除の競合, B-3

制限事項, B-4  
パラメータ, B-2  
複数層マテリアライズド・ビュー, B-4  
例, B-6  
列オブジェクト, B-2  
列サブセット化, B-5  
優先グループによるメソッド, 6-17  
列グループ, 6-4, 6-7, 6-12, 6-15, 6-18, 6-23  
許可  
テンプレート・ユーザー, 4-12

## け

---

計画  
レプリケーション, 1-4

## こ

---

コメント  
更新, 7-43  
コメント・フィールド  
ビュー内の更新, 7-43

## さ

---

サイトの優先順位  
変更, 20-34  
サイト優先グループ  
削除, 20-75  
作成  
構文, 20-61  
メンバーの削除, 20-76  
メンバーの追加, 20-17

## し

---

受信者  
登録, 2-6  
使用禁止  
伝播, 14-22  
ジョブ  
キュー  
ジョブの削除, 14-25  
シリアル化  
トランザクション, 7-48

## す

---

スケジュール・リンク  
作成, 2-17  
ステータス  
伝播, 14-8  
スナップショット, 「マテリアライズド・ビュー」を  
参照

## せ

---

静止  
競合解消メソッドの設定, 6-3  
マスター・グループ, 20-106  
マスター・サイトの追加, 7-4, 7-30  
レプリケート・オブジェクトの変更, 9-2  
生成  
レプリケーション・サポート, 3-12  
プロシージャ・レプリケーション, 7-48  
セキュリティ  
トラステッド・セキュリティとアントラステッド・  
セキュリティの比較, A-3, A-8  
マテリアライズド・ビュー・レプリケーション,  
A-7  
トラステッド・セキュリティとアントラステッ  
ド・セキュリティの比較, A-8  
マルチマスター・レプリケーション, A-2  
トラステッド・セキュリティとアントラステッ  
ド・セキュリティの比較, A-3  
レプリケーション, A-1

## ち

---

遅延トランザクション  
DEFDEFAULTDEST ビュー  
接続先の削除, 14-6  
接続先の追加, 14-4  
DEFSCHEDULE ビュー  
接続先の削除, 14-6  
統計の消去, 14-5  
開始, 12-7  
監視, 10-20  
バージ・ジョブ, 10-23, 10-24  
プッシュ・ジョブ, 10-21, 10-22  
キューからの削除, 14-7  
再実行, 14-10  
実行のスケジュール, 14-20



- 遅延リモート・プロシージャ・コール (RPC)
  - 作成, 12-2
  - 即時実行, 14-14
  - 引数, 12-5
  - 引数値, 13-7
  - 引数のタイプ, 13-4
- データ・ディクショナリ・ビュー, 27-1
- 遅延トランザクション・キュー
  - 管理, 9-13
  - 遅延コール
    - 値の判定, 9-15
  - 伝播されたトランザクションのページ, 9-14
  - プッシュ, 9-13
- 違い
  - 表, 18-2
  - 調整, 18-5
- 調整
  - 表, 9-9, 18-5

## つ

---

- 通知パッケージ
  - 競合
    - 作成, B-9
- 通知方法
  - ユーザー定義, B-8
- 通知ログ表
  - 競合
    - 作成, B-8
    - 例, B-8

## て

---

- データ定義言語
  - 非同期, 20-79
  - レプリケート・オブジェクトの変更, 20-27
- データ・ディクショナリ・ビュー
  - コメント
    - 更新, 7-43
  - マテリアライズド・ビュー, 28-1
  - リフレッシュ・グループ, 28-1
  - レプリケーション, 10-1, 25-1
- データベース・リンク
  - 作成, 2-16, 2-29, 4-20, 5-5, 5-6, 5-13, 5-14
- 伝播
  - 使用禁止, 14-22
  - ステータス, 14-8

- パラレル
  - 監視, 10-31, 10-32
  - 変更, 20-25
  - 方法の変更, 20-25, 20-30
- テンプレート, 「配置テンプレート」を参照

## と

---

- 統計
  - 競合解消
    - 監査, 6-42
    - 収集, 6-42, 20-92
    - 消去, 6-43, 20-88
    - 取消し, 6-42
    - 表示, 6-42
  - 伝播に関する
    - 消去, 10-32, 14-5
- 動的所有権
  - 競合防止, 6-37
  - 行の所有者の検索, 6-40
  - 所有権の取得, 6-40
  - ワークフロー・パーティション分割, 6-37
- 動的パフォーマンス・ビュー
  - レプリケーション, 26-1
- トークン渡し, 6-38
  - 実現例, 6-37
- トラステッド・セキュリティ, A-3, A-8
- トランザクション
  - シリアル化, 7-48
- トリガー
  - サイトの優先順位による競合解消, 6-23
  - タイムスタンプによる競合解消, 6-11
  - レプリケート, 9-7

## は

---

- ページ
  - DBA\_REPCATLOG 表, 20-87
  - 遅延トランザクション・キュー, 9-14
    - 監視, 10-23
    - マスター・サイト, 2-6
    - マテリアライズド・ビュー・サイト, 2-25, 2-30
- 配置テンプレート
  - インスタンス化, 4-18
  - インスタンス化スクリプト, 4-16

- オブジェクト
  - 削除, 23-37
  - 作成, 23-20
  - すべて削除, 23-30
- オブジェクトの追加, 4-6
- オブジェクトの変更, 23-6
- オフライン・インスタンス化セッション, 4-13, 22-3, 23-43
- オンライン・インスタンス化セッション, 22-5, 23-45
- 概要, 4-2
- 監視, 10-10
- 既存のオブジェクトからのオブジェクトの作成, 23-17
- サイト
  - 削除, 23-36
  - すべて削除, 23-32
- サイトのインスタンス化の削除, 22-2
- 削除, 23-35
- 作成, 4-3, 4-5
- 作成用フローチャート, 4-4
- すべて削除, 23-33
- データ・ディクショナリ・ビュー, 25-6
- テンプレートのコピー, 23-15
- テンプレートの作成, 23-18
- テンプレートの比較, 23-14
- テンプレートの変更, 23-4
- テンプレートのロック, 23-48
- パッケージ化, 4-13, 4-14
  - オフライン・インスタンス化セッション, 4-14
  - オンライン・インスタンス化セッション, 4-15
- パラメータ
  - 削除, 23-38
  - 作成, 4-10, 23-23
  - すべて削除, 23-31
  - ユーザー値, 4-11
- パラメータの変更, 23-8
- ファイルの配布, 4-18
- マテリアライズド・ビュー・グループの削除, 8-3
- ユーザー定義型, 4-2
- ユーザー認証
  - 削除, 23-39
  - 作成, 23-25
  - すべて削除, 23-33
- ユーザー認証の変更, 23-10
- ユーザーの許可, 4-12

- ユーザー・パラメータ値
  - 削除, 23-40
  - 作成, 23-27
  - すべて削除, 23-34
- ユーザー・パラメータ値の変更, 23-12
- ランタイム・パラメータ
  - IDの取得, 23-41
  - 削除, 23-29
  - 作成, 23-41
  - 挿入, 23-41
- パッケージ化
  - 配置テンプレート, 4-13
- パフォーマンス
  - レプリケーション
    - 監視, 10-29
- パラメータ
  - 配置テンプレート, 4-10
  - ユーザー値, 4-11
- パラレル伝播
  - 監視, 10-31, 10-32

## ひ

---

- 比較
  - 表, 18-2
- 日付式, 2-6
- 表
  - オンラインでの再定義
    - レプリケーション, 8-20
  - コメントの更新, 7-43
  - 違い, 9-9
  - 調整, 9-9, 18-5
  - 比較, 18-2
  - 変更
    - 変更をレプリケートしない, 9-5
    - レプリケーションの変更, 9-2
- 表のオンライン再定義, 8-20
- 表の再定義
  - オンライン
    - レプリケーション, 8-20

## ふ

---

- 複数層マテリアライズド・ビュー
  - 設定, 2-20
- プッシュ
  - 遅延トランザクション・キュー, 9-13

- プロキシ・マテリアライズド・ビュー管理者
  - 作成, 2-7, 2-11, 2-15, 2-26
- プロシージャ・レプリケーション
  - 競合, 7-48
  - 使用, 7-45
  - 制限事項, 7-45
  - トランザクションのシリアル化, 7-48
  - ユーザー定義型, 7-47
  - レプリケーション・サポートの生成, 7-48
- プロパゲータ
  - 登録, 2-5, 2-6, 14-17

## ま

- マスター・グループ
  - オブジェクトの追加, 3-6
  - 監視, 10-4
  - 削除, 20-66
  - 作成, 3-2, 3-5, 20-49
  - 作成用フローチャート, 3-4
  - 静止, 20-106
  - マスター・サイトの削除, 7-40
  - マスター・サイトの追加
    - 静止, 7-30
    - 静止することなく, 7-4
  - レプリケーション・アクティビティの再開, 20-98
- マスター・サイト
  - 監視, 10-2, 10-6
  - クリーン・アップ, 8-11
  - 削除, 7-40, 20-94
  - 作成, 20-8
  - スケジュール・ページ対象, 2-6
  - スケジュール・リンクの対象, 2-17
  - 設定, 2-4
  - 設定用フローチャート, 2-3
  - 対象ユーザーの作成, 2-7, 2-11, 2-15, 2-26
  - 違いの判別, 9-9
  - 追加, 3-9, 7-4
    - オブジェクトレベルのエクスポートまたはインポートの使用, 7-19
    - オフライン・インスタンスエーションの使用, 7-33
    - 外部キー制約, 7-4
    - 自己参照型制約, 3-10, 7-4
    - 循環依存, 3-10, 7-4
    - 制限事項, 7-7
    - 静止, 7-30

- 静止することなく, 7-4
- 全データベースのエクスポートまたはインポートに対する制限事項, 7-5
- 全データベースのエクスポートまたはインポートの使用, 7-8
- フローチャート, 7-9, 7-20
- 変更ベースのリカバリ, 7-8
- 変更ベースのリカバリに対する制限事項, 7-5
- 方法決定のフローチャート, 7-6
- データベース・リンク, 2-16
- 変更の伝播, 14-20
- マスター定義サイトの変更, 7-2
- マスター定義サイト
  - 再配置, 20-93
- マスター表
  - オンラインでの再定義, 8-20
  - 切捨て, 8-20
  - 再編成, 8-20
    - 方法, 8-21
  - への列の追加, 20-96
- マスター・マテリアライズド・ビュー
  - 監視, 10-6
  - 再編成, 8-20
- マテリアライズド・ビュー
  - オフライン・インスタンスエーション, 17-2, 17-4
  - 監視, 10-11, 10-13
  - 削除, 8-10
  - サポートの生成, 20-80
  - セキュリティ, A-7
    - トラステッド・セキュリティとアントラステッド・セキュリティの比較, A-8
  - データ・ディクショナリ・ビュー, 28-1
  - 配置テンプレート
    - ユーザー定義型, 4-2
  - 複数層
    - 設定, 5-4
    - ユーザー定義の競合解消, B-4
  - マスターからの登録解除, 8-14
  - マテリアライズド・ビュー・ログからのページ, 8-11, 8-14
  - リフレッシュ, 4-21, 8-2, 8-34, 15-12, 15-14, 15-16
  - リフレッシュ・グループ
    - 作成, 5-7, 5-15
- マテリアライズド・ビュー・グループ
  - オブジェクトの追加, 5-9, 5-16, 8-39
  - 監視, 10-12

- グループ所有者, 8-36
- 削除, 8-3, 8-9
- 作成, 5-4, 5-7, 5-14, 20-54
- マスターからの登録解除, 8-11
- マスターの変更, 8-2
- リフレッシュ, 20-89
- マテリアライズド・ビュー・サイト
  - オブジェクトの削除, 8-9
  - 監視, 10-11
  - 管理者
    - 作成, 2-21
  - グループ所有者
    - 使用, 8-36
  - 削除, 8-3, 20-69
  - スケジュール・ページ, 2-25, 2-30
  - 設定用フローチャート, 2-20
  - 追加
    - オフライン・インスタンスーションの使用, 8-25
  - データベース・リンク
    - 作成, 2-29, 4-20, 5-5, 5-6, 5-13, 5-14
  - 複数層
    - 設定, 2-20
  - マスターの変更, 20-107
  - マスターへの変更の伝播, 14-20
  - ユーザー
    - 作成, 2-21
  - リフレッシュ
    - 作成, 2-21, 2-28
- マテリアライズド・ビュー・ログ
  - 監視, 10-8
  - 管理, 8-16
    - 領域, 8-17
  - 行のページ
    - 手動, 8-18
  - 切捨て, 8-19
  - 削除, 8-24
  - ページ
    - ページからのマテリアライズド・ビュー, 8-11, 8-14
    - 必要な権限, 8-18
  - 変更, 8-16
    - 必要な権限, 8-16
  - マスターの再編成, 8-20
  - マスター表
    - ページ, 15-8, 15-10
  - マスター表の切捨て, 8-20

- 列の追加, 8-17
- 割り当てられた領域の削減, 8-19
- マルチマスター・レプリケーション
  - 監視, 10-2
  - セキュリティ
    - トラステッド・セキュリティとアントラステッド・セキュリティの比較, A-3

## ゆ

---

- ユーザー
  - 配置テンプレート用の許可, 4-12
  - マスター・サイト, 2-7, 2-11, 2-15
  - マスター・マテリアライズド・ビュー・サイト, 2-26
  - マテリアライズド・ビュー・サイト, 2-21
- 優先グループ
  - サイト優先グループ
    - メンバーの追加, 20-17
  - 削除, 20-72
  - 作成, 20-60
  - メンバーの削除, 20-71, 20-73
  - メンバーの追加, 20-16
  - メンバーの変更
    - 値, 20-33
  - 優先順位, 20-31

## り

---

- リフレッシュ
  - 作成, 2-21, 2-28
- リフレッシュ
  - 監視, 10-16, 10-17
  - マテリアライズド・ビュー, 8-2, 8-34, 15-12, 15-14, 15-16
  - マテリアライズド・ビュー・サイト, 20-89
- リフレッシュ・グループ
  - オブジェクトの追加, 5-11, 5-18
  - 監視, 10-15
  - 削除, 19-6
  - 作成, 5-7, 5-15, 19-6
  - データ・ディクショナリ・ビュー, 28-1
  - メンバーの削除, 19-10
  - メンバーの追加, 19-2
  - リフレッシュ, 8-2
    - 手動, 19-9
  - リフレッシュ間隔
    - 変更, 19-3

## れ

### 列

- マスター表への追加, 20-96
- 列グループ, 6-4, 6-7, 6-12, 6-15, 6-18, 6-23
  - 削除, 20-63
  - 作成, 20-59, 20-84
  - メンバーの削除, 20-64
  - メンバーの追加, 20-7

### 列オブジェクト

- ユーザー定義の競合解消, B-2

### 列サブセット化

- ユーザー定義の競合解消メソッド, B-5

### レプリケーション

#### LONG 列

- LOB への変換, 9-7

#### エラー・キュー

- 管理, 9-18

#### オブジェクト

- 配置テンプレートへの追加, 4-6
- 変更, 6-10, 9-2
- マスター・グループへの追加, 3-6
- マスター・サイトからの削除, 20-67

#### カタログ・ビュー, 10-1, 25-1

#### 環境構築用フローチャート, 1-2

#### 環境の管理, 7-1, 8-1

#### 環境の構築, 1-2

#### 監視, 10-1

- エラー・トランザクション, 10-26
- 遅延トランザクション, 10-20
- パフォーマンス, 10-29
- マスター・環境, 10-2
- マテリアライズド・ビュー環境, 10-11

#### 期間データ型

- 略称, 11-4

#### 競合解消, 6-1

- 一意性競合, 6-26

#### 計画対象, 1-4

#### 再開, 3-14

#### サイト

- 設定, 2-2

#### サイトの設定, 2-2

#### サポートの生成, 3-12

#### 受信者

- 登録, 2-6

#### 使用禁止, 7-48, 9-5, 9-6, 24-3

#### スケジュール・リンク

- 作成, 2-17

#### セキュリティ, A-1

#### 遅延トランザクション

- データ・ディクショナリ・ビュー, 27-1

#### 遅延トランザクション・キュー

- 管理, 9-13

#### データ・ディクショナリ・ビュー, 10-1, 25-1

#### データベース・リンク

- 作成, 2-16

#### 統計

- 消去, 10-32

#### 動的パフォーマンス・ビュー, 26-1

#### トリガー, 9-7

#### 日時データ型

- 略称, 11-4

#### 配置テンプレート

- ユーザー定義型, 4-2

#### パラレル伝播

- 監視, 10-31, 10-32

#### 表間の違いの判別, 9-9

#### プロシージャ・レプリケーション, 7-45

- 制限事項, 7-45

- ユーザー定義型, 7-47

#### プロパゲータ

- 登録, 2-5, 2-6

#### マスター・グループ

- 作成, 3-2

#### マスター・サイト

- 追加, 3-9

#### マテリアライズド・ビュー・グループ

- 作成, 5-4, 5-7, 5-14

#### マテリアライズド・ビュー・ログ

- 管理, 8-16

#### 有効化, 7-48, 9-5, 9-6, 24-3

#### 列グループ, 6-4, 6-7, 6-12, 6-15, 6-18, 6-23

#### レプリケーション・キュー, 9-1

#### レプリケーション・アクティビティの再開, 20-98

#### レプリケーション・オブジェクト

#### 表

- 変更, 9-5

#### 変更, 9-2

#### レプリケーション・カタログ・ビュー, 25-1

#### コメント

- 更新, 7-43

#### レプリケーションの監視, 10-1

レプリケーション・マネージメント API, 11-1

概要, 1-1

競合解消, 6-1

配置テンプレート

    インスタンス化, 4-18

    作成, 4-3

    パッケージ化, 4-13

パッケージ, 35, 36, 11-1

マスター・グループ

    作成, 3-2

マテリアライズド・ビュー・グループ

    作成, 5-4

例, 11-2

レプリケート・オブジェクトの管理, 9-1

レプリケーション環境の管理, 7-1, 8-1

レプリケーション・キューの管理, 9-1

レプリケーション・サイトの設定, 2-2

## わ

---

ワークフロー, 6-37