

Oracle8 Server

レプリケーション

リリース 8.0

1998 年 2 月

部品番号 A56819-1

ORACLE®

Enabling the Information Age™

Oracle8 Server レプリケーション

部品番号 A56819-1

リリース 8.0

第 1 版 : 1998 年 2 月

原本名 : Oracle8 Replication

原本部品番号 : A58245-01

Copyright© Oracle Corporation, 1997, 1998.

All rights reserved.

Printed in Japan.

原著者 : Steve Bobrowski, Gordon Smith.

制限付権利の説明

プログラムの使用、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。

本書の情報は、予告なしに変更されることがあります。本書に問題を見つけたら、当社にコメントをお送りください。オラクル社は、本書の無謬性を保証しません。

危険な用途への使用について

当社製品は、原子力、航空産業、大量輸送、又は医療の分野など、本質的に危険が伴うアプリケーションを用途として特に開発されておりません。当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って行っていただきたく、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、当社および開発元である米国 Oracle Corporation（その関連会社も含みます）は一切責任を負いかねます。

ORACLE は、Oracle Corporation の登録商標です。

本文中の他社の商品名は、それぞれ各社の商標または登録商標です。

目 次

はじめに	i
『 Oracle8 レプリケーション 』 の概要	ii
対象読者	iii
前提知識	iii
『 Oracle8 レプリケーション 』 の構成	iii
表記規則	v
特記事項	v
本文について	v
サンプル・コード	vi
 1 レプリケーションについて	
レプリケーションとは?	1-2
基本レプリケーション	1-2
アドバンスト・レプリケーション	1-3
基本レプリケーションの概念	1-4
基本レプリケーションの使用方法	1-4
読み込み専用の表スナップショット	1-5
スナップショット・リフレッシュ	1-7
基本レプリケーションのその他のオプション	1-10
アドバンスト・レプリケーションの概念	1-11
アドバンスト・レプリケーションの使用方法	1-11
アドバンスト・レプリケーションの構成	1-12
アドバンスト・レプリケーションおよび Oracle Replication Manager	1-15
レプリケーション・オブジェクト、グループ、サイト、カタログ	1-16

Oracle のアドバンスド・レプリケーション・アーキテクチャ	1-17
レプリケーション管理者および伝播担当者、受信者	1-21
レプリケーションの競合	1-22
アドバンスド・レプリケーション固有のオプション	1-25
2 基本レプリケーションの使用方法	
クイック・スタート：基本レプリケーション環境の作成	2-2
簡単な例	2-2
スナップショット用データベースの設定	2-4
必要なスキーマ	2-4
必要なデータベース・リンク	2-5
スナップショット・リフレッシュの計画	2-5
スナップショット・ログの作成	2-6
スナップショット・ログ名	2-8
必要な権限	2-8
スナップショット・ログ作成のタイミング	2-8
主キー・スナップショットの特別要件	2-8
スナップショット・ログ記憶領域パラメータ	2-8
副問合せスナップショットの特別要件	2-9
スナップショット・ログ作成の内部メカニズム	2-10
単純スナップショットの作成	2-10
スナップショット名	2-11
必要な権限	2-11
主キー・スナップショットの特別要件	2-12
スナップショット・リフレッシュの設定	2-12
スナップショットの定義問合せ	2-12
スナップショットの記憶領域の設定	2-13
クラスタ化スナップショット	2-14
データ・ロード・オプション	2-15
副問合せを使用するスナップショットの作成	2-15
副問合せを使用する高度なサブセット化	2-16
副問合せを使用するスナップショットに関する制限事項	2-23

リフレッシュ・グループの作成	2-24
リフレッシュの設定	2-25
リフレッシュ・タイプ	2-27
ロールバック・セグメントの設定	2-27
基本レプリケーション環境の管理	2-27
スナップショット・ログの管理	2-27
読み込み専用スナップショットの管理	2-33
スナップショット・リフレッシュとリフレッシュ・グループの管理	2-36
スナップショットのパフォーマンスのチューニング	2-39
スナップショットの索引作成	2-39
副問合せスナップショットのチューニング	2-40
基本レプリケーションのその他のオプション	2-40
複合スナップショット	2-40
ROWID スナップショット	2-42
スナップショットの個別リフレッシュ	2-42
基本レプリケーション環境の監視	2-43

3 マルチマスター・レプリケーションの使用法

クイック・スタート：マルチマスター・レプリケーション環境の作成	3-2
簡単な例	3-2
マルチマスター・レプリケーションの設定	3-3
レプリケーション・セットアップ・ウィザード	3-3
SNP バックグラウンド・プロセスの起動	3-7
スケジュール・リンクの管理	3-7
スケジュール・リンクの作成	3-8
スケジュール・リンクの編集	3-10
スケジュール・リンクの状態の表示	3-10
スケジュール・リンクの削除	3-11
サイトの遅延トランザクション・キューのページ	3-11
サイトのページ・スケジュールの指定	3-12
サイトの遅延トランザクション・キューの手動ページ	3-13

マスター・グループの管理	3-13
マスター・グループの作成	3-13
マスター・グループの削除	3-15
マスター・グループのレプリケーション・アクティビティの中断	3-15
マスター・グループのレプリケーション・アクティビティの再開	3-16
マスター・グループへのオブジェクトの追加	3-17
マスター・グループのオブジェクトの変更	3-20
マスター・グループからのオブジェクトの削除	3-21
マスター・グループへのマスター・サイトの追加	3-21
マスター・グループからのマスター・サイトの削除	3-22
マスター・グループ・オブジェクトのレプリケーション・サポートの生成	3-23
マスター・グループに関する情報の表示	3-25
マスター・サイト管理に関するその他の情報	3-27
アドバンスド・マルチマスター・レプリケーション・オプション	3-27
パラレル伝播の計画	3-27
レプリケーションの保護メカニズムについて	3-29

4 スナップショット・サイト・レプリケーションの使用方法

クイック・スタート: アドバンスド・レプリケーション・システムへのスナップショット・サイトの追加	4-2
簡単な例	4-2
スナップショット・サイト・レプリケーションの準備	4-4
レプリケーション・セットアップ・ウィザード	4-5
スケジュール・リンクおよびスナップショット・リフレッシュの計画	4-9
SNP バックグラウンド・プロセスの起動	4-9
スナップショット・ログの管理	4-10
スナップショット・ログの作成	4-10
スナップショット・ログの変更	4-10
スナップショット・ログの削除	4-11
スナップショット・グループの管理	4-11
スナップショット・グループの作成	4-11
マスター・サイトにスナップショット・グループを登録する	4-13
マスター・サイトのスナップショット・グループを登録解除する	4-13

スナップショット・グループにオブジェクトを追加する	4-13
スナップショット・グループのオブジェクトの変更	4-14
スナップショット・グループのオブジェクトの削除	4-14
スナップショット・グループの編集	4-14
スナップショット・グループの削除	4-15
スナップショットの管理	4-15
スナップショットの作成	4-16
副問合せを使用するスナップショットの作成	4-18
更新可能スナップショットのレプリケーション・サポートの再生成	4-18
スナップショットの変更	4-19
スナップショットの削除	4-19
リフレッシュ・グループの管理	4-20
リフレッシュ・グループの作成	4-20
リフレッシュ・グループにスナップショットを追加する	4-21
リフレッシュ・グループからスナップショットを削除する	4-21
スナップショット・グループのリフレッシュ設定を変更する	4-21
スナップショットのグループを手動でリフレッシュする	4-21
リフレッシュ・グループの削除	4-22
スナップショット・サイトの管理に関するその他の情報	4-22
データ・ディクショナリ・ビュー	4-22

5 競合の解消

レプリケーションの競合の概要	5-2
データ要件およびアプリケーション要件の理解	5-2
レプリケーションの競合の種類	5-3
競合の防止	5-3
マスター・サイトでの競合の検出	5-5
競合の解消	5-6
競合解消方法	5-7
競合解消の構成の概要	5-10
競合解消の設計および準備のガイドライン	5-10
競合解消のインプリメント	5-11

更新の競合解消方法の構成	5-11
列グループの作成	5-11
列グループの列の追加および削除	5-12
列グループの削除	5-12
更新の競合解消方法の管理	5-13
ビルトインの更新の競合解消方法	5-13
更新の競合解消のための優先順位グループの使用	5-18
更新の競合解消のためのサイト優先順位グループの使用	5-23
サンプル・タイムスタンプおよびサイト・メンテナンス・トリガー	5-26
一意性の競合解消方法の構成	5-28
一意性の競合解消方法の割当て	5-28
一意性の競合解消方法の削除	5-28
ビルトインの一意性の競合解消方法	5-29
削除の競合解消の構成	5-30
削除の競合解消方法の割当て	5-30
削除の競合解消方法の削除	5-30
データ収束の保証	5-31
順序の競合の防止	5-32
更新の競合解消のためのデータ伝播の最小化	5-34
通信の最小化の例	5-34
データ伝播をさらに削減する	5-37
ユーザー定義の競合解消方法	5-39
競合解消方法のパラメータ	5-39
更新の競合解消	5-39
一意性の競合解消	5-40
削除の競合解消	5-40
制限	5-41
ユーザー定義の競合解消方法の例	5-41
ユーザー定義の競合通知方法	5-43
競合通知ログの作成	5-43
競合通知パッケージの作成	5-43
競合解消情報の表示	5-46

6 レプリケート環境の管理方法

マスター・グループおよびスナップショット・グループの上級管理テクニック	6-2
マスター・グループ内での DDL の実行	6-2
マスター・グループの妥当性検査	6-2
マスター・グループの定義サイトの再配置	6-3
スナップショット・グループのマスター・サイトの変更	6-3
アドバンスド・レプリケーション・システムの監視	6-4
管理要求の管理	6-4
遅延トランザクションの管理	6-8
エラー・トランザクションの管理	6-9
ローカル・ジョブの管理	6-11
レプリケーション・システムにおけるデータベースのバックアップおよび回復	6-14
インポートされたデータに対するチェックの実行	6-14
正常に終了した競合解消の監査	6-15
競合解消の統計収集	6-15
競合解消の統計の表示	6-15
競合解消の統計収集の取消し	6-15
統計情報の削除	6-15
レプリケート表相互間の相違の判別	6-16
DIFFERENCES	6-16
RECTIFY	6-16
一般的な問題のトラブルシューティング	6-19
データベース・リンクに関する問題の診断方法	6-19
マスター・サイトに関する問題の診断方法	6-19
遅延トランザクション・キューに関する問題の診断方法	6-22
スナップショットに関する問題の診断方法	6-23
ビュー内のコメント・フィールドの更新	6-25

7 上級テクニック

プロシージャ・レプリケーションの使用方法	7-2
プロシージャ・レプリケーションの制限	7-2
トランザクションの直列化	7-3

レプリケートされたプロシージャのサポートの生成	7-3
同期データ伝播の使用方法	7-6
同期データ伝播について	7-6
アドバンスト・レプリケーション環境への新規サイトの追加	7-8
マスター・サイトのデータ伝播モードの変更	7-10
動作保証の設計	7-10
Oracle Parallel Server とアドバンスト・レプリケーション	7-11
動作保証の設計	7-12
動作保証が可能なシステムの実現	7-13
スナップショット・クローニングとオフライン・インスタンス化	7-13
基本レプリケーション環境のスナップショット・クローニング	7-14
アドバンスト・レプリケーション・システムでのマスター・サイトのオフライン・ インスタンス化	7-15
アドバンスト・レプリケーション・システムでのスナップショット・サイトのオフライン・ インスタンス化	7-16
マルチマスター・レプリケーションのセキュリティ設定	7-17
スナップショット・レプリケーションのセキュリティ設定	7-21
アドバンスト・レプリケーション・スナップショット・サイトに対する代替セキュリティの設定	7-23
削除の競合の防止	7-24
動的所有権の競合防止の使用方法	7-26
ワークフロー	7-26
トークン渡し	7-27
行の所有者の判別	7-29
所有権の取得	7-29
変更の適用	7-30
変更をレプリケートしないで表を変更する	7-30
アドバンスト・レプリケーションを使用禁止にする	7-31
アドバンスト・レプリケーションを再び使用可能にする	7-32
トリガーおよびレプリケーション	7-32
スナップショットのレプリケーションを使用可能 / 使用禁止にする	7-32

8 遅延トランザクションの使用方法

遅延トランザクションに関する情報のリスト	8-2
遅延トランザクションの作成.....	8-2
セキュリティ	8-3
接続先の指定	8-3
遅延トランザクションの開始	8-4
遅延リモート・プロシージャ・コール	8-4
遅延コール用パラメータ値をキューに入れる	8-5
DEFDEFAULTDEST ビューへの接続先の追加.....	8-6
DEFDEFAULTDEST ビューからの接続先の取消し.....	8-6
遅延トランザクションの実行	8-6
LOB の格納	8-7
RPC のための格納の DEFLOB ビュー	8-7

9 レプリケーション管理 API リファレンス

パッケージ	9-2
Oracle レプリケーション管理 API の使用例.....	9-2
前提条件	9-3
Replication Manager と Oracle レプリケーション管理 API.....	9-3
DBMS_DEFER パッケージ.....	9-4
DBMS_DEFER.CALL.....	9-4
目的	9-4
構文	9-4
DBMS_DEFER.COMMIT_WORK.....	9-5
目的	9-5
構文	9-5
DBMS_DEFER.datatype_ARG	9-6
目的	9-6
DBMS_DEFER.TRANSACTION.....	9-7
目的	9-7
構文	9-7
DBMS_DEFER_QUERY パッケージ	9-8

DBMS_DEFER_QUERY.GET_ARG_FORM	9-8
目的	9-8
構文	9-8
DBMS_DEFER_QUERY.GET_ARG_TYPE.....	9-9
目的	9-9
構文	9-9
DBMS_DEFER_QUERY.GET_CALL_ARGS.....	9-10
目的	9-10
構文	9-10
DBMS_DEFER_QUERY.GET_datatype_ARG	9-11
目的	9-11
構文	9-12
DBMS_DEFER_SYS パッケージ	9-13
DBMS_DEFER_SYS.ADD_DEFAULT_DEST	9-14
目的	9-14
構文	9-14
DBMS_DEFER_SYS.DELETE_DEFAULT_DEST.....	9-14
目的	9-14
構文	9-14
DBMS_DEFER_SYS.DELETE_DEF_DESTINATION	9-15
目的	9-15
構文	9-15
DBMS_DEFER_SYS.DELETE_ERROR.....	9-15
目的	9-15
構文	9-15
DBMS_DEFER_SYS.DELETE_TRAN.....	9-16
目的	9-16
構文	9-16
DBMS_DEFER_SYS.DISABLED	9-16
目的	9-16
構文	9-17
DBMS_DEFER_SYS.EXCLUDE_PUSH.....	9-17

目的	9-17
構文	9-17
DBMS_DEFER_SYS.EXECUTE_ERROR.....	9-18
目的	9-18
構文	9-18
DBMS_DEFER_SYS.EXECUTE_ERROR_AS_USER.....	9-19
目的	9-19
構文	9-19
DBMS_DEFER_SYS.PURGE	9-20
目的	9-20
構文	9-20
DBMS_DEFER_SYS.PUSH.....	9-22
目的	9-22
構文	9-22
DBMS_DEFER_SYS.REGISTER_PROPAGATOR.....	9-24
目的	9-24
構文	9-24
DBMS_DEFER_SYS.SCHEDULE_PURGE	9-25
目的	9-25
構文	9-25
DBMS_DEFER_SYS.SCHEDULE_PUSH.....	9-26
目的	9-26
構文	9-27
DBMS_DEFER_SYS.SET_DISABLED.....	9-28
目的	9-28
構文	9-29
DBMS_DEFER_SYS.UNREGISTER_PROPAGATOR.....	9-29
目的	9-29
構文	9-29
DBMS_DEFER_SYS.UNSCHEDULE_PURGE	9-30
目的	9-30
構文	9-30

DBMS_DEFER_SYS.UNSCHEDULE_PUSH	9-30
目的	9-30
構文	9-30
DBMS_OFFLINE_OG パッケージ	9-31
DBMS_OFFLINE_OG.BEGIN_INSTANTIATION	9-31
目的	9-31
構文	9-31
DBMS_OFFLINE_OG.BEGIN_LOAD	9-32
目的	9-32
構文	9-32
DBMS_OFFLINE_OG.END_INSTANTIATION	9-33
目的	9-33
構文	9-33
DBMS_OFFLINE_OG.END_LOAD	9-34
目的	9-34
構文	9-34
DBMS_OFFLINE_OG.RESUME_SUBSET_OF_MASTERS	9-35
目的	9-35
構文	9-35
DBMS_OFFLINE_SNAPSHOT パッケージ	9-36
DBMS_OFFLINE_SNAPSHOT.BEGIN_LOAD	9-36
目的	9-36
構文	9-37
DBMS_OFFLINE_SNAPSHOT.END_LOAD	9-38
目的	9-38
構文	9-38
DBMS_RECTIFIER_DIFF パッケージ	9-39
DBMS_RECTIFIER_DIFF.DIFFERENCES	9-39
目的	9-39
構文	9-39
制限	9-41
DBMS_RECTIFIER_DIFF.RECTIFY	9-42

目的	9-42
構文	9-42
DBMS_REFRESH パッケージ	9-43
DBMS_REFRESH.ADD	9-44
目的	9-44
構文	9-44
DBMS_REFRESH.CHANGE	9-45
目的	9-45
構文	9-45
DBMS_REFRESH.DESTROY	9-47
目的	9-47
構文	9-47
DBMS_REFRESH.MAKE.....	9-48
目的	9-48
構文	9-48
DBMS_REFRESH.REFRESH.....	9-50
目的	9-50
構文	9-50
DBMS_REFRESH.SUBTRACT	9-51
目的	9-51
構文	9-51
DBMS_REPCAT パッケージ	9-52
DBMS_REPCAT.ADD_GROUPED_COLUMN.....	9-54
目的	9-54
構文	9-54
DBMS_REPCAT.ADD_MASTER_DATABASE.....	9-55
目的	9-55
構文	9-55
DBMS_REPCAT.ADD_PRIORITY_datatype.....	9-57
目的	9-57
構文	9-57
DBMS_REPCAT.ADD_SITE_PRIORITY_SITE.....	9-58

目的	9-58
構文	9-59
DBMS_REPCAT.ADD_conflicttype_RESOLUTION.....	9-60
目的	9-60
構文	9-60
DBMS_REPCAT.ALTER_MASTER_PROPAGATION.....	9-63
目的	9-63
構文	9-63
DBMS_REPCAT.ALTER_MASTER_REPOBJECT	9-64
目的	9-64
構文	9-64
DBMS_REPCAT.ALTER_PRIORITY	9-66
目的	9-66
構文	9-66
DBMS_REPCAT.ALTER_PRIORITY_datatype	9-67
目的	9-67
構文	9-67
DBMS_REPCAT.ALTER_SITE_PRIORITY	9-68
目的	9-68
構文	9-69
DBMS_REPCAT.ALTER_SITE_PRIORITY_SITE	9-70
目的	9-70
構文	9-70
DBMS_REPCAT.ALTER_SNAPSHOT_PROPAGATION.....	9-71
目的	9-71
構文	9-71
DBMS_REPCAT.CANCEL_STATISTICS	9-72
目的	9-72
構文	9-72
DBMS_REPCAT.COMMENT_ON_COLUMN_GROUP	9-72
目的	9-72
構文	9-72

DBMS_REPCAT.COMMENT_ON_PRIORITY_GROUP/	9-73
DBMS_REPCAT.COMMENT_ON_SITE_PRIORITY	9-73
目的	9-73
構文	9-73
DBMS_REPCAT.COMMENT_ON_REPGROUP	9-74
目的	9-74
構文	9-74
DBMS_REPCAT.COMMENT_ON_REPSITES.....	9-75
目的	9-75
構文	9-75
DBMS_REPCAT.COMMENT_ON_REPOBJECT	9-76
目的	9-76
構文	9-76
DBMS_REPCAT.COMMENT_ON_conflicttype_RESOLUTION.....	9-77
目的	9-77
構文	9-78
DBMS_REPCAT.CREATE_MASTER_REPGROUP	9-79
目的	9-79
構文	9-79
DBMS_REPCAT.CREATE_MASTER_REPOBJECT	9-80
目的	9-80
構文	9-80
DBMS_REPCAT.CREATE_SNAPSHOT_REPGROUP	9-83
目的	9-83
構文	9-83
DBMS_REPCAT.CREATE_SNAPSHOT_REPOBJECT	9-84
目的	9-84
構文	9-85
DBMS_REPCAT.DEFINE_COLUMN_GROUP	9-87
目的	9-87
構文	9-87
DBMS_REPCAT.DEFINE_PRIORITY_GROUP	9-88

目的	9-88
構文	9-88
DBMS_REPCAT.DEFINE_SITE_PRIORITY	9-89
目的	9-89
構文	9-89
DBMS_REPCAT.DO_DEFERRED_REPCAT_ADMIN	9-90
目的	9-90
構文	9-90
DBMS_REPCAT.DROP_COLUMN_GROUP	9-91
目的	9-91
構文	9-91
DBMS_REPCAT.DROP_GROUPED_COLUMN	9-92
目的	9-92
構文	9-92
DBMS_REPCAT.DROP_MASTER_REPGROUP	9-93
目的	9-93
構文	9-93
DBMS_REPCAT.DROP_MASTER_REPOBJECT	9-94
目的	9-94
構文	9-94
DBMS_REPCAT.DROP_PRIORITY	9-95
目的	9-95
構文	9-95
DBMS_REPCAT.DROP_PRIORITY_GROUP	9-96
目的	9-96
構文	9-96
DBMS_REPCAT.DROP_PRIORITY_datatype	9-97
目的	9-97
構文	9-97
DBMS_REPCAT.DROP_SITE_PRIORITY	9-98
目的	9-98
構文	9-98

DBMS_REPCAT.DROP_SITE_PRIORITY_SITE	9-99
目的	9-99
構文	9-99
DBMS_REPCAT.DROP_SNAPSHOT_REPGROUP.....	9-100
目的	9-100
構文	9-100
DBMS_REPCAT.DROP_SNAPSHOT_REPOBJECT.....	9-101
目的	9-101
構文	9-101
DBMS_REPCAT.DROP_conflictype_RESOLUTION.....	9-102
目的	9-102
構文	9-102
DBMS_REPCAT.EXECUTE_DDL	9-104
目的	9-104
構文	9-104
DBMS_REPCAT.GENERATE_REPLICATION_PACKAGE	9-105
目的	9-105
構文	9-105
DBMS_REPCAT.GENERATE_REPLICATION_SUPPORT	9-106
目的	9-106
構文	9-106
DBMS_REPCAT.GENERATE_REPLICATION_TRIGGER.....	9-108
目的	9-108
構文	9-108
伝播モードの変更	9-110
DBMS_REPCAT.GENERATE_SNAPSHOT_SUPPORT	9-110
目的	9-110
構文	9-110
DBMS_REPCAT.MAKE_COLUMN_GROUP	9-112
目的	9-112
構文	9-112
DBMS_REPCAT.PURGE_MASTER_LOG	9-113

目的	9-113
構文	9-113
DBMS_REPCAT.PURGE_STATISTICS	9-113
目的	9-113
構文	9-114
DBMS_REPCAT.REFRESH_SNAPSHOT_REPGROUP	9-114
目的	9-114
構文	9-114
DBMS_REPCAT.REGISTER_SNAPSHOT_REPGROUP	9-116
目的	9-116
構文	9-116
DBMS_REPCAT.REGISTER_STATISTICS	9-117
目的	9-117
構文	9-117
DBMS_REPCAT.RELOCATE_MASTERDEF	9-117
目的	9-117
構文	9-117
使用上の注意事項	9-118
DBMS_REPCAT.REMOVE_MASTER_DATABASES	9-119
目的	9-119
構文	9-119
DBMS_REPCAT.REPCAT_IMPORT_CHECK	9-120
目的	9-120
構文	9-120
DBMS_REPCAT.RESUME_MASTER_ACTIVITY	9-121
目的	9-121
構文	9-121
DBMS_REPCAT.SEND_AND_COMPARE_OLD_VALUES	9-121
目的	9-121
構文	9-122
DBMS_REPCAT.SET_COLUMNS	9-123
目的	9-123

構文	9-123
DBMS_REPCAT.SUSPEND_MASTER_ACTIVITY	9-124
目的	9-124
構文	9-124
DBMS_REPCAT.SWITCH_SNAPSHOT_MASTER	9-125
目的	9-125
構文	9-125
DBMS_REPCAT.UNREGISTER_SNAPSHOT_REPGROUP	9-126
目的	9-126
構文	9-126
DBMS_REPCAT.VALIDATE	9-126
目的	9-126
構文	9-126
DBMS_REPCAT.WAIT_MASTER_LOG	9-129
目的	9-129
構文	9-129
DBMS_REPCAT_ADMIN パッケージ	9-129
DBMS_REPCAT_ADMIN.GRANT_ADMIN_ANY_SCHEMA	9-130
目的	9-130
構文	9-130
DBMS_REPCAT_ADMIN.GRANT_ADMIN_SCHEMA	9-130
目的	9-130
構文	9-130
DBMS_REPCAT_ADMIN.REVOKE_ADMIN_ANY_SCHEMA	9-131
目的	9-131
構文	9-131
DBMS_REPCAT_ADMIN.REVOKE_ADMIN_SCHEMA	9-132
目的	9-132
構文	9-132
DBMS_REPCAT_AUTH パッケージ	9-132
DBMS_REPCAT_AUTH.GRANT_SURROGATE_REPCAT	9-132
目的	9-132

構文	9-132
DBMS_REPCAT_AUTH.REVOKE_SURROGATE_REPCAT	9-133
目的	9-133
構文	9-133
DBMS_REPUTIL パッケージ	9-133
DBMS_REPUTIL.REPLICATION_OFF	9-134
目的	9-134
構文	9-134
DBMS_REPUTIL.REPLICATION_ON	9-134
目的	9-134
構文	9-134
DBMS_SNAPSHOT パッケージ	9-134
DBMS_SNAPSHOT.BEGIN_TABLE_REORGANIZATION	9-135
目的	9-135
構文	9-135
DBMS_SNAPSHOT.END_TABLE_REORGANIZATION	9-135
目的	9-135
構文	9-135
DBMS_SNAPSHOT.I_AM_A_REFRESH	9-136
目的	9-136
構文	9-136
DBMS_SNAPSHOT.PURGE_LOG	9-136
目的	9-136
構文	9-136
DBMS_SNAPSHOT.REFRESH	9-137
目的	9-137
構文	9-137
DBMS_SNAPSHOT.REGISTER_SNAPSHOT	9-139
目的	9-139
構文	9-140
DBMS_SNAPSHOT.SET_I_AM_A_REFRESH	9-141
目的	9-141

構文	9-141
DBMS_SNAPSHOT.UNREGISTER_SNAPSHOT.....	9-141
目的	9-141
構文	9-141
パッケージ変数	9-143

10 データ・ディクショナリ・ビュー

レプリケーション・カタログ・ビュー	10-2
REPGROUP ビュー	10-3
REPCATLOG ビュー	10-5
REPCOLUMN ビュー	10-6
REPCOLUMN_GROUP ビュー	10-6
REPCONFLICT ビュー	10-7
REPDDL ビュー	10-7
REPGENERATED ビュー	10-8
REPGROUPED_COLUMN ビュー	10-8
REPKEY_COLUMNS ビュー	10-9
REPOBJECT ビュー	10-9
REPPARAMETER_COLUMN ビュー	10-10
REPPRIORITY ビュー	10-11
REPPRIORITY_GROUP ビュー	10-12
REPPROP ビュー	10-12
REPRESOLUTION ビュー	10-13
REPRESOL_STATS_CONTROL ビュー	10-14
REPRESOLUTION_METHOD ビュー	10-14
REPRESOLUTION_STATISTICS ビュー	10-15
REPSITES ビュー	10-16
REPGENOBJECTS ビュー	10-16
遅延トランザクション・ビュー	10-17
DEFCALL ビュー	10-18
DEFCALLDEST ビュー	10-18
DEFDEFAULTDEST ビュー	10-18

DEFERRCOUNT ビュー	10-19
DEFERROR ビュー	10-19
DEFLOB ビュー	10-20
DEFPROPAGATOR ビュー	10-20
DEFSCCHEDULE ビュー	10-21
DEFTRAN ビュー	10-22
DEFTRANDEST ビュー	10-22
スナップショットおよびスナップショット・リフレッシュ・グループ・ビュー.....	10-22
SNAPSHOTS ビュー	10-23
REGISTERED_SNAPSHOTS ビュー	10-24
SNAPSHOTS_LOGS ビュー	10-25
SNAPSHOT_REFRESH_TIMES ビュー	10-26
REFRESH ビュー	10-26
REFRESH_CHILDREN ビュー	10-28

A 新機能

パフォーマンスの拡張.....	A-2
遅延トランザクションの平行伝播	A-2
内部化されたレプリケーション・トリガー	A-2
データ伝播の縮小	A-2
副問合せに基づくデータのサブセット化.....	A-3
大容量オブジェクト・データ型 (LOB) のサポート	A-3
管理方法の改善と容易な使用性	A-3
きめ細かな休止	A-3
主キー・スナップショット	A-3
マスター・サイトでのスナップショット登録	A-3
高速リフレッシュ機能での表の再編成	A-4
オフライン・インスタンスエーションのサポート	A-4
更新可能スナップショットの遅延制約	A-4
検査プロシージャ	A-4
パーティション表と索引	A-4
拡張されたシステム・ベースのセキュリティ・モデル	A-5

Replication Manager の新しい機能	A-6
B 移行および互換性	
移行の概要	B-2
すべてのサイトの一括移行	B-3
段階的な移行	B-5
Oracle7 マスター・サイトの段階的移行の準備	B-5
スナップショット・サイトの段階的な移行	B-6
マスター・サイトの段階的移行	B-7
エクスポート/インポートを使用する移行	B-10
主キー・スナップショットへのアップグレード	B-11
マスター・サイトでの主キー・スナップショットの変換	B-11
スナップショット・サイトでの主キー・スナップショットの変換	B-11
Oracle8 への移行を必要とする機能	B-12
旧プロシージャ	B-13
索 引	

はじめに

ここでは次のトピックについて説明します。

- 『Oracle8 レプリケーション』の概要
- 対象読者
- 『Oracle8 レプリケーション』の構成
- 表記規則

『Oracle8 Server レプリケーション』には、Oracle8 と Oracle8 Enterprise Edition の両方に関する情報が含まれています。このマニュアルで説明する機能のなかには、Oracle8 Enterprise Edition だけで使用できるものがあります。さらに、そのような機能のなかには、特定のオプション（たとえば、Objects Option ）を購入した場合だけに使用できるものがあります。

Oracle8 と Oracle8 Enterprise Edition の違いの詳細は、『Oracle8 と Oracle8 Enterprise Edition の解説』を参照してください。このマニュアルでは、両製品に共通の機能と、Oracle8 Enterprise Edition または特定のオプションだけで使用できる機能について説明します。

『 Oracle8 レプリケーション 』の概要

このマニュアルではOracle8 Serverのレプリケーション機能について説明します。同期レプリケーション機能を使用するには、Oracle の拡張レプリケーション・オプションをインストールしておく必要があります。基本レプリケーション（読み込み専用スナップショットと更新可能スナップショット）は、Oracle の標準分散機能です。プロシージャ・レプリケーションを使用するには、PL/SQL と拡張レプリケーション・オプションが必要です。

このマニュアルの情報は、すべてのオペレーティング・システムで実行されている Oracle8 Server に適用します。次のトピックがあります。

- 読み込み専用スナップショット
- アドバンスド・レプリケーション機能
 - 更新可能スナップショット
 - マルチマスター・レプリケーション
 - スナップショット・サイト・レプリケーション
 - 競合の解消
- レプリケート環境の管理
- 上級テクニック
- トラブルシューティング
- ジョブ・キューの使用方法
- 遅延トランザクション

対象読者

このマニュアルは、アドバンスド Oracle8 分散システムを開発およびメンテナンスするアプリケーションの開発者とデータベース管理者を対象にしています。

前提知識

このマニュアルの読者は、リレーショナル・データベースの概念、および分散データベースの管理、PL/SQL（プロシージャ・レプリケーションを使用する場合）、Oracle レプリケート環境を動作させるオペレーティング・システムに精通している必要があります。

また、このマニュアルの読者は、次に示すマニュアルを読み、その内容を理解しておく必要があります。

- 『Oracle8 Server 概要』
- 『Oracle8 Server 管理者ガイド』
- 『Oracle8 Server 分散システム』
- 『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）

『Oracle8 レプリケーション』の構成

このマニュアルには、次のとおり 10 の章と 2 つの付録があります。

第 1 章：レプリケーションについて

Oracle レプリケーションの概念および用語を紹介します。

第 2 章：基本レプリケーションの使用方法

読み込み専用スナップショットを使用して、基本的なプライマリ・サイト・レプリケーションを実行する方法を説明します。

第 3 章：マルチマスター・レプリケーションの使用方法

Oracle アドバンスド・レプリケーション機能を使用して、マルチマスター・レプリケート環境を作成し、メンテナンスする方法を説明します。

第 4 章：スナップショット・サイト・レプリケーションの使用方法

Oracle アドバンスド・レプリケーション機能を使用して、スナップショット・サイトを作成しメンテナンスする方法を説明します。更新可能スナップショットについても説明します。

第 5 章：競合の解消

Oracle 提供の競合解消方法を使用して、レプリケート環境でのデータの動的所有または共有が原因で発生する競合を解消する方法について説明します。

第 6 章：レプリケート環境の管理方法

未解決のレプリケーション・エラーの検出および解決方法、さらに正常な競合解消の監視方法について説明します。

第7章：上級テクニック

独自の競合解消ルーチンの作成、フェイルオーバー・サイトのインプリメント、トークン受渡しのインプリメント、プロシージャ・レプリケーションの使用、削除の処理など上級のレプリケーション・テクニックについて説明します。

第8章：遅延トランザクションの使用法

リモート・サイトで遅延実行用のトランザクションを実行する方法について説明します。

第9章：レプリケーション管理 API リファレンス

レプリケート環境の実現に必要なパッケージ・プロシージャのパラメータ、およびこれらのプロシージャで発生する可能性のある例外について説明します。

第10章：データ・ディクショナリ・ビュー

遅延トランザクションおよび読み込み専用スナップショット、アドバンスト・レプリケーション機能を使用するときに便利なビューについて説明します。

付録 A: 新機能

このリリースの新機能について簡単に説明し、詳しい情報が記載されている項を示します。

付録 B: 移行および互換性

拡張レプリケーション・オプションのリリース⁸とそれより前のリリースとの互換性の問題について説明します。

表記規則

このマニュアルで使用する表記上の規則について説明します。

特記事項

このマニュアルの本文では、特別な情報は次のように示します。

注意：特別な情報または予備の情報を示します。

注意：特別な注意または警戒が必要な事柄に関する、特に重要な項目を示します。

追加情報：詳細な情報を得られる参照箇所を示します。

本文について

次に、このマニュアルの本文で使用する規則について説明します。

大文字

大文字テキストは、コマンド・キーワード、オブジェクト名、パラメータ、ファイル名などを明示する場合に使用します。

次に例文を示します。プライベート・ロールバック・セグメントを作成する場合、ロールバック・セグメントの名前はパラメータ・ファイルの ROLLBACK_SEGMENTS パラメータに含めなければなりません。

かぎカッコ

かぎカッコは、語句の定義またはマニュアルのタイトル、強調する語句を示します。

定義の例：「データベース」は1つの単位として取り扱われるデータの集合体です。データベースの一般的な目的は、関連する情報を保存および検索することです。

他の資料を参照する例：詳細は、『Oracle8 Server チューニング』を参照してください。

強調する語句の例：データベースは定期的に「バックアップ」する必要があります。

サンプル・コード

SQLおよびServer Manager行モード、SQL*Plusのコマンドまたは文は、本文と区別するために、クーリエ・フォントで示されています。たとえば、次のようになります。

```
INSERT INTO emp (empno, ename) VALUES (1000, 'SMITH');  
ALTER TABLESPACE users ADD DATAFILE 'users2.ora' SIZE 50K;
```

例文には、カンマや引用符などの句読点を含むことがあります。例文の句読点は、すべて必須です。例文は、すべてセミコロン(;)で終了します。セミコロンまたはその他の終了文字が文を終了するために必須であるかどうかは、アプリケーションによって異なります。

例文の大文字は、Oracle SQL のキーワードを示しています。ただし、文を書くときには、キーワードの大文字と小文字は区別されません。

例文の小文字は、その単語が単なる例として使われていることを示しています。たとえば、表、列、ファイルの名前などを表します。

レプリケーションについて

この章では、Oracle のレプリケーション機能に関する基本的な概念および用語について説明します。

- レプリケーションとは？
- 基本レプリケーションの概念
- アドバンスド・レプリケーションの概念

注意: Trusted Oracleを使用している場合、その環境でレプリケーションを使用する方法については、Trusted Oracle のマニュアルを参照してください。

このマニュアルの第 3 章およびそれ以降の項目で説明するように、アドバンスド・レプリケーションは Oracle8 Enterprise Edition がないと使用できません。

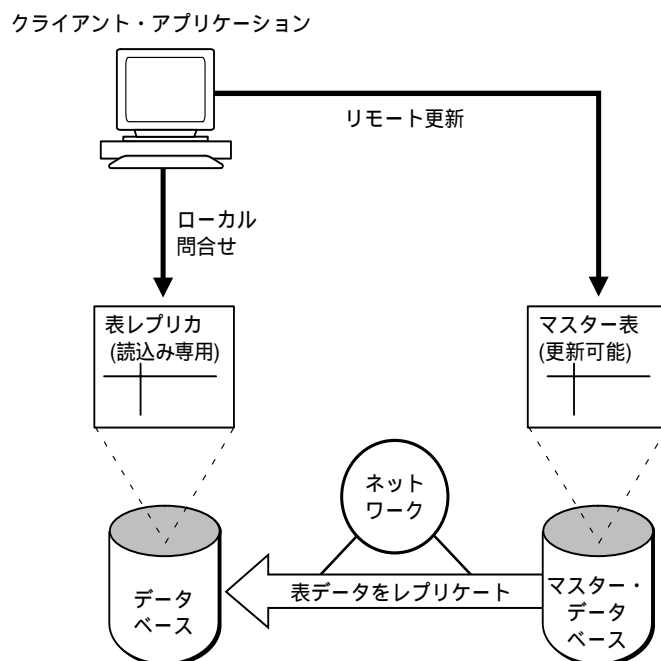
レプリケーションとは？

レプリケーションとは、分散データベースを構成する複数のデータベースのスキーマ・オブジェクトをコピーおよびメンテナンスする処理です。レプリケーションを使用すると、代替データにアクセスできるため、パフォーマンスが向上し、アプリケーションの可用性が保証されます。たとえば、ネットワーク・トラフィックを最小に抑え、パフォーマンスを最大にするために、アプリケーションでは通常リモート・サーバーではなく、ローカル・データベースにアクセスすることができます。ローカル・サーバーでエラーが発生しても、レプリケート・データが存在する他のサーバーがアクセス可能な場合は、アプリケーションは処理を継続できます。Oracle では、基本レプリケーションおよびアドバンスド・レプリケーションという2つの形式のレプリケーションをサポートしています。

基本レプリケーション

基本レプリケーションでは、データ・レプリカを使用して、プライマリ（"マスター"）サイトから作成された表データに読み専用でアクセスすることができます。アプリケーションは、ネットワークの可用性にかかわらず、ネットワークにアクセスせずにローカルのデータ・レプリカのデータを問い合わせることができます。ただし、システム全体で使用するアプリケーションの場合、更新が必要なときはプライマリ・サイトでデータにアクセスしなければなりません。図 1-1 は、基本レプリケーションの例です。

図 1-1 基本的な読み専用のレプリケーション

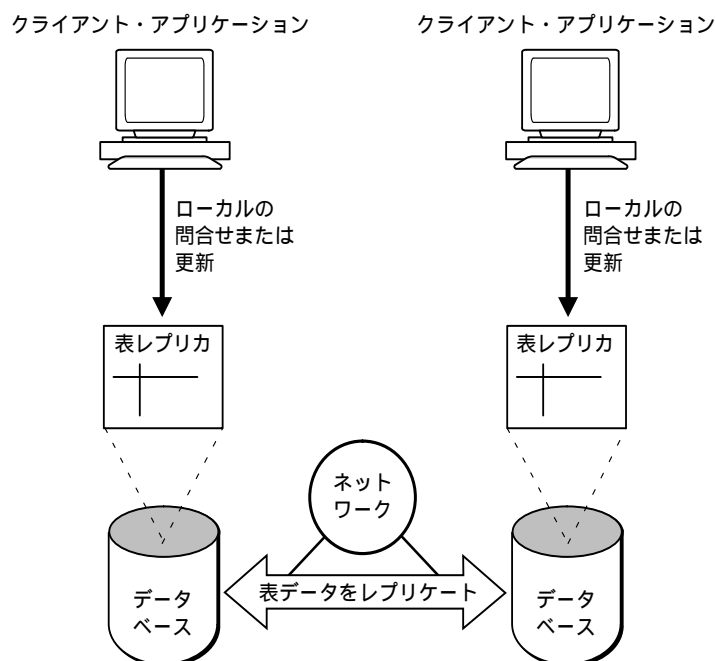


Oracle Serverでは、読み専用表スナップショットを使用する基本的な読み専用レプリケーション環境をサポートしています。基本レプリケーションおよび読み専用スナップショットの詳細は、1-4 ページの「基本レプリケーションの概念」を参照してください。

アドバンスド・レプリケーション

アドバンスド・レプリケーションでは、基本的な読み専用レプリケーションの機能が拡張されており、アプリケーションはレプリケート・データベース・システムのどこからでも表レプリカを更新できます。アドバンスド・レプリケーションの場合、システムのどこに位置するデータ・レプリカに対しても、表のデータに読みおよび更新アクセスができます。関連する Oracle データベース・サーバーで、すべての表レプリカのデータが自動的に収集され、グローバルなトランザクション一貫性およびデータ整合性が保証されます。図 1-2 は、アドバンスド・レプリケーションの例です。

図 1-2 アドバンスド・レプリケーション



Oracle Serverでは、何通りの構成によって、アドバンスド・レプリケーション環境の要件をサポートできます。アドバンスド・レプリケーション・システムの詳細は、1-11ページの「アドバンスド・レプリケーションの概念」を参照してください。

基本レプリケーションの概念

基本レプリケーション環境では、プライマリ・サイトから作成された表データに読み専用でアクセスするアプリケーションがサポートされます。以降の項では、基本レプリケーション環境の基礎概念を説明します。

- 基本レプリケーションの使用方法
- 読み専用の表スナップショット
- スナップショット・リフレッシュ

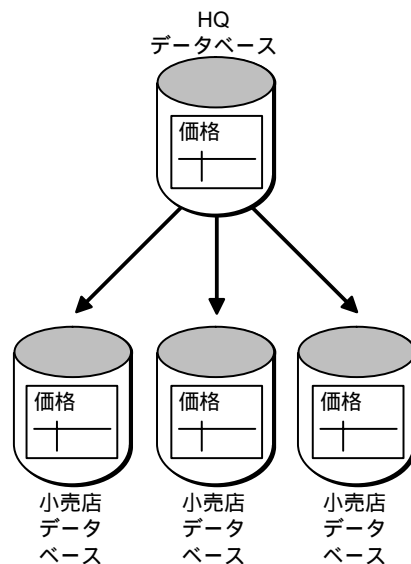
基本レプリケーションの使用方法

基本的な読み専用のデータ・レプリケーションは次に説明するいくつかのタイプのアプリケーションに有効です。

情報の配布

基本レプリケーションは情報の配布に便利です。たとえば、膨大な数の消費者がいるデパート・チェーンの運用を考えてみます。この種の業務では、製品価格情報は常に使用可能で、比較的新しく、すべての小売店で一貫していることが重要です。これらの要件を満たすため、各小売店は独自の製品価格データのコピーを管理して、主価格表をもとにそのデータを夜間にリフレッシュできます。

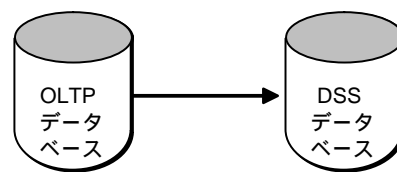
図 1-3 情報の配布



情報のオフロード

基本レプリケーションは、データベース全体をレプリケートする、または情報をオフロードする手段として便利です。たとえば、大量のトランザクションを処理するシステムのパフォーマンスが大幅に低下しているときは、複製データベースを管理して、意思決定支援アプリケーションの問合せ要求を分離すると便利です。

図 1-4 情報のオフロード



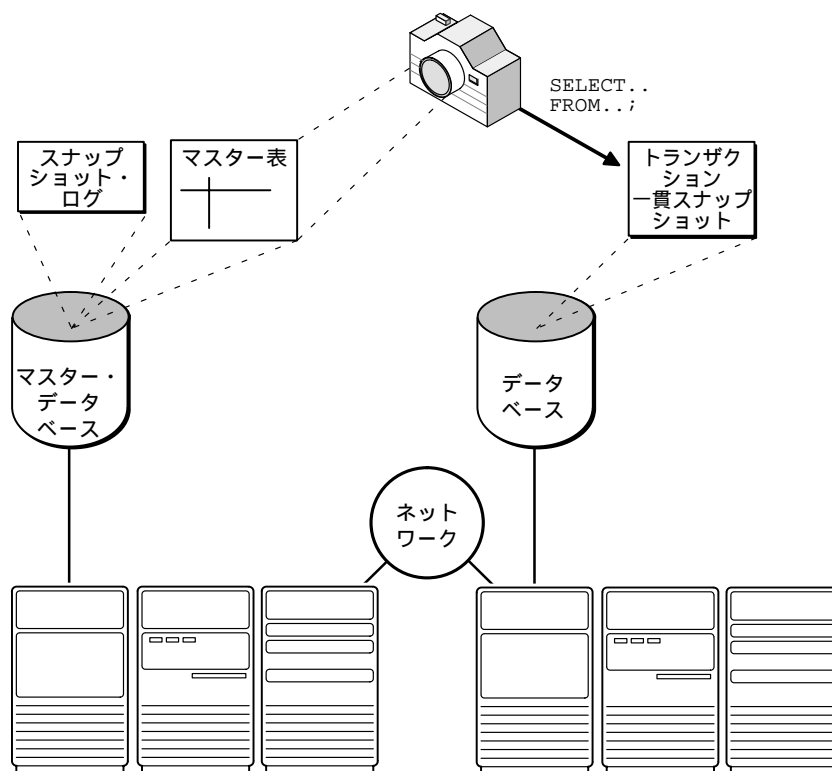
情報の移送

基本レプリケーションは、情報移送手段として便利です。たとえば、基本レプリケーションを使用すると、製品トランザクション処理データベースからデータ・ウェアハウスにデータを定期的に移送できます。

読み専用の表スナップショット

読み専用の表スナップショットは、1つ以上のリモート・マスター表から作成された表データのローカル・コピーです。アプリケーションでは、読み専用の表スナップショットのデータの問合せはできますが、スナップショットの行の挿入または更新、削除はできません。図 1-5 および以降の項で、読み専用の表スナップショットと基本レプリケーションについてさらに詳しく説明します。

図 1-5 基本レプリケーションのコンポーネント：読み専用スナップショットおよびマスター表、スナップショット・ログ



読み専用スナップショットのアーキテクチャ

Oracle では、表スナップショット機能の付いた基本データ・レプリケーションをサポートしています。以降の項では、単純な読み専用表スナップショットのアーキテクチャを説明します。

注意：固有のアプリケーション要件に対して、複合スナップショットおよびROWIDスナップショットなど、基本レプリケーションのその他の機能が用意されています。これらの特別な構成の詳細は、1-10 ページの「基本レプリケーションのその他のオプション」を参照してください。

スナップショットの定義問合せ

表スナップショットの論理的なデータ構造は、1 つ以上のリモート・マスター表のデータを参照する問合せで定義されています。スナップショットの定義問合せにより、スナップショットに含まれるデータが決まります。

スナップショットの定義問合せでは、スナップショットの各行が1つのマスター表の行または行の一部に直接対応しなければなりません。特に、スナップショットの定義問合せには、個別関数や集約関数、またはGROUP BY句やCONNECT BY句、結合、制限型の副問合せ、集合演算を入れることはできません。次のコードは、簡単な表スナップショットの定義例です。

```
CREATE SNAPSHOT sales.customers AS
SELECT * FROM sales.customers@hq.acme.com
```

注意：スナップショットの定義問合せでは、必ずマスター表の主キー列をすべて参照しなければなりません。

スナップショットの定義問合せには、複数の表を参照する制限型の副問合せを組み込んで、スナップショットのマスター表の行をフィルタリングできます。副問合せスナップショットを使用すると、複数のレベルに関連する、子表から親表への複数対1参照を"ウォーク・アップ"するスナップショットを作成できます。次のコードは、簡単な副問合せスナップショットの作成例です。

```
CREATE SNAPSHOT sales.orders AS
SELECT * FROM sales.orders@hq.acme.com o
WHERE EXISTS
( SELECT c_id FROM sales.customers@hq.acme.com c
  WHERE o.c_id = c.c_id AND zip = 19555);
```

内部スナップショット・オブジェクト

新しい読み込み専用の表スナップショットを作成すると、スナップショットをサポートするために、基礎となるデータベース・オブジェクトがいくつか作成されます。

- スナップショットのデータを保存する実表（常時作成）。スナップショットの実表の名前は、SNAP\$_snapshotname です。
- スナップショットの実表の主キー用の索引。索引の名前は、マスター・サイトで対応する PRIMARY KEY 制約を実施するために使用された索引の名前になります。
- 各スナップショットのビュー（常時作成）。ビューの名前はスナップショットの名前になり、スナップショットに読み込み専用でアクセスできます。
- 副問合せで定義されたスナップショットの追加索引（必要な場合）。

スナップショット・リフレッシュ

スナップショットのデータは、そのマスター表の現行データと一致する必要はありません。表スナップショットは、ある時点で存在していたマスター・データとして、トランザクションの一貫性を保つそのデータを反映するものです。スナップショットのデータを比較的新しいマスターデータと一致させるために、スナップショットを定期的にリフレッシュします。スナップショット・リフレッシュは、より最新のマスターの状態をスナップショットに反映させる効果的なバッチ処理です。

より新しいマスター・データを反映するために、各スナップショットをいつ、どのようにリフレッシュするかを決定しなければなりません。たとえば、アプリケーションで更新されることが多いマスター表のスナップショットは、頻繁にリフレッシュする必要があります。これに対して、比較的变化の少ないマスター表のスナップショットは、あまり頻繁にリフレッシュする必要はありません。スナップショットの適切なリフレッシュ間隔を判断するためには、アプリケーションの特性および要件を分析しなければなりません。

スナップショットをリフレッシュするために、異なるタイプのリフレッシュ ("完全" および "高速" リフレッシュ、スナップショット・リフレッシュ・グループ、"手動" および "自動" リフレッシュ) がサポートされています。

完全リフレッシュと高速リフレッシュ

完全リフレッシュまたは高速リフレッシュのどちらを使用しても、スナップショットをリフレッシュできます。

完全リフレッシュ

スナップショットの完全リフレッシュを実行するには、スナップショットを管理するサーバーで、スナップショットの定義問合せを実行します。問合せの結果セットで、既存のスナップショット・データが置き換えられ、スナップショットがリフレッシュされます。完全リフレッシュはあらゆるスナップショットで実行できます。

高速リフレッシュ

高速リフレッシュの実行時は、スナップショットを管理しているサーバーで、スナップショットの最新リフレッシュ後にマスターに加えられた変更が識別され、その変更内容がスナップショットに適用されます。高速リフレッシュは、関連サーバーおよびネットワークでレプリケートするデータが少なくすむため、マスターに加えられた変更が少ないときは完全リフレッシュよりも効率的です。高速リフレッシュは、マスター表にスナップショット・ログがある場合だけ実行できます。

スナップショット・ログ

マスター表が1つ以上のスナップショットに対応するときは、スナップショットの高速リフレッシュを利用できるように表のスナップショット・ログを作成します。マスター表のスナップショット・ログには、対応するすべてのスナップショットの高速リフレッシュ・データが記録されます (1つのマスター表に対して作成できるスナップショット・ログは1つだけです)。サーバーでは、スナップショットの高速リフレッシュを実行するとき、スナップショットを効率的にリフレッシュするために、マスター表のスナップショット・ログのデータを使います。すべてのスナップショットでリフレッシュが実行され、ログ・データが必要なくなると、スナップショット・ログから特定のリフレッシュ・データが自動的にパージされます。

マスター表にスナップショット・ログを作成すると、スナップショット・ログをサポートする基礎表が作成されます。スナップショット・ログの表には、主キーおよびタイムスタンプ、アプリケーションで更新されるマスター表の行のROWID (オプション) が含まれます。スナップショット・ログには、副問合せを使用するスナップショットの高速リフレッシュをサポートするために、フィルタ列を組み込むこともできます。スナップショット・ログの表の名前は、MLOG\$_master_table_name です。

スナップショット・リフレッシュ・グループ

複数の関連するマスター表のスナップショット間の参照整合性およびトランザクションの一貫性を保つために、各スナップショットは、リフレッシュ・グループの一部として編成およびリフレッシュされます。1回の処理でグループのすべてのスナップショットがリフレッシュされます。リフレッシュ・グループ内のすべてのスナップショットがリフレッシュされると、グループ内のすべてのスナップショット・データが、トランザクションの一貫性を保つある同じ時点のデータに対応します。

自動スナップショット・リフレッシュ

スナップショットのリフレッシュ・グループ作成時に、管理者は通常スナップショットが自動的にリフレッシュされるようにグループを構成します。このように構成しない場合は、管理者は必要なときにグループを手動でリフレッシュしなければなりません。

自動リフレッシュ用のリフレッシュ・グループを構成するときは、次のことを行わなければなりません。

- グループのリフレッシュ間隔を指定する。
- リフレッシュする必要のあるスナップショットを定期的によりリフレッシュする、1つ以上のSNPバックグラウンド・プロセスでスナップショットを管理するサーバーを構成する。

自動リフレッシュ間隔

スナップショットのリフレッシュ・グループを作成するときに、グループの自動リフレッシュ間隔を指定できます。グループのリフレッシュ間隔を設定するときには、次の特性を考慮してください。

- リフレッシュ間隔を指定する日付または日付式の値は、将来のある時点でなければならない。
- リフレッシュ間隔は、リフレッシュの実行に要する時間よりも長くなければならない。
- 相対日付式の値は、最新のリフレッシュ日付から計算されるある時点でなければならない。ネットワークまたはシステムの障害により、グループのリフレッシュがスケジュール通りに実行されない場合、相対日付式の結果は状況に応じて変更されることがあります。
- 明示日付式の値は、最新のリフレッシュ日付に関係なく特定の時点でなければならない。

リフレッシュ・タイプ

デフォルトでは、リフレッシュ・グループの各スナップショットに対して、高速リフレッシュの実行が試行されます。個々のスナップショットに対して高速リフレッシュが実行できない場合は（たとえば、マスター表にスナップショット・ログがない）、サーバーでスナップショットの完全リフレッシュが実行されます。

SNP バックグラウンド・プロセス

Oracle Serverの自動スナップショット・リフレッシュ機能は、内部システム・プロシージャを周期的に実行するようにジョブ・キューに設定されて稼働します。ジョブ・キューを使用するには、少なくとも1つのSNPバックグラウンド・プロセスが実行中でなければなりません。SNP バックグラウンド・プロセスはジョブ・キューを周期的にチェックし、未処理のジョブを実行します。

手動スナップショット・リフレッシュ

スケジュールされた自動スナップショット・リフレッシュが適当ではない場合があります。たとえば、大量のデータをマスター表にロードした直後は、依存するスナップショットにマスター表のデータは反映されていません。次回実行予定の自動グループ・リフレッシュを待たずに、対応付けられたスナップショットにマスター表の新規行をすぐに反映するために、依存するスナップショット・グループを手動でリフレッシュすることもできます。

基本レプリケーションのその他のオプション

他にも、特定の状況で役立つ基本レプリケーションの機能がサポートされています。

- 複合スナップショット
- ROWID スナップショット

複合スナップショット

スナップショットの定義問合せに、個別関数や集約関数、またはGROUP BY 句やCONNECT BY句、結合、制限型の副問合せ、集合演算などが含まれるとき、スナップショットは複合スナップショットと呼ばれます。次のコードは、複合表スナップショットの定義例です。

```
CREATE SNAPSHOT scott.emp AS
SELECT ename, dname
FROM scott.emp@hq.acme.com a, scott.dept@hq.acme.com b
WHERE a.deptno = b.deptno
SORT BY dname
```

複合スナップショットの主な欠点は、スナップショットの高速リフレッシュを実行できないことです。複合スナップショットでは完全リフレッシュしか実行できません。このため、複合スナップショットを使用すると、完全スナップショット・リフレッシュの実行中、ネットワークのパフォーマンスに影響することがあります。

ROWID スナップショット

この章の前の項でも説明したとおり、Oracle では、主キー・スナップショットがデフォルトです。主キー・スナップショットは、マスター表の主キーに基づいて作成されます。この構造を使用すると、次のことができます。

- スナップショットの完全リフレッシュを実行しなくても、スナップショットのマスター表を再編成できる。
- 制限型の副問合せを含む定義問合せを使用したスナップショットを作成する。

下位互換性のためだけに必要な、マスター表の行の物理的な行識別子 (ROWID) に基づいた ROWID スナップショットもサポートされています。ROWID スナップショットは、Oracle7.3 データベースのマスター表のスナップショットにだけ使用し、Oracle8 データベースのマスター表の新規スナップショットを作成するときは使用しないでください。

注意: ROWID スナップショットをサポートするために、L_SNAP\$_snapshotname という名前のスナップショットの実表に、追加索引が作成されます。

アドバンスト・レプリケーションの概念

アドバンスト・レプリケーション環境では、システムのどこに位置するデータ・レプリカを使用しても、表のデータの読み込みおよび更新アクセスができます。以降の項では、アドバンスト・レプリケーション・システムの基本的な概念を説明します。

- アドバンスト・レプリケーションの使用方法
- アドバンスト・レプリケーションの構成
- レプリケーション・オブジェクト、グループ、サイト、カタログ
- Oracle のアドバンスト・レプリケーション・アーキテクチャ
- レプリケーション管理者および伝播担当者、受信者
- レプリケーションの競合
- アドバンスト・レプリケーション固有のオプション

アドバンスト・レプリケーションの使用方法

アドバンスト・データ・レプリケーションは、特別な要件があるさまざまなタイプのアプリケーション・システムに有効です。

非接続環境

アドバンスト・レプリケーションは、非接続コンポーネントを使用して動作するトランザクション処理アプリケーションを配置する際に便利です。たとえば、生命保険会社の一般的な販売部門自動化システムを考えてみます。各販売員は、ラップトップ・コンピュータを携帯して顧客を定期的に訪問し、会社のコンピュータ・ネットワークおよび中央制御データベース・システムとは切断された状態で、個人データベースに取引内容を記録しなければなりません。会社に戻ってから、各販売員は取引内容をすべて中央制御の会社のデータベースに転送する必要があります。

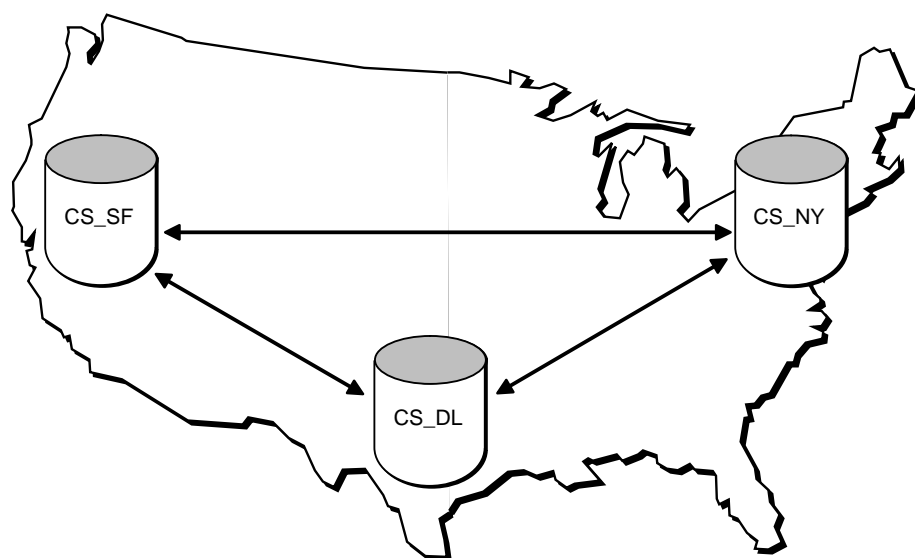
フェイルオーバー・サイト

アドバンスト・レプリケーションは、ミッション・クリティカルなデータベースの可用性を保護するために役立ちます。たとえば、アドバンスト・レプリケーション・システムでは、システムまたはネットワーク障害が発生してプライマリ・サイトが利用できなくなった場合に備えて、フェイルオーバー・サイトを確立するためにデータベース全体をレプリケートできます。スタンバイ・データベース機能とは異なり、このようなフェイルオーバー・サイトは、プライマリ・サイトが同時に稼動しているときは、アプリケーション・アクセスをサポートする完全な機能をもつデータベースの役割も果たします。

アプリケーション負荷の分配

アドバンスト・レプリケーションは、過度のアプリケーション負荷の分配、または継続的な可用性の保証、よりローカライズされたデータ・アクセスの提供を目的として、複数ポイントのデータベース情報にアクセスする必要があるトランザクション処理アプリケーションに便利です。このような要件があるアプリケーションには、通常サービス優先のアプリケーションが含まれます。

図 1-6 複数ポイントの更新アクセスをサポートするアドバンスト・レプリケーション・システム



情報の移送

アドバンスト・レプリケーションは、情報の移送手段として便利です。たとえば、アドバンスト・レプリケーション・システムでは、更新操作中のデータベースのデータを、データ・ウェアハウスまたはデータ・マートに定期的にオフロードできます。

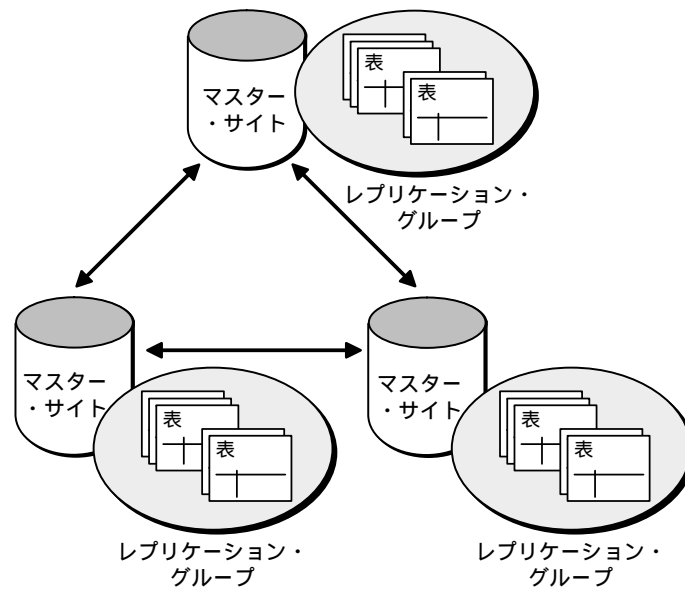
アドバンスト・レプリケーションの構成

スナップショット・サイトと同様に、マルチマスター・レプリケーションを使用するアドバンスト・レプリケーション環境の要件もサポートされています。

マルチマスター・レプリケーション

マルチマスター・レプリケーションを使用すると、同等ピアとして機能する複数サイトで、レプリケート・データベース・オブジェクトのグループを管理できます。アプリケーションでは、マルチマスター構成のあらゆるサイトで任意のレプリケート表を更新できます。図 1-7 は、マルチマスター・アドバンスト・レプリケーション・システムの例です。

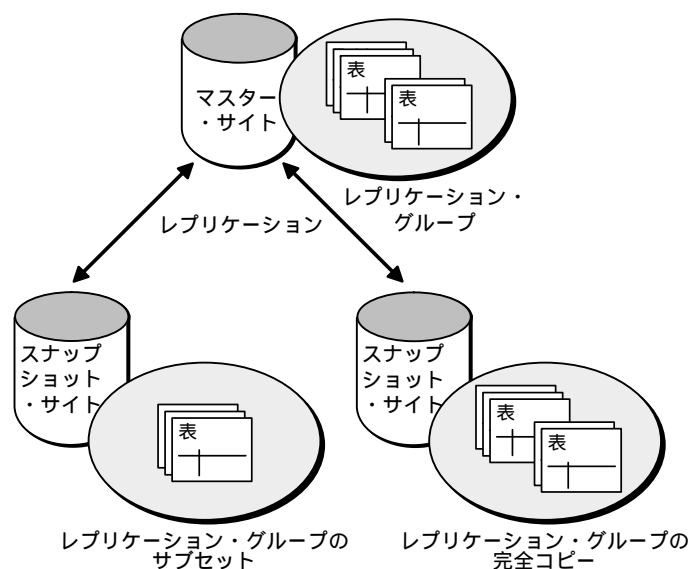
図 1-7 マルチマスター・レプリケーション・システム



スナップショット・サイトおよび更新可能スナップショット

リモートのスナップショット・サイトでアプリケーションが更新する情報を、アドバンスド・レプリケーション・システムのマスター・サイトで整理統合できます。アドバンスド・レプリケーション機能を使用すると、アプリケーションは更新可能スナップショットを使用して表の行の挿入および更新、削除ができます。図 1-8 は、更新可能なスナップショットをもつアドバンスド・レプリケーション環境の例です。

図 1-8 更新可能スナップショットを使用するアドバンスド・レプリケーション・システム



更新可能スナップショットには次の特性があります。

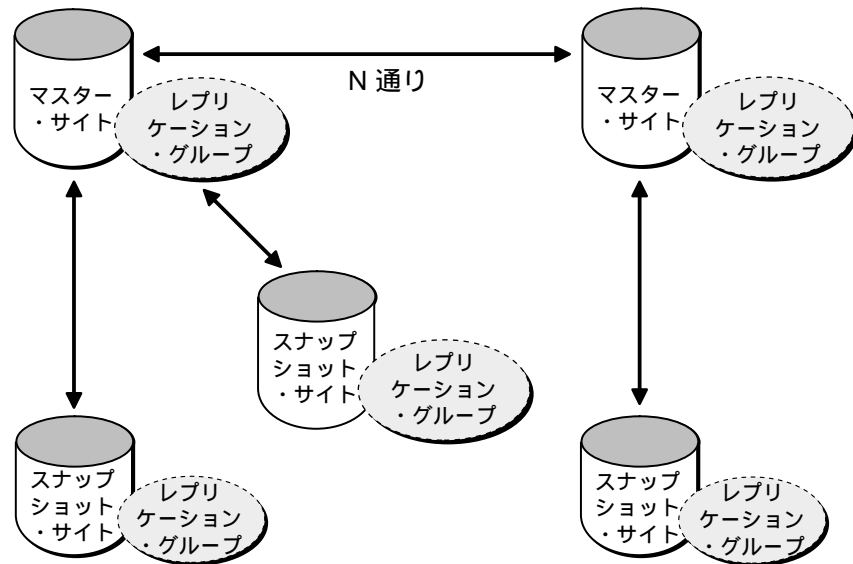
- 常に単純な、高速リフレッシュが可能な表スナップショットである。
- 更新可能スナップショットを通して加えられた変更は、スナップショットのリモートのマスター表に伝播される。必要ならば、続けてその他のすべてのマスター・サイトに更新内容が順に伝播されます。
- 読み専用スナップショットと同様に、リフレッシュ・グループの一部としてリフレッシュされる。
- 読み専用スナップショットと同じ基本オブジェクト（実表および索引、ビュー）がある。他に、サポート用に `USLOG$_snapshotname` という表が作成されます。

ハイブリッド構成

さまざまなアプリケーション要件に合わせて、マルチマスター・レプリケーションおよび更新可能スナップショットをハイブリッド（混在）構成で結合できます。混在構成では、任意の数のマスター・サイトと、各マスター用の複数のスナップショット・サイトを作成できます。

たとえば、図 1-9 に示すように、2つのマスター間の n 通りのレプリケーションで、2つの地域をサポートするデータベース間の全表レプリケーションをサポートできます。スナップショットをマスター上に定義すると、全表または表のサブセットを各地域内のサイトにレプリケートできます。

図 1-9 ハイブリッド構成



更新可能スナップショットとレプリケート・マスターとの主な相違点は次のとおりです。

- レプリケート・マスターには、レプリケートしている表全体のデータが含まなければならないが、スナップショットでは、マスター表のデータのサブセットをレプリケートできる。
- マルチマスター・レプリケーションを使用すると、変更が起きるたびに各トランザクションの変更内容がレプリケートされる。スナップショット・リフレッシュを使用すると、ある決まった間隔で処理されるので、マルチマスター・レプリケーションより少ない頻度で複数のトランザクションの内容をより効率的に伝播できます。
- 同じデータの複数のコピーに変更が行われた結果なんらかの競合が起こった場合、競合はマスター・サイトで検出され、解消される。

アドバンスド・レプリケーションおよび Oracle Replication Manager

アドバンスド・レプリケーション環境では、任意の場所で更新可能なデータ・モデルをサポートする構成および管理が可能です。アドバンスド・レプリケーション環境の管理に利用できるように、高度な管理ツール、Oracle Replication Manager が用意されています。Replication Manager の使用方法についての情報および例は、このマニュアルの他の項で説明しています。

レプリケーション・オブジェクト、グループ、サイト、カタログ

以降の項では、レプリケーション・オブジェクトおよびグループ、サイト、カタログなど、アドバンスト・レプリケーション・システムの基本コンポーネントについて説明します。

レプリケーション・オブジェクト

レプリケーション・オブジェクトは、分散データベース・システム内の複数のサーバーに存在するデータベース・オブジェクトです。アドバンスト・レプリケーション機能を使用すると、表とそのサポート・オブジェクト（ビューおよびデータベース・トリガー、パッケージ、索引、シノニムなど）をレプリケートできます。

レプリケーション・グループ

アドバンスト・レプリケーション環境では、レプリケーション・グループを使用してレプリケーション・オブジェクトが管理されます。レプリケーション・グループ内に関連するデータベース・オブジェクトを編成すると、多数のオブジェクトをまとめて簡単に管理できます。通常、レプリケーション・グループを作成および使用して、特定のデータベース・アプリケーションのサポートに必要なスキーマ・オブジェクトを編成します。ただし、レプリケーション・グループとスキーマが相互に対応する必要はありません。レプリケーション・グループのオブジェクトは、複数のデータベース・スキーマから作成でき、スキーマには異なるレプリケーション・グループのメンバーであるオブジェクトを入れることができます。制限は、レプリケーション・オブジェクトが1つのグループのメンバーにしか入れないことです。

レプリケーション・サイト

レプリケーション・グループには、複数のレプリケーション・サイトが存在することがあります。アドバンスト・レプリケーション環境では、マスター・サイトおよびスナップショット・サイトという、2つの基本的なタイプのサイトがサポートされています。

- マスター・サイトでは、レプリケーション・グループのすべてのオブジェクトの完全なコピーが管理されます。マルチマスターのアドバンスト・レプリケーション環境のすべてのマスター・サイトで、レプリケーション・グループのデータおよびスキーマ変更を伝播するために、別のマスター・サイトと直接通信を行います。マスター・サイトのレプリケーション・グループは、特にマスター・グループと呼ばれます。
- また、各レプリケーション・グループには、マスター定義サイトが1つだけあります。レプリケーション・グループのマスター定義サイトは、レプリケーション・グループおよびグループのオブジェクトを管理する制御ポイントとして機能するマスター・サイトです。
- スナップショット・サイトでは、対応付けられたマスター・サイトの表データの単純な読み込み専用スナップショットおよび更新可能スナップショットがサポートされます。スナップショット・サイトの表スナップショットには、レプリケーション・グループ内の表データ全体またはそのサブセットを含めることができます。ただし、どちらの場合も、そのスナップショットはマスター・サイトの表に1対1で対応する単純スナップショットでなければなりません。たとえば、スナップショット・サイトには、レプリケーション・グループの選択した表だけのスナップショットを作成できます。特定のスナップショットに、あるレプリケート表の選択した部分だけを入れることもできます。スナップ

ショット・サイトのレプリケーション・グループは、特にスナップショット・グループと呼ばれます。スナップショット・グループにはその他のレプリケーション・オブジェクトを入れることもできます。

レプリケーション・カタログ

アドバンスト・レプリケーション環境の各マスター・サイトおよびスナップショット・サイトには、レプリケーション・カタログがあります。サイトのレプリケーション・カタログは、データ・ディクショナリ表の固有のセット、およびサイトのレプリケーション・オブジェクトとレプリケーション・グループに関する管理情報を管理するビューです。アドバンスト・レプリケーション環境に関係する各サーバーでは、そのレプリケーション・カタログの情報を使用して、レプリケーション・グループのオブジェクトのレプリケーションを自動化できます。

レプリケーション管理 API および管理要求

アドバンスト・レプリケーション環境を構成および管理するために、関係する各サーバーで、Oracle のレプリケーション・アプリケーション・プログラミング・インタフェース (API) が使用されます。サーバーのレプリケーション管理 API は、アドバンスト・レプリケーション機能の構成に管理者が利用できるプロシージャおよびファンクションがカプセル化された PL/SQL パッケージのセットです。また、Oracle Replication Manager では、作業を実行するために、各サイトのレプリケーション管理 API のプロシージャとファンクションを使用します。

管理要求とは、Oracle のレプリケーション管理 API のプロシージャまたはファンクションをコールすることです。たとえば、Replication Manager を使用して新しいマスター・グループを作成するときに、Replication Manager は、DBMS_REPCAT.CREATE_MASTER_REPGROUP プロシージャをコールして作業を完了させます。管理要求の中には、他のレプリケーション管理 API を生成して要求を処理するものもあります。

Oracle のアドバンスト・レプリケーション・アーキテクチャ

非同期データ伝播の行レベル・レプリケーションを使用して、一般的なアドバンスト・レプリケーション構成のデータが収集されます。以降の項では、これらのメカニズムの機能について説明します。

注意：アプリケーション固有の要件のために、プロシージャ・レプリケーションおよび同期データ伝播など、その他のアドバンスト・レプリケーション機能が用意されています。これらの特別な構成の詳細は、1-25 ページの「アドバンスト・レプリケーション固有のオプション」を参照してください。

行レベル・レプリケーション

一般的なトランザクション処理アプリケーションでは、1 つのトランザクションで変更されるのは数行です。アドバンスト・レプリケーション環境で動作するこのようなアプリケーションは、通常行レベル・レプリケーション・メカニズムに依存します。行レベル・レプリケーションの場合、アプリケーションは標準の DML 文を使用して、ローカルのデータ・レプリカのデータを変更します。トランザクションによりローカルのデータが変更されると、サーバーは変更に関する情報を自動的に取得し、リモート・サイトにローカルの変更内容を転送するために、対応する遅延トランザクションをキューに入れます。

生成レプリケーション・オブジェクト

アドバンスト・レプリケーション環境でトランザクションのレプリケーションをサポートするには、各レプリケート表またはパッケージ、プロシージャをサポートする 1 つ以上の内部システム・オブジェクトを生成する必要があります。

- 表をレプリケートするとき、対応するパッケージが 2 つ生成されます。レプリケート表の `tablename$RP` パッケージは、表を含むトランザクションをレプリケートするために使用されます。レプリケート表の `tablename$RR` パッケージは、表を含むレプリケーションの競合を解消するために使用されます。
- プロシージャ・レプリケーションをサポートするためにパッケージ仕様部およびパッケージ本体をレプリケートするときに、対応する Wrapper パッケージ仕様部およびパッケージ本体を生成できます。デフォルトでは、パッケージ仕様部およびパッケージ本体の Wrapper の名前は、接頭辞が "defer_" のオブジェクトの名前を使用して設定されます。

注意：Oracle の以前のバージョンでは、サーバーで、レプリケート表をサポートする PL/SQL トリガーも生成されます。しかし、表のレプリケートを選択すると、内部トリガーが "アクティブ" になります。

非同期（蓄積交換）データ伝播

行レベル・レプリケーションに依存する一般的なアドバンスト・レプリケーション構成では、非同期データ・レプリケーションを使用してデータ・レベルの変更が伝播されます。非同期データ・レプリケーションが発生するのは、アプリケーションで表のローカル・レプリカを更新し、レプリケーション情報をローカル・キューに保存し、後でそのレプリケーション情報を他のレプリケーション・サイトに転送するときです。このため、非同期データ・レプリケーションは蓄積交換データ・レプリケーションとも呼ばれます。

図 1-10 非同期データレプリケーションのメカニズム

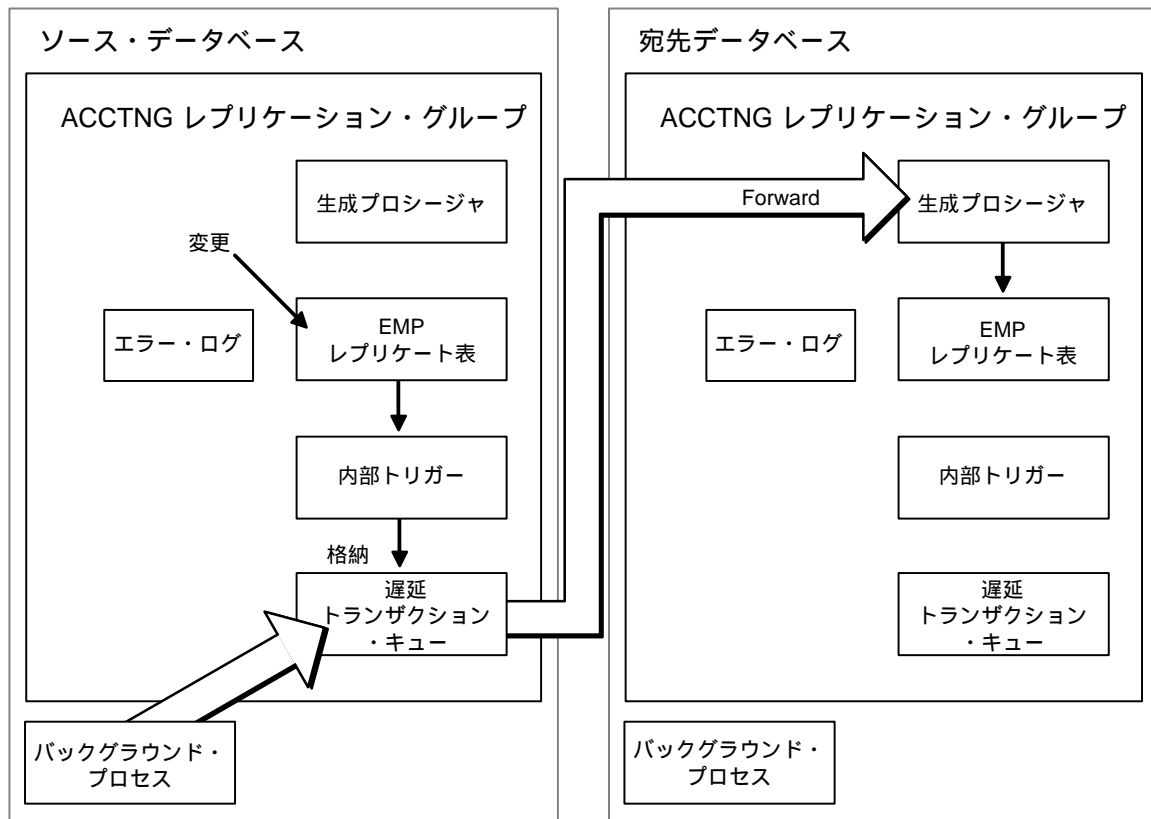


図 1-10 に示すように、トリガー内部システムおよび遅延トランザクション、遅延トランザクション・キュー、ジョブ・キューを使用して、更新可能スナップショットからマスター表にだけでなく、アドバンスド・レプリケーション・システムのマスター・サイト間でもデータ・レベルの変更が非同期で伝播されます。

- アプリケーションがアドバンスド・レプリケーション環境で動作するとき、レプリケート・データの更新内容に関する情報の取得および保存に、内部トリガーが使用されます。内部トリガーにより、ローカル・サイトで加えられたデータ変更をリモートのレプリケーション・サイトに再生成するリモート・プロシージャ・コール (RPC) が作成されます。データ・レプリケーションをサポートする内部トリガーは、Oracle Server 実行モジュール内の必須コンポーネントです。これを利用すると、システム・リソースの使用を最小限に抑え、更新内容の取得およびレプリケート・データの保存を非常にすばやく実行できます。

- 遅延伝播用に、サイトの遅延トランザクション・キューの内部トリガーで生成される RPC が保存されます。また、トランザクションを構成するすべての RPC が、同様にトランザクションとしてリモートで伝播および適用されるように、開始トランザクションに関する情報も記録されます。アドバンスト・レプリケーション機能では、アドバンスト・キュー・メカニズムを使用して遅延トランザクション・キューがインプリメントされます。
- 伝播プロセスは、ジョブ・キュー・メカニズムおよび遅延トランザクションを使用して管理されます。アドバンスト・レプリケーション・システムに関連する各サーバーには、ローカル・ジョブ・キューがあります。サーバーのジョブ・キューは、ジョブの実行を要求する PL/SQL コール、ジョブの実行時期など、ローカル・ジョブに関する情報を保存するデータベース表です。アドバンスト・レプリケーション環境の一般的なジョブには、遅延トランザクションをリモートのマスター・サイトに送信するジョブ、および遅延トランザクション・キューから適用済のトランザクションをバージするジョブ、スナップショットのリフレッシュ・グループをリフレッシュするジョブがあります。
- 遅延トランザクションの一部として RPC を実行することにより、データ・レプリケーション情報が転送されます。分散トランザクション・プロトコルを使用して、グローバルなデータベース整合性が自動的に保護され、データ保管が保証されます。

シリアル伝播

シリアル伝播の場合、レプリケート・トランザクションは、元のサイトでコミットされた順に一度に 1 つずつ非同期で伝播されます。

パラレル伝播

パラレル伝播では、複数のパラレル伝送ストリームを使用してレプリケート・トランザクションが非同期で伝播されるため、スループットがより高くなります。必要に応じて、グローバルなデータベースの整合性を保証するために依存トランザクションの実行と順序づけします。

パラレル伝播では、パラレル問合せおよびロード、回復、その他のパラレル操作に使用されるメカニズムと同じ実行メカニズムが使用されます。各サーバー・プロセスでは、シングル・ストリームを介してトランザクションが伝播されます。これらのサーバー・プロセスは、パラレル・コーディネータ・プロセスにより制御されます。コーディネータでは、トランザクションの依存関係を追跡してサーバー・プロセスに作業を割り当て、その進行状況を追跡します。

遅延トランザクション・キューのバージ

サイトから接続先に遅延トランザクションを送信した後、適用済のトランザクションは、別のジョブでキューからバージされるまで、遅延トランザクション・キューに残ります。

スナップショット伝播メカニズム

アドバンスト・レプリケーション環境の更新可能スナップショットでは、データをマスター表に "送信" したり、マスター表から "取り出す" ことができます。

マスター表の更新

更新可能スナップショットに対する更新内容は、行レベルの非同期データ伝播メカニズム（RPC および遅延トランザクション、ジョブ・キュー）を使用して、マスター表に非同期で送信されます。

スナップショット・リフレッシュ

基本レプリケーション環境と同様に、アドバンスト・レプリケーション・システムでは、スナップショット・リフレッシュ・メカニズムを使用して、マスター表から変更内容を非同期で取り出し、対応付けられた更新可能（および読み専用）スナップショットに送信します。

その他の考慮事項

更新可能スナップショットの送信および取出しのタスクは、まとめてまたは個々に構成できる独立した処理です。

- スナップショット・サイトでは、リフレッシュ・グループを構成すると、メンバー・スナップショットに加えられた変更をマスター・サイトに自動的に送信して、スナップショットをリフレッシュできます。
- スナップショット・サイトでは、更新可能スナップショットを構成すると、変更内容のマスター・サイトへの送信およびスナップショットのリフレッシュを、異なる時間および間隔で実行できます。

たとえば、マスター・サイトで情報を整理統合するアドバンスト・レプリケーション環境では、更新可能スナップショットを構成して、変更内容をマスター・サイトに1時間ごとに送信し、更新可能スナップショットのリフレッシュはあまり実行しないようにできます。

レプリケーション管理者および伝播担当者、受信者

アドバンスト・レプリケーション環境では、正常に機能させるためには、レプリケーション管理者および伝播担当者、受信者など、一意のデータベース・ユーザー・アカウントがいくつか必要です。

- アドバンスト・レプリケーション・システムの各サイトで、レプリケーション管理者（レプリケート・データベース・オブジェクトを構成および管理するユーザー）が最低1人は必要です。
- アドバンスト・レプリケーション・システムの各レプリケーション・サイトで、レプリケート・データの変更内容を伝播および適用する特別なユーザー・アカウントが必要です。

構成オプション

ほとんどのアドバンスト・レプリケーション構成では、1つのアカウントがすべての目的（レプリケーション管理者および伝播担当者、受信者）に使用されます。しかし、一意の構成用に固有のアカウントもサポートされています。

レプリケーションの競合

任意の場所で更新可能なデータ・レプリカのモデルをサポートするアドバンスト・レプリケーション・システムでは、レプリケーションが競合することがあるので注意が必要です。以降の項では、異なる種類のレプリケーションの競合、およびその発生時期、検出方法と解消方法を説明します。

レプリケーションの競合の種類

アドバンスト・レプリケーション環境では、一意性の競合および更新の競合、削除の競合の3種類の競合が発生する可能性があります。

一意性の競合

一意性の競合は、行のレプリケーションがエンティティの整合性（PRIMARY KEY 制約または UNIQUE 制約）に違反する処理を実行しようとしたときに発生します。たとえば、2つの異なるサイトから発行された2つのトランザクションで、同じ主キー値を使用してそれぞれの表レプリカに行を挿入すると何が起きるかを考えてみます。この場合、トランザクションのレプリケーションが原因で一意性の競合が発生します。

更新の競合

更新の競合は、ある行に対する更新のレプリケーションが同じ行に対する別の更新と競合したときに発生します。異なるサイトから発行された2つの異なるトランザクションにより、同じ行がほとんど同時に更新されたときに発生します。

削除の競合

削除の競合は、異なるサイトから2つのトランザクションが発行され、一方のトランザクションが更新または削除する行を、もう一方のトランザクションが削除する場合に発生します。

レプリケート・データ・モデルと競合

アドバンスト・レプリケーションを使用するデータベース・システムの最上位層で動作するアプリケーションを設計するときは、レプリケーションの競合の可能性を考慮しなければなりません。このような場合、アプリケーションでいくつかの異なるレプリケート・データ所有権モデルの1つを使用して、レプリケーションの競合を防止または解消することによりグローバルなデータベース整合性が保証されるようにしなければなりません。

プライマリ・サイト、静的所有権

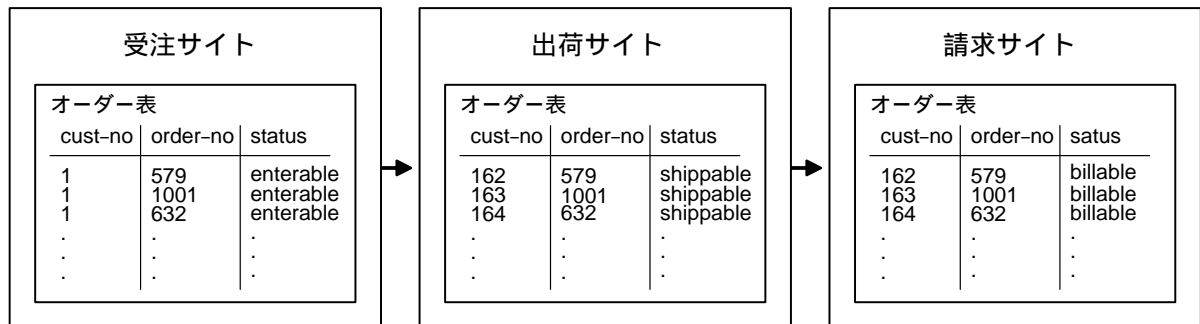
プライマリ所有権（静的所有権ともいいます）は、基本的な読み専用レプリケーション環境でサポートされるレプリケート・データ・モデルです。プライマリ所有権により、一連のレプリケート・データに更新アクセスできるサーバーは1つに限定されるため、すべての種類のレプリケーションの競合が発生しなくなります。

表レベルでデータの所有権を制御するかわりに、アプリケーションで水平パーティション化および垂直パーティション化を使用すると、データの静的所有権をより細かく設定できます。たとえば、サイト別の基準でレプリケート表の特定の列または行に更新アクセスすることもできます。

動的所有権

動的所有権レプリケート・データ・モデルは、プライマリ・サイト所有権よりも制限の少ないモデルです。動的所有権の場合、データ・レプリカを更新する権利はサイトからサイトへ移動しますが、ある時点で特定のデータに更新アクセスできるサイトは1つであることが保証されます。ワークフロー・システムに動的所有権の概念がわかりやすく示されています。たとえば、関連する部門のアプリケーションで、注文を更新できる時期を判断するために、製品注文のステータス・コードを読み込むことができます。図 1-11 は、動的所有権モデルを使用したアプリケーションの例です。

図 1-11 オーダー処理システムの動的所有権



共有所有権

競合の発生を抑えるプライマリ・サイト所有権および動的所有権レプリケーション・データ・モデルは、制限が多いため一部のデータベース・アプリケーションでは実施できないことがあります。アプリケーションによっては、いつでも表レプリカのデータを更新できる共有所有権レプリケート・データ・モデルを使用して処理する必要があります。

共有データ所有権システムで変更内容が非同期でレプリケートされる（蓄積交換レプリケーション）とき、対応するアプリケーションでは、レプリケーションの競合を確実に回避するか、または発生した場合は検出して解消しなければなりません。

競合の解消方法

通常は、レプリケーションの競合（特に、一意性の競合と削除の競合）をすべてまたはほとんど防止できるようにアドバンスド・レプリケーション・システムを設計することができます。

競合の検出

競合は防止するべきですが、避けられない場合もあります。アプリケーションで非同期の行レベル・レプリケーションを使用する共有所有権データ・モデルを使用していて、レプリケーションの競合が発生する可能性がある場合、一意性および更新、削除の競合が自動的に検出されます。レプリケーション中の競合を検出するために、発行サイトの最小単位の行データと、受信サイトの対応する行の情報が比較されます。相違点があると競合が検出されます。

レプリケーションの競合を正確に検出するには、データ・レプリケーション時に別々のサイトで対応する行を一意に識別して、一致させることができればなりません。通常、アドバンスト・レプリケーション機能では、表の主キーを使用して表の行が一意に識別されます。表に主キーがない場合は、代替キーを指定しなければなりません。代替キーとは、データ・レプリケーション中に表の行を識別するために使用する列または列のセットのことです。どちらの場合も、アプリケーションで表の識別列を更新できないようにする必要があります。そうすることにより、行を識別してレプリケート・データの整合性を保つことができます。

競合の解消

アドバンスト・レプリケーション・システムの受信サイトで非同期の行レベル・レプリケーションを使用していて、トランザクションで競合が検出された場合、デフォルトの動作では、その競合とトランザクション全体のログが作成され、ローカルデータはそのままの状態に残されます。ほとんどの場合、アドバンスト・レプリケーション機能を使用して、レプリケーションの競合の解消を自動化することをお勧めします。

列グループ

非同期の行レベル・アドバンスト・レプリケーション中の更新の競合を検出および解消するために、列グループが使用されます。列グループは、表の1つ以上の列を論理的にグループ化したものです。レプリケート表の各列は、1つの列グループの一部になります。レプリケート表を構成するときに、列グループを作成して、列および対応する更新の競合解消方法を各グループに割り当てることができます。

レプリケート表の各列グループには、1つ以上の更新の競合解消方法のリストがある場合があります。グループの競合解消方法を複数指定しておく、他の方法で競合の解消に失敗したときに、異なる方法で更新の競合を解消できます。グループの更新の競合を解消しようとすると、そのグループ用にリストした順でグループの競合解消方法が実行されます。

デフォルトでは、各レプリケート表にシャドウ列グループがあります。表のシャドウ列グループには、特定の列グループ内にはない列がすべて含まれます。表のシャドウ・グループには、競合解消方法を割り当ててはできません。

競合解消方法

列グループを設計するときに、多数のビルトインの競合解消方法から選択できます。たとえば、更新の競合を解消するには、接続先サイトの列値を発行サイトの列値で上書きする方法を選択できます。更新の競合解消方法は、その他に多数用意されています。

また、PRIMARY KEY 制約および UNIQUE 制約に一意性の競合解消方法を割り当てることもできます。しかし、削除の競合解消方法は用意されていません。このため、非同期の共有所有権データ・モデル内で動作するアプリケーションは、行の削除に DELETE 文を使用しないようにして削除の競合を防止しなければなりません。または、アプリケーションで行に削除のマークを設定して、削除した行をプロシージャ・レプリケーションを使用して定期的に削除するようにシステムを構成することもできます。

アドバンスト・レプリケーション固有のオプション

一部のアプリケーションには、アドバンスト・レプリケーション・システムに特別な要件があります。以降の項では、次のアドバンスト・レプリケーション固有のオプションについて説明します。

- プロシージャ・レプリケーション
- 同期（即時）データ伝播

プロシージャ・レプリケーション

バッチ処理アプリケーションでは、1つのトランザクション内で大量のデータを変更できます。このような場合、一般的な行レベル・レプリケーションでは、大量のデータ変更のためにネットワークが飽和状態になることがあります。このような問題を防止するために、アドバンスト・レプリケーション環境で動作するバッチ処理アプリケーションでは、プロシージャ・レプリケーションを使用して、データ・レプリカを収集する単純なストアド・プロシージャ・コールをレプリケートできます。プロシージャ・レプリケーションでは、アプリケーションが表の更新に使用するストアド・プロシージャのコールだけがレプリケートされます。プロシージャ・レプリケーションでは、データの変更内容はレプリケートされません。

プロシージャ・レプリケーションを使用するには、すべてのサイトで、システムのデータを変更するパッケージをレプリケートする必要があります。パッケージをレプリケートした後に、各サイトでこのパッケージ用の Wrapper を生成しなければなりません。アプリケーションがデータを変更するパッケージ・プロシージャをローカル・サイトでコールすると、Wrapper は最終的にレプリケート環境内の他のすべてのサイトで同じパッケージ・プロシージャをコールします。プロシージャ・レプリケーションは、非同期にまたは同期をとって発生する場合があります。

競合の検出とプロシージャ・レプリケーション

アドバンスト・レプリケーション・システムでプロシージャ・レプリケーションを使用してデータをレプリケートするとき、データをレプリケートするプロシージャは、レプリケート・データの整合性を保証しなければなりません。つまり、このようなプロシージャは、レプリケーションの競合を防止する、または検出して適切に解消するように設計しなければなりません。このため、プロシージャ・レプリケーションは、大規模なバッチ操作の処理にだけデータベースが使用可能なときに主に使用されます。このような状況では、多数のトランザクションが同じデータで競合することはないため、レプリケーションの競合はほとんど発生しません。

追加情報: 7-2ページの「プロシージャ・レプリケーションの使用方法」を参照してください。

同期（即時）データ伝播

非同期データ伝播は、アドバンスド・レプリケーション環境の標準構成です。ただし、特別な要件があるアプリケーションに対しては同期データ伝播もサポートされています。同期データ伝播は、アプリケーションで表のローカル・レプリカを更新して、同じトランザクション内で同じ表のその他のレプリカもすべて更新するときに発生します。このため、同期データ・レプリケーションは即時データ・レプリケーションとも呼ばれます。同期レプリケーションの使用は、アプリケーションでレプリケート・サイトを継続的に同期化する必要があるときだけにしてください。

注意：レプリケーション・データの即時伝播を使用するレプリケーション・システムは、システムのすべてのサイトが同時に使用可能な場合にしか機能しないため、システムおよびネットワークの可用性に大きく依存します。

図 1-12 同期データ・レプリケーションのメカニズム

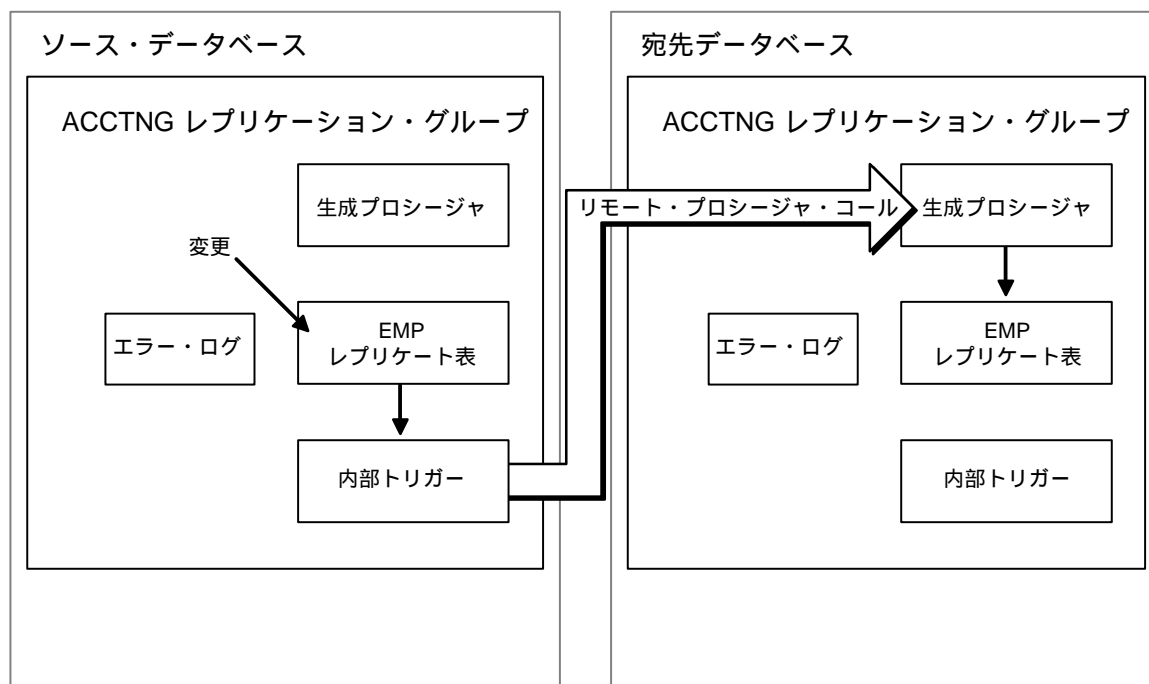


図 1-12 に示すように、同期式の行レベル・データ・レプリケーションをサポートするために、同じシステムの内部データベース・トリガーを使用して、データ・レベルの変更内容を他のレプリケーション・サイトにレプリケートする RPC が生成されます。ただし、このような RPC の実行は延期されません。かわりに、データ・レプリケーション RPC は、ローカル・レプリカを変更する同一のトランザクションの境界内で実行されます。このため、レプリケート表を管理するすべてのサイトでデータレベルの変更が可能でなければなりません。可能でない場合は、トランザクションがロールバックされます。

一部のサイトは同期をとって変更を伝播し、他のサイトは非同期伝播（遅延トランザクション）を使用するレプリケート環境も作成できます。

レプリケーションの競合および同期データ・レプリケーション

共有所有権システムで変更内容をすべて同期をとってレプリケートするとき（即時レプリケーション）、レプリケーションの競合は発生しません。即時レプリケーションの場合、アプリケーションでは分散トランザクションを使用して表のレプリカをすべて同時に更新します。分散データベースでない環境の場合、各分散トランザクションのかわりに行を自動的にロックして、トランザクション間の障害を起こすようなあらゆる種類の干渉を防止します。即時レプリケーション・システムはレプリケーションの競合を防止できますが、このタイプのシステムはシステムのすべてのサイトが使用可能な場合にしか機能しないため、システムおよびネットワークの可用性に大きく依存します。

基本レプリケーションの使用方法

この章では、表の読み込み専用のスナップショットを使用して、基本レプリケーションの環境を構成および管理する方法を説明します。この章のトピックは次のとおりです。

- クイック・スタート：基本レプリケーション環境の作成
- スナップショット用データベースの設定
- スナップショット・ログの作成
- 単純スナップショットの作成
- 副問合せを使用するスナップショットの作成
- リフレッシュ・グループの作成
- 基本レプリケーション環境の管理
- スナップショットのパフォーマンスのチューニング
- 基本レプリケーションのその他のオプション
- 基本レプリケーション環境の監視

注意：この章では、SQL コマンドと Oracle Enterprise Manager を使用して基本レプリケーション・システムを作成および管理する方法について、例を示して説明します。Enterprise Manager の詳細は、Enterprise Manager のマニュアルを参照してください。

クイック・スタート：基本レプリケーション環境の作成

マスター・サイトへの読み込み専用アクセスを可能にする表のスナップショットを使用し基本レプリケーション環境を作成するには、次のステップをすべて実行してください。

1. 基本レプリケーション環境を設計する。どのマスター表を読み込み専用の表スナップショットを使用してレプリケートするか、およびどのデータベースにこのようなスナップショットが必要かを決定する。
2. 各スナップショット・サイトで、スナップショットをサポートするために必要なスキーマおよびデータベース・リンクを作成する。
3. マスター・サイトで、すべてのスナップショットの高速リフレッシュをサポートするために必要なスナップショット・ログを作成する。
4. 各スナップショット・サイトでスナップショットを作成する。
5. 各スナップショット・サイトで、スナップショットのリフレッシュに使用されるリフレッシュ・グループを作成し、各スナップショットをリフレッシュ・グループに割り当てる。
6. アプリケーション・ユーザーに、スナップショットにアクセスするために必要な権限を付与する。

各ステップの詳細は、この章の後述の項を参照してください。

簡単な例

次の簡単な例で、基本レプリケーション環境を作成するために必要なステップを説明します。

ステップ 1: 環境の設計

最初に、基本レプリケーション環境を設計します。この例では、マスター・サイト DBS1 の表 SCOTT.EMP と SCOTT.DEPT を、DBS2 の対応するスナップショットを使用してレプリケートする方法を紹介します。

注意：SCOTT.EMP の主キーは EMPNO 列で、SCOTT.DEPT の主キーは DEPTNO 列です。

ステップ 2: スナップショット・サイトのスキーマおよびデータベース・リンクの作成

マスター・サイト DBS1 には、主キーを持つ表 EMP および DEPT を使用するスキーマ SCOTT がすでに存在しています。スナップショット・サイトには、指定されたスナップショットを入れるために、対応するスキーマ SCOTT がなければなりません。さらに、スナップショットのスキーマ SCOTT には、マスター・サイトの対応するスキーマとの接続を確立するプライベート・データベース・リンクが必要です。

次の SQL コマンド・スクリプトを使用すると、スナップショット・サイトのスキーマおよびデータベース・リンクの設定が完了します。

```
CONNECT system/manager@db2;  
CREATE USER scott IDENTIFIED BY tiger QUOTA UNLIMITED ON data;
```

```
GRANT CONNECT TO scott;  
CONNECT scott/tiger@dbs2;  
CREATE DATABASE LINK dbs1 CONNECT TO scott IDENTIFIED BY tiger;
```

ステップ 3: 必要なマスター・サイトのスナップショット・ログの作成

スナップショットを作成する前に、スナップショットの高速リフレッシュをサポートするために必要な、マスター・サイトのスナップショット・ログを作成してください。次の SQL コマンド・スクリプトは、スナップショットをサポートする、マスター・サイトのスナップショット・ログの作成例です。

```
CONNECT system/manager@dbs1;  
CREATE SNAPSHOT LOG ON scott.emp;  
CREATE SNAPSHOT LOG ON scott.dept;
```

ステップ 4: スナップショットの作成

必要なスナップショット・ログを設定したら、スナップショットを作成できます。次の SQL コマンド・スクリプトは、スナップショット・サイトのスナップショット SCOTT.EMP および SCOTT.DEPT の作成例です。

```
CONNECT system/manager@dbs2;  
CREATE SNAPSHOT scott.emp AS SELECT * FROM scott.emp@dbs1.acme.com;  
CREATE SNAPSHOT scott.dept AS SELECT * FROM scott.dept@dbs1.acme.com;
```

ステップ 5: スナップショット・サイト・リフレッシュ・グループの作成

スナップショットを作成したら、関連するすべてのスナップショットを、リフレッシュに使用するリフレッシュ・グループに必ず割り当てます。次の SQL コマンド・スクリプトは、スナップショット・サイトにリフレッシュ・グループ SCOTT.REFGRP1 を作成およびスケジューリングして、それに新規のスナップショット EMP および DEPT を割り当てる例です。

```
CONNECT system/manager@dbs2;  
DBMS_REFRESH.MAKE(  
    name => 'scott.refgrp1',  
    list => 'scott.dept,scott.emp',  
    next_date => SYSDATE,  
    interval => 'SYSDATE+1/24');  
COMMIT;
```

ステップ 6: スナップショットへのアクセス権限の付与

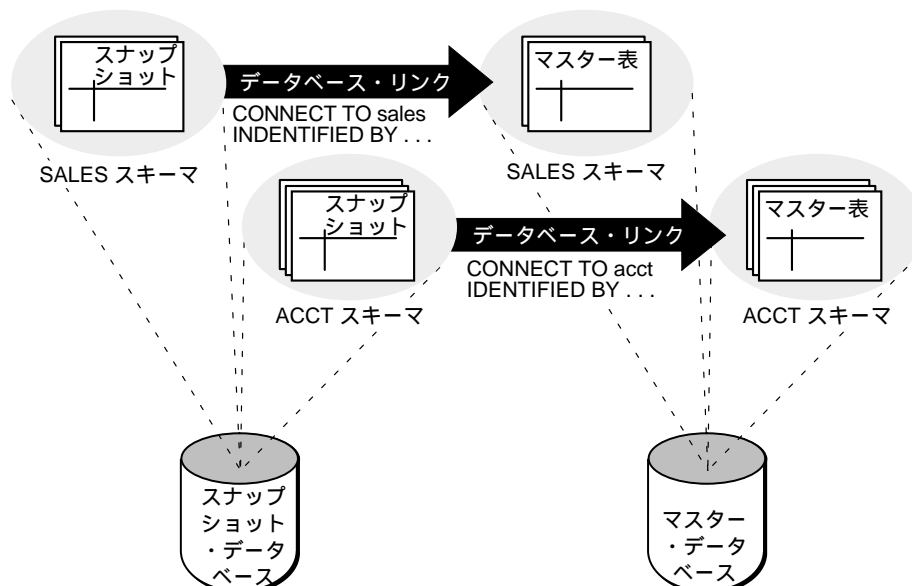
スナップショットを構成したら、ユーザーにアクセス権限を付与します。

```
GRANT SELECT ON scott.emp TO ... ;
```

スナップショット用データベースの設定

基本レプリケーション環境のスナップショットを作成する前に、各スナップショット・サイトでスナップショット用の設定が必要です。特に、各スナップショット・サイトには、指定されたスナップショットをサポートするためのスキーマおよびデータベース・リンクがなければなりません。基本レプリケーションの構成を簡単にするには、スキーマおよびデータベース・リンクの設計を行います。この方法は、図 2-1 および以降の項で説明します。

図 2-1 基本レプリケーション環境用の推奨スキーマおよびデータベース・リンク構成



必要なスキーマ

スナップショット・データベースのスナップショットを含むスキーマは、マスター・データベースのマスター表を含むスキーマに対応しなければなりません。したがって、読み込み専用の表のスナップショットを使用してレプリケートするマスター・データベースの表およびそれを含むスキーマを特定してください。次に、スナップショット・データベースで、マスター・データベースのマスター表を含むスキーマと同じ名前で作成します。たとえば、すべてのマスター表が DB1 データベースの SALES スキーマ内にある場合、スナップショット・データベース DB2 に、対応する SALES アカウントを作成します。

必要なデータベース・リンク

スナップショットを定義する問合せでは、1 つ以上のデータベース・リンクを使用して、リモート表のデータを参照します。スナップショットを作成するには、スナップショットで使用する予定のデータベース・リンクが使用可能になっていなければなりません。さらに、データベース・リンクでリモート・データベースへのアクセスに使用するアカウントでは、セキュリティ・コンテキストが定義され、それに基づいてスナップショットの作成、および後続のリフレッシュが実行されます。

正しい動作を保証するために、スナップショットの定義問合せでは、定義にユーザー名およびパスワードが埋め込まれているデータベース・リンクを使用する必要があります。このタイプのデータベース・リンクでは、指定したアカウントを使用して、常にリモート・データベースとの接続が確立されます。また、リンクで使用するリモート・アカウントには、スナップショットの定義問合せの中で参照されるデータにアクセスするために必要な権限がなければなりません。

基本レプリケーション環境の設定を簡単にするには、スナップショット・データベースの各スナップショットのスキーマから、マスター・データベースの対応するスキーマへのプライベート・データベースを作成します。スナップショット・データベースの各プライベート・データベース・リンクに、対応するマスター・データベースのアカウント情報を必ず埋め込んでください。たとえば、スナップショット・データベース DB2 の SALES スキーマには、SALES ユーザー名およびパスワードを使用して接続するプライベート・データベース・リンク DB1 がなければなりません。

注意：分散システム間でのスナップショット問合せの詳細は、『Oracle8 Server 分散システム』を参照してください。

スナップショット・リフレッシュの計画

スナップショットを作成する前に、スナップショット・リフレッシュ・グループを使用して、スナップショットをどのようにリフレッシュするかを考えておいてください。事前にいくつかの項目を検討しておく、リフレッシュ・グループの構成がより簡単になります。

必要な権限の付与

基本的なレプリケート環境では、権限の管理が可能です。スナップショットを作成およびリフレッシュするには、作成者とスナップショットの所有者が、ともにスナップショットの定義問合せを発行できなければなりません。これは、スナップショットの定義問合せでスナップショットのマスター表へのアクセスに使用されるデータベース・リンクに依存します。基本レプリケーション環境でのスナップショット・リフレッシュの権限の管理を容易にするには、前述の項で説明した推奨スキーマ/データベース・リンクを使用します。

リフレッシュ・グループの設計

リフレッシュ・グループを作成すると、多数のスナップショットをグループとして効率的にリフレッシュできるだけでなく、複数の関連するマスター表の表スナップショット間の参照整合性およびトランザクション一貫性が保証されます。リフレッシュ・グループのスナップショットがすべてリフレッシュされると、グループのすべてのスナップショットのデータは、トランザクションの一貫性を保つある同じ時点のデータに対応します。

基本レプリケーション環境を作成する前に、参照整合性およびトランザクション整合性を保つ必要があるスナップショットを特定します。次にそれに従って、関連するスナップショットを同じリフレッシュ・グループにグループ分けします。

注意：何の規則性もなしにスナップショットを1つにまとめて大きいリフレッシュ・グループを作成しないでください。不必要に大きいリフレッシュ・グループのスナップショット・リフレッシュを実行すると、大量のロールバック・セグメントを使用するために必要となる、膨大な量のロールバック・データが生成される可能性があります。

また、アプリケーションで必要とされるスナップショットのリフレッシュ頻度を判断することが重要です。マスター表での更新の頻度が予測できる場合は、対応付けられたスナップショットがあらかじめ適切な間隔で自動的にリフレッシュされるようにします。

SNP バックグラウンド・プロセスの起動

管理を容易にするために、ほとんどの基本レプリケーション環境では、リフレッシュ・グループを構成してスナップショットをすべて自動的にリフレッシュします。基本レプリケーション環境の各スナップショット・データベースでは、スナップショット・リフレッシュ・グループの自動リフレッシュをサポートするために、1つ以上の SNP バックグラウンド・プロセスを開始する必要があります。各サーバーの SNP バックグラウンド・プロセスの設定は、次の初期化パラメータで制御されます。

- `JOB_QUEUE_PROCESSES`。サーバーごとに `SNPn` バックグラウンド・プロセスの数を指定します。`n` は0から9までの数値で、その後に A から Z のアルファベットが続きます。リフレッシュ・グループを構成してスナップショットを自動的に更新するときは、このパラメータの値を1以上に設定します。同時にリフレッシュするリフレッシュ・グループが多数ある場合を除いて、通常はスナップショット・リフレッシュ・プロセスは1つで十分です。
- `JOB_QUEUE_INTERVAL` で、サーバーのバックグラウンド・プロセスの起動の間隔を秒単位で指定します。一般的なレプリケーション環境では、デフォルトの60秒が適当です。

スナップショット・ログの作成

各マスター表には、対応するスナップショットの効率的な高速リフレッシュをサポートするために、関連するスナップショット・ログを設定する必要があります。マスター表に関連するスナップショット・ログが設定されていない場合、対応するスナップショットでは完全リフレッシュしか実行できません。

マスター表のスナップショット・ログは、同じデータベースに表として作成されます。つまり、マスター表のスナップショット・ログはそれ自体が表です。トランザクションによりマスター表の情報が変更されると、マスター表の内部トリガーにより、対応するスナップショット・ログに行が自動的に挿入されます。スナップショット・ログの行には、マスター表に加えられた変更とともに、どのスナップショットがマスター表の変更を反映するために更新され、どのスナップショットが更新されていないかについての情報も記録されます。

1つのスナップショット・ログは、1つのマスター表に対応付けられます。つまり、1つのマスター表には1つのスナップショット・ログしか設定できません。マスター表が（おそらく異なるデータベースの）複数の異なるスナップショットのデータ・ソースである場合、マスター表のすべてのスナップショットで同じスナップショット・ログが使用されます。

注意：スナップショット・ログの管理の詳細は、2-27 ページの「スナップショット・ログの管理」を参照してください。

マスター表のスナップショット・ログを作成するには、Enterprise Manager の Schema Manager アプリケーションまたはSQLコマンドCREATE SNAPSHOT LOGを使用できます。次は、スナップショット・ログの作成例です。



これと同等の CREATE SNAPSHOT LOG 文は次のようになります。

```
CREATE SNAPSHOT LOG ON scott.emp;
```

マスター表のスナップショット・ログを作成する前に、以降の項で説明するいくつかの事項について考慮する必要があります。

スナップショット・ログ名

マスター表のスナップショット・ログを作成すると、マスター表を含むスキーマに、MLOG\$_master_table_name という名前の表としてログが自動的に作成されます。マスター表の名前が 20 バイトを超える場合、master_table_name の部分が 20 バイトに切り捨てられ、一意性を保証するために 4 桁の数値が付加されます。これによって、オブジェクトがスキーマ・オブジェクトの命名規則に従うことが保証されます。

必要な権限

スナップショット・ログの作成に必要な権限は、スナップショット・ログに関連するオブジェクトを作成するために必要な権限と同じです。

- ユーザーは、自身のマスター表を所有している場合、そのマスター表のスナップショット・ログを作成できる（ただし、CREATE TABLE 権限が必要です）。
- 別のユーザーのスキーマの表に対するスナップショット・ログを作成している場合は、マスター表のSELECT権限の他に、CREATE ANY TABLEおよびCOMMENT ANY TABLE権限が必要である。

いずれの場合も、スナップショット・ログの所有者には、スナップショット・ログを保持するために必要な割当て制限が表領域に対してなければなりません。

スナップショット・ログ作成のタイミング

スナップショットに高速リフレッシュを実行する場合は、スナップショットを作成する前に、そのマスター表に対応するスナップショット・ログを必ず作成してください。スナップショットを作成した後にスナップショット・ログを作成すると、最初のリフレッシュは、高速リフレッシュではなく完全リフレッシュが実行されます。

主キー・スナップショットの特別要件

デフォルトでは、主キー・スナップショットをサポートするスナップショット・ログが作成されます。したがって、スナップショット・ログを作成するには、マスター表に有効な PRIMARY KEY 制約が含まれていなければなりません。

スナップショット・ログ記憶領域パラメータ

Schema Managerの「スナップショット・ログの作成」プロパティ・シートの「記憶領域」ページを使用すると、作成時にスナップショット・ログの記憶領域オプションを設定できます。



通常は、スナップショット・ログの記憶領域オプションは、次の規則に従って設定してください。

- PCTFREE を 0 に、PCTUSED を 99 に設定する。
- エクステント記憶領域パラメータは、マスター表での更新アクティビティ (INSERT および UPDATE、DELETE 文の数) に応じて設定する。

副問合せスナップショットの特別要件

定義問合せで副問合せを使用する単純スナップショットをサポートするには、次の要件を満たす必要があります。

- 副問合せに表示される各マスター表に、対応するスナップショット・ログが設定されていないといけない。
- 各スナップショット・ログで、スナップショットの副問合せで参照する各列をフィルタ列に指定しなければならない (主キー列は除く)。

注意: LOB データ型を使用する列はフィルタ列に指定できません。

副問合せを使用するスナップショットをサポートするために必要なフィルタ列を指定したスナップショット・ログの例は、2-16 ページの「副問合せを使用する高度なサブセット化」を参照してください。

スナップショット・ログ作成の内部メカニズム

スナップショット・ログが作成されると、いくつかの操作が内部で実行されます。

- MLOG\$_master_table_name という名前の実表が作成され、マスター表内の更新した行の主キー（または ROWID、あるいはその両方）およびフィルタ列、タイムスタンプが保存される。タイムスタンプ列は、ログがスナップショット・リフレッシュに最初に使用されるまでは更新されません。
- マスター表上で内部トリガーが起動され、挿入および更新、削除された行の主キー（または ROWID、あるいはその両方）およびフィルタ列、タイムスタンプがマスターのスナップショット・ログに挿入される。

マスター表を含むスキーマには、スナップショット・ログの実表も含まれます。

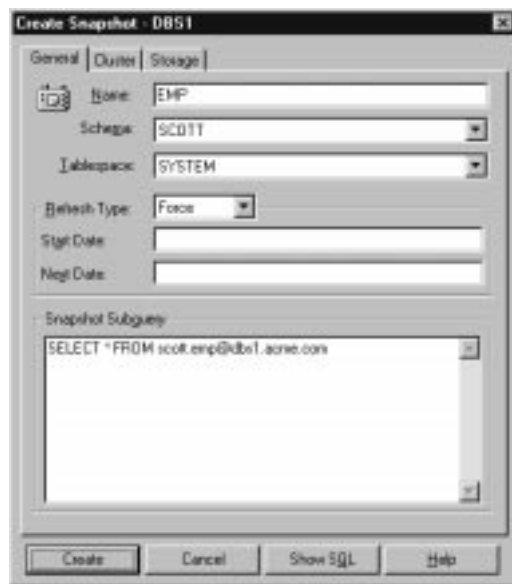
注意：スナップショット・ログの実表や表のデータは変更しないでください。

単純スナップショットの作成

読み専用スナップショットを作成するには、Enterprise Manager の Schema Manager アプリケーションの「スナップショットの作成」プロパティ・シートまたは SQL コマンド CREATE SNAPSHOT を使用できます。

注意：スナップショットの管理の詳細は、2-33 ページの「読み専用スナップショットの管理」を参照してください。

次は、単純な読み専用スナップショットの作成例です。



これと同等の CREATE SNAPSHOT 文は次のようになります。

```
CREATE SNAPSHOT scott.emp  
AS SELECT * FROM scott.emp@db1.acme.com;
```

読み込み専用の表スナップショットを作成する前に、以降の項で説明する事項を考慮してください。

スナップショット名

スナップショット名は、それを含むスキーマ内で一意でなければなりません。スナップショット名は最大 30 バイトまで指定できますが、可能な限り 19 バイト以下にしてください。スナップショット名を 20 バイト以上にすると、対応する表の名前の接頭部が自動的に切り捨てられ、その部分には、新規オブジェクト名の一意性を保証しスキーマ・オブジェクトの命名規則に従うために、4 桁の数値が付加されます。

必要な権限

完全に機能するスナップショットを作成するにはいくつかの権限が必要です。

- スナップショットをユーザー自身のスキーマに作成する場合は、SNAPSHOT および CREATE TABLE、CREATE VIEW システム権限が必要である。
- スナップショットを他のユーザーのスキーマに作成する場合は、CREATE ANY SNAPSHOT システム権限が必要である。

- スナップショットを含むスキーマには、スナップショットの実表および索引を保存するために必要な割当て制限がターゲット表領域に対してなければならない。
- スナップショットを作成およびリフレッシュするには、作成者とスナップショットの所有者が、ともにスナップショットの定義問合せを発行できなければなりません。これは、スナップショットの定義問合せで使用するデータベース・リンクに直接依存します。

主キー・スナップショットの特別要件

デフォルトでは、新規スナップショットはすべて主キー・スナップショットとして作成されます。スナップショットの作成に必要な要件は次のとおりです。

- 新規スナップショットを作成する前に、スナップショットのマスター表に、使用可能な PRIMARY KEY 制約がなければならない。
- スナップショットの定義問合せでは、マスター表の主キーの列をすべて参照しなければならない。

スナップショット・リフレッシュの設定

Schema Manager および CREATE SNAPSHOT コマンドを使用すると、個々のスナップショットに対するリフレッシュの設定を指定できますが、スナップショットを常にリフレッシュ・グループの一部としてリフレッシュしなければなりません。リフレッシュ・グループを使用してスナップショット・リフレッシュを構成する方法の詳細は、2-24 ページの「リフレッシュ・グループの作成」を参照してください。また、リフレッシュ用に個々のスナップショットを構成する方法の詳細は、2-42 ページの「スナップショットの個別リフレッシュ」を参照してください。

スナップショットの定義問合せ

新規スナップショットを作成するときは、スナップショットの構造を定義する問合せに関して、次の事項を考慮してください。

参照するマスター表の列のデータ型

スナップショットの定義問合せでは、次の Oracle データ型を使用するマスター表の列を参照できます。NUMBER および DATE、CHAR、VARCHAR2、NCHAR、NVARCHAR2、RAW、ROWID、BLOB、CLOB、NCLOB です。スナップショットには、LONG または LONG RAW、BFILE データ型を使用する列を組み込むことはできません。また、スナップショット内では、ユーザー定義オブジェクト型はサポートされていません。

高速リフレッシュ中に BLOB および CLOB、NCLOB が伝播されるのは、次の場合だけです。

- LOB が更新されたとき（部分的な更新および追加も含む）
- スナップショットの選択条件を満たす新規行が存在するとき

スナップショットの定義問合せでは、WHERE 句で LOB 列は参照できません。

単純スナップショットに関する制限事項

単純スナップショットを作成する場合、スナップショットの定義問合せに次の SQL 属性を組み込むことはできません。

- 個別関数または集約関数
- GROUP BY 句または CONNECT BY 句
- 結合（副問合せに許可されている型を除く）
- 集合演算

スナップショットの定義で上記の属性のいずれかが使用されているときは、スナップショットは複合スナップショットになります。複合スナップショットに対しては、スナップショット・ログを使用して高速リフレッシュを実行することはできません。

明示的な表参照

スナップショットの定義問合せは、常にそのリモート・データを明示的に参照するように設計してください。このように設計しないと、スナップショットの作成およびその後のリフレッシュ時に、別のリモート・データが参照される可能性があります。たとえば、SCOTT が次のコマンドを使用してスナップショットを作成したとします。

```
CONNECT scott/tiger
CREATE DATABASE LINK sales.hq.com USING 'hq.sales.com';
CREATE SNAPSHOT emp AS SELECT * FROM emp@sales.hq.com;
```

この例では、スナップショットの定義で、リモート表 SCOTT.EMP を暗黙的に参照しています。これは、データベース・リンクにより、スナップショットの作成時に SCOTT/TIGER としてリモート・データベースへの接続が確立されるためです。このかわりに、次の例のように、リモート表を明示的に参照してください。

```
CREATE SNAPSHOT emp AS SELECT * FROM scott.emp@sales.hq.com;
```

スナップショットに対応するリモート・データを明確に識別するには、スナップショットの定義問合せで次のことを行う必要があります。

- 部分的な表名ではなく、省略のない形の表名を参照する。
- リモート表だけを参照して、リモートのマスター・ビューまたはシノニムは参照しない。
- コンテキスト依存データを生成しない。たとえば、SQL 関数 SYSDATE または UID、USER を使用する問合せを持つ単純スナップショットは作成しないでください。

スナップショットの記憶領域の設定

Schema Manager の「スナップショットの作成」プロパティ・シートの「記憶領域」ページを使用すると、作成時にスナップショットの記憶領域オプションを設定できます。



一般的に、単純なスナップショットはマスター表と同じ特性を共有するので、記憶領域オプションには、そのマスター表と同じ記憶領域オプションを使用してください。ただし、単純スナップショットにマスター表の列のコピーがすべて含まれていない場合は、スナップショットの記憶領域の項目をそれに応じて変更します。

クラスタ化スナップショット

多くのマスター表がマスター・データベース内でクラスタ化されている場合は、リモート・データベース内の対応するスナップショットもクラスタ化することを考慮してください。データ・クラスタの一部としてスナップショットを作成するには、Schema Manager の「スナップショットの作成」プロパティ・シートの「クラスタ」ページを使用します。



データ・クラスタ内にスナップショットを作成すると、常にクラスタのデータ・セグメントの記憶領域パラメータが使用されます。

データ・ロード・オプション

デフォルトでは、スナップショットを作成すると、対応するマスター表の行でスナップショットの実表を組み入れるために、その直後にスナップショットの定義問合せが実行されます。大規模なレプリケート環境では、スナップショット・データベースの作成に必要なネットワークのオーバーヘッドを減らすために、スナップショット・クローニングの使用を考慮する必要があります。スナップショット・クローニングの詳細は、7-13 ページの「スナップショット・クローニングとオフライン・インスタンスエーション」を参照してください。

副問合せを使用するスナップショットの作成

多くの場合、スナップショットは、マスター表の行のサブセットに対応する必要があります。スナップショットの定義問合せで簡単な WHERE 句を使用して、マスター表の行のサブセットを特定することもできます。たとえば、顧客の郵便番号で各販売員の担当地域を定義する、企業の販売部門自動化システムを考えてみます。各販売員のパソコンのデータベースには、担当地域の顧客に関する情報と、その顧客の注文だけが必要です。たとえば、郵便番号 19555 の地域の顧客を担当する販売員は、その担当地域の顧客に読み専用でアクセスするスナップショットを作成できます。

```
CREATE SNAPSHOT sales.customers AS
```

副問合せを使用するスナップショットの作成

```
SELECT * FROM sales.customers@dbs1.acme.com
WHERE zip = 19555;
```

この簡単な例は、単純スナップショットの基本的なサブセット化の作成方法を示しています。より高度なサブセット化を行うには、スナップショットの定義問合せで副問合せを使用しなければならない場合があります。以降の項では、上記の例を基に、副問合せスナップショットの利点について説明します。それぞれ関連する例で、副問合せを使用するスナップショットの作成方法を説明します。

副問合せを使用する高度なサブセット化

副問合せを使用するスナップショットは、高度なサブセット化要件に対する柔軟な解決方法です。次の例では、上述した仮定の販売部門自動化システムで、副問合せを使用するスナップショットの作成方法および利点を紹介します。次の例で、中央制御の会社のデータベース名はhq.acme.comとします。次のCREATE TABLE コマンドは、中央制御の会社のデータベースのSALESスキーマに、販売部門自動化システムの重要なマスター表をいくつか記述しています。

CUSTOMERS 表

```
CREATE TABLE sales.customers
( c_id INTEGER PRIMARY KEY,
  zip INTEGER
  -- other columns defined here
);
```

ORDERS 表

```
CREATE TABLE sales.orders
( o_id INTEGER PRIMARY KEY,
  c_id INTEGER
  -- other columns defined here
);
```

ORDER_LINES 表

```
CREATE TABLE sales.order_lines
( ol_id INTEGER,
  o_id INTEGER,
  PRIMARY KEY (ol_id, o_id)
  -- other columns defined here );
```

ASSIGNMENTS 表

```
CREATE TABLE sales.assignments
( c_id INTEGER,
  s_id INTEGER,
  PRIMARY KEY (c_id, s_id)
);
```

SALESPERSONS 表

```
CREATE TABLE sales.salespersons
( s_id INTEGER PRIMARY KEY,
  s_name VARCHAR2(30) UNIQUE
  -- other columns defined here
);
```

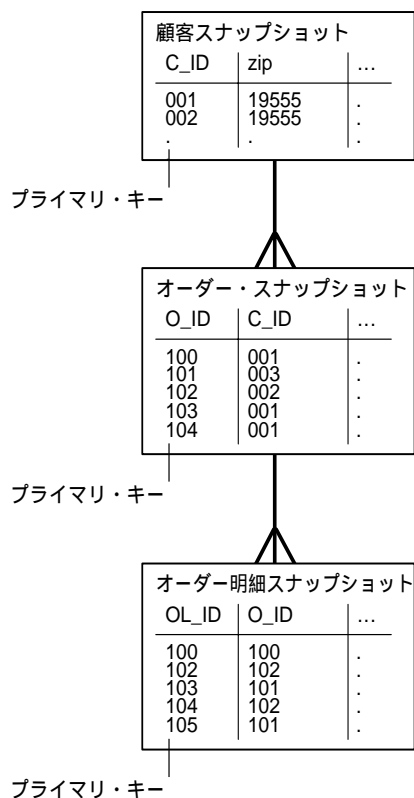
高速リフレッシュを実行できるスナップショットを作成するには、対応するマスター表にスナップショット・ログが存在しなければなりません。副問合せを使用するスナップショットが参照する各マスター表に対しては、対応するスナップショット・ログに適切なフィルタ列が組み込まれていなければなりません。次の SQL コマンドは、中央制御のデータベース hq.acme.com に必要なスナップショット・ログを作成します。

```
CREATE SNAPSHOT LOG ON sales.customers
  WITH PRIMARY KEY (zip);
CREATE SNAPSHOT LOG ON sales.orders
  WITH PRIMARY KEY (c_id);
CREATE SNAPSHOT LOG ON sales.order_lines
  WITH PRIMARY KEY;
CREATE SNAPSHOT LOG ON sales.assignments
  WITH PRIMARY KEY;
CREATE SNAPSHOT LOG ON sales.salespersons
  WITH PRIMARY KEY (s_name);
```

次の例では、スナップショットをサポートするために必要な、対応する SALES スキーマおよびプライベート・データベース・リンクが、spdb1.acme.com データベースにすでに存在していることも仮定しています。

例 1

図 2-2 副問合せを使用する高度なサブセット化



前述の項の CUSTOMERS スナップショットを使用すると、販売員の顧客レコードに読み専用でアクセスできます。販売員には、自分の顧客の注文情報に読み専用でアクセスできるスナップショットも必要です。1 つの方法としては、ORDERS および ORDER_LINES マスター表に郵便番号列を追加し、CUSTOMERS スナップショットと同様に単純スナップショットを定義できます。ただし、マスター表のこのように冗長化するのは通常好ましくなく、管理も難しくなります。よりよい解決方法は、副問合せを使用して、CUSTOMERS 表の郵便番号を参照するスナップショットを ORDERS および ORDER_LINES マスター表上に定義することです。

次の SQL コマンドは、販売員の担当地域に対応する注文と明細項目を含む、ORDERS マスター表と ORDER_LINES マスター表のスナップショットの作成方法を示しています。

ORDERS スナップショット

```
CREATE SNAPSHOT sales.orders AS
SELECT * FROM sales.orders@hq.acme.com o
WHERE EXISTS
( SELECT c_id FROM sales.customers@hq.acme.com c
  WHERE o.c_id = c.c_id AND zip = 19555);
```

ORDER_LINES スナップショット

```
CREATE SNAPSHOT sales.order_lines AS
SELECT * FROM sales.order_lines@hq.acme.com ol
WHERE EXISTS
( SELECT o_id FROM sales.orders@hq.acme.com o
  WHERE ol.o_id = o.o_id
  AND EXISTS
    ( SELECT c_id FROM sales.customers@hq.acme.com c
      WHERE o.c_id = c.c_id AND zip = 19555));
```

例の副問合せでは、1つ以上のレベルに関連する、子表から親表への複数対1参照をウォーク・アップします。これらのスナップショットが作成されると、郵便番号列の値がスナップショットの選択条件と一致する顧客に属するすべての注文行または注文ライン行でスナップショットの実表が埋められます。後続の高速リフレッシュでは、スナップショットの作成後または最終リフレッシュ後に変更された行だけが戻されます。

郵便番号列が更新されて、ある顧客がスナップショットの選択条件を満たさなくなった場合は、次のリフレッシュ時にその顧客の注文行と注文ライン行がスナップショットから削除されます。郵便番号列が更新されて別の顧客がスナップショットの選択条件を満たすようになった場合は、次のリフレッシュ時にその顧客の注文行と注文ライン行が追加されます。

例 2: より優れた方法

前述の項で定義したスナップショットでは、各スナップショットの定義問合せに郵便番号を埋め込むことで、CUSTOMERS および ORDERS、ORDER_LINES 表のサブセットを取り出しています。このため、スナップショットは販売員の担当地域の単純な変更に対応できません。たとえば、販売員の担当地域の郵便番号が 19555 から 19500 に変更された場合、スナップショットを再作成する必要があります。柔軟性を向上させるために、次の例のスナップショットで、副問合せを使用して、選択されたケースの表間での複数対複数参照の問合せを可能にする方法を示します。

この例では、SALESPERSONS 表および ASSIGNMENTS 表を使用します。SALESPERSONS 表にはそれぞれの販売員用の行が 1 行ずつあり、S_ID が各販売員の名前にマップされています。ASSIGNMENTS 表は、顧客を販売員に（たとえば、C_ID を S_ID に）マップする交差部表です。ASSIGNMENTS 表は、CUSTOMER と SALESPERSON との間での複数対複数の関連を表します。つまり、販売員に複数の顧客を割り当てることも、顧客に複数の販売員を割り当てることもできます。

次の SQL コマンドでは、高速リフレッシュが可能な新規の CUSTOMERS および ORDERS、ORDER_LINES スナップショットが作成されます。

CUSTOMERS スナップショット

```
CREATE SNAPSHOT sales.customers AS
SELECT * FROM sales.customers@hq.acme.com c
    -- conditions for customers
WHERE EXISTS
( SELECT * FROM sales.assignments@hq.acme.com a
  WHERE a.c_id = c.c_id
  AND EXISTS
    ( SELECT * FROM sales.salespersons@hq.acme.com s
      WHERE s.s_id = a.s_id AND s_name = 'gsmith'));
```

ORDERS スナップショット

```
CREATE SNAPSHOT sales.orders AS
SELECT * FROM sales.orders@hq.acme.com o
    -- conditions for customers
WHERE EXISTS
( SELECT c_id FROM sales.customers@hq.acme.com c
  WHERE o.c_id = c.c_id
  AND EXISTS
    ( SELECT * FROM sales.assignments@hq.acme.com a
      WHERE a.c_id = c.c_id
      AND EXISTS
        ( SELECT * FROM sales.salespersons@hq.acme.com s
          WHERE s.s_id = a.s_id AND s_name = 'gsmith'))));
```

ORDER_LINES スナップショット

```
CREATE SNAPSHOT sales.order_lines AS
SELECT * FROM sales.order_lines@hq.acme.com ol
WHERE EXISTS
( SELECT o_id FROM sales.orders@hq.acme.com o
```



```

WHERE ol.o_id = o.o_id
AND EXISTS
( SELECT c_id FROM sales.customers@hq.acme.com c
  WHERE o.c_id = c.c_id
  -- conditions for customers
  AND EXISTS
  ( SELECT * FROM sales.assignments@hq.acme.com a
    WHERE a.c_id = c.c_id
    AND EXISTS
    ( SELECT * FROM sales.salespersons@hq.acme.com s
      WHERE s.s_id = a.s_id AND s_name = 'gsmith' ) ) ) );

```

これらの副問合せでは、ORDER_LINES から ORDERS、ORDERS から CUSTOMERS への複数対1参照がウォーク・アップされ、ASSIGNMENTS 交差部表を使用して、CUSTOMERS と SALESPERSONS 間の複数対複数参照が問い合わせられます。この場合も、CUSTOMERS および ORDERS、ORDER_LINES スナップショットが作成されると、販売員に割り当てられている顧客に属するすべての顧客行および注文行、注文ライン行でスナップショットの実表が埋められます。後続の高速リフレッシュでは、スナップショットの作成後または最終リフレッシュ後に変更された行だけが戻されます。ASSIGNMENTS レコードが削除された場合、または顧客が販売員に属さなくなるような更新が行われた場合は、次のリフレッシュ時にその顧客の顧客行および注文行、注文ライン行がスナップショットから削除されます。ASSIGNMENTS レコードが挿入された場合、または顧客が販売員に属するように更新された場合は、次のリフレッシュ時に適切な顧客行および注文行、注文ライン行が追加されます。

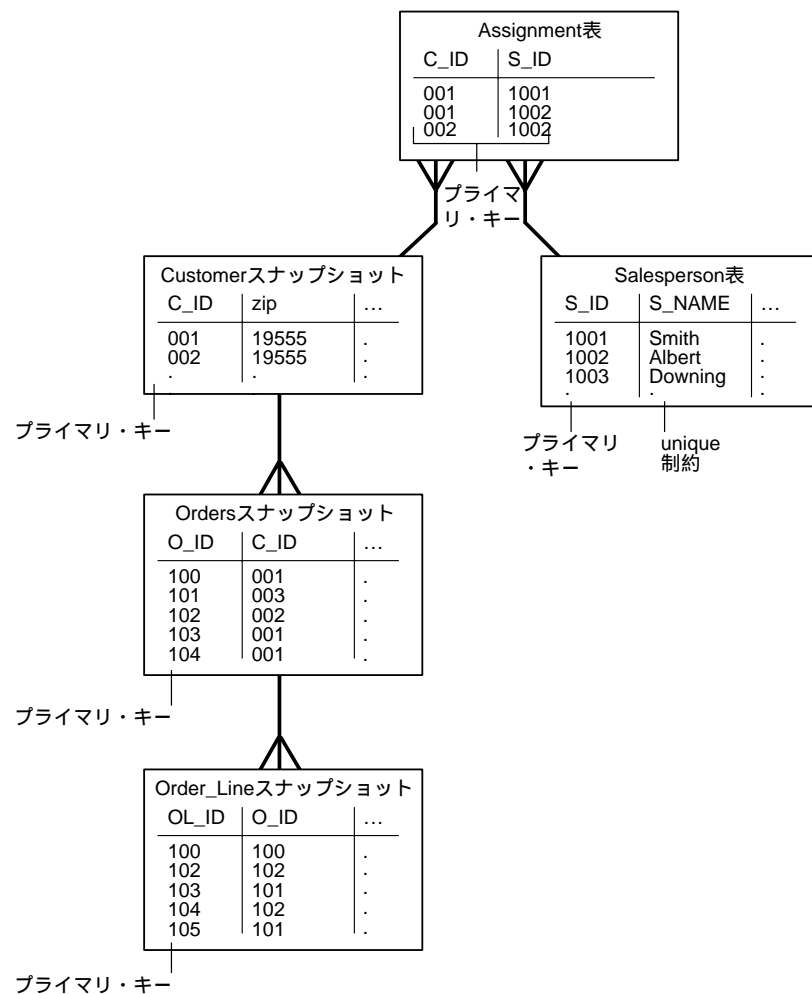
SALESPERSONS 表への結合を防止するために、副問合せを使用する前述のスナップショットを、ASSIGNMENTS 交差部表だけを使用して定義することもできます。たとえば、次のように ASSIGNMENTS 表の S_ID 列への参照だけを使用して、CUSTOMERS スナップショットを作成できます。

```

CREATE SNAPSHOT sales.customers AS
SELECT * FROM sales.customers@dbsl.acme.com c
-- conditions for customers
WHERE EXISTS
( SELECT * FROM sales.assignments@dbsl.acme.com a
  WHERE a.c_id = c.c_id AND a.s_id = 1001 );

```

図 2-3 副問合せを使用する高度なサブセット化



複数対複数関連を問い合わせる副問合せを使用すると、多くの場合柔軟性が向上します。例 2 では、ASSIGNMENTS を適切に変更することにより、販売員に異なる地域、つまり異なる顧客を割り当てることができます。最初の例では、販売員に別の郵便番号が割り当てられると、新しい郵便番号地域の割当てをスナップショット定義に組み込むために、スナップショットを削除して再作成する必要がありました。

同様に、例 2 では、会社は地域の定義と割当ての方法を変更できました。この会社は、ASSIGNMENTS 表に対する適切な更新を取り出して実行するだけで、郵便番号に基づいた地域別の構造から、産業分野別またはナショナル・アカウントに基づくより精巧な構造に移行できます。最初の例では、業界情報別またはナショナル・アカウント情報（あるいはその両方）を組み込むためには、列を CUSTOMERS 表に追加し、スナップショットを削除してから再作成してこれらの列を使用するようにしなければなりません。

副問合せを使用するスナップショットに関する制限事項

単純スナップショットに関する問題点はすべて、副問合せを使用するスナップショットにも関係します。詳細は、2-10 ページの「単純スナップショットの作成」を参照してください。この他に、副問合せを使用するスナップショットの定義問合せには、単純スナップショットの高速リフレッシュ機能を保つためにその他の制限事項がいくつかあります。

注意：単純副問合せスナップショットが次の多数の制限事項を満たしているかどうかを判断するには、「高速リフレッシュ」を使用するスナップショットを作成してください。単純副問合せスナップショットに関する制限事項に違反している場合は、エラーが返されます。

- 副問合せでは、次の場合に子表から親表への複数対 1 参照を "ウォーク・アップ" できる。
 - スナップショットを作成する前に、各親表の参照される結合列すべてに PRIMARY KEY 制約または UNIQUE 制約があるとき。
 - 結合式で、完全一致または等価比較 ("equi-joins" など) を使用するとき。
- 副問合せでは、次のような条件で複数対複数参照を問い合わせることもできる。
 - スナップショットを作成する前に、交差部表の結合列の両方をセットとした PRIMARY KEY 制約または UNIQUE 制約があり、かつ、交差部表が結合している表の結合列すべてに PRIMARY KEY 制約または UNIQUE 制約、フィルタ列すべてに別の UNIQUE 制約がある。交差部表しかない場合は、その交差部表の結合列すべてとフィルタ列の両方を含む PRIMARY KEY 制約または UNIQUE 制約がなければなりません。
 - 副問合せで、完全一致または等価比較が指定される。

注意：これら 2 つのプロパティを組み合わせると、複数対複数参照で 1 行だけが戻されます。詳細は、2-19 ページの「例 2: より優れた方法」を参照してください。ASSIGNMENT 表の PRIMARY KEY 制約には、C_ID 結合列と S_ID 結合列の両方が含まれます。SALESPERSON 表の S_ID 結合列には PRIMARY KEY 制約があり、S_NAME フィルタ列には UNIQUE 制約があります。副問合せは S_NAME 内の値が定数（この場合は 'gsmith'）と等しいことを指定しています。
- スナップショットは主キー・スナップショットでなければならない。
- マスター表のスナップショット・ログには、副問合せで参照されるフィルタ列がすべて含まれていなければならない。注意：フィルタ列は、副問合せ述語で参照されている主キー以外のキー列です。詳細は、2-9 ページの「副問合せスナップショットの特別要件」を参照してください。

- 副問合せは、肯定副問合せでなければならない。たとえば、EXISTSは使用できますが、NOT EXISTS は使用できません。
- 副問合せに AND 式を組み込むことができるが、各 OR 式は 1 行に含まれている列だけを参照できる。1 つの副問合せ内の複数の OR 式は、AND 処理されます。
- 各表は、副問合せ内で 1 度だけ結合できる。たとえば、各表は単一の EXISTS 式にだけ存在できます。
- 副問合せで参照される各表は、同じマスター・データベースに存在しなければならない。
- 選択された列数と、副問合せ内で使用される各表の主キー列の数の合計は、表で認められている最大列数 (1,000 列) より少なくななければならない。

リフレッシュ・グループの作成

表スナップショットには、ある時点で存在していたとおりにそのマスター・データが表示されます。スナップショットのデータを比較的新しいマスターのデータと一致させるには、スナップショットを定期的にリフレッシュする必要があります。スナップショット・リフレッシュは、より最新のマスターの状態をスナップショットに反映させる効果的なバッチ処理です。

複数の関連するマスター表の表スナップショット間の参照整合性およびトランザクション一貫性を保つために、各スナップショットは、リフレッシュ・グループの一部として編成およびリフレッシュされます。リフレッシュ・グループのスナップショットがすべてリフレッシュされると、グループのすべてのスナップショットのデータは、トランザクションの一貫性を保つある同じ時点のデータに対応します。

注意：リフレッシュ・グループの管理の詳細は、2-36 ページの「スナップショット・リフレッシュとリフレッシュ・グループの管理」を参照してください。

リフレッシュ・グループを作成するには、Schema Manager または同等の DBMS_REFRESH-MAKE プロシージャのコールを使用できます。



- 「リフレッシュ・グループの作成」プロパティ・シートの「一般」ページで、新しいグループの名前およびスキーマ、リフレッシュの設定、対応するロールバック・セグメントを指定する。
- 「リフレッシュ・グループの作成」プロパティ・シートの「スナップショット」ページで、新しいグループのメンバー・スナップショットのリストを指定する。リフレッシュ・グループ内のスナップショットの最大数は 100 です。

DBMS_REFRESH.MAKE プロシージャの同等のコールは次のようになります。

```
DBMS_REFRESH.MAKE(
  name           => 'sales.refgrp_1',
  list           => 'sales.customers,sales.orders,sales.order_lines',
  next_date      => SYSDATE,
  interval       => 'SYSDATE + 1/24',
  implicit_destroy => TRUE,
  rollback_seg   => 'trs_1'
);
```

リフレッシュ・グループを作成する前に、考慮すべき事項がいくつかあります。

リフレッシュの設定

リフレッシュ・グループを作成すると、グループの 1 回目のリフレッシュ日およびその後のリフレッシュ間隔を決定できます。

次回日付

リフレッシュ・グループの1回目のリフレッシュは、Schema Managerの「リフレッシュ・グループの作成」プロパティ・シートの「次回日付」フィールド(DBMS_REFRESH.MAKEプロシージャのNEXT_DATEパラメータ)の設定で決まります。

間隔

リフレッシュ・グループのリフレッシュ間隔で、グループのすべてのスナップショットの自動リフレッシュ間隔が決まります。グループの自動リフレッシュ間隔を設定するときには、次の性質を理解しておいてください。

- リフレッシュ間隔を指定する日付または日付式の値は、将来のある時点でなければならない。
- リフレッシュ間隔は、リフレッシュの実行に要する時間よりも長くなければならない。
- 相対日付式の値は、最新のリフレッシュ日付から計算されるある時点でなければならない。ネットワークまたはシステムの障害により、グループ・リフレッシュがスケジュールどおりに実行されない場合、相対日付式の結果は状況に応じて変更されます。
- 明示日付式の値は、最新のリフレッシュ日付に関係なく特定の時点でなければならない。

日付式の例

次の例は、リフレッシュ・グループのリフレッシュ間隔の指定に使用できる一般的な日付式です。

- SYSDATE + 7
最新リフレッシュ日から計算して7日後にリフレッシュを実行する相対日付式。
- SYSDATE + 1/48
最新リフレッシュ日から計算して30分後にリフレッシュを実行する相対日付式。
- NEXT_DAY(TRUNC(SYSDATE), 'MONDAY') + 3/24
毎週月曜日の午前3時にリフレッシュを実行する明示日付式。
- NEXT_DAY(ADD_MONTHS(TRUNC(SYSDATE, 'Q'), 3), 'THURSDAY')
各四半期の第1木曜日にリフレッシュを実行する明示日付式。

更新可能スナップショットのリフレッシュ設定

リフレッシュ・グループを作成すると、アドバンスド・レプリケーション環境の更新可能スナップショットを含むリフレッシュ・グループ用に特別に設計されたその他の設定を指定できます。基本的な読み込みレプリケーション環境の場合は、デフォルト設定をそのまま使用してください。アドバンスド・レプリケーション・システムを使用するこれらの特別なリフレッシュ・グループ設定の構成方法の詳細は、4-20ページの「リフレッシュ・グループの管理」を参照してください。

リフレッシュ・タイプ

デフォルトでは、高速リフレッシュを使用して、グループの各スナップショットをリフレッシュしようとしています。次の状況のいずれかに該当する場合に限り、リフレッシュ・グループの個々のスナップショットの完全リフレッシュが実行されます。

- スナップショットに対してスナップショット・ログが使用可能でないとき
- スナップショットが複合であるとき（複合スナップショットは高速リフレッシュできません）
- スナップショットが個別のリフレッシュ設定で作成されているとき（できる限り避けてください）
- スナップショットのマスター表が一度削除されてから再作成されたとき

ロールバック・セグメントの設定

リフレッシュ・グループのスナップショットがリフレッシュされるときに、サーバーで大量のロールバック・データが生成されることがあります。リフレッシュ・グループを作成するときに、グループのリフレッシュ用に十分な大きさのロールバック・セグメントを確保してください。

基本レプリケーション環境の管理

以降の項では、スナップショット・ログおよびスナップショット、スナップショット・リフレッシュ・グループなど、基本レプリケーション環境のさまざまなコンポーネントの管理方法を説明します。

スナップショット・ログの管理

以降の項では、スナップショット・ログの管理方法を説明します。次のトピックが含まれています。

- スナップショット・ログの変更
- スナップショット・ログ領域の管理
- スナップショット・ログがあるマスター表の再編成
- スナップショット・ログの削除

スナップショット・ログの変更

スナップショット・ログを作成すると、その記憶領域パラメータを変更して、対応するスナップショットをサポートできます。以降の項では、スナップショット・ログの変更方法について詳しく説明します。

必要な権限

マスター表の所有者またはマスター表の SELECT 権限を持つユーザーだけが、スナップショット・ログを変更できます。

スナップショット・ログ記憶領域パラメータの変更

スナップショット・ログの記憶領域パラメータを変更するには、「スナップショット・ログ」プロパティ・シートの「記憶領域」ページと「オプション」ページ、または同等の ALTER SNAPSHOT LOG 文を使用します。次に例を示します。

```
ALTER SNAPSHOT LOG ON sales.customers
  PCTFREE 25
  PCTUSED 40;
```

スナップショット・ログを変更してフィルタ列を追加する

スナップショット・ログに新しいフィルタ列を追加するには、SQL コマンド ALTER SNAPSHOT LOG を使用します。次に例を示します。

```
ALTER SNAPSHOT LOG ON sales.customers
  ADD (zip);
```

スナップショット・ログ領域の管理

スナップショットのリフレッシュ時にスナップショット・ログ内のどの行が使用されたかが自動的に追跡され、ログが際限なく大きくなることのないように不要な行がログから削除されます。複数の単純スナップショットで同じスナップショット・ログが使用できるため、あるスナップショットのリフレッシュにすでに使用された行が、まだ別のスナップショットのリフレッシュに必要な場合があります。このため、すべてのスナップショットで使用されるまで行は削除されません。

たとえば、SPDB1 データベースの CUSTOMERS スナップショットがリフレッシュされたとします。しかし、マスター表および関連するスナップショット・ログを管理するサーバーは、そのスナップショットのリフレッシュに使用されたスナップショット・ログの行を、SPDB2 データベースの CUSTOMERS スナップショットもこれを使用してリフレッシュされるまで削除されません。

スナップショット・ログからの行の削除方法によっては、複数のスナップショットが同じマスター表から作成されているときに、スナップショット・ログが際限なく大きくなるという状況が発生します。このような状況は、たとえば1つのマスター表から複数のスナップショットが作成されていて、かつ次の条件に該当する場合に発生することがあります。

- あるスナップショットが自動リフレッシュ用に構成されていないで、長時間手動でリフレッシュされていない場合。
- あるスナップショットのリフレッシュ間隔が非常に長い場合（たとえば、1年おき）。
- ネットワーク障害によって、マスター表をベースとする1つ以上のスナップショット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、スナップショットが削除されたことをマスターが認識できない場合。

スナップショット・ログからの行の削除

スナップショット・ログが使用するデータベース領域は、常にできる限り小さくなるようにしてください。スナップショット・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ために、次の処理を実行できます。

- ログに対応付けられたスナップショットをリフレッシュする。これで、スナップショット・ログから行が削除されます。
- n 番前にリフレッシュされたスナップショットだけ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削除する。

スナップショット・ログから行を手動で削除するには、ログを含むデータベースで、DBMS_SNAPSHOT パッケージの PURGE_LOG ストアド・プロシージャを実行します。たとえば、一番最後にリフレッシュされたスナップショットだけに必要な項目を CUSTOMERS 表のスナップショット・ログから削除するには、次のプロシージャを実行します。

```
DBMS_SNAPSHOT.PURGE_LOG(
  master => 'sales.customers',
  num    => 1,
  flag   => 'DELETE');
```

追加情報：DBMS_SNAPSHOT.PURGE_LOG プロシージャのパラメータは、表 9-209 で説明します。

必要な権限

スナップショット・ログの所有者または DBMS_SNAPSHOT パッケージの EXECUTE 権限を持つユーザーだけが、PURGE_LOG プロシージャを実行してスナップショット・ログから行を削除できます。

スナップショット・ログの切捨て

スナップショット・ログが大きくなり、多数のエクステン트가割り当てられた場合、その行のログを削除しても、ログに割り当てられた領域の量は減少しません。スナップショット・ログに割り当てられた領域を削減するには、次の処理を実行します。

1. マスター表に排他ロックを設定して、その後のプロセス中に更新が発生しないようにする。

```
LOCK TABLE sales.customers IN EXCLUSIVE MODE;
```

2. 2 番目のデータベース・セッションで、スナップショット・ログ (MLOG\$ 実表など) の行を一時表にコピーする。

```
CREATE TABLE sales.templog AS SELECT * FROM sales.mlog$_customers;
```

3. 2 番目のセッションで、SQL コマンド TRUNCATE を使用してログを切り捨てる。

```
TRUNCATE sales.mlog$_customers;
```

4. 2 番目のセッションで、元の行を再挿入する。この作業によって、依存するスナップショットの完全リフレッシュを実行する手間が省けます。

```
INSERT INTO sales.mlog$_customers SELECT * FROM sales.templog;  
DROP TABLE sales.templog;
```

5. 最初のセッションで、マスター表の排他ロックを解放する。

```
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に対して行われた変更は、次の完全リフレッシュ時まで反映されません。

必要な権限

スナップショット・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が、スナップショット・ログを切り捨てることができます。

スナップショット・ログがあ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、表を定期的に再編成 ("reorg") できます。この項では、マスター表の再編成および対応するスナップショットの高速リフレッシュ機能の維持方法を説明します。

再編成通知

表を再編成するときに、スナップショット・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だけ自動的に検出されます。2-32 ページの「表 t の再編成の方法 2」を参照してください。

表が切り捨てられない場合は、Oracle に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には、2 つのプロシージャ DBMS_SNAPSHOT.BEGIN_TABLE_REORGANIZATION および DBMS_SNAPSHOT.END_TABLE_REORGANIZATION で、指定した表が再編成されたことを通知します。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、表のスナップショット・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表の所有者と名前です。出力はありません。

マスター表の切捨て

表が切り捨てられると、スナップショット・ログも切り捨てられます。しかし、主キー・スナップショットの場合、スナップショット・ログを切り捨てずに、高速リフレッシュを継続できます。スナップショット・ログに格納されている情報は保存されますが、マスター表が切り捨てられるとスナップショット・ログは ROWID に関して無効になります。スナップショット・ログ内の ROWID 情報は、新しく作成されたとみなされ、高速リフレッシュ用に ROWID スナップショットでこの情報を使用することはできません。

PRESERVE SNAPSHOT LOGオプションが指定されている場合、またはオプションが何も指定されていない場合は、マスター表のスナップショット・ログ内の情報は保存されますが、現行のROWIDスナップショットは完全リフレッシュが実行された後でしか高速リフレッシュ用にログを使用できません。これがデフォルトです。

注意：以前に高速リフレッシュが可能だったスナップショットの高速リフレッシュを確実に可能なままにするためには、2-31 ページの「データベースの表の再編成方法」のガイドラインに従ってください。

PURGE SNAPSHOT LOG オプションが指定されている場合は、スナップショット・ログはマスター表とともに削除されます。

例

次の2つの文は、マスター表が切り捨てられたときにスナップショット・ログ情報を保存します。

```
TRUNCATE TABLE tablename PRESERVE SNAPSHOT LOG;
TRUNCATE TABLE tablename;
```

次の文は、マスター表とともにスナップショット・ログを切り捨てます。

```
TRUNCATE TABLE tablename PURGE SNAPSHOT LOG
```

データベースの表の再編成方法

高速リフレッシュを保ったまま表を再編成できる方法が4つ提供されています。これらの方法については、以降の項目で説明します。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しなければなりません。

注意：マスター表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スナップショットで使われるログ情報を無効にすることがあります。)

表 t の再編成の方法 1

1. 表 t に対して dbms_snapshot.begin_table_reorganization をコールする。
2. 表の名前を t から t_old に変更する。
3. select * from t_old として表 t を作成する。
4. 新しい表 t に対して dbms_snapshot.end_table_reorganization をコールする。

・警告：表が改名されると、関連付けられている PL/SQL トリガーも表の新規名に合わせて調整されます。

・ dbms_snapshot.begin_table_reorganization のコールと dbms_snapshot.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 t の再編成の方法 2

1. 表 t に対して `dbms_snapshot.begin_table_reorganization` をコールする。
2. 表 t をエクスポートする。
3. PRESERVE SNAPSHOT LOG オプションで表 t を切り捨てる。
4. 従来型パスを使用して表 t をインポートする。
5. 新しい表 t に対して `dbms_snapshot.end_table_reorganization` をコールする。

・ 警告：再編成中にマスター表を切り捨てる場合、切り捨てる表のデータ定義言語の PRE-SERVE SNAPSHOT LOG 句を使用する必要があります。

・ `dbms_snapshot.begin_table_reorganization` のコールと `dbms_snapshot.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 t の再編成の方法 3

1. 表 t に対して `dbms_snapshot.begin_table_reorganization` をコールする。
2. 表 t をエクスポートする。
3. 表の名前を t から t_old に変更する。
4. 従来型パスを使用して表 t をインポートする。
5. 新しい表 t に対して `dbms_snapshot.end_table_reorganization` をコールする。

・ 警告：表が改名されると、関連付けられている PL/SQL トリガーも表の新規名に合わせて調整されます。

・ `dbms_snapshot.begin_table_reorganization` のコールと `dbms_snapshot.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 t の再編成の方法 4

1. 表 t に対して `dbms_snapshot.begin_table_reorganization` をコールする。
2. 表 t の中身を検索してフラット・ファイルに格納する。
3. 表の名前を t から t_old に変更する。
4. t_old と同型で表 t を作成する。
5. 従来型パスを使用して SQL*Loader を実行する。

6. 新しい表 t に対して `dbms_snapshot.end_table_reorganization` をコールする。

・ 警告：表が改名されると、関連付けられている PL/SQL トリガーも表の新規名に合わせて調整されます。

・ `dbms_snapshot.begin_table_reorganization` のコールと `dbms_snapshot.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。

スナップショット・ログの削除

スナップショット・ログは、マスター表または既存のスナップショットに関係なく削除できます。たとえば、次の状況の1つに該当する場合、スナップショット・ログを削除できます。

- ・ マスター表のすべてのスナップショットが削除された場合。
- ・ マスター表のすべてのスナップショットで、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。

スナップショット・ログを削除するには、Schema Manager を使用してログを削除するか、または同等の `DROP SNAPSHOT LOG ON sales.customers;` を使用できます。

```
DROP SNAPSHOT LOG ON sales.customers;
```

必要な権限

マスター表の所有者または `DROP ANY TABLE` システム権限を持つユーザーだけが、スナップショット・ログを削除できます。

読み込み専用スナップショットの管理

以降の項では、基本レプリケーション環境の読み込み専用スナップショットの作成および管理方法を説明します。次のトピックが含まれています。

- ・ 読み込み専用スナップショットの使用方法
- ・ マスター・サイトでのスナップショットの登録
- ・ 読み込み専用スナップショットの記憶領域パラメータの変更
- ・ 読み込み専用スナップショットの削除
- ・ 媒体障害からの読み込み専用スナップショットの回復

読み込み専用スナップショットの使用方法

アプリケーションでは、表またはビューと同様にスナップショットを問い合わせることができます。たとえば、次の問合せでは、CUSTOMERS という名前のローカル・スナップショットを参照します。

```
SELECT * FROM sales.customers;
```

読み込み専用スナップショットの使用中は、INSERT または UPDATE、DELETE 文を発行できません。発行すると、エラーが戻されます。

注意：INSERT および UPDATE、DELETE 文をスナップショットの実表に対して発行することはできますが、これらの処理はスナップショットを破壊します。アプリケーションではマスター表だけを更新して、最新に近いマスター表の状態がスナップショットに反映されるようにリフレッシュが行われるようにしてください。アプリケーションで、リモートのマスター表ではなくローカル・スナップショットを更新しなければならないときは、更新可能スナップショットを使用するアドバンスド・レプリケーション・システムを構成する必要があります。詳細は、第4章「スナップショット・サイト・レプリケーションの使用法」を参照してください。

注意：管理者は、いかなる方法でも読み込み専用スナップショットの構造を変更しないようにしてください。たとえば、スナップショットの実表に、トリガーまたは整合性制約を追加しないでください。

必要な権限

スナップショットに対して問合せを実行するには、スナップショットを所有している、またはスナップショットに対する SELECT 権限がある、SELECT ANY TABLE システム権限がある、のいずれかの条件を満たしていなければなりません。

スナップショットにアクセスする必要があるすべてのユーザーに、スナップショットに対する SELECT 権限を付与してください。スナップショットを所有するアカウントを使用したスナップショット・データベースへの接続は、許可しないでください。許可した場合、ユーザーはプライベート・データベース・リンクを使用して、なんらかの方法でリモートのマスター表にアクセスできてしまいます。

スナップショットに基づくビューおよびシノニムの作成

オプションで、スナップショットを基にビューおよびシノニムを作成できます。たとえば、次の文は、EMP スナップショットに基づくビューを作成します。

```
CREATE VIEW sales_dept AS
SELECT ename, empno
FROM scott.emp
WHERE deptno = 10;
```

マスター・サイトでのスナップショットの登録

マスター・サイトで、マスター表から作成されたスナップショットに関する情報が Oracle データベースに自動的に登録されます。以降の項では、スナップショットの登録メカニズムについて詳しく説明します。

登録済スナップショットに関する情報の参照

Schema Manager の「スナップショット・ログ」プロパティ・シートの「一般」ページを使用すると、スナップショット・ログに対応付けられたスナップショットを表示できます。その他の情報については、DBA_REGISTERED_SNAPSHOTS データ・ディクショナリ・ビューを使用すると、リモート・スナップショットに関する次の情報が表示されます。

- 所有者および名前、スナップショットを含むデータベース

- スナップショットの定義問合せ
- リフレッシュ方法（高速または完全）など、その他のスナップショットの特性

DBA_REGISTERED_SNAPSHOT ビューをマスター・サイトの DBA_SNAPSHOT_LOGS ビューと結合して、各スナップショットの最終リフレッシュ時刻を取得することもできます。管理者は、この情報を使用してマスター・サイトからスナップショット・アクティビティを監視し、マスター表の削除または変更、再配置が必要な場合にスナップショット・サイトへの変更を調整できます。

内部メカニズム

スナップショットを作成すると、そのマスター・データベースに読み込み専用スナップショットが自動的に登録され、スナップショットを削除すると登録解除されます。

注意：Oracle7 のマスター・サイトでは、スナップショットは登録できません。

注意：スナップショットの作成または削除時のマスター・サイトでのスナップショットの登録または登録解除は保証されていません。作成時にスナップショットが正常に登録されなかった場合は、次のスナップショットのリフレッシュ時にスナップショットが登録されます。スナップショットの削除時にスナップショットが正常に登録解除されなかった場合は、手動で登録解除するまで登録情報はマスター・データベースに残ります。

スナップショットの手動登録

必要ならば、登録は手動でメンテナンスできます。マスター・サイトで DBMS_SNAPSHOT パッケージの REGISTER_SNAPSHOT および UNREGISTER_SNAPSHOT プロシージャを使用して、スナップショットの登録情報を追加または修正、削除します。

追加情報：REGISTER_SNAPSHOT プロシージャおよび UNREGISTER_SNAPSHOT プロシージャについては、それぞれ表 9-193 と表 9-196 を参照してください。

読み込み専用スナップショットの記憶領域パラメータの変更

データ・クラスタ内にないスナップショットの記憶領域パラメータを変更するには、Schema Manager または SQL コマンド ALTER SNAPSHOT を使用できます。

```
ALTER SNAPSHOT sales.customers  
STORAGE (NEXT 500K);
```

必要な権限

スナップショットの記憶領域パラメータを変更するには、そのスナップショットがユーザーのスキーマになければなりません。または、ALTER ANY SNAPSHOT システム権限が必要です。

読み込み専用スナップショットの削除

スナップショットは、マスター表またはスナップショット・ログに関係なく削除できます。ローカル・スナップショットを削除するには、Schema Manager または SQL コマンド DROP SNAPSHOT を使用します。

```
DROP SNAPSHOT sales.customers;
```

あるマスター表の唯一のスナップショットを削除する場合は、そのマスター表のスナップショット・ログ（存在する場合）も削除してください。

スナップショットのマスター表の削除

マスター表を削除すると、対応するスナップショット・ログが存在する場合は自動的に削除されます。マスター表を削除しても、対応するスナップショットはすべて残り、引き続きアクセス可能な状態が保たれます。ただし、存在しないマスター表をベースとするスナップショットをリフレッシュしようとする、エラーが戻されます。

その後マスター表を再作成する場合、スナップショットの定義問合せをその新しいマスター表に対して正常に発行できれば、スナップショットを再び正常に完全リフレッシュできます。ただし、新しいマスター表用にスナップショット・ログを作成するまでは、スナップショットの高速リフレッシュは実行できません。マスター表を削除して再作成した後、スナップショットを正常にリフレッシュできない場合は、一度そのスナップショットも削除して、再作成してください。

スナップショットの削除に必要な権限

DROP ANY SNAPSHOT システム権限を持つユーザーまたはスナップショットの所有者だけが、スナップショットを削除できます。

媒体障害からの読み専用スナップショットの回復

媒体の障害が発生した場合は、スナップショットのマスター表を含むデータベースまたはスナップショットを含むデータベースを回復させる必要がある場合があります。マスター・データベースが独立的に過去のある時点まで回復した（つまり、対応する時間ベースの分散データベース回復が実行されなかった）場合、障害が起こっていた期間にリフレッシュしたリモート・スナップショットは、マスター表との整合性が失われます。このような場合、マスター・データベースの管理者は、リモート側の管理者に整合性が失われたスナップショットの完全リフレッシュを実行するよう指示してください。

追加情報：媒体の障害から回復する方法は、『Oracle8 Server バックアップおよびリカバリ』を参照してください。

スナップショット・リフレッシュとリフレッシュ・グループの管理

以降の項では、Oracle データベースのスナップショットのリフレッシュ・グループの管理方法を説明します。次のトピックが含まれています。

- リフレッシュ・グループの変更
- リフレッシュ・グループの削除
- スナップショット・リフレッシュ・グループの手動リフレッシュ
- リフレッシュの問題のトラブルシューティング

リフレッシュ・グループの変更

リフレッシュ・グループを作成した場合、いくつかの方法でグループを変更できます。以降の項では、リフレッシュ・グループへの新しいメンバー・スナップショットの追加、およびリフレッシュ・グループからのメンバーの削除、リフレッシュ・グループの自動リフレッシュ間隔の変更方法を説明します。

リフレッシュ・グループへのメンバーの追加

リフレッシュ・グループにメンバー・スナップショットを追加するには、Schema Manager の「リフレッシュ・グループ」プロパティ・シートの「スナップショット」ページを使用するか、または DBMS_REFRESH.ADD プロシージャをコールします。

```
DBMS_REFRESH.ADD
  name => 'sales_refgrp_1',
  list => 'sales.order_lines'
);
```

注意: 1 つのリフレッシュ・グループ内のスナップショットの最大数は 100 です。

リフレッシュ・グループからのメンバーの削除

リフレッシュ・グループからメンバー・スナップショットを削除するには、Schema Manager の「リフレッシュ・グループ」プロパティ・シートの「スナップショット」ページを使用するか、または DBMS_REFRESH.SUBTRACT プロシージャをコールします。

```
DBMS_REFRESH.SUBTRACT
  name  => 'sales_refgrp_1',
  list  => 'sales.assignments'
);
```

リフレッシュ・グループからスナップショットを削除した後は、別のリフレッシュ・グループに追加するか、または手動でリフレッシュしない限り、そのスナップショットはリフレッシュされません。

グループのリフレッシュ設定の変更

リフレッシュ・グループの次回リフレッシュを実行する日付または自動リフレッシュ間隔を変更するには、Schema Manager の「リフレッシュ・グループ」プロパティ・シートの「一般」ページを使用するか、または DBMS_REFRESH.CHANGE プロシージャをコールします。

```
DBMS_REFRESH.CHANGE
  name          => 'sales_refgrp_1',
  next_date     => SYSDATE,
  interval      => 'SYSDATE + 1'
);
```

リフレッシュ・グループの削除

リフレッシュ・グループを削除して、グループからすべてのメンバー・スナップショットを暗黙的に削除するには、Schema Manager を使用するか、DBMS_REFRESH.DESTROY プロシージャをコールします。

```
DBMS_REFRESH.DESTROY( name => 'sales_refgrp_1');
```

リフレッシュ・グループを削除した後は、そのグループの以前のメンバー・スナップショットは、別のリフレッシュ・グループに追加するか、または手動でリフレッシュしない限りリフレッシュされません。

スナップショット・リフレッシュ・グループの手動リフレッシュ

リフレッシュ・グループを使用すると、レプリケーション環境でのスナップショットの管理を容易にする自動リフレッシュ設定を構成できます。ただし、状況によっては、スナップショットを手動でリフレッシュする必要があります。たとえば、大量のデータをマスター表にロードした直後は、依存するスナップショットにマスター表のデータは反映されていません。次回実行予定の自動グループ・リフレッシュを待たずに、対応付けられたスナップショットにマスター表の新規行をすぐに反映するために、依存するスナップショット・グループを手動でリフレッシュすることもできます。

スナップショットのグループを手動でリフレッシュするには、Schema Manager の「リフレッシュ・グループ」プロパティ・シートの「一般」ページの「リフレッシュ」ボタンを使用するか、DBMS_REFRESH.REFRESH プロシージャをコールします。

```
DBMS_REFRESH.REFRESH('sales_refgrp_1');
```

手動でリフレッシュ・グループをリフレッシュしても、そのグループの次の自動リフレッシュには影響しません。

リフレッシュの問題のトラブルシューティング

以降の項では一般的なスナップショット・リフレッシュの問題を説明します。

一般的な問題

次のようないくつかの一般的な要因で、スナップショットのグループの自動リフレッシュが実行されない場合があります。

- スナップショット・データベースの SNP バックグラウンド・プロセスが起動していない
- ネットワークまたはサーバー障害の影響
- サーバー・シャットダウンの影響

スナップショット・リフレッシュ・グループで問題が発生したときに、上記の状況が原因かどうかを確認してください。

自動リフレッシュの再試行

グループの自動リフレッシュが失敗すると、グループはリフレッシュ実行予定のままになります。次の条件に基づいて、グループの自動リフレッシュが再試行されます。

- グループのリフレッシュを初回は1分後、次は2分後、その次は4分後というように、失敗するたびに再試行間隔を倍にする。
- 再試行間隔はリフレッシュ間隔自体より長い時間にはならない。
- 最高16回まで自動リフレッシュを再試行する。

リフレッシュ・グループのリフレッシュを16回再試行してもエラーが発生する場合は、グループは破損しているとみなされます。Schema Managerの「リフレッシュ・グループ」プロパティ・シートの「一般」ページを参照すると、いつリフレッシュ・グループが破損したかがわかります。USER_REFRESH および USER_REFRESH_CHILDREN データ・ディクショナリ・ビューのBROKEN列を問い合わせるリフレッシュ・グループの現在の状態を知ることもできます。

スナップショット・リフレッシュ・グループが破損しているとみなされる原因となったエラーは、トレース・ファイルに記録されます。リフレッシュ・グループが正常にリフレッシュされない原因となった問題を修正し、グループを手動でリフレッシュする必要があります。この結果、破損フラグがリセットされ、自動リフレッシュが実行できるようになります。

追加情報: スナップショット・トレース・ファイルの名前の形式はSNPnで、nはプラットフォーム固有のものです。各システムにおける名前は、プラットフォーム固有のOracle マニュアルを参照してください。

スナップショットのリフレッシュの継続的実行

スナップショットのグループのリフレッシュが継続的に繰り返されている場合、グループのリフレッシュ間隔をチェックしてください。グループの自動リフレッシュ間隔は、リフレッシュを開始する前に評価されます。グループのリフレッシュ間隔がグループのすべてのスナップショットのリフレッシュに要する時間よりも短い場合、SNP バックグラウンド・プロセスで未処理ジョブのキューがチェックされるたびに、グループ・リフレッシュが繰返し開始されることになります。

スナップショット・ログの無制限の増大

スナップショット・ログが無制限に増大する場合は、ネットワークまたはサイトの障害が原因で、スナップショットが削除されたことをマスターが認識できなくなっているかをチェックしてください。2-29 ページの「スナップショット・ログからの行の削除」に記述されているように、ログの一部を削除する必要があります。

スナップショットのパフォーマンスのチューニング

以降の項では、スナップショットのパフォーマンスのチューニング方法を説明します。

スナップショットの索引作成

スナップショットを使用するアプリケーションの問合せパフォーマンスを最大にするために、スナップショットに索引を作成できます。スナップショットの列に索引を付けるには、スナップショットの対応する実表に索引を作成します。

注意: スナップショットに遅延不可 UNIQUE 制約を宣言しないでください。

副問合せスナップショットのチューニング

参照整合性制約（可能な場合）および、スナップショットの定義にある問合せの中で参照している等価結合列に一意でない索引を定義することにより、副問合せを使用するスナップショットの高速リフレッシュのパフォーマンスを改善することができます。スナップショットを作成する前に、必要なFOREIGN KEY制約を宣言しなければなりません。スナップショットが存在している間は、これらの制約を変更または削除しないでください。

コストベースのオプティマイザを使用する場合は、マスター表およびスナップショットの実表、スナップショット・ログ（マスター側）、更新可能スナップショット・ログ（スナップショット・サイト、更新可能な場合）に対して必ずANALYZEコマンドを実行してください。

推奨：一般に、ログは空または空に近いときにANALYZEコマンドを実行するのが最適です。

副問合せを持つ単純スナップショットで使用される、基礎になっているスナップショット表に、さまざまな数の索引が自動的に追加されます。これらの索引によって、マスター表内の行の変更がスナップショットのパーティション化条件に影響する場合に、リフレッシュ時に全表検索が行われずに済みます。

2-19 ページの「例 2: より優れた方法」を参照してください。ORDER_LINES スナップショットの例では、SNAP\$_ORDER_LINE 表の C_ID および O_ID、OL_ID の各列に一意でない索引があります。

注意：等価結合またはフィルタ処理を実行するために、制限型の副問合せを持つスナップショットで使用される列を頻繁に修正すると、リフレッシュ時のパフォーマンスが低下することがあります。

基本レプリケーションのその他のオプション

他にも特定の状況で役立つ基本レプリケーションの機能がいくつかサポートされています。

- 複合スナップショット
- ROWID スナップショット
- スナップショットの個別リフレッシュ

複合スナップショット

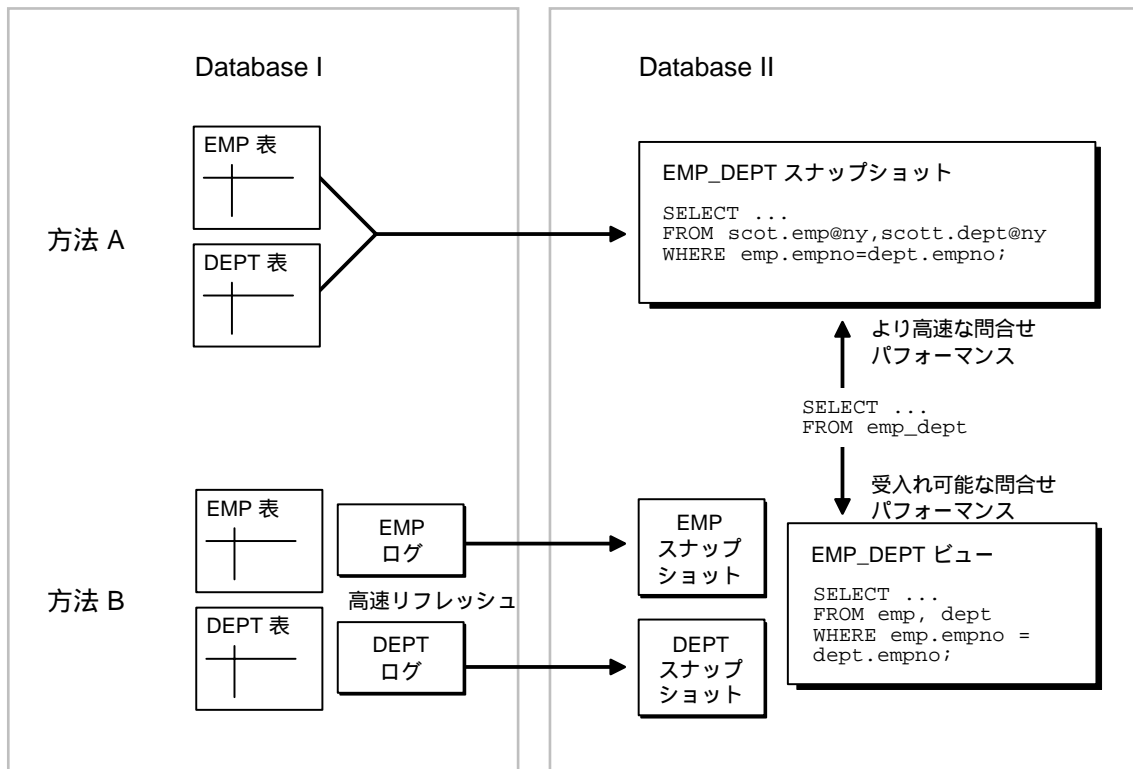
スナップショットの定義問合せに、個別関数や集約関数、またはGROUP BY句やCONNECT BY句、結合、制限型の副問合せ、集合演算が含まれるとき、スナップショットは複合スナップショットと呼ばれます。

注意：複合スナップショットの高速リフレッシュはできないため、複合スナップショットはできる限り使用しないでください。このため、複合スナップショットを使用すると、完全スナップショット・リフレッシュの実行中、ネットワークのパフォーマンスが低下することがあります。

単純スナップショットと複合スナップショットの比較

特定のアプリケーションでは、複合スナップショットの使用を考慮することもできます。図 3-1 および以降の項で、考慮すべき事項をいくつかまとめています。

図 2-4 単純スナップショットと複合スナップショットの比較



複合スナップショット

方法 A は、複合スナップショットを示しています。Database II 内のスナップショットは、スナップショットのリフレッシュ中にすでに結合操作が完了しているため、非常に効率的に問い合わせが行われます。ただし、複合スナップショットであるため、完全リフレッシュを実行しなければなりません。

結合ビューを持つ単純スナップショット

方法 B は、スナップショット内のデータベースで結合を実行するビューと、Database II 内の 2 つの単純スナップショットを示しています。ビューに対する問い合わせのパフォーマンスは、方法 A の複合スナップショットに対する問い合わせのパフォーマンスほど良くはありませんが、単純スナップショットの方がスナップショット・ログを使用してより効率的にリフレッシュできます。

どちらの方法を使用するかは、次のように要約できます。

- リフレッシュを実行する頻度が少なく、より速い問合せのパフォーマンスを優先する場合には、方法 A を使用する。
- 頻繁にリフレッシュを実行し、問合せのパフォーマンスを優先する必要がある場合は、方法 B を使用する。

複合スナップショットの記憶領域設定

複合スナップショットでは常に完全リフレッシュが実行されるので、効率を最大にするために、PCTFREE には 0、PCTUSED には 100 を設定します。

ROWID スナップショット

下位互換性のため、デフォルトの主キー・スナップショットの他に、ROWID スナップショットがサポートされています。ROWID スナップショットは、マスター表の行の物理的な行識別子 (ROWID) に基づいて作成されます。ROWID スナップショットは、Oracle7 データベースのマスター表のスナップショットにだけ使用し、Oracle データベースのマスター表の新規スナップショットを作成するときには使用しないでください。

ROWID スナップショットの作成

Oracle7、リリース 7.3 のマスター表の ROWID スナップショットを明示的に作成する場合は、CREATE SNAPSHOT 文に WITH ROWID 句を指定する必要があります。

```
CREATE SNAPSHOT ...  
  WITH ROWID  
  AS ...
```

スナップショット・ログと ROWID スナップショット

ROWID スナップショットの高速リフレッシュをサポートするには、CREATE SNAPSHOT LOG 文に WITH ROWID 句を指定してマスター表のスナップショット・ログを作成する必要があります。

マスター表の再編成と ROWID スナップショット

マスター表のすべての ROWID スナップショットは、表の再編成後、完全リフレッシュする必要があります。その後は、すべての ROWID スナップショットが高速リフレッシュされます。

スナップショットの個別リフレッシュ

一般的な環境では、スナップショットはすべてリフレッシュ・グループの一部としてリフレッシュされます。ただし、特定の環境では、スナップショット間の参照整合性を考慮する必要がなく、スナップショットを個別にリフレッシュする方が効率的な場合もあります。たとえば、特定のスナップショットを他のスナップショットよりも頻繁にリフレッシュすることもできます。

個別リフレッシュ用にスナップショットを構成するには、1 つのリフレッシュ・グループを 1 つのスナップショットに対して作成します。次の場合に、個々のスナップショットのリフレッシュ・グループが自動的に作成されます。

- Schema Manager でスナップショットを作成または変更して、リフレッシュ設定を指定した場合
- SQL コマンド CREATE SNAPSHOT を使用してスナップショットを作成し、REFRESH 句を指定した場合
- SQL コマンド ALTER SNAPSHOT を使用してスナップショットを変更し、REFRESH 句を指定した場合

基本レプリケーション環境の監視

Enterprise Manager の Schema Manager コンポーネントを使用すると、基本レプリケーション環境で、スナップショット・ログのプロパティおよびスナップショット、リフレッシュ・グループをすばやく参照できます。または、データベースのデータ・ディクショナリを問い合わせ、スナップショット・ログおよびスナップショット、リフレッシュ・グループに関する情報を参照することもできます。次のデータ・ディクショナリ・ビューでは、基本レプリケーション環境のコンポーネントに関する情報が提供されます。

- DBA_SNAPSHOTS
- DBA_SNAPSHOT_LOGS
- DBA_REFRESH、DBA_RGROUP、DBA_REFRESH_CHILDREN
- DBA_REGISTERED_SNAPSHOTS

注意：ほとんどのデータ・ディクショナリ・ビューには 3 つのタイプがあり、タイプによって異なる接頭辞、USER_* および ALL_*、SYS.DBA_* が付いています。

マルチマスター・レプリケーションの使用方法

この章では、マルチマスター・レプリケーションを使用するアドバンスド・レプリケーション・システムの構成方法および管理方法を説明します。アドバンスド・レプリケーションは Oracle8 Enterprise Edition でのみ使用可能です。Oracle[®]製品と Oracle8 Enterprise Edition との違いの詳細は、『Oracle8 と Oracle8 Enterprise Edition の解説』を参照してください。

この章のトピックは次のとおりです。

- クイック・スタート: マルチマスター・レプリケーション環境の作成
- マルチマスター・レプリケーションの設定
- スケジュール・リンクの管理
- サイトの遅延トランザクション・キューのバージ
- マスター・グループの管理
- アドバンスド・マルチマスター・レプリケーション・オプション 多数の発生するどのトランザクションも同一の小さい表を更新する状態は、できれば避けてください。たとえば、複数のトランザクションが 1 つの小さい表に読み込みと更新を行って主キーの順序番号生成をシミュレートするのは、優れたアプリケーション設計とは言えません。この設計の場合、すべてのトランザクションが同一のデータ・ブロックを更新しなければなりません。より優れた解決方法として、順序を作成し順序番号をキャッシュに格納すれば、主キーの生成を最適化しパラレル伝播のパフォーマンスを改善できます。

注意: この章では、デフォルトのレプリケーション・アーキテクチャ（非同期伝播を使用する行レベル・レプリケーション）を使用するマルチマスター・レプリケーション・システムの管理方法を説明します。プロシージャ・レプリケーションおよび同期データ伝播の構成方法の詳細は、第 7 章「上級テクニック」を参照してください。また、この章では、Oracle Replication Manager ツールを使用してマルチマスター・レプリケーション・システムを管理する方法の例を示します。参考として、等価のレプリケーション管理 API プロシージャを各項に記載しています。Oracle のレプリケーション管理 API の詳細は、第 9 章「レプリケーション管理 API リファレンス」を参照してください。

クイック・スタート：マルチマスター・レプリケーション環境の作成

マルチマスター・アドバンスド・レプリケーション環境を作成するには、少なくとも次のステップを行う必要があります。

1. アドバンスド・レプリケーション環境を設計する。複数のデータベースにレプリケートする表およびサポート・オブジェクトを決定し、レプリケーション・オブジェクトを適切なマスター・グループ内に編成します。
2. Replication Manager のセットアップ・ウィザードを使用して、マルチマスター・レプリケーションをサポートする多数のデータベースを構成する。Replication Manager セットアップ・ウィザードにより、マルチマスター・レプリケーション・システムをサポートするために必要なすべてのコンポーネントがすばやく構成されます。
3. マスター定義サイトへの Replication Manager のデータベース接続を使用して、1 つ以上のマスター・グループを作成して、表および関連するオブジェクトを複数のマスター・サイトにレプリケートする。
4. 各サイトのデータにアクセスするために必要な権限を、アプリケーション・ユーザーに付与する。

各ステップおよびその他のオプションの構成ステップの詳細は、この章の後述の項を参照してください。

簡単な例

以下に、マルチマスター・レプリケーション環境を作成するために必要なステップを、簡単な例で示します。

ステップ 1: 環境の設計

最初に、基本レプリケーション環境を設計します。この例では、マスター・サイト DBS1 および DBS2 の表 SCOTT.EMP と SCOTT.DEPT をレプリケートする方法を紹介します。DBS1 は、システムのマスター定義サイトとして指定されます。

注意：SCOTT.EMP の主キーは EMPNO 列で、SCOTT.DEPT の主キーは DEPTNO 列です。

ステップ 2: Replication Manager セットアップ・ウィザードの使用法

Replication Manager セットアップ・ウィザードは、マルチマスター・レプリケーション・システムのすべてのマスター・サイトのサポート・アカウントおよびリンク、スキーマ、スケジュール作成を構成するときに役に立ちます。この例では、セットアップ・ウィザードを使用して次のことを行います。

- マスター・サイト DBS1 および DBS2 を指定する。
- グローバルなレプリケーション管理者および伝播担当者、受信者のアカウントの役割をする、デフォルトの REPADMIN アカウントを各マスター・サイトに作成する。
- マルチマスター・システム全体で、指定されたレプリケーション・オブジェクトをサポートするスキーマ SCOTT を各マスター・サイトに作成する。

- 各マスター・サイトのデータ伝播およびパージのデフォルトのスケジュールを作成する（この例では、各マスター・サイトのスケジュール情報をカスタマイズする必要はありません）。

ステップ 3: マスター・グループの作成

マルチマスター・レプリケーション環境では、表および関連するレプリケーション・オブジェクトは、マスター・グループの一部としてレプリケートされます。マスター定義サイト DBS1へのデータベース接続を使用して、Replication Managerの「マスター・グループの作成」プロパティ・シートをオープンして、EMPLOYEEという名前の新規マスター・グループを作成します。プロパティ・シートのページで、グループのレプリケーション・オブジェクト SCOTT.EMP と SCOTT.DEPT、および他のマスター・サイト DBS2 を指定します。デフォルトでは、Replication Managerによりグループ内のすべてのオブジェクトのレプリケーション・サポートが生成され、グループのレプリケーション・アクティビティが再開されます。

ステップ 4: レプリケーション・オブジェクトへのアクセス権限の付与

マルチマスター・レプリケーション環境を構成したら、各サイトに接続するユーザーがそれを使用できるように、さまざまなレプリケーション・オブジェクトへのアクセス権限を付与します。

```
GRANT SELECT ON scott.emp TO ... ;
```

その他のステップ

この例では、特定の種類のマルチマスター・レプリケーション・システムを構成するために必要な場合があるいくつかのオプションのステップについては説明されていません。たとえば、アドバンスド・レプリケーションで共有データ・モデルを使用しているとき、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する前に、すべてのレプリケート表に対して競合の解消を構成する場合があります。マルチマスター・レプリケーション・システムの構成の詳細は、この章の以降の記述を参照してください。

マルチマスター・レプリケーションの設定

マルチマスター・アドバンスド・レプリケーション環境の作成を開始する前に、関連するデータベースに次の項目を設定する必要があります。

- レプリケーション管理者
- レプリケーション伝播担当者
- レプリケーション受信者
- データベース間通信を提供するデータベース・リンク

レプリケーション・セットアップ・ウィザード

デフォルトのマルチマスター・レプリケーション構成用のすべてのサイトの設定は、Replication Managerのレプリケーション・セットアップ・ウィザードを使用すると簡単なプロセスで行うことができます。指定する各マスター・サイトで、このウィザードにより次のステップが実行されます。

- レプリケーション管理者の役割をするデータベース・アカウントの作成。デフォルトでは、このアカウントはレプリケーション伝播担当者および受信者の役割もします。
- レプリケーション管理者 / 伝播担当者 / 受信者アカウントへの必要な権限の付与。
- 新しい管理者アカウントに対応する Replication Manager データベース接続の作成。
- 環境内のその他のすべてのマスター・サイトへのスケジュール・リンクの作成。
- システム内のすべてのマスター・サイトに対する遅延トランザクション・キューのバージ・スケジュール作成。

Replication Manager セットアップ・ウィザードを起動するには：

1. 「ファイル」をクリックする。
2. 「セットアップ・ウィザード」をクリックする。

以降の項では、Replication Manager セットアップ・ウィザードを使用して、マルチマスター・レプリケーション・システムにマスター・サイトを設定する方法を説明します。

マスター・サイトの作成

レプリケーション・セットアップ・ウィザードの最初のページで、実行するレプリケーション環境セットアップのタイプを指定するように要求されます。

1. 「マスター・サイトのセットアップ」を選択する。
2. 「次」をクリックする。

その次のページで、新しいマルチマスター・レプリケーション・システム内のマスター・サイトのリストを作成できます。この時点で、セットアップ・ウィザードで利用できる Replication Manager データベース接続がないことがよくあります。この場合、次のステップを実行します。

1. 「新規作成」をクリックする。
2. 表示される「新規マスター・サイト」ダイアログに、指定したシステムのマスター・サイトのグローバルなデータベース名と、そのサイトの SYSTEM アカウントのパスワードを入力する。（セットアップ・ウィザードでは、SYSTEM アカウントを使用して後続の構成タスクを実行します）。
3. 終了したら「追加」をクリックして、セットアップ・ウィザードのマスター・サイトのリストにそのサイトを追加する。
4. 各マスター・サイトについてステップ 2 を繰り返す。最後のサイトの情報の入力が終わったら、「OK」をクリックしてそのサイトを追加し、ダイアログを終了します。
5. マスター・サイトのリストを確認したら、「次」をクリックして処理を続行する。

レプリケーション管理者および伝播担当者、受信者のアカウントの作成

次のページで、各マスター・サイトのレプリケーション管理者および伝播担当者、受信者として機能するデータベース・アカウントの情報を指定できます。ウィザードにより、システム内のすべてのマスター・サイトに同じ名前とパスワードのアカウントが作成されます。

セットアップ・ウィザードでは、2つの異なるタイプのマスター・サイトにおけるアカウントのセットアップがサポートされています。

- デフォルトでは、管理者および伝播担当者、受信者の機能を実行できるアカウントが1つ作成される。アカウント名とパスワードを入力し、「次」を押して処理を続行します。
- また、伝播担当者/受信者アカウントとは別の独立したレプリケーション管理用アカウントを作成する場合は、「別の伝播担当者/受信者」をクリックし、各ユーザー・アカウント用の別のアカウント情報を入力する。それから、「次」を押して処理を続行します。

レプリケーション・オブジェクトを編成するスキーマの作成

次のページで、レプリケーション・オブジェクトを格納するスキーマとして作成するスキーマを指定できます。ウィザードにより、システム内のすべてのマスター・サイトに同じ名前とパスワードのスキーマが作成されます。

新規スキーマをリストに追加するには：

1. 「新規作成」をクリックする。
2. 表示される「新規レプリケート・オブジェクト・スキーマ」ダイアログに、レプリケーション・オブジェクトの格納に使用するスキーマ名と、そのスキーマのパスワードを入力する。終了したら「追加」をクリックして、セットアップ・ウィザードのスキーマのリストにそのスキーマを追加します。
3. 各スキーマについてステップ1および2を繰り返す。最後のスキーマの情報の入力が終わったら、「OK」をクリックして追加し、「新規レプリケート・オブジェクト・スキーマ」ダイアログを終了する。
4. スキーマのリストの確認が終わったら、「次」をクリックして処理を続行します。

スケジュール・リンクの作成

次のページで、システム内のすべてのマスター・サイトのデフォルトの伝播特性を指定できます。セットアップ・ウィザードでは、この情報を使用して各マスター・サイトから他のすべてのマスター・サイトへの対応するスケジュール・リンクを作成します。ウィザードのこのページの各設定の説明は、3-8ページの「スケジュール・リンクの作成」を参照してください。

デフォルトのスケジュール設定の確認が終わったら「次」をクリックして処理を続行します。

ページ・スケジュールの指定

次のページで、システム内の各マスター・サイトの遅延トランザクション・キューに対するデフォルトのページ・スケジュールを構成できます。ウィザードのこのページの各設定の説明は、3-11ページの「サイトの遅延トランザクション・キューのページ」を参照してください。

デフォルトのページ設定の確認が終わったら、「次」をクリックして処理を続行します。

各マスター・サイトのカスタマイズ

次のページでは、システム内の個々のマスター・サイトの設定をカスタマイズできます。システム内のマスター・サイトのカスタマイズを行わない場合、各サイトでは次の設定が同じになります。

- レプリケーション管理者および伝播担当者、受信者
- データベース・リンク仕様部
- スケジュールされた伝播設定およびパージ設定

マスター・サイトの設定をカスタマイズするには：

1. カスタマイズするターゲット・マスター・サイトを選択する。
2. 「カスタマイズ」をクリックする。

次に、「マスター・サイトのカスタマイズ」プロパティ・シートを使用して、ターゲット・マスター・サイトの次の設定をカスタマイズします。

- レプリケーション管理者または伝播担当者のアカウント（またはその両方）
- 管理者 / 伝播担当者 / 受信者データベース・リンクの USING 句
- 別のマスター・サイトへのスケジュールされた伝播設定

マスター・サイトのカスタマイズ設定を確認したら「OK」をクリックします。別のマスター・サイトの設定をカスタマイズするには、上記のプロセスを繰り返します。すべてのマスター・サイトのカスタマイズが終了したら、「次」をクリックして処理を続行します。

構成の調査および構築

次のページで、マルチマスター・アドバンスド・レプリケーション・システムの構成を終了するかどうか尋ねられます。終了する場合は、「終了」をクリックして、処理を続行します。この結果、設定をすばやく確認できる情報ダイアログが表示されます。

- すべて正しく設定されていたら、「OK」をクリックしてマルチマスター・レプリケーション環境を作成する。
- 設定に誤りがある場合は、「取消し」をクリックして「終了」ページに戻る。次に、「戻る」および「次」をクリックしてウィザードのページ内を移動し、設定を変更する。環境を構築する準備ができたなら、ウィザードの最終ページに戻る。

「終了」をクリックすると、マルチマスター・レプリケーション環境が作成されます。

注意：セットアップ・プロセス中に実行される API プロシージャのスクリプトを記録する場合は、システムを構築する前に「スクリプトを記録する」をクリックします。さらに、Replication Managerにより、現行の作業ディレクトリ内のファイルRepsetup.logに情報が記録されます。

終了後の作業

Replication Managerセットアップ・ウィザードを終了したら、次のステップを実行して構成を続行してください。

1. 各グループのマスター定義サイトとして機能するサイトにマスター・グループを作成する。マスター・グループの作成および管理の詳細は、3-13 ページの「マスター・グループの管理」を参照してください。
2. マスター定義サイトで、各マスター・グループのレプリケーション・オブジェクトを作成または指定する。レプリケーション・オブジェクトの作成および管理の詳細は、3-17 ページの「マスター・グループへのオブジェクトの追加」を参照してください。
3. 各マスター・グループのすべてのレプリケート表に対して競合の解消を構成する。レプリケーション・オブジェクトの競合の解消の構成の詳細は、第 5 章「競合の解消」を参照してください。
4. 各マスター・グループのレプリケーション・アクティビティを再開する。マスター・グループのレプリケーション・アクティビティの管理の詳細は、3-16 ページの「マスター・グループのレプリケーション・アクティビティの再開」を参照してください。
5. マルチマスター環境を監視し、システムが正常に動作していることを確認する。マスター・グループのアクティビティの監視の詳細は、6-4 ページの「アドバンスト・レプリケーション・システムの監視」を参照してください。

SNP バックグラウンド・プロセスの起動

管理を容易にするために、大部分のアドバンスト・レプリケーション環境ではデータ伝播が自動的に行われるように構成されます。このため、アドバンスト・レプリケーション環境の各マスター・サイトで、1 つ以上の SNP バックグラウンド・プロセスを起動する必要があります。各サーバーの SNP バックグラウンド・プロセスの設定は、次の初期化パラメータで制御されます。

- `JOB_QUEUE_PROCESSES` では、サーバーごとに `SNPn` バックグラウンド・プロセス番号を指定します。n は 0 ~ 9、A ~ Z を表します。このパラメータには 1 以上を設定します。大規模なシステムでないかぎり、通常はバックグラウンド・プロセスは 2 つか 3 つで十分です。
- `JOB_QUEUE_INTERVAL` で、サーバーのバックグラウンド・プロセスの起動の間隔を秒単位で指定します。一般的なレプリケーション環境では、デフォルトの 60 秒が適当です。

スケジュール・リンクの管理

スケジュール・リンクは、あるレプリケーション・サイトから別のレプリケーション・サイトにレプリケート・データを伝播するために必要です。マルチマスター・レプリケーション環境では、他の各マスター・サイトにデータを移送するために、各マスター・サイトにスケジュール・リンクが必要です。更新可能スナップショットを使用するスナップショット・サイトでも、対応するマスター・サイトにデータを移送するためにスケジュール・リンクが必要です。

特に、Replication Managerのセットアップ・ウィザードにより、必要なスケジュール・リンクで各マルチマスター・サイトまたはスナップショット・サイトの環境が設定されます。また、スケジュール・リンクをユーザーが管理できる機能もあります。以降の項では、スケジュール・リンクの管理について詳しく説明します。

スケジュール・リンクの作成

新規スケジュール・リンクを作成するには：

1. リンクを作成するサイトに接続するオープン・データベース接続内の任意の場所をクリックする。
2. 「新規作成」ツールバー・ボタンをクリックする。
3. 「新規スケジュール・リンク」をクリックする。

「新規スケジュール・リンクの作成」プロパティ・シートを使用して、新規リンクを作成します。以降の項では、このプロパティ・シートの「一般」および「オプション」ページで設定可能な項目について説明します。

「データベース・リンク」

新規スケジュール・リンクに使用するデータベース・リンク。「ブラウズ」をクリックして「スケジュール・リンクの設定」ダイアログを表示し、データベース・リンクを選択します。データベース・リンクはあらかじめ作成しておかなければなりません。

「次回日付」

新規接続先に変更内容を初めて送信する時間。「編集」をクリックして「日付の設定」ダイアログを表示し、「次回日付」フィールドの時間を設定します。

「間隔の式」

新規接続先に変更内容を送信する間隔。「編集」をクリックして「間隔の設定」ダイアログを表示し、「間隔」フィールドの時間を設定します。

「使用可能」

チェックすると新規スケジュール・リンクがすぐに使用可能になり、新規接続先に変更内容が送信される。

注意：リンクの作成時にターゲットの接続先が利用できない場合は、新規スケジュール・リンクを使用禁止にしてください。使用禁止にすると、利用できない接続先には変更内容は送信されません。

「パラレル伝播」

スケジュール・リンクにパラレル伝播（またはシリアル伝播）を使用するかどうかを指定する。チェックすると、リンクに対してパラレル伝播を設定できます。チェックしないと、新規スケジュール・リンクでシリアル伝播が使用されます。詳細は、3-27 ページの「パラレル伝播の計画」を参照してください。

「プロセス」

スケジュール・リンクで、接続先への情報のパラレル伝播に使用するバックグラウンド・プロセスの数。デフォルト値の 0 は、リンクにシリアル伝播を指定したことになります。0 より大きい値は、パラレル伝播で n 個のバックグラウンド・プロセスを使用することを示します。

「遅延秒数」

キューのポーリングを継続する時間（キューが空の場合も適用）。この設定の詳細は、3-9 ページの「スケジュール・リンクのガイドライン」を参照してください。

「バッチ・サイズ」(Oracle7 のデータベースのみ)

ローカル遅延トランザクション・キューを送信するときに、トランザクションをコミットする間隔を決定する。デフォルトの 0 は、リモートの接続先に各トランザクションが送信されるたびにコミットすることを示します。スケジュール・リンクにシリアル伝播を使用するときは、バッチ・サイズの設定を大きい値にすると 1 回の操作で複数の遅延トランザクションをコミットできるので、多数のトランザクションをコミットする場合のオーバーヘッドを削減できます。

「エラー時に停止」

ローカル遅延トランザクション・キューの送信中にエラーが発生した後の処理方法。デフォルトのチェックされていない状態は、ローカル遅延トランザクション・キューの伝播を継続することを示します。チェックすると、遅延トランザクションの実行は停止されます。

同等の API:

DBMS_DEFER_SYS.SCHEDULE_PUSH,DBMS_DEFER_SYS.SCHEDULE_EXECUTION

(Oracle7 のデータベースのみ)

スケジュール・リンクのガイドライン

スケジュール・リンクにより、あるマスター・サイトから別のマスター・サイトへ（あるスナップショット・サイトからマスター・サイトへ）の遅延トランザクション・キューの伝播方法が決まります。スケジュール・リンクを作成すると、遅延トランザクション・キューをシステムの別のサイトに送信するジョブがローカル・ジョブ・キューに作成されます。リモートのマスター・サイトに遅延トランザクションが伝播される場合は、レプリケーション伝播担当者のセキュリティ・コンテキスト内で行われます。また、スケジュール・リンクを構成すると、シリアル伝播またはパラレル伝播を使用して情報を送信できます。アドバンスド・レプリケーション・システムのスケジュール・リンクを作成する前に、どのようにシステム全体でレプリケーションをグローバルに発生させるかを慎重に検討します。

たとえば、ほぼリアルタイムのレプリケーションをシミュレートするために、各スケジュール・リンクにより、マスター・サイトの遅延トランザクション・キューをその接続先に絶えず送信する場合があります。または、一定の間隔で遅延トランザクションを伝播するときは、送信間隔および送信スケジュールを決定しなければなりません。必ず接続できるときまたは夜間などの通信コストが最低の時間帯にリンクをスケジュールする場合もあります。さらに、レプリケーションによるネットワーク・リソースへの負荷を分配するために、すべてのマスター・サイト間で、リンクのスケジュールが重複しないように設定することもできます。

連続的な送信のスケジュール

非同期レプリケーション・メカニズムを使用しているときでも、スケジュール・リンクを構成すると、連続的な即時レプリケーションをシミュレートできます。Replication Manager セットアップ・ウィザード、または「スケジュール・リンクの作成」プロパティ・シート、「スケジュール・リンクの編集」プロパティ・シートでスケジュール・リンクを構成するときに、「オプション」ページの「遅延秒数」を 500,000 に設定します。

定期的な送信のスケジュール

もう 1 つの方法として、サイトの遅延トランザクション・キューをリモートの接続先に定期的に送信するようにスケジュールできます。Replication Manager セットアップ・ウィザード、または「スケジュール・リンクの作成」プロパティ・シート、「スケジュール・リンクの編集」プロパティ・シートでスケジュール・リンクを構成するときに、「オプション」ページの「遅延秒数」をデフォルト値の 0 に設定します。次に、「一般」ページの「次回日付」および「間隔」を使用して、遅延トランザクションを送信する間隔を構成します。

シリアル伝播およびパラレル伝播の決定

スケジュール・リンクでは、シリアル伝播またはパラレル伝播のどちらかを使用してサイトの遅延トランザクション・キューを送信できます。シリアル伝播とパラレル伝播の詳細は、3-27 ページの「パラレル伝播の計画」を参照してください。

スケジュール・リンクの編集

スケジュール・リンクのリフレッシュ間隔または伝播特性を編集する、またはスケジュール・リンクを使用不可にするには：

1. ターゲットのスケジュール・リンクをクリックする。
2. 「プロパティ」ツールバー・ボタンをクリックする。

「スケジュール・リンクの編集」プロパティ・シートを使用して、スケジュール・リンクのプロパティを変更し、変更内容を適用します。スケジュール・リンクを調整できるプロパティの詳細は、3-8 ページの「スケジュール・リンクの作成」を参照してください。

同等の API: DBMS_DEFER_SYS.SCHEDULE_PUSH, DBMS_DEFER_SYS.SET_DISABLED, DBMS_DEFER_SYS.SCHEDULE_EXECUTION (Oracle7 のみ)

スケジュール・リンクの状態の表示

サイトのすべてのスケジュール・リンクの状態を表示するには、Replication Manager を使用します。

1. リンクを表示するサイトに接続しているオープン・データベース接続内の任意の場所をクリックする。
2. 「スケジュール」フォルダをクリックする。
3. 「リンク」フォルダをクリックする。

Replication Manager の詳細パネルには、各リンクの現在の状態（使用可能かどうか）など、サイトのすべてのスケジュール・リンクのリストが表示されます。

同等の API: DBMS_DEFER_SYS.DISABLED

スケジュール・リンクの削除

スケジュール・リンクを削除するには:

1. ターゲットのスケジュール・リンクをクリックする。
2. 「削除」ツールバー・ボタンをクリックする。

同等の API: DBMS_DEFER_SYS.UNSCHEDULE_PUSH

サイトの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がその接続先マスター・サイトに正常に送信されたら、サイトの遅延トランザクション・キューに残しておく必要はありません。適用済の遅延トランザクションをサイトの遅延トランザクション・キューから定期的にパージすると、キューは管理しやすいサイズに保たれます。Replication Manager セットアップ・ウィザードを使用してアドバンスド・レプリケーション・システムを構成すると、システムのすべてのマスター・サイトおよびスナップショット・サイトのパージが構成されます。サイトのパージ・スケジュールの設定には次の項目があります。

「次回日付」

ローカル遅延トランザクション・キューから適用済のトランザクションを次回パージする時間。「編集」をクリックして「日付の設定」ダイアログを表示し、「次回日付」フィールドの時間を設定します。

「間隔の式」

ローカル遅延トランザクション・キューから適用済のトランザクションを自動的にパージする間隔。「編集」をクリックして「間隔の設定」ダイアログを表示し、「間隔」フィールドの時間を設定します。

「ロールバック・セグメント」

ローカル遅延トランザクション・キューのパージを実行するときに使用するロールバック・セグメント。「ブラウズ」をクリックして「ロールバック・セグメントの選択」ダイアログを表示して、データベースのロールバック・セグメントを選択します。この設定を NULL 値にしておくと、遅延トランザクション・キューをパージするときにロールバック・セグメントを選択できます。

注意: ローカル遅延トランザクション・キューのパージで大量のロールバック・データが生成されることが予測されるときは、十分な大きさのロールバック・セグメントを確保してください。

「遅延秒数」

キューのポーリングを継続する時間 (キューが空の場合も適用)。スケジュールされたパージ間隔が比較的長いと、オーバーヘッドの削減に有効です。

同等の API:

DBMS_DEFER_SYS.SCHEDULE_PURGE,DBMS_DEFER_SYS.SCHEDULE_EXECUTION
(Oracle7 のみ)

遅延トランザクション・キューのスケジュール・ページのガイドライン

スケジュール・ページにより、マスター・サイトまたはスナップショット・サイトで、その遅延トランザクション・キューから適用済のトランザクションをページする方法が決まります。Replication Managerのセットアップ・ウィザードを使用してマスター・サイトまたはスナップショット・サイトを作成すると、一定の基準でローカル遅延トランザクション・キューをページするジョブが、各サイトのローカル・ジョブ・キューに作成されます。アドバンスド・レプリケーション・システムのサイトを構成する前に、ページ方法を慎重に検討してください。次に例を示します。

- サイトの遅延トランザクション・キューの送信（スケジュール・リンク）とページを同期化できます。たとえば、トランザクション・キューの送信とページを連続的に行うように構成することができます。このタイプの構成の場合、パフォーマンスが向上することがあります。これは、最後に送信されたトランザクションに関する情報が、対応するページ操作に対するサーバーのバッファ・キャッシュにすでに存在していることがよくあるためです。
- サーバーがCPU によるボトルネックの状態になっていない場合、遅延トランザクション・キューの連続的なページをスケジュールすると、可能な限りキューのサイズを小さくできます。
- 通常の業務時間帯はトランザクションのスループットが非常に大きくなるサーバーの場合、オフピーク時にページが行われるようにスケジュールできます。

連続的なページのスケジュール

Replication Manager セットアップ・ウィザード、または「DB 接続の編集」プロパティ・シートの「ページ・スケジュール」ページの使用時に、サイトの遅延トランザクション・キューの連続的なページを構成するには、「遅延秒数」を 500,000 に設定します。

定期的なページのスケジュール

もう1つの方法として、サイトの遅延トランザクション・キューを定期的にページするようにスケジュールできます。Replication Manager セットアップ・ウィザード、または「DB 接続の編集」プロパティ・シートの「ページ・スケジュール」ページを使用して、サイトのスケジュール・ページを構成するには、「遅延秒数」をデフォルト値の0に設定します。次に、「次回日付」および「間隔」の設定を使用して、遅延トランザクション・キューをページする間隔を構成します。

サイトのページ・スケジュールの指定

マスター・サイトまたはスナップショット・サイトを手動で構成する、またはサイトのページ・スケジュールを変更する場合は、「DB 接続の編集」プロパティ・シートを使用します。サイトのページ・スケジュールを編集するには：

1. サイトに対するデータベース接続をクリックする。
2. 「プロパティ」ツールバー・ボタンをクリックする。

「DB 接続の編集」プロパティ・シートの「ページ・スケジュール」を使用して、サイトのページ・スケジュールを変更し、変更内容を適用します。

同等の API:

DBMS_DEFER_SYS.SCHEDULE_PURGE,DBMS_DEFER_SYS.SCHEDULE_EXECUTION
(Oracle7 のみ)

サイトの遅延トランザクション・キューの手動ページ

マスター・サイトまたはスナップショット・サイトの遅延トランザクション・キューを手動でページするには、「DB 接続の編集」プロパティ・シートを使用します。サイトのページ・スケジュールを編集するには:

1. サイトに対するデータベース接続をクリックする。
2. 「プロパティ」ツールバー・ボタンをクリックする。
3. 「ページ・スケジュール」ページをクリックする。
4. 「即時ページ」をクリックする。

同等の API: DBMS_DEFER_SYS.PURGE

マスター・グループの管理

アドバンスド・レプリケーション・システムのマスター・サイトでは、レプリケーション・グループのすべてのオブジェクトの完全なコピーが管理されます。マスター・サイトのレプリケーション・グループは、特にマスター・グループと呼ばれます。Replication Managerにはマスター・グループを作成および管理するときに役に立つ多数の機能があります。

以降の項では、マスター・グループの管理方法について詳しく説明します。

マスター・グループの作成

アドバンスド・レプリケーションで新規マスター・グループを作成するには、Replication Manager の「マスター・グループの作成」プロパティ・シートを使用します。新規マスター・グループを作成するには:

1. 新規マスター・グループのマスター定義サイトに接続するオープン・データベース接続内の任意の場所をクリックする。新規マスター・グループのマスター定義サイトを選択するときは、常時最も使用可能なサイトを選択します。必要ならば、後でグループのマスター定義サイトを変更できます。
2. 「新規作成」ツールバー・ボタンをクリックする。
3. 「新規マスター・グループ」をクリックする。

「マスター・グループの作成」プロパティ・シートには、「一般」および「オブジェクト」、「接続先」の3つのページがあります。「オブジェクト」および「接続先」ページはオプションです。使用すると、マスター・グループの作成時にReplication Managerで追加の構成ステップを処理できます。

- グループのレプリケーション・オブジェクトなしに、かつマスター・グループをレプリケートするその他のマスター・サイトを登録せずにマスター・グループを作成するには、「一般」ページだけを設定し、オプションの「オブジェクト」および「接続先」ページは無視する。
- 作成時に新規マスター・グループにオブジェクトを追加する場合だけ、「オブジェクト」ページを使用する。マスター・グループにオブジェクトを追加するときは、いくつか考慮すべき事項があります。マスター・グループにレプリケーション・オブジェクトを追加する方法の詳細は、3-17 ページの「マスター・グループへのオブジェクトの追加」を参照してください。
- 作成時に新規マスター・グループに接続先のマスター・サイトを追加する場合だけ、「接続先」ページを使用する。マスター・グループにマスター・サイトを追加するときは、いくつか考慮すべき事項があります。レプリケーション・グループにマスター・サイトを追加する方法の詳細は、3-21 ページの「マスター・グループへのマスター・サイトの追加」を参照してください。

注意：新規マスター・グループの作成時に、Replication Manager により、グループおよび指定するレプリケーション・オブジェクトを作成するための補足情報を入力するよう要求されることがあります。たとえば、主キーが設定されていないレプリケート表といっしょに新規マスター・グループを作成すると、レプリケート表の代替の識別列または列のセットを指定できるように、「代替キー列の設定」ダイアログが表示されます。作成後、グループのレプリケーション・アクティビティを可能にするかどうかも尋ねられます。

警告：グループの作成時にマスター・グループに 1 つ以上の表を追加する場合、レプリケーション・アクティビティを再開していないことを確認してください。最初にレプリケーションの競合の可能性を考慮し、グループのレプリケート表に対して競合の解消を構成します。マスター・グループ・オブジェクトに対する競合解消方法の詳細は、第 5 章「競合の解消」を参照してください。

同等の API: DBMS_REPCAT.CREATE_MASTER_REPGROUP,DBMS_REPCAT.SET_COLUMNS

マスター・グループの接続修飾子の使用方法

接続修飾子を使用すると、同じリモート・データベースを示す複数のデータベース・リンクで、異なるパスを使用して接続を確立できます。たとえば、異なるパスを使用してリモート・データベースに接続する 2 つのパブリック・データベース・リンク DBS1 がデータベースにあるとします。

- DBS1@ETHERNET。イーサネット・リンクを使用して DBS1 に接続するリンク。
- DBS1@MODEM。モデム・リンクを使用して DBS1 に接続する別のリンク。

追加情報：データベース・リンクの接続修飾子の定義方法の詳細は、『Oracle8 Server 分散システム』の第 2 章を参照してください。

新規マスター・グループを作成するときに、グループに対応するすべてのスケジュール・リンクに対して接続修飾子を使用することを指定できます。ただし、マスター・グループに接続修飾子を使用すると、各マスター・サイトで接続修飾子を使用してデータベース・リンクを作成しないと情報が伝播されません。

たとえば、2つのマスター・サイト DBS1 と DBS2、および2つのマスター・グループ MG1 と MG2 があるマルチマスター構成の場合を考えてみます。グループ MG1 では接続修飾子 ETHERNET、グループ MG2 では接続修飾子 MODEM を使用します。この構成を実現するには：

- DBS1 データベースにデータベース・リンク DBS2@ETHERNET および DBS2@MODEM がなければならない。
- DBS2 データベースにデータベース・リンク DBS1@ETHERNET および DBS1@MODEM がなければならない。
- グループ MG1 の作成時に、接続修飾子 ETHERNET を使用することを指定する。
- グループ MG2 の作成時に、接続修飾子 MODEM を使用することを指定する。

注意：接続修飾リンクおよび複数のマスター・グループを使用するマルチマスター環境でトランザクションの整合性を保持するために、1つのトランザクションで複数グループのレプリケーション・オブジェクトを処理することはできません。

注意：接続修飾子を使用する場合は、おそらくすべてのマスター・サイトで初期化パラメータ OPEN_LINKS の値を増やす必要があります。デフォルトでは、プロセスごとのオープン・リンク数は4つです。用途に応じて必要な値を判断してください。パラメータ OPEN_LINKS の詳細は、『Oracle8 Server リファレンス・マニュアル』を参照してください。

マスター・グループの削除

あるマスター・グループをアドバンスド・レプリケーション環境のすべてのマスター・サイトから削除するには：

1. マスター定義サイトでターゲット・マスター・グループをクリックする。
2. 「削除」ツールバー・ボタンをクリックする。

同等の API: DBMS_REPCAT.DROP_MASTER_REPGROUP

マスター・グループのレプリケーション・アクティビティの中断

マスター・グループまたはそのレプリケーション・オブジェクトの管理操作がほぼ完了するまでは、すべてのマスター・サイトのマスター・グループのレプリケーション・アクティビティを中断するように要求されます。レプリケーション・アクティビティの中断は、マスター・グループの休止とも呼ばれます。

次の管理作業が完了するまでは、レプリケーション・アクティビティを中断してください。

- マスター・グループへのオブジェクトの追加
- マスター・グループのオブジェクトの定義の変更
- レプリケーション・オブジェクトのレプリケーション・サポートの再生成
- レプリケート表の競合の解消の追加または変更
- マスター・グループへのマスター・サイトの接続先の追加

- サイトからサイトへの伝播方法の変更

その他の状況でも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る場合があります。たとえば、予期しない競合（解消されなかったもの）が検出された場合、管理者はアクティビティを中断し、整合性を復元するためにマスター・グループの表のレプリカ上で問合せおよび更新を手動で実行しなければならない場合があります。

警告：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る管理作業を実行するには、マスター定義サイトでグループの状態が" 休止中 (quiescing) " になるまで待ちます。あるサイトに空でない遅延トランザクション・キューまたはレプリケーション・トリガーが存在すると問題が起きる場合があるので、作業前にグループの状態が" 休止状態 " になるまで待つ必要があります。

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するには：

1. グループのマスター定義サイトに接続するデータベース接続のターゲット・マスター・グループをクリックする。
2. 「プロパティ」ツールバー・ボタンをクリックする。
3. 「マスター・グループの編集」プロパティ・シートの「操作」ページをクリックする。
4. 「レプリケーションの中断」をクリックする。

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たら、マスター定義サイトのあらゆる管理操作を完了する前に、すべてのマスター・サイトでマスター・グループの状態を監視します。

注意：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するように要求すると、グループを" 休止 " する前に、最初にすべてのマスター・サイトの遅延トランザクション・キューが送信されます。プロセス中、Replication Manager によりグループの状態は" コールバック待機中 (Await Callback) " と表示されます。すべてのサイトでプロセスが完了すると、Replication Manager によりグループの状態は" 休止状態 " と表示されます。

同等の API: DBMS_REPCAT.SUSPEND_MASTER_ACTIVITY

マスター・グループのレプリケーション・アクティビティの再開

マスター・グループまたはそのレプリケーション・オブジェクトの管理操作が完了したら、すべてのマスター・サイトのマスター・グループのレプリケーション・アクティビティを再開できます。

注意：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する前に、グループの管理要求のステータスをチェックして、予期しないエラーが発生していないことを確認してください。

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するには：

1. グループのマスター定義サイトに接続するデータベース接続のターゲット・マスター・グループをクリックする。
2. 「プロパティ」ツールバー・ボタンをクリックする。

3. 「マスター・グループの編集」プロパティ・シートの「操作」ページをクリックする。
4. 「レプリケーションの再開」をクリックする。

マスター・グループのレプリケーション・アクティビティが再開したら、マスター・グループの状態を監視して、レプリケーション・アクティビティが正常に再開したことを確認します。

同等の API: DBMS_REPCAT.RESUME_MASTER_ACTIVITY

マスター・グループへのオブジェクトの追加

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すると、グループに追加する新規レプリケーション・オブジェクトを指定できます。マスター・グループの一部として表およびビュー、シノニム、索引、トリガー、プロシージャ、ファンクション、パッケージをレプリケートできます。マスター・グループに1つ以上のオブジェクトを追加するには:

1. マスター定義サイトの「データベース・オブジェクト」フォルダをオープンする。
2. 選択したデータベース・オブジェクトを、同じマスター定義サイトのターゲット・マスター・グループにドラッグアンドドロップする。

「グループへのオブジェクトの追加」ダイアログおよび「マスター・グループの編集」プロパティ・シートを使用して、マスター・グループにオブジェクトを追加することもできます。

警告: 生成オブジェクトの名前の競合を避けるために、レプリケート表の名前は 27 バイトを超えないようにしてください。また、マスター・グループのレプリケート表の PRIMARY KEY 制約および UNIQUE 制約に対応する索引を明示的にレプリケートしないでください。表制約の定義はすべて自動的にレプリケートされ、制約の施行に必要な索引はその後レプリケートされます。

マスター・グループにオブジェクトを追加するときは、次の管理操作も考慮する必要があります。

- すべてのマスター・サイトでオブジェクト定義（およびデータ）をどのようにレプリケートするか。このトピックの詳細は、3-18 ページの「オブジェクト定義のマスター・サイトへのレプリケート」を参照してください。
- 表をいつ追加し、すべてのマスター・サイトで表データをどのようにレプリケートするか。このトピックの詳細は、3-19 ページの「表データのマスター・サイトへのレプリケート」を参照してください。
- 表をいつ追加し、表の競合の解消をどのように構成するか。このトピックの詳細は、第 5 章「競合の解消」を参照してください。

同等の API: DBMS_REPCAT.CREATE_MASTER_REPOBJECT

レプリケート表の代替キーの指定

データ・レプリケーション時には別々のサイトで対応する行を個々に識別して、一致させることができなければなりません。通常、アドバンスド・レプリケーション機能では、表の主キーを使用して表の行が個々に識別されます。表に主キーがない場合は、代替キーを指定する必要があります。代替キーとは、データ・レプリケーション時に表の行を識別するために使用できる列または列のセットのことです。

警告：行を識別してレプリケート・データの整合性を保つことを保証するために、アプリケーションで表の識別列を更新できないようにする必要があります。

主キーのない表といっしょに新規マスター・グループを作成する、または主キーのない表をマスター・グループに追加しようとする、Replication Managerにより、「代替キー列の設定」ダイアログが自動的に表示されます。このダイアログで、レプリケート表の代替の識別列または列のセットを指定できます。

同等の API: DBMS_REPCAT.SET_COLUMNS

レプリケート表のデータ型の考慮

マルチマスター・レプリケーションでは、NUMBER、DATE、VARCHAR2、CHAR、NVARCHAR2、NCHAR、RAW、ROWIDというデータ型を使用する列を含む表のレプリケーションをサポートしています。

また、バイナリ型LOB (BLOB) および文字型LOB (CLOB)、各国文字型LOB (NCLOB) というラージ・オブジェクト型を使用する列を含む表のレプリケーションもサポートしています。部分単位の更新と追加がこれらのLOB列に適用されるとき、マルチ・マスターのレプリケーションに使用する遅延および同期リモート・プロシージャ・コール・メカニズムでは、サポートされるLOBデータ型に対する部分単位の変更だけを伝播します。

注意：Oracle8 では、一部のサイトがOracle7 リリース 7.3 を実行しているレプリケーション環境で、LOBデータ型のレプリケーションをサポートしていません。

Oracle では、LONGおよびLONG RAWデータ型を使用する列のレプリケーションはサポートしていません。これらのデータ型を含む列は、レプリケート表から単純に削除されます。

また、ユーザー定義オブジェクト型および外部LOB、ファイル・ベースLOB (BFILE) もサポートされていません。これらのデータ型の列を含む表をマスターとして構成しようとすると、エラー・メッセージが戻されます。

オブジェクト定義のマスター・サイトへのレプリケート

オブジェクトをマスター・グループに追加すると、Replication Manager により、「存在する場合既存のオブジェクトを使用する」かどうか尋ねられます。

オブジェクトの自動作成

デフォルトでは、マスター定義サイトのグループにオブジェクトを追加すると、オブジェクトの定義を使用して、すべてのマスター・サイトに同じオブジェクトが作成されます。このオプションを使用すると管理作業は軽減しますが、初期オブジェクト作成のためネットワーク・トラフィックが発生します。

注意：マスター・グループにパーティション表（または索引）を追加すると、他のすべてのマスター・サイトにも表のパーティションがレプリケートされます。マスター定義サイトの表領域と同じ名前の表領域がマスター・サイトにない場合、マスター・サイトのスキーマのデフォルトの表領域を使用して、レプリケート表のパーティションが作成されます。

オブジェクトの手動作成

マスター定義サイトのグループにオブジェクトを追加する前に、各マスター・サイトに同じオブジェクト定義を手動で作成できます。後で、グループにオブジェクトを追加するときに、既存のオブジェクトを使用できるため、各マスター・サイトでオブジェクトを作成せずに済みます。

レプリケーション・オブジェクトの手動作成は、大規模なレプリケーション環境を構成するときに、ネットワーク・トラフィックを最小にするために役立ちます。循環型依存を使用する表または自己参照型制約を含む特定の表がマスター・グループに含まれるときも、このオプションを考慮する必要がある場合があります。

レプリケーション・オブジェクトを事前作成するときは、次の事を考慮します。

- ユーザー自身でレプリケート表を事前作成するときは、"表の形式"（数および名前、表のすべての列のデータ型）がすべてのサイトで必ず同じになるようにしなければならない。マスター・サイトの表がマスター定義サイトの表と同じ形式ではない場合、エラーが戻されます。
- ユーザー自身で表ではないレプリケート・オブジェクトを事前作成するときは、オブジェクトのSQL定義がすべてのサイトで必ず同じになるようにしなければならない。マスター・サイトのオブジェクトにマスター定義サイトのオブジェクトと同じSQL定義がない場合、エラーが戻されます。

表データのマスター・サイトへのレプリケート

表をマスター・グループに追加すると、Replication Managerにより、"行データをコピーする"かどうか尋ねられます。

表データの自動レプリケート

デフォルトでは、マスター定義サイトのグループに表を追加すると、マスター定義サイトの表のデータが、すべてのマスター・サイトの表にレプリケートされます。このオプションを使用すると管理作業は軽減しますが、初期オブジェクト作成のためネットワーク・トラフィックが発生します。

表データの手動ロード

マスター定義サイトのグループに表を追加する前に、各マスター・サイトに同じ表構造を事前に作成して、各表レプリカに同じデータを手動でロードできます。後で、マスター定義サイトのグループにオブジェクトを追加したときに、既存の表レプリカを使用できるため、各マスター・サイトで表データを作成およびレプリケートせずに済みます。このオプションは、大きい表を構成していて、初期オブジェクト作成およびデータ・レプリケーションによるネットワーク・トラフィックを最小に抑えるときに適しています。

ユーザー自身でマスター・サイトのレプリケート表を作成するときは、システム内のすべてのレプリカ内で表データの一貫性が必ず保たれるようにしなければなりません。たとえば、データとともにレプリケート表を手動で作成するときは、表をマスター・グループに追加する前に行ってください。さらに、表がマスター・グループに追加され、レプリケーション・アクティビティが再開されるまでは、アプリケーションがレプリケート表にアクセスできないようにします。アクセスを許可すると、マスター・サイト間で、表の一貫性が失われる可能性があります。

オフライン・インスタンスエーション

現在大量のデータをレプリケートしていて、システムに新規サイトを追加する場合は、オフライン・インスタンスエーションを検討する必要があります。オフライン・インスタンスエーションの詳細は、7-13 ページの「スナップショット・クローニングとオフライン・インスタンスエーション」を参照してください。

マスター・グループのオブジェクトの変更

マスター・グループのレプリケーション・オブジェクトの定義を変更するときは、常に Replication Manager（または同等の API コール）を使用してください。Enterprise Manager または SQL の DDL コマンド（ALTER TABLE など）を使用してレプリケート環境のオブジェクトを直接変更すると、システム内のすべてのサイトのオブジェクトに DDL の変更内容が伝播されない場合があります。

注意：スナップショット・サイトまたはマスター・サイトでの個々のレプリカのローカルでのカスタマイズは、Oracle のアドバンスト・レプリケーション機能では行えません。レプリケーション管理者として、Replication Manager で実行されるグローバルなカスタマイズが、ローカルなカスタマイズにより妨げられることがないようにしてください。

マスター・グループのレプリケーション・アクティビティが正常に中断してから、次の手順でグループのオブジェクトの定義を変更します。

1. マスター定義サイトで、マスター・グループのターゲット・オブジェクトをクリックする。
2. 「プロパティ」ツールバー・ボタンをクリックする。
3. 「レプリケーション・オブジェクトの編集」プロパティ・シートの「一般」ページをクリックする。
4. 「レプリケーション・オブジェクトの変更」をクリックする。
5. 「レプリケーション・オブジェクトの変更」ダイアログを使用して、レプリケーション・オブジェクトの定義を変更する。たとえば、新規列をレプリケート表に追加できます。すべてのオブジェクト参照を必ず完全に修飾してください。

同等の API: DBMS_REPCAT.ALTER_MASTER_REPOBJECT

考慮事項

マスター・グループのオブジェクトの変更前または変更後には、次の事項を考慮してください。

- 依存する更新可能スナップショットがあるレプリケート表を変更するときは、その前にスナップショット・サイトからすべての変更内容を必ず送信する。
- レプリケーション・オブジェクトを変更した後に、オブジェクトのレプリケーション・サポートを再生成する必要がある場合がある。
- レプリケーション・オブジェクトを変更した後に、マスター定義サイトで管理要求をチェックして、各マスター・サイトでオブジェクトが正常に変更されたことを確認する。オブジェクトおよびサポート・オブジェクトに対するDDL変更は、各マスター・サイトで非同期で適用されます。
- 依存するスナップショットがあるレプリケート表を変更した後は、スナップショットを削除して再作成しなければならない。スナップショット・サイトにおける他のすべてのオブジェクトは、依存するスナップショットが次回リフレッシュされるとき自動的に再作成されます。

マスター・グループからのオブジェクトの削除

マスター・グループからオブジェクトを削除するには：

1. マスター定義サイトのターゲット・マスター・グループをクリックする。
2. グループから削除するターゲット・オブジェクトを選択する。
3. 「削除」ツールバー・ボタンをクリックする。

注意：マスター・グループからオブジェクトを削除する前に、そのオブジェクトに依存しているスナップショットがないことを確認してください。

注意：マスター・グループからレプリケーション・オブジェクトを削除すると、そのレプリケーション・オブジェクトのサポートに必要な対応するすべてのシステム生成オブジェクトが、Replication Manager により自動的に削除されます。

同等の API: DBMS_REPCAT.DROP_MASTER_REPOBJECT

マスター・グループへのマスター・サイトの追加

新規マスター・サイトをマスター・グループに追加する前に、次のことを行う必要があります。

- 必要なレプリケーション管理者および伝播担当者、受信者、スキーマ、既存のサイトへのリンクを新規マスター・サイトに設定する。
- 新規サイトを認識するように既存のサイトを変更する。

新規マスター・サイトを追加するためにマルチマスター・レプリケーション・システムを設定するには、Replication Manager セットアップ・ウィザードを使用します。セットアップ・ウィザードを使用するときは、次の事に注意します。

- システム内のすべてのマスター・サイト（すべての新規マスター・サイトとすでにシステムでサポートされているサイト）を指定しなければならない。システムにすでに存在しているマスター・サイトを指定するには、ウィザードの「マスター・サイトの作成」

ページの「ブラウズ」ボタンを使用します。新規マスター・サイトを指定するには、同じページの「新規作成」ボタンを使用します。

- SYSTEM アカウントのパスワードおよびシステムの既存の伝播担当者を入力するように要求される。
- オプションで、管理者または伝播担当者 / 受信者アカウント（あるいはその両方）、およびデフォルトとサイト固有の伝播設定など、マルチマスター環境のシステム設定を変更できる。

注意: マルチマスター構成にセットアップ・ウィザードを使用する方法の詳細は、3-3 ページの「レプリケーション・セットアップ・ウィザード」を参照してください。

セットアップ・ウィザードを使用して新規マスター・サイトを追加するようにマルチマスター・レプリケーション・システムを設定したら、新規マスター・サイトをグループに追加できます。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た後、新規接続先をマスター・グループに追加するには:

1. マスター定義サイトのターゲット・マスター・グループをクリックする。
2. 「プロパティ」ツールバー・ボタンをクリックする。
3. 「マスター・グループの編集」プロパティ・シートの「接続先」ページをクリックする。
4. 接続先マスター・サイトを1つ以上グループに追加する。新規マスター・サイトへの送信を試行するには、新規マスター・サイトへのスケジュール・リンクがすでに存在していなければなりません。

注意: 循環型依存がある表またはまたは自己参照型制約を含む特定の表があるマスター・グループにマスター・サイトを追加するときは、マスター・サイトで表を事前に作成して、新規サイトでデータを手動でロードする必要があります。詳細は、3-18 ページの「オブジェクト定義のマスター・サイトへのレプリケート」を参照してください。

同等の API: DBMS_REPCAT.ADD_MASTER_DATABASE

マスター・グループからのマスター・サイトの削除

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たら、グループから接続先（マスター・サイト）を削除できます。マスター・グループからマスター・サイトの接続先を削除するには:

1. マスター定義サイトのターゲット・マスター・グループをクリックする。
2. 「プロパティ」ツールバー・ボタンをクリックする。
3. 「マスター・グループの編集」プロパティ・シートの「接続先」ページをクリックする。
4. 接続先マスター・サイトを1つ以上グループから削除する。

同等の API: DBMS_REPCAT.REMOVE_MASTER_DATABASES

特別な状況

マスター・グループから削除されたサイトには、アクセスする必要はありません。システムまたはネットワーク障害のためにマスター・サイトが非常に長い時間使用できないときは、マスター・グループからマスター・サイトを削除することをお勧めします。ただし、サイトは利用できないため、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できないことがよくあります。この場合、次の処理を実行する必要があります。

- 遅延トランザクション・キューの消去
- データの非一貫性の除去

特に、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た後、利用できないマスター・サイトが削除されたら、できるだけ早く次のステップを次の順序で実行しなければなりません。

1. マスター・グループのレプリケーション・アクティビティを中断する。
2. トランザクションの接続先が削除済のマスター・サイトである遅延トランザクションを各マスター・サイトから削除する。
3. 削除済のマスター・サイトから遅延トランザクションをすべて削除する。
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する。
5. 残りの各マスターに遅延またはエラー・トランザクションがないことを確認する。残りのマスターに削除できない遅延トランザクションが1つでもある場合は、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。
6. すべてのレプリケート・データが矛盾していないことを確認する。DBMS_RECTIFIER_DIFF パッケージを使用して相違点を判別して修正します。
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する。

注意：利用できないマスター・サイトをマスター・グループから削除した後、サイトからもマスター・グループを削除して、クリーン・アップを完了しなければなりません。

マスター・グループ・オブジェクトのレプリケーション・サポートの生成

マスター・グループの管理操作を実行した後、グループのレプリケーション・アクティビティを再開する前に、Oracle により、変更に応じたレプリケーション・サポートが生成されなければなりません。たとえば、マスター・グループに表を追加したら、レプリケート表をサポートするには、\$RR および \$RP、\$RL パッケージが生成され、内部トリガーがアクティブにならなければなりません。後で表に競合の解消を追加するときは、すべてのマスター・サイトで表に対して同じ競合解消方法が使用されるように、表のレプリケーション・サポートを再生成する必要があります。

注意：レプリケーション・オブジェクトの状態を表示するには、オブジェクトを含むマスター・グループをクリックします。「状態」フィールドには、グループの各レプリケーション・オブジェクトの状態が表示されます。オブジェクトの状態が「有効」のときは、処理は必要ありませんが、オブジェクトの状態が「生成が必要」の場合は、オブジェクトのレプリケーション・サポートを生成する必要があります。

次の2つのフェーズを使用してオブジェクトのレプリケーション・サポートが生成されます。

1. オブジェクトのレプリケーション・サポートを生成すると、必要な生成パッケージ作成の要求が同期をとってサイトすべてに通知されて、フェーズ1が開始されます。これらのパッケージは非同期で作成されます。プロシージャ・レプリケーションの場合、フェーズ1でパッケージ仕様部が生成されます。
2. レプリケーションをサポートするために必要なパッケージを生成したことを各サイトがマスター定義サイトに示すまでは、フェーズ2は開始されません。その後、各サイトで必要な内部トリガーを起動する要求を同期をとって同報通信することによって、フェーズ2が開始されます（Oracle7 リリース 7.3 のサイトでは、同報通信要求が、必要な PL/SQL トリガーとこれに関連付けられたパッケージを生成します）。この場合も、これらのオブジェクトは非同期で作成されます。プロシージャ・レプリケーションの場合、フェーズ2でパッケージ本体が生成されます。

注意：Oracleは、各サイトでパッケージ作成の要求を同報通信した後で生成を再び要求し、続けてマスター・グループを作成できるように最適化されています。この種類の要求を処理するために、すべてのパッケージが実際に全サイトで作成されるのを待つ必要はありません。新しい管理要求は、レプリケーション・サポートを生成する第2フェーズが完了するまで実行されません。

個々のオブジェクトに対するレプリケーション・サポートの生成

マスター・グループの個々のオブジェクトに対するレプリケーション・サポートを生成するには：

1. マスター定義サイトで、マスター・グループのターゲット・オブジェクトをクリックする。
2. 「プロパティ」ツールバー・ボタンをクリックする。
3. 「レプリケーション・オブジェクトの編集」プロパティ・シートの「一般」ページをクリックする。
4. 「サポートの生成」をクリックして、ターゲット・オブジェクトのレプリケーション・サポートを生成する。

注意：1つ以上のオブジェクトのレプリケーション・サポートを生成したら、Replication Manager を使用してオブジェクトの状態をチェックすると、操作が正常に行われたか確認できます。

マスター・グループのすべての表に対するレプリケーション・サポートの生成

マスター・グループのすべての表に対するレプリケーション・サポートを生成するには：

1. マスター定義サイトでターゲット・マスター・グループをクリックする。
2. 「プロパティ」ツールバー・ボタンをクリックする。
3. 「マスター・グループの編集」プロパティ・シートの「操作」ページをクリックする。
4. 「生成」をクリックして、ターゲット・マスター・グループのすべての表に対するレプリケーション・サポートを生成する。

同等の API: DBMS_REPCAT.GENERATE_REPLICATION_SUPPORT

注意: 1 つ以上のオブジェクトのレプリケーション・サポートを生成したら、Replication Manager を使用してオブジェクトの状態をチェックすると、操作が正常に行われたか確認できます。

データ伝播の最小化

「レプリケーション・オブジェクトの編集」プロパティ・シートの「通信の最小化」設定により、表の競合検出を実行するためにサイトで転送しなければならないデータの量が決まります。この設定は、Oracle8 のデータベースでだけ有効で、グループのマスター定義サイトへのデータベース接続を使用しているときだけ使用できます。

注意: レプリケート環境に Oracle7 リリース 7.3 で実行されているマスター・サイトがある場合は、この設定は使用禁止にする必要があります。使用禁止にすると、Oracle は行内の任意の列が更新されたときに、行内のすべての列の古い値と新しい値を伝播します。これは、Oracle7 リリース 7.3 での動作です。

「通信の最小化」が使用可能な場合（デフォルト）、更新された列の新しい値、および更新された各列グループ内の主キーと列の古い値だけが伝播されます。

追加情報: データ伝播を最小にするその他の方法は、5-34 ページの「更新の競合解消のためのデータ伝播の最小化」を参照してください。

マスター・グループに関する情報の表示

Replication Manager で、アドバンスド・レプリケーション・システムのマスター・グループに関する情報を表示できます。

マスター・グループの表示

サイトのすべてのマスター・グループのリストを表示するには:

1. サイトの「構成」フォルダをオープンする。
2. その中の「マスター・グループ」フォルダをクリックする。

サイトの各マスター・グループについて、マスター・グループの名前、およびそのサイトがマスター定義サイトかどうか、グループの状態（通常または休止状態など）が詳細パネルに表示されます。

マスター・グループのオブジェクトの表示

サイトのマスター・グループのすべてのオブジェクトのリストを表示するには:

1. サイトの「構成」フォルダをオープンする。

2. その中の「マスター・グループ」フォルダをオープンする。
3. その中のマスター・グループをクリックする。

ターゲットのマスター・グループの各オブジェクトについて、オブジェクトの名前、およびオブジェクトを含むスキーマ、オブジェクトのタイプ（表および索引、プロシージャなど）、状態（有効またはレプリケーション・サポートの生成が必要、など）が詳細パネルに表示されます。

マスター・グループの接続先マップの表示

Replication Managerでは、接続先マップを使用してアドバンスド・レプリケーション環境のマスター・グループの構成を視覚的に表示します。マスター・サイトのマスター・グループの接続先マップを表示するには：

1. サイトの「構成」フォルダをオープンする。
2. その中の「マスター・グループ」フォルダをオープンする。
3. ターゲット・マスター・グループをオープンする。
4. ターゲット・マスター・グループの「接続先マップ」フォルダをクリックする。

マスター・グループの接続先マップでは、マスター・グループに関して次の視覚的な情報が提供されます。

- マスター・グループのマスター定義サイトおよびマスター・サイト
- 各マスター・サイト間の伝播方法

接続先マップでは、マスター・サイト間に表示されるスケジュール・リンクのプロパティを編集することもできます。接続先マップのリンクを編集するには、Replication Manager の「データベース接続先の編集」プロパティ・シートを使用します。ダイアログにアクセスするには、スケジュール・リンクをクリックして「入力」を押すか、リンクを右クリックして「プロパティ」をクリックします。

「データベース接続先の編集」プロパティ・シートを使用すると、次のことが実行できます。

- スケジュール・リンクの伝播モードの変更
- リンクの伝播間隔の再スケジュール

マスター・グループに関連する生成オブジェクトの表示

マスター・サイトのマスター・グループのレプリケーション・オブジェクトに関連する生成オブジェクトを表示するには：

1. サイトの「構成」フォルダをオープンする。
2. その中の「マスター・グループ」フォルダをオープンする。
3. ターゲット・マスター・グループをオープンする。
4. ターゲット・マスター・グループの「生成オブジェクト」フォルダをクリックする。

データ・ディクショナリ・ビュー

アドバンスト・レプリケーション環境に関する情報を参照するためにReplication Managerを使用する以外に、次のデータ・ディクショナリ・ビューを使用することもできます。

- REPGROUP
- REPSITES
- REPOBJECT
- REPCATLOG

マスター・サイト管理に関するその他の情報

この章のこれまでの項目では、最も一般的なマスター・グループの管理プロシージャについて説明しました。あまり一般的でないマスター・グループの管理プロシージャの詳細は、6-2ページの「マスター・グループおよびスナップショット・グループの上級管理テクニック」を参照してください。

アドバンスト・マルチマスター・レプリケーション・オプション

以降の項では、マルチマスター・レプリケーション・システムを作成および管理するときに考慮すべきその他のトピックについて説明します。

- パラレル伝播の計画
- レプリケーションの保護メカニズムについて
- 多数の発生するどのトランザクションも同一の小さい表を更新する状態は、できれば避けてください。たとえば、複数のトランザクションが1つの小さい表に読み込みと更新を行って主キーの順序番号生成をシミュレートするのは、優れたアプリケーション設計とは言えません。この設計の場合、すべてのトランザクションが同一のデータ・ブロックを更新しなければなりません。より優れた解決方法として、順序を作成し順序番号をキャッシュに格納すれば、主キーの生成を最適化しパラレル伝播のパフォーマンスを改善できます。

パラレル伝播の計画

アドバンスト・レプリケーション環境のスケジュール・リンクを作成すると、各リンクで、シリアルまたはパラレル伝播のどちらかを使用して接続先に変更内容を非同期で伝播できます。

- シリアル伝播では、レプリケート・トランザクションは、ソース・システムでコミットされた順に一度に1つずつ伝播されます。シリアル伝播を使用するスケジュール・リンクを構成するには、「スケジュール・リンクの作成」または「スケジュール・リンクの編集」プロパティ・シートを使用して、「オプション」ページの「パラレル伝播」設定を使用禁止にします。
- パラレル伝播では、複数のパラレル・ストリームを使用してレプリケート・トランザクションが伝播されるので、スループットがより高くなります。必要に応じて、Oracle は

データの整合性を保つために依存関係にあるトランザクションの実行を命令します。パラレル伝播を使用するスケジュール・リンクを構成するには、「スケジュール・リンクの作成」または「スケジュール・リンクの編集」プロパティ・シートを使用して、「オプション」ページの「パラレル伝播」設定を使用可能にします。「プロセス」を必要な並行度（1以上）に設定します。

パラレル伝播では、使用可能なパラレル・サーバー・プロセスのプールが使用されます。これは、パラレル問合せ、パラレル・ロード、パラレル回復などの他のパラレル操作に使用されるプールと同じです。各サーバー・プロセスでは、シングル・ストリームを介してトランザクションが伝播されます。これらのサーバー・プロセスは、パラレル・コーディネータ・プロセスにより制御されます。コーディネータは、トランザクションの依存関係を追跡してサーバー・プロセスに作業を割り当て、その進行状況を追跡します。

操作の実行中は、パラレル・サーバー・プロセスがパラレル操作に関連付けられたままとなります。操作が完了すると、サーバー・プロセスは他のパラレル操作の処理に使用できるようになります。たとえば、接続先への遅延トランザクション・キューのパラレル送信が実行されるとき、キューの送信に使用されるすべてのパラレル・サーバー・プロセスは、送信が完了するまで占有されたままになります。

サーバーに対するパラレル・サーバー・プロセスのプールを正しく構成するには、アドバンスト・レプリケーション・システムの構成に関する問題をいくつか考慮する必要があります。

- すべてのスケジュール・リンクをシリアル伝播を使用するように構成すると、レプリケーション・システムでは、パラレル・サーバー・プロセスは使用されません。このため、レプリケーションを行うためにパラレル・サーバーのサーバーのプールのサイズを調整する必要はありません。
- スケジュール・リンクを1つでもパラレル伝播を使用するように構成すると、各リンクでその接続先に変更内容を送信するために使用するパラレル・サーバー・プロセスの数を検討する必要があります。さらに、各送信で、別の操作に使用されないようにパラレル・サーバーをホールドする時間も考慮しなければなりません。たとえば、連続的な伝播（「遅延秒数」の値を大きくする）用にスケジュール・リンクを構成すると、その接続先にトランザクションを送信するために使用するパラレル・サーバー・プロセスがホールドされます。このため、サーバーのその他のパラレル操作用のプロセスの数が十分になるように、対応するデータベース・サーバーのパラレル・サーバー・プロセスの数を増加する必要があります。

パラレル問合せプロセスのデータベース・サーバーのプールを構成するには、次の初期化パラメータを使用します。

- PARALLEL_MAX_SERVERS
- PARALLEL_MIN_SERVERS
- PARALLEL_SERVER_IDLE_TIME

追加情報：『Oracle8 Server 概要』を参照してください。

レプリケーションの保護メカニズムについて

Oracle では、障害の発生時に、リモート・サイトに伝播されたトランザクションが失われたり、2 回以上伝播されたりしないことが保証されています。

- 単一のローカル・トランザクション内で実行依頼された複数のプロシージャ・コールは、単一のトランザクションとしてリモートで実行されます。
- 伝播の実行中にネットワークまたはリモート・データベースに障害が発生した場合は、トランザクションがリモート・サイトでロールバックされ、リモート・データベースが再度アクセス可能になり、伝播が成功するまで、トランザクションがローカル・キューに残ります。
- トランザクションは、接続先サイトに正常に伝播されて適用されるまで、ローカル・サイトのキューから削除されることはありません。

注意：伝播が正常に完了しても、リモート・サイトでのトランザクションが正常に適用されるとは限りません。回復不能な競合または記憶領域の不足などのエラーが発生すると、トランザクションでエラーが発生する原因となることがあり、リモート・サイトに記録されます。エラー・トランザクションの表示方法および管理方法の詳細は、6-10 ページの「エラー・トランザクションの表示」を参照してください。

障害からの保護は、シリアル伝播とパラレル伝播の両方に対して提供されます。

- シリアル伝播の場合、アドバンスト・レプリケーション機能で 2 フェーズ・コミットが使用されます。
- パラレル伝播の場合、アドバンスト・レプリケーション機能でパラレル操作用に最適化された特殊な分散トランザクション・プロトコルが使用されます。リモート・サイトは、正常に伝播されたトランザクションを追跡して記録し、この情報を定期的にローカル・サイトに戻します。ローカル・サイトは、この情報を記録し、すべての接続先サイトに伝播されたローカル・キュー内のエントリを消去します。障害が発生した場合、ローカル・サイトは、適切な時点から伝播を続行できるように、正常に伝播されたトランザクションに関する情報を要求します。

データ伝播の依存性の維持

Oracle は、レプリケート・トランザクションがリモート・システムに伝播されたときの依存性の順序を維持します。次に例を示します。

1. トランザクション A が注文を取り消します。
2. トランザクション B が取消しを検出して返金を処理します。

トランザクション B は、ローカル・システムでの注文の取消しのコミット済み更新（トランザクション A）を参照するので、トランザクション B はトランザクション A に依存しています。

トランザクション A（注文の取消し）が正常に伝播された後で、トランザクション B（返金）が伝播されます。取消しを適用した後で返金を処理するという更新が適用されます。

パラレル伝播の依存性の追跡

ローカル・システムで新しいトランザクションが実行されるときに、次のことが行われます。

1. データを更新した最新のシステム変更番号を記録します。これは、新規トランザクションにより従属システム変更番号として参照されます。
2. Oracle では、システム変更番号が従属システム変更番号以下のトランザクションが、リモート・システムに正常に伝播することを確認します。
3. Oracle は、待機中の依存トランザクションを伝播します。

注意：トランザクション間で依存関係の可能性がないときは、トランザクションはパラレルで伝播されます。

パラレル伝播は、シリアル伝播とは異なる方法でデータの整合性を維持します。シリアル伝播では、Oracle は依存関係を維持するために、ローカル・システム上でコミットするときと同じ順序ですべてのトランザクションを適用します。パラレル伝播では、Oracle は依存関係を追跡し、依存関係があるときはコミットされた順に、依存関係がないときはパラレルで実行します。シリアル伝播でもパラレル伝播でも、Oracle はトランザクション内の実行順序を保ちます。遅延トランザクションは、ローカル・トランザクション内で実行された場合と同じ順序で、各システムですべてのリモート・プロシージャ・コールを実行します。

注意：シングルのコーディネータ・プロセスは、リモート・サイトに対する各データベース・リンクごとに存在しています。同じリモート・サイトに対する各データベース・リンクには、異なる接続修飾子が必要です。

追加情報：3-14 ページの「マスター・グループの接続修飾子の使用方法」を参照してください。

並列性を改善するためのトランザクション依存性の最小化

アプリケーション条件によっては、トランザクション間に依存性が生じ、そのため遅延トランザクションの伝播をシリアル化しなければならない場合があります。複数の関連付けられていないトランザクションによってレプリケート表の同じデータ・ブロックを変更する場合、対応するトランザクションはリモート宛先へシリアル伝播されます。

データ・ブロック・レベルで発生したトランザクション依存性を最小化するには、1 つまたは少数のデータ・ブロックにデータ・ブロック変更が集中しないようにする必要があります。次に例を示します。

- レプリケート表に頻繁な INSERT アクティビティが発生する場合、複数の表の空きリストを作成して、新規の行の記憶域を複数のデータ・ブロックに分配することができます。
- 多数の発生するどのトランザクションも同一の小さい表を更新する状態は、できれば避けてください。たとえば、複数のトランザクションが 1 つの小さい表に読み込みと更新を行って主キーの順序番号生成をシミュレートするのは、優れたアプリケーション設計とは言えません。この設計の場合、すべてのトランザクションが同一のデータ・ブロックを更新しなければなりません。より優れた解決方法として、順序を作成し順序番号をキ

キャッシュに格納すれば、主キーの生成を最適化しパラレル伝播のパフォーマンスを改善できます。

スナップショット・サイト・レプリケーションの 使用方法

この章では、スナップショット・サイトを使用するアドバンスド・レプリケーション環境の構成方法および管理方法について説明します。この章のトピックは次のとおりです。

- クイック・スタート: アドバンスド・レプリケーション・システムへのスナップショット・サイトの追加
- スナップショット・サイト・レプリケーションの準備
- スナップショット・ログの管理
- スナップショット・グループの管理
- スナップショットの管理
- リフレッシュ・グループの管理
- スナップショット・サイトの管理に関するその他の情報

注意：この章では、デフォルトのレプリケーション・アーキテクチャ（非同期伝播を使用する行レベル・レプリケーション）を使用するアドバンスド・レプリケーション・システムの管理方法について説明します。プロシージャ・レプリケーションおよび同期伝播の構成方法については、第7章「上級テクニック」を参照してください。また、この章では、Oracle Replication Managerツールを使用して、更新可能なスナップショット・サイトのアドバンスド・レプリケーション・システムの構築および管理する方法の例を示します。参考として、等価のレプリケーション管理 API コールを各項に記載しています。

クイック・スタート：アドバンスド・レプリケーション・システムへのスナップショット・サイトの追加

スナップショット・サイトをアドバンスド・レプリケーション環境に追加するには、次のステップを行ってください。

1. アドバンスド・レプリケーション環境用のスナップショット・サイトを設計する。各スナップショット・サイトでレプリケートする表およびサポート・オブジェクトを決定し、各スナップショット・サイトをサポートするシステム内のマスター・サイトを決定する。
2. Replication Manager セットアップ・ウィザードを使用して、多数のデータベースをスナップショット・サイトとして構成する。各スナップショット・サイトをサポートするために必要なすべてのコンポーネントがすばやく構成されます。
3. 各マスター・サイトで、指定したスナップショットをサポートするためのスナップショット・ログを作成する。
4. 各スナップショット・サイトで、指定したスナップショットをサポートするためのリフレッシュ・グループを作成する。
5. 各スナップショット・サイトで、対応するマスター・グループ・オブジェクトをレプリケートする 1 つ以上のスナップショット・グループを作成する。
6. 各スナップショット・サイトのデータにアクセスするために必要な権限をアプリケーション・ユーザーに付与する。

各ステップおよびその他のオプションの構成ステップの詳細は、この章で後述します。

簡単な例

以下に、スナップショット・サイトをアドバンスド・レプリケーション環境に追加するために必要なステップを簡単な例で示します。アドバンスド・レプリケーション環境ですべてのマスター・サイトを構成したら、次のステップを実行してください。

ステップ 1: スナップショット・サイトを設計する

最初に、各スナップショット・サイトを設計します。スナップショット・サイトは、それぞれに対応付けられたマスター・サイトの表データの単純読み専用スナップショットおよび更新可能スナップショットをサポートします。スナップショット・サイトの表スナップショットには、マスター・グループの表データのすべてまたは 1 つのサブセットだけを含めることができます。どちらの場合も、そのスナップショットはマスター・サイトの表に 1 対 1 で対応する単純スナップショットでなければなりません。スナップショット・グループには、プロシージャおよびパッケージ、ファンクション、シノニム、ビューなどのレプリケーション・オブジェクトを含めることもできます。例では、マスター・サイト DBS1 にあるマスター表 SCOTT.EMP および SCOTT.DEPT の更新可能スナップショットの作成方法を示しています。

注意：SCOTT.EMP の主キーは EMPNO 列で、SCOTT.DEPT の主キーは DEPTNO 列です。

ステップ 2: Replication Manager セットアップ・ウィザードの使用方法

Replication Manager セットアップ・ウィザードは、アドバンスド・レプリケーション・システムのすべてのスナップショット・サイトのサポート・アカウントおよびリンク、スキーマ、スケジュール作成を構成するときに役に立ちます。この例では、セットアップ・ウィザードを使用して次のことを行います。

- マスター・サイト DBS1 を識別する。
- スナップショット・サイト DBS3 を識別する。
- グローバル・レプリケーション管理者および伝播担当者アカウントの役割をする、デフォルトの SNAPADMIN アカウントをスナップショット・サイトに作成する。
- 指定したスナップショットをサポートするスキーマ SCOTT をスナップショット・サイトに作成する。
- スナップショット・サイトからマスター・サイトへのデータ伝播、および遅延トランザクション・キューのパージの、デフォルトのスケジュールを構成する。

ステップ 3: 必要なスナップショット・ログをマスター・サイトに作成する

スナップショット・サイトのスナップショット・グループおよびスナップショットを作成する前に、必要なスナップショット・ログをマスター・サイトに必ず作成しておかなければなりません。スナップショット・ログは、単純スナップショットを 1 つでも高速リフレッシュでサポートしているすべてのマスター表に必要です。Replication Manager の「スナップショット・ログの作成」プロパティ・シートを使用して、マスター表 SCOTT.EMP および SCOTT.DEPT のスナップショット・ログを作成します。

注意: Replication Manager を使用してスナップショット・ログを作成するには、必要な権限が付与されたデータベース管理者 (例、SYSTEM) として、マスター・サイトへのデータベース接続を確立する必要があります。

ステップ 4: 必要なリフレッシュ・グループをスナップショット・サイトに作成する

スナップショット・サイトのスナップショット・グループおよびスナップショットを作成する前に、スナップショットをリフレッシュするために使用するリフレッシュ・グループをスナップショット・サイトに必ず作成しておかなければなりません。Oracle では、個々のスナップショットをリフレッシュ・グループの一部として編成およびリフレッシュします。Replication Manager の「リフレッシュ・グループの作成」プロパティ・シートを使用して、リフレッシュ・グループ SCOTT.REFGRP_1 を作成します。新規リフレッシュ・グループの場合は、単純にデフォルトのスケジュール設定を使用します。

ステップ 5: Replication Manager スナップショット・グループ・ウィザードを使用する

アドバンスド・レプリケーション環境でスナップショット・サイトを使用するときは、表および関連レプリケーション・オブジェクトはスナップショット・グループの一部としてレプリケートされます。スナップショット・サイト DBS3 へのデータベース接続を使用して、Replication Manager のスナップショット・グループ・ウィザードをオープンすると、次の作業を含む新規スナップショット・グループの作成手順が示されます。

- 新規スナップショット・グループのマスター・サイトの選択
- 新規スナップショット・グループのマスター・グループの選択
- 新規スナップショット・グループにレプリケートするマスター・グループ・オブジェクトの選択
- グループの設定および個々のスナップショットの設定の調整

ウィザードを使用して、マスター・サイト DBS1 の EMPLOYEE マスター・グループに対応する新規スナップショット・グループを作成します。ウィザードを使用するときは、マスター表 SCOTT.EMP および SCOTT.DEPT の完全なスナップショットを作成します。両方のスナップショットに対して、デフォルトの伝播および記憶領域設定とリフレッシュ・グループ SCOTT.REFGRP_1 を使用します。

ステップ 6: レプリケーション・オブジェクトへのアクセス権限を付与する

新規スナップショット・グループを構成したら、スナップショット・サイトに接続しているユーザーがそれを使用できるように、スナップショットへのアクセス権限を付与します。

```
GRANT SELECT ON scott.emp TO ... ;
```

その他のステップ

この例では、特定の種類のアドバンスド・レプリケーション・システムを構成するために必要な場合があるその他のステップについては説明されていません。たとえば、アドバンスド・レプリケーション・システムで共有所有権データ・モデルを使用して処理を行う場合、アプリケーションで更新可能なスナップショットを処理する前に、すべてのマスター表に対して競合解消を構成する場合があります。競合解消の構成の詳細は、第 5 章「競合の解消」を参照してください。

スナップショット・サイト・レプリケーションの準備

アドバンスド・レプリケーション環境の一部としてスナップショット・サイトの構築を開始する前に、各スナップショット・サイト・データベースに次のものを用意する必要があります。

- レプリケーション管理者
- レプリケーション伝播担当者
- スナップショット・サイトとマスター・サイト間のデータベース通信を提供するデータベース・リンク

Replication Manager セットアップ・ウィザードでは、上記のコンポーネントを使用してスナップショット・サイトをすばやく構築できます。Replication Manager セットアップ・ウィザードでスナップショット・サイトを構築したら、次の追加の設定ステップを行う必要があります。

- マスター・サイトへの変更の伝播方法を決定する
- スナップショットのリフレッシュ方法を決定する

- スナップショット・サイトで1つ以上のSNPバックグラウンド・プロセスを起動する
以降の項では、スナップショット・サイト・レプリケーション用のデータベースを設定する方法を説明します。

レプリケーション・セットアップ・ウィザード

デフォルト構成用のスナップショット・サイトの設定は、Replication Managerセットアップ・ウィザードを使用すると簡単なプロセスで行うことができます。作成する各スナップショット・サイトで、このウィザードにより次のステップが実行されます。

- スナップショット・サイトのレプリケーション管理者の役割をするデータベース・アカウントが作成される。デフォルトでは、このアカウントはレプリケーション伝播担当者の役割もします。
- レプリケーション管理者 / 伝播担当者アカウントに、必要な権限が付与される。
- スナップショット・サイトから対応するマスター・サイトへのスケジュール・リンクが作成される。

Replication Manager セットアップ・ウィザードを起動するには：

1. 「ファイル」をクリックする。
2. 「セットアップ・ウィザード」をクリックする。

次の項では、Replication Managerセットアップ・ウィザードを使用して、アドバンスト・レプリケーション・システムにスナップショット・サイトを設定する方法を説明します。

スナップショット・サイトの作成

レプリケーション・セットアップ・ウィザードの最初のページで、実行するレプリケーション環境セットアップのタイプを指定するように要求されます。

1. 「スナップショット・サイトのセットアップ」を選択する。
2. 「次」をクリックする。

マスター・サイトの指定

次のページで、作成する新規スナップショット・サイトのマスター・サイトを指定するように要求されます。

1. マスター・サイトの名前（たとえば、DBS1）をタイプする。
2. マスター・サイトの SYSTEM ユーザー用のパスワードをタイプする。

スナップショット・サイトの作成

その次のページで、レプリケーション・システム内のスナップショット・サイトにリストを作成できます。この時点で、セットアップ・ウィザードで利用できるReplication Managerデータベース接続がないことがよくあります。この場合、次のステップを実行します。

1. 「新規作成」をクリックする。

2. 表示される「新規スナップショット・サイト」ダイアログに、スナップショット・サイトのグローバル・データベース名と、そのサイトの SYSTEM アカウントのパスワードを入力する（セットアップ・ウィザードでは、SYSTEM アカウントを使用して後続の構成タスクを実行します）。
3. 終了したら「追加」をクリックして、セットアップ・ウィザードのスナップショット・サイトのリストにそのサイトを追加する。
4. 各スナップショット・サイトについてステップ2を繰り返す。最後のサイトの情報の入力が終了したら、「OK」をクリックしてそのサイトを追加し、「新規スナップショット・サイト」ダイアログを終了します。
5. スナップショット・サイトのリストの確認が終わったら、「次」をクリックして処理を続行する。

レプリケーション管理者アカウントおよび伝播担当者アカウントの作成

その次のページで、各スナップショット・サイトのレプリケーション管理者および伝播担当者として機能するデータベース・アカウントの情報を指定できます。ウィザードにより、システム内のすべてのスナップショット・サイトに同じ名前とパスワードのアカウントが作成されます。

セットアップ・ウィザードでは各スナップショット・サイトで2つの異なるタイプのアカウント構成が可能です。

- デフォルトでは、管理者と伝播担当者の両方の機能を実行できるアカウントが1つ作成される。アカウント名とパスワードを入力し、「次」を押して処理を続行します。
- また、伝播担当者アカウントとは別の独立したレプリケーション管理用アカウントを作成する場合は、「別の伝播担当者」をクリックし、伝播担当者ユーザー・アカウント用の別のアカウント情報を入力する。それから、「次」を押して処理を続行します。

一般的には、両方の機能に対して1つのアカウントを作成するデフォルトのウィザード・セットアップ・オプションが好まれます。ただし、独自のセキュリティ上の問題があるシステムでは、固有の管理者および伝播担当者アカウントが必要な場合があります。アドバンスト・レプリケーション環境のスナップショット・サイト用の特殊なセキュリティ構成の詳細は、7-23 ページの「アドバンスト・レプリケーション・スナップショット・サイトに対する代替セキュリティの設定」を参照してください。

レプリケーション・オブジェクトを編成するスキーマの作成

同じページで、新規スナップショット・サイトのレプリケーション・オブジェクトを格納するスキーマとして作成するスキーマを指定できます。ウィザードにより、マスター・サイトのレプリケーション・オブジェクトを格納するのに見合ったスキーマが作成され、スナップショット・サイトのスキーマから対応するマスター・サイトのスキーマへのプライベート・データベース・リンクが作成されます。

新規スキーマをリストに追加するには：

1. 「オブジェクト・スキーマの選択」をクリックする。

2. 表示される「マスター・スキーマから選択」リストから、新規スナップショット・サイトへレプリケートするレプリケーション・オブジェクトが格納されているすべてのマスター・サイト・スキーマを選択する。スキーマのリストの確認が終わったら、「次」をクリックして処理を続行します。次に、指定した各マスター・サイト・スキーマのパスワードを入力するようにReplication Managerから要求されます。それぞれのパスワードを入力したら、「OK」をクリックして処理を続行します。

スケジュール・リンクの作成

その次のページで、すべてのスナップショット・サイトのデフォルトの伝播特性を指定できます。セットアップ・ウィザードではこの情報を使用して、各スナップショット・サイトからマスター・サイトへの対応するスケジュール・リンクを作成します。ウィザードのこのページの各設定の説明は、3-7 ページの「スケジュール・リンクの管理」を参照してください。

デフォルトのスケジュール設定の確認が終わったら「次」をクリックして処理を続行します。

ページ・スケジュールの指定

次のページで、システム内の各スナップショット・サイトの遅延トランザクション・キューに対するデフォルトのページ・スケジュールを構成できます。ウィザードのこのページの各設定の説明は、3-11 ページの「サイトの遅延トランザクション・キューのページ」を参照してください。

デフォルトのページ設定の確認が終わったら、「次」をクリックして処理を続行します。

各スナップショット・サイトのカスタマイズ

その次のページで、システム内の個々のスナップショット・サイトの設定をカスタマイズできます。システム内のスナップショット・サイトのカスタマイズを行わない場合、各サイトでは次の設定が同じになります。

- レプリケーション管理者および伝播担当者
- データベース・リンク仕様部
- スケジュールされた伝播設定

スナップショット・サイトの設定をカスタマイズするには：

1. カスタマイズするターゲット・スナップショット・サイトを選択する。
2. 「カスタマイズ」をクリックする。

次に、「スナップショット・サイトのカスタマイズ」プロパティ・シートを使用して、ターゲット・スナップショット・サイトの次の設定をカスタマイズします。

- レプリケーション管理者または伝播担当者のアカウント（またはその両方）
- 管理者 / 伝播担当者データベース・リンクの USING 句
- スケジュールされた伝播設定およびページ設定
- サポート・レプリケーション・スキーマ

スナップショット・サイトのカスタマイズされた設定の確認が終わったら「OK」をクリックします。別のスナップショット・サイトの設定をカスタマイズするには、上記のプロセスを繰り返します。すべてのスナップショット・サイトのカスタマイズが終了したら、「次」をクリックして処理を続行します。

構成の調査および構築

その次のページで、アドバンスド・レプリケーション・システムの構成を終了するかどうか尋ねられます。「終了」をクリックして処理を続行します。この結果、設定をすばやく確認できる情報ダイアログが表示されます。

- すべて正しく設定されていたら、「OK」をクリックしてレプリケーション環境用の新規スナップショット・サイトを設定する。
- 設定に誤りがある場合は、「取消し」をクリックして「終了」ページに戻る。次に、「戻る」および「次」をクリックしてウィザードのページ内を移動し、設定を変更する。環境を構築する準備ができたなら、ウィザードの最終ページに戻る。

「終了」をクリックすると、Replication Managerによりスナップショット・サイトが構築されます。

注意：セットアップ・プロセス中に実行される API プロシージャのスクリプトを記録する場合は、システムを構築する前に「スクリプトを記録する」をクリックします。さらに、Replication Managerにより、現行の作業ディレクトリ内に別のスクリプト Repsetup.log が記録されます。

追加情報: 7-23ページの「アドバンスド・レプリケーション・スナップショット・サイトに対する代替セキュリティの設定」を参照してください。

終了後の作業

Replication Manager セットアップ・ウィザードを終了したら、次のステップを実行して構成を続行してください。

1. 各スナップショット・サイトにスナップショット・グループを作成する。スナップショット・グループの作成および管理に関する詳細は、4-11 ページの「スナップショット・グループの管理」を参照してください。
2. 各レプリケーション・グループのすべての更新可能スナップショットのマスター表に対して、競合解消を構成する。レプリケーション・オブジェクトの競合の解消の構成に関する詳細は、第5章「競合の解消」を参照してください。
3. 各マスター・グループのレプリケーション・アクティビティを再開する。マスター・グループのレプリケーション・アクティビティの管理の詳細は、3-16 ページの「マスター・グループのレプリケーション・アクティビティの再開」を参照してください。
4. レプリケーション環境を監視し、システムが正しく動作していることを確認する。マスターおよびスナップショット・グループのアクティビティの監視の詳細は、6-4 ページの「アドバンスド・レプリケーション・システムの監視」および 4-22 ページの「スナップショット・サイトの管理に関するその他の情報」を参照してください。

スケジュール・リンクおよびスナップショット・リフレッシュの計画

アドバンスド・レプリケーション環境の更新可能スナップショットでは、データをマスター表に"送信"したり、マスター表から"取り出す"ことができます。アドバンスド・レプリケーション・システムでは、スナップショット・リフレッシュ機能を使用して、マスター表から変更内容を非同期で取り出し、対応付けられた更新可能スナップショットへ送信します。それに対して、更新スナップショットに対する更新内容は、行レベルのデータ伝播メカニズムを使用して、マスター表に非同期（または同期）で送信されます。更新可能スナップショットの送信および取出しタスクは独立した処理であるため、まとめて構成することも別々に構成することもできます。

- スナップショット・サイトでは、リフレッシュ・グループを構成すると、メンバー・スナップショットに加えられたすべての変更をマスター・サイトに自動的に送信して、スナップショットをリフレッシュできます。
- スナップショット・サイトでは、更新可能スナップショットを構成すると、変更内容のマスター・サイトへの送信およびスナップショットのリフレッシュを、異なる時間および間隔で実行できます。

たとえば、マスター・サイトで情報を整理統合するアドバンスド・レプリケーション環境では、更新可能スナップショットを構成して、変更内容をマスター・サイトに1時間ごとに送信し、更新可能スナップショットのリフレッシュはあまり実行しないようにできます。

スナップショット・サイトを作成する前に、スナップショット・リフレッシュ・グループを使用してどのようにリフレッシュするか、変更内容をマスター・サイトに伝播するための各スナップショットのスケジュール・リンクをどのように構成するかを決めます。計画の立てかたによって、後続の構成ステップを簡単にすることができます。

追加情報：スケジュール・リンクの作成および管理の詳細は、3-7 ページの「スケジュール・リンクの管理」を参照してください。

追加情報：リフレッシュ・グループの構成および管理の詳細は、4-20 ページの「リフレッシュ・グループの管理」を参照してください。

SNP バックグラウンド・プロセスの起動

管理をより簡単に行うために、大部分のアドバンスド・レプリケーション環境ではデータ伝播およびスナップショットのリフレッシュが自動的に行われるように構成されています。このため、アドバンスド・レプリケーション環境の各スナップショット・サイトで、1つ以上の SNP アドバンスド・レプリケーションを起動する必要があります。各サーバーの SNP バックグラウンド・プロセスの設定は、次の初期化パラメータで制御されます。

- `JOB_QUEUE_PROCESSES` では、サーバーごとに SNPn バックグラウンド・プロセス番号を指定します。n は 0 ~ 9、A ~ Z を表します。このパラメータには 1 以上を設定します。大規模なシステムでないかぎり、通常はバックグラウンド・プロセスは 2 つか 3 つで十分です。
- `JOB_QUEUE_INTERVAL` で、サーバーのバックグラウンド・プロセスの起動の間隔を秒単位で指定します。一般的なレプリケーション環境では、デフォルトの 60 秒が適当です。

スナップショット・ログの管理

マスター表のスナップショット・ログには、対応する単純スナップショットすべての高速リフレッシュ・データが記録されます。サーバーは、スナップショットの高速リフレッシュを実行するとき、マスター表のスナップショット・ログを使用して、スナップショットを非常に効率よくリフレッシュします。Replication Managerにはスナップショット・ログを作成および管理するときに役に立つ多数の機能があります。

次の項では、スナップショット・ログの管理方法の詳細を説明します。

スナップショット・ログの作成

スナップショット・サイトのスナップショット・グループおよびスナップショットを作成する前に、必要なスナップショット・ログをマスター・サイトに必ず作成しておかなければなりません。スナップショット・ログは、単純スナップショットを1つでも高速リフレッシュでサポートしているすべてのマスター表に必要です。

スナップショット・サイトのマスター・サイトにスナップショット・ログを作成するには：

1. マスター・サイトに接続するオープン・データベース接続内の任意の場所をクリックする。
2. 「新規作成」ツールバー・ボタンをクリックする。
3. 「新規スナップショット・ログ」をクリックする。

「スナップショット・ログの作成」プロパティ・シートには、「一般」および「表領域およびエクステント」、「フィルタ列」の3つのページがあります。

- 「一般」ページの設定を使用して、マスター表のスキーマおよび名前を決定し、ログで主キー・スナップショットとROWID スナップショットのどちらを（または両方を）サポートするかを指定する。
- 「表領域およびエクステント」ページを使用して、ログの記憶領域の仕様を指定する。
- 「フィルタ列」ページを使用して、副問合せを含むスナップショットのフィルタ列を指定する。

スナップショット・ログの変更

スナップショット・ログの記憶領域設定またはフィルタ列を編集するには：

1. マスター・サイトのターゲット・スナップショットをクリックする。
2. 「プロパティ」ツールバー・ボタンをクリックする。

「スナップショット・ログの編集」プロパティ・シートには、「一般」および「表領域およびエクステント」、「フィルタ列」の3つのページがあります。

- 「一般」ページの設定を使用して、ログで主キー・スナップショットとROWID スナップショットのどちらを（または両方を）サポートするかを追加する。
- 「表領域およびエクステント」ページを使用して、ログの記憶領域の仕様を変更する。

- ・ 「フィルタ列」ページを使用して、副問合せを含むスナップショットのフィルタ列を変更する。

スナップショット・ログの削除

マスター表で、高速リフレッシュを行う必要があるスナップショットがサポートされなくなったときは、対応するスナップショット・ログを削除できます。スナップショット・ログを削除するには：

1. マスター・サイトのターゲット・スナップショットをクリックする。
2. 「削除」ツールバー・ボタンをクリックする。

スナップショット・グループの管理

アドバンスド・レプリケーション・システムのスナップショット・グループでは、対応するマスター・グループのすべてまたはいくつかのオブジェクトの一部または完全なコピーをメンテナンスします。Replication Managerにはスナップショット・グループを作成および管理するときに役に立つ多数の機能があります。次の項では、スナップショット・グループの管理方法の詳細を説明します。

スナップショット・グループの作成

新規スナップショット・グループを作成する前に、次の準備作業が完了していることを確認する必要があります。

- ・ 新規スナップショットが更新内容をマスター・サイトに伝播するために使用するスケジュール・リンクが作成されている。
- ・ 新規スナップショット・グループの作成作業の一環として、高速リフレッシュを行う単純スナップショットを作成する場合は、グループを作成する前に必要なスナップショット・ログが作成されていることを確認する。

新規スナップショット・グループを作成するには：

1. スナップショット・サイトに接続するオープン・データベース接続内の任意の場所をクリックする。
2. 「新規作成」ツールバー・ボタンをクリックする。
3. 「新規スナップショット・グループ」をクリックする。

すると、Replication Managerにより、新規スナップショット・グループの作成手順が示されます。次の項では、スナップショット・グループ・ウィザードの各ページで指定できる設定について詳しく説明します。

同等の API:

DBMS_REPCAT.CREATE_SNAPSHOT_REPGROUP, DBMS_REPCAT.CREATE_SNAPSHOT_REPOBJECT, DBMS_REFRESH.MAKE

スナップショット・グループの設計

スナップショット・グループ・ウィザードの最初の 3 ページで、スナップショット・グループの基本構造を設計できます。

- 最初に、新規グループのマスター・サイトを指定する。
- 次に、新規スナップショット・グループに対応するマスター・グループを選択する。
- 最後に、新規スナップショット・グループの一部としてレプリケートするマスター・グループ内のオブジェクトを選択する。マスター・グループのオブジェクトはすべて選択しても、一部を選択しても、何も選択しなくてもかまいません。

スナップショット・グループ設定およびデフォルトの設定

スナップショット・グループ・ウィザードでは、新規スナップショット・グループのすべてのスナップショットのグループ設定およびデフォルト設定を指定できます。次の項で各設定について説明します。

グループ伝播モード

スナップショット・グループに更新可能スナップショットがある場合、スナップショット・サイトでスナップショットに対して行った DML 変更を、スナップショット・サイトのマスター・サイトにどのように伝播するかを指定しなければなりません。デフォルト設定は非同期伝播用です。アドバンスト・レプリケーション環境での同期伝播の詳細は、7-6 ページの「同期データ伝播の使用方法」を参照してください。

個々のスナップショット設定

「グループ内のスナップショット」リストからスナップショットを選択した後で、スナップショットの多数のプロパティを調整できます。

- 「更新可能」をクリックして、選択したスナップショットを更新可能スナップショットとして作成する。
- 「高速リフレッシュ」をクリックして、選択したスナップショットに高速リフレッシュを使用する。
- 「Where 句」をクリックし、次に「編集」をクリックして、選択したスナップショットの WHERE 句を指定する。詳細は、4-17 ページの「WHERE 句を使用する更新可能スナップショットを作成する」を参照してください。
- 「表領域およびエクステンツ」をクリックして、選択したスナップショットの記憶領域の設定を指定する。
- 「通信を最小化する」をクリックして、選択した更新可能スナップショットの競合検出の実行に必要な通信を最小化する。この設定の詳細は、4-18 ページの「データ伝播の最小化」を参照してください。
- 「リフレッシュ・グループ」を選択して、選択したスナップショットのリフレッシュ・グループを選択する。「編集」をクリックして、選択したリフレッシュ・グループのプロパティを編集する。新規リフレッシュ・グループを作成する場合は、「新規作成」をクリックする。

注意：デフォルトでは、スナップショット・グループ・ウィザードはスナップショット・グループ内のすべてのスナップショットに対して、そのスナップショット・グループと同じ名前でリフレッシュ・グループを作成します。

デフォルトのスナップショット設定

「グループ内のスナップショット」リストから「デフォルトのスナップショット設定」を選択すると、特定の設定をされていないグループのすべてのスナップショット用のデフォルト設定を調整できます。ウィザードを使用すると、新規スナップショット・グループのすべてのスナップショットに対して、デフォルトの記憶領域設定、デフォルトの最小通信設定、デフォルトのリフレッシュ・グループを設定できます。

マスター・サイトにスナップショット・グループを登録する

スナップショット・グループを作成すると、Oracle ではマスター・サイトにグループを自動的に登録しようとします。なんらかの理由で登録が失敗した場合は、レプリケーション管理 API を使用するとスナップショット・グループをマスター・サイトに手動で登録できます。

同等の API: DBMS_REPCAT.REGISTER_SNAPSHOT_REPGROUP

マスター・サイトのスナップショット・グループを登録解除する

スナップショット・グループを削除するか、またはスナップショット・グループのマスター・サイトを切り替えると、Oracle では以前のマスター・サイトのグループを自動的に登録解除しようとします。このプロセスがなんらかの理由で失敗した場合は、レプリケーション管理 API を使用するとマスター・サイトのスナップショット・グループを手動で登録解除できます。

同等の API: DBMS_REPCAT.UNREGISTER_SNAPSHOT_REPGROUP

スナップショット・グループにオブジェクトを追加する

スナップショット・グループにオブジェクトを追加するには：

1. スナップショット・サイトのターゲット・スナップショット・グループをクリックする。
2. 「グループに追加」ツールバー・ボタンをクリックする。
3. 「スナップショット・グループにオブジェクトを追加する」をクリックする。
4. スナップショット・グループ・ウィザードを使用して、マスター・グループから選択したオブジェクトを追加する。
5. 「次」をクリックする。
6. ステップ 5 で 1 つ以上のマスター表を選択した場合は、ウィザードを使用して対応するスナップショットの個々の設定を指定できる。
7. 「終了」をクリックすると、ウィザードにより新規オブジェクトがスナップショット・グループに追加される。

同等の API: DBMS_REPCAT.CREATE_SNAPSHOT_REPOBJECT

スナップショット・グループのオブジェクトの変更

スナップショット・グループのレプリケーション・オブジェクトの定義を変更するときは、常に Replication Manager（または同等の API コール）を使用してください。Enterprise Manager や SQL DDL コマンド（ALTER TABLE など）を使用してレプリケート環境を変更すると、矛盾が生じる可能性があります。

注意：スナップショット・サイトでの個々のレプリカのローカルでのカスタマイズは、Oracle のアドバンスド・レプリケーション機能では行えません。レプリケーション管理者として、Replication Manager で実行されるグローバルなカスタマイズが、ローカルなカスタマイズにより妨げられることがないようにしてください。

スナップショット・グループのスナップショット以外のオブジェクトの定義は変更できません。スナップショット・グループ内の個々のスナップショットの変更方法の詳細は、4-19 ページの「スナップショットの変更」を参照してください。

スナップショット・グループのオブジェクトの削除

スナップショット・グループからオブジェクトを削除するには：

1. スナップショット・サイトのターゲット・スナップショット・グループをクリックする。
2. グループから削除するターゲット・オブジェクトを選択する。
3. 「削除」ツールバー・ボタンをクリックする。

注意：スナップショット・グループからオブジェクトを削除すると、そのオブジェクトのサポートに必要なすべての対応するシステム生成オブジェクトが、Replication Manager により自動的に削除されます。

同等の API: DBMS_REPCAT.DROP_SNAPSHOT_REPOBJECT

スナップショット・グループの編集

スナップショット・グループを編集するには：

1. スナップショット・サイトのターゲット・スナップショット・グループをクリックする。
2. 「プロパティ」ツールバー・ボタンをクリックする。

「スナップショット・グループの編集」プロパティ・シートには「一般」と「オブジェクト」の2つのページがあります。

- 「一般」ページを使用して、ターゲット・スナップショット・グループのマスター・サイトとそのマスター・サイトへの伝播モードを編集します。
- 「オブジェクト」ページを使用して、スナップショット・グループのオブジェクトを追加および削除します。

同等の API:

DBMS_REPCAT.SWITCH_SNAPSHOT_MASTER,
DBMS_REPCAT.ALTER_SNAPSHOT_PROPAGATION

スナップショット・グループの削除

スナップショット・グループを削除するには:

1. スナップショット・サイトのターゲット・スナップショット・グループをクリックする。
2. 「削除」ツールバー・ボタンをクリックする。

注意: スナップショット・グループを削除するときに、そのグループのオブジェクトをデータベースから削除するかどうかを決めることもできます。グループを削除するときにそのグループのオブジェクトを削除しないことを選択すると、オブジェクトはスナップショット・サイトのデータベースに残ります。対応するスナップショット・グループの削除後に残っていたスナップショットは、非グループ化スナップショットとして残り、リフレッシュ・グループ内にあるかぎりリフレッシュされます。ただし、グループ化されていない更新可能スナップショットに加えられた変更内容は、マスター・サイトには転送されません。

同等の API: DBMS_REPCAT.DROP_SNAPSHOT_REPGROUP

スナップショットの管理

表スナップショットは、1つ以上のリモート・マスター表から作成された表データのローカル・コピーです。アプリケーションでは、読み専用表スナップショットでのデータの問合せは行えますが、スナップショットの行の挿入または更新、削除は行えません。一方、更新可能スナップショットでは、行の問合せおよび挿入、更新、削除が可能です。

Oracle のデータ・レプリケーション機能では独立した表スナップショットとともに、スナップショット・グループの一部である単純スナップショットをサポートしています。新規スナップショットをスナップショット・グループの一部として作成するかどうかを決定するには、次の事項を検討してください。

- スナップショット・グループには単純な読み専用スナップショットおよび更新可能スナップショットを含めることができます。
- スナップショット・グループでは複合読み専用スナップショットはサポートされていない。Oracle では、複合読み専用スナップショットは、すべてスナップショット・グループから独立して作成されます。

Replication Managerには、アドバンスド・レプリケーション環境でスナップショットを作成および管理するときに役に立つ多数の機能があります。次の項では、スナップショットの管理方法の詳細を説明します。

注意: Replication Manager は、基本レプリケーション環境で読み専用スナップショットを管理するようには設計されていません。読み専用スナップショットを使用する基本レプリケーション環境の作成および管理方法の詳細は、第2章「基本レプリケーションの使用法」を参照してください。

スナップショットの作成

新規単純スナップショットを作成する前に、次の準備作業を完了していることを確認する必要があります。

- 高速リフレッシュを使用する単純スナップショットを作成する場合は、スナップショットを作成する前にマスター表に必要なスナップショット・ログを作成する。
- スナップショットの自動リフレッシュに使用するリフレッシュ・グループを作成する。

スナップショット・グループから独立した読み専用スナップショットを作成するには：

1. スナップショット・サイトに接続するオープン・データベース接続内の任意の場所をクリックする。
2. 「新規作成」ツールバー・ボタンをクリックする。
3. 「新規スナップショット」をクリックする。

この結果、Replication Managerのスナップショット・ウィザードにより、新規スナップショットの作成手順が示されます。

注意：Replication Manager を使用して更新可能スナップショットを作成する場合、スナップショット・ウィザードでは、新規スナップショットを常に適切なスナップショット・グループの一部として作成します。

読み専用または更新可能スナップショットをスナップショット・グループの一部として作成するには：

1. スナップショット・サイトのターゲット・スナップショット・グループをクリックする。
2. 「グループに追加」ツールバー・ボタンをクリックする。
3. 「スナップショット・グループにオブジェクトを追加する」をクリックする。

この結果、Replication Managerにより、スナップショットなどの新規スナップショット・オブジェクトの作成手順が示されます。スナップショット・グループ・ウィザードの設定の情報は、4-11 ページの「スナップショット・グループの作成」を参照してください。

同等の API: DBMS_REPCAT.CREATE_SNAPSHOT_REPOBJECT

スナップショットのデータ型の考慮

Oracle では NUMBER、DATE、VARCHAR2、CHAR、NVARCHAR2、NCHAR、RAW、ROWID というデータ型を使用するマスター表列のスナップショットをサポートしています。

また Oracle は、バイナリ型 LOB (BLOB) および文字型 LOB (CLOB)、各国文字型 LOB (NCLOB) というラージ・オブジェクト型を使用するマスター表列のスナップショットもサポートしています。ただし、スナップショットの定義問合せの WHERE 句で LOB 列は参照できません。部分単位の更新と追加がこれらの LOB 列に適用されるとき、レプリケーションで使用する遅延および同期リモート・プロシージャ・コール・メカニズムは、サポートされる LOB データ型に部分単位の変更だけを伝播します。

注意：一部のサイトが Oracle7 リリース 7.3 を実行しているレプリケーション環境では、LOB データ型のレプリケーションをサポートしていません。

Oracle では LONG データ型を使用する列のレプリケーションはサポートしていません。LONG データ型を含む列のデータはスナップショットから単純に削除されます。

また、ユーザー定義オブジェクト型および外部 LOB、ファイル・ベース LOB (BFILE) もサポートしていません。これらのデータ型の列を含むスナップショットを構成しようとすると、エラー・メッセージが戻されます。

WHERE 句を使用する更新可能スナップショットを作成する

WHERE 句を使用する更新可能スナップショットを作成して、マスター表内の行のサブセットをスナップショットに反映させることができます。たとえば、マスター・サイトの注文の STATUS 値によって、ORDERS マスター表の更新可能スナップショットに異なる注文のセットが反映される次のようなワークフロー環境を検討してみます。

ORDER データベース：

```
CREATE SNAPSHOT sales.orders FOR UPDATE
AS SELECT * FROM sales.orders@dbsl.acme.com
WHERE status = 'SHIPPABLE';
```

SHIP データベース：

```
CREATE SNAPSHOT sales.orders FOR UPDATE
AS SELECT * FROM sales.orders@dbsl.acme.com
WHERE status = 'BILLABLE';
```

ACCT データベース：

```
CREATE SNAPSHOT sales.orders FOR UPDATE
AS SELECT * FROM sales.orders@dbsl.acme.com
WHERE status = 'COMPLETE';
```

Oracle では、アプリケーションが更新可能スナップショットの結果セットの選択条件になる列を更新することを制限していません。たとえば、ORDER データベースに接続されたアプリケーションは、注文のステータスを "BILLABLE" に変更できます。この動作により、少なくともスナップショットがリフレッシュされるまでは、スナップショットの選択条件に一致しなくなった行を更新可能スナップショットに含めることができます。この動作は、多くのアプリケーションで可能です。

それとは逆に、更新可能スナップショットの行を常にスナップショットの選択条件に一致合致させたいときは、CHECK 制約を使用してスナップショットまたはスナップショットの実表のビューを定義します。

データ伝播の最小化

Replication Managerウィザードのいくつかのページとプロパティ・シートにある「通信を最小化する」設定を使用して、更新可能スナップショットの競合検出を実行するために、スナップショット・サイトが送信しなければならないデータ量を決定できます（主キー・スナップショットのみ）。デフォルト設定を使用して通信を最小化する場合、Oracleは、更新された列の新しい値と、更新された各列グループ内の主キーおよび列の古い値だけを伝播します。次の Replication Manager コンポーネントにより、「通信を最小化する」設定にアクセスできます。

- スナップショット・グループ・ウィザード
- スナップショット・ウィザード
- 「レプリケーション・オブジェクトの編集」プロパティ・シート

注意：通信を最小化するためのデフォルトの設定は、Oracle8 データベースにだけ有効です。Oracle7 リリース 7.3 のマスター表の更新可能スナップショットを基にしている場合は、「通信を最小化する」設定を使用禁止にしなければなりません。使用禁止にすると、Oracle は行内の任意の列が更新されたときに、行内のすべての列の古い値と新しい値を伝播します。これは、Oracle7 リリース 7.3 での動作です。

追加情報：5-34 ページの「更新の競合解消のためのデータ伝播の最小化」を参照してください。

副問合せを使用するスナップショットの作成

スナップショット・グループ・ウィザードまたはスナップショット・ウィザードのどちらかを使用すると、新規スナップショットを副問合せスナップショットとして作成できます。

スナップショット・グループ・ウィザード使用時に副問合せスナップショットを作成するには：

1. スナップショット・グループ・ウィザードの「個々のスナップショット設定」ページの「グループ内のスナップショット」リスト内のスナップショットをクリックする。
2. 「Where句」オプションをクリックし、次に、対応する「編集」ボタンをクリックする。
3. 新規スナップショットの副問合せをタイプする。

スナップショット・ウィザード使用時に副問合せスナップショットを作成するには、スナップショット・ウィザードの「Where 句」ページで新規スナップショットに対する副問合せをタイプします。

追加情報：副問合せスナップショットの詳細は、2-15 ページの「副問合せを使用するスナップショットの作成」を参照してください。

追加情報：スナップショットの WHERE 句の指定方法の詳細は、4-17 ページの「WHERE 句を使用する更新可能スナップショットを作成する」を参照してください。

更新可能スナップショットのレプリケーション・サポートの再生成

更新可能スナップショットのサポートを生成するには：

1. ターゲット・スナップショットをクリックする。
2. 「プロパティ」ツールバー・ボタンをクリックする。
3. 「レプリケーション・オブジェクトの編集」プロパティ・シートの「一般」ページの「サポートの生成」をクリックする。

同等の API: DBMS_REPCAT.GENERATE_SNAPSHOT_SUPPORT

スナップショットの変更

次の項ではアドバンスド・レプリケーション環境でスナップショットを変更する方法を説明します。

スナップショットの記憶領域設定の編集

スナップショットの記憶領域設定を編集するには：

1. ターゲット・スナップショットをクリックする。
2. 「プロパティ」ツールバー・ボタンをクリックする。
3. 「スナップショットの編集」プロパティ・シート（グループから独立したスナップショットの場合）または「レプリケーション・オブジェクトの編集」プロパティ・シート（グループ内のスナップショットの場合）の「表領域およびエクステンツ」ページの設定を変更する。

スナップショットの実表の操作

スナップショットの実表内のデータに対して、操作または修正、加算、減算をしないでください。スナップショットの実表に対しては、参照制約や一意制約などの統合制約を定義できません。ただし、このような制約は遅延制約チェック用に構成しなければなりません。

スナップショットの実表でPL/SQLトリガーを定義することもできます。ただし、このようなトリガーはスナップショット・リフレッシュ中に起動しないようにコーディングする必要があります。スナップショット・リフレッシュ中にトリガーが起動すると、予期しない結果が生じることがあります。

追加情報：7-32 ページの「トリガーおよびレプリケーション」を参照してください。

スナップショット定義の変更

スナップショット定義を変更するときには、Enterprise ManagerやSQL DDLコマンド（ALTER TABLE など）を絶対に使用しないでください。スナップショットの定義を変更するには、スナップショットを削除してから再作成します。

注意：スナップショット・サイトでの個々のレプリカのローカルでのカスタマイズは、Oracle のアドバンスド・レプリケーション機能では行えません。レプリケーション管理者として、Replication Managerで実行されるグローバルなカスタマイズが、ローカルなカスタマイズにより妨げられることがないようにしてください。

スナップショットの削除

スナップショット・グループからスナップショットを削除するには：

1. スナップショット・サイトのターゲット・スナップショット・グループをクリックする。
2. グループから削除するターゲット・スナップショットを選択する。
3. 「削除」ツールバー・ボタンをクリックする。

注意：スナップショット・グループからスナップショットを削除すると、そのスナップショットのサポートに必要なすべての対応するシステム生成オブジェクトが、Replication Manager により自動的に削除されます。

同等の API: DBMS_REPCAT.DROP_SNAPSHOT_REPOBJECT

リフレッシュ・グループの管理

いくつかの関連マスター表の表スナップショット間の参照整合性およびトランザクション一貫性を保つために、Oracle では、個々のスナップショットをリフレッシュ・グループの一部として編成およびリフレッシュします。リフレッシュ・グループ内のすべてのスナップショットのリフレッシュが終了すると、グループ内のすべてのスナップショットのデータはトランザクション一貫性を保つある時点のデータに対応します。Replication Managerにはアドバンスト・レプリケーション環境でリフレッシュ・グループを作成および管理するときに役に立つ多数の機能があります。次の項では、リフレッシュ・グループの管理方法の詳細を説明します。

追加情報：リフレッシュ・グループの作成および管理の詳細は、2-24 ページの「リフレッシュ・グループの作成」を参照してください。

リフレッシュ・グループの作成

スナップショット・サイト用のリフレッシュ・グループを作成するには：

1. スナップショット・サイトに接続するデータベース内の任意の場所をクリックする。
2. 「新規作成」ツールバー・ボタンをクリックする。
3. 「新規リフレッシュ・グループ」をクリックする。

「リフレッシュ・グループの作成」プロパティ・シートには、「一般」および「スナップショット」、「スケジュール作成」の3つのページがあります。

- 「一般」ページを使用して、グループの名前および所有者を指定する。
- 「スナップショット」ページを使用して、リフレッシュ・グループにスナップショットを追加する。
- 「スケジュール作成」ページを使用して、リフレッシュ・グループのリフレッシュ設定を構成する。

注意：スナップショットをリフレッシュするには、スナップショットで使用されるデータベース・リンクのユーザー・アカウントが、マスター実表の SELECT 権限を持ち、高速リフレッシュを使用する場合はさらに対応するスナップショット・ログの SELECT 権限を持っていなければなりません。

同等の API: DBMS_REFRESH.MAKE

リフレッシュ・グループにスナップショットを追加する

リフレッシュ・グループに 1 つ以上のスナップショットを追加するには:

1. スナップショット・サイトのスナップショット・グループまたは「グループ化されていないスナップショット」フォルダをクリックする。
2. 選択したスナップショットを同じスナップショット・サイトのターゲット・リフレッシュ・グループにドラッグアンドドロップする。

同等の API: DBMS_REFRESH.ADD

リフレッシュ・グループからスナップショットを削除する

リフレッシュ・グループから 1 つ以上のスナップショットを削除するには:

1. スナップショット・サイトのターゲット・リフレッシュ・グループをクリックする。
2. グループから削除するターゲット・オブジェクトを選択する。
3. 「削除」ツールバー・ボタンをクリックする。

同等の API: DBMS_REFRESH.SUBTRACT

スナップショット・グループのリフレッシュ設定を変更する

スナップショット・グループのリフレッシュ設定を編集するには:

1. スナップショット・サイトのターゲット・リフレッシュ・グループをクリックする。
2. 「プロパティ」ツールバー・ボタンをクリックする。
3. 「リフレッシュ・グループの編集」プロパティ・シートの「スケジュール作成」ページをクリックする。
4. リフレッシュ・グループの新規リフレッシュ設定を指定する。

同等の API: DBMS_REFRESH.CHANGE

スナップショットのグループを手動でリフレッシュする

リフレッシュ・グループの即時リフレッシュを手動で実行するには:

1. スナップショット・サイトのターゲット・リフレッシュ・グループをクリックする。
2. 「プロパティ」ツールバー・ボタンをクリックする。
3. 「リフレッシュ・グループの編集」プロパティ・シートの「一般」ページをクリックする。
4. 「即時リフレッシュ」をクリックする。

同等の API: DBMS_REFRESH.REFRESH

リフレッシュ・グループの削除

リフレッシュ・グループを削除するには：

1. スナップショット・サイトのターゲット・リフレッシュ・グループをクリックする。
2. 「削除」ツールバー・ボタンをクリックする。

注意：リフレッシュ・グループを削除すると、そのグループの親の存在しないスナップショットの自動リフレッシュは行われなくなります。スナップショットをリフレッシュするには、そのスナップショットを別のスナップショット・リフレッシュ・グループに追加するか、手動でリフレッシュしなければなりません。

同等の API: DBMS_REFRESH.DESTROY

スナップショット・サイトの管理に関するその他の情報

この章のこれまでの項目では、最も一般的なスナップショット・サイトの管理プロシージャについて説明しました。あまり一般的でないスナップショット・サイトの管理プロシージャの詳細は、6-2ページの「マスター・グループおよびスナップショット・グループの上級管理テクニック」を参照してください。

データ・ディクショナリ・ビュー

アドバンスド・レプリケーション環境でスナップショット・サイトに関する情報を検索および管理するには、Replication Managerを使用する他に、次のデータ・ディクショナリ・ビューを問合せすることもできます。

スナップショット・サイト：

- DBA_SNAPSHOTS
- DBA_REFRESH
- DBA_REFRESH_CHILDREN

マスター・サイト：

- DBA_REGISTERED_SNAPSHOT
- DBA_SNAPSHOT_LOGS

競合の解消

この章のトピックは次のとおりです。

- レプリケーションの競合の概要
- 競合解消の構成の概要
- 更新の競合解消の構成
- 一意性の競合解消の構成
- 削除の競合解消の構成
- データ収束の保証
- ユーザー定義の競合解消方法
- ユーザー定義の競合通知方法
- 競合の解消情報の表示

注意: この章では、Oracle Replication Manager ツールを使用してアドバンスド・レプリケーション・システムにおける競合の解消を管理する方法の例を示します。参考として、等価のレプリケーション管理 API プロシージャを各項に記載しています。Oracle のレプリケーション管理 API の詳細は、第9章「レプリケーション管理 API リファレンス」を参照してください。

レプリケーションの競合の概要

レプリケーションの競合は、複数のサイトから同じデータを同時に更新できるアドバンスト・レプリケーション環境で発生します。たとえば、別々のサイトから発行された2つのトランザクションによって同じ行がほとんど同時に更新されたときに、競合が発生します。アドバンスト・レプリケーション環境を構成するときには、レプリケーションの競合が発生する可能性の有無を考慮する必要があります。システム設計に競合を発生させる可能性があり、競合が発生した場合には、なんらかの方法で競合を解消しないかぎりシステム・データが収束しません。

一般的に、第一に必要なのは、競合が発生しないレプリケート環境を設計することです。ほとんどのシステム設計では、いくつかのテクニックを使用することによって、レプリケートするデータの全体または大部分で競合を回避できます。しかし、多くのアプリケーションでは、データの一部を複数のサイトから更新できるようにする必要があります。その場合、レプリケーションの競合が発生する可能性に対処しなければなりません。

以降の項では、レプリケーションの競合に関する一般的な情報、レプリケーションの競合を考慮してアドバンスト・レプリケーション・システムを設計する方法、レプリケート・システム設計においてレプリケーションの競合を防止する方法、設計で競合を防止できない場合に競合を検出および解消する方法について説明します。

データ要件およびアプリケーション要件の理解

どのような種類のデータベース・アプリケーションおよびサポート・データベースを設計する場合にも、データベースまたはアプリケーション自体を設計する前にアプリケーション要件を理解しておくことが非常に重要です。たとえば、各アプリケーションは、機能境界および依存関係が正しく定義され、モジュール化されている必要があります（受入力、出荷、請求など）。さらに、サポート・データベースのデータを正規化して、アプリケーション・システムのモジュール間の隠れた依存関係を減らさなければなりません。

アドバンスト・レプリケーション環境で操作するデータベースを設計する場合、基本的なデータベース設計のルールに加えて、考慮しなければならない追加の要件があります。最初に、レプリケート・データを扱うアプリケーションの一般的な要件を考慮します。たとえば、基本的な読み専用表スナップショットだけを使用するアプリケーションでは、結果的にレプリケーションの競合が発生する可能性を完全に避けることができます。これに対して、大部分のレプリケート・データが読み専用で、ごく一部のデータ（たとえば、1つまたは2つの表、あるいは特定の表内の1つまたは2つの列）だけをすべてのレプリケーション・サイトで更新可能にする必要があるアプリケーションもあります。この場合は、レプリケーションの競合が発生したときにレプリケート・データの整合性が完全に保たれるように、競合解消方法を決定しなければなりません。

例

競合を考慮したレプリケート・データベースの設計方法をよりよく理解するために、競合の検出および解消が可能な場合と不可能な場合がある、次のような環境を考えてみます。

- 同じ商品に対して複数の予約を受け付けることができない予約システムでは、多くの場合、競合を解消できない。たとえば、コンサートの特定の座席を予約するときには、競

合を解消する方法がないため、予約システムの異なるレプリカにアクセスする異なるエージェントが、複数の顧客のために同じ座席を予約することはできません。

- 顧客管理システムでは、多くの場合、競合を解消できる。たとえば、複数のセールス担当者が、レプリケート環境内の異なるデータベースから顧客住所情報を維持することが可能です。競合が発生しても、レコードへの最新の更新を適用することによって、システムは競合する更新を解消できます。

レプリケーションの競合の種類

アドバンスド・レプリケーションでは、更新の競合および一意性の競合、削除の競合という、3種類の競合を検出および解消するための機能が用意されています。

更新の競合

更新の競合は、ある行に対する更新のレプリケーションが同じ行に対する別の更新と競合したときに発生します。異なるサイトから発生した2つの異なるトランザクションによって、同じ行がほとんど同時に更新されたときに発生することがあります。

一意性の競合

一意性の競合は、行のレプリケーションがエンティティの整合性（PRIMARY KEY 制約または UNIQUE 制約）に違反する処理を実行しようとしたときに発生します。たとえば、2つの異なるサイトから発生した2つのトランザクションで、同じ主キー値を使用してそれぞれの表レプリカに行を挿入すると何が起るかを考えてみます。この場合、トランザクションのレプリケーションが原因で一意性の競合が発生します。

削除の競合

削除の競合は、異なるサイトから2つのトランザクションが発生し、一方のトランザクションが更新または削除する行を、もう一方のトランザクションが削除する場合に発生します。

競合の防止

アプリケーション要件上可能であれば、第一に、レプリケーションの競合を完全に防止できるようなアドバンスド・レプリケーション・システムを設計しなければなりません。以降の項では、レプリケーションの競合をすべてまたは部分的に防止するためのテクニックを簡単に説明します。

プライマリ・サイトおよび動的サイト所有権データ・モデル

レプリケーションの競合の発生を防止する1つの方法として、システム内でレプリケート・データに同時更新アクセスできるサイトの数を制限する方法があります。このアプローチは、プライマリ・サイト所有権および動的サイト所有権という、2つのレプリケート・データ所有権モデルによってサポートされます。

プライマリ・サイト所有権

プライマリ所有権は、基本的な読み専用レプリケーション環境でサポートされるレプリケート・データ・モデルです。プライマリ所有権によって、一連のレプリケート・データに更新アクセスできるサーバーは1つに限定されるため、すべての種類のレプリケーションの競合が発生しなくなります。

表レベルでデータの所有権を制御するかわりに、アプリケーションで水平パーティション化および垂直パーティション化の機能を使用すると、データの静的所有権をより細かく設定できます。たとえば、サイト別の基準でレプリケート表の特定の列または行に更新アクセスすることができます。

追加情報：Oracle の基本的な読み込み専用レプリケーション機能の詳細は、第 2 章を参照してください。

動的サイト所有権

動的所有権レプリケート・データ・モデルは、プライマリ・サイト所有権よりも制限の少ないモデルです。動的所有権の場合、データ・レプリカを更新する権利はサイトからサイトへ移動しますが、ある時点で特定のデータに更新アクセスできるサイトは 1 つだけであることが保証されます。ワークフロー・システムに動的所有権の概念がわかりやすく示されています。たとえば、関連する部門のアプリケーションで、注文を更新できる時期を判断するために、製品注文のステータス・コード (ENTERABLE、 SHIPPABLE、 BILLABLE など) を読み込むことができます。

追加情報：動的所有権データ・モデルの使用方法的詳細は、7-26 ページの「動的所有権の競合防止の使用法」を参照してください。

特定タイプの競合の防止

プライマリ・サイト所有権および動的所有権データ・モデルを使用するとアプリケーション要件に対して制限が多すぎる場合には、共有所有権データ・モデルを使用する必要があります。その場合でも、通常は、特定のタイプの競合を防止するために簡単な計画を使用する必要があります。

一意性の競合の防止

一意性の競合が発生しないようにアドバンスド・レプリケーション環境を構成するのは非常に簡単です。たとえば、互いに重複しない一連の順序番号を生成するように、各サイトでレプリカ順を作成できます。ただし、この方法を使用する場合、サイト数の増加またはレプリケート表内の項目数の増加につれて問題が発生します。かわりに、各サイトのレプリカ順で全範囲の順序値を使用することを許可し、さらに一意のサイト識別子を複合主キーの一部として組み入れることができます。

削除の競合の防止

削除の競合は、すべてのレプリケート・データ環境で必ず防止しなければなりません。一般的に、非同期の共有所有権データ・モデル環境で動作するアプリケーションでは、DELETE 文を使用して行を削除してはなりません。かわりに、アプリケーションで行に削除のマークを付けて、論理的に削除された行をプロシージャ・レプリケーションを使用して定期的にパージするようにシステムを構成できます。

更新の競合の防止

アドバンスド・レプリケーション・システムでは、一意性の競合と削除の競合の可能性を排除するように試みた後、さらに更新の競合の発生数を少なく抑える必要があります。ただし、共有所有権データ・モデルでは、更新の競合をすべて防止することはできません。更新の競合をすべて防止することができない場合は、発生する可能性があるレプリケーションの競合のタイプを正確に理解し、競合が発生したときにそれを解消できるようにシステムを構成する必要があります。

マスター・サイトでの競合の検出

アドバンスド・レプリケーション・システムの各マスター・サイトは、レプリケーションの競合が発生するとそれを自動的に検出し、解消します。たとえば、あるマスター・サイトがシステム内の別のマスター・サイトに遅延トランザクション・キューを送信するとき、レプリケーションの競合があれば、呼び出された受信サイトのリモート・プロシージャによって自動的に検出されます。

スナップショット・サイトが関連マスター・サイトに遅延トランザクションを送信するときには、受信マスター・サイトによって競合が検出および解消されます。スナップショット・サイトは、スナップショット・リフレッシュを実行することによってデータをリフレッシュします。このリフレッシュの仕組みによって、リフレッシュ完了後には、すべての競合解消結果を含むスナップショットのデータが関連マスターのデータと同一になります。したがって、スナップショット・サイトでレプリケーションの競合の検出または解消を実行する必要はありません。

Oracle での異なるタイプの競合の検出方法

アドバンスド・レプリケーション・システムの受信マスター・サイトでは、次のように更新の競合および一意性の競合、削除の競合が検出されます。

- レプリケートされた行の古い値（変更前の値）と受信サイトの同じ行の現在の値との間に違いがある場合、受信サイトで更新の競合が検出されます。
- レプリケート行の INSERT または UPDATE の実行中に一意制約違反が発生した場合、受信サイトで一意性の競合が検出されます。
- 行の主キーが存在しないために受信サイトが UPDATE または DELETE 文の行を見つけられない場合、受信サイトで削除の競合が検出されます。

注意：行の更新の競合を検出および解消するには、伝播サイトから受信サイトに行の新旧バージョンについてある特定の量のデータを送信する必要があります。パフォーマンスを最大にするには、更新の競合の検出および解消サポートに使用するデータの量を調整します。詳細は、5-34ページの「更新の競合解消のためのデータ伝播の最小化」を参照してください。

競合検出時の行の識別

レプリケーションの競合を正確に検出するには、データ・レプリケーション時に別々のサイトで対応する行を個々に識別して、一致させることができればなりません。通常、アドバンスド・レプリケーション機能では、表の主キーを使用して行が一意に識別されます。表に主キーがない場合は、代替キーを指定する必要があります。代替キーとは、データ・レプリケーション時に表の行を識別するために使用できる列または列のセットのことです。

警告：アプリケーションで表の識別列を更新できないようにする必要があります。そうすることによって、行を識別してレプリケート・データの整合性を保つことができます。

競合の解消

受信マスター・サイトでレプリケーションの競合が発生した場合には、競合を解消して、システム全体のデータを確実に収束させなければなりません。「データ収束」とは、レプリケート・データを管理するすべてのサイトが最終的に同じ一連の情報に一致するということです。発生したレプリケーションの競合を解消しないで放置すると、サイト間にデータの不整合が生じます。さらに、次々と望ましくない影響が出る場合もあります。不整合によってさらに競合が発生し、それによってさらに不整合が発生する、という悪循環になる可能性があります。

システムですべてのタイプのレプリケーションの競合を防止できない場合は、Oracle の自動競合解消機能を使用するようにシステムを構成できます。以降の項では、各タイプのレプリケーションの競合に対する Oracle の競合解消機能について説明します。

自動競合解消方法と手動競合解消方法

競合は、発生したら必ず Oracle の自動競合解消機能を使用して解消しなければなりません。レプリケート表の自動競合解消を構成していない場合は、各サイトで競合の記録だけが行われます。この場合は、レプリケート・データの整合性を保つために、手動で競合を解消しなければなりません。手動で競合解消を実行するのは非常に手間がかかります。また、手動で競合解消を実行するために発生した遅れによってデータの不整合が生じ、前の項で説明したように、その不整合によってさらに次々と影響が出る可能性があります。

更新の競合解消と列グループ

更新の競合を検出および解消するために、列グループが使用されます。列グループは、レプリケート表の1つ以上の列を論理的にグループ化したものです。レプリケート表の各列は、1つの列グループの一部になります。マスター定義サイトでレプリケート表を構成するときに、列グループを作成して、各グループに列および対応する競合解消方法を割り当てることができます。

複数の列グループでのデータ整合性の保証

列グループを用いると、異なるタイプのデータに異なる競合解消方法を指定できます。たとえば、数値データの多くは算術的な解消方法に向いており、文字データの多くはタイムスタンプ解消方法に向いています。ただし、列グループの列を選択するときには、十分に考えた上で列をグループ化することが重要です。表内の複数の列が相互に一貫性を保っていない場合、データの整合性を保証するために、それらの列を同じ列グループに入れます。たとえば、顧客表の郵便番号の列に1つの解消方法を使用し、都市名の列には別の解消方法を使用している場合、サイトは都市名に一致する郵便番号に収束します。したがって、一般的には、競合解消方法が住所全体に対して適用されるように、1つの住所のすべてのコンポーネントを単一の列グループに入れなければなりません。

シャドウ列グループ

デフォルトで、各レプリケート表にシャドウ列グループがあります。表のシャドウ列グループには、特定の列グループ内にはない列がすべて含まれます。表のシャドウ・グループには、競合解消方法を割り当てることはできません。このため、列に競合解消方法を使用する必要がある場合には、列グループに必ずその列を入れてください。

一意性の競合解消

通常、アドバンスド・レプリケーション・システムおよび関連アプリケーションは、一意性の競合が起こらないように設計しなければなりません。しかし、一意性の競合を防止できない場合には、レプリケート表の PRIMARY KEY 制約または UNIQUE 制約に1つ以上の競合解消方法を割り当て、一意性の競合の発生時に競合を解消できます。Oracle には、ビルトインの一意性の競合解消方法がいくつか用意されています。一般的には、解消した一意性の競合の正確度を検証するために、これらの方法を競合通知とともに使用します。

削除の競合解消

アドバンスド・レプリケーション環境は、必ず削除の競合を防止するように設計しなければなりません。アプリケーションの設計上、削除の競合を防止すると制限が大きすぎる場合は、ユーザー定義の削除の競合解消方法を作成して、それをレプリケート表に割り当てることができます。Oracle には、ビルトインの削除の競合解消方法は用意されていません。ユーザー独自の競合解消方法の作成の詳細は、5-39 ページの「ユーザー定義の競合解消方法」を参照してください。

競合解消方法

レプリケーションの競合を自動的に解消するために、1つ以上の競合解消方法を割り当てることができます。Oracle には、競合を解消するために使用できる、多数のビルトイン競合解消方法が用意されています。必要な場合は、ユーザー独自の競合解消方法を作成できます。以降の項では、ビルトインおよびユーザー定義の競合解消方法の詳細について説明します。

ビルトインの更新の競合解消方法

Oracle では、列グループに割当て可能なビルトインの更新の競合解消方法として、次の方法が用意されています。

- 上書きおよび廃棄値
- 最小値および最大値

- 最古および最新のタイムスタンプ値
- 加算および平均値
- 優先順位グループおよびサイト優先順位

追加情報：ビルトインの更新の競合解消方法の詳細は、5-13 ページの「ビルトインの更新の競合解消方法」を参照してください。

ビルトインの更新の競合解消方法には、レプリケート・データの収束ができるものとできないものがあります。たとえば、加算の競合解消方法では、3 つ以上のマスター・サイトで管理されているレプリケート・データが収束しますが、最古のタイムスタンプ方法では収束しません。

追加情報：それぞれのビルトインの競合解消方法のデータ収束プロパティの詳細は、5-31 ページの「データ収束の保証」を参照してください。

ビルトインの一意性の競合解消方法

Oracle では、PRIMARY KEY 制約および UNIQUE 制約に割当て可能なビルトインの一意性の競合解消方法として、次の方法が用意されています。

- サイト名を重複値に追加する。
- 順序を重複値に追加する。
- 重複値を廃棄する。

Oracle のビルトインの一意性の競合解消方法では、レプリケート環境のデータは収束しません。PRIMARY KEY 制約違反および UNIQUE 制約違反を解決するテクニックを提供するだけです。したがって、Oracle の一意性の競合解消方法を使用して競合を解消する場合は、一意性の競合の発生を知らせる通知メカニズムを併用して、必要に応じて手動でレプリケート・データを収束しなければなりません。

追加情報：ビルトインの一意性の競合解消方法の詳細は、5-29 ページの「ビルトインの一意性の競合解消方法」を参照してください。

ビルトインの競合解消方法の制限

Oracle のビルトインの競合解消方法では、次の状況はサポートされません。

- 削除の競合
- 主キー（または識別キー）列への変更
- 競合解消のために列に NULL を代入する場合
- 参照整合性制約の違反

これらの状況については、独自の競合解消方法を用意するか、またはエラー・トランザクションを手動で解決する方法を決定してください。

ユーザー定義の競合解消方法および通知方法

Oracle のビルトインの競合解消方法に加えて、競合解消の補助的な手段として、競合のロギングおよび競合の通知の機能も使用できます。Oracle では、ユーザー定義の競合解消方法をコールするようにレプリケート表を構成し、競合情報の記録や、競合が解消できない場合の通知を行うことができます。すべての競合について通知するか、または解消できない競合についてだけ通知するように、列グループおよび制約、レプリケート表を構成できます。

追加情報: ユーザー独自の競合解消方法および通知方法の作成の詳細は、5-39ページの「ユーザー定義の競合解消方法」を参照してください。

複数の競合解消方法の使用

列グループの競合解消方法を複数指定しておくと、競合解消が失敗したときに、他の異なる方法で解消できます。競合を解消しようとするときに、指定した順に各グループの方法が実行されます。更新の競合を解消するときに Oracle が使用するアルゴリズムは、次のとおりです。

1. 受信マスター・サイトで、最初の列グループについて、グループの各フィールドが変更されているかどうかをチェックする。変更されている場合には、古い値、新しい値、現在の値の間に競合が起きているかどうかを判別する。
2. 競合が起きていない場合は、次の列グループに進む。競合が起きている場合は、その列グループに対して、最も小さい順序番号が割り当てられている競合解消方法と呼び出す。
3. その競合解消方法で競合が解消された場合は、状態の判別が保留されている列の適正な値を保持する。
4. その競合解消方法で競合が解消できない場合は、競合が解消されるまで、または使用可能な方法がなくなるまで、順に解消方法を実行する。
5. すべての列グループ（シャドウ列グループを含む）を評価し、競合を解消した後、列の新しい値を保存する。
6. 割り当てた方法で競合が解消できない場合は、受信サイトのレプリケーション・カタログにトランザクション全体をエラー・トランザクションとしてロギングし、ローカル行の値は変更しない。

複数の競合解消方法を使用するのには、次のような理由があります。

- 指定した方法で競合が解消できなかった場合に、代替の競合解消方法を使用する。
- レプリケーションの競合が発生したときに、自動通知を受信する。

以降の項で、それぞれの詳細を説明します。

注意: 一意性の競合を解消するために、PRIMARY KEY 制約または UNIQUE 制約に複数の競合解消方法を割り当てることができます。また、削除の競合を解消するために、レプリケート表に複数の競合解消方法を割り当てることができます。

予備のための複数の競合解消方法の使用

状況によっては、列グループおよび制約、表に設定した競合解消方法で競合を解消できない場合があります。できる限り、1 つ以上の代替方法を順序付きで指定しておく必要があります。そうすることで、手動の解消方法を実行しなくても競合を解消できる可能性が高くなります。

システム定義の競合解消方法には、すべての状況における競合解消を保証できないものがあります。たとえば、最新タイムスタンプによる更新の競合解消方法は、特殊なタイムスタンプ列を使用して最新の変更を判別し、適用します。可能性は少ないですが、発行側サイトの行と別のサイトの行がまったく同じ時刻に変更された場合、Oracle は時間に関する情報を秒刻みで記録するため、最新のタイムスタンプの方法で競合を解消することはできません。サイト優先順位など、予備の更新の競合解消方法を宣言することで、競合を自動的に解消することが可能になります。

通知のための複数の競合解消方法の使用

複数の競合解消方法を使用するもう 1 つの理由は、競合情報の記録や競合を解消できない場合の通知を行うユーザー定義の方法をコールできるようにすることです。たとえば、最初にユーザー定義の競合通知方法をコールし、次にサイト名の追加による一意性の競合解消方法を使用して競合を解消するというように、PRIMARY KEY 制約を構成することも可能です。

追加情報：競合の通知の詳細は、5-43 ページの「ユーザー定義の競合通知方法」を参照してください。

競合解消の構成の概要

アドバンスド・レプリケーション・システムで競合解消が必要だと判断した場合は、最初に競合解消方針を設計し、次に、レプリケート表の作成時に競合解消方針をインプリメントする必要があります。以降の項では、競合解消の構成の各段階を完了するために必要な手順の概要を説明します。

競合解消の設計および準備のガイドライン

競合解消計画を設計および準備するときは、次のガイドラインに従ってください。

- データを分析して、更新の競合解消に適した列グループと、各列グループに適した競合解消方法を判断する。一意性の競合および削除の競合を防止できない場合は、これらの競合解消方法も計画する必要があります。
- 一部のビルトインの更新の競合解消方法を使用するには、前もって特別な準備の手順を実行しなければならない。次に例を示します。
 - 最古および最新のタイムスタンプの更新の競合解消方法を使用する場合、レプリケート表に、タイムスタンプの比較に使用可能な DATE 列がなければならない。さらに必要に応じて、同時更新の問題に対処するために、レプリケート表にタイムスタンプのメンテナンス・トリガーを追加します。タイムスタンプの競合解消方法の詳細は、5-15 ページの「最古および最新のタイムスタンプ」を参照してください。
 - レプリケート表内の列グループで、競合解消にサイト優先順位または優先順位グループを使用する場合、各サイトまたは値に関して優先順位を定義する。優先順位

グループおよびサイト優先順位グループの構成方法の詳細は、5-18 ページの「更新の競合解消のための優先順位グループの使用」および 5-23 ページの「更新の競合解消のためのサイト優先順位グループの使用」をそれぞれ参照してください。

- 必要に応じて、競合通知方法を設定する。
 - 各マスター・サイトで、競合通知情報を格納する表を作成する。
 - 表に競合通知情報を記録する PL/SQL プロシージャを作成する。
 - ユーザー定義の競合解消方法をパッケージに追加するか、電子メールによる通知を自動化する方法を追加する（オプション）。

競合通知の構成の詳細は、5-43 ページの「ユーザー定義の競合通知方法」を参照してください。

競合解消のインプリメント

計画の後、Oracle の Replication Manager およびレプリケーション管理 API を使用して、マスター・グループ内のレプリケート表の競合解消を構成します。一般的な手順は次のとおりです。

1. マスター・グループのレプリケーション・アクティビティを中断する。
2. マスター定義サイトで、マスター・グループのレプリケート表に対して使用する競合解消を構成する。たとえば、表に対して更新の競合解消を構成する場合は、Replication Manager を使用して必要な列グループを作成し、列グループに更新の競合解消方法を割り当てます。
3. 競合解消の構成が完了した後、レプリケート表またはマスター・グループ内のすべてのオブジェクトに対するレプリケーション・サポートを再生成する。
4. すべての変更を行い、競合解消の構成をテストした後、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する。

以降の項では、更新および一意性、削除の競合解消方法の構成について説明します。

更新の競合解消方法の構成

一般的なアドバンスド・レプリケーション環境では、一意性の競合および削除の競合は発生しません。したがって、システムの設計および構成で最大の注意を払う必要があるのは、更新の競合です。Oracle のアドバンスド・レプリケーション機能では、列グループを使用して更新の競合を検出し、解消します。以降の項では、レプリケート表の列グループの構成方法と、列グループに使用する更新の競合解消方法について説明します。

列グループの作成

マスター定義サイトのマスター・グループに表を追加した後、グループのレプリケーション・アクティビティを中断している間に、表の列グループを構成し競合解消を設定できます。

1. マスター定義サイトのマスター・グループにあるターゲット表をクリックする。

2. 「プロパティ」ツールバー・ボタンをクリックする。
3. 「レプリケーション・オブジェクトの編集」ダイアログの「競合解消」ページをクリックする。
4. 「追加」(上のボタン)をクリックして「列グループの作成」ダイアログを表示し、ターゲット表に新規の列グループを作成する。列グループを作成するときは、列グループの名前およびコメント(オプション)、グループのメンバーに入れる1つ以上の列を指定してください。
5. 「OK」をクリックし、新規の列グループを作成する。
6. 列グループに使用する更新の競合解消方法を選択する。詳細は、5-13 ページの「ビルトインの更新の競合解消方法」を参照してください。

同等の API: DBMS_REPCAT.MAKE_COLUMN_GROUP

列グループの列の追加および削除

マスター・グループのレプリケーション・アクティビティが中断している間は、表の列グループの列を追加または削除できます。

1. マスター定義サイトのマスター・グループにあるターゲット表をクリックする。
2. 「プロパティ」ツールバー・ボタンをクリックする。
3. 「レプリケーション・オブジェクトの編集」ダイアログの「競合解消」ページをクリックする。
4. 編集する列グループをクリックする。
5. 「編集」(上のボタン)をクリックして、「列グループの編集」ダイアログを表示する。
6. 列グループに列を追加する場合は、使用可能な列のリストから列を選択し、「追加」をクリックする。列グループから列を削除する場合は、グループの列リストから列を選択し、「削除」をクリックする。

同等の API: DBMS_REPCAT.ADD_GROUPED_COLUMN、
DBMS_REPCAT.DROP_GROUPED_COLUMN

列グループの削除

マスター・グループのレプリケーション・アクティビティが中断している間は、表の列グループを削除できます。

1. マスター定義サイトのマスター・グループにあるターゲット表をクリックする。
2. 「プロパティ」ツールバー・ボタンをクリックする。
3. 「レプリケーション・オブジェクトの編集」ダイアログの「競合解消」ページをクリックする。
4. 削除する列グループをクリックする。

5. 「削除」(上のボタン)をクリックして、表からターゲット列グループを削除する。

同等の API: DBMS_REPCAT.DROP_COLUMN_GROUP

更新の競合解消方法の管理

マスター・グループのレプリケーション・アクティビティが中断している間は、Oracle の Replication Managerを使用して、レプリケート表の列グループに対する更新の競合解消方法の割当ておよび削除、順序付けができます。

1. マスター定義サイトのマスター・グループにあるターゲット表をクリックする。
2. 「プロパティ」ツールバー・ボタンをクリックする。
3. 「レプリケーション・オブジェクトの編集」ダイアログの「競合解消」ページをクリックする。
4. 管理するターゲット列グループをクリックする。

この時点で、選択した列グループに対して更新の競合解消方法の割当てまたは削除、順序付けができます。以降の項では、それぞれの手順を説明します。

注意：表の競合解消を作成または編集する前に、レプリケーション・アクティビティを中断しなければなりません。また、「競合解消」ページで実行したすべての変更内容は、ただちにレプリケート環境にコミットされます。

更新の競合解消方法の割当て

選択した列グループに新しい更新の競合解消方法を割り当てるには、「追加」(下のボタン)をクリックして「競合解消方法の追加」ダイアログを表示し、ターゲット列グループに新しい更新の競合解消方法を追加します。

注意：一部の更新の競合解消方法では、列グループへの割当て前に準備作業（たとえば、優先順位グループ）を行う必要があります。各タイプの競合解消方法と、それらの方法を使用するための特別な要件については、この章の後の項で説明します。

同等の API: DBMS_REPCAT.ADD_UPDATE_RESOLUTION

更新の競合解消方法の削除

選択した列グループから更新の競合解消方法を削除するには、「削除」(下のボタン)をクリックして、列グループから選択した競合解消方法を削除します。

同等の API: DBMS_REPCAT.DROP_UPDATE_RESOLUTION

列グループの更新の競合解消方法の順序付け

選択した列グループに対して競合解消方法を適用する順序を設定または変更するには、選択した解消方法の順位を上下の矢印キーを使用して指定し、「再編」をクリックして新しい順序を適用します。

ビルトインの更新の競合解消方法

以降の項では、更新の競合を解消するために使用できる Oracle のビルトインの方法を説明します。次の方法があります。

- 加算および平均
- 最小値および最大値
- 最古および最新のタイムスタンプ
- 上書きおよび廃棄
- 優先順位グループおよびサイト優先順位

以降の項では、ビルトインの更新の競合解消方法のそれぞれについて、詳しく説明します。

注意：競合解消方法を割り当てる場合は、データの収束を保証し、業務で使用するデータとして適した結果になるようにする必要があります。データ収束および Oracle のビルトインの競合解消方法の詳細は、5-31 ページの「データ収束の保証」を参照してください。

加算および平均

加算および平均による方法は、数値型の単一列だけで構成される列グループに作用します。

- 加算による方法では、発行元サイトの古い値と新しい値の差を、接続先サイトの現在の値に加算します。

$$\text{現在の値} = \text{現在の値} + (\text{新しい値} - \text{古い値})$$

加算による競合解消方法では、マスター・サイトがいくつあっても値は収束します。

- 平均による競合解消方法では、発行元サイトの新しい列値と接続先サイトの現在の値の平均をとります。

$$\text{現在の値} = (\text{現在の値} + \text{新しい値}) / 2$$

レプリケート環境に複数のマスターが存在する場合、平均による方法では収束は保証できません。この方法は、単一マスター・サイトおよび複数の更新可能なスナップショットに対して有効です。

最小値および最大値

アドバンスド・レプリケーション機能が列グループを使用して競合を検出し、最小値による競合解消方法をコールすると、列グループの指定された列について、発行元サイトの新しい値と接続先サイトの現在の値が比較されます。最小値による競合解消方法を選択する場合は、この列を指定しなければなりません。

指定した列の新しい値が現在の設定値より小さい場合は、発行元サイトの列グループは接続先サイトに適用されます（その行のその他のエラーはすべて解消されたとみなされます）。指定の列の新しい値が現在の設定値より大きい場合は、現在の列グループの設定値を変更しないでくと競合は解消されます。

注意：指定した列の 2 つの値が同じである場合（たとえば、指定した列が競合が発生した列でない場合）競合は解消されず列グループの列の値は変更されません。この場合は、予備の競合解消方法を指定してください。

最大値による方法は、発行元サイトの指定した列の値が接続先サイトの指定した列の値よりも大きい場合だけ適用されます。それ以外は、最小値による方法と同じです。

列グループ内の列のデータ型に制約はありません。次の条件が満たされる場合にだけ、3つ以上のマスター・サイトで収束が保証されます。

- 最大値による方法では、列値が常に増加している場合。
- 最小値による方法では、列値が常に減少している場合。

注意：競合解消が制約によって妨げられる可能性があるため、CHECK 制約を使用して常に増加する値の制限を施行することは避けてください。

最古および最新のタイムスタンプ

最古および最新のタイムスタンプによる方法は、最小値および最大値による方法を応用したものです。タイムスタンプを使用するには、レプリケート表内の DATE 型の列を指定しなければなりません。アプリケーションで列グループ内の列を更新するときは、ローカル SYS-DATE で指定されたタイムスタンプ列の値も更新する必要があります。他のサイトから適用される変更については、タイムスタンプ値は発行元サイトのタイムスタンプ値に設定されます。

最新のタイムスタンプ競合解消方法の適切な使用例を次に示します。

1. フェニックスの顧客が地元の営業担当者に電話して、自分の住所情報を更新します。
2. 電話を切った後で、顧客は営業担当者に伝えた郵便番号が間違っていたことに気がきます。
3. 顧客は営業担当者に電話して正しい郵便番号を伝えようとしてますが、担当者に電話がつながりません。
4. 顧客はニューヨーク本社に電話します。フェニックス・サイトではなく、ニューヨーク・サイトで住所情報が正しく更新されます。
5. ニューヨーク本社とフェニックス営業所を接続するネットワークが一時的にダウンします。
6. ニューヨークとフェニックスを接続するネットワークが回復すると、Oracle は同じ住所に対する 2 つの更新を認識して、各サイトでの競合を検出します。
7. Oracle では、最新タイムスタンプによる方法を使用して、最も最近の更新内容を選択し、正しい郵便番号を住所に適用します。

最古のタイムスタンプによる方法では、最も古いタイムスタンプを持つサイトからの変更が適用され、最新タイムスタンプによる方法では、最も新しいタイムスタンプを持つサイトからの変更が適用されます。

注意：タイムスタンプによる競合解消方法を使用する場合は、2 つのサイトのタイムスタンプが同一であるときにコールされる、サイト優先順位などの予備方法を指定しなければなりません。

構成に関するその他の情報

タイムスタンプによる解消方法を使用する場合は、レプリケート・データを管理している異なるサイトの標準時間帯を慎重に考慮してください。たとえば、レプリケート環境が複数の標準時間帯にわたっている場合、システムを使用するアプリケーションは、すべてのタイムスタンプをグリニッジ標準時 (GMT) などの共通の標準時間帯に変換しなければなりません。さらに、システム内の 2 つのサイトのシステム時計が正しく同期化されていない場合、タイムスタンプの比較が不正確になり、アプリケーション要件を満たすことができない可能性があります。

EARLIEST または LATEST タイムスタンプによる更新の競合解消方法を使用する場合、タイムスタンプ列のメンテナンスには 2 つの方法があります。

- 各アプリケーションに、タイムスタンプを同期化する論理を組み込む。
- すべてのアプリケーションのタイムスタンプを自動的に同期化させる、レプリケート表のトリガーを作成する。

時刻は、秒数を増加する値としてカウントします。タイムスタンプのメカニズムを正しく指定し、2 つのサイトのタイムスタンプが同一である場合に備えて予備方法を設定してあると想定すると、(最大値による方法と同様に) 最新タイムスタンプによる方法で収束が保証されます。ただし、最古のタイムスタンプによる方法では、3 つ以上のマスターについての収束は保証されません。

追加情報: タイムスタンプおよびサイト・メンテナンス・トリガーの例は、5-26 ページの「サンプル・タイムスタンプおよびサイト・メンテナンス・トリガー」を参照してください。

上書きおよび廃棄

上書きおよび廃棄による方法では、発行元サイトまたは接続先サイトのいずれかの値が無視されるため、複数のマスター・サイトでの収束は保証されません。これらの方法は、単一マスター・サイトと複数のスナップショット・サイトの間で使用するか、またはなんらかのユーザー定義による通知機能とともに使用してください。

たとえば、主に問合せ用に使用する単一マスター・サイトがあり、すべての更新をそのスナップショット・サイトで実行する場合は、上書き方法を選択できます。次の場合にも、上書きおよび廃棄による方法が役立ちます。

- 最大の目的がデータ収束である場合。
- マスター・サイトが 1 つしかない場合。
- 更新の選択について、業務上の規則が特にならない場合。
- 複数のマスター・サイトがあるときに、DEFERROR ビューで競合をロギングしてその解消をローカル・データベース管理者にまかせるのではなく、通知機能を使用して、データが正しく適用されていることを確認する人物に通知する場合。

上書きによる方法では、接続先サイトの現在の値を、発行元サイトの新しい値で上書きします。逆に、廃棄による方法では、発行元サイトの新しい値が無視されます。

優先順位グループおよびサイト優先順位

優先順位グループを使用すると、特定の列の可能な値それぞれに優先順位レベルを割り当てることができます。競合が検出されると、「優先順位」列の値が小さい表は、より高い優先順位を持つ表のデータによって更新されます。

図 5-1 に示すように、各優先順位グループのメンバーに割り当てた優先順位レベル（「優先順位」列に入れることのできる値）は、REPPRIORITY ビューに表示されます。「優先順位」列のすべての可能な値に対して優先順位を指定しなければなりません。

図 5-1 優先順位グループの使用方法

CUSTOMER 表				
custno	name	addr1	addr2	site
153	Kelly	104 First St.	Jones, NY	new_york.world
118	Klein	22 Iris Ln.	Planes, NE	houston.world
121	Lee	71 Blue Ct.	Aspen, CO	houston.world
204	Potter	181 First Av.	Aspen, CO	houston.world
.
.
.

REPPRIORITY ビュー				
...	priority-group	priority	...	value
	site-priority	1		houston.world
	site-priority	2		new_york.world
	order-status	1		ordered
	order-status	2		shipped
	order-status	3		billed

REPPRIORITY ビューには、現在の位置で定義されたすべての優先グループの値が表示されます。図 5-1 の例では、サイト優先順位と順序ステータスという2つの異なる優先順位グループが示されています。CUSTOMER 表は、サイト優先順位グループを使用しています。

Replication Managerを使用して優先順位グループによる更新の競合解消方法を選択するには、その前に、表内の「優先順位」列を指定しなければなりません。

追加情報：更新の競合解消のために優先順位を構成する方法の詳細は、5-18 ページの「更新の競合解消のための優先順位グループの使用」を参照してください。

サイト優先順位は、特殊な種類の優先順位グループです。サイト優先順位を使用する場合、指定する「優先順位」列は、更新が発行されたサイトのグローバル・データベース名に自動的に更新されます。各データベース・サイトに割り当てた優先順位は、REPPRIORITYビューに表示されます。あるサイトが最も正確な情報を持っている可能性が高いとみなされる場合、サイト優先順位が役に立つことがあります。たとえば、図 5 - 1 では、ニューヨーク・サイト（優先順位値=2）は本社で、ヒューストン・サイト（優先順位値=1）は販売本社です。本社オフィスは、各顧客のクレジット限度額について、販売本社より正確な情報を持っているとみなされます。

注意：REPPRIORITY ビューの優先グループ列には、サイト優先グループと順序ステータス・グループの両方が表示されます。

サイト優先順位を使用している場合、3 つ以上のマスターでの収束は保証されません。しかし、優先順位グループを使用している場合、「優先順位」列の値が必ず増加するなら、3 つ以上のマスターでの収束を保証できます。つまり、優先順位列の値は、イベントの順序、たとえば、ordered、shipped、billed に対応します。

優先順位グループと同様、Replication Managerを使用して列グループに対するサイト優先順位による競合解消方法を選択するには、その前に準備作業を実行しておく必要があります。

追加情報：サイト優先順位の構成方法は、5-23 ページの「更新の競合解消のためのサイト優先順位グループの使用」を参照してください。

更新の競合解消のための優先順位グループの使用

優先順位グループによる方法を使用して更新の競合を解消する場合、Replication Managerを使用して列グループに優先順位グループ方法を割り当てる前に、レプリケーション管理 API を使用して特別な手順を実行する必要があります。最初に、優先順位グループを作成しなければなりません。優先順位グループを作成するには、次の手順を実行します。

1. 優先順位グループの名前と、グループ内の値のデータ型を定義します。
2. 「優先順位」列に格納される可能性のある各値について優先順位を定義します。この情報は、REPPRIORITY ビューに表示されます。

複数の表で、1 つの優先順位グループを使用できます。したがって、優先順位グループに選択する名前は、マスター・グループ内において一意でなければなりません。この優先順位に対応する列は、別の表では別の名前を持つことができます。

表に優先順位グループの競合解消方法を追加するときは、その表内で、特定の優先順位グループに対応付けられている列を示さなければなりません。したがって、優先順位グループには、その優先順位グループに関連付けられた列全部に入り得る値をすべて入れなければなりません。

たとえば、レプリケート表 INVENTORY があり、その表に VARCHAR2 型の STATUS という列があり、さらにその列には 3 つの可能な値、ORDERED および SHIPPED、BILLED があるとします。ここで、この STATUS 列の値に基づいて更新の競合を解消するとします。対応付けられたマスター・グループ ACCT のレプリケーション・アクティビティを中断した後、マスター定義サイトで次の手順を実行し、INVENTORY 表に対して優先順位グループによる競合解消方法を構成します。

1. Replication Manager を使用して、INVENTORY 表に STATUS 列を含む列グループを作成します。
2. マスター・グループ ACCT に対応付けられた STATUS 優先順位グループを作成して、組み入れます。これには、次の API コールを使用します。

```
DBMS_REPCAT.DEFINE_PRIORITY_GROUP(
    gname          => 'acct',
    pgroup         => 'status',
    datatype       => 'varchar2');
DBMS_REPCAT.ADD_PRIORITY_VARCHAR2(
    gname          => 'acct',
    pgroup         => 'status',
    value          => 'ordered',
    priority       => 1);
DBMS_REPCAT.ADD_PRIORITY_VARCHAR2(
    sname          => 'acct',
    pgroup         => 'status',
    value          => 'shipped',
    priority       => 2);
DBMS_REPCAT.ADD_PRIORITY_VARCHAR2(
    sname          => 'acct',
    pgroup         => 'status',
    value          => 'billed',
    priority       => 3);
```

3. Replication Manager を使用して、ターゲット列グループに PRIORITY GROUP 競合解消方法を指定します。

注意：優先順位グループを作成または管理する前に、Replication Manager を使用して、マスター定義サイトの関連するマスター・グループのレプリケーション・アクティビティを中断してください。いずれかの方法で優先順位グループを管理した後は、マスター・グループのレプリケーション・アクティビティを再開する前に、対応付けられたレプリケート表のレプリケーション・サポートを再生成してください。

以降の項では、優先順位グループの管理方法を詳しく説明します。

優先順位グループの作成

DBMS_REPCAT パッケージ内の DEFINE_PRIORITY_GROUP プロシージャを使用して、次の例に示すように、マスター・グループに新規の優先順位グループを作成します。

```
DBMS_REPCAT.DEFINE_PRIORITY_GROUP(
    gname          => 'acct',
```

```
pgroup      => 'status',  
datatype    => 'varchar2');
```

この例では、ACCT マスター・グループに STATUS という名前の優先順位グループを作成します。この優先順位グループのメンバーには、VARCHAR2 型の値があります。

追加情報: DEFINE_PRIORITY_GROUP プロシージャのパラメータは、表 9-122 を参照してください。例外は、表 9-123 を参照してください。

優先順位グループへのメンバーの追加

DBMS_REPCAT パッケージ内には、メンバーを優先順位グループに追加するプロシージャが数種類あります。これらのプロシージャは、ADD_PRIORITY_type の形になります。type は、優先順位グループを作成したときに指定したデータ型と同じ値です。

- ADD_PRIORITY_CHAR
- ADD_PRIORITY_VARCHAR2
- ADD_PRIORITY_NUMBER
- ADD_PRIORITY_DATE
- ADD_PRIORITY_RAW
- ADD_PRIORITY_NCHAR
- ADD_PRIORITY_NVARCHAR2

コールする必要があるプロシージャは、「優先順位」列のデータ型によって決まります。このプロシージャは、「優先順位」列の各有効値ごとに 1 度コールしてください。

次の例では、値 SHIPPED を STATUS 優先順位グループに追加します。

```
DBMS_REPCAT.ADD_PRIORITY_VARCHAR2(  
    gname      => 'acct',  
    pgroup     => 'status',  
    value      => 'shipped',  
    priority   => 2);
```

追加情報: ADD_PRIORITY_datatype プロシージャのパラメータは、表 9-77 を参照してください。例外は、表 9-78 を参照してください。

メンバーの値の変更

DBMS_REPCAT パッケージ内には、優先順位グループのメンバーの値を変更するための異なるプロシージャがいくつかあります。これらのプロシージャは、ALTER_PRIORITY_type の形になります。type は、優先グループを作成したときに指定したデータ型と同じ値です。

- ALTER_PRIORITY_CHAR
- ALTER_PRIORITY_VARCHAR2

- ALTER_PRIORITY_NUMBER
- ALTER_PRIORITY_DATE
- ALTER_PRIORITY_RAW
- ALTER_PRIORITY_NCHAR
- ALTER_PRIORITY_NVARCHAR2

コールする必要があるプロシージャは、「優先順位」列のデータ型によって決まります。優先順位グループのメンバーは、特定の値にそれぞれ優先順位がつけられた形になっているので、このプロシージャによって指定した優先順位の値を変更できます。

次の例では、優先順位 2 の項目の値を、SHIPPED から IN_SHIPPING に変更します。

```
DBMS_REPCAT.ALTER_PRIORITY_VARCHAR2(
    gname          => 'acct',
    pgroup         => 'status',
    old_value       => 'shipped',
    new_value       => 'in_shipping');
```

追加情報：ALTER_PRIORITY_datatype プロシージャのパラメータは、表 9-89 を参照してください。例外は、表 9-90 を参照してください。

メンバーの優先順位の変更

DBMS_REPCAT パッケージ内の ALTER_PRIORITY プロシージャを用いて、指定の優先順位グループ・メンバーに対応する優先順位を変更します。優先順位グループのメンバーは、特定の値にそれぞれ優先順位がつけられた形になっていて、このプロシージャによって指定の列値の優先順位の上げ下げを効率よく行えます。高位の優先順位値を持つメンバーには、より高位の優先順位が与えられて競合が解消されます。

次の例では、IN_SHIPPING のマークの付いた項目の優先順位を、レベル 2 からレベル 4 に変更します。

```
DBMS_REPCAT.ALTER_PRIORITY(
    gname          => 'acct',
    pgroup         => 'status',
    old_priority    => 2,
    new_priority    => 4);
```

追加情報：ALTER_PRIORITY プロシージャのパラメータは、表 9-87 を参照してください。例外は、表 9-88 を参照してください。

値によるメンバーの削除

DBMS_REPCAT パッケージ内には、値によって優先順位グループのメンバーを削除するためのプロシージャがいくつかあります。これらのプロシージャは、DROP_PRIORITY_type の形になります。type は、優先グループを作成したときに指定したデータ型と同じ値です。

- DROP_PRIORITY_CHAR
- DROP_PRIORITY_VARCHAR2
- DROP_PRIORITY_NUMBER
- DROP_PRIORITY_DATE
- DROP_PRIORITY_RAW
- DROP_PRIORITY_NCHAR
- DROP_PRIORITY_NVARCHAR2

コールする必要があるプロシージャは、「優先順位」列のデータ型によって決まります。

次の例では、STATUS 優先順位グループ内の項目として、IN_SHIPPING は有効なステータスではなくなります。

```
DBMS_REPCAT.DROP_PRIORITY_VARCHAR2(  
    gname      => 'acct',  
    pgroup     => 'status',  
    value      => 'in_shipping');
```

追加情報: DROP_PRIORITY_datatype プロシージャのパラメータは、表 9-140 を参照してください。例外は、表 9-141 を参照してください。

優先順位によるメンバーの削除

DBMS_REPCAT パッケージ内の DROP_PRIORITY プロシージャを使用して、優先順位レベルによって優先順位グループのメンバーを削除します。

次の例では、STATUS 優先順位グループ内の項目として、(優先順位レベル4に割り当てられた) IN_SHIPPING は有効なステータスではなくなります。

```
DBMS_REPCAT.DROP_PRIORITY(  
    gname      => 'acct',  
    pgroup     => 'status',  
    priority_num => 4);
```

追加情報: DROP_PRIORITY プロシージャのパラメータは、表 9-136 を参照してください。例外は、表 9-137 を参照してください。

優先順位グループの削除

DBMS_REPCAT パッケージ内の DROP_PRIORITY_GROUP プロシージャを使用して、指定したマスター・グループの優先順位グループを削除します。たとえば、STATUS優先順位グループを削除するには、次のようにコールします。

```
DBMS_REPCAT.DROP_PRIORITY_GROUP(
    gname      => 'acct',
    pgroup     => 'status');
```

注意：優先順位グループを削除する前に、その優先順位グループに依存するすべての列グループから、優先順位グループの更新の競合解消方法を削除しておいてください。どの列グループが優先順位グループに依存しているかは、REPRESOLUTION ビューに問合せを実行することによって判断できます。

追加情報：DROP_PRIORITY_GROUP プロシージャのパラメータは、表 9-138 を参照してください。例外は、表 9-139 を参照してください。

更新の競合解消のためのサイト優先順位グループの使用

サイト優先順位は、特殊な形式の優先順位グループです。したがって、サイト優先順位に対応するプロシージャの多くは、優先順位グループに対応付けられたプロシージャと類似した動作をします。

サイト優先順位方法を使用して更新の競合を解消する場合は、Replication Managerを使用して列グループにこの競合解消方法を追加する前に、まずサイト優先順位グループを作成する必要があります。サイト優先順位グループを作成する手順には、次の2つのステップがあります。

1. サイト優先順位グループの名前を定義する。
2. 各サイトをサイト優先順位グループに追加し、優先順位レベルを定義する。この情報は、REPPRIORITY ビューに表示されます。

一般に、1つのマスター・グループに対して必要なサイト優先順位グループは1つだけです。このサイト優先順位グループは、任意の数のレプリケート表で使用できます。以降の項では、サイト優先順位グループを管理する方法を説明します。

サイト優先順位管理の前後の手順

サイト優先順位を構成または管理する場合は、次に示す作業をこのとおりの順序で実行しなければならないことに注意してください。

- サイト優先順位グループを作成または管理する前に、Replication Manager を使用して、関連するマスター・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。
- サイト優先順位グループは、関連するマスター・グループのマスター定義サイトで管理しなければなりません。
- いずれかの方法でサイト優先順位グループを管理した後、ReplicationManager を使用して、対応付けられたレプリケート表のレプリケーション・サポートを再生成してください。

い。その後、Replication Managerを使用して、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できます。

サイト優先順位グループの作成

DBMS_REPCAT パッケージ内の DEFINE_SITE_PRIORITY プロシージャを使用して、次の例に示すように、マスター・グループの新規のサイト優先順位グループを作成します。

```
DBMS_REPCAT.DEFINE_SITE_PRIORITY(  
    gname      => 'acct',  
    name       => 'site');
```

この例では、ACCT オブジェクト・グループに SITE と呼ばれるサイト優先順位グループを作成します。

追加情報: DEFINE_SITE_PRIORITY プロシージャのパラメータは、表 9-124 を参照してください。例外は、表 9-125 を参照してください。

グループへのサイトの追加

DBMS_REPCAT パッケージ内の ADD_SITE_PRIORITY_SITE プロシージャを用いて、次の例に示すように、サイト優先順位グループに新規サイトを追加します。

```
DBMS_REPCAT.ADD_SITE_PRIORITY_SITE(  
    gname      => 'acct',  
    name       => 'site',  
    site       => 'hq.widgetek.com',  
    priority   => 100);
```

この例では、HQ サイトを SITE グループに追加し、その優先順位を 100 に設定しています。

注意: 優先順位の値が最も高いサイトに、最も高い優先順位が割り当てられています。優先順位値は連続する整数でなくてもかまいません。

追加情報: ADD_SITE_PRIORITY_SITE プロシージャのパラメータは、表 9-79 を参照してください。例外は、表 9-80 を参照してください。

サイトの優先順位レベルの変更

DBMS_REPCAT パッケージ内の ALTER_SITE_PRIORITY プロシージャを使用して、次の例に示すように、所定のサイトに対応付けられた優先順位を変更します。

```
DBMS_REPCAT.ALTER_SITE_PRIORITY(  
    gname      => 'acct',  
    name       => 'site',  
    old_priority => 100,  
    new_priority => 200);
```

この例では、SITE グループ内のサイトの優先順位を 100 から 200 に変更します。

注意：優先順位の値が最も高いサイトに、最も高い優先順位が割り当てられています。優先順位値は連続する整数でなくてもかまいません。

追加情報：ALTER_SITE_PRIORITY プロシージャのパラメータは、表 9-91 を参照してください。例外は、表 9-92 を参照してください。

優先順位に対応付けられたサイトの変更

DBMS_REPCAT パッケージ内の ALTER_SITE_PRIORITY_SITE プロシージャを用いて、次の例に示すように、所定の優先順位に対応付けられたサイトを変更します。

```
DBMS_REPCAT.ALTER_SITE_PRIORITY_SITE(  
    gname          => 'acct',  
    name           => 'site',  
    old_site       => 'hq.widgetek.com',  
    new_site       => 'hq.widgetworld.com');
```

この例では、HQ サイトのグローバル・データベース名を HQ.WIDGETWORLD.COM に変更しますが、その優先順位はそのままです。

追加情報：ALTER_SITE_PRIORITY_SITE プロシージャのパラメータは、表 9-93 を参照してください。例外は、表 9-94 を参照してください。

サイト名によるサイトの削除

DBMS_REPCAT パッケージ内の DROP_SITE_PRIORITY_SITE プロシージャを用いて、次の例に示すように、サイト優先順位から、所定のサイトを名前によって削除します。

```
DBMS_REPCAT.DROP_SITE_PRIORITY_SITE(  
    gname          => 'acct',  
    name           => 'site',  
    site           => 'hq.widgetek.com');
```

この例では、SITE グループから HQ サイトを削除します。

追加情報：DROP_SITE_PRIORITY_SITE プロシージャのパラメータは、表 9-144 を参照してください。例外は、表 9-145 を参照してください。

優先順位によるサイトの削除

5-27 ページで説明した DBMS_REPCAT.DROP_PRIORITY プロシージャを使用して、優先順位レベルでサイト優先順位グループからサイトを削除します。

サイト優先順位グループの削除

DBMS_REPCAT パッケージ内の DROP_SITE_PRIORITY プロシージャを使用して、次の例に示すように、指定したマスター・グループのサイト優先順位グループを削除します。

```
DBMS_REPCAT.DROP_SITE_PRIORITY(  
    gname => 'acct',  
    name  => 'site');
```

この例では、SITE は有効なサイト優先順位グループではなくなりました。

注意：このプロシージャをコールする前に、このサイト優先順位グループによる SITE PRIORITY 競合解消方法を使用しているマスター・サイトのすべての列グループに対し、DROP_UPDATE_RESOLUTION プロシージャをコールする必要があります。REPRESOLUTION ビューを問い合わせることによって、影響を受ける列グループを判断できます。

追加情報：DROP_SITE_PRIORITY プロシージャのパラメータは、表 9-142 を参照してください。例外は、表 9-143 を参照してください。

サンプル・タイムスタンプおよびサイト・メンテナンス・トリガー

最初に、トリガー内またはアプリケーション内で、タイムスタンプ情報およびサイト情報を保持できるようにロジックを組み立てなくてはなりません。次のサンプル・トリガーではクロック同期の問題を考慮していますが、アプリケーションが複数の時間帯にまたがる場合は修正する必要があります。

サンプル・トリガーでは、トリガーを実際に起動する必要があるかどうかをチェックするために、生成されたプロシージャの 1 つを使用するため、トリガーを作成する前に関連する CUSTOMERS 表のレプリケーション・サポートを生成する必要があります。これによって CUSTOMERS 表に対するトランザクションも伝播できます。

```
dbms_repcat.generate_replication_support(sname => 'SALES',  
    oname => 'CUSTOMERS',  
    type => 'TABLE');
```

これで、トリガーを定義できます。

```
CREATE OR REPLACE TRIGGER sales.t_customers  
    BEFORE INSERT OR UPDATE ON sales.customers  
    FOR EACH ROW  
DECLARE  
    timestamp$x DATE := SYSDATE;  
    site$x VARCHAR2(128) := dbms_reputil.global_name;  
BEGIN  
    -- Don't fire if a snapshot refreshing;  
    -- Don't fire if a master and replication is turned off  
    IF (NOT (dbms_snapshot.i_am_a_refresh)  
    AND dbms_reputil.replication_is_on) THEN  
        IF NOT dbms_reputil.from_remote THEN
```



```
IF INSERTING THEN
    -- set site and timestamp columns.
    :new."TIMESTAMP" := TIMESTAMP$X;
    :new."SITE" := SITE$X;
ELSIF UPDATING THEN
    IF (:old."ADDR1" = :new."ADDR1" OR
        (:old."ADDR1" IS NULL AND :new."ADDR1" IS NULL)) AND
        (:old."ADDR2" = :new."ADDR2" OR
        (:old."ADDR2" IS NULL AND :new."ADDR2" IS NULL)) AND
        (:old."FIRST_NAME" = :new."FIRST_NAME" OR
        (:old."FIRST_NAME" IS NULL AND :new."FIRST_NAME"
            IS NULL)) AND
        (:old."LAST_NAME" = :new."LAST_NAME" OR
        (:old."LAST_NAME" IS NULL AND :new."LAST_NAME" IS NULL)) AND
        (:old."SITE" = :new."SITE" OR
        (:old."SITE" IS NULL AND :new."SITE" IS NULL)) AND
        (:old."TIMESTAMP" = :new."TIMESTAMP" OR
        (:old."TIMESTAMP" IS NULL AND :new."TIMESTAMP" IS NULL)) THEN
        -- column group was not changed; do nothing
        NULL;
    ELSE
        -- column group was changed; set site and timestamp columns.
        :new."SITE" := SITE$X;
        :new."TIMESTAMP" := TIMESTAMP$X;
        -- consider time synchronization problems;
        -- previous update to this row may have originated from a site
        -- with a clock time ahead of the local clock time.
        IF :old."TIMESTAMP" IS NOT NULL AND
            :old."TIMESTAMP" > :new."TIMESTAMP" THEN
            :new."TIMESTAMP" := :old."TIMESTAMP" + 1 / 86400;
        ELSIF :old."TIMESTAMP" IS NOT NULL AND
            :old."TIMESTAMP" = :new."TIMESTAMP" AND
            (:old."SITE" IS NULL OR :old."SITE" != :new."SITE") THEN
            :new."TIMESTAMP" := :old."TIMESTAMP" + 1 / 86400;
        END IF;
    END IF;
END IF;
END IF;
```

```
END IF;  
END;
```

次に、Replication Managerを使用してレプリケート表を含むマスター・グループにトリガーを追加し、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。

一意性の競合解消方法の構成

一般的なアドバンスド・レプリケーション環境では、一意性の競合は未然に防止されているはずです。しかし、一意性の競合が発生する可能性が避けられない場合は、レプリケート表の PRIMARY KEY 制約または UNIQUE 制約に 1 つ以上の競合解消方法を割り当てて、発生した競合を解消できます。以降の項では、レプリケート表の構成方法、および PRIMARY KEY 制約と UNIQUE 制約に対する一意性の競合解消方法について説明します。

一意性の競合解消方法の割当て

制約に一意性の競合解消方法を割り当てるには、DBMS_REPCAT パッケージの ADD_UNIQUE_RESOLUTION プロシージャを使用します。たとえば、次の文を使用して、APPEND_SEQUENCE という一意性の競合解消方法を CUSTOMERS 表の C_CUST_NAME 制約に割り当てます。

```
DBMS_REPCAT.ADD_UNIQUE_RESOLUTION(  
    sname           => 'acct',  
    oname           => 'customers',  
    constraint_name => 'c_cust_name',  
    sequence_no     => 1,  
    method          => 'APPEND SEQUENCE',  
    comment         => 'Resolve Conflict',  
    parameter_column_name => 'last_name');
```

追加情報: ADD_UNIQUE_RESOLUTION プロシージャのパラメータは、第 10 章を参照してください。

一意性の競合解消方法の削除

制約から一意性の競合解消方法を削除するには、DBMS_REPCAT パッケージの DROP_UNIQUE_RESOLUTION プロシージャを使用します。たとえば、次の文を使用して、前述の例で割り当てた一意性の競合解消方法を削除します。

```
DBMS_REPCAT.DROP_UNIQUE_RESOLUTION(  
    sname           => 'acct',  
    oname           => 'customers',  
    constraint_name => 'c_cust_name',  
    sequence_no     => 1);
```

追加情報: DROP_UNIQUE_RESOLUTION プロシージャのパラメータは、第9章を参照してください。

ビルトインの一意性の競合解消方法

Oracle には、一意性の競合を解消するビルトインの方法が3つ用意されています。

- 発行側サイトのグローバル名を、発行側サイトの列値に追加する。
- 生成された順序番号を、発行側サイトの列値に追加する。
- 発行側サイトから行の値を廃棄する。

以降の項では、一意性の競合解消方法のそれぞれについて、詳しく説明します。

注意: Oracle に用意されているビルトインの一意性の競合解消方法は、実際にはレプリケート環境のデータを収束しません。制約違反を解決するテクニックを提供するだけです。Oracle の一意性の競合解消方法を使用する場合には、一意性の競合の発生を知らせる通知メカニズムを同時に使用して、必要に応じて手動でレプリケート・データを収束しなければなりません。データ収束の詳細は、5-31 ページの「データ収束の保証」を参照してください。

サイト名 / 順序の追加

サイト名の追加による方法および順序の追加による方法は、DUP_VAL_ON_INDEX 例外が発生する列に文字列を追加して実行します。これらの方法を使用すると、一意の整合性制約に違反することなく列を挿入または更新できますが、複数のマスター・サイト間での収束は提供されません。結果として起こる相違は、手動で解決しなければなりません。したがって、これらの方法は、通知機能の中のある形式で使用するよう想定されています。

注意: サイト名の追加および順序の追加は、どちらも文字型の列にしか使用できません。

これらの方法は、データの完全な正確さよりもデータの可用性の方が重要な場合に有用です。データをレプリケートした後で、すぐに使用可能な状態にするには、次のようにします。

- サイト名の追加または順序の追加を選択する。
- 競合をロギングするのではなく、通知スキームを使用して、適切な担当者に重複を解消するように警告する。

サイト名の追加による方法では、一意性の競合が発生すると、トランザクションを発行しているサイトのグローバル・データベース名がレプリケートされた列値に追加されます。名前は最初のピリオド (.) ままで追加されます。たとえば、HOUSTON.WORLD は HOUSTON になります。

同様に、順序の追加による方法では、生成された順序が列値に追加されます。列値は必要に応じて切り捨てられます。列値の生成された部分が列の長さを超えた場合、この競合解消方法はエラーを解決しません。

廃棄

廃棄の一意性による競合解消方法では、エラーが発生した発行元サイトから列を単純に廃棄することによって、一意性の競合を解消します。この方法では、複数のマスター間での収束は保証されないため、通知機能とともに使用するようにしてください。

追加による方法とは異なり、廃棄の一意性による方法では、データの正確さを検証できるまで、データの伝播が最小限にとどめられます。

削除の競合解消の構成

一般的なアドバンスド・レプリケーション環境では、削除の競合は未然に防止されているはずですが、削除の競合が発生する可能性が避けられない場合は、独自の削除の競合解消方法を作成して、レプリケート表に割り当てることができます。以降の項では、レプリケート表の構成方法、および表に対して使用するユーザー定義の削除の競合解消方法について説明します。

削除の競合解消方法の割り当て

レプリケート表にユーザー定義の削除の競合解消方法を割り当てるには、DBMS_REPCAT パッケージの ADD_DELETE_RESOLUTION プロシージャを使用します。たとえば、次の文を使用して、CUSTOMERS_DELETE_M1 というユーザー定義ファンクションを、CUSTOMERS 表の削除の競合解消方法に割り当てます。

```
DBMS_REPCAT.ADD_DELETE_RESOLUTION(  
    sname           => 'acct',  
    oname           => 'customers',  
    sequence_no     => 1,  
    parameter_column_name => 'last_name',  
    function_name    => 'customers_delete_m1' );
```

追加情報: ADD_DELETE_RESOLUTION プロシージャのパラメータは、第 10 章を参照してください。

削除の競合解消方法の削除

レプリケート表から削除の競合解消方法を削除するには、DBMS_REPCAT パッケージの DROP_DELETE_RESOLUTION プロシージャを使用します。たとえば、次の文を使用すると、前述の例で割り当てた削除の競合解消方法を削除します。

```
DBMS_REPCAT.DROP_DELETE_RESOLUTION(  
    sname           => 'acct',  
    oname           => 'customers',  
    sequence_no     => 1);
```

追加情報：DROP_DELETE_RESOLUTION プロシージャのパラメータは、第 10 章を参照してください。

データ収束の保証

データ収束はアドバンスド・レプリケーション・システムの必要条件です。データ収束は、すべてのレプリケーション・サイトで、ある行の値が最終的に一致する場合に起こります。アドバンスド・レプリケーション・システムを構成するときには、設計する競合解消方針でデータ収束が保証されていなければなりません。表 5-1 に、Oracle ビルトインの更新の競合解消方法と、それらの方法がどの構成で複数マスター・サイトと対応付けられたスナップショット・サイト間のデータ収束を保証するかを示します。

表 5-1 更新の競合解消方法のデータ収束プロパティ

解消の方法	1 つのマスター・サイト	2 つのマスター・サイト	任意の数のマスター・サイト
最小値	収束する	収束する	収束する (列値は常に減少しなければならない)
最大値	収束する	収束する	収束する (列値は常に増加しなければならない)
最古のタイムスタンプ	収束する (予備方法を設定する場合)	収束する (予備方法を設定する場合)	収束しない
最新タイムスタンプ	収束する (予備方法を設定する場合)	収束する (予備方法を設定する場合)	収束する (予備方法を設定する場合)
優先順位グループ	収束する	収束する (順序付けられた更新値がある場合)	収束する (順序付けられた更新値がある場合)
サイト優先順位	収束する	収束する	収束しない
上書き	収束する	収束しない	収束しない
廃棄	収束する	収束しない	収束しない
平均	収束する	収束しない	収束しない
加算	収束する	収束する	収束する

注意: Oracle ビルトインの一意性の競合解消方法では、レプリケート環境でのデータ収束は保証されません。したがって、一意性の競合解消方法とともに競合通知を構成し、さらに、必要に応じて手動でデータを収束する必要があります。

複数のマスター・サイトがある場合は、上書きおよび廃棄、平均による方法を使用してもデータの収束は保証されません。これらの方法を使用する場合は、必ず通知機能を併用する必要があります。

3 つ以上のマスター・サイトがある場合は、その他の方法のいくつかで、データの収束が保証されません。また、ネットワーク障害が発生したり、遅延リモート・プロシージャ・コール (RPC) を処理する頻度が少なかったりすると、これらの方法でデータが収束しない可能性が大きくなります。

順序の競合の防止

順序の競合は、3 つ以上のマスター・サイトがあるアドバンスド・レプリケーション構成で発生します。なんらかの理由でマスター・サイト X への伝播がブロックされた場合に、レプリケート・データへの更新を他のマスター・サイト間で伝播できます。伝播が再開されると、これらの更新は、他のマスターで発生した順序と異なる順序でサイト X に伝播され、これらの更新が競合します。デフォルトでは、発生した競合をエラー・ログに記録し、競合が依存するトランザクションが伝播および適用された後に、再実行できます。しかし、できる限り順序の競合を防止するか、自動的に解消するのが最善です。たとえば、順序の競合が発生する可能性のあるマルチマスター構成で収束を保証するための、競合解消ルーチンを選択する必要があります。

表 5-2 は、3 つのマスター・サイトでどのような場合に順序の競合が発生する可能性があるかを示しています。マスター・サイト A の優先順位は 30 で、マスター・サイト B の優先順位は 25 で、マスター・サイト C の優先順位は 10 です。x は、サイト優先順位による競合解消方法が割り当てられている列グループ内の特定の行の列です。優先順位の値が最も高いサイトに、最も高い優先順位が割り当てられています。優先順位値は任意の Oracle 番号で、連続する整数でなくてもかまいません。

表 5-2 サイト優先順位による順序の競合 - 3 つ以上のマスター

時間	アクション	サイト A	サイト B	サイト C
1	すべてのサイトで $x=2$ に更新され、一致している。	2	2	2
2	サイト A で $x=5$ に更新する。	5	2	2
3	サイト C が使用不能になる。	5	2	down
4	サイト A がサイト B に更新を送る。 サイト A とサイト B が $x=5$ で一致する。 サイト C は使用不能のままである。 更新トランザクションはサイト A のキューに残留している。	5	5	down
5	サイト C が $x=2$ で使用可能になる。 サイト A とサイト B が $x=5$ で一致する。	5	5	2
6	サイト B で $x=5$ を $x=7$ に更新する。	5	7	2
7	サイト B がトランザクションをサイト A に送る。 サイト A とサイト B が $x=7$ で一致する。 サイト C は $x=2$ のままである。	7	7	2
8	サイト B がトランザクションをサイト C に送る。 サイト C の古い値は $x=2$ である。 サイト B の古い値は $x=5$ である。 競合が検出され、サイト C の優先順位 (10) より高い優先順位 (25) を持つ サイト B の更新を適用することで、競合が解消される。 すべてのサイトが $x=7$ で一致する。	7	7	7
9	サイト A がそのトランザクション ($x=5$) を正常にサイト C に送る。 サイト C ($x=7$) がサイト A の古い値 ($x=2$) と一致しないので、 Oracle が競合を検出する。 サイト A はサイト C (10) より高い優先順位 (30) を持つ。 サイト A ($x=5$) の古い更新を適用することで、競合が解消される。 この競合処理の順序が原因で、サイトのデータは収束しなくなる。	7	7	5

所有権のフローを順序付ける必要がある場合は、優先順位グループを使用すると、収束を保証できます。たとえば、ワークフロー・モデルは、次の 3 つのステップ順序で情報が 1 つの方向に流れることを示しています。

1. ORDERING サイトから
2. SHIPPING サイトへ
3. BILLING サイトへ

BILLING サイトが ORDERING サイトからある行の変更を受信したが、その前にすでに SHIPPING サイトからその行への変更を受信していた場合、SHIPPING サイトの変更の方が優先順位が高いので、ORDERING サイトの変更を古い変更として無視します。

提案：使用する競合解消方法の決定をやすくするために、ダイアグラムまたは（表 5-2 のような）時間 - アクション表を作成して、競合解消方法にもれがないようにしてください。

更新の競合解消のためのデータ伝播の最小化

行の更新の競合を検出および解消するために、伝播サイトから受信サイトに行の新旧バージョンのデータの一部を送信する必要があります。環境によって、更新の競合の検出および解消をサポートするために伝播されるデータの量は異なります。

たとえば、レプリケート表を作成し、すべての関連サイトが Oracle8 データベースの場合、表の変更された各行の競合を検出するために通信する必要があるデータの量を最小化できます。この場合、Oracle は次の値を伝播します。

- 主キーおよび列グループの各列の、古い値（変更前の値）。
- 列グループの更新された各列の新しい値。

注意：挿入された行には、古い値はありません。また、削除された行には、新しい値はありません。

一般的に、Oracle8 のみで構成されるアドバンスド・レプリケーション・システムでは、ネットワークで送信する必要のあるデータの量を減らすために、データ伝播の最小化を実行する必要があります。それによって、システム全体のパフォーマンスが向上します。

あるいは、レプリケート環境で Oracle7 のサイトと Oracle8 のサイトを併用している場合は、更新の競合解消のための行データの通信を最小化することはできません。この場合、競合を解消するために、変更された各行の新旧の値をすべて伝播する必要があります。

Replication Manager を使用してレプリケート表のサポートを生成する場合は、「レプリケーション・オブジェクトの編集」プロパティ・シートの「最小限の通信」の設定を使用可能にして、データ伝播を最小化できます。レプリケーション API を使用する場合は、次の DBM-S_REPCAT プロシージャの min_communication パラメータを TRUE に設定することによって、データ伝播を最小化できます。

- CREATE_SNAPSHOT_REPOBJECT
- GENERATE_REPLICATION_SUPPORT
- GENERATE_REPLICATION_TRIGGER
- GENERATE_SNAPSHOT_SUPPORT

通信の最小化の例

下のレプリケート表では、列 1 と列 3 が主キーを構成しています。列 1-3 および列 4-6 という 2 つの列グループがあります。

C1	C2	C3	C4	C5	C6

列グループ 1 = C1, C2, & C3
列グループ 2 = C4, C5, & C6
主キー = C1 & C3

表のレプリケーション・サポートの生成時に「最小限の通信」の設定を使用禁止にすると、Oracle は更新のために 6 つの古い値 (C1-C6) と 6 つ新しい値 (C1-C6) を送信します。たとえば、列 C4 を更新すると、次のようになります。

	C1	C2	C3	C4	C5	C6
古い	古い値	古い値	古い値	古い値	古い値	古い値
新しい	新しい値	新しい値	新しい値	更新 新しい値	新しい値	新しい値

また、表のレプリケーション・サポートの生成時に「最小限の通信」の設定を使用可能にすると、Oracle は通信を最小化します。たとえば、列 C4 を更新すると、次のものが送信されます。

- 主キーの古い値
- 更新された列グループの古い値
- 主キーおよび変更された列グループ内にはない古い値については、NULL
- 更新されていないすべての列の新しい値については、NULL
- 更新された列の実際の新しい値

	C1	C2	C3	C4	C5	C6
古い	主キー	NULL	主キー	変更された 列グループ	変更された 列グループ	変更された 列グループ
新しい	NULL	NULL	NULL	更新 新しい値	NULL	NULL

列 C2 および C4 を更新すると、次のものが送信されます。

- 変更された両方の列グループの古い値
- 主キー列の古い値
- 2 つの更新の新しい値
- 他の 4 つの新しい値については、NULL

	C1	C2	C3	C4	C5	C6
古い	変更された 列グループ 1 と 主キー	変更された 列グループ 1	変更された 列グループ 1 と 主キー	変更された 列グループ 2	変更された 列グループ 2	変更された 列グループ 2
新しい	NULL	更新	NULL	更新	NULL	NULL

列 2 を更新すると、次のものが送信されます。

- 主キーの古い値
- 変更された列グループのメンバーの古い値
- 更新された列の新しい値
- 他の 5 つの新しい値については、NULL

古い	変更された 列グループ と主キー	変更された 列グループ	変更された 列グループ と主キー	NULL	NULL	NULL
新しい	NULL	更新	NULL	NULL	NULL	NULL

データ伝播をさらに削減する

前述した方法を使用して伝播を最小化した場合は、DBMS_REPCAT.SEND_AND_COMPARE_OLD_VALUES プロシージャを使用して、競合の検出と解消のために必要な古い値のみを送信することによって、場合によってはデータ伝播をさらに削減できます。たとえば、最新のタイムスタンプ競合の検出と解消方法は、非キーおよび非タイムスタンプ列の古い値を必要としません。

提案：LOB データ型をレプリケートしていて、これらの列で競合が発生しないと考えられる場合、古い値の伝播をさらに最小化することには特に価値があります。

注意：予想される競合を検出および解消するために必要な古い値が確実に伝播されるようにしてください。ユーザーが提供する競合解消プロシージャは、送信される古い列値 NULL を適切に処理する必要があります。SEND_AND_COMPARE_OLD_VALUES を使用して、データ伝播をさらに削減すると、予期しない競合に対する保護が低下します。

データ伝播をさらに削減するには、次のプロシージャを実行します。

```
DBMS_REPCAT.VARCHAR2SDBMS_REPCAT.SEND_AND_COMPARE_OLD_VALUES(
    sname          IN    VARCHAR2,
    oname          IN    VARCHAR2,
    column_list    IN    VARCHAR2 |
    column_table   IN    DBMS_REPCAT.VARCHAR2s
    operation      IN    VARCHAR2 := 'UPDATE',
    send          IN    BOOLEAN  := TRUE);
```

このプロシージャを実行した後で、min_communication を TRUE に設定してレプリケーション・サポートを再度生成し、この変更を有効にする必要があります。

注意：行が削除されたとき、またはキー以外の列が更新されたとき、あるいはその両方の場合に、操作パラメータを使用すると、キー以外の列の古い値を送信するかどうかを決定できます。古い値を送信しない場合、Oracle は古い値のかわりに NULL を送信し、更新または削除が適用されたときに送信先の列の現在の値が古い値と等しいことを想定します。

古い列値に指定した動作は、REPCOLUMNビューの2つの列、COMPARE_OLD_ON_DELETE ('Y' または 'N') と COMPARE_OLD_ON_UPDATE ('Y' または 'N') に表示されます。

次の例では、SEND_AND_COMPARE_OLD_VALUESを使用してデータ伝播をさらに削減する方法を示します。3つの列を持つ 'SCOTT.REPORTS' という表で考えてみましょう。列1は主キーであり、それ自身の列グループ (列グループ1) を構成します。列2および列3は2番目の列グループ (列グループ2) にあります。

列 1	列 2	列 3
主キー	サイト	LOB
列グループ 1	列グループ 2	

2番目の列グループの競合解消方法は、サイト優先順位です。列2は、サイト名を含む VARCHAR2列です。列3は、LOB列です。そのLOBを更新するときは常に、更新が行われたサイトのグローバル名で列2も更新する必要があります。LOBに対する個々の更新にはトリガーがないので、LOBに対して個々の更新を行う場合は常に列2を明示的に更新する必要があります。

min_communication が TRUE に設定されている SCOTT.REPORTS のレプリケーション・サポートを生成し、UPDATE文を使用して列2 (サイト名) と列3 (LOB) を変更するとします。サイト名の新しい値とLOBの新しい値は更新されているので、遅延リモート・プロシージャ・コール (RPC) にはこれらが含まれます。遅延RPCには、主キー (列1) の古い値およびサイト名 (列2) の古い値、LOB (列3) の古い値も含まれます。

注意: 競合の検出および解消計画には、LOBの古い値は不要です。競合の検出と解消の両方には、列C2 (サイト名) だけが必要です。LOBの古い値を送信すると、伝播時間が著しく増加します。

列C2または列C3が更新されたときにLOBの古い値が伝播されないように、次のコールを実行してください。

```
dbms_repcat.send_and_compare_old_values(  
    sname      => 'SCOTT',  
    oname      => 'REPORTS',  
    column_list => 'C3',  
    operation   => 'UPDATE',  
    send       => FALSE);
```

min_communication を TRUE に設定した SCOTT.REPORTS のレプリケーション・サポートを再度生成し、変更を有効にする必要があります。続いて UPDATE 文を使用して列 2 (サイト名) と列 3 (LOB) を変更するとします。遅延 RPC には、主キー (列 1) の古い値およびサイト名 (列 2) の古い値と新しい値、LOB (列 3) の新しい値が含まれます。遅延 RPC は、主キーの新しい値と LOB の古い値については NULL を含みます。

追加情報: SEND_AND_COMPARE_OLD_VALUES プロシージャのパラメータは、9-122 ページの表 9-182 を参照してください。例外は、9-123 ページの表 9-183 を参照してください。

ユーザー定義の競合解消方法

Oracle では、ユーザー独自の競合解消方法または通知方法を作成できます。ユーザー定義の解消方法とは、TRUE または FALSE のどちらかを戻す 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方法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、FALSE が戻されます。競合解消方法が TRUE を戻すか、または使用できる方法がなくなるまで、Oracle は順番に使用できる競合解消方法の評価を続けます。

競合解消方法が例外を呼び出すと、Oracle はその方法の評価を停止し、競合解消のために他の方法が (より大きな順序番号で) 提供されていても評価しません。

競合解消方法のパラメータ

競合解消方法に必要なパラメータは、解消する競合のタイプ (一意または更新、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。すべての競合解消方法は、表の古い列の値および新しい列の値、現在の列の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。
- 現在の値は、受信サイトでの対応する行の値を表します。

注意: Oracle が、比較する行の判別に主キー (または SET_COLUMNS で指定されたキー) を使用することに再度注意してください。

競合解消機能は、DBMS_REPCAT.ADD_conflicttype_RESOLUTION プロシージャの PARAMETER_COLUMN_NAME 引数で指定した列の値をパラメータとして受取らなければなりません。列パラメータは競合解消方法に、PARAMETER_COLUMN_NAME 引数にリストされた順序で、またはこの引数に '*' を指定した場合は昇順のアルファベット順で渡されます。古い列の値と新しい列の値の両方が (更新の競合の場合) 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方法では、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・サポートが再生成されるまで実行されません。

更新の競合解消

更新の競合の場合、ユーザー定義のファンクションは、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の値。このパラメータのモードは IN。この値は変更できない。
- 発信元サイトからの新しい列の値。このパラメータのモードは IN OUT。競合を正常に解消できる場合、そのファンクションは必要に応じて新しい列の値を変更する必要がある。
- 受信サイトからの現在の列の値。このパラメータのモードは IN。

列の値は、古いもの、新しいもの、そして現在のものと連続して受信されます。競合解消方法に渡す最後の引数は、ブール型フラグであることが必要です。このフラグが FALSE の場合、IN OUT パラメータの値をすでに更新したこと、および現在の列値を新しい値に更新しなくてはならないことを示します。このフラグが TRUE の場合、現在の列値を変更してはならないことを示します。

一意性の競合解消

一意性の競合は、INSERT または UPDATE の結果として生じます。一意性の競合解消方法は、列グループ内の各列に、IN OUT モードの発信元サイトからの新しい列値を使用する必要があります。競合解消方法に渡す最後の引数は、ブール型フラグであることが必要です。

競合解消方法で競合が解消されたら、新しい列値を変更して、Oracle で新しい列値による現在行への挿入または現在行の更新ができるようにしなければなりません。新しい列値を廃棄したい場合はファンクションでブール型フラグを TRUE に設定し、その他の場合は FALSE に設定します。

競合解消方法では一意性の競合が収束することを保証できないので、ユーザー定義の一意性の競合解消方法に通知メカニズムを組み入れる必要があります。

削除の競合解消

削除の競合が生じるのは、ローカル・サイトからの削除が正常に終了したものの、対応付けられた行がリモート・サイトに（たとえば、更新されたため）ない場合です。削除の競合では、ファンクションは古い列の値を行全体で IN OUT モードで使用する必要があります。競合解消方法に渡す最後のパラメータは、ブール型フラグであることが必要です。

競合解消方法では競合が解消されると、古い列値が変更され、Oracle ですべての古い列値に一致する現在行を削除できるようになります。これらの列値を廃棄したい場合はファンクションでブール型フラグを TRUE に設定し、その他の場合は 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・サイトでは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されません。この競合は NO_DATA_FOUND 例外を呼び出し、Oracle でエラー・トランザクションが記録されます。

これらのタイプの更新/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。削除された行をただマーク付けしてから、プロシージャ・レプリケーションを使用してそれらを削除することによって、これらのタイプの競合を完全に防止するほうがはるかに容易です。

追加情報：7-20 ページの「削除の競合の防止」を参照してください。

制限

ユーザー定義の競合解消方法で次の型の SQL コマンドを使用することは避けてください。これらの型のコマンドを使用すると、予測できない結果になる可能性があります。

- データ定義言語文（たとえば、CREATE、ALTER、DROP）
- トランザクション制御文（たとえば、COMMIT、ROLLBACK）
- セッション制御（たとえば、ALTER SESSION）
- システム制御（たとえば、ALTER SYSTEM）

ユーザー定義の競合解消方法の例

標準的な MAXIMUM および ADDITIVE のビルトイン競合解消方法を変形した、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にある NULL 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, use new values.
-- If new is null or new < curr, use current values.
-- If both are null, 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;
END;
```

```
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, old = 0.
-- If new is null, new = 0.
-- If curr is null, 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、競合解消の代わりまたは追加として競合の通知を提供するユーザー定義の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て、競合が発生した時点か、または次の競合解消方法を実行する前か、競合解消を試みて失敗した後に、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。

1. 競合通知ログを作成する。
2. パッケージ内にユーザー定義の競合通知方法を作成する。

次の項で各手順について説明します。

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には、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・サイトに競合通知ログを作成するには、レプリケーション実行DDL機能を使用します。詳細は、6-2 ページの「マスター・グループ内でのDDLの実行」を参照してください。競合通知表の項目は競合を検出するサイトに固有の項目であるため、競合通知表にレプリケーション・サポートを生成しないでください。

サンプルの競合通知ログ表

次のCREATE TABLE文を使用して、マスター・グループの複数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、その方法をPL/SQLパッケージ内で定義し、さらに、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しなければなりません。

競合通知方法は、競合の通知だけを実行するか、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 のビルトインの競合解消方法の 1 つを使用して競合を解消しなければなりません。ユーザー定義の競合通知方法で競合の通知だけを実行する場合は、競合を解消する競合解消方法とともに、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独立して通知が必要な場合は、DBMS_PIPE パッケージまたは Oracle Office とのデータベース・インタフェースを使用する通知メカニズムを設計できます。

サンプルの競合通知パッケージ

次のパッケージおよびパッケージ本体は、CUSTOMERS 表の一意性の競合を定義済みの CONF_REPORT 表へロギングすることによって、簡単な形式の競合の通知を実行します。

注意：この競合の通知の例では、競合は解消されません。したがって、競合解消方法（たとえば、廃棄や上書き）を用意するか、エラーが解消されないで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を（電子メールなどで）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、さらにアクティブな手段をとることができます。たとえば、このパッケージでは、通知メッセージを記録するだけでなく、DBMS_OFFICE ユーティリティ・パッケージを使用して管理者に Oracle Office 電子メールのメッセージを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 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 will be 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、レプリケート環境の各表および列グループで利用できる競合解消方法を判断するために使用するレプリケーション・カタログ (REPCAT) が用意されています。各ビューには、3つのバージョン、USER_*、ALL_*、SYS.DBA_* があります。利用できるビューを次に示します。

REPRESOLUTION_METHOD	使用可能な競合解消方法をすべて記述する。
REPCOLUMN_GROUP	データベースに定義された列グループをすべて記述する。
REPGROUPED_COLUMN	データベース内の各列グループ内の列をすべて記述する。
REPPRIORITY_GROUP	データベースに定義された優先順位グループおよびサイト優先順位グループをすべて記述する。
REPPRIORITY	各優先順位グループまたは各サイト優先順位グループの値および対応する優先順位を記述する。
REPCONFLICT	データベース内の表および列グループ、一意制約に関して解消方法を指定した競合のタイプ (削除または更新、一意) を記述する。
REPRESOLUTION	各オブジェクトで発生する競合を解消するための方法に関する、より具体的な情報を表示する。
REPPARAMETER_COLUMN	競合解消方法が競合を解消するために使用した列を表示する。

追加情報：第 10 章「データ・ディクショナリ・ビュー」を参照してください。

レプリケート環境の管理方法

この章では、レプリケート・データベース環境の管理方法について説明します。トピックには次の事柄が含まれます。

- マスター・グループおよびスナップショット・グループの上級管理テクニック
- アドバンスド・レプリケーション・システムの監視
- レプリケーション・システムにおけるデータベースのバックアップおよび回復
- 正常に終了した競合解消の監査
- レプリケート表相互間の相違の判別
- ビュー内のコメント・フィールドの更新

マスター・グループおよびスナップショット・グループの上級管理テクニック

このマニュアルの第3章と第4章では、アドバンスド・レプリケーション環境でマスター・グループおよびスナップショット・グループを管理するのに使用する最も一般的なプロシージャについて説明しました。以降の項では、マスター・グループおよびスナップショット・グループの管理に関するあまり一般的でないいくつかの管理プロシージャについて説明します。

マスター・グループ内での DDL の実行

Replication Managerを使用すると、1つ以上のSQLのDDL文をマスター・グループのマスター・サイトの一部またはすべてに伝播できます。このオプションでは、Oracle のレプリケーション管理 API 内で特にサポートされていない固有のDDLを実行できます。たとえば、レプリケーション環境をサポートするために必要なロールバック・セグメントおよびユーザーを作成することもできます。

警告：マルチマスター環境のグローバルなデータベース整合性に影響するようなDDLは実行しないでください。たとえば、任意のサイトでレプリケーション・オブジェクトを変更するDDL文は実行しないでください。

マスター・グループの選択したサイトでDDLを実行するには：

1. マスター定義サイトのターゲット・マスター・グループをクリックする。
2. 「プロパティ」ツールバー・ボタンをクリックする。
3. 「マスター・グループの編集」プロパティ・シートの「操作」ページをクリックする。
4. 「DDLの実行」をクリックする。
5. 「DDLの実行」ダイアログを使用して、選択した接続先でDDL文を実行する。

同等の API: DBMS_REPCAT.EXECUTE_DDL

マスター・グループの妥当性検査

マルチマスター・レプリケーション環境内のすべてのマスター・サイト間のマスター・グループの整合性を検査するために、マスター・グループを検査できます。マスター・グループを検査するには：

1. マスター定義サイトのターゲット・マスター・グループをクリックする。
2. 「プロパティ」ツールバー・ボタンをクリックする。
3. 「マスター・グループの編集」プロパティ・シートの「妥当性チェック」ページをクリックする。

「妥当性チェック」ページで、次のような特定の項目を検査できます。

- マスター・グループのオブジェクトのレプリケーション・サポートおよび妥当性
- マスター・グループのスケジュール・リンクの存在およびスケジュール

マスター・グループを検査する準備が整ったら、「検査」をクリックします。

マスター・グループの定義サイトの再配置

マスター・グループのマスター定義サイトが利用できなくなった場合、または単にグループのマスター接続先サイトを再配置する場合は、次の手順で行います。

1. 任意のマスター・サイトのターゲット・マスター・グループをクリックする。
2. 「プロパティ」ツールバー・ボタンをクリックする。
3. 「マスター・グループの編集」プロパティ・シートの「接続先」ページをクリックする。
4. 「変更」をクリックして、グループのマスター定義サイトを変更する。
5. 「マスター定義の変更」ダイアログを使用して、グループのマスター定義サイトを変更する。

マスター・グループのマスター定義サイトを再配置するときに、次の項目を通知できます。

- その他のすべてのマスター・サイト
- 現在のマスター定義サイト（現在のマスター定義サイトが利用できないときはこの設定をチェックしないでください）

同等の API: DBMS_REPCAT.RELOCATE_MASTERDEF

スナップショット・グループのマスター・サイトの変更

スナップショット・グループのマスター・サイトを別のマスター・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_SNAPSHOT_MASTERプロシージャをコールします。

```
DBMS_REPCAT.SWITCH_SNAPSHOT_MASTER(  
    gname          => 'sales',  
    master         => 'dbs2.acme.com',  
    execute_as_user => 'FALSE');
```

この例では、ACCT オブジェクト・グループのマスター・サイトがDBS2 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・サイトを変更するスナップショット・サイトでコールする必要があります。新しいデータベースは、レプリケート環境でのマスター・サイトであることが必要です。

このプロシージャをコールするとき、Oracle は新しいマスターを使用して、ローカル・スナップショット・グループの各スナップショットを完全にリフレッシュします。

サイトを切り換えたスナップショットの元のマスター・サイトのSYS.SLOG\$表にあるエントリは削除できません。その結果、元のマスター・サイトにある切り換えられた更新可能スナップショットのMLOG\$表は、DBMS_SNAPSHOT.PURGE_LOGをコールして消去しない限り、無限に大きくなる可能性があります。

追加情報：SWITCH_SNAPSHOT_MASTER プロシージャのパラメータは9-125 ページの表 9-188 を、例外は9-126 ページの表 9-189 を参照してください。

アドバンスト・レプリケーション・システムの監視

Oracle では、遅延トランザクションおよびジョブ・キューの内部システムを使用して、アドバンスト・レプリケーションのサイト間で変更内容を伝播します。レプリケーション環境の内部作業を常に監視して、スムーズに実行されているか確認することが重要です。以降の項では、管理要求および遅延トランザクション、エラー・トランザクション、ジョブ・キューを表示および管理するためにOracle Replication Managerを使用する方法について説明します。

管理要求の管理

管理要求とは、Oracle のレプリケーション管理 API のプロシージャまたはファンクションに対して特定のコールを実行することです。たとえば、Replication Managerを使用して新しいマスター・グループを作成するときに、Replication Manager は、DBMS_REPCAT.CREATE_MASTER_REPGROUP プロシージャをコールして作業を完了させます。レプリケーション・グループおよびその中のオブジェクトに対するすべてのDDL変更により管理要求が生成されます。上位レベルの管理要求の多くは、他のレプリケーション管理 API を生成して要求を処理します。

管理要求は、マスター定義サイトにより、すべてのマスター・サイトの管理要求キューに分散トランザクションとして挿入されます。各マスター・サイトには、各サイトで要求を同時に実行するスケジュール・ジョブ、DO_DEFERRED_REPCAT_ADMINがあります。管理要求が処理されるたびに、各サイトからマスター定義サイトにレポートが戻されます。正常に完了した要求は、マスター定義サイトの管理要求キューから削除されます。しかし、なんらかのエラーが発生した場合は、管理要求はエラーの状態でマスター定義サイトの管理要求キューに残ります。

Replication Manager を使用すると、管理要求の状態を参照および更新できます。以降の項では、マルチマスター・アドバンスト・レプリケーション環境のマスター・グループに対する管理要求の管理方法について詳しく説明します。

管理要求の表示

マスター・サイトのマスター・グループの管理要求を表示するには：

1. サイトの「構成」フォルダをオープンする。
2. その中の「マスター・グループ」フォルダをオープンする。
3. ターゲット・マスター・グループをオープンする。
4. ターゲット・マスター・グループの「管理要求」フォルダをクリックする。

Replication Managerの詳細パネルには、次に示すように対応するマスター・サイトの保留中のすべてのマスター・グループ管理要求の情報が表示されます。

- 要求に対応するレプリケーション管理 API コール
- 要求の現在の状態（処理前およびエラー、コールバック待機中など）
- 管理要求のソース・データベース
- 管理要求に関連するエラー

Replication Managerでは、詳細パネルの表示は自動的に更新されません。保留中の管理要求のより新しいリストを取得するには、画面をリフレッシュします。

追加情報: REPCATLOGデータ・ディクショナリ・ビューの問合せを実行して管理要求を調べることができます。

管理要求の適用

アドバンスド・レプリケーション機能の各インスタンスに1つ以上のSNPプロセスがある限り、各サーバーで周期的な間隔でジョブが起動され、すべての管理要求が実行されます。管理要求の自動実行を待たずに実行する場合は、Replication Managerを使用して保留中の管理要求をすべて手動で適用します。

1. ターゲット・マスター・サイトのマスター・グループの管理要求を表示する。
2. 詳細ウィンドウの内側を右クリックする。
3. 「すべての管理要求をすぐに適用する」をクリックする。

注意: Oracle では、自動的にすべてのマスター・サイト間で同時に管理要求を適用しようとしません。

同等の API: DBMS_REPCAT.DO_DEFERRED_REPCAT_ADMIN

管理要求の削除

マスター・グループから保留中の管理要求を削除する必要がある場合があります。たとえば、特定の管理要求でエラーが戻されるとします。そのエラー状況を解決した後も、手動でパージしない限り管理要求はサーバーのキューに残ります。

マスター・グループの管理要求を削除するには:

1. ターゲット・マスター・サイトのマスター・グループの管理要求を表示する。
2. ターゲット要求を1つ以上クリックする。
3. 「削除」ツールバー・ボタンをクリックする。

同等の API: DBMS_REPCAT.PURGE_MASTER_LOG

管理要求メカニズムの詳細

Replication Manager を使用して、または DBMS_REPCAT パッケージのプロシージャをコールしてアドバンスド・レプリケーション・システムを管理するとき、Oracle では内部メカニズムを使用して、同期レプリケーションを使った要求を同報通信します。なんらかの理由で同期同報通信が失敗した場合は、エラー・メッセージを戻し、それを含むトランザクションをロールバックします。

Oracle Server で管理要求を受信すると、REPCATLOG ビューに要求、および REPCATLOG ビューの子表に対応する DDL が記録されます。マスター・サイトのマスター・グループの管理要求を表示すると、別のマスター・サイトからのコールバックを待機している要求が含まれている場合があります。Replication Manager を使用してレプリケーション・グループの管理要求を作成するたびに、そのグループに対してジョブがまだ存在していない場合はローカル・ジョブ・キューにジョブが自動的に挿入されます。このジョブはプロシージャ DO_DEFERRED_REPCAT_ADMIN を定期的に行います。要求を同期をとって同報通信すると常に、Oracle はレプリケート変更を各マスター・サイトに適用するため、ただちにこのジョブを開始しようとします。

何もエラーがなければ、バックグラウンド・プロセスが使用可能なときは常に DO_DEFERRED_REPCAT_ADMIN が実行されてジョブを実行します。初期化パラメータ JOB_QUEUE_INTERVAL によって、バックグラウンド・プロセスの起動頻度が決定されます。未処理ジョブを実行するために十分な数のバックグラウンド・プロセスがない場合、処理が遅れることがあります。

注意: サイトの JOB_QUEUE_PROCESSES = 0 のとき、サイトのすべてのグループに対して管理要求を手動で適用しなければなりません。詳細は、6-5 ページの「管理要求の適用」を参照してください。

マスター・サイトで DO_DEFERRED_REPCAT_ADMIN をコールするたびに、サイトでは REPCATLOG ビューをチェックして、実行する必要がある要求が存在するかどうか判断します。管理要求が 1 つでも存在すると、要求が適用され、ローカル・ビューが適切に更新されます。このイベントは、各マスター・サイトで非同期に起こります。

DO_DEFERRED_REPCAT_ADMIN は、所定の順序でローカルの管理要求を実行します。DO_DEFERRED_REPCAT_ADMIN がマスター定義サイトではないマスターで実行されると、可能な範囲で処理が試行されます。レプリケート表の移植などのいくつかの非同期アクティビティでは、マスター定義サイトとの通信が必要になります。通信が不可能な場合、DO_DEFERRED_REPCAT_ADMIN は管理要求の実行を停止し、管理要求が順序を無視して実行されるのを防ぎます。マスター定義サイトで管理要求を更新または削除する最終ステップの時のように、マスター定義サイトとのいくつかの通信は遅延可能であり、その通信によって DO_DEFERRED_REPCAT_ADMIN の追加の管理要求の実行が妨げられることはありません。

各マスター・サイトで、管理要求が成功したか失敗したかは、各サイトの REPCATLOG ビューに示されています。マスター・グループごとに、Replication Managerにより各管理要求の対応する状態が表示されます。最後に、各マスター・サイトは、その管理要求の状態をマスター定義サイトに伝播します。マスター・サイトで要求が正常に完了した場合、マスター定義サイトで REPCATLOG ビューからそのサイトのコールバックが削除されます。イベントがすべてのサイトで正常に完了した場合、すべてのサイト（マスター定義サイトを含む）の REPCATLOG ビューにあるエントリがすべて削除されます。

同期をとって変更を同報通信することによって、Oracle は、すべてのサイトに変更を確実に認識させ、同期のとれた状態に保つことができます。後でそのサイトに変更を適用できるようにしているため、Oracle では、サイトに変更を適用するのに最も適した時点を自由に選択できます。

自動生成されるレプリケーション・サポートがオブジェクトに必要な場合、オブジェクトを変更したらレプリケーション・サポートを再生成する必要があります。Oracle では、すべてのマスター・サイトで、変更したオブジェクトのレプリケーションをサポートするために内部トリガーを起動し、パッケージを再生成します。

注意：これらのプロシージャをエラーを出さずに完了させるために、DDL をマスター定義サイトで正常に適用する必要がありますが、これによって DDL が各マスター・サイトで正常に適用されることが保証されるわけではありません。Replication Managerにより、すべての管理要求の状態が表示されます。REPCATLOG ビューには、一時的な状態および要求によって生成される非同期のエラー・メッセージが含まれます。

DDL 変更によって影響を受けるスナップショット・サイトがあれば、次にスナップショット・サイトのリフレッシュを実行するときに更新されます。すべてのマスター・サイトが別のマスター・サイトと通信できるのに対して、スナップショット・サイトが通信できるのは、対応するマスター・サイトだけです。

スナップショットのマスターに変更を加えた結果、スナップショットの形式を変更しなければならぬ場合、スナップショットを削除し、再作成しなければなりません。

管理要求に関する問題の診断方法

アドバンスド・レプリケーション環境の管理を実行した後、管理要求を調べて、すべての処理が円滑に実行されていることを確認しなければなりません。システムの監視中に、管理要求に関する問題が発生している場合があります。管理要求に関する問題のトラブルシューティング方法を次にいくつか示します。

- エラー条件がないのに管理要求がバックアップされる場合は、DBMS_REPCAT.DO_DEFERRED_REPCAT_ADMIN を実行しているローカル・ジョブに問題が発生していないか確認する。ジョブ・キューに関する問題の診断方法の詳細は、6-13 ページの「ジョブに関する問題の診断方法」を参照してください。

注意：さらに、レプリケーション・ジョブがレプリケーション管理者として実行中であるかどうか確認するため、DBA_JOBS ビュー内の LOG_USER 列をチェックします。要求を送ったユーザーがレプリケーション管理者だったかどうかを確認するために、DBA_REPCATLOG ビューの USERID 列をチェックしてください。DBMS_REPCAT-

DO_DEFERRED_REPCAT_ADMIN は、このプロシージャのコール元のユーザーから送られてくる管理要求だけを実行します。

- 関連したデータベースが実行中であり、通信が可能かどうか確認する。
- アドバンスト・レプリケーション機能に必要なプライベート・データベース・リンクが使用可能であり、CONNECT TO句で指定されたユーザー（通常はレプリケーション管理者）が必要な権限を持っていることを確認する。Replication Managerセットアップ・ウィザードを使用してサイトをセットアップした場合は、何も問題はありません。

遅延トランザクションの管理

非同期データ伝播を使用しているアドバンスト・レプリケーションでは、遅延トランザクションおよびジョブ・キューの内部システムを使用して、あるサイトから別のサイトに変更内容が送信されます。Replication Managerを使用すると、アドバンスト・レプリケーション・システムの各サーバーでキューに入れられた、遅延トランザクションとエラー・トランザクションを参照および管理できます。

遅延トランザクションの表示

接続先別にレプリケーション・サイトの遅延トランザクションのサマリー・リストを表示するには：

1. サイトの「管理」フォルダをオープンする。
2. 「接続先別遅延トランザクション」フォルダをクリックする。

Replication Managerの詳細パネルには、接続先別に保留中の遅延トランザクションのサマリー情報がすべて表示されます。

特定の接続先の個々の遅延トランザクションを表示するには：

1. 「接続先別遅延トランザクション」フォルダで接続先をクリックする。
2. ツリー・パネルの接続先を拡張する。

個々の遅延トランザクションのプロパティを表示するには：

1. 個々の遅延トランザクションをクリックする。
2. 「プロパティ」ツールバー・ボタンをクリックする。

個々の遅延トランザクションの遅延コールを表示するには：

1. 個々の遅延トランザクションを拡張する。
2. トランザクションの遅延コールとコール先、およびトランザクションの接続先を確認する。

注意：Replication Manager では、詳細パネルの表示は自動的に更新されません。保留中の遅延トランザクションのより新しいリストを取得するには、画面をリフレッシュします。

追加情報：DEFCALL および DEFCALLDEST、DEFLOB、DEFSCHEDULE、DEFTRAN、DEFTRANDEST ビューに問合せを行って、遅延トランザクションを調べることもできます。

遅延トランザクションの実行

アドバンスト・レプリケーション機能の各インスタンスに 1 つ以上の SNP プロセスがあり、必要なスケジュール・リンクで、定期的な間隔でジョブを自動的に起動し、ターゲットの接続先で遅延トランザクションを自動的に実行します。遅延トランザクションの自動実行を待機せずに実行する場合は、Replication Managerを使用して特定の接続先の保留中の遅延トランザクションをすべて手動で送信します。

1. サイトの「接続先別遅延トランザクション」フォルダの接続先をクリックする。
2. 「プロパティ」ツールバー・ボタンをクリックする。
3. 「データベース接続先の表示」プロパティ・シートの「すべてのトランザクションをすぐに再実行する」をクリックして、対応する遅延トランザクションをすべて強制的に実行する。

同等の API: DBMS_DEFER_SYS.PUSH

遅延トランザクションの削除

ローカル・サイトのキューにある遅延トランザクションが受信サイトに送信されたため受信サイトでエラー・トランザクションが発生する場合があります。そのような場合、遅延トランザクションをローカル・キューから削除して、エラー・トランザクションの発生を防止するのが適切な処置です。

警告：エラー・トランザクションの詳細、および遅延トランザクションを削除する場合の詳細は、6-9 ページの「エラー・トランザクションの管理」を参照してください。

個々の遅延トランザクションを削除するには：

1. 個々の遅延トランザクションを右クリックする。
2. 「遅延トランザクションの削除」をクリックしてターゲット・トランザクションをローカルで削除するか、または「すべての接続先から削除」をクリックしてすべてのターゲット・レプリケーション・サイトからターゲット・トランザクションを削除する。

同等の API: DBMS_DEFER_SYS.DELETE_TRAN

エラー・トランザクションの管理

Oracle が遅延トランザクションをスナップショット・サイトまたはマスター・サイトから別のマスター・サイトへ送信するとき、Oracle はトランザクションがリモート・サイトに正常に伝播するまでローカル・キューから削除されないようにします。トランザクションがマスター・サイトに正常に伝播しても、サイトで正常に適用されないことがあります。遅延トランザクションの適用のエラーは、更新しようとする表に使用可能な領域が足りないなどのデータベースに関わる問題、または解消不可能なレプリケーション競合の結果であることがあります。エラーが生じた場合、Oracle は、受信マスター・サイトで次のアクションを実行します。

- トランザクションのロールバック
- エラー・トランザクションの受信サイトのレプリケーション・カタログへの記録

エラー・トランザクションの表示

レプリケート環境内の各サイトはエラーがないかどうか頻繁にチェックし、エラーがあれば解決しなければなりません。レプリケーション・サイトのローカル・エラー・トランザクションのサマリー・リストを表示するには：

1. サイトの「管理」フォルダをオープンする。
2. 「ローカル・エラー」フォルダをクリックする。

Replication Manager の詳細パネルには、サイトのすべてのエラー・トランザクションのサマリー情報が表示されます。

エラー・トランザクションを正しく解決するには、トランザクションが実行しているジョブに関する詳しい情報が必要です。遅延トランザクションは、一連の遅延リモート・プロシージャ・コールから構成され、トランザクションの一貫性を維持するために必ず所定の順序で適用されます。

個々のエラー・トランザクションの詳細情報を表示するには：

1. 個々のエラー・トランザクションをクリックする。
2. 「プロパティ」ツールバー・ボタンをクリックする。

個々のエラー・トランザクションの遅延コールを表示するには：

1. 個々のエラー・トランザクションを拡張する。
2. トランザクションの遅延コールとコール先、およびトランザクションの接続先を確認する。

エラー・トランザクションの解決

エラー・トランザクションの原因を判別した後、エラーを修正し、さらに接続先サイトで次のアクションを1つ以上実行しなければならない場合があります。

- エラー・トランザクションを再実行する。
- エラー・トランザクションをローカル・サイトから削除する。

エラー・トランザクションの実行

エラー・トランザクションの原因となる問題を解決したら、エラー・トランザクションを再実行できます。エラー・トランザクションが正常に実行されると、そのトランザクションはローカル・サイトのレプリケーション・カタログから自動的に削除されます。

エラー・トランザクションを再実行するには：

1. 個々のエラー・トランザクションを右クリックする。
2. 「トランザクションの再実行」をクリックする。

同等の API: DBMS_DEFER_SYS.EXECUTE_ERROR

エラー・トランザクションが受信サイトで再実行されるときに、元の受信者のセキュリティ・コンテキストでそのトランザクションが実行されます。元の受信者が有効なユーザーでなくなった場合（つまり、ユーザーが削除された場合）は、DBMS_DEFER_SYS.EXECUTE_ERROR_AS_USER プロシージャを使用して別のセキュリティ・コンテキストのもとで遅延トランザクションを再実行できます。このプロシージャは、表 9-25 「DISABLED の戻り値」を参照してください。

注意：単一のエラー・トランザクションを再実行した場合、正常に実行されても Oracle はトランザクションをコミットしません。トランザクションの結果が正しければ、SQL コマンドの COMMIT WORK を発行する必要があります。EXECUTE_ERROR によって複数のトランザクションが再実行される場合、各トランザクションは完了するたびにコミットされます。

エラー・トランザクションの削除

エラー・トランザクションを（トランザクションを再実行して解決するのではなく）手動で解決しなければならない場合もあります。エラー・トランザクションの原因となる問題を解決した後、エラー・トランザクションを削除しなければなりません。エラー・トランザクションを削除するには：

1. 個々のエラー・トランザクションをクリックする。
2. 「削除」ツールバー・ボタンをクリックする。

同等の API: DBMS_DEFER_SYS.DELETE_ERROR

ローカル・ジョブの管理

Oracle では、各サイトのジョブ・キューのジョブを実行して、アドバンスド・レプリケーション環境の管理に必要な特定の作業を自動的に完了します。たとえば、Replication Manager を使用してリモート・サイトへのスケジュール・リンクを作成および使用可能にすると、ローカル・サーバーでは、ローカルの変更内容をリモート・マスターに送信するために定期的に実行されるジョブがジョブ・キューに入れられます。

Replication Manager に用意されているいくつかの機能を使用すると、アドバンスド・レプリケーション・システムの各サーバーのジョブ・キューを参照および管理できます。以降の項では、サーバーのローカル・ジョブの参照方法および管理方法を詳しく説明します。

サイトのローカル・ジョブの表示

次の操作で、サイトのローカル・ジョブ・キューにジョブが作成されます。

- スケジュール・リンクを作成すると、遅延トランザクションをターゲットのリモート・マスター・サイトに送信するジョブが作成されます。このタイプのジョブの PL/SQL には、API コール DBMS_DEFER_SYS.PUSH が含まれます。
- サイトの遅延トランザクション・キューのバージをスケジュールすると、この操作を実行するジョブが作成されます。このタイプのジョブの PL/SQL には、API コール DBMS_DEFER_SYS.PURGE が含まれます。
- マスター・グループを作成すると、グループを管理する別のマスター・サイトに対応する管理要求を送信するジョブが作成されます（接続先ごとに 1 ジョブ）。このタイプのジ

ジョブの PL/SQL には、API コール DBMS_REPCAT.DO_DEFERRED_REPCAT_ADMIN が含まれます。

- リフレッシュ・グループを作成すると、グループの定期リフレッシュを実行するジョブが作成されます。このタイプのジョブの PL/SQL には、API コール DBMS_REFRESH.REFRESH が含まれます。

レプリケーション・サイトのローカル・ジョブ・キューのサマリー・リストを表示するには：

1. サイトの「管理」フォルダをオープンする。
2. 「ローカル・ジョブ」フォルダをクリックする。

Replication Managerの詳細パネルには、すべてのローカル・ジョブのサマリー情報が表示されます。

個々のローカル・ジョブのプロパティを表示するには：

1. 個々のジョブをクリックする。
2. 「プロパティ」ツールバー・ボタンをクリックする。

「ジョブの編集」プロパティ・シートのページでは、次のようなジョブのさまざまなプロパティを参照できます。

- ジョブの次回実行日付および実行間隔
- ジョブの PL/SQL テキスト
- ジョブの最終実行日付
- ジョブの現在の状態（標準またはブレイク）
- ジョブの実行失敗回数
- ジョブが実行されるユーザー環境

ローカル・ジョブのプロパティの編集

特定の状況では、ローカル・ジョブのプロパティを編集場合があります。たとえば、マスター・グループの管理要求のデフォルトの実行間隔は10分です。この設定を編集する唯一の方法は、対応するジョブのスケジュール・プロパティを編集することです。ローカル・ジョブのプロパティを編集するには：

1. ターゲット・ジョブをクリックする。
2. 「プロパティ」ツールバー・ボタンをクリックする。

「ジョブの編集」プロパティのシートでは、次のようなジョブのプロパティを編集できます。

- ジョブの次回実行日付および実行間隔
- ジョブの PL/SQL テキスト

ジョブの手動実行

ジョブの次回実行日を待つかわりに、ジョブを強制的に実行できます。ジョブをすぐに実行するには：

1. ターゲット・ジョブをクリックする。
2. 「プロパティ」ツールバー・ボタンをクリックする。
3. 「即時実行」をクリックする。

ジョブを使用禁止および使用可能にする

特定の状況で、ジョブを一時的に使用禁止に（ブレイク）して、後でそのジョブを再度使用可能にする必要がある場合があります。または、ジョブが正しく実行されない原因となっている問題を修正した後で、ブレイク・ジョブを使用可能にする場合があります。たとえば、ジョブで使用するデータベース・リンクを誤って削除した場合、ジョブは失敗し、結果的に（16回連続で失敗した後）ブレイクします。必要なデータベース・リンクを再作成したら、ブレイク・ジョブを使用可能にできます。ローカル・ジョブをブレイクまたは使用可能（標準）にするには：

1. ターゲット・ジョブをクリックする。
2. 「プロパティ」ツールバー・ボタンをクリックする。
3. 「ブレイク / 標準」をクリックする。

ジョブに関する問題の診断方法

アドバンスド・レプリケーション環境での操作の多くは、作業を実行するジョブに依存します。システムの監視中に、ジョブに関する問題が判明する可能性があります。Replication Manager を使用して、ローカル・ジョブを定期的に監視しなければなりません。

ジョブの実行が失敗したら、まずジョブの状態をチェックしてください。ジョブの状態がブレイクの場合、エラー情報のALERTログおよびトレース・ファイルをチェックして、エラーを修正します。ジョブがブレイクでない（標準）場合、Oracle はジョブを再実行します。

ジョブに問題がないのにジョブが実行されない場合は、SNP バックグラウンド・プロセスの可用性の問題が発生している可能性があります。初期化パラメータJOB_QUEUE_PROCESSES をチェックして使用可能なバックグラウンド・プロセスの数を判断し、JOB_QUEUE_INTERVAL をチェックして各バックグラウンド・プロセスが起動される頻度を判断します。また、DBA_JOBS_RUNNING ビューに問合せを実行してこれらのプロセスがどのジョブを現在実行しているかを調べたり（ジョブの暴走の問題が起きている場合があります）、アラート・ログおよびトレース・ファイルから、バックグラウンド・プロセスで発生している問題についての追加情報を得ることもできます。

レプリケーション・システムにおけるデータベースのバックアップおよび回復

アドバンスト・レプリケーションを使用するデータベースは分散データベースです。アドバンスト・レプリケーション・データベースのバックアップを作成するときは、『Oracle8 Server 管理者ガイド』で概説している分散データベースのバックアップのガイドラインに従ってください。アドバンスト・レプリケーション・データベースを回復するときは、『Oracle8 Server 管理者ガイド』で説明している協調分散回復のガイドラインに従ってください。

協調分散回復のガイドラインに従わないと、アドバンスト・レプリケーション・データベースの一貫性が保証されません。たとえば、復旧されたマスター・サイトが、異なるトランザクションを異なるマスターに伝播していることがあります。不正確な回復操作を修正するために、追加のステップを実行しなければならない場合があります。その方法の1つは、回復されたデータベース内のすべてのレプリケート・オブジェクトを削除して、再作成することです。

推奨：復元されたデータベースから未解決の遅延トランザクションおよび遅延エラー・レコードを削除し、レプリケート・オブジェクトを削除および再作成する前に未解決の分散トランザクションを解決してください。回復されたデータベースがレプリケート環境のマスター定義サイトであった場合、オブジェクトを削除および再作成する前に新しいマスター定義サイトを指定する必要があります。復元されたデータベースがマスターであるスナップショット、および復元されたデータベース内のスナップショットは完全にリフレッシュする必要があります。

データに対する継続的なアクセスを可能にするため、マスター定義サイトの変更（復元されるデータベースがマスター定義サイトであった場合）またはスナップショット・サイトのマスターの変更（それらのマスター・サイトが復元されている場合）が必要な場合があります。

インポートされたデータに対するチェックの実行

レプリケート・オブジェクト、およびアドバンスト・レプリケーション機能（たとえば、REPSITES ビュー）で使用するオブジェクトのエクスポート/インポートを実行したら、DBMS_REPCAT パッケージの REPCAT_IMPORT_CHECK プロシージャを実行してください。

次の例では、このプロシージャがスナップショット・サイトの ACCT レプリケート・オブジェクト・グループのオブジェクトをチェックし、適切なオブジェクト識別子および状態値が存在することを確認します。

```
DBMS_REPCAT.REPCAT_IMPORT_CHECK(  gname    =>    'acct',
                                   master    =>    FALSE);
```

追加情報：REPCAT_IMPORT_CHECK プロシージャのパラメータは、9-120 ページの表 9-178 を参照してください。例外は、9-120 ページの表 9-179 を参照してください。

正常に終了した競合解消の監査

Oracle で更新または削除、一意性の競合が検出され正常に解消された場合はいつでも、REPRESOLUTION_STATISTICS ビューを問い合わせることによって、競合解消にどの方法が使用されたかの情報を見ることができます。このビューが更新されるのは、競合に関与した表の競合解消の統計収集がオンに設定されている場合だけです。

競合解消の統計収集

表に対する更新および削除、一意性の競合の正常な解消に関する情報を収集するには、DBMS_REPCAT パッケージの REGISTER_STATISTICS プロシージャを使用します。次の例では、ACCT_REC スキーマ内の EMP 表に関する統計が収集されます。

```
DBMS_REPCAT.REGISTER_STATISTICS(sname => 'acct_rec',
                                oname  => 'emp');
```

追加情報：REGISTER_STATISTICS プロシージャのパラメータは、9-117 ページの表 9-172 を参照してください。例外は、9-117 ページの表 9-173 を参照してください。

競合解消の統計の表示

表に対して REGISTER_STATISTICS をコールした後、その表に関して正常に解消された各競合は REPRESOLUTION_STATISTICS ビューにロギングされます。解消されていない競合に関する情報は、そのオブジェクトが登録されたかどうかにかかわらず、常に DEFERROR ビューにロギングされます。

追加情報：REPRESOLUTION_STATISTICS ビューのパラメータは、10-15 ページの「REPRESOLUTION_STATISTICS ビュー」を参照してください。DEFERROR ビューの詳細は、10-19 ページの「DEFERROR ビュー」を参照してください。

競合解消の統計収集の取消し

表に対する更新または削除、一意性の競合の正常終了した解消に関する情報の収集を取り消す場合、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、ACCT_REC スキーマ内の EMP 表に関する統計の収集が取り消されます。

```
DBMS_REPCAT.CANCEL_STATISTICS(sname => 'acct_rec',
                               oname  => 'emp');
```

追加情報：CANCEL_STATISTICS プロシージャのパラメータは、9-72 ページの表 9-97 を参照してください。

統計情報の削除

更新および削除、一意性の競合の正常終了した解消に関する情報を記録する表を登録しておく、DBMS_REPCAT パッケージの PURGE_STATISTICS プロシージャをコールして、REPRESOLUTION_STATISTICS ビューからこの情報を削除できます。

次の例では、1月1日から3月31日までの期間にEMP表での挿入および更新、削除によって生じた競合の解消に関して収集された統計が削除されます。

```
DBMS_REPCAT.PURGE_STATISTICS(sname      =>    'acct_rec',
                              oname       =>    'emp',
                              start_date  =>    '01-JAN-95',
                              end_date    =>    '31-MAR-95');
```

追加情報: PURGE_STATISTICS プロシージャのパラメータは、9-114 ページの表 9-166 を参照してください。例外は、9-114 ページの表 9-167 を参照してください。

レプリケート表相互間の相違の判別

レプリケート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ことを定期的にチェックしたい場合があります。両方のサイトがリリース 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したり、任意に調整することができます。

DIFFERENCES

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の表は各行を含むサイトを示すために使用されます。

RECTIFY

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の表にはあって第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の表にはあって第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復元する順序は、次のとおりです。

1. 表のコピーを1つ選択して「基準」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。
2. 表の中のすべての行および列について違いをチェックしなければならないのか、またはサブセットだけをチェックすればよいのかを判別します。たとえば、前回このチェックを行った後、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なければなりません。
3. 表の中のどの列をチェックするかを決定した後、比較の結果を保持する表を2つ作成しなくてはなりません。

比較する列のデータを保持する表を1つ作成しなくてはなりません。たとえば、EMPLOYEE 表の EMPNO 列および SAL 列、BONUS 列を比較する場合、CREATE 文は次に示す CREATE 文に類似していなくてはなりません。

```
CREATE TABLE missing_rows_data
(
    empno    NUMBER,
    sal      NUMBER,
    bonus    NUMBER)

```

行位置を示す表も作成しなくてはなりません。この表には、次の例に示すデータ型の列が3つなければなりません。

```
CREATE TABLE missing_rows_location
(
    present   VARCHAR2(128),
    absent    VARCHAR2(128),
    r_id      ROWID
)

```

4. 比較する表があるオブジェクト・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、最初に表を休止しないで調整すると、データに矛盾が生じることがあります。
5. 「基準」表があるサイトで DBMS_RECTIFIER_DIFF.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 EMPLOYEE 表を比較する場合、次のようなプロシージャ・コールになります。

```
DBMS_RECTIFIER_DIFF.DIFFERENCES(
    sname1           => 'hr',
    oname1           => 'employee',
    reference_site   => 'ny.com',
    sname2           => 'hr',
    oname2           => 'employee',
    comparison_site  => 'sf.com',
    where_clause     => '',
    column_list      => 'empno,sal,bonus',
    missing_rows_sname => 'scott',
    missing_rows_oname1 => 'missing_rows_data',
    missing_rows_oname2 => 'missing_rows_location',
    missing_rows_site  => 'ny.com',
    commit_rows      => 50);

```

図 6-1 に、EMPLOYEE 表の2つのレプリカと、これらのレプリカで DIFFERENCES プロシージャを実行した場合、欠落行出力表がどのようなになるかの例を示します。

図 6-1 レプリカ間の相違の判別

NY.COM の EMPLOYEE 表				
empno	ename	deptno	sal	bonus
100	Jones	20	55,000	3,500
101	Kim	20	62,000	1,000
102	Braun	20	43,500	1,500

SF.COM の EMPLOYEE 表				
empno	ename	deptno	sal	bonus
100	Jones	20	55,000	3,500
101	Kim	20	62,000	2,000
102	Braun	20	43,500	1,500
103	Rama	20	48,750	2,500

MISSING_ROWS_DATA 表			
empno	sal	bonus	rowid
101	62,000	1,000	000015E8.0000.0002
101	62,000	2,000	000015E8.0001.0002
103	48,750	2,500	000015E8.0002.0002

MISSING_ROWS_LOCATION 表		
present	absent	rowid
nycom	sf.com	000015E8.0000.0002
sf.com	nycom	000015E8.0001.0002
sf.com	nycom	000015E8.0002.0002

2 つの欠落行出力表が、ROWID と R_ID 列によって関連付けられたことに注目してください。

6. ここで、次の例に示すように、DBMS_RECTIFIER_DIFF.RECTIFY プロシージャをコールして、「基準」サイトの表に等しくなるように「比較」サイトの表を調整できます。

```
DBMS_RECTIFIER_DIFF.RECTIFY(
    sname1          => 'hr',
    oname1          => 'employee',
    reference_site   => 'ny.com',
    sname2          => 'hr',
    oname2          => 'employee',
    comparison_site  => 'sf.com',
    column_list      => 'empno,sal,bonus',
    missing_rows_sname => 'scott',
    missing_rows_oname1 => 'missing_rows_data',
    missing_rows_oname2 => 'missing_rows_location',
    missing_rows_site  => 'ny.com',
    commit_rows      => 50);
```

必要な挿入および削除を実行している間、その変更を伝播したくない場合、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、まず必要な DELETE をすべて実行し、次に INSERT をすべて実行します。これによって、PRIMARY KEY 制約の違反はなくなります。

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ていないことを確認する必要があります。欠落行出力表が持つ情報を使用して、表を直接更新しなくてはならない場合もあります。そのような場合は、必ず欠落行出力表から該当の行を DELETE してください。

7. RECTIFY プロシージャが正常に実行されると、欠落行出力表は空になります。次に、レプリケート表の残りのコピーに対して、ステップ5および6を繰り返すことができます。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「基準」表を使用してください。
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できます。

一般的な問題のトラブルシューティング

以降の項では、アドバンスド・レプリケーション環境の管理に関するトラブルシューティングのガイドラインについて説明します。

データベース・リンクに関する問題の診断方法

データベース・リンクが正常に機能していないと思ったら、Oracle Enterprise Manager または Oracle Server Manager、その他のツールを使用してデータベース・リンクを削除して作成しなおすことができます。

- スケジュールされた間隔が希望のものであることを確認します。
- スケジュールされた間隔が、必要な実行時間よりも短くないことを確認します。

指定のサイトへのデータベース・リンクで、ある接続修飾子を使用した場合、そのサイトにリンクするその他のサイトでもまったく同じ接続修飾子を使用する必要があります。たとえば、次のようにデータベース・リンクを作成するとします。

```
CREATE DATABASE LINK myethernet CONNECT TO repsys IDENTIFIED BY secret
USING 'connect_string_myethernet'
```

この場合、myethernet データベース・リンクに関連するすべてのサイトは、マスターであれスナップショットであれ、'connect_string_myethernet' 接続文字列を含んでいなければなりません。

マスター・サイトに関する問題の診断方法

マルチマスター・アドバンスド・レプリケーション・システムで発生する問題にはさまざまなものがあります。以降の項では、これらの問題とトラブルシューティング方法をいくつか説明します。

新しいマスター・サイトでレプリケート・オブジェクトが作成されない場合

マスター・グループに新しいマスター・サイトを追加したにもかかわらず、適切なオブジェクトが新しいサイトで作成されない場合には、次のようにしてください。

- 必要なプライベート・データベース・リンクが、新しいマスター・サイトと既存のマスター・サイトとの間に存在することを確認する。Replication Manager セットアップ・ウィザードを使用してサイトをセットアップした場合は、何も問題はありません。各既存サイトから新しいサイトへのリンクおよび新しいサイトから各既存サイトへのリンクの両方を持っていないとなりません。
- すべてのサイトで管理要求が正常に終了していることを確認する。管理要求がまだ実行されていない場合、保留になっている管理要求を手動で実行し、即時に操作を完了させることができます。

DDL の変更がマスター・サイトに伝播されない場合

新しいマスター・グループ・オブジェクトを作成した、またはマスター定義サイトでマスター・グループ・オブジェクト定義を変更したのに、変更がマスター・サイトに伝播されない場合には、すべてのサイトで管理要求が正常に終了していることを最初に確認してください。管理要求が保留になっている場合は、手動で実行して即時に操作を完了させることができます。

レプリケーション API を通して DDL 文を実行すると、DDL を送信するユーザー名において DDL 文が実行されます。DDL 文を送信者のスキーマ以外のスキーマ内のオブジェクトに適用するときは、送信者が文を実行する権限を所有していなければなりません。また、文の中でスキーマを明示的に指定する必要があります。たとえば、レプリケーション管理者のユーザーが DBMS_REPCAT.CREATE_MASTER_REPOBJECT プロシージャの DDL_TEXT パラメータとして次のように記述するとします。

```
CREATE TABLE scott.new_emp AS SELECT * FROM hr.emp WHERE...
```

それぞれの表名にスキーマ名が含まれているので、レプリケーション管理者が SCOTT または HR、別のユーザーであっても、必要な権限を所有していれば、この文が正しく機能します。

提案：各スキーマ・オブジェクトの名前を適切なスキーマで修飾してください。

DML 変更が他のサイトに非同期で伝播されない場合

マスター・サイトのデータを更新し、その変更がレプリケート環境の他のサイトに非同期で正しく伝播されない場合は、次の処理を行ってください。

- Replication Manager を使用して、対応する遅延トランザクションが接続先に送信されているかどうかをチェックする。送信されていない場合、スケジュールされたリンクがキューを接続先サイトに送信するまでの待ち時間をチェックすることもできます。リンクによるスケジュール送信を待てない場合、遅延トランザクションを手動で実行することができます。6-9 ページの「遅延トランザクションの実行」を参照してください。
- スケジュールされたリンクの送信間隔が経過しても関連する遅延トランザクションが送信されない場合は、リンクに対応するジョブをチェックします。6-11 ページの「サイトのローカル・ジョブの表示」を参照してください。

- 遅延トランザクションが接続先に伝播されても、エラーによって実行できない場合があります。エラー・トランザクションの詳細は、6-9ページの「エラー・トランザクションの管理」を参照してください。

DML をレプリケート表に適用できない場合

レプリケート表を修正しようとしたときに DEFERRED_RPC_QUIESCE 例外を受け取った場合、ローカル・サイトにある1つ以上のマスター・グループは、「休止中」または「休止状態」になっています。先に進むには、レプリケーション管理者がグループのレプリケーション・アクティビティを再開しなければなりません。

複数更新および制約の違反

レプリケート表に適用される単一の UPDATE 文は、複数行を更新できます。UPDATE 文によって、ゼロ以上の更新要求が出され、更新される行ごとに1つずつ遅延実行のためのキューに入れられます。Oracle は、文単位で最後に制約のチェックをまとめて行うので、この機能は制約が存在するときに問題になります。たとえば、複数行の UPDATE 文で一意性制約違反が発生しなくても、場合によっては、同じ順序で1行ずつ UPDATE すると一意性制約違反が発生します。

UPDATE の順序が重要である場合、適切な順序で1度に1つの行を更新してください。これによって、遅延 RPC キューにおける更新要求の順序を定義できます。

レプリケート・オブジェクトの再作成

パッケージまたプロシージャ、ビューなどのオブジェクトをマスター・グループに追加する場合、オブジェクトの状態は VALID でなければなりません。オブジェクトの状態が INVALID の場合は、マスター・グループにオブジェクトを追加する前に、オブジェクトを再コンパイルするか、またはオブジェクトを削除して再作成してください。

表のレプリケーション・サポートを生成できない場合

表のレプリケーション・サポートを生成すると、Oracle はローカル・サイトで内部トリガーをアクティブにします。表が変更を非同期で伝播する場合、このトリガーは DBMS_DEFER パッケージを使用して、ローカル遅延トランザクション・キューに入れられるコールを作成します。DBMS_REPCAT および DBMS_DEFER などのアドバンスト・レプリケーションとともに含まれるほとんどのパッケージ用の EXECUTE 権限は、レプリケート・オブジェクトを所有するレプリケーション管理者およびユーザーに付与しておく必要があります。Replication Manager セットアップ・ウィザードおよび DBMS_REPCAT_ADMIN パッケージは、数多くの一般的なレプリケーション例のレプリケーション管理者が必要とする権限付与を実行します。ただし、レプリケート・オブジェクトの所有者がレプリケーション管理者でない場合には、オブジェクト所有者に DBMS_DEFER に対する EXECUTE 権限を明示的に付与しなければなりません。

レプリケート・プロシージャまたはレプリケート・トリガーに関する問題

予期しない未解消の競合が発見され、しかもプロシージャのレプリケーションおよび行レベルのレプリケーションが表に混在している場合、プロシージャを注意深く検証して、レプリケート・プロシージャが競合の原因ではないかどうか確認します。プロシージャの更新と行レベルの更新の実行順序による競合がないかどうか確認します。レプリケート・プロシージャの中で更新を実行する前に、表を EXCLUSIVE モードにロックしたかどうか（または、行レベルの更新の競合を避けるための他の方法を使用しているかどうか）をチェックします。レプリケート・プロシージャの最初に、行レベルのレプリケーションを使用禁止にし、最後に再び使用可能にしていることをチェックします。プロシージャの実行時に例外が発生した場合も、行レベルのレプリケーションを再び使用可能にしていることを確認します。さらに、レプリケート・プロシージャがすべてのマスター・サイトにおいて実行されたことをチェックします。レプリケート表で定義したレプリケート・トリガーのすべてについて、同様のチェックを行わなければなりません。

遅延トランザクション・キューに関する問題の診断方法

サイトの遅延トランザクションが接続先に送信されない場合、いろいろな理由が考えられます。以降の項では、考えられる原因のいくつかを説明します。

スケジュール・リンクのジョブのチェック

スケジュール・リンクを作成すると、対応するジョブがサイトのジョブ・キューへ追加されます。スケジュール・リンクが周期的な間隔で遅延トランザクションを送信するように設定していて、問題が検出される場合、まず最初にジョブ・キューに関する問題ではないかどうか確認しなければなりません。

分散トランザクションの問題

Oracle でシリアル伝播を使用してリモート・サイトに遅延トランザクションを送信するときは、分散トランザクションを使用して、トランザクションがリモート・サイトで正しくコミットされてからローカル・サイトのキューから削除されたことを確認します。分散トランザクション（2 フェーズ・コミット）に関する問題の診断の詳細は、『Oracle8 Server 分散システム』を参照してください。

データベース・リンク指定の不備

指定したリモート・サイトにトランザクションが送信されていない場合、トランザクションの接続先の指定方法に問題がある場合があります。スケジュール・リンクを作成したら、完全なデータベース・リンク名を指定しなければなりません。Replication Managerを使用している場合、問題はありません。

不適切なレプリケーション・カタログ・ビュー

ビューの定義が不適切であると、遅延トランザクションが正しく動作しない場合があります。DEFCALLDEST ビューと DEFTRANDEST ビューは、CATDEFER.SQL と CATREPC.SQL で別々に定義されます。CATREPC.SQL の定義は、アドバンスド・レプリケーションを使用するときは常に使用しなければなりません。CATDEFER.SQL を（再）ロードする場合は、その後で CATREPC.SQL のビュー定義がロードされるようにしてください。

スナップショットに関する問題の診断方法

アドバンスド・レプリケーション・システムのスナップショット・サイトで発生する問題にはさまざまなものがあります。以降の項では、これらの問題とトラブルシューティング方法をいくつか説明します。

スナップショット・サイトでレプリケート・オブジェクトの作成に関する問題
スナップショット・サイトで新しいオブジェクトの作成が失敗した場合、次のようにしてください。

- 更新可能なスナップショットの場合、関連したマスター表がスナップショット・ログを持っているかどうかチェックする。
- オブジェクトを作成するために必要な権限を持っていることを確認する。4-4 ページの「スナップショット・サイト・レプリケーションの準備」を参照してください。
- 既存のスナップショットをスナップショット・グループに追加しようとする場合、スナップショットをグループに追加するときにスナップショットを再作成する。

スナップショット・リフレッシュに関する問題

スナップショットのリフレッシュに関する問題が発生している場合には、次の処理を行ってください。

- リフレッシュがスケジュールされているかどうかを判断するために、DBA_SNAPSHOTS ビュー内の NEXT 値をチェックする。
- リフレッシュ間隔が経過している場合は、スナップショット・リフレッシュに対応するジョブ番号の DBA_REFRESH ビューをチェックして、6-11 ページの「ローカル・ジョブの管理」のジョブ・キューに関する問題の診断方法の指示に従う。
- また、2つのスナップショット間のマスター・ディテール関係を定義しようとした場合に、エラーを検出することもある。宣言参照整合性制約を使用して、マスター表でだけマスター・ディテール関係を定義する必要があり、この関係を保つため、関連のあるスナップショットは同じリフレッシュ・グループに入れる必要があります。
- スナップショットのリフレッシュが絶え間なく繰り返されている場合、指定したリフレッシュ間隔をチェックする。この間隔は、スナップショットがリフレッシュする前に評価されます。指定した間隔がスナップショットをリフレッシュするのにかかる時間より短い場合、SNP バックグラウンド・プロセスが未解決ジョブのキューをチェックするたびに、スナップショットがリフレッシュすることになります。
- スナップショットのマスター・サイトで未解決の競合が記録されている場合、パラメータ REFRESH_AFTER_ERRORS を TRUE に設定するとスナップショットをリフレッシュできる。このパラメータは、スナップショット・リフレッシュ・グループの作成または変更時に設定できます (Replication Manager プロパティ・シートに対応するパラメータがあります)。
- スナップショット・ログが大きくなりすぎる場合は、2-27 ページの「スナップショット・ログの管理」を参照してください。

- 同じリフレッシュ・グループ内のスナップショットは、行が単一トランザクションで更新される。このようなトランザクションは非常に大きくなる場合があり、スナップショット・サイトで大きなロールバック・セグメント（リフレッシュ中に使用するロールバック・セグメント）が必要になります。または、リフレッシュをより頻繁に実行してトランザクションを小さくする必要があります。
- Oracle エラー ORA-12004 が発生した場合は、マスター・サイトでスナップショット・ログを管理しようとしてロールバック・セグメントが足りなくなったか、スナップショット・ログの情報が古い可能性がある。たとえば、スナップショット・ログがパージまたは再作成された可能性があります。2-27 ページの「スナップショット・ログの管理」を参照してください。
- 単一表の完全リフレッシュは内部的に TRUNCATE 機能を使用して、処理速度を向上させ、ロールバック・セグメントの必要量を少なくする。しかし、スナップショットのリフレッシュが完了するまで、ユーザーは一時的にスナップショット内のデータが確認できない場合があります。複数のスナップショットをリフレッシュするときは（リフレッシュ・グループなど）、TRUNCATE 機能を使用しないでください。
- マスター表を再編成するときは（システム資源を再利用する目的などで）、マスター表を TRUNCATE して、ROWID スナップショットの完全リフレッシュを実行する必要があります。そうしないと、スナップショットがマスター表の ROWID を不適切に参照することになります。詳細は、2-42 ページの「マスター表の再編成と ROWID スナップショット」を参照してください。

追加情報: 2-38 ページの「リフレッシュの問題のトラブルシューティング」を参照してください。

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付けられたさまざまなビュー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 6-1 に、各ビューをコールするための適切なプロシージャを一覧表示します。

表 6-1 アドバンスド・レプリケーション機能ビューのコメントの更新

ビュー	DBMS_REPCAT プロシージャ	その他の情報
REPGROUP	COMMENT_ON_REPGROUP(gname IN VARCHAR2, Comment IN VARCHAR2)	COMMENT_ON_REPGROUP プロシージャのパラメータは、表 9-103 を参照してください。例外は、表 9-104 を参照してください。
REPOBJECT	COMMENT_ON_REPOBJECT(sname IN VARCHAR2, oname IN VARCHAR2, type IN VARCHAR2, comment IN VARCHAR2)	COMMENT_ON_REPOBJECT プロシージャのパラメータは、表 9-107 を参照してください。例外は、表 9-108 を参照してください。
REPSITES	COMMENT_ON_REPSITES(gname IN VARCHAR2, master IN VARCHAR, comment IN VARCHAR2)	COMMENT_ON_REPSITES プロシージャのパラメータは、表 9-105 を参照してください。例外は、表 9-106 を参照してください。
REPCOLUMN_ GROUP	COMMENT_ON_COLUMN_GROUP(sname IN VARCHAR2, oname IN VARCHAR2, column_group IN VARCHAR2, comment IN VARCHAR2)	COMMENT_ON_COLUMN_GROUP プロシージャのパラメータは、表 9-99 を参照してください。例外は、表 9-100 を参照してください。
REPPRIORITY_ GROUP	COMMENT_ON_PRIORITY_GROUP(gname IN VARCHAR2, pgroup IN VARCHAR2, comment IN VARCHAR2)	COMMENT_ON_PRIORITY_GROUP プロシージャのパラメータは、表 9-101 を参照してください。例外は、表 9-102 を参照してください。
REPPRIORITY_ GROUP (サイト優先グルー プ)	COMMENT_ON_SITE_PRIORITY(gname IN VARCHAR2, name IN VARCHAR2, comment IN VARCHAR2)	COMMENT_ON_SITE_PRIORITY プロシージャのパラメータは、表 9-101 を参照してください。例外は、表 9-102 を参照してください。

表 6-1 アドバンスド・レプリケーション機能ビューのコメントの更新

ビュー	DBMS_REPCAT プロシージャ	その他の情報
REPRESOLUTION (一意性の競合)	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、表 9-109 を参照してください。例外は、表 9-110 を参照してください。
REPRESOLUTION (更新の競合)	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、表 9-109 を参照してください。例外は、表 9-110 を参照してください。
REPRESOLUTION (削除の競合)	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、表 9-109 を参照してください。例外は、表 9-110 を参照してください。

上級テクニック

この章では、Oracle のレプリケート・データベース環境を実現するときに使用できる上級テクニックのいくつかについて説明します。

- プロシージャ・レプリケーションの使用方法
- 同期データ伝播の使用方法
- 動作保証の設計
- スナップショット・クローニングとオフライン・インスタンスエーション
- マルチマスター・レプリケーションのセキュリティ設定
- スナップショット・レプリケーションのセキュリティ設定
- アドバンスド・レプリケーション・スナップショット・サイトに対する代替セキュリティの設定
- 削除の競合の防止
- 動的所有権の競合防止の使用方法
- 変更をレプリケートしないで表を変更する

プロシージャ・レプリケーションの使用方法

プロシージャ・レプリケーションを使用すると、レプリケート環境で直列的に実行する大量の行を操作する、大規模なバッチ指向操作のパフォーマンスが向上します。

アプリケーションの好例は、古いデータや"論理的に"オンライン・データベースから消されたデータを削除するために、休業時間中に（15分ごとなど）実行されるバッチ操作（アーカイブ操作とも呼ばれる）です。プロシージャ・レプリケーションを使用して、削除した行をバッチする例は、7-24 ページの「削除の競合の防止」を参照してください。

プロシージャ・レプリケーションの制限

レプリケートされたプロシージャのすべてのパラメータは、IN パラメータでなければなりません。OUT モードおよび 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サポートされるデータ型は、NUMBER および DATE、VARCHAR2、CHAR、ROWID、RAW、BLOB、CLOB、NCHAR、NVARCHAR、NCLOB です。

Oracle では、レプリケートされたプロシージャによって発生した更新の競合は検出できません。レプリケートされたプロシージャは、自分自身で競合を検出し解消する必要があります。独自の競合解消ルーチンを作成することは困難なので、単に競合の発生を防止することが最善です。

次のガイドラインを遵守すると、プロシージャ・レプリケーションの使用時に表がすべてのサイトで一貫性を持つようになります。

- 遅延プロシージャ内での行レベルのレプリケーションの使用禁止にする必要がある。詳細は、7-30 ページの「変更をレプリケートしないで表を変更する」を参照してください。
- 次の項の 7-3 ページの「トランザクションの直列化」で説明しているように、同時に実行できるレプリケート・プロシージャは 1 つだけである。
- 遅延トランザクションは直列的に伝播されなければならない。詳細は、3-9 ページの「スケジュール・リンクのガイドライン」を参照してください。
- レプリケートされたプロシージャはパッケージ化する必要があり、このパッケージに機能を含むことはできない。スタンドアロンの遅延プロシージャとスタンドアロンのまたはパッケージ化した遅延機能は現在サポートされていない。
- 遅延プロシージャが参照してよいのは、ローカルで所有するデータだけである。
- プロシージャに、ローカルで生成したフィールドや値、または環境に依存する SQL 関数を使用してはならない（たとえば、SYSDATE をコールしてはならない）。
- データ所有権は、静的に区別されている必要がある（つまり、行の所有権がサイト間で移動してはならない）。

トランザクションの直列化

直列処理によって、データの一貫性が保証されます。アドバンスド・レプリケーション機能は、レプリケート・トランザクションを同時に伝播および実行します。たとえば、ローカル・データの更新を実行する2つのプロシージャ、AおよびBがあると想定します。ここで、次の動作を順序どおりに実行すると想定します。

1. AおよびBをローカルで実行します。
2. 他のノードにあるAおよびBのレプリケートを実行する要求をキューに入れます。
3. コミットします。

別のノードにあるAおよびBのレプリケートは、発行されたサイトでコミットされた順序と同じ順序で、完全に直列的に実行されます。しかし、AおよびBが発行されたサイトで同時に実行されると、生じる結果がローカルとリモートとで異なる場合があります。発行されたサイトでAおよびBを直列的に実行することで、すべてのサイトに同じ結果が生じることを保証します。トランザクションを直列的に伝播すると、どのような場合にもAとBが目的のサイトで直列的に実行されることが保証されます。

また、直列化が確実になるように考慮したプロシージャを作成できます。たとえば、問合せにSELECT ... FOR UPDATEを使用すると、パラレル伝播を使用する場合に発行元のサイトと宛先のサイトで直列性を保証できます。

レプリケートされたプロシージャのサポートの生成

プロシージャの開始では、行レベルのレプリケーション・サポートを使用禁止にして、その後、最後に再びサポートを使用可能にする必要があります。これによって、プロシージャを実行した結果として生じた更新が他のサイトに伝播しないようにできます。行レベルのレプリケーションは、それぞれ次のプロシージャをコールすることによって、使用可能および使用禁止になります。

- DBMS_REPUTIL.REPLICATION_ON
- DBMS_REPUTIL.REPLICATION_OFF

追加情報: 7-31ページの「アドバンスド・レプリケーションを使用禁止にする」を参照してください。

レプリケートされたパッケージ用のレプリケーション・サポートを生成すると、レプリケーション伝播担当者のスキーマにWrapperパッケージが作成されます。

注意: 現行の伝播担当者を登録解除すると、伝播担当者のスキーマに生成された既存のWrapperがすべて削除されます。新しい伝播担当者を登録した後で、ラップ・ストアド・プロシージャのレプリケーション・サポートを再生成する必要があります。

Wrapper パッケージの名前は元の名前と同じですが、接頭辞には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が使用されます。接頭辞を指定しない場合、デフォルトの接頭辞「defer_」が使用されます。Wrapper プロシージャには元のプロシージャと同じパラメータがあり、さらに、2つのパラメータ CALL_LOCAL および CALL_REMOTE が追加されます。これらの2つの BOOLEAN 型パラメータによって、プロシージャが実行される場所が決まります。CALL_LOCAL が TRUE の場合は、プロシージャがローカルに実行されます。CALL_REMOTE が TRUE の場合は、プロシージャは最終的にレプリケート環境内の他のすべてのマスター・サイトで実行されます。

変更を同期伝播している場合は、リモート・プロシージャが直接コールされます。変更を非同期で伝播している場合は、そのプロシージャのコールが遅延トランザクション・キューへ追加されます。デフォルトでは、CALL_LOCAL は FALSE、CALL_REMOTE は TRUE です。

次の2つのフェーズを使用してパッケージのレプリケーション・サポートが生成されます。最初のフェーズでは、すべてのサイトでパッケージ仕様部が生成されます。フェーズ2では、すべてのサイトでパッケージ本体が生成されます。これら2つのフェーズは、同期レプリケーションをサポートするために必要です。

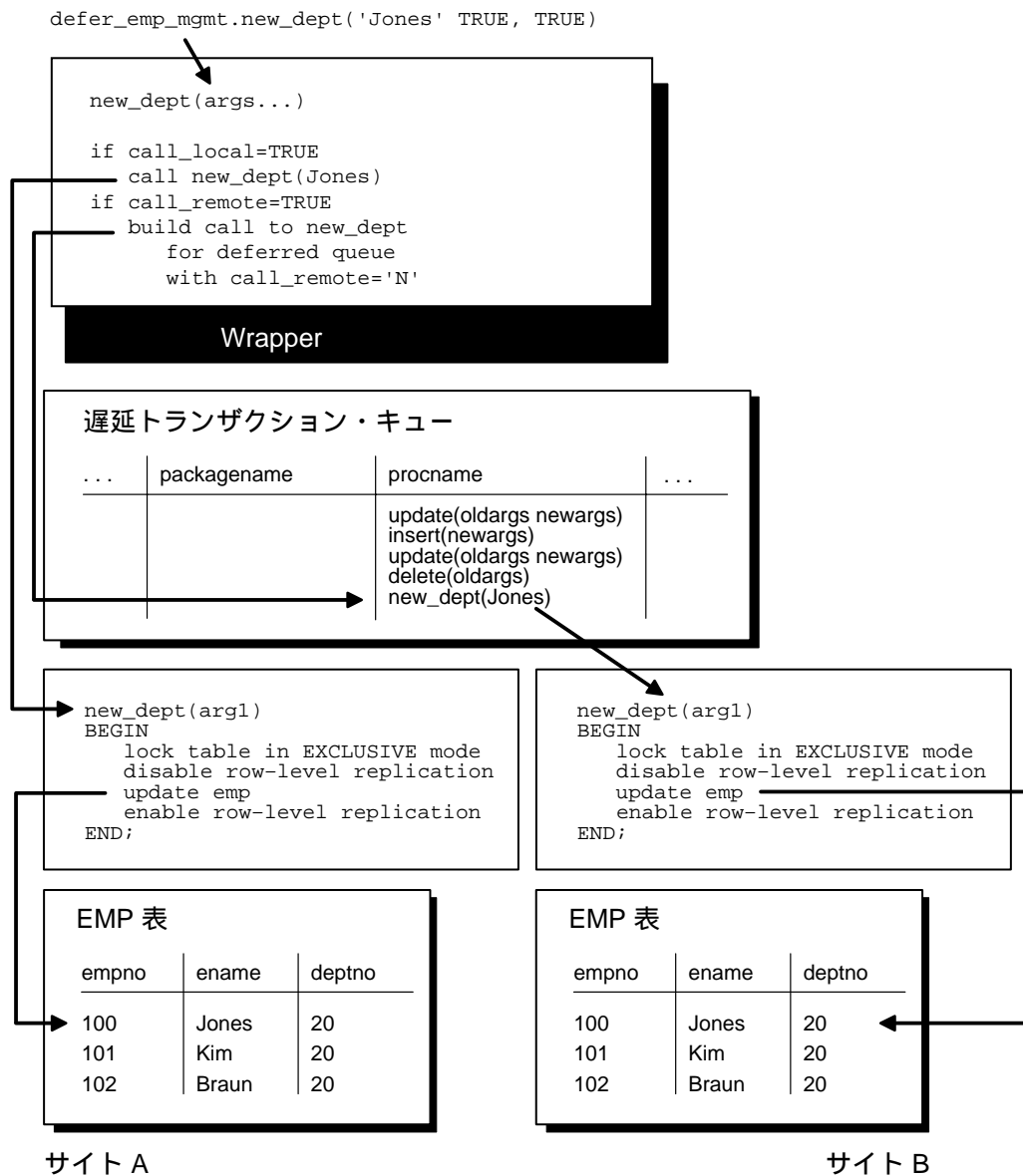
たとえば、引数 ENAME を使用するプロシージャ NEW_DEPT を含むパッケージ EMP_MGMT を生成すると想定します。Replication Manager を使用してこのパッケージをマスター・グループに追加し、オブジェクトのレプリケーション・サポートを生成することによって、このパッケージをシステムのすべてのマスター・サイトにレプリケートすることができます。マスター・グループおよびレプリケーション・オブジェクトの管理に Replication Manager を使用する詳細は、3-13 ページの「マスター・グループの管理」を参照してください。上記の手順が完了したら、次に示すように、アプリケーションでレプリケートされたパッケージのプロシージャをコールすることができます。

```
defer_emp_mgmt.new_dept( ename          => 'Jones',  
                          call_local     => TRUE,  
                          call_remote    => TRUE);
```

図 7-1 に示すように、Wrapper プロシージャのロジックによって、そのプロシージャは確実にローカル・サイトでコールされ、続いてすべてのリモート・サイトでコールされます。また、Wrapper プロシージャのロジックによって、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る場合は確実に CALL_REMOTE が FALSE となり、そのプロシージャがそれ以上伝播されることはありません。

データ所有権の静的パーティション化によって複合レプリケート環境で稼働している場合（つまり、行レベル・レプリケーションを防止していない場合）は、行レベル・レプリケーションとプロシージャ・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ので、レプリケーション機能によってリモート・ノードでの操作の順序が保持されます。

図 7-1 非同期プロシージャ・レプリケーション



同期データ伝播の使用方法

非同期データ伝播は、アドバンスド・レプリケーション環境の標準構成です。ただし、特別な要件があるアプリケーションに対しては同期データ伝播もサポートされています。同期データ伝播は、アプリケーションで表のローカル・レプリカを更新して、同じトランザクション内で同じ表のその他のレプリカもすべて更新するときに発生します。このため、同期データ・レプリケーションは即時データ・レプリケーションとも呼ばれます。同期レプリケーションの使用は、アプリケーションでレプリケート・サイトを継続的に同期化する必要があるときだけにしてください。

注意：レプリケーション・データの即時伝播を使用するレプリケーション・システムは、システムのすべてのサイトが同時に使用可能な場合にだけ機能するため、システムとネットワークの可用性に大きく依存します。

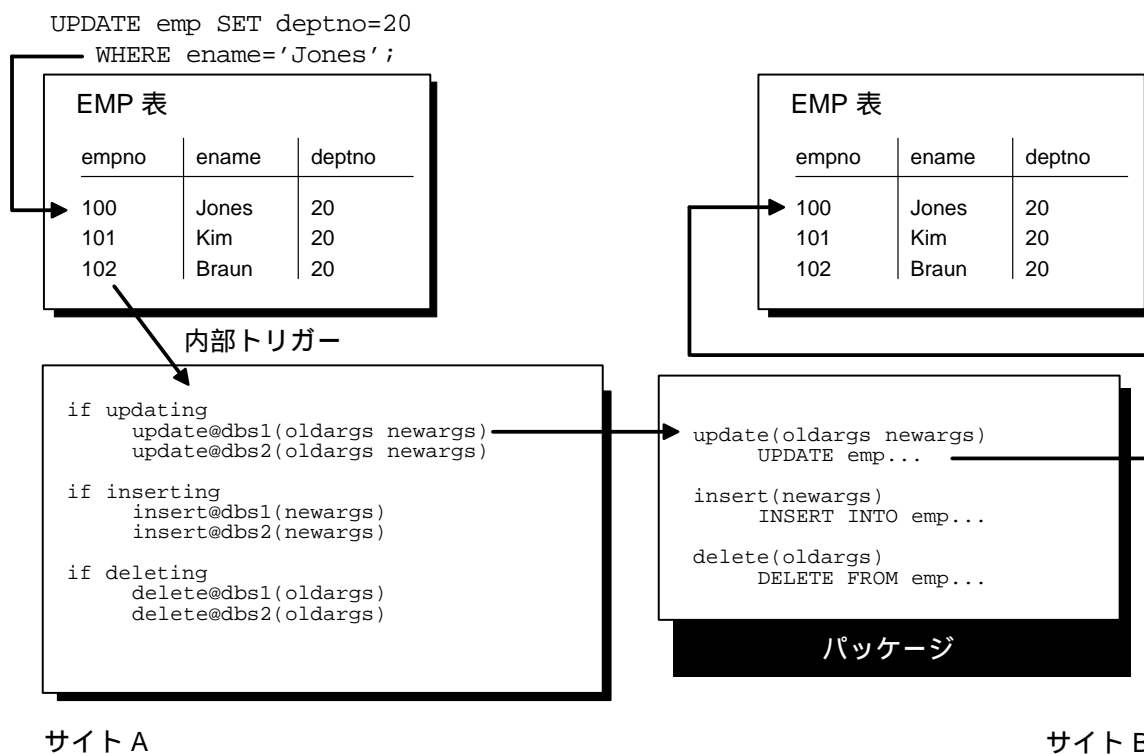
次の項では、同期データ伝播の詳細とそれを使用するレプリケート・データベースの管理方法について説明します。

同期データ伝播について

図 7-2 に示すように、アプリケーションでローカルなレプリケート表に DML 変更を加えるときに、レプリケーション・グループが同期的な行レベル・レプリケーションを使用していても、変更内容はレプリケート環境の他のマスター・サイトに内部トリガーを使用して同時伝播されます。アプリケーションでローカルな変更を適用すると、内部トリガーは、レプリケーション伝播担当者のセキュリティ・コンテキスト内のリモート・マスター・サイトで生成されたプロシージャに対してコールを発行します。Oracle は、障害が発生した場合に、すべての分散トランザクションをコミットするかロールバックします。

追加情報：『Oracle8 Server 分散システム』にある分散更新の説明を参照してください。

図 7-2 同期行レベル・レプリケーションを使った変更の伝播



制限

同期レプリケーションで使われるロック・メカニズムが原因で、デッドロックが発生することがあります。アプリケーションでレプリケート表への同期更新を実行すると、Oracle は、最初にローカル行をロックし、次に AFTER ROW トリガーを使用して対応するリモート行をロックします。各サイトでトランザクションがコミットされると、ロックは解放されます。

同期をとってレプリケートされるトランザクションの接続先

同期レプリケーションのサポートに必要なリモート・プロシージャ・コールは、各オブジェクトの内部トリガーに組み込まれています。レプリケート・オブジェクトのレプリケーション・サポートを生成すると、Oracle は、そのトリガーを全マスター・サイトで起動し、新しいサイトに必要なリモート・プロシージャ・コールを追加します。反対に、マスター・グループからマスター・サイトを削除すると、内部トリガーからコールが削除されます。

競合の検出

マスター・グループのすべてのサイトが互いに同期通信している場合は、アプリケーションでレプリケーションの競合が発生することはありません。しかし、たとえ1つのサイトでも別のサイトに非同期で変更を送信すると、レプリケート環境のどこかのサイトで競合が起こる場合があります。

変更が同期伝播されている場合はエラーになり、ロールバックが必要になります。変更内容が非同期で伝播されている場合は、Oracle は、その競合を自動的に検出してログに書き出すか、または適切な解消方法を指定している場合には、その競合を解消します。

追加情報：第5章「競合の解消」を参照してください。

アドバンスド・レプリケーション環境への新規サイトの追加

新規マスター・サイトまたはそのスナップショット・サイトをアドバンスド・レプリケーション環境のレプリケーション・グループに追加するときは、Replication Managerを使用して新規サイトのデータ伝播モード（方法）を選択できます。

- 新規マスター・サイトのデータ伝播モードによって、そのサイトと他のすべてのマスター・サイトとの間で変更内容を送受信する方法が決定されます。
- 新規スナップショット・サイトのデータ伝播モードによって、そのスナップショット・サイトからマスター・サイトへの変更内容の送信方法が決定されます。

アドバンスド・レプリケーション環境へのマスター・サイトおよびスナップショット・サイトの追加方法の詳細は、第3章「マルチマスター・レプリケーションの使用方法」および第4章「スナップショット・サイト・レプリケーションの使用方法」をそれぞれ参照してください。

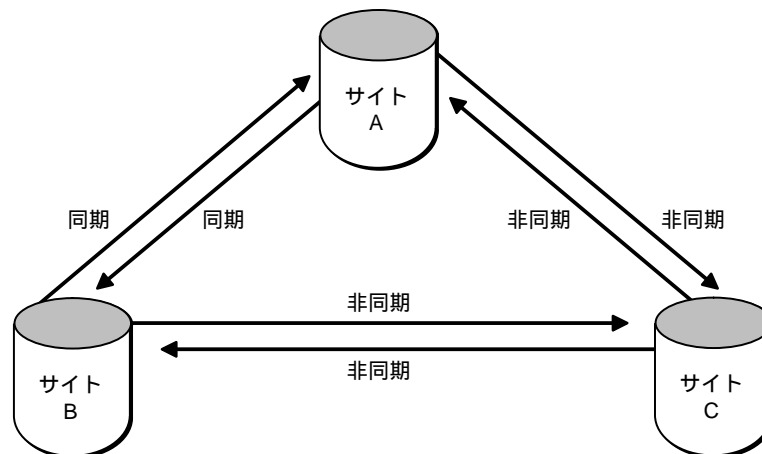
混合モード・マルチマスター・システムについて

状況によっては、一部のマスター・サイトではマスター・グループの変更内容を非同期で伝播し、他のマスター・サイトでは同期伝播するような混合モード環境を設定しなくてはならないことがあります。新規マスター・サイトを異なるデータ伝播モードで1グループに追加するときの順番が重要となる場合があります。

たとえば、3つのマスター・サイト、AおよびB、Cがあるとします。最初に、マスター定義サイトとしてサイトAを作成し、次に、同期伝播モードでサイトBを追加した場合、サイトAはサイトBに変更を同期をとって送信し、サイトBもサイトAに同期をとって変更を送信します。いずれのサイトも遅延トランザクションを作成しないため、いずれかのサイトでリンクをスケジュールする必要はありません。

次に、非同期伝播モードでサイトCを作成するとします。伝播モードは、この場合、図 7-3 に示すようになります。

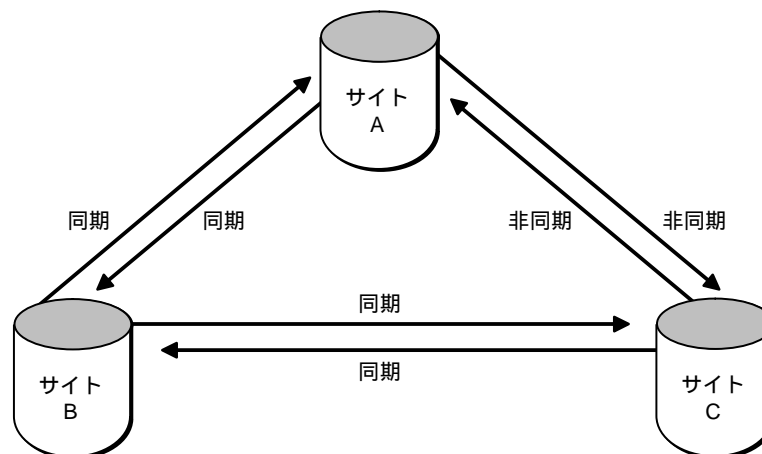
図 7-3 伝播モードの選択



次に、サイト A からサイト C、サイト B からサイト C、サイト C からサイト A と B への遅延トランザクション・キューの伝播をスケジュールしなくてはなりません。

別の例として、サイト A をマスター定義サイトとして作成してから、サイト C を非同期伝播モードで追加した後、サイト B を同期伝播モードで追加したらどうなるかを考えてください。伝播モードはこの場合、図 7-4 に示すようになります。

図 7-4 順序の考慮



新規マスター・サイトを混合モードのマルチマスター・システムに追加するたびに、その追加が既存のサイトとのデータ伝播モードに与える影響を考慮してください。

ヒント : Replication Manager の接続先マップを使用すると、マルチマスター・システムにあるマスター・グループ・サイト間のデータ伝播モードをすばやく参照できます。マスター・グループの接続先マップの詳細は、3-26ページの「マスター・グループの接続先マップの表示」を参照してください。

マスター・サイトのデータ伝播モードの変更

マスター・サイトからマスター・グループ内の別のマスター・サイトにデータ伝播モードを変更するには、Replication Manager でそのグループの接続先マップを使用します。

1. マスター・グループのレプリケーション・アクティビティを中断する。
2. 変更するリンクを右クリックする。
3. 「同期にする」または「非同期にする」をクリックする。
4. 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する。

参照 API: DBMS_REPCAT.ALTER_MASTER_PROPAGATION

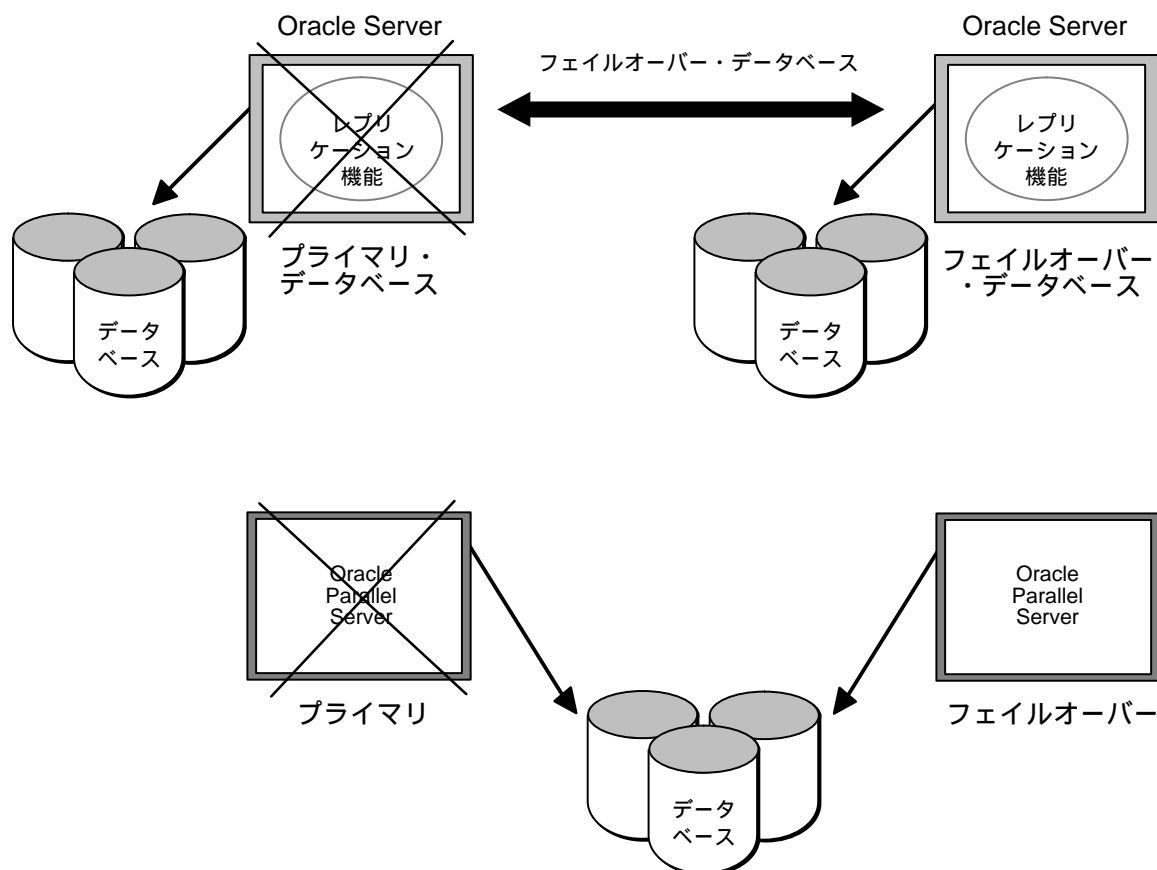
マスター・グループ内の 1 つ以上のマスター・サイト間で伝播モードを切り替えたら、次の指示に従ってください。

- Oracle8 のデータベースにあるマスター・グループ・オブジェクトに対するレプリケーション・サポートの再生成は不要です。
- Oracle7 のデータベースにあるすべてのマスター・グループ・オブジェクトに対するレプリケーション・サポートは再生成する必要があります。
- 同期を非同期に切り替える場合は、影響を受けるサイトの遅延トランザクション・キューの伝播とパージのスケジュールを必ず立ててください。

動作保証の設計

動作の保証によって、システムまたはサイトの障害に際してもアプリケーションの実行を継続することが可能になります。動作保証によって、アプリケーションをフェイルオーバー・システム上で実行でき、障害発生時にプライマリ・システム上で実行されていたときと同じか、またはほとんど同じデータにアクセスできます。図 7-5 に示すように、Oracle Server には Oracle Parallel Server とアドバンスド・レプリケーション機能という、動作保証を実現するための 2 つのテクノロジーが用意されています。

図 7-5 動作保証の方法：アドバンスド・レプリケーションと Parallel Server



Oracle Parallel Server とアドバンスド・レプリケーション

Oracle Parallel Serverは、Oracle Serverのインスタンスをサポートするシステムの障害時に、システムの動作保証をするフェールオーバーをサポートしています。Oracle Parallel Serverにはクラスタまたは超並列の平行・ハードウェアのプラットフォームが必要です。そうしておくことで、クラスタまたは超並列平行・システムが動いているローカル環境での処理システムの障害に対する保護が可能となります。

これらの環境において、Oracle Parallel Serverは動作保証のための理想的な解決策です。インスタンス障害の発生時にトランザクションの消失やデータの不一致なしに、大量のトランザクションをサポートします。インスタンスに障害が生じた場合、Oracle Parallel Serverの正常なインスタンスが自動的に未完了のトランザクションを復元します。障害が生じたシステムで実行されていたアプリケーションはフェイルオーバー・システムで実行され、データベースのすべてのデータにアクセスすることができます。

しかし、Oracle Parallel Serverは、サイト全体におよぶサイト障害（洪水、火事、破壊行為など）によってクラスタまたは大型の平行・システム全体が操作不能になるような状態に対しては動作の保証を行いません。サイト障害に対する動作保証のために、アドバンスド・レプリケーション機能を使用し、地理的に遠隔な場所でデータベースのレプリカを管理できます。

ローカル・システムに障害が生じて、アプリケーションはリモート・サイトで実行を継続できます。ただし、アドバンスド・レプリケーションで、すべてのトランザクションの保護が保証されるわけではありません。また、プライマリ・サイトを復元するときは、データの不一致を防ぐために細心の注意が必要です。

注意：スタンバイ・データベースを構成して、サイト障害から Oracle データベースを保護することもできます。Oracle のスタンバイ・データベース機能の詳細は、『Oracle8 Server バックアップおよびリカバリ』を参照してください。

動作保証の設計

動作保証のためにアドバンスド・レプリケーションを使用する場合は、次の事項を検討してください。

- アドバンスド・レプリケーション機能は、プライマリ・システムと同じ量のトランザクションを保持できなければならない。これはアプリケーションで指定できますが、Oracle Parallel Serverを使用する場合よりも、一般的にスループットがかなり低くなります。
- プライマリ・サイトで障害が生じた場合、プライマリ・サイトで少し前にコミットされたトランザクションは、フェイルオーバー・サイトにまだ非同期に伝播していないことがある。これらのトランザクションは消失したように見える。
- これらの「消失した」トランザクションは、プライマリ・サイトが復元したときに処理を施す必要がある。

たとえば、レプリケーションを使用してリモートのフェイルオーバー受注入力システムを保守する受注入力システムを実行していたときに、プライマリ・システムに障害が生じたとします。

障害発生するとき、少し前にプライマリ・サイトで処理された2つのトランザクションの変更はフェイルオーバー・サイトにはまだ伝播されず適用されていませんでした。最初のトランザクションは新規の受注の入力で、2番目のトランザクションは既存の受注の取消しです。

まず、フェイルオーバー・システムで処理が継続しているときにその受注が存在しないことを誰かが発見して、再入力することが考えられます。次に、受注の取消しが発見さ

れずに、その受注の処理が行われることが考えられます。つまり、取り消された項目が出荷されて、顧客に請求が行われます。

さてここで、プライマリ・サイトが復元したらどうなるでしょう。フェイルオーバー・システムで処理されたすべての変更をプライマリ・システムにただ戻すと、競合が発生します。

具体的には、基本システムに障害が生じる直前に受注された項目が重複受注となります。さらに、プライマリ・システムでは取り消されているはずの受注の出荷および請求トランザクションにより、データが変更されます。

これらの状況に対処するには、システムを注意深く設計する必要があります。次の項でこのプロセスについて説明します。

動作保証が可能なシステムの実現

Oracle のアドバンスド・レプリケーション機能を利用し、複数のレプリケート・マスター・サイトを使用する方法でサイト障害に対する動作保証を提供できます。次の方法の 1 つを使用して、システムを構成する必要があります。この一覧の下の方ほど実現が難しくなります。

- フェイルオーバー・サイトは読取りアクセス専用で使用する。つまり、プライマリ・サイトに障害が生じたときでも、フェイルオーバー・サイトでは更新を行えない。
- 障害の後、プライマリ・サイトはエクスポート/インポートを使用して、または完全バックアップによって、フェイルオーバー・サイトから復元される。
- すべてのデータ/トランザクションに対して完全な競合解消が適用される。このためには、注意深く設計して組み込む必要がある。プライマリ・サイトが復旧したときに生じる可能性のある重複トランザクションなどの競合に対して、適切な解消が確実に行われるようにする必要がある。
- アプリケーション・レベルのルーチンまたはプロシージャ、またはその両方を独自に作成してプライマリ・サイトが復元したとき、アクティブなフェイルオーバー・システムでキューに入れられていたトランザクションをプライマリ・サイトに適用するときに生じる不一致を処理する。

スナップショット・クローニングとオフライン・インスタンスエーション

デフォルトでは、Oracle は次のときにレプリカを作成および移入します。

- 基本レプリケーション環境でスナップショットを作成するとき
- アドバンスド・レプリケーション環境にマスター・サイトまたはスナップショット・サイトを追加するとき

大規模なレプリケート環境の構築中に、システム全体にわたるレプリカの作成と移入に必要なデータ量が過剰な量のネットワーク・トラフィックを生成する可能性があります。大規模なレプリケート環境の構築に必要なデータが原因のネットワークの飽和状態を防止するために、基本とアドバンストの両レプリケーション・システムで新規サイトのオフライン・インスタンスエーションを実行できます。次の項で、基本レプリケーション環境でのスナップショット・クローニングの作成方法とアドバンスト・レプリケーションでマスターとスナップショットのサイトをオフラインでインスタンスエーションする方法を説明します。

基本レプリケーション環境のスナップショット・クローニング

多数のデータベースに同じスナップショット・セットを作成することによるネットワーク・オーバーヘッドを削減するには、次のステップでスナップショットの「クローニング」を実行します。

1. マスター表を含むデータベースとは別のデータベースに、クローニングするスナップショットと対応するリフレッシュ・グループを作成する。マスター表データのスナップショットが作成されたデータベースは、クローニング処理のためのテンプレート・スナップショット・データベースです。基本レプリケーション環境でのスナップショットの作成方法の詳細は、第2章「基本レプリケーションの使用法」を参照してください。
2. テンプレート・スナップショット・データベースで、クローニングするスナップショットとリフレッシュ・グループを含んだスキーマを、別のスナップショット・データベースにエクスポートする。Export ユーティリティは、個々のスナップショットのエクスポートはサポートしません。このため、スナップショットはスキーマ・レベルでエクスポートする必要があります。
3. マスター・データベースのスナップショット用に他のすべてのスナップショット・データベースを準備する。スナップショット・データベースの準備に必要なアカウントとデータベース・リンクの設定方法の詳細は、2-4 ページの「スナップショット用データベースの設定」を参照してください。

注意：スナップショット・データベースが正しく準備されたことを確認するには、データベースの各スナップショット・スキーマに接続します。次に、指定したスナップショットの定義問合せを実行し、エラーなしに実行されていることを確認します。さらに、各新規スナップショット・データベースに、新規スナップショットを保持するために十分な空き領域があることを確認します。

4. 各新規スナップショット・データベースに、テンプレート・スナップショット・データベースからエクスポートしたスナップショット・スキーマをインポートする。これにより、テンプレート・スナップショット・データベースからエクスポートしたスナップショットと同じスナップショットが作成され、データが挿入されます。
5. 高速リフレッシュ可能なすべてのスナップショットの高速リフレッシュを実行する。この処理により、新規スナップショットがマスター・データベースに登録されます。

アドバンスド・レプリケーション・システムでのマスター・サイトのオフライン・インスタンス化

マスター・サイトのオフライン・インスタンス化により、マルチマスター・レプリケーション・システムの既存サイトに要求されるダウン時間を制限しながら、新規マスター・サイトをマスター・グループに追加できます。非常に大規模なデータベースが存在するサイトで、ネットワーク・リンクを介して新しいサイトにデータを転送する時間がかかりすぎる場合は、マスター・サイトのオフライン・インスタンス化が特に役立ちます。

非同期の行レベル・レプリケーションを使用するデフォルトのレプリケーション構成を使用している場合、オフライン・インスタンス化によるスナップショット・サイトの作成に必要なステップは次のとおりです。

1. Replication Managerのセットアップ・ウィザードで新規マスター・サイトを準備する。新規マスター・サイトに、マスター・グループのオブジェクトを格納するスキーマを必ず作成します。マスター・サイトの準備方法の詳細は、3-3ページの「レプリケーション・セットアップ・ウィザード」および3-21ページの「マスター・グループへのマスター・サイトの追加」を参照してください。
2. マスター・グループをレプリケートするマスター・サイトを同期化する。このプロセスには、各マスター・サイトの未処理の管理要求の追求、エラー状態の管理要求の解決、各マスター・サイトのエラー・トランザクションの解決などがあります。
3. 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する。

警告：マスター定義サイトに新規マスター・サイトが表示されるまでは、レプリケーション・アクティビティを再開したり、マスター・グループのその他のレプリケーションの管理を実行したりしないでください。これらの処理を行うと、サイトで行った変更が新規サイトに伝播されず、グループのデータの同期化に関する問題が発生する場合があります。

4. グループのマスター定義サイトから、レプリケーション管理 API プロシージャ DBMS_OFFLINE_OG.BEGIN_INSTANTIATIONをパラメータ gname (マスター・グループ名) と new_site (新規マスター・サイト名) 付きでコールします。このプロシージャにより、新規サイトがマスター・グループに追加されます。
5. マスター・サイトからマスター・グループの個々のオブジェクトをエクスポートする。マスター・グループに2つ以上のスキーマから作成されたレプリケーション・オブジェクトが含まれているときは、必ずそれぞれのスキーマからオブジェクトをエクスポートしてください。
6. マスター定義サイトから、レプリケーション管理 API プロシージャ DBMS_OFFLINE_OG.RESUME_SUBSET_OF_MASTERS をパラメータ gname (マスター・グループ名) と new_site (新規マスター・サイト名) 付きでコールします。

既存のマスター・サイトでは、この時点で管理以外の通常のアクティビティを再開できます。ただし、新規サイトではそのグループのレプリケーション・アクティビティは中断したままです。

7. エクスポート・ファイルを新規マスター・サイトに転送する。

8. 新規マスター・サイトで、レプリケーション管理 API プロシージャ DBMS_OFFLINE_OG.BEGIN_LOAD をパラメータ gname (マスター・グループ名) と new_site (新規マスター・サイト名) 付きでコールします。このプロシージャによって、アプリケーションは、次のステップで行うレプリケーション・データの更新中にマスター・グループ・オブジェクトを更新できなくなり、さらに内部レプリケーション・トリガーが使用できなくなり、他のすべてのマスター・サイトから新規サイトへ遅延 RPC キューを伝播できなくなります。
9. インポート・ユーティリティを使用してエクスポート・ファイルからデータをインポートする。
10. 新規マスター・サイトでインポートが完了したら、レプリケーション管理 API プロシージャ DBMS_OFFLINE_OG.END_LOAD(gname、 new_site) をコールして新規サイトを使用可能にする。
11. マスター定義サイトに戻り、プロシージャ DBMS_OFFLINE_OG.END_INSTANTIATION をパラメータ gname (マスター・グループ名) と new_site (新規マスター・サイト名) 付きでコールします。このプロシージャにより、新規サイトのレプリケーション・アクティビティが再開されます。

追加情報: Import ユーティリティおよび Export ユーティリティの詳細は、『Oracle8 Server ユーティリティ』を参照してください。

次の点に注意してください。

- 複数の新規サイトを追加する場合は、各新規サイトに対して上記のステップを実行する必要があります。いくつかの新規サイトをグループとしてインスタンスエーションすることはできません。
- 上記のプロシージャは適切なサイトからコールするように注意してください。これに失敗すると、予期せぬ結果が生じることがあります。

アドバンスド・レプリケーション・システムでのスナップショット・サイトのオフライン・インスタンスエーション

アドバンスド・レプリケーションでのスナップショットのオフライン・インスタンスエーションは、大規模なスナップショット・サイトを作成する必要があり、ネットワークを介して新規サイトにレプリケーション・データを転送するには時間がかかりすぎる場合に便利です。

オフライン・インスタンスエーションによるスナップショット・サイトの作成に必要なステップは次のとおりです。

マスター・サイト

1. 開始する前に、各マスター・サイトのスナップショット・ログを作成する。
2. 各マスター・グループ表のスナップショットを作成する。各スナップショットをそのマスター表として同じスキーマで作成します。これを行うには、スナップショットの名前をそれと対応するマスター表と異なるものにする必要があります。さらに、スナップショットの定義問合せには、マスター表を参照するためのループバック・データベース・

リンクを使用する必要があります。たとえば、EMP 表のスナップショット EMPLOYEE を同じデータベース DBS1 に作成する場合は、次のようになります。

```
CREATE SNAPSHOT sales.employee AS SELECT * FROM sales.emp@dbs1
```

注意：スナップショットの作成前に、マスター・データベースに十分な記憶領域があることを確認してください。

3. スキーマの所有者として、Export を使用してスナップショットを含むすべてのスキーマをエクスポートする。
4. オフライン・インスタンス用として作成したマスター・サイトのスナップショットを削除する。
5. エクスポート・ファイルを新規スナップショット・サイトに転送する。

新規スナップショット・サイト

1. Replication Manager のセットアップ・ウィザードを使用して、マスター・サイトと通信する新規のスナップショット・サイトを準備する。このプロセスの詳細は、4-4 ページの「スナップショット・サイト・レプリケーションの準備」を参照してください。
2. Replication Manager のスナップショット・グループ・ウィザードを使用して、マスター・グループに対応する空のスナップショット・グループを作成する。マスター・グループのオブジェクトはレプリケートしないでください。
3. それぞれのスキーマおよびスナップショットに対して、プロシージャ DBMS_OFFLINE_SNAPSHOT.BEGIN_LOAD(gname, sname, master_site, snapshot_onsame) を使用して、新規スナップショット・サイトの指定したスキーマおよびオブジェクト・グループ内に空のスナップショットを作成し、必要なサポート・オブジェクトもすべて作成する。
4. エクスポート・ファイルからスナップショットの実表をインポートする。
5. インポートが完了したら、それぞれのスキーマおよびスナップショットに対して、DBMS_OFFLINE_SNAPSHOT.END_LOAD(gname, sname, snapshot_onsame) プロシージャを実行する。

追加情報：『Oracle8 Server ユーティリティ』を参照してください。

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構成するには、Replication Manager 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、一部のケースではこれらの設定を実行するのにレプリケーション管理 API を使用する必要があります。

レプリケーション環境を構成するには、レプリケーション管理者は SYS への接続を含む DBA 権限を持っていなければなりません。

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しなければなりません。

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伝播担当者
- 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、単一のユーザーを管理者および伝播担当者、受信者として設定します。しかし、各機能ごとに別々のユーザーを設定することも可能です。単一のグローバル・レプリケーション管理者を設定するか、またはレプリケーション・グループがスキーマの境界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することも可能です。ただし、各データベースに登録できる伝播担当者は1人だけです。

表 7-1 に、これらの特別アカウントに割り当てなければならない権限を示します。これらのユーザーに必要な権限のほとんどは、レプリケーション管理 API へのコールを通じて付与されます。また、一部の権限は直接付与される必要があります。

表 7-1 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
伝播担当者	DBMS_DEFER_SYS.REGISTER_PROPAGATOR
受信者	EXECUTE ANY PROCEDURE 権限 CREATE SESSION 権限 または、より厳重なセキュリティ要件の場合： CREATE SESSION 権限 DBMS_DEFER_INTERNAL_SYS の EXECUTE 権限 必要な \$RP および \$RL、Wrapper パッケージの EXECUTE 権限

これらのアカウントを作成したら、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・リンクを作成してください。

- 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
- ローカルの伝播担当者からリモートの受信者へ

レプリケーション管理者および伝播担当者、受信者として単一のユーザー・アカウントを指定した場合は、 $N(N-1)$ 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ら、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バージする頻度を定義しなければなりません。各サイトで、各リモート・ロケーションごとに1度ずつ複数回DBMS_DEFER_SYS.SCHEDULE_PUSHを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サンプル・スクリプトを次に示します。

```
/*--- 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 ---*/
grant execute any procedure to prop_rec
grant create session to prop_rec

/*--- 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 ---*/
grant execute any procedure to prop_rec
grant create session to prop_rec

/*--- 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,
    delay_seconds => 0,
    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,  
delay_seconds => 0,  
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
```

スナップショット・レプリケーションのセキュリティ設定

ほとんどの場合、スナップショット・レプリケーションのセキュリティを構成するには、Replication Manager 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、これらの設定を実行するのにレプリケーション管理 API を使用する必要があります。

レプリケーション環境を構成するには、SYS への接続を含む DBA 権限を持っていなければなりません。

最初に、各スナップショット・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・サイトへのリンクを定義しなければなりません。対応付けられたマスター・サイトで追加のユーザーを作成するか、またはユーザーへの追加の権限を作成する必要がある場合があります。

スナップショット・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伝播担当者
- 関連付けられたマスター表からスナップショットに加えられた変更をプルダウンする、リフレッシュ担当者

通常は、上記の各機能を 1 人のユーザーが実行します。しかし、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、スナップショット・サイト管理者がスナップショットを作成し、別のエンド・ユーザーがリフレッシュを実行する場合などです。

表 7-2 に、スナップショット・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 7-2 スナップショット・サイトに必要なユーザー・アカウント

ユーザー	権限
スナップショット・サイト・レプリケーション管理者	DBMS_REPCAT_ADMIN.GRANT_ADMIN_ANY_SCHEMA
伝播担当者	DBMS_DEFER_SYS.REGISTER_PROPAGATOR
リフレッシュ担当者	CREATE ANY SNAPSHOT ALTER ANY SNAPSHOT

スナップショット・サイトに適切なユーザーを作成することに加えて、対応付けられたマスター・サイトにも追加ユーザーを作成する必要があります。表 7-3 に、新規スナップショット・サイトをサポートするマスター・サイト・ユーザーに必要な権限を示します。

表 7-3 マスター・サイトに必要なユーザー・アカウント

ユーザー	権限
プロキシ・スナップショット・サイト管理者	DBMS_REPCAT_ADMIN.GRANT_ADMIN_ANY_SCHEMA または、次の権限： DBMSOBJGWRAPPER および DBMS_REPCAT_UTL2、DBMS_REPCAT_UNTRUSTED への CREATE SESSION および CREATE ANY TRIGGER、CREATE ANY PROCEDURE、SELECT ANY TABLE、EXECUTE さらに、プロキシ管理者にはスナップショット・ログを作成するための次の権限が必要です。 マスター表の COMMENT ANY TABLE および CREATE ANY TABLE、SELECT ANY TABLE または SELECT
受信者	EXECUTE ANY PROCEDURE 権限 CREATE SESSION 権限 または CREATE SESSION 権限 DBMS_DEFER_INTERNAL_SYS の EXECUTE 権限 必要な \$RP および \$RL、Wrapper パッケージの EXECUTE 権限
プロキシ・リフレッシュ担当者	CREATE SESSION 権限 SELECT ANY TABLE 権限 または CREATE SESSION 権限 必要なマスター表およびスナップショット・ログの SELECT 権限

スナップショット・サイトと対応付けられたマスター・サイトでアカウントを作成したら、スナップショット・サイトからマスターへ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・リンクを作成する必要があります。

- スナップショット・レプリケーション管理者からプロキシ・スナップショット・レプリケーション管理者へ
- 伝播担当者から受信者へ
- リフレッシュ担当者からプロキシ・リフレッシュ担当者へ

レプリケーション管理者および伝播担当者、受信者として単一のユーザー・アカウントを指定した場合は、各スナップショット・サイトに1つずつリンクを作成する必要があります。マスター・サイトからスナップショット・サイト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ら、スナップショット・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しなければなりません。また、スナップショット・サイトで DBMS_REFRESH.REFRESH をコールして、対応付けられたマスター・サイトからの変更を受信する頻度を設定しなければなりません。

アドバンスド・レプリケーション・スナップショット・サイトに対する代替セキュリティの設定

簡潔さと安全の両方が向上する代替アプローチを使用して、スナップショット・サイトのセキュリティを構成できます。この構成は、レプリケーション管理 API を使用して設定してください。Replication Manager のセットアップ・ウィザードは、このアプローチをサポートしません。また、大部分のケースで、この構成を使用するスナップショット・サイトの管理に Replication Manager は使用できません。

このアプローチでは、スナップショット・サイトにあるすべてのスナップショットとそれに対応するマスター・サイトのマスター表が1つのスキーマに格納されていなければなりません。このときのスキーマ所有者には、特定のスキーマのプライマリ・スナップショット・サイト管理者とスナップショット・サイトの伝播担当者としての権限が与えられています。同様に、このスキーマ所有者にはマスター・サイトでのスナップショット・レプリケーション受信者としての権限も与えられています。このアプローチは、スナップショット・サイト管理者に他のスキーマのオブジェクトを管理する権限を付与する必要がないため、より簡潔で安全性の高いアプローチです。

order_entry スキーマのマスター・サイト HQ とスナップショット・サイト Sales の間にこのアプローチをインプリメントするためのサンプル・スクリプトを次に示します。

```
/*--- Grant snap_repadmin privileges to schema owner at snapshot site ---*/
connect system/manager@sales.world
execute dbms_repcat_admin.grant_admin_schema(username => 'order_entry')

/*--- Register schema owner as propagator at snapshot site ---*/
```

```
execute dbms_defer_sys.register_propagator(username => 'order_entry')

/*--- Grant proxy snap_repadmin privileges to schema owner at master site ---*/
connect system/manager@hq.world
grant execute on sys.dbmsobjgwrapper to order_entry
grant execute on sys.dbms_defer_internal_sys to order_entry
grant execute on sys.dbms_repcat_untrusted to order_entry
grant execute on sys.dbms_repcat_utl2 to order_entry

/*--- Create link from snapshot schema owner to master schema owner ---*/
connect order_entry/order_entry@sales.world
create database link hq.world
  connect to 'order_entry' identified by 'order_entry' using 'hq.world'
```

削除の競合の防止

削除競合の発生を防止するための最も簡単な方法として、行に削除マークを付けて後で削除する方法があります。この項では、プロシージャ・レプリケーションを使用して、これらマーク付けされた行を削除する簡単なテクニックを概説します。

データベースに次の MAIL_LIST 表が含まれると想定します。

Name	Null?	Type	
-----	-----	-----	-----
CUSTNO	NOT NULL	NUMBER(4)	PRIMARY KEY
CUSTNAME		VARCHAR2(10)	
ADDR1		VARCHAR2(30)	
ADDR2		VARCHAR2(30)	
CITY		VARCHAR2(30)	
STATE		VARCHAR2(2)	
ZIP		NUMBER(9)	
PHONE		NUMBER(10)	
REMOVE_DATE		DATE	

顧客がメール・リストからの除外を要求したときに、削除するかわりに、顧客だったことを示すために、REMOVE_DATE 列が使用されます。現在の顧客には NULL 値を使用します。顧客がメール・リストからの除外を要求した後は、それらの行は更新しないようにします。この規則によって、それらの行が後に削除されたときに競合が生じることを防止できます。現在の顧客のビューは、次のように定義できます。

```
CREATE OR REPLACE VIEW corp.current_mail_list AS
  SELECT custno, custname, addr1, addr2, city, state, zip, phone
```

```
FROM corp.mail_list WHERE remove_date IS NULL;
```

定期的に、たとえば年に一度の祝日セール後に、これらの顧客をREMOVE_DATEフィールドを使用して表から削除できます。次のような文を実行するだけで、行レベル・レプリケーションを使用してこのような削除を実行できます。

```
DELETE corp.mail_list
WHERE remove_date IS NOT NULL AND remove_date<'01-JAN-95';
```

しかし、大量の郵便注文ビジネスを営む大会社の場合、顧客だった人の数が非常に多くなってネットワーク通信量の増大、データベースのオーバーヘッドが発生することがあります。そこで、かわりに次のパッケージを使用して、プロシージャ・レプリケーションを使用できます。

```
CREATE OR REPLACE PACKAGE corp.purge AS
  PROCEDURE remove_cust(purge_date IN DATE);
END;
/
CREATE OR REPLACE PACKAGE BODY corp.purge AS
  PROCEDURE remove_cust(purge_date IN DATE) IS
  BEGIN
    -- turn off row-level replication for set delete
    dbms_reputil.replication_off;
    -- prevent phantom reads
    LOCK TABLE corp.mail_list IN EXCLUSIVE MODE;
    DELETE corp.mail_list 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;
```

DBMS_REPCAT.GENERATE_REPLICATION_SUPPORT プロシージャを使用して、レプリケーション・セットアップの際に DEFER_PURGE パッケージを生成できます。その後、単一のマスター・サイトによって、次のようにプロシージャ・レプリケーションをコールすることができます。

```
BEGIN
  defer_purge.remove_cust('14-APR-97','Y');
END;
```

プロシージャ `PURGE.REMOVE_CUST` をローカルに、そして各マスターで非同期に実行すると、最少のネットワーク通信量で多くの行を削除できます。

パージする行に未処理のトランザクションがないことを確認するには、論理的に削除した行は更新しないようにアプリケーションを作成し、行の論理的削除がその行がパージされる前に伝播されていることを確認できるような古い日付を `REMOVE_DATE` に指定します。こうしておく、前の例では、おそらく表を `EXCLUSIVE` モードにロックする必要はありません。このようにロックしておく方法を使用しても、これらの行が削除中に更新されないようにできます。

動的所有権の競合防止の使用方法

この項では、競合を防ぐアプリケーションを設計するさらに上級の方法を説明します。この方法はトークン・パッシングと呼ばれ、第1章で説明したワークフロー方式と同じものです。この項では、この方法を使用して行全体の所有権を制御する方法を説明しますが、この方法を修正したものを使用すれば、行内の個々の列グループの所有権を制御できます。

ワークフローおよびトークン渡しはどちらも、データの動的所有権の1つです。動的所有権を使用すると、行の更新を行えるのは1つのサイトですが、行の所有権はサイトからサイトに渡すことができます。ワークフローおよびトークン渡しはどちらも、1つ以上の「識別子」列の値を使用して、現在、行を更新できるのは誰かを判別します。

ワークフロー

ワークフロー・パーティション分割を使用すると、データの所有権がサイトからサイトへと送信されると考えることができます。「識別子」列の値を変更することによって行の所有権を別のサイトに送信できるのは、その行の現在の所有者だけです。

受注および出荷、請求のための個別のサイトの簡単な例を取り上げます。ここで、識別子列は受注の状態を示すために使用されます。その状態によって、行を更新できるサイトがどれであるかが判別されます。受注サイトのユーザーは、受注を入力した後でこの行の状態を `SHIP` に更新します。受注サイトのユーザーはこの行をもはや変更できません。所有権は出荷サイトに送信されています。

受注を出荷した後で、出荷サイトのユーザーは、この行の状態を `BILL` に更新するので、所有権は請求サイトに送信されます。以下同様です。

競合を適切に防止するには、動的データ所有権を組み込むアプリケーションは、次の条件に適合することを確認する必要があります。

- 行の所有者だけがその行を更新できる。
- 行が複数のサイトによって所有されることはない。
- 順序競合はすべてのサイトで正常に解消される。

ワークフローをパーティション分割することによって、行の現在の所有者だけが「識別子」列を更新して、その行の所有権を次のサイトに送信できます。あるサイトが所有権を手放すまで、他のサイトは所有権を取得できません。したがって、複数の所有者が存在しないことが確実にになります。

作業のフローが順序付けられているので、順序競合は作業のフロー上で一番後に生じる変更を適用することによって解消されます。すべての順序競合は、ワークフロー・プロセスの各ステップで優先値が増加する、優先順位競合解消方式を使用することによって解消できます。

優先順位競合解消方式は、優先値が常増加する場合、複数のマスターを正常に収束します。

トークン渡し

トークン渡しは、動的所有権の基準を満たす、より一般的な方法です。トークン渡しを組み込むには、「識別子」列のかわりに、レプリケートされた表に所有者およびエポック（移転回数）列が必要です。所有者列には、現在その行を所有していると思われるサイトのグローバル・データベース名が格納されます。

トークン渡し方式を使用すると、ワークフローを含む、データ所有権を動的にパーティション分割する多様な用途に適用できます。

トークン渡しを自動的に実行するようにアプリケーションを設計してください。このアプリケーション管理外で、所有者またはエポック列が更新されることを許容すべきではありません。

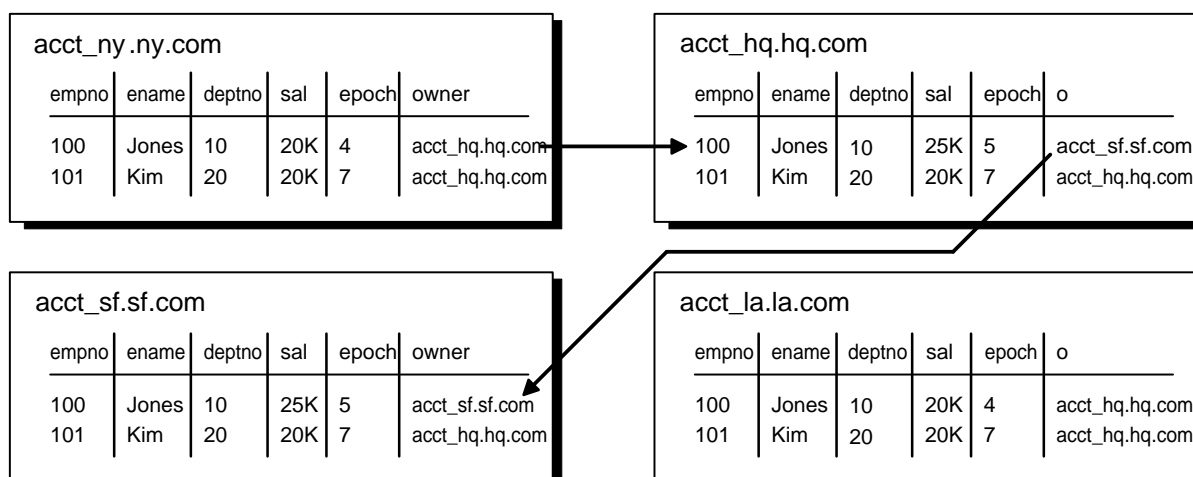
行を更新するときに、アプリケーションは次の処理を必ず実行しなくてはなりません。

1. 行の現在の所有者を判別する。
2. 所有権の変更中に更新されることを防ぐため、行をロックする。
3. 行の所有権を取得する。
4. 更新を実行する（Oracle はトランザクションがコミットされたときにロックを解除します）。

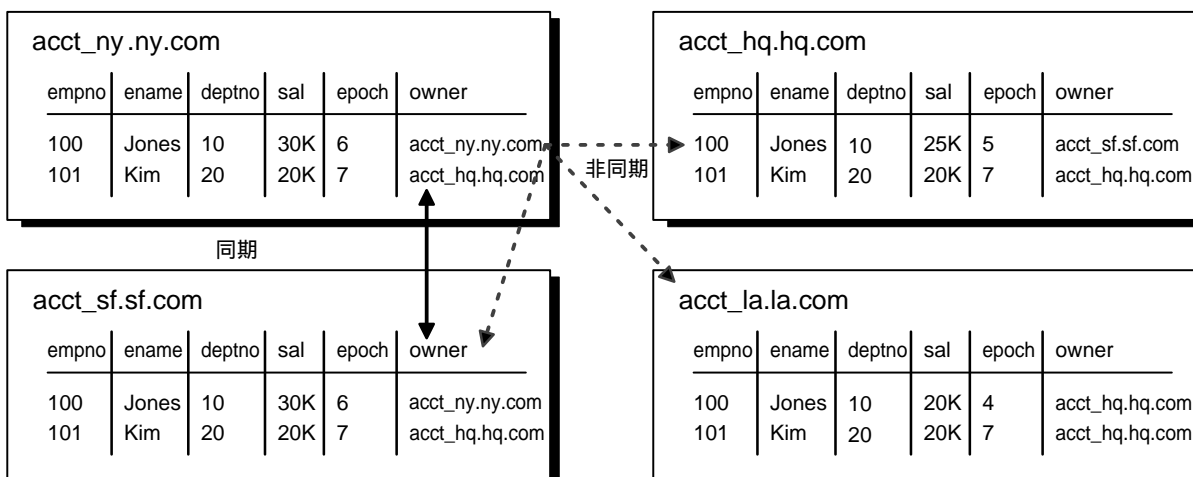
たとえば、図 7-6 では、従業員 100 の所有権が ACCT_SF データベースから ACCT_NY データベースへどのように渡されるかを示します。

図 7-6 トークンの取得

ステップ 1. 真の所有者の判別



ステップ 2. 所有権の取得と変更のプロードキャスト



行の所有者の判別

所有権を取得するために、ACCT_NY データベースは簡単な再帰的アルゴリズムを使用して行の所有者を検出します。このアルゴリズムの疑似コードは次のとおりです。

```
-- Pseudo 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 (dms_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

行の所有者を判別した後、次のステップを完了することによって、ACCT_NY サイトは所有権を ACCT_SF サイトから取得します。

1. 所有権の交換中に変更が生じることを防止するため、SF サイトの行をロックします。
2. SF および NY サイトの両方で、同期に所有者情報を更新します。これによって、常に 1 つのサイトだけが所有者となることが保証されます。SF サイトでの更新は、DBMS_REPUTIL.REPLICATION_OFF を使用してレプリケートしてはいけません。NY サイトでの所有権の変更はステップ 4 でレプリケートされて、最後には（影響はないが SF サイトを含む）他のすべてのサイトに伝播します。

3. 現在の所有者サイト SF からの情報を使用して、新しい所有者サイト NY の行の情報を更新します。このデータは最新のものであることが保証されます。この時、NY サイトでの変更はレプリケートされる必要はありません。SF サイトのこのデータに対するキューに入れられたすべての変更は、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 に伝播するとき、次の項目で説明しているように、エポック番号の値のためにこの変更は無視されます。
4. 新しい所有者サイトのエポック番号を、以前のサイトでの値より 1 つ大きくなるように更新します。この更新を新しい所有者に対してだけ実行し、その後、この更新を他のマスター・サイトに非同期に伝播します。新しい所有者サイトのエポック番号を増やすことによって、順序競合が防止されます。

（上記のステップ 2 で遅延キュー内にあった）SF の変更が最終的に NY サイトに伝播したとき、それらのエポック番号は NY サイトの同じデータのエポック番号よりも低いので、NY サイトはそれらを無視します。

別の例として、HQ サイトが NY の変更を受信した後に SF の変更を受信した場合、NY サイトからの変更により大きなエポック番号があるために、HQ サイトは SF の変更を無視します。

変更の適用

更新時には常に自動的にトークン渡しが行われるように、アプリケーションを設計する必要があります。このアプリケーション管理外で、所有者またはエポック列が更新されることを許容すべきではありません。所有権を変更するときに取得したロックは、実際の更新を適用すると解除されます。変更された情報および更新された所有者とエポック情報は、通常の方法で非同期に他のサイトに伝播します。

変更をレプリケートしないで表を変更する

レプリケート・オブジェクトを変更して、その変更をレプリケート環境内の他のサイトにレプリケートしない場合もあります。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場合があります。

- 変更の伝播にプロシージャ・レプリケーションを使用する場合、プロシージャの最初で常に行レベルのレプリケーションを使用禁止にする必要がある。
- レプリケート・トリガーが複数回動作することを防止するため、レプリケートされた表に定義されているトリガーのレプリケーションを使用禁止にしなければならない場合がある。7-32 ページの「トリガーおよびレプリケーション」を参照してください。
- 手動で競合を解消したときに、この変更を表の他のコピーにレプリケートする必要がない場合がある。

たとえば、あるサイトのレコードの状態を訂正し、エラー・トランザクションの再実行時に競合するレプリケート更新を成功させる必要がある場合、この処理が必要となります。あるいは、トランザクションが接続先サイトで適用されなかったため、レプリケートされていない変更を用いて送信元サイトの変更を戻さなくてはならない場合もあります。

す。この例では、Replication Manager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しないで表を変更するには、DBMS_REPUTIL パッケージの REPLICATION_ON 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必要とせず、生成されたレプリケーション・トリガー中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および使用禁止にするには、DBMS_REPUTIL パッケージに対する EXECUTE 権限が必要です。

アドバンスト・レプリケーションを使用禁止にする

DBMS_REPUTIL.REPLICATION_OFF プロシージャは、現行のセッションの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・サポート・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更の状態をチェックするため、行レベルのレプリケーションを使用するレプリケート表が変更されても、遅延トランザクションがキューに入ることはありません。

注意：レプリケーションのオンまたはオフは、現行のセッションだけに影響を与えます。つまり、同じサーバーに接続している他のユーザーは、コミットされた変更を遅延トランザクション・キューに入れることについて制限は受けません。

プロシージャ・レプリケーションを使用している場合、次の例に示すように、プロシージャの最初で REPLICATION_OFF をコールする必要があります。これにより、アドバンスト・レプリケーション機能は、行レベル・レプリケーションを使用して、加える変更を伝播しなくなります。

```
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;
```

アドバンスト・レプリケーションを再び使用可能にする

競合を解消した後、またはレプリケートされたプロシージャの最後で、確実に DBMS_REPUTIL.REPLICATION_ON をコールして、レプリケートされた表またはスナップショットの変更に対する通常のレプリケーションを再開してください。このプロシージャには引数がありません。 REPLICATION_ON をコールすると、内部レプリケーション変数が TRUE に設定されます。

トリガーおよびレプリケーション

レプリケートされたトリガーをレプリケートされた表に定義した場合、トリガーが起動するのは変更ごとに 1 回だけであることを確認する必要があります。典型的な場合では、最初の変更時にだけトリガーが起動して、その変更がリモート・サイトにレプリケートされたときにはリモートのトリガーが起動しないようにします。

DBMS_REPUTIL.FROM_REMOTE パッケージ変数の値をトリガーの最初にチェックする必要があります。トリガーが表を更新するようにするのは、この変数の値が FALSE であるときだけです。

また、トリガーが起動される原因となった行以外の行を修正するときに、トリガーの最初にレプリケーションを使用禁止にし、トリガーの最後に再び使用可能にできます。この方法を使用すると、元の変更だけがリモート・サイトにレプリケートされます。その後、レプリケート・トリガーがそれぞれのリモート・サイトで起動します。レプリケート・トリガーが実行した更新は、他のどのサイトにも送信されません。

このアプローチを使用すると、競合の解消は起動されません。そのため、トリガーによる変更がデータの一貫性に影響しないことを確認する必要があります。

スナップショットのレプリケーションを使用可能 / 使用禁止にする

現行のサイトでのスナップショットに対するすべてのローカル・レプリケーション・トリガーを使用禁止にするには、次の例に示すように、 SET_I_AM_A_REFRESH をコールして内部リフレッシュ変数を TRUE に設定します。

```
DBMS_SNAPSHOT.SET_I_AM_A_REFRESH(value => TRUE);
```

トリガーを再び使用可能にするには、次に示すように、内部リフレッシュ変数を FALSE に設定します。

```
DBMS_SNAPSHOT.SET_I_AM_A_REFRESH(value => FALSE);
```

内部リフレッシュ変数の値を判別するには、次に示すように、 I_AM_A_REFRESH ファンクションをコールします。

```
ref_stat := DBMS_SNAPSHOT.I_AM_A_REFRESH;replication:advanced  
techniques<$startrange>;advanced replication:techniques<$startrange>
```

遅延トランザクションの使用方法

この章では、次のトピックを説明します。

- 遅延トランザクションに関する情報のリスト
- 遅延トランザクションの作成
- LOB の格納

遅延トランザクションに関する情報のリスト

Oracle は、遅延トランザクションの管理で使用する表やビューをいくつか用意しています。これらのビューによって、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実行中に生じたエラーなど、それぞれの遅延トランザクションに関する情報が得られます。

注意：これらの表は直接修正しないで、DBMS_DEFER および DBMS_DEFER_SYS パッケージのプロシージャを使用して修正してください。

これらのビューについて簡単に説明します。詳細は、第 10 章「データ・ディクショナリ・ビュー」を参照してください。

データ・ディクショナリ・ビュー 説明	
DEFCALL	すべての遅延リモート・プロシージャ・コール (RPC) を記録する。
DEFCALLDEST	遅延リモート・プロシージャ・コールごとの接続先リストを表示する。
DEFDEFAULTDEST	遅延リモート・プロシージャ・コールのデフォルトの接続先リストを表示する。
DEFERROR	適用できなかったトランザクションについて情報を提供する。
DEFLOB	遅延 RPC への LOB パラメータの格納。
DEFSCHEDULE	ジョブの次回実行スケジュールについての情報を表示する。
DEFTRAN	すべての遅延トランザクションを記録する。
DEFTRANDEST	遅延トランザクションの接続先リストを表示する。

遅延トランザクションの作成

正しく形成された遅延トランザクションはすべて、ゼロまたは 1 つの DBMS_DEFER.TRANSACTION コールに続く、ゼロ以上の正しく形成された遅延リモート・プロシージャ・コール、それに続く SQL COMMIT 文で構成されなければなりません。

注意：プロシージャに対する遅延コールを作成する場合、そのプロシージャはパッケージの一部でなければなりません。スタンドアロン・プロシージャに対する遅延コールはサポートされません。

正しく形成された遅延リモート・プロシージャ・コールはすべて、1つの DBMS_DEFER.-CALL コールに続く、ゼロ以上の DBMS_DEFER.datatype_ARG コールで構成されなければなりません。適切な datatype_ARG プロシージャをコールする回数は、CALL プロシージャの ARG_COUNT パラメータの値によって決まります。

DBMS_DEFER.TRANSACTION を使用してトランザクションの開始を指示しない場合、Oracle は、DBMS_DEFER.CALL への最初のコールを新規トランザクションの開始とみなします。

セキュリティ

遅延トランザクションを作成するには、DBMS_DEFER パッケージに対する EXECUTE 権限が必要です。このパッケージは SYS が所有しています。遅延トランザクションはレプリケーション伝播担当者の権限ドメインで実行されるので、DBMS_DEFER パッケージに対する EXECUTE 権限はあまり多くのユーザーに付与しないようにしてください。

提案：これらのプロシージャへのアクセスを制御するには、レプリケーション伝播担当者のスキーマにカバークパッケージを作成して、このカバークパッケージに EXECUTE 権限を付与する必要があります。

接続先の指定

遅延トランザクションのコールを作成する以外に、トランザクションの接続先も指定しなければなりません。アドバンスト・レプリケーション機能により遅延トランザクション・キューに入れられたトランザクションは、DBA_REPPROP ビューにリストされたとおり、レプリケート・オブジェクトのすべての非同期ロケーション (dblink) のキューに入れられます。DBMS_DEFER パッケージのプロシージャを使用して遅延トランザクションをキューに追加するときに、次のいずれかの方法で接続先を指定しなければなりません。これらの方法を、優先順位が高い方から記述します。

1. 次の条件に一致する場合、DBA_REPPROP ビューを使用して遅延トランザクションの接続先が判断されます。
 - DBMS_DEFER.TRANSACTION のコールでの接続先指定に NODES パラメータを使用していない。
 - DBMS_DEFER.CALL のコールでの接続先指定に NODES パラメータを使用していない。
 - 各コールが DBA_REPOBJECT のオブジェクト用に生成されたパッケージのプロシージャに対応する。

注意：この方法は次のどの方法とも組み合わせられません。

2. DBMS_DEFER.CALL プロシージャの NODES パラメータで、省略のない形のデータベース名を1つ（または複数）指定します。この値は、現在の遅延リモート・プロシージャにだけ適用されます。
3. DBMS_DEFER.TRANSACTION プロシージャの NODES パラメータで、省略のない形のデータベース名を1つ（または複数）指定します。この値は、トランザクションを作成するすべての遅延コールに適用されます。

4. 上記の方法の1つを使用して接続先を指定しない場合、10-18ページの「DEFDEFAULTDEST ビュー」の内容を使用してコール先が判断されます。

遅延トランザクションの開始

新しい遅延トランザクションを開始するには、次の例で示されているように、DBMS_DEFER パッケージの TRANSACTION プロシージャをコールします。

```
nodes dbms_defer.node_list_t;  
node(1) := 'acct_hq.hq.com';  
node(2) := 'acct_ny.ny.com';  
DBMS_DEFER.TRANSACTION(nodes);
```

この例は、遅延トランザクションを作成するコールで、DBMS_DEFER.CALLをコールするときに接続先を指定しなければ、すべて ACCT_HQ および ACCT_NY データベースに送るキューに入れます。

TRANSACTION はコールしなくてもかまいません。TRANSACTION をコールしない場合、Oracle は、DBMS_DEFER.CALL への最初のコールを新規トランザクションの開始とみなします。遅延コールの転送先ノードのリストを指定するときに TRANSACTION をコールすると便利であり、そのリストは、遅延トランザクションのどのコールについても同じです。

すべての遅延トランザクションは、DEFTRAN ビューに記録されます。トランザクションの各接続先は DEFTRANDEST ビューに記されます。

追加情報: TRANSACTION プロシージャのパラメータは、9-7 ページ表 9-7、例外は 9-8 ページ表 9-8 を参照してください。

遅延リモート・プロシージャ・コール

遅延リモート・プロシージャ・コールを作成するには、DBMS_DEFER パッケージの CALL プロシージャをコールします。次に例を示します。

```
DBMS_DEFER.CALL(  
    schema_name      => 'accts_rec',  
    package_name      => 'hr',  
    proc_name         => 'hire_emp',  
    arg_count         => 3);
```

この例では、ACCTS_REC スキーマの HR.HIRE_EMP プロシージャへの遅延コールを作成しています。この HIRE_EMP プロシージャには3つの引数が必要です。遅延コールには接続先が指定されていないので、8-3 ページに説明したその他の方法の1つを使用して指定する必要があります。

遅延リモート・プロシージャ・コールは、すべて DEFCALL ビューに記録されます。コールの接続先はそれぞれ DEFCALLDEST ビューに記されます。

追加情報：CALL プロシージャのパラメータは 9-5 ページの表 9-1、例外は 9-5 ページの表 9-2 を参照してください。

遅延コール用パラメータ値をキューに入れる

リモート・プロシージャへのコールを遅延したら、そのプロシージャに渡すデータを用意しなければなりません（IN パラメータだけがサポートされます）。リモート・プロシージャに渡される引数ごとに 1 回のコールが必要であり、また、これらのコールは引数が渡される順序で実行されなければなりません。渡されるデータの型によって、DBMS_DEFER パッケージのどのプロシージャをコールするかが決まります。たとえば、HIRE_EMP プロシージャへのコールを遅延する場合、引数を 3 つ使用すると次のようになります。

```
HIRE_EMP(ename IN VARCHAR2, empno IN NUMBER, salary IN NUMBER)
```

HIRE_EMP への遅延コールを作成したら、次の 3 つのコールによって、必要なデータをプロシージャに渡します。

```
DBMS_DEFER.VARCHAR2_ARG('scott');  
DBMS_DEFER.NUMBER_ARG(12345);  
DBMS_DEFER.NUMBER_ARG(30000);
```

プロシージャに渡すデータ型によっては、プロシージャに対する引数ごとに DBMS_DEFER パッケージ内の次のプロシージャの 1 つをコールしなければなりません。

```
DBMS_DEFER.NUMBER_ARG(arg IN NUMBER);  
DBMS_DEFER.DATE_ARG(arg IN DATE);  
DBMS_DEFER.VARCHAR2_ARG(arg IN VARCHAR2);  
DBMS_DEFER.NVARCHAR2_ARG(arg IN NVARCHAR2);  
DBMS_DEFER.CHAR_ARG(arg IN CHAR);  
DBMS_DEFER.NCHAR_ARG(arg IN NCHAR);  
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);
```

注意：RAW_ARG および CHAR_ARG、NCHAR_ARG、VARCHAR2_ARG、NCHAR_ARG プロシージャに渡すユーザーが長すぎると、ORA-23323 例外が発生する場合があります。

DEFDEFAULTDEST ビューへの接続先の追加

DBMS_DEFER パッケージを使用して遅延トランザクションを作成したときに、遅延トランザクションの接続先またはそのトランザクション内のコールを指定しない場合、Oracle では DEFDEFAULTDEST ビューを使用して、リモート・プロシージャ・コールを遅延させる接続先データベースを判断します。

この表に接続先データベースを追加するには、DBMS_DEFER_SYS パッケージの ADD_DEFAULT_DEST プロシージャをコールしてください。次に例を示します。

```
DBMS_DEFER_SYS.ADD_DEFAULT_DEST( dblink => 'acct_ny.ny.com' );
```

この例では、接続先が指定されていない遅延予定トランザクションは、ACCT_NY データベースのキューに入れられます。

追加情報: ADD_DEFAULT_DEST プロシージャのパラメータは、9-14 ページの表 9-18、例外は 9-14 ページの表 9-19 を参照してください。

DEFDEFAULTDEST ビューからの接続先の取消し

DEFDEFAULTDEST ビューから接続先データベースを削除するには、次の例に示すように、DBMS_DEFER_SYS パッケージ内の DELETE_DEFAULT_DEST プロシージャをコールします。

```
DBMS_DEFER_SYS.DELETE_DEFAULT_DEST( dblink => 'acct_ny.ny.com' );
```

この例では、ユーザーが作成する遅延トランザクションは、デフォルトでは ACCT_NY データベースのキューに入れられなくなります。

遅延トランザクション・キューからトランザクションを削除するには、Replication Manager を使用できます。詳細は、3-11 ページの「サイトの遅延トランザクション・キューのページ」を参照してください。

追加情報: DELETE_DEFAULT_DEST プロシージャのパラメータは、9-14 ページの表 9-20 を参照してください。

遅延トランザクションの実行

作成された遅延トランザクションは、ローカル・サイトの遅延トランザクション・キューに追加されます。このキューが送信されるまで、リモート・プロシージャは実行されません。このキューが一定の間隔で定期的に送信されるようにスケジュール・リンクを作成するか DBMS_DEFER_SYS.SCHEDULE_PUSH をコールしてスケジュール化するか、または Replication Manager が DBMS_DEFER_SYS.PUSH コールを使用して、キューをただちに強制的に送信できます。アドバンスド・レプリケーション機能を利用すると、DML の変更を伝播するのと同じ方法でこれらのトランザクションを伝播できます。

LOB の格納

Oracle では大容量の内部オブジェクト (LOB) として、バイナリ型の LOB (BLOB)、文字型の LOB (CLOB)、各国文字型の LOB (NCLOB) をサポートします。

注意：データ操作言語 (DML) または部分更新の場合、LOB (更新) のサイズが大きくなるほど、伝播時間も増加します。

注意：上記のオブジェクトの遅延 RPC をサポートするには、すべてのサイトが Oracle8 サイトでなければなりません。

注意：セキュリティ上、(遅延) RPC に対する LOB パラメータは、RPC の実行中だけトランザクションで参照可能であることに注意してください。

RPC のための格納の DEFLOB ビュー

Oracle では、遅延 RPC に対する内部 LOB パラメータをサイド表に格納します。この表は別名を使用しなければ参照できません。これによって、記憶領域パラメータおよび格納するスキーマの柔軟性が高くなります。次に、LOB パラメータを格納するデフォルトの表を示します。

```
CREATE TABLE system.def$_lob(
  id RAW(16) CONSTRAINT def$_lob_primary PRIMARY KEY,
  deferred_tran_db VARCHAR2(128), -- origin db
  deferred_tran_id VARCHAR2(22), -- transaction id
  blob_col BLOB,
  clob_col CLOB
  nclob_col NCLOB)
/

-- make deletes fast
CREATE INDEX system.def$_lob_n1 ON system.def$_lob(
  deferred_tran_db,
  deferred_tran_id)
/

-- use a synonym in case underlying table is moved
CREATE SYNONYM sys.def$_lob FOR system.def$_lob
/

CREATE OR REPLACE VIEW DefLOB AS SELECT * FROM sys.def$_lob
/

CREATE PUBLIC SYNONYM DefLOB FOR DefLOB
/
```

レプリケーション管理 API リファレンス

Oracle アドバンスド・レプリケーションのすべてのインストレーションには、レプリケーション管理アプリケーション・プログラム・インタフェース（API）が含まれています。サーバーのレプリケーション管理 API は、アドバンスド・レプリケーション機能の構成に管理者が利用できるプロシージャおよびファンクションがカプセル化された PL/SQL パッケージのセットです。また、Oracle Replication Manager では、作業を実行するために、各サイトのレプリケーション管理 API のプロシージャとファンクションを使用します。この章では、Oracle レプリケーション API を構成するパッケージについて説明します。次の内容が含まれます。

- 各パッケージのプロシージャとファンクション
- 各パッケージ・プロシージャとファンクションのパラメータ
- 各プロシージャとファンクションで呼び出せる例外

パッケージ

Oracle のレプリケーション管理 API には、次のパッケージが含まれています。

- DBMS_DEFER
- DBMS_DEFER_QUERY
- DBMS_DEFER_SYS
- DBMS_OFFLINE_OG
- DBMS_OFFLINE_SNAPSHOT
- DBMS_RECTIFIER_DIFF
- DBMS_REFRESH
- DBMS_REPCAT
- DBMS_REPCAT_ADMIN
- DBMS_REPCAT_AUTH
- DBMS_REPUTIL
- DBMS_SNAPSHOT

Oracle レプリケーション管理 API の使用例

Oracle のレプリケーション管理 API を使用するには、Enterprise Manager の SQL Worksheet または Server Manager のコマンド・プロンプト、SQL*Plus など、非定型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・コールを発行します。たとえば、次の DBMS_REPCAT.CREATE_MASTER_REPOBJECT プロシージャの呼出しでは、ACCT レプリケーション・グループに、新規のレプリケートする表 SALES.EMP を作成します。

```
DBMS_REPCAT.CREATE_MASTER_REPOBJECT(  
    sname           => 'sales',  
    oname           => 'emp',  
    type            => 'table',  
    use_existing_object => TRUE,  
    ddl_text        => 'CREATE TABLE acct_rec.emp AS . . .',  
    comment         => 'created by . . .',  
    retry           => FALSE,  
    copy_rows       => TRUE,  
    gname           => 'acct');
```


レプリケーション管理 API ファンクションをコールするには、そのファンクションの戻り値を受け取る環境を準備する必要があります。たとえば、次の名前なしの PL/SQL ブロックでは、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;
```

前提条件

レプリケーション管理 API のプロシージャとファンクションの多くについて、重要な前提条件があります。次に例を示します。

- 一部のプロシージャまたはファンクションは、マルチマスター構成のマスター定義サイトからのコールだけに適しています。
- マスター・グループに対してある管理操作を実行するには、レプリケーション管理 API のプロシージャおよびファンクションをコールする前に、最初にレプリケーション・アクティビティを中断する必要があります。
- Oracle のレプリケーション管理 API で異なるプロシージャおよびファンクションをコールする場合、その順序は非常に重要です。レプリケーション管理コールを正しく発行する方法の詳細は、次の章を参照してください。

Replication Manager と Oracle レプリケーション管理 API

Oracle の Replication Manager では、ファンクションの大部分を実行するのに、レプリケーション管理 API を使用します。Replication Manager を使用すると、個別にレプリケーション管理 API のコールを発行するよりもかなり便利です。これは、次のようにユーティリティが機能するからです。

- GUI インタフェースを使用して、API コールのパラメータを入力したり調整したりできます。
- 複数の関連する API コールを、適切な順序で自動的に順序付けます。
- API コールから戻された出力を、メッセージ・ボックスとエラー・ファイルで表示します。

Oracle のレプリケーション管理 API の使用方法を習得する簡単な方法は、Replication Manager のスクリプト機能を使用することです。Replication Manager の管理セッションを起動するとき、スクリプト機能をオンにします。終了したら、スクリプト機能をオフにし、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、セッション中にコールされたすべてのレプリケーション管理 API コールが含まれています。スクリプト機能の詳細は、Replication Manager のヘルプ・ドキュメンテーションを参照してください。

DBMS_DEFER パッケージ

DBMS_DEFER パッケージには次のプロシージャが含まれます。

- DBMS_DEFER.CALL
- DBMS_DEFER.COMMIT_WORK
- DBMS_DEFER.datatype_ARG
- DBMS_DEFER.TRANSACTION

各プロシージャについて説明します。

DBMS_DEFER.CALL

目的

リモート・プロシージャに対する遅延コールを発行します。

構文

CALL プロシージャのパラメータは、表 9-1 を参照してください。例外は、表 9-2 を参照してください。このプロシージャの構文は、次のとおりです。

```
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: CALL プロシージャはオーバーロードされます。NODES と GROUP_NAME パラメータは、両方同時に指定できません。

表 9-1 CALL 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ているスキーマの名前。
package_name	ストアド・プロシージャが含まれているパッケージの名前。ストアド・プロシージャはパッケージの一部である必要があります。スタンドアロン・プロシージャに対する遅延コールはサポートされません。
proc_name	コールを中断するリモート・プロシージャの名前。
arg_count	プロシージャのパラメータの数。これらの各パラメータについて、DBMS_DEFER.datatype_ARGに対するコールが必ず1つ必要があります。
nodes	遅延コールを伝播する、省略のない形のデータベース名の PL/SQL 表。表は、位置 1 から始まり、NULL 入力が見つかった場合、または NO_DATA_FOUND 例外状況が発生した場合に終わるように索引が付けてあります。大文字と小文字は区別されません。この引数はオプションです。
group_name	内部的に使用される予約名。

表 9-2 CALL の例外

例外	説明
ORA-23304 (malformedcall)	前のコールは正しく発行されませんでした。
ORA-23319	パラメータ値が不適切です。
ORA-23352	(NODES、または前の DBMS_DEFER.TRANSACTION コールによって指定された) 接続先リストが、重複しています。

DBMS_DEFER.COMMIT_WORK

目的

構成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・コミットを実行します。

構文

COMMIT_WORK プロシージャのパラメータは、表 9-3 を参照してください。例外は、表 9-4 を参照してください。このプロシージャの構文は、次のとおりです。

```
DBMS_DEFER.COMMIT_WORK(commit_work_comment IN VARCHAR2)
```

表 9-3 COMMIT_WORK のパラメータ

パラメータ	説明
commit_work_ comment	DEF\$_CALL 表のトランザクションを説明するのに 50 バイト まで使用できます。

表 9-4 COMMIT_WORK の例外

例外	説明
ORA-23304 (malformedcall)	トランザクションが正しく発行されなかった、または終了さ れませんでした。

DBMS_DEFER.datatype_ARG

目的

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。プロシージャに渡すデータ型によっては、プロシージャに対する引数ごとに DBMS_DEFER パッケージ内の次のプロシージャの 1 つをコールしなくてはなりません。

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.BFILE_ARG	(arg IN BFILE)
DBMS_DEFER.CFILE_ARG	(arg IN CFILE)

表 9-5 datatype_ARG のパラメータ

パラメータ	説明
arg	直前に設定した遅延コールのリモート・プロシージャに渡すパラメータの値。

表 9-6 datatype_ARG の例外

例外	説明
ORA-23323	引数の値が長すぎます。

DBMS_DEFER.TRANSACTION

目的

新しい遅延トランザクションの開始します。このコールを省略すると、Oracle は、DBMS_DEFER.CALL への最初のコールを新しいトランザクションの開始とみなします。

構文

TRANSACTION プロシージャのパラメータは、表 9-7 を参照してください。例外は、表 9-8 を参照してください。このプロシージャの構文は次のとおりです。

```
DBMS_DEFER.TRANSACTION
```

```
DBMS_DEFER.TRANSACTION( nodes    IN    node_list_t)
```

注意：TRANSACTION プロシージャはオーバーロードされます。このプロシージャは、入力パラメータが指定されていてもいなくても同様に動作しますが、入力パラメータが指定されていない場合は、NODESパラメータのノードを使用するかわりにDEFDEFAULTDESTビュー内のノードを使用します。

表 9-7 TRANSACTION 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、省略のない形のデータベース名の PL/SQL 表。表は、位置 1 から始まり、NULL 入力が見つかった場合、または NO_DATA_FOUND 例外状況が発生した場合に終わるように索引が付けてあります。大文字と小文字は区別されません。

表 9-8 TRANSACTION の例外

例外	説明
ORA-23304 (malformedcall)	前のトランザクションが正しく形成されなかった、または終了されませんでした。
ORA-23319	パラメータ値が不適切です。
ORA-23352	ノード・リストに重複名が含まれている場合に、DBMS_DEFER.CALL によって発生します。

DBMS_DEFER_QUERY パッケージ

DBMS_DEFER_QUERY パッケージには、次のプロシージャおよびファンクションが含まれます。

- DBMS_DEFER_QUERY.GET_ARG_FORM
- DBMS_DEFER_QUERY.GET_ARG_TYPE
- DBMS_DEFER_QUERY.GET_CALL_ARGS
- DBMS_DEFER_QUERY.GET_datatype_ARG

各プロシージャおよびファンクションについて説明します。

DBMS_DEFER_QUERY.GET_ARG_FORM

目的

遅延コールの引数フォームを返します。遅延トランザクションの表示方法の詳細は、6-8ページの「遅延トランザクションの表示」を参照してください。エラー・トランザクションの表示方法の詳細は、6-10ページの「エラー・トランザクションの表示」を参照してください。

構文

GET_ARG_FORM ファンクションのパラメータは、表 9-9 を参照してください。例外は、表 9-10 を参照してください。このプロシージャの構文は、次のとおりです。

```
DBMS_DEFER_QUERY.GET_ARG_FORM(  
    callno           IN NUMBER,  
    arg_no           IN NUMBER,  
    deferred_tran_id IN VARCHAR2)  
RETURN NUMBER
```

表 9-9 GET_ARG_FORM 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。
arg_no	目的のパラメータのコール引数リスト内の位置。パラメータ位置は、コール内の 1 から開始するパラメータ数です。
deferred_tran_id	遅延トランザクション ID。

表 9-10 GET_ARG_FORM 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ていません。

DBMS_DEFER_QUERY.GET_ARG_TYPE

目的

遅延コールの引数型を返します。遅延トランザクションの表示方法の詳細は、6-8 ページの「遅延トランザクションの表示」を参照してください。エラー・トランザクションの表示方法の詳細は、6-10 ページの「エラー・トランザクションの表示」を参照してください。

構文

GET_ARG_TYPE ファンクションのパラメータは、表 9-11 を参照してください。例外については表 9-12 を、戻り値については表 9-13 を参照してください。このプロシージャの構文は、次のとおりです。

```
DBMS_DEFER_QUERY.GET_ARG_TYPE(  
    callno          IN    NUMBER,  
    arg_no          IN    NUMBER,  
    deferred_tran_id IN    VARCHAR2)  
RETURN NUMBER
```

表 9-11 GET_ARG_TYPE 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 ビューの ID 番号。
arg_no	型を返すコールの引数の位置 (数値)。プロシージャの最初の引数は、位置 1 にあります。
deferred_tran_id	遅延トランザクションの識別子。

表 9-12 GET_ARG_TYPE 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ていません。

表 9-13 GET_ARG_TYPE の戻り値

戻り値	対応するデータ型
1	VARCHAR2
2	NUMBER
11	ROWID
12	DATE
23	RAW
96	CHAR

DBMS_DEFER_QUERY.GET_CALL_ARGS

目的

このプロシージャは、指定したコールのさまざまな引数をテキストで戻します。

構文

GET_CALL_ARGS プロシージャのパラメータは、表 9-14 を参照してください。例外は、表 9-15 を参照してください。このプロシージャの構文は次のとおりです。

```
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)

```

表 9-14 GET_CALL_ARGS のパラメータ

パラメータ	説明
callno	遅延 RPC の DEFCALL の ID 番号。
startarg	記述したい最初の引数の位置（数値）。
argcnt	コール内の引数の数。
argsize	戻される引数の最大サイズ。
tran_id	遅延トランザクションの識別子。
date_fmt	日付が戻される書式。
types	引数の型を含む配列。
forms	引数のキャラクタ・セット・フォームを含む配列。
vals	テキスト形式で引数の値を含む配列。

表 9-15 GET_CALL_ARGS 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ていません。

DBMS_DEFER_QUERY.GET_datatype_ARG

目的

遅延コールの引数値を返します。遅延トランザクションの表示方法の詳細は、6-8 ページの「遅延トランザクションの表示」を参照してください。エラー・トランザクションの表示方法の詳細は、6-10 ページの「エラー・トランザクションの表示」を参照してください。

構文

取り出す引数値の型に応じて、該当するファンクションの構文は次のようになります。これらのファンクションのパラメータは、表 9-16 を参照してください。例外は、表 9-17 を参照してください。これらの各々のファンクションは、指定された引数の値を戻します。

```
DBMS_DEFER_QUERY.GET_datatype_ARG (  
    callno          IN    NUMBER,  
    arg_no          IN    NUMBER,  
    deferred_tran_id IN    VARCHAR2 DEFAULT NULL)  
RETURN datatype
```

datatype には、次の引数を指定します。

```
{ NUMBER  
| VARCHAR2  
| CHAR  
| DATE  
| RAW  
| ROWID  
| BLOB  
| CLOB  
| NCLOB  
| NCHAR  
| NVARCHAR2 }
```

表 9-16 GET_datatype_ARG 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 ビューの ID 番号。
arg_no	値を返すコールの引数の位置（数値）。プロシージャの最初の引数は、位置 1 にあります。
deferred_tran_id	デフォルトは NULL。遅延トランザクションの識別子。GET_ARG_TYPE に渡された最後のトランザクション識別子がデフォルトになります。

表 9-17 GET_datatype_ARG 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ていません。
ORA-26564	この位置の引数は、指定された型のも的一致しません。

DBMS_DEFER_SYS パッケージ

DBMS_DEFER_SYS パッケージには、次のプロシージャおよびファンクションが含まれます。

- DBMS_DEFER_SYS.ADD_DEFAULT_DEST
- DBMS_DEFER_SYS.DELETE_DEFAULT_DEST
- DBMS_DEFER_SYS.DELETE_DEF_DESTINATION
- DBMS_DEFER_SYS.DELETE_ERROR
- DBMS_DEFER_SYS.DELETE_TRAN
- DBMS_DEFER_SYS.DISABLED
- DBMS_DEFER_SYS.EXCLUDE_PUSH
- DBMS_DEFER_SYS.EXECUTE_ERROR
- DBMS_DEFER_SYS.EXECUTE_ERROR_AS_USER
- DBMS_DEFER_SYS.PURGE
- DBMS_DEFER_SYS.PUSH
- DBMS_DEFER_SYS.REGISTER_PROPAGATOR
- DBMS_DEFER_SYS.SCHEDULE_PURGE
- DBMS_DEFER_SYS.SCHEDULE_PUSH
- DBMS_DEFER_SYS.SET_DISABLED
- DBMS_DEFER_SYS.UNREGISTER_PROPAGATOR
- DBMS_DEFER_SYS.UNSCHEDULE_PURGE
- DBMS_DEFER_SYS.UNSCHEDULE_PUSH

次のページからは、各プロシージャおよびファンクションについて説明します。

DBMS_DEFER_SYS.ADD_DEFAULT_DEST

目的

DEFDEFAULTDEST ビューに接続先データベースを追加します。

構文

ADD_DEFAULT_DEST プロシージャのパラメータは、表 9-18 を参照してください。例外は、表 9-19 を参照してください。このプロシージャの構文は、次のとおりです。

```
DBMS_DEFER_SYS.ADD_DEFAULT_DEST( dblink    IN    VARCHAR2)
```

表 9-18 ADD_DEFAULT_DEST 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、省略のない形のデータベース名。

表 9-19 ADD_DEFAULT_DEST の例外

例外	説明
ORA-23352	指定された DBLINK は、すでにデフォルトのリスト中で指定されています。

DBMS_DEFER_SYS.DELETE_DEFAULT_DEST

目的

DEFDEFAULTDEST ビューから接続先データベースを削除します。

構文

DELETE_DEFAULT_DEST プロシージャのパラメータは、表 9-20 を参照してください。このプロシージャの構文は、次のとおりです。

```
DBMS_DEFER_SYS.DELETE_DEFAULT_DEST(dblink    IN    VARCHAR2)
```

表 9-20 DELETE_DEFAULT_DEST 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、省略のない形のデータベース名。このdblinkがビューの中に見つからなければ、無視されます。

DBMS_DEFER_SYS.DELETE_DEF_DESTINATION

目的

DEFSCHEDULE ビューから接続先データベースを削除します。

構文

DELETE_DEF_DESTINATION プロシージャのパラメータは、表 9-21 を参照してください。このプロシージャの構文は、次のとおりです。

```
DBMS_DEFER_SYS.DELETE_DEF_DESTINATION(  
    destination    IN    VARCHAR2,  
    force          IN    BOOLEAN := FALSE)
```

表 9-21 DELETE_DEF_DESTINATION 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、省略のない形のデータベース名。Oracleでこの接続先がビューの中に見つからなければ無視されます。
force	TRUE に設定すると、Oracle はすべての安全チェックを無視し、接続先を削除します。

DBMS_DEFER_SYS.DELETE_ERROR

目的

DEFERROR ビューからトランザクションを削除します。

構文

DELETE_ERROR プロシージャのパラメータは、表 9-22 を参照してください。このプロシージャの構文は、次のとおりです。

```
DBMS_DEFER_SYS.DELETE_ERROR(  
    deferred_tran_id    IN    VARCHAR2,  
    destination         IN    VARCHAR2)
```

表 9-22 DELETE_ERROR 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 ビューの ID 番号。このパラメータが NULL であれば、他のパラメータの要件に適応しているすべてのトランザクションが削除されます。

表 9-22 DELETE_ERROR のパラメータ

パラメータ	説明
destination	トランザクションが最初にキューされたデータベースの、DEFERROR ビューの 省略のない形のデータベース名。このパラメータがNULLであれば、他のパラメータの要件に適用してい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。

DBMS_DEFER_SYS.DELETE_TRAN

目的

DEFTRANDEST ビューからトランザクションを削除します。そのトランザクションに対する DEFTRANDEST または DEFERROR の項目が他になければ、そのトランザクションは DEFT-RAN ビューおよび DEFCALL ビューの両方からも削除されます。

構文

DELETE_TRAN プロシージャのパラメータは、表 9-23 を参照してください。このプロシージャの構文は、次のとおりです。

```
DBMS_DEFER_SYS.DELETE_TRAN(
    deferred_tran_id    IN    VARCHAR2,
    destination         IN    VARCHAR2)
```

表 9-23 DELETE_TRAN 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 ビューの ID 番号。このパラメータがNULLであれば、他のパラメータの要件に適用してい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最初にキューされたデータベースの、DEFTRANDEST ビューの省略のない形のデータベース名。このパラメータがNULLであれば、他のパラメータの要件に適用しているすべてのトランザクションが削除されます。

DBMS_DEFER_SYS.DISABLED

目的

現行のサイトから指定した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かどうかを判断します。指定した接続先に対して遅延リモート・プロシージャ・コール (RPC) キューが使用禁止になっている場合は、TRUE が返ります。

構文

DISABLED ファンクションのパラメータは、表 9-24 を、戻り値については表 9-25 を、例外については表 9-26 を参照してください。このファンクションの構文は、次のとおりです。

```
DBMS_DEFER_SYS.DISABLED(  
    destination IN VARCHAR2)  
RETURN BOOLEAN
```

表 9-24 DISABLED 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したいノードの、省略のない形のデータベース名。

表 9-25 DISABLED の戻り値

値	説明
TRUE	現行のサイトから指定したサイトへの伝播が使用禁止です。
FALSE	現行のサイトから指定したサイトへの伝播が使用可能です。

表 9-26 DISABLED の例外

例外	説明
NO_DATA_FOUND	DESTINATION が DEFSCHEDULE ビューに表示されません。

DBMS_DEFER_SYS.EXCLUDE_PUSH

目的

遅延トランザクション PUSH（シリアルまたはパラレル）を防ぐ排他ロックを獲得します。このファンクションはコミットを実行します。ロックは、RELEASE_ON_COMMIT => TRUE の場合に獲得され、遅延トランザクション・キューの送信は、次のコミット後に再開されます。

構文

パラメータおよび戻り値を以下に示します。

```
DBMS_DEFER_SYS.EXCLUDE_PUSH(  
    timeout IN INTEGER)
```

RETURN INTEGER

表 9-27 EXCLUDE_PUSH のパラメータ

パラメータ	説明
timeout	タイムアウトの秒数。この時間内に（エラーのために、または PUSH が現在進行中であるために）ロックが獲得されない場合、コールは値 1 を戻します。タイムアウト値 DBMS_LOCK.MAX-WAIT では無期限に待機します。

表 9-28 EXCLUDE_PUSH の戻り値

値	説明
0	成功、ロックが獲得されました。
1	タイムアウト、ロックは獲得されていません。
2	デッドロック、ロックは獲得されていません。
4	すでにロックがあります。

DBMS_DEFER_SYS.EXECUTE_ERROR

目的

正常に完了しなかった遅延トランザクションを再実行します。このプロシージャでは、NULL パラメータと NULL 以外のパラメータを不適切に組み合わせて使用すると、ORA-24275 エラーが発生します。

構文

EXECUTE_ERROR プロシージャのパラメータは、表 9-29 を参照してください。このプロシージャの構文は、次のとおりです。

```
DBMS_DEFER_SYS.EXECUTE_ERROR(  
    deferred_tran_id IN  VARCHAR2,  
    destination      IN  VARCHAR2)
```


表 9-29 EXECUTE_ERROR 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 ビューの ID 番号。このパラメータが NULL の場合、DESTINATION のキューに入っ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最初にキューされたデータベースの、DEFERROR ビューの省略のない形のデータベース名。このパラメータは NULL にできません。

表 9-30 EXECUTE_ERROR の例外

例外	説明
badparam	パラメータ値が入力されていないか、無効です（たとえば、接続先が NULL の場合）。
missinguser	無効なユーザー。

DBMS_DEFER_SYS.EXECUTE_ERROR_AS_USER

目的

正常に完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。このプロシージャでは、NULL パラメータと NULL 以外のパラメータを不適切に組み合わせて使用すると、ORA-24275 エラーが発生します。

構文

EXECUTE_ERROR_AS_USER プロシージャのパラメータは、表 9-31 を参照してください。このプロシージャの構文は、次のとおりです。

```
DBMS_DEFER_SYS.EXECUTE_ERROR_AS_USER(
    deferred_tran_id IN    VARCHAR2,
    destination      IN    VARCHAR2)
```

表 9-31 EXECUTE_ERROR_AS_USER 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 ビューの ID 番号。このパラメータが NULL の場合、DEFERRED_TRAN_DB でコールされた、DESTINATION のキューに入っ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最初にキューされたデータベースの、DEFERROR ビューの省略のない形のデータベース名。このパラメータは NULL にできません。

表 9-32 EXECUTE_ERROR_AS_USER の例外

例外	説明
badparam	パラメータ値が入力されていないか、無効です（たとえば、接続先が NULL の場合）。
missinguser	無効なユーザー。

DBMS_DEFER_SYS.PURGE

目的

現行のマスター・サイトまたはスナップショット・サイトの遅延トランザクション・キューから、送信済みトランザクションを削除します。

構文

PURGE ファンクションのパラメータは、表 9-33 を参照してください。このファンクションの構文は、次のとおりです。

```
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
```

表 9-33 PURGE 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・キューのページ方法を制御します。 purge_method_quick を指定するとコストが削減でき、 purge_method_precise を指定すると精度が高くなります。
rollback_segment	ページするために使用するロールバック・セグメント名。デ フォルトは NULL。
startup_seconds	同じ遅延トランザクション・キューの直前のページを待つ最 大秒数。
execution_seconds	>0 の場合、指定の秒数後に実行を正常に停止します。
delay_seconds	遅延トランザクション・キューが delay_seconds の間空になっ た後、実行を正常に停止します。
transaction_count	>0 の場合、 transaction_count のトランザクション数をページ 後、正常にシャットダウンします。
write_trace	TRUE に設定すると、Oracle では、サーバーのトレース・ファ イルのファンクションから戻された結果値を記録します。

表 9-34 PURGE の戻り値

値	説明
0	OK。 delay_seconds が経過後、終了しました。
1	起動中にロック・タイムアウトにより終了しました。
2	execution_seconds が経過したので終了しました。
3	transaction_count が経過したので終了しました。
4	delivery_order_limit が経過したので終了しました。
5	エラー発生後に終了しました。

表 9-35 PURGE の例外

例外	説明
argoutofrange	パラメータ値が有効範囲外です。
executiondisabled	遅延 RPC の実行が接続先で使用禁止です。

表 9-35 PURGE の例外

例外	説明
defererror	内部エラー。

DBMS_DEFER_SYS.PUSH

目的

現在のマスター・サイトまたはスナップショット・サイトの遅延リモート・プロシージャ・コール・キューを、シリアル伝播またはパラレル伝播を使用して別のマスター・サイトに強制的に送信（実行、伝播）します。

構文

PUSH ファンクションのパラメータは、表 9-36 を参照してください。このファンクションの構文は、次のとおりです。

```
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
```

表 9-36 PUSH 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の省略のない形のデータベース名。
parallelism	0 = シリアル伝播、n > 0 = n個のパラレル・サーバー・プロセスを使用したパラレル伝播、1 = 1つのパラレル・サーバー・プロセスだけを使用したパラレル伝播。
heap_size	パラレル伝播スケジューリングのために、同時に検査されるトランザクションの最大数。Oracleが自動的に最適なパフォーマンスのためのデフォルト設定を計算します。オラクル・サポートで指示されない限り、このパラメータを設定しないでください。

表 9-36 PUSH のパラメータ

パラメータ	説明
stop_on_error	デフォルトの FALSE では、競合などのエラーが発生しても、executor が継続します。TRUE では、トランザクションが接続先サイトで最初にエラーを検出したときに（可能であれば正常に）停止します。
write_trace	TRUE に設定すると、Oracle では、サーバーのトレース・ファイルのファンクションから戻された結果値を記録します。
startup_seconds	同じ接続先の直前の送信を待つ最大秒数。
execution_seconds	>0 の場合、指定の秒数後に実行を正常に停止します。transaction_count および execution_seconds がゼロ（デフォルト）の場合、キューに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。
delay_seconds	キューが空でも、指定の秒数が経過するまでは戻りません。タイトなループから DBMS_DEFER_SYS.PUSH がコールされた場合に、実行オーバーヘッドを削減するのに有効です。
transaction_count	> 0 の場合、停止までに送られるトランザクションの最大数。transaction_count および execution_seconds がゼロ（デフォルト）の場合、キューに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。
delivery_order_limit	delivery_order >= delivery_order_limit の場合、トランザクションを送信する前に実行を正常に停止します。

表 9-37 PUSH の戻り値

値	説明
0	OK。delay_seconds が経過後、終了しました。
1	起動中にロック・タイムアウトにより終了しました。
2	execution_seconds が経過したので終了しました。
3	transaction_count が経過したので終了しました。
4	delivery_order_limit が経過したので終了しました。
5	エラー発生後に終了しました。PUSH

表 9-38 PUSH の例外

例外	説明
deferror incompleteparallepush	内部エラー。
executiondisabled	遅延 RPC の実行が接続先で使用禁止です。
cat_err_err	DEFERROR で項目を作成中にエラーが発生しました。
deferred_rpc_qiesce	オブジェクト・グループのレプリケーション・アクティビティが中断しました。
commfailure	遅延 RPC 中に通信に失敗しました。
missingpropagator	伝播担当者が存在しません。

DBMS_DEFER_SYS.REGISTER_PROPAGATOR

目的

指定のユーザーをローカル・データベースの伝播担当者として登録します。また、指定のユーザーに、CREATE SESSIONおよびCREATE PROCEDURE、CREATE DATABASE LINK、EXECUTE ANY PROCEDURE権限を付与します（これにより、そのユーザーはWrapperを作成できます）。

構文

REGISTER_PROPAGATOR プロシージャのパラメータは、表 9-39 を参照してください。このプロシージャの構文は、次のとおりです。また、例外については、表 9-40 を参照してください。

```
DBMS_DEFER_SYS.REGISTER_PROPAGATOR(username IN VARCHAR2)
```

表 9-39 REGISTER_PROPAGATOR のパラメータ

パラメータ	説明
username	ユーザーの名前。

例外

表 9-40 REGISTER_PROPAGATOR の例外

例外	説明
missinguser	指定のユーザーが存在しません。
alreadypropagator	指定のユーザーはすでに伝播担当者です。
duplicatepropagator	すでに別の伝播担当者が存在します。

DBMS_DEFER_SYS.SCHEDULE_PURGE

目的

現行のマスター・サイトまたはスナップショット・サイトの遅延トランザクション・キューから、送信済みトランザクションを削除するジョブをスケジュールします。サイトごとに、1つのページだけスケジュールできます。

構文

SCHEDULE_PURGE プロシージャのパラメータは、表 9-41 を参照してください。このプロシージャの構文は、次のとおりです。

```
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)
```

表 9-41 SCHEDULE_PURGE のパラメータ

パラメータ	説明
interval	パージする次の時間を計算するファンクションを提供できます。この値は、DEFSCHEDULE ビューの INTERVAL フィールドに格納され、このビューの NEXT_DATE フィールドを計算します。このパラメータのデフォルト値である NULL を使用すると、このフィールドの値は変更されません。フィールドに直前の値がなかった場合は、フィールドが値 NULL で再作成されます。このフィールドに値を指定しない場合、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送信済みのトランザクションをパージする時間を指定できます。この値は、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である NULL を使用すると、このフィールドの値は変更されません。このフィールドに直前の値がなかった場合は、フィールドが値 NULL で再作成されます。このフィールドに値を指定しない場合、INTERVAL に対して値を指定する必要があります。
reset	LAST_TXN_COUNT および LAST_ERROR、LAST_MSG を NULL にリセットするには、TRUE に設定します。
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 を指定するとコストが削減でき、purge_method_precise を指定すると精度が高くなります。
rollback_segment	パージするために使用するロールバック・セグメント名。デフォルトは NULL。
startup_seconds	同じ遅延トランザクション・キューの直前のパージを待つ最大秒数。
execution_seconds	>0 の場合、指定の秒数後に実行を正常に停止します。
delay_seconds	遅延トランザクション・キューが delay_seconds の間空になった後、実行を正常に停止します。
transaction_count	>0 の場合、transaction_count のトランザクション数をパージ後、正常にシャットダウンします。
write_trace	TRUE に設定すると、Oracle では、サーバーのトレース・ファイルのファンクションから戻された結果値を記録します。

DBMS_DEFER_SYS.SCHEDULE_PUSH

目的

遅延トランザクション・キューをリモート・マスター接続先に送信するジョブをスケジュールします。このプロシージャはコミットを実行します。

構文

SCHEDULE_PUSHプロシージャのパラメータは、表 9-42を参照してください。このプロシージャの構文は、次のとおりです。

```
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)
```

表 9-42 SCHEDULE_PUSH 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の省略のない形のデータベース名。
interval	送信する次の時間を計算するファンクションを提供できます。この値は、DEFSCHEDULE ビューの INTERVAL フィールドに格納され、このビューのNEXT_DATEフィールドを計算します。このパラメータのデフォルト値であるNULLを使用すると、このフィールドの値は変更されません。フィールドに直前の値がなかった場合は、フィールドが値NULLで再作成されます。このフィールドに値を指定しない場合、NEXT_DATEに値を設定する必要があります。
next_date	マスター・サイトの接続先から遅延トランザクションを送信する時間を指定できます。この値は、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であるNULLを使用すると、このフィールドの値は変更されません。このフィールドに直前の値がなかった場合は、フィールドが値NULLで再作成されます。このフィールドに値を指定しない場合、INTERVAL に対して値を指定する必要があります。
reset	LAST_TXN_COUNT および LAST_ERROR、LAST_MSG を NULL にリセットするには、TRUE に設定します。

表 9-42 SCHEDULE_PUSH のパラメータ

パラメータ	説明
parallelism	0 = シリアル伝播、n > 0 = n個のパラレル・サーバー・プロセスを使用したパラレル伝播、1 = 1つのパラレル・サーバー・プロセスだけを使用したパラレル伝播。
heap_size	パラレル伝播スケジューリングのために、同時に検査されるトランザクションの最大数。Oracleが自動的に最適なパフォーマンスのためのデフォルト設定を計算します。オラクル・サポートで指示されない限り、このパラメータを設定しないでください。
stop_on_error	デフォルトの FALSE では、競合などのエラーが発生しても、executor が継続します。TRUE では、トランザクションが接続先サイトで最初にエラーを検出したときに（可能であれば正常に）停止します。
write_trace	TRUE に設定すると、Oracle では、サーバーのトレース・ファイルのファンクションから戻された結果値を記録します。
startup_seconds	同じ接続先の直前の送信を待つ最大秒数。
execution_seconds	>0 の場合、指定の秒数後に実行を正常に停止します。transaction_count および execution_seconds がゼロ（デフォルト）の場合、キューに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。
delay_seconds	キューが空でも、指定の秒数が経過するまでは戻りません。タイトなループから DBMS_DEFER_SYS.PUSH がコールされた場合に、実行オーバーヘッドを削減するのに有効です。
transaction_count	> 0 の場合、停止までに送られるトランザクションの最大数。transaction_count および execution_seconds がゼロ（デフォルト）の場合、キューに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。

DBMS_DEFER_SYS.SET_DISABLED

目的

現行のサイトから指定した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。disabled パラメータが TRUE の場合は、指定した接続先への伝播が使用禁止になり、この後で DBMS_DEFER_SYS.EXECUTE をコールしても、遅延リモート・プロシージャ・コール（RPC）キューは送られません。SET_DISABLED は、所定の接続先にキューをすでに送っているセッション、または DBMS_DEFER でキューに追加しているセッションには影響しません。disabled パラメータが FALSE の場合は、プロシージャで、指定の接続先への伝播が可能になります。このプロシージャではキューを送信しませんが、将来、DBMS_DEFER_SYS.EXECUTE を起動して、キューを指定の接続先へ送信することができます。disabled パラメータが TRUE でも FALSE でも、他のセッションにその設定を適用するには「コミット」する必要があります。

```
DBMS_DEFER_SYS.SET_DISABLED(
    destination IN VARCHAR2,
    disabled    IN BOOLEAN := TRUE)
```

パラメータ	説明
destination	伝播ステータスを変更したいノードの、省略のない形のデータベース名。
disabled	デフォルトでは、このパラメータは、現行の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・キューの伝播を使用禁止にします。伝播を使用可能にするには、このパラメータを FALSE に設定します。

例外	説明
NO_DATA_FOUND	指定の DESTINATION の DEFSCHEDULE ビューに、入力が見つかりませんでした。

DBMS DEFER SYS.UNREGISTER PROPAGATOR

目的

- DEFPROPAGATOR から指定の伝播担当者を削除する
- REGISTER_PROPAGATOR で付与した権限を、指定のユーザーから取り消す（独立して付与された同一の権限も含む）
- 指定の伝播担当者のスキーマで生成されたすべての Wrapper を削除し、レプリケーション・カタログで削除済みとしてマーク付けする

構文

DBMS DEFER SYS.UNREGISTER PROPAGATOR(

DBMS_DEFER_SYS.UNSCHEDULE_PURGE

```
username IN VARCHAR2
timeout  IN INTEGER DEFAULT DBMS_LOCK.MAXWAIT)
```

表 9-45 UNREGISTER_PROPAGATOR のパラメータ

パラメータ	説明
username	伝播担当者ユーザーの名前。
timeout	タイムアウトの秒数。伝播担当者が使われている場合、プロシージャはタイムアウトまで待機します。デフォルトは DBMS_LOCK.MAXWAIT です。

例外

表 9-46 UNREGISTER_PROPAGATOR の例外

パラメータ	説明
missingpropagator	指定のユーザーは伝播担当者ではありません。
propagator_inuse	伝播担当者が使われているので、登録解除できません。後で再試行してください。

DBMS_DEFER_SYS.UNSCHEDULE_PURGE

目的

スナップショット・サイトまたはマスター・サイトの遅延トランザクション・キューからの送信済みトランザクションの自動削除を停止します。

構文

このプロシージャの構文は、次のとおりです。

```
DBMS_DEFER_SYS.UNSCHEDULE_PURGE
```

DBMS_DEFER_SYS.UNSCHEDULE_PUSH

目的

スナップショット・サイトまたはマスター・サイトから別のマスター・サイトへの、遅延トランザクションの自動送信を停止します。

構文

UNSCHEDULE_PUSH プロシージャのパラメータは、表 9-47 を参照してください。例外は、表 9-48 を参照してください。このプロシージャの構文は、次のとおりです。

```
DBMS_DEFER_SYS.UNSCHEDULE_PUSH( dblink IN VARCHAR2)
```

表 9-47 UNSCHEDULE_PUSH 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のスケジュールを解除するデータベース・サイトへの省略のない形のパス名。

表 9-48 UNSCHEDULE_PUSH の例外

例外	説明
NO_DATA_FOUND	指定の DBLINK の DEFSCHEDULE ピューに、入力が見つかりませんでした。

DBMS_OFFLINE_OG パッケージ

DBMS_OFFLINE_OG パッケージには次のプロシージャが含まれます。

- DBMS_OFFLINE_OG.BEGIN_INSTANTIATION
- DBMS_OFFLINE_OG.BEGIN_LOAD
- DBMS_OFFLINE_OG.END_INSTANTIATION
- DBMS_OFFLINE_OG.END_LOAD
- DBMS_OFFLINE_OG.RESUME_SUBSET_OF_MASTERS

各プロシージャについて説明します。

DBMS_OFFLINE_OG.BEGIN_INSTANTIATION

目的

レプリケート・オブジェクト・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でコールしなければなりません。

構文

BEGIN_INSTANTIATION プロシージャのパラメータは、表 9-49 を参照してください。例外については、表 9-50 を参照してください。このプロシージャの構文は、次のとおりです。

```
DBMS_OFFLINE_OG.BEGIN_INSTANTIATION(  
    gname      IN   VARCHAR2,  
    new_site   IN   VARCHAR2)
```

表 9-49 BEGIN_INSTANTIATION のパラメータ

パラメータ	説明
gname	新規サイトにレプリケートするオブジェクト・グループの名前。
new_site	オブジェクト・グループのレプリケート先となる新規サイトの、省略がない形のデータベース名。

表 9-50 BEGIN_INSTANTIATION の例外

例外	説明
badargument	オブジェクト・グループまたは新規マスター・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でコールしなければなりません。
sitealreadyexists	指定されたサイトは、このオブジェクト・グループのマスター・サイトです。
wrongstate	マスター定義サイトのステータスは休止状態でなければなりません。
dbms_repcat.missingrepgroup	GNAMEがレプリケート・オブジェクト・グループとして存在しません。

DBMS_OFFLINE_OG.BEGIN_LOAD

目的

オフライン・インスタンスーションの一部としてデータを新規マスター・サイトにインポートしている間、トリガーを使用禁止にします。このプロシージャは、新規マスター・サイトでコールしなければなりません。7-13 ページの「スナップショット・クローニングとオフライン・インスタンスーション」を参照してください。

構文

BEGIN_LOAD プロシージャのパラメータは、表 9-51 を参照してください。例外は、表 9-52 を参照してください。このプロシージャの構文は、次のとおりです。

```
DBMS_OFFLINE_OG.BEGIN_LOAD(  
    gname      IN   VARCHAR2,  
    new_site   IN   VARCHAR2)
```

表 9-51 BEGIN_LOAD のパラメータ

パラメータ	説明
gname	インポートするメンバーが属しているオブジェクト・グループの名前。
new_site	オブジェクト・グループ・メンバーのインポート先となる新規サイトの、省略がない形のデータベース名。

表 9-52 BEGIN_LOAD の例外

例外	説明
badargument	オブジェクト・グループまたは新規マスター・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、新規マスター・サイトでコールしなければなりません。
unknownsite	指定されたサイトは、オブジェクト・グループによって認識されません。
wrongstate	新規マスター・サイトのステータスは休止状態でなければなりません。
dbms_repcat.missingrepgroup	GNAMEがレプリケート・オブジェクト・グループとして存在しません。

DBMS_OFFLINE_OG.END_INSTANTIATION

目的

レプリケート・オブジェクト・グループのオフライン・インスタンス化を完了します。このプロシージャは、マスター定義サイトでコールしなければなりません。7-13ページの「スナップショット・クローニングとオフライン・インスタンス化」を参照してください。

構文

END_INSTANTIATION プロシージャのパラメータは、表 9-53 を参照してください。例外は、表 9-54 を参照してください。このプロシージャの構文は、次のとおりです。

```
DBMS_OFFLINE_OG.END_INSTANTIATION(
    gname      IN  VARCHAR2,
    new_site   IN  VARCHAR2)
```

表 9-53 END_INSTANTIATION のパラメータ

パラメータ	説明
gname	新規サイトにレプリケートするオブジェクト・グループの名前。
new_site	オブジェクト・グループのレプリケート先となる新規サイトの、省略がない形のデータベース名。

表 9-54 END_INSTANTIATION の例外

例外	説明
badargument	オブジェクト・グループまたは新規マスター・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でコールしなければなりません。
unknownsite	指定されたサイトは、オブジェクト・グループによって認識されません。
wrongstate	マスター定義サイトのステータスは休止状態でなければなりません。
dbms_repcat.missingrepgroup	GNAMEがレプリケート・オブジェクト・グループとして存在しません。

DBMS_OFFLINE_OG.END_LOAD

目的

オフライン・インスタンスーションの一部として、データを新規マスター・サイトにインポートした後、トリガーを再び使用可能にします。このプロシージャは、新規マスター・サイトでコールしなければなりません。詳細は、7-13 ページの「スナップショット・クローニングとオフライン・インスタンスーション」を参照してください。

構文

END_LOAD プロシージャのパラメータは、表 9-55 を参照してください。例外は、表 9-56 を参照してください。このプロシージャの構文は、次のとおりです。

```
DBMS_OFFLINE_OG.END_LOAD(  
    gname      IN   VARCHAR2,  
    new_site   IN   VARCHAR2)
```


表 9-55 END_LOAD のパラメータ

パラメータ	説明
gname	インポートの終わったメンバーが属しているオブジェクト・グループの名前。
new_site	オブジェクト・グループのメンバーのインポート先となった新規サイトの、省略がない形のデータベース名。

表 9-56 END_LOAD の例外

例外	説明
badargument	オブジェクト・グループまたは新規マスター・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、新規マスター・サイトでコールしなければなりません。
unknownsite	指定されたサイトは、オブジェクト・グループによって認識されません。
wrongstate	新規マスター・サイトのステータスは休止状態でない必要があります。
dbms_repcat.missingrepgroup	GNAMEがレプリケート・オブジェクト・グループとして存在しません。

DBMS_OFFLINE_OG.RESUME_SUBSET_OF_MASTERS

目的

レプリケート・オブジェクト・グループのオフライン・インスタンス化中に、新規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、マスター定義サイトでコールしなければなりません。詳細は、7-13ページの「スナップショット・クローニングとオフライン・インスタンス化」を参照してください。

構文

RESUME_SUBSET_OF_MASTERS プロシージャのパラメータは、表 9-57 を参照してください。例外は、表 9-58 を参照してください。このプロシージャの構文は、次のとおりです。

```
DBMS_OFFLINE_OG.RESUME_SUBSET_OF_MASTERS(  
    gname      IN  VARCHAR2,  
    new_site   IN  VARCHAR2)
```

表 9-57 RESUME_SUBSET_OF_MASTERS のパラメータ

パラメータ	説明
gname	新規サイトにレプリケートするオブジェクト・グループの名前。
new_site	オブジェクト・グループのレプリケート先となる新規サイトの、省略がない形のデータベース名。

表 9-58 RESUME_SUBSET_OF_MASTERS の例外

例外	説明
badargument	オブジェクト・グループまたは新規マスター・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でコールしなければなりません。
unknownsite	指定されたサイトは、オブジェクト・グループによって認識されません。
wrongstate	マスター定義サイトのステータスは休止状態でなければなりません。
dbms_repcat.missingrepgroup	GNAMEがレプリケート・オブジェクト・グループとして存在しません。

DBMS_OFFLINE_SNAPSHOT パッケージ

DBMS_OFFLINE_SNAPSHOT パッケージには次のプロシージャが含まれます。

- DBMS_OFFLINE_SNAPSHOT.BEGIN_LOAD
- DBMS_OFFLINE_SNAPSHOT.END_LOAD

各プロシージャについて説明します。

DBMS_OFFLINE_SNAPSHOT.BEGIN_LOAD

目的

新規スナップショットをオフライン・インスタンスーションの一部としてインポートするためのスナップショットを準備します。このプロシージャは、新規スナップショット用のスナップショット・サイトでコールしなければなりません。詳細は、7-13 ページの「スナップショット・クローニングとオフライン・インスタンスーション」を参照してください。

構文

BEGIN_LOAD プロシージャのパラメータは、表 9-59 を参照してください。例外は、表 9-60 を参照してください。このプロシージャの構文は、次のとおりです。

```
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)
```

表 9-59 BEGIN_LOAD 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を使用して作成するスナップショット用のオブジェクト・グループの名前。
sname	新規スナップショット用のスキーマの名前。
master_site	スナップショットのマスター・サイトの、省略がない形のデータベース名。
snapshot_otype	マスター・サイトで作成される一時スナップショットの名前。
storage_c	スナップショット・サイトで新規スナップショットを作成するときに使用する格納オプション。
comment	ユーザーのコメント。
min_communication	TRUE の場合、更新トリガーが列の新しい値を送るのは、UPDATE 文がその列を修正するときだけです。更新トリガーがその列の古い値を送るのは、その列がキー列であるか、修正された列グループ内の列であるときだけです。

表 9-60 BEGIN_LOAD の例外

例外	説明
badargument	オブジェクト・グループまたはスキーマ、マスター・サイト、スナップショットの名前が、NULL または空の文字列です。
dbms_repcat.missingrepgroup	GNAME がレプリケート・オブジェクト・グループとして存在しません。
missingremotesnap	指定されたマスター・サイトで、指定されたスナップショットが見つかりませんでした。
dbms_repcat.missingschema	指定されたスキーマが存在しません。
snaptabmismatch	マスターおよびスナップショットで、スナップショットの実表名が合致しません。

DBMS_OFFLINE_SNAPSHOT.END_LOAD

目的

スナップショットのオフライン・インスタンスエーションを完了します。このプロシージャは、新規スナップショット用のスナップショット・サイトでコールしなければなりません。詳細は、7-13 ページの「スナップショット・クローニングとオフライン・インスタンスエーション」を参照してください。

構文

END_LOAD プロシージャのパラメータは、表 9-61 を参照してください。例外は、表 9-62 を参照してください。このプロシージャの構文は、次のとおりです。

```
DBMS_OFFLINE_SNAPSHOT.END_LOAD(
    gname          IN  VARCHAR2,
    sname          IN  VARCHAR2,
    snapshot_ename IN  VARCHAR2)
```

表 9-61 END_LOAD 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を使用して作成するスナップショット用のオブジェクト・グループの名前。
sname	新規スナップショット用のスキーマの名前。
snapshot_ename	スナップショットの名前。

表 9-62 END_LOAD の例外

例外	説明
badargument	オブジェクト・グループまたはスキーマ、スナップショットの名前が、NULL または空の文字列です。
dbms_repcat.missingrepgroup	GNAMEがレプリケート・オブジェクト・グループとして存在しません。
dbms_repcat.nonsnapshot	このプロシージャは、スナップショット・サイトでコールしなければなりません。

DBMS_RECTIFIER_DIFF パッケージ

DBMS_RECTIFIER_DIFF パッケージには次のプロシージャが含まれます。

- DBMS_RECTIFIER_DIFF.DIFFERENCES
- DBMS_RECTIFIER_DIFF.RECTIFY

各プロシージャについて説明します。

DBMS_RECTIFIER_DIFF.DIFFERENCES

目的

2 つの表の間の違いをチェックします。

構文

DIFFERENCES プロシージャのパラメータは、表 9-63 を参照してください。例外は、表 9-64 を参照してください。このプロシージャの構文は、次のとおりです。

```
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ます。COLUMN_LIST と ARRAY_COLUMNS パラメータは、両方同時に指定できません。

表 9-63 DIFFERENCES のパラメータ

パラメータ	説明
sname1	REFERENCE_SITE のスキーマの名前。
oname1	REFERENCE_SITE の表の名前。
reference_site	リファレンス・データベース・サイトの名前。デフォルトの NULL は、現行のサイトを指定します。
sname2	COMPARISON_SITE のスキーマの名前。
oname2	COMPARISON_SITE の表の名前。
comparison_site	比較データベース・サイトの名前。デフォルトの NULL は、現行のサイトを指定します。
where_clause	比較するために、この制約を満たす行だけが選択されます。デフォルトの NULL は、現行のサイトを指定します。
column_list	2 つの表について比較される 1 つまたは複数の列名のカンマで区切られたリスト。カンマの前後にはスペースを入れてはいけません。デフォルトの NULL は、すべての列を比較する指定です。
array_columns	2 つの表について比較される列名の PL/SQL 表。索引は 1 から始まり、配列の最後の要素は NULL でなければなりません。位置 1 が NULL の場合、すべての列が使用されます。
missing_rows_sname	欠落行の情報がある表を含むスキーマの名前。
missing_rows_oname1	REFERENCE サイトの表にはあって COMPARISON サイトの表にはない行、および COMPARISON サイトの表にはあって REFERENCE サイトの表にはない行についての情報を格納する、MISSING_ROWS_SITE の表の名前。
missing_rows_oname2	欠落行についての情報を格納する MISSING_ROWS_SITE の表の名前。この表には、MISSING_ROWS_ONAME1 表の行の rowid および行が存在するサイトの名前、行が欠落しているサイトの名前の 3 つの列があります。
missing_rows_site	MISSING_ROWS_ONAME1 表および MISSING_ROWS_ONAME2 表が置かれるサイトの名前。デフォルトの NULL は、表を現行サイトに置く指定です。

表 9-63 DIFFERENCES のパラメータ

パラメータ	説明
max_missing	"missing_rows_ename" 表に挿入される最大行数を示す整数。 "max_missing" の数値を超える行が欠落している場合、その行数が"missing_rows_ename"に挿入されます。そして、ルーチンは、さらに行が欠落しているかどうかを判断することなく通常に復帰します。この引数は、断片部分が違いすぎるため、欠落行の表の項目が多すぎて継続できなくなる場合に役に立ちます。"max_missing" が 0 以下または NULL の場合は、例外の badnumber が呼び出されます。
commit_rows	COMMIT が発生する前に参照表または比較表に挿入される行またはこれらの表から削除される行の最大数。デフォルトでは、500 行挿入されるかまたは 500 行削除されると COMMIT が発生します。空の文字列 ('') または NULL は、1 つの表のすべての行が挿入されたか削除された後で初めて COMMIT が発行される指定です。

表 9-64 DIFFERENCES 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。
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、SET_COLUMNS に相当するもの）が含まれる必要があります。
badname	表またはスキーマ名が NULL または空の文字列です。
cannotbenull	パラメータに NULL を指定してはなりません。
notshapeequivalent	比較される表の形態が同じではありません。形態とは、列数および表の列名、列のデータ型のことです。
unknowncolumn	列が存在していません。
unsupportedtype	型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

MISSING_ROWS_DATA 表に一意キー制約または主キー制約がある場合は、エラー ORA-00001（一意制約の違反）が発行されます。

DBMS_RECTIFIER_DIFF.RECTIFY

目的

2 つの表の間の違いを解消します。

構文

RECTIFY プロシージャのパラメータは、表 9-65 を参照してください。例外は、表 9-66 を参照してください。このプロシージャの構文は、次のとおりです。

```
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ます。COLUMN_LIST と ARRAY_COLUMNS パラメータは、両方同時に指定できません。

表 9-65 RECTIFY のパラメータ

パラメータ	説明
sname1	REFERENCE_SITE のスキーマの名前。
oname1	REFERENCE_SITE の表の名前。
reference_site	リファレンス・データベース・サイトの名前。デフォルトの NULL は、現行のサイトを指定します。
sname2	COMPARISON_SITE のスキーマの名前。
oname2	COMPARISON_SITE の表の名前。
comparison_site	比較データベース・サイトの名前。デフォルトの NULL は、現行のサイトを指定します。

表 9-65 RECTIFY のパラメータ

パラメータ	説明
column_list	2 つの表について比較される 1 つまたは複数の列名のカンマで区切られたリスト。カンマの前後にはスペースを入れてはいけません。デフォルトの NULL は、すべての列を比較する指定です。
array_columns	2 つの表について比較される列名の PL/SQL 表。索引は 1 から始まり、配列の最後の要素は NULL でなければなりません。位置 1 が NULL の場合、すべての列が使用されます。
missing_rows_sname	欠落行の情報がある表を含むスキーマの名前。
missing_rows_onsame1	REFERENCE サイトの表にはあって COMPARISON サイトの表にはない行、および COMPARISON サイトの表にはあって REFERENCE サイトの表にはない行についての情報を格納する、MISSING_ROWS_SITE の表の名前。
missing_rows_onsame2	欠落行についての情報を格納する MISSING_ROWS_SITE の表の名前。この表には、MISSING_ROWS_ONAME1 表の行の rowid および行が存在するサイトの名前、行が欠落しているサイトの名前の 3 つの列があります。
missing_rows_site	MISSING_ROWS_ONAME1 表および MISSING_ROWS_ONAME2 表が置かれるサイトの名前。デフォルトの NULL は、表を現行サイトに置く指定です。
commit_rows	COMMIT が発生する前に参照表または比較表に挿入される行またはこれらの表から削除される行の最大数。デフォルトでは、500 行挿入されるかまたは 500 行削除されると COMMIT が発生します。空の文字列 ('') または NULL は、1 つの表のすべての行が挿入されたか削除された後で初めて COMMIT が発行される指定です。

表 9-66 RECTIFY 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。
badnumber	COMMIT_ROWS パラメータが 1 未満です。
badname	表またはスキーマ名が NULL または空の文字列です。
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。

DBMS_REFRESH パッケージ

DBMS_REFRESH パッケージには次のプロシージャが含まれます。

- DBMS_REFRESH.ADD
- DBMS_REFRESH.CHANGE
- DBMS_REFRESH.DESTROY
- DBMS_REFRESH.MAKE
- DBMS_REFRESH.REFRESH
- DBMS_REFRESH.SUBTRACT

各プロシージャについて説明します。

DBMS_REFRESH.ADD

目的

リフレッシュ・グループにスナップショットを追加します。詳細は、2-36 ページの「スナップショット・リフレッシュとリフレッシュ・グループの管理」を参照してください。

構文

ADD プロシージャのパラメータは、表 9-67 を参照してください。このプロシージャの構文は、次のとおりです。

```
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ます。LIST と TAB パラメータは、両方同時に指定できません。

表 9-67 ADD 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。
list	リフレッシュ・グループに追加する、カンマで区切られたスナップショットのリスト。(シノニムはサポートされていません)

表 9-67 ADD のパラメータ

パラメータ	説明
tab	カンマ区切りリストのかわりに、DBMS_UTILITY.UNCL_ARRAY 型の PL/SQL 表を提供します。このとき、各要素がスナップショット名です。最初のスナップショットは位置 1 に入ります。最後の位置は NULL でなければなりません。
lax	1 つのスナップショットは、一度に 1 つのリフレッシュ・グループにだけ所属できます。スナップショットの別のグループへの移動を正常実行するには、LAX フラグを TRUE に設定する必要があります。このようにすると、Oracle は自動的にスナップショットを元のリフレッシュ・グループから削除し、そのリフレッシュ間隔を新規グループの間隔に更新します。そうしないと、ADD へのコールでエラー・メッセージが生成されます。

DBMS_REFRESH.CHANGE

目的

スナップショット・グループのリフレッシュ間隔を変更します。詳細は、2-36 ページの「スナップショット・リフレッシュとリフレッシュ・グループの管理」を参照してください。

構文

CHANGE プロシージャのパラメータは、表 9-68 を参照してください。このプロシージャの構文は、次のとおりです。

```
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)
```

表 9-68 CHANGE 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。
next_date	次にリフレッシュを実行する日付。デフォルトでは、この日付は変更されずにそのまま残ります。
interval	グループ中のスナップショットの次のリフレッシュがいつになるかを計算するのに使用するファンクション。この間隔は、リフレッシュの直前に計算されます。このため、リフレッシュを実行するの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、この間隔は変更されずにそのまま残ります。
implicit_destroy	ユーザーがIMPLICIT_DESTROY フラグの値をリセットできるようにします。このフラグを設定すると、グループにメンバーが含まれていない場合、Oracleは自動的にそのグループを削除します。デフォルトでは、このフラグは変更されずにそのまま残ります。
rollback_seg	ユーザーが、使用するロールバック・セグメントを変更可能にします。デフォルトでは、ロールバック・セグメントは変更されずにそのまま残ります。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、引用符も含めてNULLと指定してください。引用符を付けずにNULLを指定すると、現在使用しているロールバック・セグメントを変更しないことを示します。
push_deferred_rpc	更新可能なスナップショットによってだけ使用されます。スナップショットをリフレッシュする前に、スナップショットから、それに関連付けられたマスターに変更を送信するには、このパラメータをTRUEに設定します。そうしないと、これらの変更は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ずにそのまま残ります。
refresh_after_errors	更新可能なスナップショットによってだけ使用されます。スナップショットのマスターの DEFERROR ビューに、未解消の競合がまだ記録されていても、リフレッシュを続行する場合は、このパラメータにTRUEを設定します。デフォルトでは、このフラグは変更されずにそのまま残ります。

表 9-68 CHANGE のパラメータ

パラメータ	説明
purge_option	パラレル伝播のメカニズムを使用する場合（つまり、parallelism に 1 以上を設定する場合）の指定は次のとおりです。0 = パージなし、1 = レイジー・パージ（デフォルト）、2 = アグレッシブ。ほとんどの場合、レイジー・パージの設定が最適です。複数のマスター・レプリケーション・グループがさまざまなターゲット・サイトに送られて、1 つ以上のレプリケーション・グループに対する更新がまれであり、まれにしか送信されない場合、パージをアグレッシブに設定してキューを減らします。すべてのレプリケーション・グループがまれにしか更新されず、送信されない場合は、パージをパージなしに設定しておき、ときおりパージをアグレッシブに設定して PUSH を実行してキューを低減します。
parallelism	0 = シリアル伝播、n > 0 = n 個のパラレル・サーバー・プロセスを使用したパラレル伝播、1 = 1 つのパラレル・サーバー・プロセスだけを使用したパラレル伝播。
heap_size	パラレル伝播スケジューリングのために、同時に検査されるトランザクションの最大数。Oracle が自動的に最適なパフォーマンスのためのデフォルト設定を計算します。オラクル・サポートで指示されない限り、このパラメータを設定しないでください。

DBMS_REFRESH.DESTROY

目的

リフレッシュ・グループからすべてのスナップショットを削除し、リフレッシュ・グループを削除します。詳細は、2-36 ページの「スナップショット・リフレッシュとリフレッシュ・グループの管理」を参照してください。

構文

DESTROY プロシージャのパラメータは、表 9-69 を参照してください。このプロシージャの構文は、次のとおりです。

```
DBMS_REFRESH.DESTROY(name IN VARCHAR2)
```

表 9-69 DESTROY のパラメータ

パラメータ	説明
name	削除するリフレッシュ・グループの名前。

DBMS_REFRESH.MAKE

目的

リフレッシュ・グループのメンバー、およびリフレッシュするために用いられる時間間隔を指定します。詳細は、2-36 ページの「スナップショット・リフレッシュとリフレッシュ・グループの管理」を参照してください。

構文

MAKE プロシージャのパラメータは、表 9-70 を参照してください。このプロシージャの構文は、次のとおりです。

```
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ます。LIST と TAB パラメータは、両方同時に指定できません。

表 9-70 MAKE のパラメータ

パラメータ	説明
name	リフレッシュ・グループを識別するのに使用される一意の名前。リフレッシュ・グループは、表と同じ命名規則に従っている必要があります。

表 9-70 MAKE のパラメータ

パラメータ	説明
list	リフレッシュするスナップショットのカンマで区切られたリスト。(シノニムはサポートされていません)。これらのスナップショットは別々のスキーマに置かれていても、別々のマスター表を持っていたりもかまいませんが、リストされたスナップショットはすべて現行データベースに入っている必要があります。
tab	カンマ区切りリストのかわりに、DBMS_UTILITY.UNCL_ARRAY 型を使用して、リフレッシュしたいスナップショットの名前の PL/SQL 表を提供できます。表に N 個のスナップショットの名前が含まれる場合、最初のスナップショットが位置 1 にあり、N + 1 の位置は NULL に設定されている必要があります。
next_date	次にリフレッシュを実行する日付。
interval	グループ中のスナップショットの次のリフレッシュがいつになるかを計算するのに使用するファンクション。このフィールドは、NEXT_DATE 値といっしょに使用します。たとえば、自分が使用する間隔として NEXT_DAY (SYSDATE+1, "MONDAY") を指定し、NEXT_DATE の値が月曜日になると、Oracle はスナップショットを毎月曜日にリフレッシュします。この間隔は、リフレッシュの直前に計算されます。このため、リフレッシュを実行するのにかかる時間より長い間隔を選択する必要があります。2-26 ページの「日付式の例」を参照してください。
implicit_destroy	リフレッシュ・グループのメンバーが空になった時点で、そのリフレッシュ・グループを自動削除する場合は、この引数を TRUE に設定します。Oracle では、SUBTRACT プロシージャをコールしたときだけ、このフラグをチェックします。つまり、このフラグを設定していても、空のリフレッシュ・グループを作成することは可能です。
lax	1 つのスナップショットは、一度に 1 つのリフレッシュ・グループにだけ所属できます。既存のグループから新規のリフレッシュ・グループへのスナップショットの移動を成功させるには、LAX フラグを TRUE に設定する必要があります。このようにすると、Oracle は自動的にスナップショットを元のリフレッシュ・グループから削除し、そのリフレッシュ間隔を新規グループの間隔に更新します。そうしないと、MAKE へのコールでエラー・メッセージが生成されません。
job	このパラメータは、インポート・ユーティリティで必要です。デフォルト値の 0 を使用してください。
rollback_seg	スナップショットをリフレッシュする際に使用するロールバック・セグメントの名前。デフォルトの NULL では、デフォルト・ロールバック・セグメントが使用されます。

表 9-70 MAKE のパラメータ

パラメータ	説明
push_deferred_rpc	更新可能なスナップショットによってだけ使用されます。スナップショットをリフレッシュする前にスナップショットから関連マスターへの変更を送信する場合、デフォルト値のTRUEを使用します。そうしないと、これらの変更は一時的に失われたように見えることがあります。
refresh_after_errors	更新可能なスナップショットによってだけ使用されます。スナップショットのマスターの DEFERROR ビューに、未解消の競合がまだ記録されていても、リフレッシュを続行する場合は、このパラメータに TRUE を設定します。
purge_option	パラレル伝播のメカニズムを使用する場合（つまり、parallelism に 1 以上を設定する場合）の指定は次のとおりです。0 = パージなし、1 = レイジー・パージ（デフォルト）、2 = アグレッシブ。ほとんどの場合、レイジー・パージの設定が最適です。複数のマスター・レプリケーション・グループがさまざまなターゲット・サイトに送られて、1つ以上のレプリケーション・グループに対する更新がまれであり、まれにしか送信されない場合、パージをアグレッシブに設定してキューを減らします。すべてのレプリケーション・グループがまれにしか更新されず、送信されない場合は、パージをパージなしに設定しておき、ときおりパージをアグレッシブに設定して PUSH を実行してキューを低減します。
parallelism	0 = シリアル伝播、n > 0 = n 個のパラレル・サーバー・プロセスを使用したパラレル伝播、1 = 1つのパラレル・サーバー・プロセスだけを使用したパラレル伝播。
heap_size	パラレル伝播スケジューリングのために、同時に検査されるトランザクションの最大数。Oracle が自動的に最適なパフォーマンスのためのデフォルト設定を計算します。オラクル・サポートで指示されない限り、このパラメータを設定しないでください。

DBMS_REFRESH.REFRESH

目的

リフレッシュ・グループを手動でリフレッシュします。詳細は、2-36 ページの「スナップショット・リフレッシュとリフレッシュ・グループの管理」を参照してください。

構文

REFRESH プロシージャのパラメータは、表 9-71 を参照してください。このプロシージャの構文は、次のとおりです。

```
DBMS_REFRESH.REFRESH(name IN VARCHAR2)
```


表 9-71 REFRESH 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。

DBMS_REFRESH.SUBTRACT

目的

リフレッシュ・グループからスナップショットを削除します。詳細は、2-36 ページの「スナップショット・リフレッシュとリフレッシュ・グループの管理」を参照してください。

構文

SUBTRACT プロシージャのパラメータは、表 9-72 を参照してください。このプロシージャの構文は、次のとおりです。

```
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ます。LIST と TAB パラメータは、両方同時に指定できません。

表 9-72 SUBTRACT 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。
list	リフレッシュ・グループから削除する、カンマで区切られたスナップショットのリスト（シノニムはサポートされていません）。これらのスナップショットは別々のスキーマに置かれていても、別々のマスター表を持っていたりもかまいませんが、リストされたスナップショットはすべて現行データベースに入っている必要があります。

表 9-72 SUBTRACT のパラメータ

パラメータ	説明
tab	カンマ区切りリストのかわりに、DBMS_UTILITY.UNCL_ARRAY 型を使用して、リフレッシュしたいスナップショットの名前の PL/SQL 表を提供できます。表に N 個のスナップショットの名前が含まれる場合、最初のスナップショットが位置 1 にあり、N + 1 の位置は NULL に設定されている必要があります。
lax	削除するスナップショットが、リフレッシュ・グループのメンバーでない場合に、Oracle がエラー・メッセージを生成するようにする場合、このパラメータを FALSE に設定します。

DBMS_REPCAT パッケージ

The DBMS_REPCAT パッケージには、次のプロシージャおよびファンクションが含まれます。

- DBMS_REPCAT.ADD_GROUPED_COLUMN
- DBMS_REPCAT.ADD_MASTER_DATABASE
- DBMS_REPCAT.ADD_PRIORITY_datatype
- DBMS_REPCAT.ADD_SITE_PRIORITY_SITE
- DBMS_REPCAT.ADD_conflicttype_RESOLUTION
- DBMS_REPCAT.ALTER_MASTER_PROPAGATION
- DBMS_REPCAT.ALTER_MASTER_REPOBJECT
- DBMS_REPCAT.ALTER_PRIORITY
- DBMS_REPCAT.ALTER_PRIORITY_datatype
- DBMS_REPCAT.ALTER_SITE_PRIORITY
- DBMS_REPCAT.ALTER_SITE_PRIORITY_SITE
- DBMS_REPCAT.ALTER_SNAPSHOT_PROPAGATION
- DBMS_REPCAT.CANCEL_STATISTICS
- DBMS_REPCAT.COMMENT_ON_COLUMN_GROUP
- DBMS_REPCAT.COMMENT_ON_PRIORITY_GROUP/
- DBMS_REPCAT.COMMENT_ON_REPGROUP
- DBMS_REPCAT.COMMENT_ON_REPSITES
- DBMS_REPCAT.COMMENT_ON_REPOBJECT
- DBMS_REPCAT.COMMENT_ON_conflicttype_RESOLUTION

- DBMS_REPCAT.CREATE_MASTER_REPGROUP
- DBMS_REPCAT.CREATE_MASTER_REPOBJECT
- DBMS_REPCAT.CREATE_SNAPSHOT_REPGROUP
- DBMS_REPCAT.CREATE_SNAPSHOT_REPOBJECT
- DBMS_REPCAT.DEFINE_COLUMN_GROUP
- DBMS_REPCAT.DEFINE_PRIORITY_GROUP
- DBMS_REPCAT.DEFINE_SITE_PRIORITY
- DBMS_REPCAT.DO_DEFERRED_REPCAT_ADMIN
- DBMS_REPCAT.DROP_COLUMN_GROUP
- DBMS_REPCAT.DROP_GROUPED_COLUMN
- DBMS_REPCAT.DROP_MASTER_REPGROUP
- DBMS_REPCAT.DROP_MASTER_REPOBJECT
- DBMS_REPCAT.DROP_PRIORITY
- DBMS_REPCAT.DROP_PRIORITY_GROUP
- DBMS_REPCAT.DROP_PRIORITY_datatype
- DBMS_REPCAT.DROP_SITE_PRIORITY
- DBMS_REPCAT.DROP_SITE_PRIORITY_SITE
- DBMS_REPCAT.DROP_SNAPSHOT_REPGROUP
- DBMS_REPCAT.DROP_SNAPSHOT_REPOBJECT
- DBMS_REPCAT.DROP_conflicttype_RESOLUTION
- DBMS_REPCAT.EXECUTE_DDL
- DBMS_REPCAT.GENERATE_REPLICATION_PACKAGE
- DBMS_REPCAT.GENERATE_REPLICATION_SUPPORT
- DBMS_REPCAT.GENERATE_REPLICATION_TRIGGER
- DBMS_REPCAT.GENERATE_SNAPSHOT_SUPPORT
- DBMS_REPCAT.MAKE_COLUMN_GROUP
- DBMS_REPCAT.PURGE_MASTER_LOG
- DBMS_REPCAT.PURGE_STATISTICS
- DBMS_REPCAT.REFRESH_SNAPSHOT_REPGROUP

- DBMS_REPCAT.REGISTER_SNAPSHOT_REPGROUP
- DBMS_REPCAT.REGISTER_STATISTICS
- DBMS_REPCAT.RELOCATE_MASTERDEF
- DBMS_REPCAT.REMOVE_MASTER_DATABASES
- DBMS_REPCAT.REPCAT_IMPORT_CHECK
- DBMS_REPCAT.RESUME_MASTER_ACTIVITY
- DBMS_REPCAT.SEND_AND_COMPARE_OLD_VALUES
- DBMS_REPCAT.SET_COLUMNS
- DBMS_REPCAT.SUSPEND_MASTER_ACTIVITY
- DBMS_REPCAT.SWITCH_SNAPSHOT_MASTER
- DBMS_REPCAT.UNREGISTER_SNAPSHOT_REPGROUP
- DBMS_REPCAT.VALIDATE
- DBMS_REPCAT.WAIT_MASTER_LOG

各プロシージャおよびファンクションについて説明します。

DBMS_REPCAT.ADD_GROUPED_COLUMN

目的

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でコールしなければなりません。

構文

ADD_GROUPED_COLUMN プロシージャのパラメータは、表 9-73 を参照してください。例外は、表 9-74 を参照してください。このプロシージャの構文は、次のとおりです。

```
DBMS_REPCAT.ADD_GROUPED_COLUMN(  
    sname,                IN   VARCHAR2,  
    oname,                IN   VARCHAR2,  
    column_group          IN   VARCHAR2,  
    list_of_column_names  IN   VARCHAR2 | DBMS_REPCAT.VARCHAR2S)
```

表 9-73 ADD_GROUPED_COLUMN 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ているスキーマの名前。
oname	列グループに対応するレプリケート表の名前。
column_group	メンバーを追加するグループの名前。
list_of_column_names	指定された列グループに追加する列の名前。これは、カンマで区切られた列名のリスト、または PL/SQL 表の列名のどちらでもかまいません。PL/SQL 表は dbms_repcat.varchar2 型にしてください。表内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。

表 9-74 ADD_GROUPED_COLUMN 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ていません。
duplicatecolumn	指定された列は、すでに別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が存在しません。
notquiesced	指定の表が属するオブジェクト・グループは休止していません。

DBMS_REPCAT.ADD_MASTER_DATABASE

目的

レプリケート環境に別のマスター・サイトを追加します。このプロシージャは、すべてのトリガーおよびこれらに関連付けられたパッケージを、既存のマスター・サイトで再生成します。このプロシージャは、マスター定義サイトでコールしなければなりません。

構文

ADD_MASTER_DATABASE プロシージャのパラメータは、表 9-75 を参照してください。例外は、表 9-76 を参照してください。このプロシージャの構文は、次のとおりです。

```
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')

```

表 9-75 ADD_MASTER_DATABASE のパラメータ

パラメータ	説明
gname	レプリケートされているオブジェクト・グループの名前。 このオブジェクト・グループは、すでにマスター定義サイトに存在していなければなりません。
master	新規マスター・データベースの、省略のない形のデータベース名。
use_existing_objects	新規マスター・サイトにおいてすでにスキーマ中に存在しているオブジェクトと、同じ型および内容のオブジェクトを再使用する場合、TRUEと指定します。これらの変更の適用方法の詳細は、3-18 ページの「オブジェクト定義のマスター・サイトへのレプリケート」を参照してください。
copy_rows	新規マスター・サイトの表の初期の内容が、マスター定義サイトの表の内容と一致するようにする場合、TRUEを指定します。
comment	このコメントは、RepSite ビューの MASTER_COMMENT フィールドに追加されます。
propagation_mode	新規のマスター・データベースへの変更の転送およびこのデータベースからの変更の受信の方法。指定できる値は、SYNCHRONOUS および ASYNCHRONOUS です。

表 9-76 ADD_MASTER_DATABASE 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	レプリケート・オブジェクト・グループは、まだ中断されていません。
missingrepgroup	オブジェクト・グループは、指定したデータベース・サイトに存在していません。
commfailure	新規マスターにアクセスできません。
typefailure	誤った伝播モードを指定しました。
notcompat	互換モードは、7.3.0.0 以上でなければなりません。
duplrepgrp	マスター・サイトはすでに存在しています。

DBMS_REPCAT.ADD_PRIORITY_datatype

目的

優先順位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でコールしなければなりません。コールする必要があるプロシージャは、「優先順位」列のデータ型によって決まります。このプロシージャは、「優先順位」列の各有効値ごとに1度コールしてください。

追加情報は、5-17ページの「優先順位グループおよびサイト優先順位」を参照してください。

構文

ADD_PRIORITY_datatype プロシージャのパラメータは、表 9-77 を参照してください。例外は、表 9-78 を参照してください。ADD_PRIORITY_datatype プロシージャの構文は、次のとおりです。

```
DBMS_REPCAT.ADD_PRIORITY_datatype(
    gname          IN   VARCHAR2,
    pgroup         IN   VARCHAR2,
    value          IN   datatype,
    priority       IN   NUMBER)
```

datatype には、次の引数を指定します。

```
{ NUMBER
| VARCHAR2
| CHAR
| DATE
```

DBMS_REPCAT.ADD_SITE_PRIORITY_SITE

| RAW
| NCHAR
| NVARCHAR2 }

表 9-77 ADD_PRIORITY_datatype のパラメータ

パラメータ	説明
gname	優先順位グループを作成しているレプリケート・オブジェクト・グループの名前。
pgroup	優先グループの名前。
value	優先順位グループ・メンバーの値。これは、この優先順位グループを使用して関連付けられた表の「優先順位」列の有効値のどれかになります。
priority	この値の優先順位。数値が大きいほど、優先順位は高くなります。

表 9-78 ADD_PRIORITY_datatype 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順位グループ中にすでに存在しています。
duplicatepriority	指定された優先順位は、優先順位グループ中にすでに存在しています。
missingrepgroup	指定されたレプリケート・オブジェクト・グループは存在しません。
missingprioritygroup	指定された優先順位グループは存在しません。
typefailure	優先順位グループに指定された値のデータ型が正しくありません。
notquiesced	指定されたレプリケート・オブジェクト・グループは休止していません。

DBMS_REPCAT.ADD_SITE_PRIORITY_SITE

目的

サイト優先順位グループに新規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でコールしなければなりません。詳細は、5-24 ページの「グループへのサイトの追加」を参照してください。

構文

ADD_SITE_PRIORITY_SITE プロシージャのパラメータは、表 9-79 を参照してください。例外は、表 9-80 を参照してください。このプロシージャの構文は、次のとおりです。

```
DBMS_REPCAT.ADD_SITE_PRIORITY_SITE(
    gname          IN   VARCHAR2,
    name           IN   VARCHAR2
    site           IN   VARCHAR2,
    priority       IN   NUMBER)
```

表 9-79 ADD_SITE_PRIORITY_SITE のパラメータ

パラメータ	説明
gname	サイトをグループに追加するレプリケート・オブジェクト・グループ。
name	メンバーを追加するサイト優先順位グループの名前。
site	追加するサイトのグローバルなデータベース名。
priority	追加するサイト優先順位。数値が大きいほど、優先順位レベルが高いことを示します。

表 9-80 ADD_SITE_PRIORITY_SITE 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	指定されたレプリケート・オブジェクト・グループは存在しません。
missingpriority	指定されたサイト優先順位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、すでにグループ中の別のサイトに存在しています。
duplicatevalue	指定されたサイトは、サイト優先順位グループ中にすでに存在しています。
notquiesced	レプリケート・オブジェクト・グループは休止していません。

DBMS_REPCAT.ADD_conflicttype_RESOLUTION

目的

更新、削除または一意の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でコールする必要があります。コールする必要のあるプロシージャは、ルーチンが解消する競合のタイプによって決まります。

競合のタイプ	プロシージャ名
更新	ADD_UPDATE_RESOLUTION
一意	ADD_UNIQUE_RESOLUTION
削除	ADD_DELETE_RESOLUTION

更新の競合を解消するための設計方法の詳細は、5-13 ページの「更新の競合解消方法の割当て」を参照してください。一意性の競合解消方法の選択の詳細は、5-28 ページの「一意性の競合解消方法の割当て」を参照してください。削除の競合解消方法の指定については、5-30 ページの「削除の競合解消方法の割当て」を参照してください。

構文

ADD_conflicttype_RESOLUTION プロシージャのパラメータは、表 9-81 を参照してください。例外は、表 9-82 を参照してください。ADD_UPDATE_RESOLUTION プロシージャの構文は、次のとおりです。

```
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,
    priority_group  IN    VARCHAR2      := NULL,
    function_name   IN    VARCHAR2      := NULL,
    comment        IN    VARCHAR2      := NULL)
```

ADD_DELETE_RESOLUTION プロシージャの構文については、次のとおりです。

```
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)

```

ADD_UNIQUE_RESOLUTION プロシージャの構文は、次のとおりです。

```

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,
    function_name        IN    VARCHAR2    := NULL,
    comment              IN    VARCHAR2    := NULL)

```

表 9-81 ADD_conflicttype_RESOLUTION のパラメータ

パラメータ	説明
sname	レプリケートする表が入っているスキーマの名前。
oname	競合解消ルーチンを追加する表の名前。
column_group	競合解消ルーチンを追加する列グループの名前。列グループが必要なのは、更新競合解消ルーチンだけです。
constraint_name	競合解消ルーチンを追加する一意制約または一意の索引の名前。その名前が関連する一意制約名と異なる場合は、一意索引の名前を使います。制約名は、一意競合解消ルーチンでだけ必要になります。
sequence_no	指定された競合解消方法が適用される順序。
method	作成する競合解消ルーチンの型。アドバンスド・レプリケーションで提供する標準ルーチンの 1 つの名前を指定できます。または、ユーザー自身がルーチンを作成している場合は USER FUNCTION を選択し、引数 FUNCTION_NAME にそのルーチン名を指定します。このリリースでサポートされている方法は、MINIMUM および MAXIMUM、LATEST TIMESTAMP、EARLIEST TIMESTAMP、ADDITIVE、AVERAGE、PRIORITY GROUP、SITE PRIORITY、OVERWRITE、DISCARD（更新競合の場合）および APPEND SITE NAME、APPEND SEQUENCE NUMBER、DISCARD（一意競合の場合）です。削除競合の場合、提供される方法はないので、この引数は使用しません。

表 9-81 ADD_conflicttype_RESOLUTION のパラメータ

パラメータ	説明
parameter_column_name	競合を解消するために使用する列の名前。標準の方法では単一の列に対して操作します。たとえば、列グループに対して LATEST TIMESTAMP方法を使用している場合、タイムスタンプを含む列の名前をこの引数として渡す必要があります。USER FUNCTION を使用する場合、任意の数の列を使用して競合を解消できます。この引数では、カンマによって区切られた列名のリスト、または dbms_repat.varchar2 という型の PL/SQL表のどちらかを受け入れます。単一の値 '*' は、表（または更新競合の場合は列グループ）のすべての列を使用して、競合を解消することを示します。 '*' を指定すると、列はアルファベット順でユーザーのファンクションに渡されます。
priority_group	PRIORITY GROUPまたはSITE PRIORITY更新競合解消方法を使用している場合、作成した優先順位グループの名前を指定する必要があります。 5-17 ページの「優先順位グループおよびサイト優先順位」を参照してください。別の方法を使用している場合、この引数のデフォルト値である NULL を使用できます。この引数を適用できるのは更新競合の場合だけです。
function_name	USER FUNCTION メソッドを選択したか、または削除競合解消ルーチンを追加している場合は、自分で作成した競合解消ルーチンの名前を指定する必要があります。提供される方法のいずれかを使用している場合は、この引数のデフォルト値である NULL を使用できます。
comment	このユーザー・コメントは、RepResolution ビューに追加されます。

表 9-82 ADD_conflicttype_RESOLUTION 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	指定したオブジェクトは、行レベル・レプリケーションを使用する指定のスキーマ内の表として存在していません。
missingschema	指定されたスキーマが存在しません。
missingcolumn	PARAMETER_COLUMN_NAME 引数の一部としてユーザーが指定した列が存在しません。

表 9-82 ADD_conflicttype_RESOLUTION の例外

例外	説明
missinggroup	指定された列グループは存在しません。
missingprioritygroup	ユーザーが指定した表の優先順位グループは存在しません。
invalidmethod	ユーザーが指定した解消方法は認識できません。
invalidparameter	PARAMETER_COLUMN_NAME 引数に対して指定した列数が無効です (標準ルーチンは 1 つの列名しかとりません)。
missingfunction	ユーザーが指定したユーザーファンクションが存在しません。
missingconstraint	ユーザーが指定した一意競合に対する制約が存在しません。
notquiesced	指定の表が属するオブジェクト・グループは休止していません。
duplicateresolution	指定された競合解消方法は、すでに登録されています。
paramtype	型が優先グループに割り当てられた型と異なります。

DBMS_REPCAT.ALTER_MASTER_PROPAGATION

目的

指定したマスター・サイトの指定したオブジェクト・グループの伝播方法を変更します。このオブジェクト・グループは休止していなくてはなりません。このプロシージャは、マスター定義サイトでコールしなければなりません。伝播方法を変更するマスターを dblink_list または dblink_table で指定すると、そのデータベース・リンクは ALTER_MASTER_PROPAGATIONで無視されます。マスターからそれ自身の伝播モードを変更することはできません。

構文

ALTER_MASTER_PROPAGATION プロシージャのパラメータは、表 9-83 を参照してください。例外は、表 9-84 を参照してください。このプロシージャの構文は、次のとおりです。

```
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ます。DBLINK_LIST と DBLINK_TABLE パラメータは、両方同時に指定できません。

表 9-83 ALTER_MASTER_PROPAGATION のパラメータ

パラメータ	説明
gname	伝播モードを変更するオブジェクト・グループの名前。
master	伝播モードを変更するマスター・サイトの名前。
dblink_list	伝播を変更するデータベース・リンクの、カンマで区切られたリスト。NULLの場合、変更されるマスター・サイトを除くすべてのマスターがデフォルトで使用されます。
dblink_table	伝播を変更するデータベース・リンクの、位置 1 から索引が付けられる PL/SQL 表。
propagation_mode	データベース・リンクのリストによって識別されたサイトに、指定したマスター・サイトから変更が伝播する方法を決定します。指定できる値は SYNCHRONOUS および ASYNCHRONOUS です。
comment	このコメントは、RepProp ビューに追加されます。

表 9-84 ALTER_MASTER_PROPAGATION の例外

例外	説明
nonmasterdef	ローカル・サイトはマスター定義サイトではありません。
notquiesced	ローカル・サイトは休止していません。
typefailure	指定した伝播モードは認識されませんでした。
nonmaster	マスター・サイトではないサイトがデータベース・リンクのリストに含まれています。

DBMS_REPCAT.ALTER_MASTER_REPOBJECT

目的

レプリケート環境中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でコールしなければなりません。

構文

ALTER_MASTER_REPOBJECT プロシージャのパラメータは、表 9-85 を参照してください。例外は、表 9-86 を参照してください。このプロシージャの構文は、次のとおりです。

```
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)

```

注意：スキーマを指定しないで DDL を指定すると、レプリケーション管理者のスキーマがデフォルトで指定されます。レプリケーション管理者のスキーマを使用しない場合は、必ずスキーマを指定してください。

表 9-85 ALTER_MASTER_REPOBJECT のパラメータ

パラメータ	説明
sname	変更するオブジェクトを含んでいるスキーマの名前。
oname	変更するオブジェクトの名前。
type	変更するオブジェクトの型。サポートされている型は、TABLE および INDEX、SYNONYM、TRIGGER、VIEW、PROCEDURE、FUNCTION、PACKAGE、PACKAGE BODY です。
ddl_text	オブジェクトを変更するのに使用する DDL テキスト。Oracle はこの DDL を適用する前に、その構文を解析しません。したがって、DDL テキストの中に、変更するオブジェクトのスキーマ名とオブジェクト名を含める必要があります。
comment	NULL になっていない場合、このコメントは RepObject ビューの COMMENT フィールドに追加されます。
retry	retry が TRUE の場合、ALTER_MASTER_REPOBJECT によって、オブジェクト状態が VALID ではないマスターにおいてだけオブジェクトが変更されます。

表 9-86 ALTER_MASTER_REPOBJECT 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	関連のあるオブジェクト・グループは休止していません。
missingobject	SNAME と ONAME で指定されたオブジェクトが存在しません。
typefailure	指定された型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で DDL は成功しませんでした。
commfailure	アクセスできないマスター・サイトが少なくとも 1 つあります。

DBMS_REPCAT.ALTER_PRIORITY

目的

指定された優先順位グループ・メンバーに対応する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でコールしなければなりません。5-21 ページの「メンバーの優先順位の変更」を参照してください。

構文

ALTER_PRIORITY プロシージャのパラメータは、表 9-87 を参照してください。例外は、表 9-88 を参照してください。このプロシージャの構文は、次のとおりです。

```
DBMS_REPCAT.ALTER_PRIORITY(
    gname          IN   VARCHAR2,
    pgroup         IN   VARCHAR2,
    old_priority   IN   NUMBER,
    new_priority   IN   NUMBER)
```

表 9-87 ALTER_PRIORITY のパラメータ

パラメータ	説明
gname	優先順位グループが対応付けられているレプリケート・オブジェクト・グループの名前。
pgroup	変更する優先順位を含む優先順位グループの名前。
old_priority	優先順位グループ・メンバーの現在の優先順位レベル。
new_priority	優先順位グループ・メンバーに割り当てる新規の優先順位レベル。

表 9-88 ALTER_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。
duplicatepriority	その新規優先順位レベルは、優先順位グループ中にすでに存在しています。
missingrepgroup	指定されたレプリケート・オブジェクト・グループは存在しません。
missingvalue	値が、DBMS_REPCAT.ADD_PRIORITY_datatype のコールで登録されていません。
missingprioritygroup	指定された優先順位グループは存在しません。
notquiesced	指定されたレプリケート・オブジェクト・グループは休止していません。

DBMS_REPCAT.ALTER_PRIORITY_datatype

目的

優先グループ内の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でコールしなければなりません。コールする必要のあるプロシージャは、「優先順位」列のデータ型によって決まります。

詳細は、5-20 ページの「メンバーの値の変更」を参照してください。

構文

ALTER_PRIORITY_datatype プロシージャのパラメータは、表 9-89 を参照してください。例外は、表 9-90 を参照してください。ALTER_PRIORITY_datatype プロシージャの構文は、次のとおりです。

```
DBMS_REPCAT.ALTER_PRIORITY_datatype(
    gname          IN   VARCHAR2,
    pgroup         IN   VARCHAR2,
    old_value      IN   datatype,
    new_value      IN   datatype)
```

datatype には、次の引数を指定します。

```
{ NUMBER
| VARCHAR2
| CHAR
| DATE
| RAW
```

| NCHAR
| NVARCHAR2 }

表 9-89 ALTER_PRIORITY_datatype のパラメータ

パラメータ	説明
gname	優先順位グループが対応付けられているレプリケート・オブジェクト・グループの名前。
pgroup	変更する値を含む優先順位グループの名前。
old_value	優先順位グループ・メンバーの現在の設定値。
new_value	優先順位グループ・メンバーに割り当てる新しい値。

表 9-90 ALTER_PRIORITY_datatype 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	その新しい値は優先順位グループ中にすでに存在しています。
missingrepgroup	指定されたレプリケート・オブジェクト・グループは存在しません。
missingprioritygroup	指定された優先順位グループは存在しません。
missingvalue	現在の値は存在しません。
paramtype	優先順位グループの新しい値のデータ型が正しくありません。
typefailure	優先順位グループに指定された値のデータ型が正しくありません。
notquiesced	指定されたレプリケート・オブジェクト・グループは休止していません。

DBMS_REPCAT.ALTER_SITE_PRIORITY

目的

指定されたサイトに関連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でコールしなければなりません。5-24 ページの「サイトの優先順位レベルの変更」を参照してください。

構文

ALTER_SITE_PRIORITY プロシージャのパラメータは、表 9-91 を参照してください。例外は、表 9-92 を参照してください。このプロシージャの構文は、次のとおりです。

```
DBMS_REPCAT.ALTER_SITE_PRIORITY(
    gname          IN   VARCHAR2,
    name           IN   VARCHAR2
    old_priority   IN   NUMBER,
    new_priority   IN   NUMBER)
```

表 9-91 ALTER_SITE_PRIORITY のパラメータ

パラメータ	説明
gname	サイト優先順位グループが対応付けられているレプリケート・オブジェクト・グループの名前。
name	メンバーを変更するサイト優先順位グループの名前。
old_priority	優先順位レベルを変更するサイトの現在の優先順位レベル。
new_priority	そのサイトの新規優先順位レベル。数値が大きいほど、優先順位レベルが高いことを示します。

表 9-92 ALTER_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	指定されたレプリケート・オブジェクト・グループは存在しません。
missingpriority	現在の優先順位レベルが、どのグループ・メンバーとも関連付けられていません。
duplicatepriority	この新規優先順位レベルは、グループ中の別のサイトにすでに存在しています。
missingvalue	現在の値は存在しません。
paramtype	優先順位グループの新しい値のデータ型が正しくありません。
notquiesced	レプリケート・オブジェクト・グループは休止していません。

DBMS_REPCAT.ALTER_SITE_PRIORITY_SITE

目的

指定された優先順位レベルに関連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でコールしなければなりません。 5-25 ページの「優先順位に対応付けられたサイトの変更」を参照してください。

構文

ALTER_SITE_PRIORITY_SITE プロシージャのパラメータは、表 9-93 を参照してください。例外は、表 9-94 を参照してください。このプロシージャの構文は、次のとおりです。

```
DBMS_REPCAT.ALTER_SITE_PRIORITY_SITE(  
    gname      IN   VARCHAR2,  
    name       IN   VARCHAR2  
    old_site   IN   VARCHAR2,  
    new_site   IN   VARCHAR2)
```

表 9-93 ALTER_SITE_PRIORITY_SITE のパラメータ

パラメータ	説明
gname	サイト優先順位グループが対応付けられているレプリケート・オブジェクト・グループの名前。
name	メンバーを変更するサイト優先順位グループの名前。
old_site	優先順位レベルから分離するサイトの、現行のグローバルなデータベース名。
new_site	現在の優先順位レベルと対応した、新しいグローバルなデータベース名。

表 9-94 ALTER_SITE_PRIORITY_SITE 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	指定されたレプリケート・オブジェクト・グループは存在しません。
missingpriority	指定されたサイト優先順位グループは存在しません。
missingvalue	現在のサイトはグループ・メンバーになっていません。
notquiesced	レプリケート・オブジェクト・グループは休止していません。

DBMS_REPCAT.ALTER_SNAPSHOT_PROPAGATION

目的

指定したオブジェクト・グループの伝播方法を、現行のスナップショット・サイトで変更します。このプロシージャは、遅延トランザクション・キューをスナップショット・サイトで送信し、スナップショットの実表をロックし、トリガーおよびこれに関連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、スナップショット・サイトからコールしなければなりません。

構文

ALTER_SNAPSHOT_PROPAGATION プロシージャのパラメータは、表 9-95 を参照してください。例外は、表 9-96 を参照してください。このプロシージャの構文は、次のとおりです。

```
DBMS_REPCAT.ALTER_SNAPSHOT_PROPAGATION(  
    gname                IN  VARCHAR2,  
    propagation_mode     IN  VARCHAR2,  
    comment              IN  VARCHAR2    := '')
```

表 9-95 ALTER_SNAPSHOT_PROPAGATION のパラメータ

パラメータ	説明
gname	伝播モードを変更するオブジェクト・グループの名前。
propagation_mode	現行のスナップショット・サイトからの変更が、対応付けられたマスター・サイトに伝播される方法。指定できる値は SYNCHRONOUS および ASYNCHRONOUS です。
comment	このコメントは、RepProp ビューに追加されます。

表 9-96 ALTER_SNAPSHOT_PROPAGATION の例外

例外	説明
notcompat	リリース 7.3.0 以降のモードで稼動するデータベースだけがこのプロシージャを使用できます。
missingrepgroup	指定されたレプリケート・オブジェクト・グループは存在しません。
typefailure	伝播モードが誤って指定されました。
nonsnapshot	現行のサイトは、指定されたオブジェクト・グループのスナップショット・サイトではありません。
commfailure	マスターと接続できません。

DBMS_REPCAT.CANCEL_STATISTICS

目的

表に関する更新競合および一意性競合、削除競合の解消が成功したものについての統計の収集を停止します。

構文

CANCEL_STATISTICS プロシージャのパラメータは、表 9-97 を参照してください。例外は、表 9-98 を参照してください。このプロシージャの構文は、次のとおりです。

```
DBMS_REPCAT.CANCEL_STATISTICS(  
    sname      IN    VARCHAR2,  
    oname      IN    VARCHAR2)
```

表 9-97 CANCEL_STATISTICS のパラメータ

パラメータ	説明
sname	表が置かれているスキーマの名前。
oname	競合解消統計を収集する表の名前。

表 9-98 CANCEL_STATISTICS の例外

例外	説明
missingschema	指定されたスキーマが存在しません。
missingobject	指定された表は存在しません。
statnotreg	指定された表は、現在、統計収集されていません。

DBMS_REPCAT.COMMENT_ON_COLUMN_GROUP

目的

列グループの RepColumn_Group ビューのコメント・フィールドを更新します。このコメントは、DBMS_REPCAT.GENERATE_REPLICATION_SUPPORT を次にコールするまで、どのマスター・サイトにも追加されません。

構文

COMMENT_ON_COLUMN_GROUP プロシージャのパラメータは、表 9-99 を参照してください。例外は、表 9-100 を参照してください。このプロシージャの構文は、次のとおりです。

```
DBMS_REPCAT.COMMENT_ON_COLUMN_GROUP(  
    sname      IN    VARCHAR2,  
    oname      IN    VARCHAR2,
```

```
column_group IN      VARCHAR2,
comment      IN      VARCHAR2)
```

表 9-99 COMMENT_ON_COLUMN_GROUP のパラメータ

パラメータ	説明
sname	オブジェクトが置かれているスキーマの名前。
oname	列グループに対応するレプリケート表の名前。
column_group	列グループの名前。
comment	RepColumn_Group ビューの GROUP_COMMENT フィールドに入れる、更新済みコメントのテキスト。

表 9-100 COMMENT_ON_COLUMN_GROUP 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	オブジェクトが見つかりません。

DBMS_REPCAT.COMMENT_ON_PRIORITY_GROUP/

DBMS_REPCAT.COMMENT_ON_SITE_PRIORITY

目的

COMMENT_ON_PRIORITY_GROUP は、REPPRIORITY_GROUP ビューの中のコメント・フィールドを更新します。このコメントは、DBMS_REPCAT.GENERATE_REPLICATION_SUPPORT を次にコールするまで、どのマスター・サイトにも追加されません。

COMMENT_ON_SITE_PRIORITY は、REPPRIORITY_GROUP ビューの中のコメント・フィールドを更新します。このプロシージャは COMMENT_ON_COLUMN_GROUP プロシージャの Wrapper で、便宜上の目的だけで提供されています。このプロシージャは、マスター定義サイトで発行しなければなりません。

構文

COMMENT_ON_PRIORITY_GROUP および COMMENT_ON_SITE_PRIORITY プロシージャのパラメータは、表 9-101 を参照してください。例外は、表 9-102 を参照してください。

COMMENT_ON_PRIORITY_GROUP プロシージャの構文は、次のとおりです。

```
DBMS_REPCAT.COMMENT_ON_PRIORITY_GROUP(
    gname      IN      VARCHAR2,
```

DBMS_REPCAT.COMMENT_ON_REPGROUP

```
pgroup      IN   VARCHAR2,  
comment      IN   VARCHAR2)
```

COMMENT_ON_SITE_PRIORITY プロシージャの構文は、次のとおりです。

```
DBMS_REPCAT.COMMENT_ON_SITE_PRIORITY(  
    gname      IN   VARCHAR2,  
    name       IN   VARCHAR2,  
    comment    IN   VARCHAR2)
```

表 9-101 COMMENT_ON_PRIORITY_GROUP および COMMENT_ON_SITE_PRIORITY のパラメータ

パラメータ	説明
gname	レプリケート・オブジェクト・グループの名前。
pgroup/name	優先順位またはサイト優先順位グループの名前。
comment	RepPriority_Group ビューの PRIORITY_COMMENT フィールドに入れる、更新されたコメントのテキスト。

表 9-102 COMMENT_ON_PRIORITY_GROUP および 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	指定されたレプリケート・オブジェクト・グループは存在しません。
missingprioritygroup	指定された優先順位グループは存在しません。

DBMS_REPCAT.COMMENT_ON_REPGROUP

目的

レプリケート・オブジェクト・グループに対する REPGROUP ビューの中のコメント・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で発行しなければなりません。

構文

COMMENT_ON_REPGROUP プロシージャのパラメータは、表 9-103 を参照してください。例外は、表 9-104 を参照してください。このプロシージャの構文は、次のとおりです。

```
DBMS_REPCAT.COMMENT_ON_REPGROUP(  
    gname      IN   VARCHAR2,
```



```
comment    IN    VARCHAR2)
```

表 9-103 COMMENT_ON_REPGROUP のパラメータ

パラメータ	説明
gname	コメントを付けるオブジェクト・グループの名前。
comment	RepGroup ビューの SCHEMA_COMMENT フィールドに入れる更新されたコメント。

表 9-104 COMMENT_ON_REPGROUP 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	アクセスできないマスター・サイトが少なくとも 1 つあります。

DBMS_REPCAT.COMMENT_ON_REPSITES

目的

レプリケート・サイトで RepSite ビューのコメント・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で発行しなければなりません。

構文

COMMENT_ON_REPSITES プロシージャのパラメータは、表 9-105 を参照してください。例外は、表 9-106 を参照してください。このプロシージャの構文は、次のとおりです。

```
DBMS_REPCAT.COMMENT_ON_REPSITES(
    gname      IN    VARCHAR2,
    [ master   IN    VARCHAR, ]
    comment    IN    VARCHAR2)
```

表 9-105 COMMENT_ON_REPSITES のパラメータ

パラメータ	説明
gname	オブジェクト・グループの名前。データベースが複数のレプリケート環境のマスター・サイトである場合は、このパラメータによって混乱を防止できます。
master	このパラメータはオプションです。コメントを付けるマスター・サイトの、省略のない形のデータベース名。スナップショット・サイトのコメントを更新する場合は、このパラメータを省略します。
comment	RepSites ビューの MASTER_COMMENT フィールドに入れる更新されたコメントのテキスト。

表 9-106 COMMENT_ON_REPSITES 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	アクセスできないマスター・サイトが少なくとも 1 つあります。

DBMS_REPCAT.COMMENT_ON_REPOBJECT

目的

レプリケート・オブジェクトに対する RepObject ビュー中のコメント・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で発行しなければなりません。

構文

COMMENT_ON_REPOBJECT プロシージャのパラメータは、表 9-107 を参照してください。例外は、表 9-108 を参照してください。このプロシージャの構文は、次のとおりです。

```
DBMS_REPCAT.COMMENT_ON_REPOBJECT(  
    sname      IN   VARCHAR2,  
    oname      IN   VARCHAR2,  
    type       IN   VARCHAR2,  
    comment    IN   VARCHAR2)
```

表 9-107 COMMENT_ON_REPOBJECT のパラメータ

パラメータ	説明
sname	オブジェクトが置かれているスキーマの名前。
oname	コメントを付けるオブジェクトの名前。
type	オブジェクトの型。
comment	RepObject ビューのOBJECT_COMMENTフィールドに入れる、更新されたコメントのテキスト。

表 9-108 COMMENT_ON_REPOBJECT 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	指定された型パラメータはサポートされていません。
commfailure	アクセスできないマスター・サイトが少なくとも 1 つあります。

DBMS_REPCAT.COMMENT_ON_conflicttype_RESOLUTION

目的

競合解消ルーチンに対する RepResolution ビューの中のコメント・フィールドを更新します。コールする必要のあるプロシージャは、ルーチンが解消する競合のタイプによって決まります。これらのプロシージャは、マスター定義サイトで発行しなければなりません。

競合のタイプ	プロシージャ名
更新	COMMENT_ON_UPDATE_RESOLUTION
一意	COMMENT_ON_UNIQUE_RESOLUTION
削除	COMMENT_ON_DELETE_RESOLUTION

コメントは、DBMS_REPCAT.GENERATE_REPLICATION_SUPPORTを次にコールするまで、どのマスター・サイトにも追加されません。

構文

COMMENT_ON_conflictype_RESOLUTION プロシージャのパラメータは、表 9-109 を参照してください。例外は、表 9-110 を参照してください。

COMMENT_ON_UPDATE_RESOLUTION プロシージャの構文は、次のとおりです。

```
DBMS_REPCAT.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 プロシージャの構文は、次のとおりです。

```
DBMS_REPCAT.COMMENT_ON_UNIQUE_RESOLUTION(  
    sname                IN    VARCHAR2,  
    oname                IN    VARCHAR2,  
    constraint_name      IN    VARCHAR2,  
    sequence_no          IN    NUMBER,  
    comment              IN    VARCHAR2)
```

COMMENT_ON_DELETE_RESOLUTION プロシージャの構文は、次のとおりです。

```
DBMS_REPCAT.COMMENT_ON_DELETE_RESOLUTION(  
    sname                IN    VARCHAR2,  
    oname                IN    VARCHAR2,  
    sequence_no          IN    NUMBER,  
    comment              IN    VARCHAR2)
```

表 9-109 COMMENT_ON_conflictype_RESOLUTION のパラメータ

パラメータ	説明
sname	スキーマの名前。
oname	競合解消ルーチンに対応するレプリケート表の名前。
column_group	更新競合解消ルーチンに対応する列グループの名前。
constraint_name	一意競合解消ルーチンに対応する一意制約の名前。
sequence_no	競合解消プロシージャの順序番号。
comment	RepResolution ビューの RESOLUTION_COMMENT フィールドに入れる、更新されたコメントのテキスト。

表 9-110 COMMENT_ON_conflictype_RESOLUTION 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	SEQUENCE_NO または COLUMN_GROUP は登録されていません。

DBMS_REPCAT.CREATE_MASTER_REPGROUP

目的

新規、空、休止状態のマスター・レプリケーション・オブジェクト・グループを作成します。

構文

CREATE_MASTER_REPGROUP プロシージャのパラメータは、表 9-111 を参照してください。例外は、表 9-112 を参照してください。このプロシージャの構文は、次のとおりです。

```
DBMS_REPCAT.CREATE_MASTER_REPGROUP(  
    gname          IN    VARCHAR2,  
    group_comment  IN    VARCHAR2    := '',  
    master_comment IN    VARCHAR2    := ''),  
qualifier         IN    VARCHAR2    := '')
```

表 9-111 CREATE_MASTER_REPGROUP のパラメータ

パラメータ	説明
gname	作成するオブジェクト・グループの名前。
group_comment	このコメントは RepCat ビューに追加されます。
master_comment	このコメントは RepGroup ビューに追加されます。
qualifier	オブジェクト・グループの接続修飾子。必ず @ 記号を使用してください。3-14 ページの「マスター・グループの接続修飾子の使用方法」にある例を参照してください。

表 9-112 CREATE_MASTER_REPGROUP の例外

例外	説明
duplicaterepgroup	オブジェクト・グループはすでに存在しています。
ddlfailure	rep\$what_am_i パッケージまたはパッケージ本体を作成する際に問題があります。
norepopt	拡張レプリケーション・オプションはインストールされません。
missingrepggrp	オブジェクト・グループ名を指定していません。
qualifiertoolong	接続修飾子が長すぎます。

DBMS_REPCAT.CREATE_MASTER_REPOBJECT

目的

オブジェクトをレプリケート・オブジェクトとして定義します。

構文

CREATE_MASTER_REPOBJECT プロシージャのパラメータは、表 9-113 を参照してください。例外は、表 9-114 を参照してください。このプロシージャの構文は、次のとおりです。

```
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         := ''

```

注意：スキーマを指定しないでDDLを指定すると、レプリケーション管理者のスキーマがデフォルトで指定されます。レプリケーション管理者のスキーマを使用しない場合は、必ずスキーマを指定してください。

表 9-113 CREATE_MASTER_REPOBJECT 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ているスキーマの名前。
oname	レプリケートするオブジェクトの名前。DDL_TEXT が NULL である場合、このオブジェクトが指定されたスキーマにすでに存在している必要があります。一意性を確保するために、表名は最大で 27 バイト、パッケージは 24 バイト以内にしてください。
type	レプリケートするオブジェクトの型です。サポートされている型は、TABLE および INDEX、SYNONYM、TRIGGER、VIEW、PROCEDURE、FUNCTION、PACKAGE、PACKAGE BODY です。
use_existing_object	現行マスター・サイトにおいて同じ型と内容のオブジェクトを再使用する場合、TRUE と指定します。これらの変更の適用方法の詳細は、表 9-116 を参照してください。
ddl_text	オブジェクトが、マスター定義サイトにまだ存在していない場合、このオブジェクトを作成するのに必要なDDLテキストを指定します。PL/SQL パッケージおよびパッケージ本体、プロシージャ、ファンクションの後には、セミコロンを付ける必要があります。SQL 文の場合は、後にセミコロンを付ける必要はありません。Oracle はこのDDLを適用する前に、その構文を解析しません。したがって、DDLテキストの中に、作成するオブジェクトのスキーマ名とオブジェクト名を含める必要があります。
comment	このコメントが、RepObject ビューの OBJECT_COMMENT フィールドに追加されます。
retry	以前に作成できなかったオブジェクトの再作成を Oracle で試行する場合は、TRUE と指定します。エラーが一時的なものであったり、すでに訂正されているならば（以前には資源が十分なかったなどで）RETRYを使用します。RETRYがTRUEの場合、Oracle はオブジェクト状態が VALID ではないマスター・サイトにおいてだけ、オブジェクトを作成します。

表 9-113 CREATE_MASTER_REPOBJECT のパラメータ

パラメータ	説明
copy_rows	新規レプリケート・オブジェクトの初期の内容が、マスター定義サイトにおけるオブジェクトの内容と一致するようにする場合、TRUE を指定します。詳細は、表 9-116 を参照してください。
gname	作成するレプリケート・オブジェクトを含むオブジェクト・グループの名前。何も指定しないと、デフォルト・オブジェクト・グループ名としてスキーマ名が使用されます。

表 9-114 CREATE_MASTER_REPOBJECT 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	レプリケート・オブジェクト・グループは、まだ中断されていません。
duplicateobject	指定されたオブジェクトがレプリケート・オブジェクト・グループにすでに存在していて retry が FALSE であるか、名前競合が発生しています。
missingobject	SNAME と ONAME で指定されたオブジェクトが存在せず、適切な DDL が指定されていません。
typefailure	指定された型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で DDL は成功しませんでした。
commfailure	アクセスできないマスター・サイトが少なくとも 1 つあります。
notcompat	すべてのリモート・マスターがリリース 7.3 の互換モードではありません。

表 9-115 マスター・サイトでのオブジェクト作成

オブジェクトはすでに			
存在しますか？	COPY_ROWS	USE_EXISTING_OBJECTS	結果
はい	TRUE	TRUE	オブジェクトが合致しない場合、 duplicatedobject のメッセージ。表の場合、マスター定義サイトのデータを使用します。
はい	FALSE	TRUE	オブジェクトが合致しない場合、 duplicatedobject のメッセージ。表の場合、内容が同じであることを管理者が確認する必要があります。
はい	TRUE/FALSE	FALSE	duplicatedobject のメッセージ
いいえ	TRUE	TRUE/FALSE	オブジェクトが作成されます。表には、マスター定義サイトのデータが入れられます。
いいえ	FALSE	TRUE/FALSE	オブジェクトが作成されます。DBA では、表を移入して、すべてのサイトでの一貫性を確実にする必要があります。

DBMS_REPCAT.CREATE_SNAPSHOT_REPGROUP

目的

ローカル・データベースに、新規で空のスナップショット・レプリケーション・オブジェクト・グループを作成します。

構文

CREATE_SNAPSHOT_REPGROUP プロシージャのパラメータは、表 9-116 を参照してください。例外は、表 9-117 を参照してください。このプロシージャの構文は、次のとおりです。

```
DBMS_REPCAT.CREATE_SNAPSHOT_REPGROUP(  
    gname          IN   VARCHAR2,  
    master         IN   VARCHAR2,  
    comment        IN   VARCHAR2      := ' ',  
    propagation_mode IN   VARCHAR2      := 'ASYNCHRONOUS')
```

注意 : CREATE_SNAPSHOT_REPGROUP では、自動的に DBMS_REPCAT.REGISTER_SNAPSHOT_REPGROUP をコールします。ただし、登録中にエラーが発生しても無視します。

表 9-116 CREATE_SNAPSHOT_REPGROUP のパラメータ

パラメータ	説明
gname	レプリケート・オブジェクト・グループの名前。このオブジェクト・グループは、指定したマスター・サイトに存在していなければなりません。
master	マスターとして使用するためのレプリケート環境のデータベースで省略のない形のデータベース名。
comment	このコメントは RepCat ビューの GROUP_COMMENT フィールドに追加されます。
propagation_mode	オブジェクト・グループにあるすべての更新可能スナップショットの伝播方法。指定できる値は、SYNCHRONOUS および ASYNCHRONOUS です。

表 9-117 CREATE_SNAPSHOT_REPGROUP の例外

例外	説明
duplicaterepgroup	オブジェクト・グループは、すでに起動サイトに存在しています。
nonmaster	指定された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。
norepopt	拡張レプリケーション・オプションはインストールされません。
typefailure	伝播モードが誤って指定されました。
missingrepgroup	レプリケート・オブジェクト・グループがマスター・サイトにない場合。
notcompatible	リリース 7.3 と互換性が必要です。

DBMS_REPCAT.CREATE_SNAPSHOT_REPOBJECT

目的

レプリケート・オブジェクトをスナップショット・サイトに追加します。

構文

CREATE_SNAPSHOT_REPOBJECT プロシージャのパラメータは、表 9-118 を参照してください。例外は、表 9-119 を参照してください。このプロシージャの構文は、次のとおりです。

```
DBMS_REPCAT.CREATE_SNAPSHOT_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)
```

注意：スキーマを指定しないで DDL を指定すると、レプリケーション管理者のスキーマがデフォルトで指定されます。レプリケーション管理者のスキーマを使用しない場合は、必ずスキーマを指定してください。

表 9-118 CREATE_SNAPSHOT_REPOBJECT のパラメータ

パラメータ	説明
sname	オブジェクトが置かれているスキーマの名前。
oname	レプリケート・スナップショット・オブジェクト・グループに追加するオブジェクトの名前。ONAMEは対応するマスター・サイトに存在する必要があります。
type	レプリケートするオブジェクトの型。スナップショット・サイトでサポートされる型は PACKAGE および PACKAGE BODY、PROCEDURE、FUNCTION、SNAPSHOT、SYNONYM、VIEW です。
ddl_text	SNAPSHOT 型のオブジェクトの場合は、オブジェクトの作成に必要なDDLテキスト。その他の型の場合は、デフォルト（空の文字列）を使用してください。同じ名前のスナップショットがすでに存在する場合、Oracle はDDLを無視し、既存のスナップショットをレプリケート・オブジェクトとして登録します。スナップショットのマスター表が、このスキーマに指定されたマスター・サイトのレプリケート・オブジェクト・グループに存在しない場合、Oracle では、missingobject エラーを呼び出します。
comment	このコメントは 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・オブジェクト・グループの名前。何も指定しないと、デフォルト・グループ名としてスキーマ名が使用されます。
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。
min_communication	任意のマスター・サイトが Oracle7 リリース 7.3 を実行している場合、FALSE に設定します。新しい値および古い値の伝播を最小化するときは、TRUE に設定します。デフォルトは TRUE です。詳細は、5-34 ページの「更新の競合解消のためのデータ伝播の最小化」を参照してください。

表 9-119 CREATE_SNAPSHOT_REPOBJECT の例外

例外	説明
nonsnapshot	起動サイトはスナップショット・サイトではありません。
nonmaster	このマスターは、マスター・サイトではありません。
missingobject	指定されたオブジェクトは、マスターのレプリケート・オブジェクト・グループに存在しません。
duplicateobject	指定されたオブジェクトはすでに別の形で存在します。
typefailure	この型は許容されていません。
ddlfailure	DDL は成功しませんでした。
commfailure	マスター・サイトにアクセスできません。
missingschema	この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。
badsnapddl	DDLが実行されましたが、スナップショットは存在しません。
onlyonesnap	マスター表のスナップショットは1つだけ作成できます。
badsnapname	スナップショット実表がマスター表と違います。
missingrepgroup	レプリケート・オブジェクト・グループは存在しません。

DBMS_REPCAT.DEFINE_COLUMN_GROUP

目的

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でコールしなければなりません。詳細は、5-6ページの「更新の競合解消と列グループ」を参照してください。

構文

DEFINE_COLUMN_GROUP プロシージャのパラメータは、表 9-120 を参照してください。例外は、表 9-121 を参照してください。このプロシージャの構文は、次のとおりです。

```
DBMS_REPCAT.DEFINE_COLUMN_GROUP(  
    sname          IN   VARCHAR2,  
    oname          IN   VARCHAR2,  
    column_group   IN   VARCHAR2,  
    comment        IN   VARCHAR2 := NULL)
```

表 9-120 DEFINE_COLUMN_GROUP 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ているスキーマの名前。
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・テキストは RepColumnGroup ビューに表示されます。

表 9-121 DEFINE_COLUMN_GROUP のパラメータ

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すでに表に存在しています。
notquiesced	指定の表が属するオブジェクト・グループは休止していません。

DBMS_REPCAT.DEFINE_PRIORITY_GROUP

目的

レプリケート・オブジェクト・グループに新規の優先順位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でコールしなければなりません。5-17 ページの「優先順位グループおよびサイト優先順位」を参照してください。

構文

DEFINE_PRIORITY_GROUP プロシージャのパラメータは、表 9-122 を参照してください。例外は、表 9-123 を参照してください。このプロシージャの構文は、次のとおりです。

```
DBMS_REPCAT.DEFINE_PRIORITY_GROUP(  
    gname          IN    VARCHAR2,  
    pgroup         IN    VARCHAR2,  
    datatype       IN    VARCHAR2,  
    fixed_length   IN    INTEGER := NULL,  
    comment        IN    VARCHAR2 := NULL)
```

表 9-122 DEFINE_PRIORITY_GROUP のパラメータ

パラメータ	説明
gname	優先順位グループを作成しているレプリケート・オブジェクト・グループの名前。
pgroup	作成する優先順位グループの名前。
datatype	優先順位グループ・メンバーのデータ型。サポートされているデータ型は、CHAR および VARCHAR2、NUMBER、DATE、RAW、NCHAR、NVARCHAR2 です。
fixed_length	CHARデータ型の場合は、列の長さを必ず指定してください。他のすべての型では、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RepPriorityビューに追加されます。

表 9-123 DEFINE_PRIORITY_GROUP 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	指定されたレプリケート・オブジェクト・グループは存在しません。
duplicateprioritygroup	指定された優先順位グループは、すでにレプリケート・オブジェクト・グループに存在しています。
typefailure	指定されたデータ型はサポートされていません。
notquiesced	レプリケート・オブジェクト・グループは休止していません。

DBMS_REPCAT.DEFINE_SITE_PRIORITY

目的

レプリケート・オブジェクト・グループに新規のサイト優先順位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でコールしなければなりません。5-17 ページの「優先順位グループおよびサイト優先順位」を参照してください。

構文

DEFINE_SITE_PRIORITY プロシージャのパラメータは、表 9-124 を参照してください。例外は、表 9-125 を参照してください。このプロシージャの構文は、次のとおりです。

```
DBMS_REPCAT.DEFINE_SITE_PRIORITY(
    gname          IN   VARCHAR2,
    name           IN   VARCHAR2,
```

```
comment          IN    VARCHAR2 := NULL)
```

表 9-124 DEFINE_SITE_PRIORITY のパラメータ

パラメータ	説明
gname	サイト優先順位グループを作成しているレプリケート・オブジェクト・グループの名前。
name	作成するサイト優先順位グループの名前。
comment	このユーザー・コメントはRepPriorityビューに追加されます。

表 9-125 DEFINE_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	指定されたレプリケート・オブジェクト・グループは存在しません。
duplicateprioritygroup	指定されたサイト優先順位グループは、すでにレプリケート・オブジェクト・グループに存在しています。
notquiesced	レプリケート・オブジェクト・グループは休止していません。

DBMS_REPCAT.DO_DEFERRED_REPCAT_ADMIN

目的

現行マスター・サイトで指定されたレプリケート・オブジェクト・グループ、またはすべてのマスター・サイトに（ジョブ・キューを利用して）未処理のローカル遅延管理プロシージャを実行します。

注意：DO_DEFERRED_REPCAT_ADMIN によって実行されるのは、DO_DEFERRED_REPCAT_ADMIN をコールした接続中のユーザーによって送られた管理要求だけです。その他のユーザーが送信した要求は無視されます。

構文

DO_DEFERRED_REPCAT_ADMIN プロシージャのパラメータは、表 9-126 を参照してください。例外は、表 9-127 を参照してください。このプロシージャの構文は、次のとおりです。

```
DBMS_REPCAT.DO_DEFERRED_REPCAT_ADMIN(
    gname          IN    VARCHAR2,
    all_sites      IN    BOOLEAN := FALSE)
```


表 9-126 DO_DEFERRED_REPCAT_ADMIN のパラメータ

パラメータ	説明
gname	レプリケート・オブジェクト・グループの名前。
all_sites	ALL_SITES が TRUE の場合、ジョブを使用してマスターごとにローカル管理プロシーダを実行します。

表 9-127 DO_DEFERRED_REPCAT_ADMIN 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	all_sites を TRUE と指定したのに、アクセスできないマスター・サイトが少なくとも 1 つあります。

DBMS_REPCAT.DROP_COLUMN_GROUP

目的

列グループを削除します。このプロシーダは、マスター定義サイトでコールしなければなりません。5-18 ページの「更新の競合解消のための優先順位グループの使用」を参照してください。

構文

DROP_COLUMN_GROUP プロシーダのパラメータは、表 9-128 を参照してください。例外は、表 9-129 を参照してください。このプロシーダの構文は、次のとおりです。

```
DBMS_REPCAT.DROP_COLUMN_GROUP(
    sname          IN   VARCHAR2,
    oname          IN   VARCHAR2,
    column_group   IN   VARCHAR2)
```

表 9-128 DROP_COLUMN_GROUP 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ているスキーマの名前。
oname	削除する列グループのレプリケート表の名前。
column_group	削除する列グループの名前。

表 9-129 DROP_COLUMN_GROUP 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検出および解消で使用中です。
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	表が属するレプリケート・オブジェクト・グループは休止していません。

DBMS_REPCAT.DROP_GROUPED_COLUMN

目的

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でコールしなければなりません。詳細は、5-12 ページの「列グループの列の追加および削除」を参照してください。

構文

DROP_GROUPED_COLUMN プロシージャのパラメータは、表 9-130 を参照してください。例外は、表 9-131 を参照してください。このプロシージャの構文は、次のとおりです。

```
DBMS_REPCAT.DROP_GROUPED_COLUMN(
    sname          IN    VARCHAR2,
    oname          IN    VARCHAR2,
    column_group   IN    VARCHAR2,
    list_of_column_names IN  VARCHAR2 | DBMS_REPCAT.VARCHAR2S)
```

表 9-130 DROP_GROUPED_COLUMN 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ているスキーマの名前。
oname	列グループが置かれているレプリケート表の名前。
column_group	メンバーを削除する列グループの名前。
list_of_column_names	指定された列グループから削除する列の名前です。これは、カンマで区切られた列名のリスト、または PL/SQL 表の列名のどちらでもかまいません。PL/SQL 表は dbms_repat.varchar2 型にしてください。

表 9-131 DROP_GROUPED_COLUMN 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	表が属するレプリケート・オブジェクト・グループは休止していません。

DBMS_REPCAT.DROP_MASTER_REPGROUP

目的

現行サイトからレプリケート・オブジェクト・グループを削除します。マスター定義サイトを含めて、レプリケート・オブジェクト・グループをすべてのマスター・サイトから削除するには、マスター定義サイトで最後の引数を TRUE に設定して、このプロシージャをコールしてください。

構文

DROP_MASTER_REPGROUP プロシージャのパラメータは、表 9-132 を参照してください。例外は、表 9-133 を参照してください。このプロシージャの構文は、次のとおりです。

```
DBMS_REPCAT.DROP_MASTER_REPGROUP(  
    gname          IN VARCHAR2,  
    drop_contents  IN BOOLEAN    := FALSE,  
    all_sites      IN BOOLEAN    := FALSE)
```

表 9-132 DROP_MASTER_REPGROUP のパラメータ

パラメータ	説明
gname	現行マスター・サイトから削除するレプリケート・オブジェクト・グループの名前。
drop_contents	デフォルトの場合、マスター・サイトでオブジェクト・グループを削除すると、すべてのオブジェクトがスキーマに残ります。これらのオブジェクトはレプリケートされなくなるだけです（つまり、オブジェクト・グループのレプリケート・オブジェクトと他のマスター・サイト間で変更の受渡しは行われなくなります）。この引数を TRUE に設定すると、レプリケート・オブジェクト・グループのすべてのレプリケート・オブジェクトは、これらに対応付けられたスキーマから削除されます。
all_sites	ALL_SITES が TRUE で起動サイトがマスター定義サイトの場合、プロシージャは同期的にすべてのマスターに要求をマルチキャストします。この場合、マスター定義サイトではすぐに、他のすべてのマスター・サイトでは遅れて実行されます。

表 9-133 DROP_MASTER_REPGROUP 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なく、ALL_SITES は TRUE になっています。
commfailure	アクセスできないマスター・サイトが少なくとも 1 つあり、ALL_SITES は TRUE になっています。
fullqueue	遅延 RPC キューに、レプリケート・オブジェクト・グループの項目があります。
masternotremoved	マスターで masterdef を認識していません。

DBMS_REPCAT.DROP_MASTER_REPOBJECT

目的

レプリケート・オブジェクト・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でコールしなければなりません。

構文

DROP_MASTER_REPOBJECT プロシージャのパラメータは、表 9-134 を参照してください。例外は、表 9-135 を参照してください。このプロシージャの構文は、次のとおりです。

```
DBMS_REPCAT.DROP_MASTER_REPOBJECT(  
    sname          IN   VARCHAR2,  
    oname          IN   VARCHAR2,  
    type           IN   VARCHAR2,  
    drop_objects   IN   BOOLEAN    := FALSE)
```

表 9-134 DROP_MASTER_REPOBJECT のパラメータ

パラメータ	説明
sname	オブジェクトが置かれているスキーマの名前。
oname	レプリケート・オブジェクト・グループから削除するオブジェクトの名前。
type	削除するオブジェクトの型。
drop_objects	デフォルトの場合、オブジェクトはスキーマには残りますが、レプリケート・オブジェクト・グループからは削除されます（つまり、オブジェクトへの変更は、他のマスター・サイトおよびスナップショット・サイトにレプリケートされなくなります）。レプリケート環境からオブジェクトを完全に削除するには、この引数を TRUE に設定します。

表 9-135 DROP_MASTER_REPOBJECT 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	指定された型パラメータはサポートされていません。
commfailure	アクセスできないマスター・サイトが少なくとも 1 つあります。

DBMS_REPCAT.DROP_PRIORITY

目的

優先順位レベルごとに優先順位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でコールしなければなりません。5-22 ページの「優先順位によるメンバーの削除」を参照してください。

構文

DROP_PRIORITY プロシージャのパラメータは、表 9-136 を参照してください。例外は、表 9-137 を参照してください。このプロシージャの構文は、次のとおりです。

DBMS_REPCAT.DROP_PRIORITY_GROUP

```
DBMS_REPCAT.DROP_PRIORITY(  
    gname          IN    VARCHAR2,  
    pgroup         IN    VARCHAR2,  
    priority_num   IN    NUMBER)
```

表 9-136 DROP_PRIORITY のパラメータ

パラメータ	説明
gname	優先順位グループが対応付けられているレプリケート・オブジェクト・グループの名前。
pgroup	削除するメンバーを含む優先順位グループの名前。
priority_num	グループから削除する優先順位グループ・メンバーの優先順位レベルです。

表 9-137 DROP_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	指定されたレプリケート・オブジェクト・グループは存在しません。
missingprioritygroup	指定された優先順位グループは存在しません。
notquiesced	レプリケート・オブジェクト・グループは休止していません。

DBMS_REPCAT.DROP_PRIORITY_GROUP

目的

指定されたレプリケート・オブジェクト・グループの優先順位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でコールしなければなりません。5-23 ページの「優先順位グループの削除」を参照してください。

構文

DROP_PRIORITY_GROUP プロシージャのパラメータは、表 9-138 を参照してください。例外は、表 9-139 を参照してください。このプロシージャの構文は、次のとおりです。

```
DBMS_REPCAT.DROP_PRIORITY_GROUP(  
    gname          IN    VARCHAR2,  
    pgroup         IN    VARCHAR2)
```

表 9-138 DROP_PRIORITY_GROUP のパラメータ

パラメータ	説明
gname	優先順位グループが対応付けられているレプリケート・オブジェクト・グループの名前。
pgroup	削除する優先順位グループの名前。

表 9-139 DROP_PRIORITY_GROUP 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	指定されたレプリケート・オブジェクト・グループは存在しません。
referenced	指定された優先順位グループは競合の解消に使用中です。
notquiesced	指定されたレプリケート・オブジェクト・グループは休止していません。

DBMS_REPCAT.DROP_PRIORITY_datatype

目的

値ごとに優先順位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でコールしなければなりません。コールする必要があるプロシージャは、「優先順位」列のデータ型によって決まります。5-22 ページの「値によるメンバーの削除」を参照してください。

構文

DROP_PRIORITY_datatype プロシージャのパラメータは、表 9-140 を参照してください。例外は、表 9-141 を参照してください。DROP_PRIORITY_datatype プロシージャの構文は、次のとおりです。

```
DBMS_REPCAT.DROP_PRIORITY_datatype(
    gname      IN   VARCHAR2,
    pgroup     IN   VARCHAR2,
    value      IN   datatype)
```

datatype には、次の引数を指定します。

```
{ NUMBER
| VARCHAR2
| CHAR
```

| DATE
| RAW
| NCHAR
| NVARCHAR2 }

表 9-140 DROP_PRIORITY_datatype のパラメータ

パラメータ	説明
gname	優先順位グループが対応付けられているレプリケート・オブジェクト・グループの名前。
pgroup	削除するメンバーを含む優先順位グループの名前。
value	グループから削除する優先順位グループ・メンバーの値。

表 9-141 DROP_PRIORITY_datatype 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	指定されたレプリケート・オブジェクト・グループは存在しません。
missingprioritygroup	指定された優先順位グループは存在しません。
paramtype, typefailure	この値に指定されている優先順位グループのデータ型は間違っています。
notquiesced	指定されたレプリケート・オブジェクト・グループは休止していません。

DBMS_REPCAT.DROP_SITE_PRIORITY

目的

指定されたレプリケート・オブジェクト・グループのサイト優先順位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でコールしなければなりません。5-25ページの「サイト優先順位グループの削除」を参照してください。

構文

DROP_SITE_PRIORITY プロシージャのパラメータは、表 9-142 を参照してください。例外は、表 9-143 を参照してください。このプロシージャの構文は、次のとおりです。

```
DBMS_REPCAT.DROP_SITE_PRIORITY(  
    gname      IN   VARCHAR2,  
    name       IN   VARCHAR2)
```


表 9-142 DROP_SITE_PRIORITY のパラメータ

パラメータ	説明
gname	サイト優先順位グループが対応付けられているレプリケート・オブジェクト・グループの名前。
name	削除するサイト優先順位グループの名前。

表 9-143 DROP_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	指定されたレプリケート・オブジェクト・グループは存在しません。
referenced	指定されたサイト優先順位グループは競合の解消で使用中です。
notquiesced	指定されたレプリケート・オブジェクト・グループは休止していません。

DBMS_REPCAT.DROP_SITE_PRIORITY_SITE

目的

サイト優先順位グループから、名前ごとに指定されたサイトを削除します。このプロシージャは、マスター定義サイトでコールしなければなりません。5-25 ページの「サイト名によるサイトの削除」を参照してください。

構文

DROP_SITE_PRIORITY_SITE プロシージャのパラメータは、表 9-144 を参照してください。例外は、表 9-145 を参照してください。このプロシージャの構文は、次のとおりです。

```
DBMS_REPCAT.DROP_SITE_PRIORITY_SITE(  
    gname      IN   VARCHAR2,  
    name       IN   VARCHAR2,  
    site       IN   VARCHAR2)
```

表 9-144 DROP_SITE_PRIORITY_SITE のパラメータ

パラメータ	説明
gname	サイト優先順位グループが対応付けられているレプリケート・オブジェクト・グループの名前。
name	メンバーを削除するサイト優先順位グループの名前。
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。

表 9-145 DROP_SITE_PRIORITY_SITE 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	指定されたレプリケート・オブジェクト・グループは存在しません。
missingpriority	指定されたサイト優先順位グループは存在しません。
missingsite	指定されたサイトが存在しません。
notquiesced	指定されたレプリケート・オブジェクト・グループは休止していません。

DBMS_REPCAT.DROP_SNAPSHOT_REPGROUP

目的

レプリケート環境からスナップショット・サイトを削除します。

構文

DROP_SNAPSHOT_REPGROUP プロシージャのパラメータは、表 9-146 を参照してください。例外は、表 9-147 を参照してください。このプロシージャの構文は、次のとおりです。

```
DBMS_REPCAT.DROP_SNAPSHOT_REPGROUP(  
    gname          IN    VARCHAR2,  
    drop_contents  IN    BOOLEAN    := FALSE)
```

注意 : DBMS_REPCAT.DROP_SNAPSHOT_REPGROUP では、自動的に DBMS_REPCAT.REGISTER_SNAPSHOT_REPGROUP をコールして、スナップショットを登録解除します。ただし、登録解除中にエラーが発生しても無視します。

表 9-146 DROP_SNAPSHOT_REPGROUP のパラメータ

パラメータ	説明
gname	現行スナップショット・サイトから削除するレプリケート・オブジェクト・グループの名前。トリガーやパッケージなど、レプリケーションをサポートするため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	デフォルトの場合、スナップショット・サイトのレプリケート・オブジェクト・グループを削除したとき、オブジェクトはすべてそれらに関連付けられたスキーマに残ります（それらのオブジェクトはレプリケートされなくなるだけです）。この引数を TRUE に設定すると、レプリケート・オブジェクト・グループのすべてのレプリケート・オブジェクトは、それらのスキーマから削除されます。

表 9-147 DROP_SNAPSHOT_REPGROUP の例外

例外	説明
nonsnapshot	起動サイトはスナップショット・サイトではありません。
missrepgroup	指定したオブジェクト・グループは存在しません。

DBMS_REPCAT.DROP_SNAPSHOT_REPOBJECT

目的

スナップショット・サイト上でレプリケート・オブジェクトを削除します。

構文

DROP_SNAPSHOT_REPOBJECT プロシージャのパラメータは、表 9-148 を参照してください。例外は、表 9-149 を参照してください。このプロシージャの構文は、次のとおりです。

```
DBMS_REPCAT.DROP_SNAPSHOT_REPOBJECT(  
    sname          IN    VARCHAR2,  
    oname          IN    VARCHAR2,  
    type           IN    VARCHAR2,  
    drop_objects   IN    BOOLEAN := FALSE)
```

表 9-148 DROP_SNAPSHOT_REPOBJECT のパラメータ

パラメータ	説明
sname	オブジェクトが置かれているスキーマの名前。
oname	レプリケート・オブジェクト・グループから削除するオブジェクトの名前。
type	削除するオブジェクトの型。
drop_objects	デフォルトでは、オブジェクトは、関連付けられたスキーマには残りますが、関連付けられたオブジェクト・グループからは削除されます。現行スナップショット・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、この引数を TRUE に設定します。

表 9-149 DROP_SNAPSHOT_REPOBJECT の例外

例外	説明
nonsnapshot	起動サイトはスナップショット・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	指定された型パラメータはサポートされていません。

DBMS_REPCAT.DROP_conflicttype_RESOLUTION

目的

更新または削除、一意の競合解消ルーチンを中断します。これらのプロシージャは、マスター定義サイトでコールする必要があります。コールする必要のあるプロシージャは、ルーチンが解消する競合のタイプによって決まります。

競合のタイプ	プロシージャ名
更新	DROP_UPDATE_RESOLUTION
一意	DROP_UNIQUE_RESOLUTION
削除	DROP_DELETE_RESOLUTION

構文

DROP_conflicttype_RESOLUTION プロシージャのパラメータは、表 9-150 を参照してください。例外は、表 9-151 を参照してください。 DROP_UPDATE_RESOLUTION プロシージャの構文は、次のとおりです。

```
DBMS_REPCAT.DROP_UPDATE_RESOLUTION(
    sname          IN   VARCHAR2,
    oname          IN   VARCHAR2,
    column_group   IN   VARCHAR2,
    sequence_no    IN   NUMBER)
```

DROP_DELETE_RESOLUTION プロシージャの構文は、次のとおりです。

```
DBMS_REPCAT.DROP_DELETE_RESOLUTION(
    sname          IN   VARCHAR2,
    oname          IN   VARCHAR2,
    sequence_no    IN   NUMBER)
```

DROP_UNIQUE_RESOLUTION プロシージャの構文は、次のとおりです。

```
DBMS_REPCAT.DROP_UNIQUE_RESOLUTION(
    sname          IN   VARCHAR2,
    oname          IN   VARCHAR2,
    constraint_name IN   VARCHAR2,
    sequence_no    IN   NUMBER)
```

表 9-150 DROP_conflicttype_RESOLUTION のパラメータ

パラメータ	説明
sname	表が存在するスキーマ。
oname	競合解消ルーチンを削除する表の名前。
column_group	更新競合解消ルーチンを削除する列グループの名前。
constraint_name	一意競合解消ルーチンを削除する一意制約の名前。
sequence_no	削除する競合解消方法に割り当てられた順序番号。この番号によって一意にルーチンを識別します。

表 9-151 DROP_conflicttype_RESOLUTION 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のスキーマ内の表として存在していないか、指定の順序番号をもつ競合解消ルーチンが登録されていません。
referenced	競合解消ルーチンは、競合の解消で使用中です。

表 9-151 DROP_conflicttype_RESOLUTION の例外

例外	説明
notquiesced	レプリケート・オブジェクト・グループは休止していません。

DBMS_REPCAT.EXECUTE_DDL

目的

各マスター・サイトで実行させたい DDL を指定します。このプロシージャは、マスター定義サイト上でしかコールできません。

構文

EXECUTE_DDL プロシージャのパラメータは、表 9-152 を参照してください。例外は、表 9-153 を参照してください。このプロシージャの構文は、次のとおりです。

```
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：スキーマを指定しないで DDL を指定すると、レプリケーション管理者のスキーマがデフォルトで指定されます。レプリケーション管理者のスキーマを使用しない場合は、必ずスキーマを指定してください。このプロシージャはオーバーロードされます。MASTER_LIST と MASTER_TABLE パラメータは、両方同時に指定できません。

表 9-152 EXECUTE_DDL のパラメータ

パラメータ	説明
gname	レプリケート・オブジェクト・グループの名前。
master_list	指定された DDL を実行する、マスター・サイトのカンマで区切られたリスト。サイト名間に余分な空白が入らないようにしてください。デフォルト値 NULL は、DDL がマスター定義サイトを含めたすべてのサイトで実行されることを示します。
master_table	指定された DDL を実行するマスター・サイトの PL/SQL 表。最初のマスターはオフセット 1 に、2 番目はオフセット 2、のようになります。
ddl_text	指定されたそれぞれのマスター・サイトで実行する DDL です。

表 9-153 EXECUTE_DDL 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	マスター・サイトでないサイトが少なくとも 1 つあります。
ddlfailure	マスター定義サイトで DDL は成功しませんでした。
commfailure	アクセスできないマスター・サイトが少なくとも 1 つあります。

DBMS_REPCAT.GENERATE_REPLICATION_PACKAGE

目的

レプリケーション・サポート生成のよりきめ細かい制御を提供します。Oracle7 リリース 7.3 のサイトを含む環境で主に使われます。指定した表についてのレプリケーションをすべてのマスター・サイトでサポートするために必要なパッケージを生成します。このプロシージャは、マスター定義サイトでコールしなければなりません。

構文

GENERATE_REPLICATION_PACKAGE プロシージャのパラメータは、表 9-154 を参照してください。例外は、表 9-155 を参照してください。このプロシージャの構文は、次のとおりです。

```
DBMS_REPCAT.GENERATE_REPLICATION_PACKAGE(  
    sname IN VARCHAR2,  
    oname IN VARCHAR2)
```

表 9-154 GENERATE_REPLICATION_PACKAGE のパラメータ

パラメータ	説明
sname	表が存在するスキーマ。
oname	レプリケーション・サポートを生成しようとする表の名前。

表 9-155 GENERATE_REPLICATION_PACKAGE 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、行レベルのレプリケーション情報を待っている指定のスキーマ内の表として、または Wrapper 生成を待っているプロシージャやパッケージ（本体）として存在しているではありません。
commfailure	アクセスできないマスター・サイトが少なくとも 1 つあります。
notcompat	このプロシージャには、リリース 7.3 以降が必要です。
notquiesced	レプリケート・オブジェクト・グループは休止されませんでした。

DBMS_REPCAT.GENERATE_REPLICATION_SUPPORT

目的

レプリケーションをサポートするのに必要なトリガーおよびパッケージ、プロシージャを生成します。このプロシージャは、マスター定義サイトでコールしなければなりません。

構文

GENERATE_REPLICATION_SUPPORT プロシージャのパラメータは、表 9-156 を参照してください。例外は、表 9-157 を参照してください。このプロシージャの構文は、次のとおりです。

```
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)
```


表 9-156 GENERATE_REPLICATION_SUPPORT のパラメータ

パラメータ	説明
sname	オブジェクトが置かれているスキーマの名前。
oname	レプリケーション・サポートを生成中のオブジェクトの名前。
type	オブジェクトの型。サポートされている型は TABLE および PACKAGE、PACKAGE BODY です。
package_prefix	PACKAGE または PACKAGE BODY 型のオブジェクトの場合、この値は生成された Wrapper パッケージ名の前に付加されます。デフォルトは DEFER_ です。
procedure_prefix	PROCEDURE または PACKAGE、PACKAGE BODY 型のオブジェクトの場合、この値は生成された Wrapper プロシージャ名の前に付加されます。デフォルトの場合、接頭辞は割り当てられません。デフォルトは DEFER_ です。
distributed	COMPATIBLE パラメータを 7.3.0 以上に設定する場合、このパラメータを TRUE に設定しなくてはなりません。
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。
min_communication	任意のマスター・サイトが Oracle7 リリース 7.3 を実行している場合、FALSE に設定します。新しい値および古い値の伝播を最小化するときは、TRUE に設定します。デフォルトは TRUE です。詳細は、5-34 ページの「更新の競合解消のためのデータ伝播の最小化」を参照してください。

表 9-157 GENERATE_REPLICATION_SUPPORT 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、行レベルのレプリケーション情報を待っている指定のスキーマ内の表として、または Wrapper 生成を待っているプロシージャやパッケージ（本体）として存在しているではありません。
typefailure	指定された型パラメータはサポートされていません。
notquiesced	レプリケート・オブジェクト・グループは、まだ中断されていません。
commfailure	アクセスできないマスター・サイトが少なくとも 1 つあります。
missschema	スキーマが存在しません。

表 9-157 GENERATE_REPLICATION_SUPPORT の例外

例外	説明
dbnotcompatible	マスターの 1 つがリリース 7.3 と互換性がありません。
duplicateobject	オブジェクトがすでに存在します。

DBMS_REPCAT.GENERATE_REPLICATION_TRIGGER

目的

レプリケーション・サポート生成のよりきめ細かい制御を提供します。Oracle7 リリース 7.3 のサイトを含む環境で主に使われます。指定したオブジェクトのレプリケーションをすべてのマスター・サイトでサポートするために必要なトリガーと、それに関連付けられたパッケージを生成します。または、マスター・サイトのリストにある指定オブジェクト・グループのすべてのオブジェクトのレプリケーションをサポートするために必要なトリガーと、それに関連付けられた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でコールしなければなりません。関連付けられたオブジェクト・グループを休止しなくてはなりません。

構文

GENERATE_REPLICATION_TRIGGER プロシージャのパラメータは、表 9-158 を参照してください。例外は、表 9-159 を参照してください。このプロシージャの構文は、次のとおりです。

```
DBMS_REPCAT.GENERATE_REPLICATION_TRIGGER(  
    sname                IN  VARCHAR2,  
    oname                IN  VARCHAR2,  
    gen_objs_owner       IN  VARCHAR2 := NULL,  
    min_communication    IN  BOOLEAN  := TRUE)  
DBMS_REPCAT.GENERATE_REPLICATION_TRIGGER(  
    gname                IN  VARCHAR2,  
    gen_objs_owner       IN  VARCHAR2 := NULL,  
    min_communication    IN  BOOLEAN  := NULL)
```

注意: GENERATE_REPLICATION_TRIGGER プロシージャはオーバーロードされるので、すべてのマスター・サイトの単一のオブジェクト、またはリストのサイトのオブジェクト・グループのためにサポートを生成できます。コールのパラメータ・タイプがいずれも同じため、単一オブジェクト用プロシージャまたはオブジェクト・グループ用プロシージャのうちのどちらをコールしようとしているのかを指示するために、名前表記法を使用しなくてはならない場合があります。

注意：マスター・サイトのリストに対するサポートを生成する場合（つまり、デフォルトの NULL を使用しない場合）配列または名前表記法のいずれかを使用する必要があります。

表 9-158 GENERATE_REPLICATION_TRIGGER のパラメータ

パラメータ	説明
sname	オブジェクトが置かれているスキーマの名前。
oname	レプリケーション・サポートを生成中のオブジェクトの名前。
gname	サポートを生成するオブジェクト・グループの名前。
gen_objs_owner	このパラメータは、以前のリリースと互換性を取るために提供されています。リリース 7.3 以前のスナップショット・サイトがある場合、このパラメータを TRUE に設定しなくてはなりません。
min_communication	任意のマスター・サイトが Oracle7 リリース 7.3 を実行している場合、FALSE に設定します。新しい値および古い値の伝播を最小化するときは、TRUE に設定します。デフォルトは場合によって変わります。詳細は、5-34 ページの「更新の競合解消のためのデータ伝播の最小化」を参照してください。

表 9-159 GENERATE_REPLICATION_TRIGGER 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、行レベルのレプリケーション情報を待っている指定のスキーマ内の表として、または Wrapper 生成を待っているプロシージャやパッケージ（本体）として存在しているではありません。
notquiesced	レプリケート・オブジェクト・グループは、まだ中断されていません。
commfailure	アクセスできないマスター・サイトが少なくとも 1 つあります。
notcompat	マスターの 1 つがリリース 7.3 と互換性がありません。
missingschema	指定されたスキーマが存在しません。

伝播モードの変更

オブジェクト・グループの伝播モードを変更した後、レプリケート環境内の Oracle7 リリース 7.3 のサイトごとに、これらのオブジェクトのサポートする PL/SQL トリガーを再生成する必要があります。内部トリガーによって伝播モードは自動的に ALTER_MASTER_PROPAGATION で指定されているように変更されます。内部トリガーは、トリガー再生成を試みても影響を受けません。

サポートする PL/SQL トリガーおよびこれに関連付けられたパッケージを、指定した一連のマスター・サイトのオブジェクト・グループのすべての Oracle7 リリース 7.3 メンバー用に生成するには、次の例に示すように DBMS_REPCAT.GENERATE_REPLICATION_TRIGGER プロシージャをコールしてください。

```
DBMS_REPCAT.GENERATE_REPLICATION_TRIGGER( gname => 'acct');
```

この例ではマスター・サイトのリストを指定していないため、Oracle は、GNAME オブジェクト・グループのオブジェクトをサポートするトリガーおよびこれに関連付けられたパッケージをすべての Oracle7 リリース 7.3 マスター・サイトで再生成します。このプロシージャは、指定したレプリケート・オブジェクト・グループのマスター定義サイトからコールしなくてはなりません。このプロシージャは、マスター定義サイトで必要なトリガーが正常に生成されると正常終了します。これらのオブジェクトは、以下で説明するように、他のマスターサイトで非同期に作成されます。

DBMS_REPCAT.GENERATE_SNAPSHOT_SUPPORT

目的

トリガーを起動し、更新可能スナップショットのレプリケーションまたはプロシージャ・レプリケーションをサポートするために必要なパッケージを生成します。このプロシージャはスナップショット・サイトでコールする必要があります。

構文

GENERATE_SNAPSHOT_SUPPORT プロシージャのパラメータは、表 9-160 を参照してください。例外は、表 9-161 を参照してください。このプロシージャの構文は、次のとおりです。

```
DBMS_REPCAT.GENERATE_SNAPSHOT_SUPPORT
    sname          IN VARCHAR2,
    oname          IN VARCHAR2,
    type           IN VARCHAR2,
    gen_objs_owner  IN VARCHAR2 := '',
    min_communication IN BOOLEAN := TRUE)
```

注意: CREATE_SNAPSHOT_REPOBJECT は、更新可能なスナップショットに対するスナップショット・サポートを自動的に生成します。

表 9-160 GENERATE_SNAPSHOT_SUPPORT のパラメータ

パラメータ	説明
sname	オブジェクトが置かれているスキーマの名前。
oname	オブジェクトが、レプリケート・スナップショット・オブジェクト・グループ内に行 / 列レベル・レプリケーションを使用する更新可能スナップショットとして存在する場合、行レベルのレプリケーション・トリガーおよびストアド・パッケージを生成します。
type	オブジェクトの型。サポートされている型は SNAPSHOT および PACKAGE、PACKAGE BODY です。
gen_objs_owner	生成されるレプリケーション・トリガーおよびトリガー・パッケージまたは Wrapper がインストールされるスキーマを指定します。NULL の場合、生成されるトリガーおよびトリガー・パッケージまたは Wrapper は、sname パラメータで指定されるスキーマにインストールされます。
min_communication	TRUE の場合、更新トリガーが列の新しい値を送るのは、UPDATE 文がその列を修正するときだけです。更新トリガーがその列の古い値を送るのは、その列がキー列であるか、修正された列グループ内の列であるときだけです。

表 9-161 GENERATE_SNAPSHOT_SUPPORT の例外

例外	説明
nonsnapshot	起動サイトはスナップショット・サイトではありません。
missingobject	指定されたオブジェクトは、行 / 列レベルのレプリケーション情報を待っているレプリケート・スキーマ内の表として、または Wrapper の生成を待っているプロシージャやパッケージ（本体）として存在しているではありません。
typefailure	指定された型パラメータはサポートされていません。
missingschema	生成されたオブジェクトの指定の所有者が存在しません。
missingremoteobject	マスター・オブジェクトがレプリケーション・サポートをまだ生成していません。
commfailure	マスターにはアクセスできません。

DBMS_REPCAT.MAKE_COLUMN_GROUP

目的

1 つ以上の列メンバーで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でコールしなければなりません。詳細は、 5-11 ページの「列グループの作成」を参照してください。

構文

MAKE_COLUMN_GROUP プロシージャのパラメータは、表 9-162 を参照してください。例外は、表 9-163 を参照してください。このプロシージャの構文は、次のとおりです。

```
DBMS_REPCAT.MAKE_COLUMN_GROUP (
    sname                IN    VARCHAR2,
    oname                IN    VARCHAR2,
    column_group         IN    VARCHAR2,
    list_of_column_names IN    VARCHAR2 | DBMS_REPCAT.VARCHAR2S)
```

表 9-162 MAKE_COLUMN_GROUP 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ているスキーマの名前。
oname	新規列グループを作成するレプリケート表の名前。
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。これは、カンマで区切られた列名のリスト、または PL/SQL 表の列名のどちらでもかまいません。PL/SQL 表は dbms_repcat.varchar2 型にしてください。表内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。

表 9-163 MAKE_COLUMN_GROUP 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すでに表に存在しています。
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、すでに別の列グループのメンバーになっています。

表 9-163 MAKE_COLUMN_GROUP の例外

例外	説明
notquiesced	レプリケート・オブジェクト・グループは休止していません。

DBMS_REPCAT.PURGE_MASTER_LOG

目的

指定された識別番号または発信元、レプリケート・オブジェクト・グループに対応付けられた RepCatLog 内のローカル・メッセージを削除します。

構文

PURGE_MASTER_LOG プロシージャのパラメータは、表 9-164 を参照してください。例外は、表 9-165 を参照してください。NULL パラメータがある場合、Oracle はワイルドカードとしてそれを処理します。このプロシージャの構文は、次のとおりです。

```
DBMS_REPCAT.PURGE_MASTER_LOG(
    id      IN    NATURAL,
    source  IN    VARCHAR2,
    gname   IN    VARCHAR2)
```

表 9-164 PURGE_MASTER_LOG のパラメータ

パラメータ	説明
id	要求の識別番号。RepCatLog ビューに表示されます。
source	要求発信元のマスター・サイトの名前。
gname	この要求の対象となるレプリケート・オブジェクト・グループの名前。

表 9-165 PURGE_MASTER_LOG の例外

例外	説明
nonmaster	GNAME は NULL ではなく、起動サイトはマスター・サイトではありません。

DBMS_REPCAT.PURGE_STATISTICS

目的

RepResolution_Statistics ビューから情報を削除します。

構文

PURGE_STATISTICS プロシージャのパラメータは、表 9-166 を参照してください。例外は、表 9-167 を参照してください。このプロシージャの構文は、次のとおりです。

```
DBMS_REPCAT.PURGE_STATISTICS(  
    sname      IN   VARCHAR2,  
    oname      IN   VARCHAR2,  
    start_date IN   DATE,  
    end_date   IN   DATE)
```

表 9-166 PURGE_STATISTICS 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ているスキーマの名前。
oname	競合解消統計を削除する表の名前。
start_date/end_date	統計の削除を実行する日付の範囲です。START_DATE が NULL の場合、END_DATE まですべての統計を削除します。END_DATE が NULL の場合、START_DATE 以後のすべての統計を削除します。

表 9-167 PURGE_STATISTICS の例外

例外	説明
missingschema	指定されたスキーマが存在しません。
missingobject	指定された表は存在しません。
statnotreg	表が統計を収集するために登録されていません。

DBMS_REPCAT.REFRESH_SNAPSHOT_REPGROUP

目的

関連付けられたマスター・サイトから最新のデータで、スナップショット・サイトのオブジェクト・グループをリフレッシュします。

構文

REFRESH_SNAPSHOT_REPGROUP プロシージャのパラメータは、表 9-168 を参照してください。例外は、表 9-169 を参照してください。このプロシージャの構文は、次のとおりです。


```

DBMS_REPCAT.REFRESH_SNAPSHOT_REPGROUP (
    gname                IN    VARCHAR2,
    drop_missing_contents IN    BOOLEAN    := FALSE,
    refresh_snapshots     IN    BOOLEAN    := FALSE,
    refresh_other_objects IN    BOOLEAN    := FALSE)

```

表 9-168 REFRESH_SNAPSHOT_REPGROUP のパラメータ

パラメータ	説明
gname	レプリケート・オブジェクト・グループの名前。
drop_missing_contents	オブジェクトがレプリケート・オブジェクト・グループから削除された場合、スナップショット・サイトでオブジェクトがスキーマから自動的に削除されることはありません。オブジェクトは、レプリケートされなくなるだけです（つまり、このオブジェクトに対する変更は、関連付けられたマスター・サイトに送信されなくなります）。スナップショットは、引き続き関連付けられたマスター表からリフレッシュされますが、更新可能スナップショットに対する変更はすべて失われます。オブジェクト・グループからオブジェクトが削除される際に、この引数を TRUE に設定すると、オブジェクトをスキーマから完全に削除できます。
refresh_snapshots	レプリケート・オブジェクト・グループ内のスナップショットの内容をリフレッシュするには、このパラメータを TRUE に設定します。
refresh_other_objects	レプリケート・オブジェクト・グループ内の非スナップショット・オブジェクトの内容をリフレッシュするには、このパラメータを TRUE に設定します。

表 9-169 REFRESH_SNAPSHOT_REPGROUP の例外

例外	説明
nonsnapshot	起動サイトはスナップショット・サイトではありません。
nonmaster	このマスターは、マスター・サイトではありません。
commfailure	マスターにはアクセスできません。
missrepgroup	オブジェクト・グループ名が指定されていません。

DBMS_REPCAT.REGISTER_SNAPSHOT_REPGROUP

目的

repcat_repsite への挿入 / 修正 / 削除を行うことにより、それぞれのマスター・サイトのスナップショットの管理に役立ちます。

構文

REGISTER_SNAPSHOT_REPGROUP プロシージャのパラメータは、表 9-170 を参照してください。REGISTER_SNAPSHOT_REPGROUP プロシージャの構文は、次のとおりです。

```
DBMS_REPCAT.REGISTER_SNAPSHOT_REPGROUP(  
    gname          IN    VARCHAR2,  
    snapsite       IN    VARCHAR2,  
    comment        IN    VARCHAR2    := NULL,  
    rep_type       IN    NUMBER      := reg_unknown)
```

表 9-170 REGISTER_SNAPSHOT_REPGROUP のパラメータ

パラメータ	説明
gname	登録するスナップショット・オブジェクト・グループの名前。
snapsite	スナップショット・サイトのグローバル名。
comment	スナップショット・サイトに対するコメント、または既存のコメントの更新。
rep_type	スナップショット・グループのバージョン。代入できる有効な定数。reg_unknown (デフォルト) および reg_v7_group、reg_v8_group、reg_repapi_group を含みます。

表 9-171 REGISTER_SNAPSHOT_REPGROUP の例外

例外	説明
missrepgroup	オブジェクト・グループ名が指定されていません。
nullsitename	スナップショット・サイトが指定されていません。
nonmaster	プロシージャは、スナップショットのマスター・サイトで実行しなくてはなりません。
duplrepggrp	オブジェクトはすでに存在します。

DBMS_REPCAT.REGISTER_STATISTICS

目的

表の更新および削除、一意の競合の正常な解消に関する情報を収集します。

構文

REGISTER_STATISTICS プロシージャのパラメータは、表 9-172 を参照してください。例外は、表 9-173 を参照してください。このプロシージャの構文は、次のとおりです。

```
DBMS_REPCAT.REGISTER_STATISTICS(  
    sname IN    VARCHAR2,  
    oname IN    VARCHAR2)
```

表 9-172 REGISTER_STATISTICS のパラメータ

パラメータ	説明
sname	表が置かれているスキーマの名前。
oname	競合解消統計を収集する表の名前。

表 9-173 REGISTER_STATISTICS の例外

例外	説明
missingschema	指定されたスキーマが存在しません。
missingobject	指定された表は存在しません。

DBMS_REPCAT.RELOCATE_MASTERDEF

目的

マスター定義サイトをレプリケート環境内の別のマスター・サイトに変更します。

構文

RELOCATE_MASTERDEF プロシージャのパラメータは、表 9-174 を参照してください。例外は、表 9-175 を参照してください。このプロシージャの構文は、次のとおりです。

```
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)
```

表 9-174 RELOCATE_MASTERDEF のパラメータ

パラメータ	説明
gname	そのマスター定義を再配置するオブジェクト・グループの名前。
old_masterdef	現行マスター定義サイトの省略のない形のデータベース名。
new_masterdef	新規マスター定義サイトを作成する既存マスター・サイトの省略のない形のデータベース名。
notify_masters	NOTIFY_MASTERS が TRUE の場合、プロシージャは同期的にすべてのマスターへの変更をマルチキャストします (INCLUDE_OLD_MASTERDEF が TRUE の場合だけ OLD_MASTERDEF も含まれる)。変更できないマスターがある場合、すべてのマスターで変更がロールバックされます。
include_old_masterdef	NOTIFY_MASTERS が TRUE の場合、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。

表 9-175 RELOCATE_MASTERDEF の例外

例外	説明
nonmaster	NEW_MASTERDEF がマスター・サイトではない、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	NOTIFY_MASTERS が TRUE の場合、アクセスできないマスター・サイトが少なくとも 1 つあります。

使用上の注意事項

RELOCATE_MASTERDEF をコールする場合、古いまたは新規のマスター定義サイトのどちらでも使用できるようになっている必要はありません。計画的に再構成する場合は、NOTIFY_MASTERS および INCLUDE_OLD_MASTERDEF を TRUE にして RELOCATE_MASTERDEF を起動する必要があります。マスター定義サイトだけが異常障害を起こしている場合は、NOTIFY_MASTERS を TRUE に、INCLUDE_OLD_MASTERDEF を FALSE にして RELOCATE_MASTERDEF を起動する必要があります。いくつかのマスター・サイトおよびマスター定義サイトで異常障害が起きている場合、管理者は NOTIFY_MASTERS を FALSE にして各マスターで RELOCATE_MASTERDEF を起動する必要があります。

DBMS_REPCAT.REMOVE_MASTER_DATABASES

目的

レプリケート環境から 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、トリガーおよびこれらに関連付けられたパッケージを他のマスター・サイトで再生成します。このプロシージャは、マスター定義サイトでコールしなければなりません。

構文

REMOVE_MASTER_DATABASES プロシージャのパラメータは、表 9-176 を参照してください。例外は、表 9-177 を参照してください。このプロシージャの構文は、次のとおりです。

```
DBMS_REPCAT.REMOVE_MASTER_DATABASES(  
    gname          IN   VARCHAR2,  
    master_list    IN   VARCHAR2 |  
    master_table   IN   DBMS_UTILITY.DBLINK_ARRAY)
```

表 9-176 REMOVE_MASTER_DATABASES のパラメータ

パラメータ	説明
gname	レプリケート環境に関連付けられたオブジェクト・グループの名前。マスター・データベースが複数のレプリケート環境を含んでいる場合でも、この名前があれば混乱しません。
master_list	レプリケート環境から削除する省略のない形のマスター・データベース名の、カンマで区切られたリスト。リストの中で名前と名前の間に余分の空白を入れないでください。
master_table	リストのかわりに、DBMS_UTILITY.DBLINK_ARRAY 型の PL/SQL 表のデータベース名を指定することもできます。

表 9-177 REMOVE_MASTER_DATABASES 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少なくとも 1 つはマスター・サイトではありません。
reconfigerror	指定されたデータベースの 1 つはマスター定義サイトです。
commfailure	残っているマスター・サイトの中に、アクセスできないサイトが少なくとも 1 つあります。

DBMS_REPCAT.REPCAT_IMPORT_CHECK

目的

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するオブジェクトのエクスポート/インポートの実行後、レプリケート・オブジェクト・グループ内のオブジェクトが、適切なオブジェクト識別子とステータス値を持っていることを確認します。

構文

REPCAT_IMPORT_CHECK プロシージャのパラメータは、表 9-178 を参照してください。例外は、表 9-179 を参照してください。このプロシージャの構文は、次のとおりです。

```
DBMS_REPCAT.REPCAT_IMPORT_CHECK(  
    gname          IN   VARCHAR2,  
    master         IN   BOOLEAN)
```

表 9-178 REPCAT_IMPORT_CHECK のパラメータ

パラメータ	説明
gname	レプリケート・オブジェクト・グループの名前。両方のパラメータを省略した場合、プロシージャによって現行サイトのすべてのレプリケート・オブジェクト・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のフラグを TRUE に、スナップショット・サイトをチェックする場合は FALSE に設定します。

表 9-179 REPCAT_IMPORT_CHECK の例外

例外	説明
nonmaster	MASTER が TRUE の場合、データベースがスキーマのマスター・サイトでない、または予期されたデータベースではありません。
nonsnapshot	MASTER が FALSE の場合、データベースはスキーマのスナップショット・サイトではありません。
missingobject	スキーマに有効なレプリケート・オブジェクトが存在しません。
missingschema	指定されたグループ名は存在しません。

DBMS_REPCAT.RESUME_MASTER_ACTIVITY

目的

レプリケート環境を休止した後、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

RESUME_MASTER_ACTIVITY プロシージャのパラメータは、表 9-180 を参照してください。例外は、表 9-181 を参照してください。このプロシージャの構文は、次のとおりです。

```
DBMS_REPCAT.RESUME_MASTER_ACTIVITY(  
    gname          IN  VARCHAR2,  
    override       IN  BOOLEAN := FALSE)
```

表 9-180 RESUME_MASTER_ACTIVITY のパラメータ

パラメータ	説明
gname	レプリケート・オブジェクト・グループの名前。
override	override を TRUE にすると、保留中の RepCat 管理要求を無視し、可能な限り迅速に各マスターに通常のレプリケーション・アクティビティを復元します。これは緊急事態に限り使用できます。override を FALSE にすると、各マスターで GNAME の保留中の RepCat 管理要求がない場合だけ、そのマスターに通常のレプリケーション・アクティビティを復元します。

表 9-181 RESUME_MASTER_ACTIVITY 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	レプリケート・オブジェクト・グループは休止中または休止状態ではありません。
commfailure	アクセスできないマスター・サイトが少なくとも 1 つあります。

DBMS_REPCAT.SEND_AND_COMPARE_OLD_VALUES

目的

レプリケート表の各非キー列に対して、更新および削除のために古い列値を送るオプションがあります。デフォルトでは、すべての列に古い値を送ります。マスター定義サイトで DBMS_REPCAT.SEND_AND_COMPARE_OLD_VALUES を起動することによって、すべてのマスター・サイトおよびスナップショット・サイトでこの動作を変更できます。

注意: OPERATION パラメータを使用すると、行が削除されたとき、またはキー以外の列が更新されたときに、キー以外の列の古い値を送信するかどうかを決定できます。古い値を送信しない場合、Oracle は古い値のかわりに NULL を送信し、更新または削除が適用されたときに送信先の列の現在の値が古い値と等しいことを想定します。

注意: Oracle のデフォルト動作を変更する前に、5-34 ページの「更新の競合解消のためのデータ伝播の最小化」を参照してください。

構文

SEND_AND_COMPARE_OLD_VALUES プロシージャのパラメータは、表 9-182 を参照してください。例外は、表 9-183 を参照してください。このプロシージャの構文は、次のとおりです。

```
DBMS_REPCAT.SEND_AND_COMPARE_OLD_VALUES(
    sname          IN    VARCHAR2,
    oname          IN    VARCHAR2,
    { column_list  IN    VARCHAR2,
    | column_table IN    DBMS_REPCAT.VARCHAR2s, }
    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ます。COLUMN_LIST と COLUMN_TABLE パラメータは、両方同時に指定できません。

表 9-182 SEND_AND_COMPARE_OLD_VALUES のパラメータ

パラメータ	説明
sname	表が存在するスキーマ。
oname	表の名前。
column_list	表の列のカンマ区切りのリスト。項目と項目の間に空白を入れてはいけません。
column_table	列名を入れる際は、リストのかわりに DBMS_REPCAT.VARCHAR2S 型の PL/SQL 表も使用できます。最初の列名はオフセット 1、2 番目はオフセット 2、といった具合になります。
operation	可能な値は、UPDATE または DELETE、アスタリスクのワイルドカード '*' (これは、更新および削除を意味します) です。

表 9-182 SEND_AND_COMPARE_OLD_VALUES のパラメータ

パラメータ	説明
send	TRUE の場合、指定した列の古い値が送られます。FALSE の場合、指定した列の古い値は送られません。指定外の列や指定外の操作は影響を受けません。その表に対する min_communication が TRUE になるとすぐに、指定された変更がマスター定義サイトで有効になります。変更内容がマスター・サイトまたはスナップショット・サイトで有効になるのは、次にそのサイトで min_communication を TRUE としてレプリケーション・サポートが生成されるときです。

表 9-183 SEND_AND_COMPARE_OLD_VALUES 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、行レベルのレプリケーション情報を待っている指定のスキーマ内の表として存在しているではありません。
missingcolumn	表に入っていない列が少なくとも 1 つあります。
notquiesced	レプリケート・オブジェクト・グループは、まだ中断されていません。
typefailure	無効な操作が指定されています。

DBMS_REPCAT.SET_COLUMNS

目的

主キーのかわりに代替列または列グループを使用するために、行レベル・レプリケーションを使用するとき、表のどの列を比較するかを判断します。このプロシージャは、マスター定義サイトでコールしなければなりません。3-18ページの「レプリケート表の代替キーの指定」を参照してください。

構文

SET_COLUMNS プロシージャのパラメータは、表 9-184 を参照してください。例外は、表 9-185 を参照してください。このプロシージャの構文は、次のとおりです。

```
DBMS_REPCAT.SET_COLUMNS(
    sname          IN      VARCHAR2,
    oname          IN      VARCHAR2,
    { column_list  IN      VARCHAR2
    | column_table 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ます。COLUMN_LIST と COLUMN_TABLE パラメータは、両方同時に指定できません。

表 9-184 SET_COLUMNS のパラメータ

パラメータ	説明
sname	表が存在するスキーマ。
oname	表の名前。
column_list	「主キー」として使用する表の中の、列のカンマで区切られたリスト。項目と項目の間に空白を入れてはいけません。
column_table	列名を入れる際は、リストのかわりに DBMS_UTILITY.NAME_ARRAY 型の PL/SQL 表も使用できます。最初の列名はオフセット 1、2 番目はオフセット 2、といった具合になります。

表 9-185 SET_COLUMNS 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、行レベルのレプリケーション情報を待っている指定のスキーマ内の表として存在していません。
missingcolumn	表に入っていない列が少なくとも 1 つあります。

DBMS_REPCAT.SUSPEND_MASTER_ACTIVITY

目的

オブジェクト・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。このプロシージャは、マスター定義サイトでコールしなければなりません。

注意: 現行の SUSPEND_MASTER_ACTIVITY の実行により、各マスター・サイトのすべてのレプリケート・オブジェクト・グループが休止します。

構文

SUSPEND_MASTER_ACTIVITY プロシージャのパラメータは、表 9-186 を参照してください。例外は、表 9-187 を参照してください。このプロシージャの構文は、次のとおりです。

```
DBMS_REPCAT.SUSPEND_MASTER_ACTIVITY(gname IN VARCHAR2)
```

表 9-186 SUSPEND_MASTER_ACTIVITY 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オブジェクト・グループの名前。

表 9-187 SUSPEND_MASTER_ACTIVITY 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	レプリケート・オブジェクト・グループは標準の操作モードではありません。
commfailure	アクセスできないマスター・サイトが少なくとも 1 つあります。
missingobjectgroup	指定されたオブジェクト・グループが存在しません。

DBMS_REPCAT.SWITCH_SNAPSHOT_MASTER

目的

スナップショット・レプリケート・オブジェクト・グループのマスター・データベースを別のマスター・サイトに変更します。このプロシージャは、影響を受けるスナップショットを完全にリフレッシュし、必要に応じてトリガーおよびこれに関連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、マスターを変更する前に古いマスター・サイトにキューを送信しません。

構文

SWITCH_SNAPSHOT_MASTER プロシージャのパラメータは、表 9-188 を参照してください。例外は、表 9-189 を参照してください。このプロシージャの構文は、次のとおりです。

```
DBMS_REPCAT.SWITCH_SNAPSHOT_MASTER(
    gname          IN   VARCHAR2,
    master         IN   VARCHAR2)
```

表 9-188 SWITCH_SNAPSHOT_MASTER のパラメータ

パラメータ	説明
gname	マスター・サイトを変更するスナップショット・オブジェクト・グループの名前。
master	スナップショット・サイトに使用する新規マスター・データベースの省略のない形のデータベース名。

表 9-189 SWITCH_SNAPSHOT_MASTER の例外

例外	説明
nonsnapshot	起動サイトはスナップショット・サイトではありません。
nonmaster	指定された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。

DBMS_REPCAT.UNREGISTER_SNAPSHOT_REPGROUP

目的

repcat\$_repsite の挿入 / 修正 / 削除によって、それぞれのマスター・サイトのスナップショットの管理に役立ちます。

構文

UNREGISTER_SNAPSHOT_REPGROUP のパラメータは、表 9-190 を参照してください。UNREGISTER_SNAPSHOT_REPGROUP プロシージャの構文は、次のとおりです。

```
DBMS_REPCAT.UNREGISTER_SNAPSHOT_REPGROUP(  
    gname      IN   VARCHAR2,  
    snapsite   IN   VARCHAR2)
```

表 9-190 UNREGISTER_SNAPSHOT_REPGROUP のパラメータ

パラメータ	説明
gname	登録解除するスナップショット・オブジェクト・グループの名前。
snapsite	スナップショット・サイトのグローバル名。

DBMS_REPCAT.VALIDATE

目的

複数マスター・レプリケーション環境のキー条件の正確さを検証するために、VALIDATE プロシージャを実行できます。このプロシージャはオーバーロードされます。

構文

VALIDATE ファンクションのパラメータは、表 9-191 を参照してください。例外は、表 9-192 を参照してください。このプロシージャの構文は、次のとおりです。

```
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

```

表 9-191 VALIDATE のパラメータ

パラメータ	説明
gname	検証するマスター・グループの名前。
check_genflags	グループのすべてのオブジェクトが生成されるかどうかチェックします。これを実行する必要があるのはマスター定義サイトだけです。
check_valid_objs	有効なグループ内のオブジェクトの基礎を形成するオブジェクトをチェックします。これを実行する必要があるのはマスター定義サイトだけです。 masterdef サイトが他のすべてのサイトを調べて、基礎を形成するオブジェクトが有効であることをチェックします。オブジェクトの有効性は、接続されているユーザーのスキーマ内でチェックされます。
check_links_sched	リンクの実行がスケジュールされているかどうか調べます。これは、各マスター・サイトで起動する必要があります。
check_links	接続されたユーザー（ repadmin ）および伝播担当者に、適切に動作するためのレプリケーションの正しいリンクがあるかどうかをチェックします。データベースにリンクが存在し、アクセス可能であることをチェックします。これは、各マスター・サイトで起動する必要があります。
error_table	見つかったすべてのエラーのメッセージと数値を戻します。
error_msg_table	見つかったすべてのエラーのメッセージを戻します。

表 9-191 VALIDATE のパラメータ

パラメータ	説明
error_num_table	見つかったすべてのエラーの数値を戻します。

VALIDATE の戻り値は、見つかったエラーの数です。ファンクションの OUT パラメータが、見つかったエラーを戻します。最初のインタフェース・ファンクションでは、ERROR_TABLE はレコードの配列で構成されています。各レコードには VARCHAR2 と NUMBER が含まれます。文字列フィールドはエラー・メッセージを格納し、数値フィールドは Oracle エラー番号を格納します。

2 番目のインタフェースは、OUT 配列が 2 つあることを除いて最初のインタフェースと同様です。2 つの OUT 配列は、エラー・メッセージを入れる VARCHAR2 配列と、エラーの数値を入れる NUMBER 配列です。

表 9-192 VALIDATE の例外

例外	説明
missingdblink	データベース・リンクが、レプリケーション伝播担当者のスキーマに存在しないか、スケジュールされていません。データベース・リンクがデータベースに存在し、アクセス可能で、実行がスケジュールされていることを確認します。
dblinkmismatch	ローカル・ノードのデータベース・リンク名が、その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。グローバル名を true に設定し、リンク名がグローバル名に一致するように確認します。
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同じではありません。アドバンスド・レプリケーションでは、2 つのユーザーは同一です。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 が同一であることを確認します。
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、生成中です。masterdef サイトでオブジェクトに対して generate_replication_support および do_deferred_repcat_admin をコールすることによって、オブジェクトを生成するように確認します。
opnotsupported	オブジェクト・グループが V8 より前のノードでレプリケートされている場合、操作がサポートされていません。レプリケート・オブジェクト・グループのすべてのノードが V8 になるように確認します。

DBMS_REPCAT.WAIT_MASTER_LOG

目的

マスター・サイトに非同期で伝播した変更が適用されたかどうかを判別します。

構文

WAIT_MASTER_LOG プロシージャのパラメータは、表 9-193 を参照してください。例外は、表 9-194 を参照してください。このプロシージャの構文は、次のとおりです。

```
DBMS_REPCAT.WAIT_MASTER_LOG(
    gname          IN    VARCHAR2,
    record_count    IN    NATURAL,
    timeout         IN    NATURAL,
    true_count      OUT   NATURAL)
```

表 9-193 WAIT_MASTER_LOG のパラメータ

パラメータ	説明
gname	レプリケート・オブジェクト・グループの名前。
record_count	不完全なアクティビティの数がこのしきい値以下の値になった場合、このプロシージャから戻ってきます。
timeout	プロシージャが戻るまで待つ最大秒数。
true_count (out parameter)	未完了アクティビティの数値を戻します。

表 9-194 WAIT_MASTER_LOG 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 パッケージ

DBMS_REPCAT_ADMIN パッケージには次のプロシージャが含まれます。

- DBMS_REPCAT_ADMIN.GRANT_ADMIN_ANY_SCHEMA
- DBMS_REPCAT_ADMIN.GRANT_ADMIN_SCHEMA
- DBMS_REPCAT_ADMIN.REVOKE_ADMIN_ANY_SCHEMA
- DBMS_REPCAT_ADMIN.REVOKE_ADMIN_SCHEMA

各プロシージャについて説明します。

DBMS_REPCAT_ADMIN.GRANT_ADMIN_ANY_SCHEMA

目的

現行サイトですべてのレプリケート・オブジェクト・グループを管理するために必要な権限を、レプリケーション管理者に付与します。

構文

GRANT_ADMIN_ANY_SCHEMA プロシージャのパラメータは、表 9-195 を参照してください。例外は、表 9-196 を参照してください。このプロシージャの構文は、次のとおりです。

```
DBMS_REPCAT_ADMIN.GRANT_ADMIN_ANY_SCHEMA(username IN VARCHAR2)
```

表 9-195 GRANT_ADMIN_ANY_SCHEMA のパラメータ

パラメータ	説明
username	現行サイトのすべてのレプリケート・オブジェクト・グループ管理に必要な権限およびロールを付与するレプリケーション管理者の名前。

表 9-196 GRANT_ADMIN_ANY_SCHEMA の例外

例外	説明
ORA-01917	ユーザーが存在しません。

DBMS_REPCAT_ADMIN.GRANT_ADMIN_SCHEMA

目的

レプリケーション管理者が現行サイトでスキーマを管理する場合に必要な権限を付与します。このプロシージャは、オブジェクト・グループが単一のスキーマ内にある場合に最も役立ちます。

構文

GRANT_ADMIN_REPSchema プロシージャのパラメータは、表 9-197 を参照してください。例外は、表 9-198 を参照してください。このプロシージャの構文は、次のとおりです。

```
DBMS_REPCAT_ADMIN.GRANT_ADMIN_SCHEMA(username IN VARCHAR2)
```


表 9-197 GRANT_ADMIN_REPSchema 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。このユーザーには、現行サイトで、レプリケート・オブジェクト・グループ内の同じ名前のスキーマを管理する場合に必要な権限およびロールが付与されます。

表 9-198 GRANT_ADMIN_REPSchema の例外

例外	説明
ORA-01917	ユーザーが存在しません。

DBMS_REPCAT_ADMIN.REVOKE_ADMIN_ANY_SCHEMA

目的

GRANT_ADMIN_ANY_SCHEMA によって付与された権限およびロールを、レプリケーション管理者から取り消します。

注意：独立して付与された GRANT_ADMIN_ANY_SCHEMA の同一の権限およびロールも取り消されます。

構文

REVOKE_ADMIN_ANY_SCHEMA プロシージャのパラメータは、表 9-199 を参照してください。例外は、表 9-200 を参照してください。このプロシージャの構文は、次のとおりです。

```
DBMS_REPCAT_ADMIN.REVOKE_ADMIN_ANY_SCHEMA(username IN VARCHAR2)
```

表 9-199 REVOKE_ADMIN_ANY_SCHEMA 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。

表 9-200 REVOKE_ADMIN_ANY_SCHEMA の例外

例外	説明
ORA-01917	ユーザーが存在しません。

DBMS_REPCAT_ADMIN.REVOKE_ADMIN_SCHEMA

目的

GRANT_ADMIN_SCHEMA によって付与された権限およびロールを、レプリケーション管理者から取り消します。

注意：独立して付与された GRANT_ADMIN_SCHEMA の同一の権限およびロールも取り消されます。

構文

REVOKE_ADMIN_SCHEMA プロシージャのパラメータは、表 9-201 を参照してください。例外は、表 9-202 を参照してください。このプロシージャの構文は、次のとおりです。

```
DBMS_REPCAT_ADMIN.REVOKE_ADMIN_SCHEMA(username IN VARCHAR2)
```

表 9-201 REVOKE_ADMIN_SCHEMA 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。

表 9-202 REVOKE_ADMIN_SCHEMA の例外

例外	説明
ORA-01917	ユーザーが存在しません。

DBMS_REPCAT_AUTH パッケージ

DBMS_REPCAT_AUTH パッケージには次のプロシージャが含まれます。

- DBMS_REPCAT_AUTH.GRANT_SURROGATE_REPCAT
- DBMS_REPCAT_AUTH.REVOKE_SURROGATE_REPCAT

各プロシージャについて説明します。

DBMS_REPCAT_AUTH.GRANT_SURROGATE_REPCAT

目的

アドバンスド・レプリケーション機能で必要な権限をユーザーに付与します。

構文

GRANT_SURROGATE_REPCAT プロシージャのパラメータは、表 9-203 を参照してください。例外は、表 9-204 を参照してください。このプロシージャの構文は、次のとおりです。

```
DBMS_REPCAT_AUTH.GRANT_SURROGATE_REPCAT(userid IN VARCHAR2)
```

表 9-203 GRANT_SURROGATE_REPCAT のパラメータ

パラメータ	説明
userid	必要な権限を付与するユーザーの名前。

表 9-204 GRANT_SURROGATE_REPCAT の例外

例外	説明
ORA-01917	ユーザーが存在しません。

DBMS_REPCAT_AUTH.REVOKE_SURROGATE_REPCAT

目的

代理レプリケーション管理者に付与された権限を取り消します。

構文

REVOKE_SURROGATE_REPCAT プロシージャのパラメータは、表 9-205 を参照してください。例外は、表 9-206 を参照してください。このプロシージャの構文は、次のとおりです。

```
DBMS_REPCAT_AUTH.REVOKE_SURROGATE_REPCAT(userid IN VARCHAR2)
```

表 9-205 REVOKE_SURROGATE_REPCAT のパラメータ

パラメータ	説明
userid	必要な権限を取り消すユーザーの名前。

表 9-206 REVOKE_SURROGATE_REPCAT の例外

例外	説明
ORA-01917	ユーザーが存在しません。

DBMS_REPUTIL パッケージ

DBMS_REPUTIL パッケージには次のプロシージャが含まれます。

- DBMS_REPUTIL.REPLICATION_OFF
- DBMS_REPUTIL.REPLICATION_ON

各プロシージャについて説明します。

DBMS_REPUTIL.REPLICATION_OFF

目的

レプリケート環境の他のサイトに変更をレプリケートしないで表を変更したり、プロシージャ・レプリケーションを使用しているとき行レベル・レプリケーションを使用禁止にしたりします。通常は、このフラグを設定する前に、レプリケート環境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・アクティビティを中断する必要があります。

構文

REPLICATION_OFF プロシージャの構文は、次のとおりです。このプロシージャには引数がありません。

```
DBMS_REPUTIL.REPLICATION_OFF
```

DBMS_REPUTIL.REPLICATION_ON

目的

REPLICATION_OFF をコールしてレプリケーションを一時的に中断した後、変更のレプリケーションを再び使用可能にします。

構文

REPLICATION_ON プロシージャの構文は、次のとおりです。このプロシージャには引数がありません。

```
DBMS_REPUTIL.REPLICATION_ON
```

DBMS_SNAPSHOT パッケージ

DBMS_SNAPSHOT パッケージには、次のプロシージャまたはファンクションが含まれます。

- DBMS_SNAPSHOT.BEGIN_TABLE_REORGANIZATION
- DBMS_SNAPSHOT.END_TABLE_REORGANIZATION
- DBMS_SNAPSHOT.I_AM_A_REFRESH
- DBMS_SNAPSHOT.PURGE_LOG
- DBMS_SNAPSHOT.REFRESH
- DBMS_SNAPSHOT.REGISTER_SNAPSHOT
- DBMS_SNAPSHOT.SET_I_AM_A_REFRESH
- DBMS_SNAPSHOT.UNREGISTER_SNAPSHOT

DBMS_SNAPSHOT.BEGIN_TABLE_REORGANIZATION

目的

このプロシージャは、マスター表を再編成する前にコールする必要があります。これは、リフレッシュに必要なスナップショット・データを保存するためのプロセスを実行します。

追加情報：2-30 ページの「スナップショット・ログがあるマスター表の再編成」を参照してください。

構文

BEGIN_TABLE_REORGANIZATION プロシージャのパラメータは、表 9-207 を参照してください。BEGIN_TABLE_REORGANIZATION の構文は、次のとおりです。

```
DBMS_SNAPSHOT.BEGIN_TABLE_REORGANIZATION(  
    tabowner    IN    VARCHAR2  
    tabname     IN    VARCHAR2)
```

表 9-207 BEGIN_TABLE_REORGANIZATION のパラメータ

パラメータ	説明
tabowner	再編成する表の所有者。
tabname	再編成する表の名前。

DBMS_SNAPSHOT.END_TABLE_REORGANIZATION

目的

このプロシージャは、マスター表を再編成した後にコールする必要があります。マスター表のスナップショット・データが有効であり、マスター表が適切な状態であることが確認されます。

追加情報：2-30 ページの「スナップショット・ログがあるマスター表の再編成」を参照してください。

構文

END_TABLE_REORGANIZATION のパラメータは、表 9-208 を参照してください。END_TABLE_REORGANIZATION の構文は、次のとおりです。

```
DBMS_SNAPSHOT.END_TABLE_REORGANIZATION(  
    tabowner    IN    VARCHAR2  
    tabname     IN    VARCHAR2)
```

表 9-208 END_TABLE_REORGANIZATION のパラメータ

パラメータ	説明
tabowner	再編成する表の所有者。
tabname	再編成する表の名前。

DBMS_SNAPSHOT.I_AM_A_REFRESH

目的

I_AM_REFRESH パッケージの状態の値を戻します。

構文

I_AM_A_REFRESH ファンクションには引数がありません。戻り値が TRUE の場合、各レプリケーション・トリガーが最初にこの状態をチェックするため、このセッションではスナップショットのすべてのローカル・レプリケーション・トリガーが完全に使用禁止になります。戻り値が FALSE の場合は、そのトリガーを使用できます。このプロシージャの構文は、次のとおりです。

```
DBMS_SNAPSHOT.I_AM_A_REFRESH RETURN BOOLEAN
```

DBMS_SNAPSHOT.PURGE_LOG

目的

スナップショット・ログから行を削除します。

構文

PURGE_LOG プロシージャのパラメータは、表 9-209 を参照してください。このプロシージャの構文は、次のとおりです。

```
DBMS_SNAPSHOT.PURGE_LOG(  
    master      IN    VARCHAR2,  
    num         IN    BINARY_INTEGER := 1,  
    flag        IN    VARCHAR2      := 'NOP')
```

表 9-209 PURGE_LOG のパラメータ

パラメータ	説明
master	マスター表の名前。
num	<p>行をスナップショット・ログから削除する、リフレッシュ日付が最も古いスナップショットの数。たとえば、次の文を使用すると、リフレッシュ日付が最も古い2つのスナップショットの行を削除します。</p> <pre>dbms_snapshot.purge_log('master_table', 2);</pre> <p>スナップショット・ログのすべての行を削除するには、次の例のように、削除するスナップショットに関して大きい数を指定します。</p> <pre>dbms_snapshot.purge_log('master_table', 9999);</pre> <p>この文は、MASTER_TABLEをベースにしているスナップショットが9999より少なければ、MASTER_TABLEに対応しているスナップショット・ログを完全に排除します。行がスナップショット・ログからすでに排除されている単純スナップショットは、次のリフレッシュ時に、完全にリフレッシュする必要があります。</p>
flag	<p>少なくとも1つのスナップショットのスナップショット・ログから行が削除されることを保証する場合は、DELETEを指定します。この引数は、引数 NUM の設定に上書きできます。たとえば、次の文を使用すると、スナップショット・ログに依存する行が実際に含まれている、スナップショット日付が最も古いリフレッシュ・スナップショットから行が削除されます。</p> <pre>dbms_snapshot.purge_log('master_table', 1, 'DELETE');</pre>

DBMS_SNAPSHOT.REFRESH

目的

同じリフレッシュ・グループのメンバーではない1つ以上のスナップショットをまとめてリフレッシュします。詳細は、表 9-210 を参照してください。

構文

REFRESH プロシージャのパラメータは、表 9-210 を参照してください。このプロシージャの構文は、次のとおりです。

```

DBMS_SNAPSHOT.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)

```

表 9-210 REFRESH のパラメータ

パラメータ	説明
list tab	リフレッシュするスナップショットのカンマで区切られたリスト。(シノニムはサポートされていません)。これらのスナップショットは別々のスキーマに置かれていても、別々のマスター表を持っていてもかまいませんが、リストされたスナップショットはすべて現行データベースに入っている必要があります。他の方法として、DBMS_UTILITY.UNCL_ARRAY 型の PL/SQL 表に入れて渡すことができます。各要素はスナップショットの名前です。
method	<p>リストに含まれる各スナップショットに対して実行するリフレッシュの型です。F または f は高速リフレッシュ、C または c は完全リフレッシュ、? はデフォルトのリフレッシュを示します。スナップショットの作成時にリフレッシュ・モードを指定した場合は、デフォルト・リフレッシュを指定すると、作成次のモードが使われます。モードを指定しない場合、Oracle は、高速リフレッシュを実行できるならば高速リフレッシュを、実行できないならば完全なリフレッシュを実行します。METHOD リストに入っている要素がスナップショット LIST より少ない場合、スナップショット・リスト内の後続する要素はデフォルト・リフレッシュを使用してリフレッシュされます。たとえば、SQL*Plus の内部から次の EXECUTE 文を実行します。</p> <pre> dbms_snapshot.refresh ('emp,dept,scott.salary','CF'); </pre> <p>この文は、EMP スナップショットの完全なリフレッシュ、および DEPT スナップショットの高速リフレッシュ、SCOTT.SALARY スナップショットのデフォルト・リフレッシュを実行します。</p>

表 9-210 REFRESH のパラメータ

パラメータ	説明
rollback_seg	スナップショットをリフレッシュする際に使用するスナップショット・サイトのロールバック・セグメントの名前。 REFRESH をコールすると、リストされたスナップショットのすべてが1つの時点に対して更新されます。いずれかのスナップショットが原因でリフレッシュが失敗すると、どのスナップショットも更新されません。
push_deferred_rpc	更新可能なスナップショットによってだけ使用されます。スナップショットをリフレッシュする前にスナップショットから関連マスターへの変更を送信する場合、デフォルト値のTRUEを使用します。そうしないと、これらの変更は一時的に失われたように見えることがあります。
refresh_after_errors	更新可能なスナップショットによってだけ使用されます。スナップショットのマスターの DEFERROR ビューに、未解消の競合がまだ記録されていても、リフレッシュを続行する場合は、このパラメータに TRUE を設定します。
purge_option	パラレル伝播のメカニズムを使用する場合（つまり、parallelism に 1 以上を設定する場合）の指定は次のとおりです。0 = パージなし、1 = レイジー・パージ（デフォルト）、2 = アグレッシブ。ほとんどの場合、レイジー・パージの設定が最適です。複数のマスター・レプリケーション・グループがさまざまなターゲット・サイトに送られて、1 つ以上のレプリケーション・グループに対する更新がまれであり、まれにしか送信されない場合、パージをアグレッシブに設定してキューを減らします。すべてのレプリケーション・グループがまれにしか更新されず、送信されない場合は、パージをパージなしに設定しておき、ときおりパージをアグレッシブに設定して PUSH を実行してキューを低減します。
parallelism	0 = シリアル伝播、 $n > 0 = n$ 個のパラレル・サーバー・プロセスを使用したパラレル伝播、1 = 1 つのパラレル・サーバー・プロセスだけを使用したパラレル伝播。
heap_size	パラレル伝播スケジューリングのために、同時に検査されるトランザクションの最大数。Oracle が自動的に最適なパフォーマンスのためのデフォルト設定を計算します。オラクル・サポートで指示されない限り、このパラメータを設定しないでください。

DBMS_SNAPSHOT.REGISTER_SNAPSHOT

目的

個々のスナップショットの管理を使用可能にします。

構文

REGISTER_SNAPSHOT プロシージャのパラメータは、表 9-211 を参照してください。REGISTER_SNAPSHOT プロシージャの構文は、次のとおりです。

```
DBMS_SNAPSHOT.REGISTER_SNAPSHOT(  
    snapowner    IN    VARCHAR2,  
    snapname     IN    VARCHAR2,  
    snapsite     IN    VARCHAR2,  
    snapshot_id  IN    DATE | BINARY_INTEGER,  
    flag         IN    BINARY_INTEGER,  
    qry_txt      IN    VARCHAR2,  
    rep_type     IN    BINARY_INTEGER := DBMS_SNAPSHOT.REG_UNKNOWN)
```

注意: このプロシージャはオーバーロードされます。SNAPSHOT_IDとFLAGパラメータは、両方同時に指定できません。

表 9-211 REGISTER_SNAPSHOT のパラメータ

パラメータ	説明
sowner	スナップショットの所有者。
snapname	スナップショットの名前。
snapsite	Oracle8 マスターに登録しているスナップショットのスナップショット・サイト名。このパラメータでは、二重引用符は使用できません。
snapshot_id	スナップショットの識別番号。Oracle8 のスナップショットは BINARY_INTEGER として指定します。Oracle8 マスター・サイトに登録されている Oracle7 スナップショットは、DATE として指定します。
flag	後続の作成または移動コマンドが問合せテキストに登録されているかどうかを示す、PL/SQL パッケージ変数。
query_txt	問合せの最初の 32,000 バイト。
rep_type	スナップショットのバージョン。代入できる有効な定数。reg_unknown (デフォルト) および reg_v7_group、reg_v8_group、reg_repapi_group を含みます。

注意：このプロシージャはマスター・サイトで実行され、リモート・プロシージャ・コールによって実行されます。REGISTER_SNAPSHOT が同一の SNAPOWNER および SNAPNAME、SNAPSITE を使用して複数回コールされる場合、SNAPSHOT_ID および FLAG、QUERY_TXT の最新の値が保存されます。問合せの結果が最大の VARCHAR2 サイズを超過する場合、QUERY_TXT には問合せの最初の 32000 文字が格納され、残りは切り捨てられます。手動で起動するときは、プロシージャをコールするユーザーが、SNAPSHOT_ID および FLAG の値をスナップショット・ビューで調べる必要があります。

注意：スナップショットの問合せをマスター・サイトに登録したくない場合は、オプションを FALSE に設定して SET_REGISTER_QUERY_TEXT プロシージャをコールします。最新のオプション設定を調べるには、DDL を発行する前にスナップショット・サイトで GET_REG_QUERY_TEXT_FLAG ファンクションをコールします。

DBMS_SNAPSHOT.SET_I_AM_A_REFRESH

目的

I_AM_REFRESH パッケージ状態を適切な値に設定します。

構文

SET_I_AM_A_REFRESH プロシージャのパラメータは、表 9-212 を参照してください。このプロシージャの構文は、次のとおりです。

DBMS_SNAPSHOT.SET_I_AM_A_REFRESH(value IN BOOLEAN)

表 9-212 SET_I_AM_A_REFRESH のパラメータ

パラメータ	説明
value	I_AM_A_REFRESH パッケージ状態に設定する値。この状態を TRUE に設定すると、それぞれのレプリケーション・トリガーで最初にこの状態がチェックされるので、スナップショットのすべてのローカル・レプリケーション・トリガーは、このセッションで使用禁止になります。この状態を FALSE に設定すると、これらのトリガーは使用可能になります。

DBMS_SNAPSHOT.UNREGISTER_SNAPSHOT

目的

個々のスナップショットの管理を使用可能にします。スナップショットを登録解除するためにマスター・サイトで起動します。

構文

UNREGISTER_SNAPSHOT プロシージャのパラメータは、表 9-213 を参照してください。UNREGISTER_SNAPSHOT プロシージャの構文は、次のとおりです。

DBMS_SNAPSHOT.UNREGISTER_SNAPSHOT

```
DBMS_SNAPSHOT.UNREGISTER_SNAPSHOT(  
    snapowner      IN   VARCHAR2,  
    snapname       IN   VARCHAR2,  
    snapsite       IN   VARCHAR2)
```

表 9-213 UNREGISTER_SNAPSHOT のパラメータ

パラメータ	説明
snapowner	スナップショットの所有者。
snapname	スナップショットの名前。
snapsite	スナップショット・サイトの名前。

パッケージ変数

表 9-214 で、アドバンスド・レプリケーション機能で使用するパッケージ変数について説明します。ユーザー独自のパッケージまたはトリガー内で、1 つ以上のこれらの変数の値をチェックする必要があります。

表 9-214 レプリケーション・パッケージ変数

変数	型	説明
dbms_reutil. replication_is_on	BOOLEAN	TRUE は、生成されたレプリケーション・トリガーが使用可能であることを示します。FALSE は、レプリケーションがレプリケーション・オブジェクト・グループの現行サイトで使用禁止になっていることを示します。この変数は、DBMS_REUTIL パッケージ内の REPLICATION_ON または REPLICATION_OFF プロシージャをコールすることによって設定されます。
dbms_reutil. from_remote	BOOLEAN	この変数は、\$RP 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 に設定され、これらのプロシージャの終わりで FALSE に設定されます。\$RP パッケージによる更新の結果として起動したいトリガーがある場合、この変数をチェックする必要があります。
dbms_reutil. global_name	VARCHAR2 (128)	この変数にはローカル・データベースのグローバル・データベース名が入っています。

データ・ディクショナリ・ビュー

この章では、アドバンスド・レプリケーション機能のユーザーにとって役に立つデータ・ディクショナリ・ビューについて説明します。ビューは次のカテゴリに分けられており、カテゴリごとにアルファベット順に記載されています。

- レプリケーション・カタログ・ビュー
- 遅延トランザクション・ビュー
- スナップショットおよびスナップショット・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

Oracle では、アドバンスド・レプリケーション機能をサイトにインストールすると、そのサイトの表とビューからなるレプリケーション・カタログもインストールされます。表 10-1 に示されているように、マスター・サイトおよびスナップショット・サイトはビューを使用して、どのオブジェクトがレプリケートされているのか、どこにレプリケートされているのか、またレプリケーション中にエラーが発生していないかどうかなどを判断します。レプリケーション・カタログ表は直接修正しないで、DBMS_REPCAT パッケージのプロシージャを使用して修正してください。

各ビューには 3 つのタイプがあり、特に注記しない限り、タイプによって異なる接頭辞、つまり USER_* および ALL_*、SYS.DBA_* が付いています。この項では、それらのビューの違いは無視します。

この項では、次のビューについて説明します。

- REPGROUP ビュー
- REPCATLOG ビュー
- REPCOLUMN ビュー
- REPCOLUMN_GROUP ビュー
- REPCONFLICT ビュー
- REPDDL ビュー
- REPGENERATED ビュー
- REPGROUPED_COLUMN ビュー
- REPKEY_COLUMNS ビュー
- REPOBJECT ビュー
- REPPARAMETER_COLUMN ビュー
- REPPRIORITY ビュー
- REPPRIORITY_GROUP ビュー
- REPPROP ビュー
- REPRESOLUTION ビュー
- REPRESOL_STATS_CONTROL ビュー
- REPRESOLUTION_METHOD ビュー
- REPRESOLUTION_STATISTICS ビュー
- REPSITES ビュー
- REPGENOBJECTS ビュー

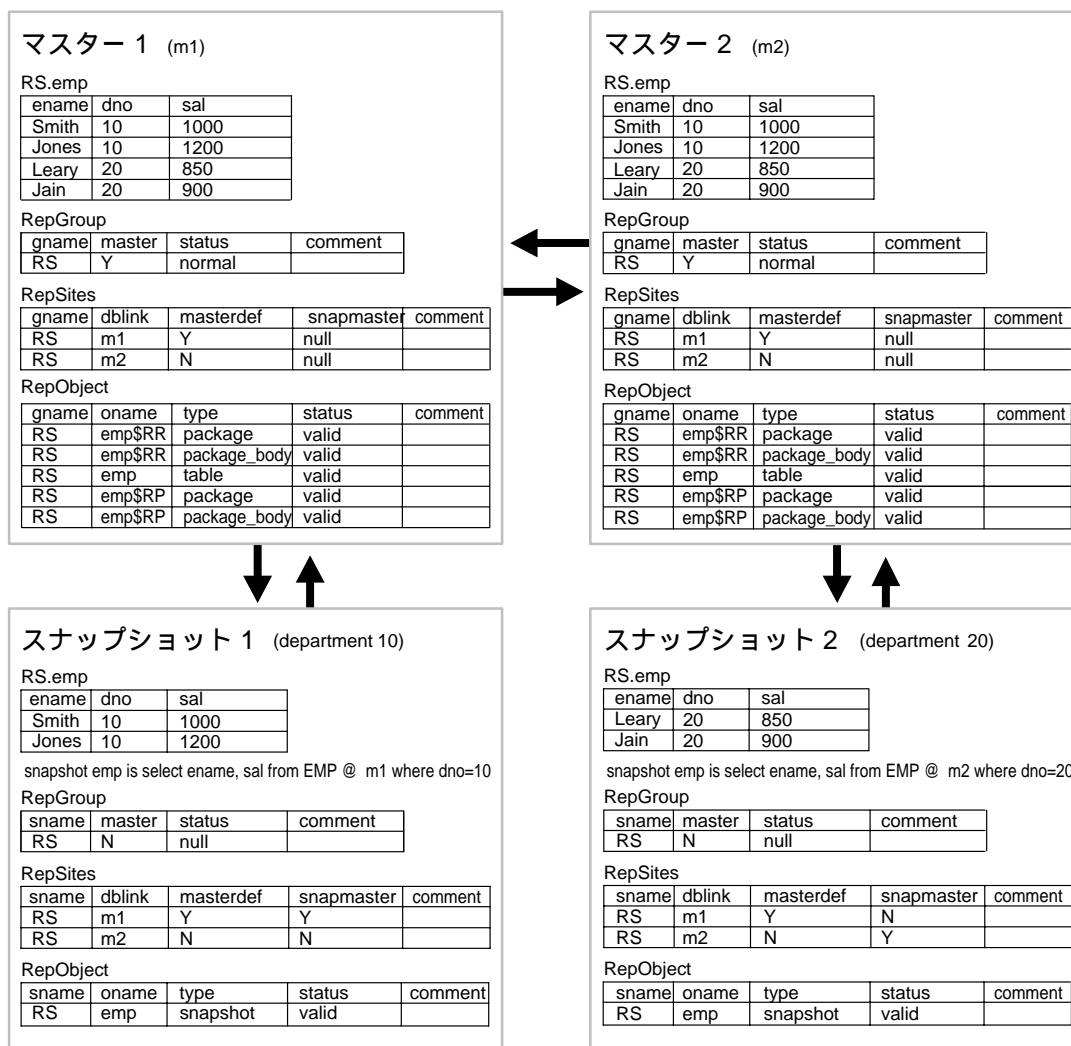
REPGROUP ビュー

REPGROUP ビューは、レプリケート中のオブジェクト・グループをすべて表示します。各オブジェクト・グループのメンバーは、別のビューである REPOBJECT に表示されます。

表 10-1 REPGROUP ビュー

列	説明
sname	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。
gname	レプリケート・オブジェクト・グループの名前。
master	'Y' は、これがマスター・サイトであることを示します。 'N' は、現行サイトがスナップショット・サイトであることを示します。
status	マスター・サイト専用。状態：正常 (normal) 休止中 (quiescing) 休止状態 (quiesced)
schema_Comment	ユーザーが入力した任意のコメント。

図 10-1 レプリケーション・カタログ・ビュー



REPCATLOG ビュー

各マスター・サイトの REPCATLOG には、非同期管理要求と生成されたエラー・メッセージの一時ステータスが含まれます。要求の実行中に検出されたすべてのメッセージは、後でその要求を出したマスターの REPCATLOG に転送されます。管理要求がエラーなしで完了すると、最終的に、その要求のトレースはすべて REPCATLOG ビューから削除されます。

表 10-2 REPCATLOG ビュー

列	説明
id	順序番号。ID および SOURCE 列によって、1 つの管理要求に関連したすべてのログ・レコードが、すべてのマスター・サイトで識別されます。
source	要求が出された位置。
userid	要求を出したユーザー ID。
timestamp	要求が出された日時。
role	サイトがマスター定義かマスターかを示します。
master	ロールがマスター定義でタスクがリモートの場合に、タスクを実行しているマスターを示します。
sname	レプリケート・オブジェクトのスキーマの名前（適用される場合）。
request	実行された DBMS_REPCAT 管理プロシージャの名前。
oname	レプリケート・オブジェクトの名前（該当する場合）。
type	レプリケート・オブジェクトの型。
status	管理要求の状態: ready または do_callback、await_callback、error。
message	戻されたすべてのエラー・メッセージ。
errnum	メッセージの Oracle エラー番号。
gname	レプリケート・オブジェクト・グループの名前。

REPCOLUMN ビュー

REPCOLUMN ビューは、表のレプリケートされた列を記述します。

表 10-3 REPCOLUMN ビュー

列	説明
sname	オブジェクト所有者の名前。
oname	オブジェクトの名前。
type	オブジェクトの型。
cname	レプリケート列の名前。
id	レプリケート列の id。
pos	レプリケート列の順序付け。
compare_old_on_delete	レプリケートされている削除内容に、Oracle がその列の古い値を送って比較するかどうかを示します。
compare_old_on_update	レプリケートされている更新内容に、Oracle がその列の古い値を送って比較するかどうかを示します。

REPCOLUMN_GROUP ビュー

REPCOLUMN_GROUP ビューは、各レプリケート表で定義したすべての列グループを記述します。

表 10-4 REPCOLUMN_GROUP ビュー

列	説明
sname	レプリケート表を含んでいるスキーマの名前。
oname	レプリケート表の名前。
group_name	列グループ名。
group_comment	ユーザーが入力した任意のコメント。

注意：sname 列は USER_ ビューにはありません。

REPCONFLICT ビュー

REPCONFLICT ビューは、競合解消方法を定義した表の名前と、解消するのに使用する競合の型を表示します。

表 10-5 REPCONFLICT ビュー

列	説明
sname	レプリケート表を含んでいるスキーマの名前。
oname	ユーザーが競合の解消方法を指定した表の名前。
conflict_type	競合解消方法に指定される競合のタイプ：削除または一意性、更新。
reference_name	ルーチンを適用するオブジェクト。削除の競合の場合、これは表名です。一意性の競合の場合、これは制約名です。更新の競合の場合、これは列グループ名です。

注意：sname 列は USER_ ビューにはありません。

REPDDL ビュー

REPDDL は、レプリケーション・オブジェクトの DDL を保持します。

表 10-6 REPDDL ビュー

列	説明
log_id	REPCAT ログ・レコードの数値を識別します。
source	要求が発行されたデータベースの名前。
role	このデータベースが要求の masterdef の場合は「 Y 」、このデータベースがマスターの場合は「 N 」。
master	この要求を処理するデータベースの名前。
line	単一要求内のレコードの配列。
text	引数または DDL テキストの一部。

REPGENERATED ビュー

REPGENERATED ビューは、システム生成されたオブジェクトについての情報を記述します。

表 10-7 REPGENERATED ビュー

列	説明
sname	オブジェクトの所有者。
oname	オブジェクト名。
type	オブジェクト型。
base_sname	実オブジェクトの所有者。
base_oname	実オブジェクトのオブジェクト名。
base_type	実オブジェクトのオブジェクト型。
package_prefix	パッケージ・ラッパー名の接頭辞。
procedure_prefix	パッケージ・ラッパー内のプロシージャに対するプロシージャ・ラッパーの接頭辞。
distributed	生成が分散されている場合は「 Y 」。 生成されたオブジェクトがクローンの場合は「 N 」。 Rep3 環境の場合は、常に「 Y 」。
reason	このオブジェクトが生成された理由。

注意：REPGENERATED ビューは廃止され、下位互換性のためだけに使用します。

REPGROUPED_COLUMN ビュー

REPGROUPED_COLUMN ビューは、各表の列グループを構成するすべての列を記述します。

表 10-8 REPGROUPED_COLUMN ビュー

列	説明
sname	レプリケート表を含んでいるスキーマの名前。
oname	レプリケート表の名前。
group_name	列グループの名前。
column_name	列グループに含まれている列の名前。

注意：sname 列はビューの USER_ パージョンにはありません。

REPKEY_COLUMNS ビュー

REPKEY_COLUMNS ビューは、主キー列に関連する情報を記述します。

表 10-9 REPKEY_COLUMNS ビュー

列	説明
sname	レプリケート表の所有者。
oname	レプリケート表の名前。
col	表内の「主キー」列名。

REPOBJECT ビュー

REPOBJECT ビューは、各レプリケート・オブジェクト・グループ内のオブジェクトについての情報を提供します。オブジェクトが所属できるのは、1つのオブジェクト・グループだけです。レプリケート・オブジェクト・グループは、複数のスキーマに所属できます。

表 10-10 REPOBJECT ビュー

列	説明
sname	レプリケート・オブジェクトを含んでいるスキーマの名前。
oname	レプリケート・オブジェクトの名前。
type	レプリケート・オブジェクトのタイプ：表またはビュー、パッケージ、パッケージ本体、プロシージャ、ファンクション、索引、シノニム、トリガー、スナップショット。
status	CREATE は、Oracle がローカルでオブジェクトを作成するために、ユーザー作成または Oracle 生成の DDL をローカル・データベースに適用していることを示します。レプリケートがローカルに存在している場合、Oracle はそのレプリケートとマスター定義を比較 (COMPARE) し、一致していることを確かめます。Oracle は、作成または比較が正常に終了した場合は状態を VALID に更新し、異常終了した場合は状態を ERROR に更新します。オブジェクトを削除した場合、Oracle では、REPOBJECT ビューから行を削除する前に、ステータスを DROPPED に更新します。
id	ローカル・データベース・オブジェクトの識別子 (存在している場合)。
object_comment	ユーザーが入力した任意のコメント。
gname	オブジェクトが所属するレプリケート・オブジェクト・グループの名前。

表 10-10 REPOBJECT ビュー

列	説明
generation_status	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す状態。
min_communication	N は、更新済みビューの OLD 値と NEW 値の両方を送ります。
trigflag	インライン・トリガー・フラグ。Y= トリガーはアクティブ; N= トリガー使用禁止。

REPPARAMETER_COLUMN ビュー

REPREOLUTION ビューに含まれる情報に加えて、REPPARAMETER_COLUMN ビューにも、競合を解消するために使用するよう指示した列についての情報が含まれます。これらの列は、DBMS_REPCAT パッケージに含められている ADD_*_RESOLUTION プロシージャの LIST_OF_COLUMN_NAMES 引数として渡される列値です。

表 10-11 REPPARAMETER_COLUMN ビュー

列	説明
sname	レプリケート表を含んでいるスキーマの名前。
oname	レプリケート表の名前。
conflict_type	ルーチンが解消する競合のタイプ: 削除または一意、更新。
reference_name	ルーチンを適用するオブジェクト。削除の競合の場合、これは表名です。一意性の競合の場合、これは制約名です。更新の競合の場合、これは列グループ名です。
sequence_no	解消方法を適用する順序。1 が最初に適用されます。
method_name	Oracle が提供した競合解消方法の名前。ユーザーが提供した方法の場合、この値は user function になります。
function_name	ユーザーが提供した競合解消ルーチンの名前 (user function タイプの方法の場合)。
priority_group	優先順位グループの名前 (priority group タイプの方法の場合)。
parameter_table_name	デフォルトは、競合解消機能に渡される列を含む PL/SQL 表のオブジェクト名です。
parameter_column_name	競合解消ルーチンの IN パラメータとして使用される列の名前。
parameter_sequence_no	IN パラメータとして使用される列の順序付け。

注意：SNAME 列は USER_ ビューにはありません。

REPPRIORITY ビュー

REPPRIORITY ビューは、各優先グループ・メンバーの値と優先順位レベルを表示します。優先順位グループの名前は、レプリケート・オブジェクト・グループの中では一意でなければなりません。優先順位および値は、優先順位グループの中では一意でなければなりません。

表 10-12 REPPRIORITY ビュー

列	説明
sname	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。
gname	レプリケート・オブジェクト・グループの名前。
priority_group	優先順位グループまたはサイト優先順位グループの名前。
priority	メンバーの優先順位。数値が大きいほど優先順位は高くなります。
data_type	優先順位グループに含まれている値のデータ型。
fixed_data_length	データ型が CHAR の値の最大長。
char_value	優先順位グループ・メンバーの値 (data_type = char の場合)
varchar2_value	優先順位グループ・メンバーの値 (data_type = varchar2 の場合)
number_value	優先順位グループ・メンバーの値 (data_type = number の場合)
date_value	優先順位グループ・メンバーの値 (data_type = date の場合)
raw_value	優先順位グループ・メンバーの値 (data_type = raw の場合)
nchar_value	優先順位グループ・メンバーの値 (data_type = nchar の場合)
nvarchar2_value	優先順位グループ・メンバーの値 (data_type = nvarchar2 の場合)
large_char_value	優先順位グループ・メンバーの値、255 文字を超える空白埋め文字列用。

注意：SNAME 列と GNAME 列は USER_ ビューにはありません。

REPPRIORITY_GROUP ビュー

REPPRIORITY_GROUP ビューは、レプリケート・オブジェクト・グループで定義した優先順位およびサイト優先順位グループを記述します。

表 10-13 REPPRIORITY_GROUP ビュー

列	説明
sname	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。USER ビューには表示されません。
gname	レプリケート・オブジェクト・グループの名前。USER ビューには表示されません。
priority_group	優先順位グループまたはサイト優先順位グループの名前。
data_type	優先順位グループに含まれている値のデータ型。
fixed_data_length	データ型が CHAR の値の最大長。
priority_comment	ユーザーが入力した任意のコメント。

REPPROP ビュー

REPPROP ビューは、あるオブジェクトに対する操作を、別のマスター・サイトにある同じオブジェクトに伝播する方法を示します。この操作は、ストアド・プロシージャまたはプロシージャ・ラップ・プログラムをコールするときに行われたり、表に対して直接発行されたりします。

表 10-14 REPPROP ビュー

列	説明
sname	レプリケート・オブジェクトを含んでいるスキーマの名前。
oname	レプリケート・オブジェクトの名前。
type	レプリケート中のオブジェクトのタイプ。
dblink	変更を伝播させるマスター・サイトの省略のない形のデータベース名。
how	伝播を実行する方法。認識される値は、ローカル・マスター・サイトの場合は「none」、その他のすべてについては「synchronous」または「asynchronous」です。
propagate_comment	ユーザーが入力した任意のコメント。

REPRESOLUTION ビュー

REPRESOLUTION ビューは、指定のスキーマの行レベルのレプリケーションを使用してレプリケートした各表について、更新または一意性、削除の競合を解消するのに使用するルーチンを示します。

表 10-15 REPRESOLUTION ビュー

列	説明
sname	レプリケート・スキーマの名前。
oname	レプリケート表の名前。
conflict_type	ルーチンが解消する競合のタイプ: 削除または一意、更新。
reference_name	ルーチンを適用するオブジェクト。削除の競合の場合、これは表名です。一意性の競合の場合、これは制約名です。更新の競合の場合、これは列グループ名です。
sequence_no	解消方法を適用する順序。1 が最初に適用されます。
method_name	Oracle が提供した競合解消方法の名前。ユーザーが提供した方法の場合、この値は user function になります。
function_name	ユーザーが提供した競合解消ルーチンの名前 (user function タイプの方法の場合)。
priority_group	優先順位グループの名前 (priority group タイプの方法の場合)。
resolution_ Comment	ユーザーが入力した任意のコメント。

注意 : SNAME 列は USER_ ビューにはありません。

REPRESOL_STATS_CONTROL ビュー

REPRESOL_STATS_CONTROL ビューは、データベース内のすべてのレプリケート表について、競合解消の統計収集についての情報を記述します。

表 10-16 REPRESOL_STATS_CONTROL ビュー

列	説明
sname	表の所有者。
oname	表名。
created	統計収集が最初に開始されたときのタイムスタンプ。
status	統計収集のステータス (ACTIVE または CANCELLED)
status_update_date	ステータスが最後に更新されたときのタイムスタンプ。
purged_date	統計データを最後にパージしたときのタイムスタンプ。
last_purged_start_date	統計パージ日付範囲の最後の開始日。
last_purged_end_date	統計パージ日付範囲の最後の終了日。

注意 : SNAME 列は USER_ ビューにはありません。

REPRESOLUTION_METHOD ビュー

REPRESOLUTION_METHOD ビューは、現在のデータベースで使用可能な競合解消ルーチンをすべて記述します。最初、このビューはアドバンスト・レプリケーション機能の標準ルーチンを記述します。ユーザー関数を新たに作成し、それをデータベース内のオブジェクトの競合解消方法として追加する場合、その関数はこのビューに追加されます。

表 10-17 REPRESOLUTION_METHOD ビュー

列	説明
conflict_type	設計されている解消ルーチンが解消する競合のタイプ : 削除または一意、更新。
method_name	Oracle が提供した方法またはユーザーが定義したルーチンの名前。

REPRESOLUTION_STATISTICS ビュー

REPRESOLUTION_STATISTICS ビューは、すべてのレプリケート表について、正常に解消した更新および一意性、削除の競合についての情報を記述します。これらの表の統計は、DBMS_REPCAT.REGISTER_STATISTICS がコールされたときだけ生成します。

表 10-18 REPRESOLUTION_STATISTICS ビュー

列	説明
sname	レプリケート・スキーマの名前。
oname	レプリケート表の名前。
conflict_type	正常に解消された競合のタイプ: 削除または一意、更新。
reference_name	競合解消ルーチンが適用されるオブジェクト。削除の競合の場合、これは表名です。一意性の競合の場合、これは制約名です。更新の競合の場合、これは列グループ名です。
method_name	Oracle が提供した競合解消方法の名前。ユーザーが提供した方法の場合、この値は user function になります。
function_name	ユーザーが定義した競合解消ルーチンの名前 (user function タイプの方法の場合)。
priority_group	優先順位グループの名前 (priority group タイプの方法の場合)。
primary_key_value	該当行の主キーの連結表示。
resolved_date	競合が解消された日付。

注意 : SNAME 列は USER_ ビューにはありません。

REPSITES ビュー

REPSITES ビューは、各レプリケート・オブジェクト・グループのメンバーを記述します。

表 10-19 USER および ALL、DBA REPSITES ビューの列

列	説明
gname	レプリケート・オブジェクト・グループの名前。
dblink	このオブジェクト・グループのマスター・サイトに至るデータベース・リンク。
masterdef	マスター定義サイトになっているデータベース・リンクを示します。
snapmaster	リフレッシュ時に使用するデータベース・リンクを示すためにスナップショット・サイトが使います。
master_Comment	ユーザーが入力したコメント。
master	サイトがレプリケート・グループのマスター・サイトかどうか。Y または N。

DBA_REPSITES ビューには次に示すその他の列があります。

prop_updates	マスターに対する更新を要求された数。
my_dblink	インポート後に問題を検出するために使います。Y の場合、データベース・リンクがグローバル名です。

REPGENOBJECTS ビュー

REPGENOBJECTS ビューは、レプリケーションをサポートするために生成されたオブジェクトについて説明します。

表 10-20 REPGENOBJECTS ビュー

列	説明
Base_otype	このオブジェクトが生成された対象のオブジェクト。
Base_sname	基本オブジェクトの所有者。
Base type	基本オブジェクトの型。
Distributed	廃棄されました。
Oname	生成されたオブジェクトの名前。
Package-prefix	パッケージ Wrapper の接頭辞。
Procedure-prefix	使われません。

表 10-20 REPGENOBJECTS ビュー

列	説明
Reason	オブジェクトが生成された理由。
Sname	レプリケート・スキーマの名前。
Type	生成されたオブジェクトの型。

遅延トランザクション・ビュー

Oracle は、遅延トランザクションの管理で使用するビューをいくつか用意しています。これらの表によって、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実行中に生じたエラーなど、それぞれの遅延トランザクションに関する情報が得られます。表は直接修正しないで、DBMS_DEFERおよびDBMS_DEFER_SYSパッケージのプロシージャを使用して修正してください。

- DEFCALL ビュー
- DEFDEFAULTDEST ビュー
- DEFERRCOUNT ビュー
- DEFCALLDEST ビュー
- DEFERROR ビュー
- DEFLOB ビュー
- DEFPROPAGATOR ビュー
- DEFSCHEDULE ビュー
- DEFTRAN ビュー
- DEFTRANDEST ビュー

DEFCALL ビュー

DEFCALL ビューは、すべての遅延リモート・プロシージャ・コールを記録します。

表 10-21 DEFCALL ビュー

列	説明
callno	トランザクション内のコールの一意 ID。
deferred_tran_id	対応するトランザクションの一意 ID。
schemaname	遅延コールのスキーマ名。
packagename	遅延コールのパッケージ名。
procname	遅延コールのプロシージャ名。
argcount	遅延コールに対する引数の数。

DEFCALLDEST ビュー

DEFCALLDEST ビューは、各遅延リモート・プロシージャ・コールの接続リストを表示します。

表 10-22 DEFCALLDEST ビュー

列	説明
callno	トランザクション内のコールの一意 ID。
deferred_tran_id	DEFTRAN ビューの deferred_tran_id に対応します。それぞれの遅延トランザクションは、1 つまたはそれ以上の遅延コールから成り立っています。
dblink	接続先データベースの省略のない形のデータベース名。

DEFDEFAULTDEST ビュー

Oracle のレプリケーション機能を使用せず、遅延トランザクションまたはそのトランザクション内のコールの接続先を提供しない場合、Oracle では、DEFDEFAULTDEST ビューを使用して、遅延するリモート・プロシージャ・コールへの接続先データベースを判断します。

表 10-23 DEFDEFAULTDEST ビュー

列	説明
dblink	トランザクションをレプリケートするデータベースの省略のない形の名前。

DEFERRCOUNT ビュー

DEFERRCOUNT ビューは、指定の接続先のエラー・トランザクションについての情報を提供します。

表 10-24 DEFERRCOUNT ビュー

列	説明
errcount	接続先に対してエラーが発生した既存のトランザクション数。
destination	接続のアドレスを指定するためのデータベース・リンク。

DEFERROR ビュー

DEFERROR ビューは、適用できなかった各トランザクションの ID を提供します。この ID を用いて、キューに入っているトランザクションに関係したコールを見つけます。これらのコールは、DEFCALL ビューに保存されます。DBMS_DEFER_QUERY パッケージのプロシージャを使用して、DEFCALL ビューに表示されているプロシージャの引数を判別できます。

表 10-25 DEFERROR ビュー

列	説明
deferred_tran_id	エラーが発生したトランザクションの ID。
callno	deferred_tran_db でコールされた一意 ID。
destination	接続のアドレスを指定するためのデータベース・リンク。
error_number	Oracle エラー番号。
error_msg	エラー・メッセージ・テキスト。
receiver	遅延トランザクションの元のレシーバ。
origin_tran_db	遅延トランザクションが発行されたデータベース。
origin_tran_id	トランザクションの元の ID。
start_time	元のトランザクションが待機状態になった時間。

DEFLOB ビュー

DEFLOB ビューは、遅延 RPC への LOB パラメータを格納します。

表 10-26 DEFLOB ビュー

列	説明
id	LOB パラメータの識別子。
deferred_tran_id	この LOB パラメータをもつ遅延 RPC のトランザクション ID。
blob_col	バイナリ型の LOB パラメータ。
clob_col	文字型の LOB パラメータ。
nclob_col	各国文字型の LOB パラメータ。

DEFPROPAGATOR ビュー

DEFPROPAGATOR ビューは、ローカル伝播担当者についての情報を表示します。

表 10-27 DEFPROPAGATOR ビュー

列	説明
username	伝播担当者のユーザー名。
userid	伝播担当者のユーザー ID。
status	伝播担当者のステータス。
created	伝播担当者が登録された時間。

DEFSCHEDULE ビュー

DEFSCHEDULE ビューは、ジョブが次回実行される予定の時点についての情報を表示します。

表 10-28 DEFSCHEDULE ビュー

列	説明
dblink	遅延リモート・プロシージャ・コールの周期的な実行が設定されているマスター・データベース・サイトの省略のない形の名前。
job	DBMS_DEFER_SYS.SCHEDULE_EXECUTION を使用してジョブを作成したときに割り当てた番号。USER_JOBS ビューの WHAT 列を問い合せて、ジョブによって何が実行されるのかを判断してください。
interval	遅延トランザクション・キューを接続先へ送信する次の時間を計算する関数。
next_date	ジョブの次の実行予定日付。
last_date	この接続先へのリモート・プロシージャ・コールで、最後にキューが送信された（または、送信を試みた）時間。
disabled	接続先への伝播が使用禁止になっているかどうか。
last_txn_count	最後の試行で送信されたトランザクションの数。
last_error_number	最後の送信で戻された Oracle エラー番号。
last_error_message	最後の送信で戻されたエラー・メッセージ。

DEFTRAN ビュー

DEFTRAN ビューは、すべての遅延トランザクションを記録します。

表 10-29 DEFTRAN ビュー

列	説明
deferred_tran_id	コールを待機状態にしたトランザクション ID。
delivery_order	キューの中にある遅延トランザクションの順番を決める識別子。この識別子は、トランザクションの発行またはコピーに与えられたシステム実行番号からとられています。
destination_list	R または D。「R」は、接続先が REPSchema ビューによって判断されることを示します。「D」は、接続先が DEF-DEFAULTDEST ビュー、または、TRANSACTION または CALL、COPY プロシージャへの引数 NODE_LIST によって判断されたことを示します。
start_time	元のトランザクションが待機状態になった時間。

DEFTRANDEST ビュー

DEFTRANDEST ビューは、遅延トランザクションの接続先リストを表示します。

表 10-30 DEFTRANDEST ビュー

列	説明
deferred_tran_id	指定のデータベース・リンクへレプリケートするトランザクションのトランザクション ID。
dblink	接続先データベースの省略のない形のデータベース名。
delivery_order	キューの中にある遅延トランザクションの順番を決める識別子。この識別子は、トランザクションの発行またはコピーに与えられたシステム実行番号からとられています。

スナップショットおよびスナップショット・リフレッシュ・グループ・ビュー

次のビューには、スナップショットおよびスナップショット・リフレッシュ・グループの情報が含まれています。

- SNAPSHOTS ビュー
- REGISTERED_SNAPSHOTS ビュー
- SNAPSHOTS_LOGS ビュー

- SNAPSHOT_REFRESH_TIMES ビュー
- REFRESH ビュー
- REFRESH_CHILDREN ビュー

SNAPSHOTS ビュー

SNAPSHOTS カタログ・ビューは、データベース内のスナップショットすべてに関する情報を示します。

表 10-31 SNAPSHOTS ビュー

列	説明
OWNER	スナップショットの所有者。
NAME	スナップショットの問合せまたは更新を行うために、ユーザーおよびアプリケーションが使用するビューの名前。
TABLE_NAME	スナップショットが格納されている表（その表の中には、マスター ROWID 用の列が追加されています）。
MASTER_VIEW	リフレッシュで使用するマスター表（スナップショット所有者が所有している）のビュー。
MASTER_OWNER	マスター表の所有者。
MASTER	スナップショットがレプリケートするマスター表。
MASTER_LINK	マスター・サイト用のデータベース・リンク名。
CAN_USE_LOG	スナップショットがスナップショット・ログを使用できる場合は YES、ログを使用するにはスナップショットが複雑すぎる場合は NO です。
UPDATABLE	「 YES 」は、スナップショットが更新可能であることを示します。「 NO 」は、読み込み専用であることを示します。
LAST_REFRESH	マスター・サイトでリフレッシュが最後に実行された日付と時刻。
ERROR	最後の正常実行以降、失敗した実行の数。
TYPE	自動リフレッシュでリフレッシュするタイプ： COMPLETE、FAST、FORCE。
NEXT	次のリフレッシュ日付を計算するための日付関数。
START_WITH	次のリフレッシュ日付を計算するための日付関数。
REFRESH_METHOD	スナップショットの高速または完全リフレッシュを駆動するために使用する値。

表 10-31 SNAPSHOTS ビュー

列	説明
FR_OPERATIONS	「 REGENERATE 」の場合は、高速リフレッシュ操作が生成されていません。
CR_OPERATIONS	「 REGENERATE 」の場合は、完全リフレッシュ操作が生成されていません。
MASTER_ROLLBACK_SEG	マスターでのリフレッシュ時に使われるロールバック・セグメント。
REFRESH_GROUP	一貫したリフレッシュ用のグループ識別子。
UPDATE_TRIG	更新可能なスナップショットのUPDATE_LOGを挿入するトリガーの名前。
UPDATE_LOG	更新可能なスナップショットへの変更を記録する表の名前。
QUERY	スナップショットを作成するための問合せ。

注意：Oracle8 では、トリガーが内部化したため、UPDATE_TRIG は NULL です。また、Oracle8 では、MASTER_VIEW の NULL の指定は廃止されました。

REGISTERED_SNAPSHOTS ビュー

このビューは、ローカル表のローカルまたはリモート・スナップショットを示します。

表 10-32 REGISTERED_SNAPSHOTS ビュー

列	説明
OWNER	スナップショットの所有者。
NAME	スナップショットの名前。
SNAPSHOT_SITE	スナップショット・サイトのグローバル名。
CAN_USE_LOG	NO の場合、スナップショットが複合で、高速リフレッシュはできません。
UPDATABLE	NO の場合、スナップショットが読み専用です。
REFRESH_METHOD	スナップショットは、ROWID、または高速リフレッシュの主キーを使用します。
SNAPSHOT_ID	高速リフレッシュのためにマスターによって使われるスナップショットの識別子。
QUERY_TXT	スナップショットを定義する問合せ（登録されている場合）。
VERSION	スナップショットのバージョン。

SNAPSHOTS_LOGS ビュー

SNAPSHOT_LOGS ビューは、データベース内のすべてのスナップショット・ログを示します。

表 10-33 SNAPSHOT_LOGS ビュー

列	説明
LOG_OWNER	スナップショット・ログの所有者。
MASTER	変更が記録されるマスター表の名前。
LOG_TABLE	マスターで変更された行の ROWID とタイムスタンプをもつログ表。
LOG_TRIGGER	ログに行を挿入するマスター上の after-row トリガー。内部化トリガーのため Oracle8 では NULL。
ROWIDS	YES の場合、スナップショット・ログに ROWID 情報が記録されます。
PRIMARY_KEY	YES の場合、スナップショット・ログに主キー情報が記録されます。
FILTER_COLUMNS	YES の場合、スナップショット・ログにフィルタ列情報が記録されます。
CURRENT_SNAPSHOTS	スナップショットごとに1つの日付。マスターのスナップショットが最後にリフレッシュされた日付。
SNAPSHOT_ID	スナップショットの一意識別子。

注意：ビューでは、ログを使用して、スナップショットごとに1行を表示します。ただし、マスター・サイトにあるすべてのスナップショットについては1行しかありません。使われているログを調べるには、一意性を使用して、SNAPSHOT_ID と CURRENT_SNAPSHOTS を選択しないで、USER_SNAPSHOT_LOGS を問い合わせます。

SNAPSHOT_REFRESH_TIMES ビュー

REFRESH_TIMES ビューは、リフレッシュが最後に実行された日付と時刻を表示します。

表 10-34 SNAPSHOT_REFRESH_TIMES ビュー

列	説明
OWNER	スナップショットの所有者。
NAME	スナップショット・ビューの名前。
MASTER_OWNER	マスター表の所有者。
MASTER	マスター表の名前。
LAST_REFRESH	リフレッシュが最後に実行された日付と時刻。

REFRESH ビュー

REFRESHビューは、データベース内の各リフレッシュ・グループを表示し、各グループのリフレッシュ間隔の情報を表示します。

表 10-35 REFRESH ビュー

列	説明
ROWNER	リフレッシュ・グループの所有者。
RNAME	リフレッシュ・グループの名前。
REFGROUP	リフレッシュ・グループの ID 番号。
IMPLICIT_DESTROY	暗黙の削除フラグ。この値が Y になっていると、リフレッシュ・グループに残っている最後のメンバーが削除された後で、Oracle はそのグループを削除します。
JOB	スナップショット・グループの自動リフレッシュを実行するジョブの ID 番号。この情報によって、USER_JOBS ビューに問い合わせたジョブの詳細を調べます。
NEXT_DATE	グループのメンバーが次回リフレッシュされる日付。
INTERVAL	リフレッシュの間隔を計算するための関数。
BROKEN	リフレッシュ・グループの問題を示すフラグ。中断フラグの値が Y になっている場合、Oracle は、リフレッシュが予定されているグループでもリフレッシュしません。

表 10-35 REFRESH ビュー

列	説明
PUSH_DEFERRED_RPC	Y の場合、リフレッシュの前に変更内容をマスターに送ります。
REFRESH_AFTER_ERRORS	Y の場合、遅延 RPC を送るときにエラーが発生しても、リフレッシュを続行します。
ROLLBACK_SEG	リフレッシュ時にスナップショット・サイトで使われるロールバック・セグメントの名前。
PURGE_OPTION	各送信後に、トランザクション・キューをパージする方法。
PARALLELISM	トランザクション伝播の並列性レベル。
HEAP_SIZE	トランザクション伝播に使用するヒープ・サイズ。

REFRESH_CHILDREN ビュー

REFRESH_CHILDREN ビューは、ユーザーが所有している各リフレッシュ・グループのメンバーを表示し、各メンバーのリフレッシュ間隔についての情報を表示します。

表 10-36 REFRESH_CHILDREN ビュー

列	説明
OWNER	リフレッシュ・グループ・メンバーの所有者。
NAME	リフレッシュ・グループ・メンバーの名前。
TYPE	リフレッシュ・グループ・メンバーのタイプ: たとえば、SNAPSHOT。
ROWNER	リフレッシュ・グループの所有者。
RNAME	リフレッシュ・グループの名前。
REFGROUP	リフレッシュ・グループの ID 番号。
IMPLICIT_DESTROY	暗黙の削除フラグ。この値が Y になっていると、リフレッシュ・グループに残っている最後のメンバーが削除された後で、Oracle はそのグループを削除します。
JOB	スナップショット・リフレッシュ・グループの自動リフレッシュを実行するジョブの ID 番号。この情報によって、USER_JOBS ビューに問い合わせるジョブの詳細を調べます。
NEXT_DATE	グループのメンバーが次回リフレッシュされる日付。
INTERVAL	リフレッシュの間隔を計算するための関数。
BROKEN	リフレッシュ・グループの問題を示すフラグ。中断フラグの値が Y になっている場合、Oracle は、リフレッシュが予定されているグループでもリフレッシュしません。
PUSH_DEFERRED_RPC	Y の場合、リフレッシュの前に変更内容をマスターに送ります。
REFRESH_AFTER_ERRORS	Y の場合、遅延 RPC を送るときにエラーが発生しても、リフレッシュを続行します。
ROLLBACK_SEG	リフレッシュ時にスナップショット・サイトで使われるロールバック・セグメントの名前。
PURGE_OPTION	各送信後に、トランザクション・キューをパージする方法。
PARALLELISM	トランザクション伝播の並列性レベル。
HEAP_SIZE	トランザクション伝播に使用するヒープ・サイズ。

この付録では、Oracle8 Server の新しいレプリケーション機能について簡単に説明し、このマニュアルの追加情報が示されている章への参照情報を示します。

新機能には次のものがあります。

- パフォーマンスの拡張
- 副問合せに基づくデータのサブセット化
- 拡張されたシステム・ベースのセキュリティ・モデル
- Replication Manager の新しい機能

パフォーマンスの拡張

Oracle8 では、次の新機能によってパフォーマンスが大幅に改善されました。

遅延トランザクションの平行伝播

一貫性とトランザクションの依存関係を維持しながら、レプリケーション・トランザクション・ストリームの伝播を並列化することによって、スループット・パフォーマンスが大幅に向上しました。

追加情報：3-10 ページの「シリアル伝播および平行伝播の決定」を参照してください。

内部化されたレプリケーション・トリガー

内部トリガーを使用すると、次の利点があります。

- 応答時間パフォーマンスの改善
- 処理オーバーヘッドの削減
- 管理作業の低減

追加情報：1-17 ページの「Oracle のアドバンスド・レプリケーション・アーキテクチャ」を参照してください。

データ伝播の縮小

Oracle8 により、ネットワーク上に伝播したレプリケート・データの量が縮小されます。伝播が縮小できるのは、更新される列の新しい値、および競合の検出と解消のために必要になる列の古い値、主キー値だけです。この機能は、LOB のレプリケート時のパフォーマンスを改善するのに重要です。

追加情報：5-34 ページの「更新の競合解消のためのデータ伝播の最小化」を参照してください。

副問合せに基づくデータのサブセット化

特定の種類の副問合せによって定義されるスナップショットは、高速でリフレッシュできます。これにより、データのサブセットの定義と維持が容易になりました。この機能は、販売部門自動化や支部自動化などの、大量配置アプリケーションにとって重要です。

追加情報：2-15 ページの「副問合せを使用するスナップショットの作成」を参照してください。

大容量オブジェクト・データ型（LOB）のサポート

Oracle8 では、次の型の大容量オブジェクトのレプリケーションをサポートします。

- バイナリ型の LOB（BLOB）
- 文字型の LOB（CLOB）
- 各国語サポート型の LOB（NCLOB）

追加情報：3-18 ページの「レプリケート表のデータ型の考慮」および 4-16 ページの「スナップショットのデータ型の考慮」を参照してください。

管理方法の改善と容易な使用性

Oracle8 では、次の機能がデータベース管理に役立ちます。

きめ細かな休止

レプリケーション・マスター・グループが、他のレプリケーション・グループに影響を与えないで個々に休止できます。マスター・グループでは、他のマスター・グループが休止されている間も、引き続き更新のレプリケートが可能で。

追加情報：3-15 ページの「マスター・グループのレプリケーション・アクティビティの中断」を参照してください。

主キー・スナップショット

主キー・スナップショットを使用すると、高速リフレッシュ機能を保ちながら、マスター表を再編成できます。Oracle8 では、デフォルトで主キーを追加するとともに、引続き ROWID スナップショットをサポートしています。

追加情報：2-10 ページの「単純スナップショットの作成」および付録 B「移行および互換性」を参照してください。

マスター・サイトでのスナップショット登録

Oracle8 では、スナップショットの情報がマスター・サイトに自動的に登録されます。これは、監視および分散管理に役立ちます。

追加情報: 2-34ページの「マスター・サイトでのスナップショットの登録」を参照してください。

高速リフレッシュ機能での表の再編成

Oracle8 では、マスター・スナップショット・ログの一貫性を保ちながらマスター表を " 再編成 " することができるユーティリティが用意されています。

追加情報: 2-42ページの「マスター表の再編成とROWIDスナップショット」を参照してください。

オフライン・インスタンスエーションのサポート

エクスポート/インポートを使用するスキーマおよびデータベースのオフライン・インスタンスエーションがさらに自動化されました。

追加情報: 7-13ページの「スナップショット・クローニングとオフライン・インスタンスエーション」を参照してください。

更新可能スナップショットの遅延制約

更新可能スナップショットで、宣言参照制約および一意性制約をサポートしています。

追加情報: 3-18ページの「オブジェクト定義のマスター・サイトへのレプリケート」を参照してください。

検査プロシージャ

新しいプロシージャが複数のマスター環境の正確さを検査し、エラーの回避に役立ちます。

追加情報: 6-2ページの「マスター・グループの妥当性検査」を参照してください。

パーティション表と索引

Oracle8 では、パーティション表および索引のレプリケーションをサポートしています。マスター定義サイトの表と同一のパーティションをレプリケート表に作成したい場合にこの機能を使います。

拡張されたシステム・ベースのセキュリティ・モデル

Oracle8 のシステム・ベースのセキュリティ・モデルでは次のことが可能です。

- 一貫性の向上：単一のシステム・レベル・モデルは、同期環境および非同期環境の両方で同一の動作をします。
- 信頼性の向上：受信側サイトでの権限の不足によるトランザクションの失敗が削減されました。
- リンクの単純化：単一のユーザーが repadmin および repsys として行動できます。
- 「ユーザー・レベル」モデルのニーズを排除します。
- 1名以上のスナップショット所有者がスナップショットのリフレッシュを実行できます。

追加情報: 3-3ページの「マルチマスター・レプリケーションの設定」および4-4ページの「スナップショット・サイト・レプリケーションの準備」、付録 B 「移行および互換性」を参照してください。

Replication Manager の新しい機能

Replication Managerには現在、システムをすばやく構成できるようにするウィザードがいくつか用意されています。

- アカウントおよびスキーマ、リンクの作成を含むマルチマスター構成の設定
- アカウントおよびスキーマ、リンクの作成を含むスナップショット・サイト構成の設定
- スナップショット・グループの設定
- スナップショットの設定

Oracle Replication Managerでは、次のようなOracle8 Serverレプリケーション機能の新機能の大部分をサポートしています。

- 各サイトのデータベースのバージョンの自動認識、および 8.x と 7.3 の両方のデータベースを管理する動的構成
- スナップショット・ログの管理のサポート
- ジョブおよびリフレッシュ・グループ、スケジュール・リンクの日付式および間隔式の簡単な入力
- Oracle8 のデータベースの場合、レプリケーション・サポートの再生成が必要な表を示すフラグ
- サイトのレプリケーション管理者および伝播担当者の登録のサポート
- マスター・グループのデータベース・オブジェクトを表示するデータベース・オブジェクト・フォルダ
- マスター・グループおよびスナップショット・リフレッシュ・グループのドラッグアンドドロップ機能
- すべてのマスター・サイトのマスター・グループの妥当性チェックのサポート

移行および互換性

この付録では、Oracle7 から Oracle8 へのレプリケーション環境の移行に必要な手順を説明します。次のトピックが含まれます。

- 移行の概要
- すべてのサイトの一括移行
- 段階的な移行
- 主キー・スナップショットへのアップグレード
- Oracle8 への移行を必要とする機能
- 旧プロシージャ

移行の概要

環境や特にマルチマスター・コンポーネントを 1 つのステップで移行するのが最も簡単な場合があります。通常、これは小規模な構成の場合だけ可能です。また、既存の Oracle7 レプリケーション環境を Oracle8 へ段階的に移行することもできます。Oracle7 と Oracle8 が共存しているレプリケーション環境でも、レプリケーション操作および管理操作を正常に実行できます。

ただし、Oracle7 と Oracle8 を正常に相互運用するには、次の制限事項を守る必要があります。

- Oracle8 のスナップショット・サイトは、Oracle7 リリース 7.3.3 以降のマスター・サイトとしか相互運用できない。
- Oracle8 のマスター・サイトは、Oracle7 リリース 7.3.4 以降のスナップショット・サイト、および Oracle7 リリース 7.3.3 以降のマスター・サイトとしか相互運用できない。

マスター・サイトを Oracle8 に移行した後で、対応付けられたすべてのスナップショット・サイトを完全にリフレッシュしてください。

Oracle8 から Oracle7 へのダウングレードはサポートされません。

特定の Oracle8 レプリケーション機能は、すべてのサイトが正しく移行されないと使用できません。たとえば、主キー・スナップショットを使用するには、スナップショット・サイトおよびそれに対応付けられたマスター・サイトの両方を Oracle8 に移行する必要があります。副問合せ機能とマスター表再編成プロシージャを持つ Oracle8 単純スナップショットでは、最初に、ROWID スナップショットから主キー・スナップショットへのアップグレードが必要です。

Oracle7 からの全データベース・エクスポートおよび Oracle8 への全データベース・インポートを使用する移行もサポートされています。

すべてのサイトの一括移行

この項では、マルチマスター環境全体を一括して Oracle8 に移行する方法について説明します。Oracle8 に移行しないスナップショット・サイトもすべて Oracle7 リリース 7.3.4 以降にアップグレードしなければなりません。

すべてのマスター・サイトおよびスナップショット・サイト（オプション）を一括して移行するには、次のようにしてください。

1. DBMS_REPCAT を実行してレプリケーション環境を停止する。すべてのマスター・レプリケーション・グループのマスター定義サイトで、SUSPEND_MASTER_ACTIVITY を実行し、次に、たとえばスナップショット・サイトでの自動的な伝播とリフレッシュを制御するジョブ・キューへのエントリを一時的に保留または「ブレーク」することによって、スナップショット・サイトからマスターへのすべての伝播とリフレッシュを停止します。また、ローカル・エラー・キューにあるエラーは、そのキューが空になるまで、解決し再実行する必要があります。詳細は、『Oracle7 Server 分散システム Vol.2: レプリケート・データ』の第4章の「DML 変更の非同期伝播」および「レプリケーション・アクティビティの中断」、第7章の「エラーの手動による解決」を参照してください。
2. 『Oracle8 Server 移行ガイド』で説明されているように、Oracle7 から Oracle8 への移行ユーティリティとレプリケーションCATREP8M.SQLスクリプトを使用してすべてのマスター・サイトを移行する。
3. Replication Managerのセットアップ・ウィザードを使用して、主マスター・レプリケーション管理者アカウントを作成し、すべてのマスター・サイトでのOracle8のレプリケーション管理者および伝播担当者、受信者の権限をユーザーに与え、次にすべてのサイトを接続する適切なリンクを設定する。3-3ページの「マルチマスター・レプリケーションの設定」を参照してください。
4. Replication Managerまたはレプリケーション管理APIを使用して、各レプリケーション・ベース・オブジェクト用のレプリケーション・サポートを再作成する。詳細は、3-23ページの「マスター・グループ・オブジェクトのレプリケーション・サポートの生成」を参照してください。他のアクティビティに加えて、レプリケーション・サポートを生成すると、登録済みの伝播担当者が、生成されるオブジェクトの所有者として確立されます。
5. Replication Managerまたはレプリケーション管理APIを使用して、環境を休止せずにレプリケーション・アクティビティを再開する。詳細は、3-16ページの「マスター・グループのレプリケーション・アクティビティの再開」を参照してください。
6. すべての関連スナップショット・サイトを、最低でも Oracle7 リリース 7.3.4 にアップグレードしなければなりません。スナップショット・サイトのOracle8への移行方法の詳細は、B-6ページの「スナップショット・サイトの段階的な移行」を参照してください。
7. すべてのスナップショット・サイトのすべてのスナップショットは、それらのマスター・サイトのOracle8への移行が終わったら、完全リフレッシュを行う必要があります。スナップショット・サイトの移行時に DBMS_JOB.BROKEN プロシージャをコールして「ブレーク」したジョブがある場合は、リフレッシュを実行する前に必ず「ブレーク解除」してください。

スナップショットがリフレッシュの "FORCE" オプションを使用して定義されている場合は、次にリフレッシュを試みると自動的に完全リフレッシュが実行されます。リフレッシュの "FAST" オプションを使用して定義されているスナップショットは、`dbms_refresh.refresh` または他のリフレッシュ用のプロシージャを使用して手動でリフレッシュする必要があります。

スナップショット・サイトでプロシージャ・レプリケーションを使用している場合は、さらに、プロシージャ・レプリケーションで使用するすべてのパッケージとその本体でスナップショット・サポートを再生成します。

注意：マスター・サイトを Oracle8 に移行する際にマスターのすべてのスナップショット・サイトを Oracle8 に移行する（つまり、スナップショット・サイトを段階的に移行する必要がない）場合は、マスターのスナップショット・ログを削除して、それらを主キー・スナップショット・ログとして再生成することができます。各スナップショット・サイトのスナップショットを変更して、それらを主キー・スナップショットに変換する必要があります。その後で、各主キー・スナップショットについて完全リフレッシュを実行できます。詳細は、B-11 ページの「主キー・スナップショットへのアップグレード」を参照してください。

8. Oracle7 マルチマスター・レプリケーション環境をメンテナンスするために使用していた、Oracle8 環境では不要な管理用のアカウントとリンクを削除してください。不要な権限を取り消してもかまいません。ただし、Oracle7 のスナップショット・サイトのメンテナンスに必要なアカウントを削除しないように注意してください。

段階的な移行

レプリケーション環境は、段階的に移行することができます。ただし、サイト間の相互依存性を慎重に分析して、移行の実行中もサイトを相互運用できることを確認する必要があります。表 B-1 で、Oracle7 と Oracle8 のレプリケーション・サイトを相互運用するための条件について説明します。

表 B-1 レプリケーション環境での相互運用性

環境	アクション	前提条件
マルチマスター	マスター・サイトを Oracle7 から Oracle8 へ移行する。	その他のすべてのマスター・サイトは、Oracle7 リリース 7.3.3 以降でなければなりません。
依存スナップショットを使用するマスター	マスター・サイトを Oracle7 から Oracle8 へ移行する。	すべての依存スナップショット・サイトは、Oracle7 リリース 7.3.4 以降でなければなりません。
依存スナップショットを使用するマスター	スナップショット・サイトを Oracle7 から Oracle8 へ移行する。	対応付けられたマスター・サイトは Oracle7 リリース 7.3.3 以降でなければなりません。

レプリケーション環境内で相互運用性の問題が発生するのを回避するために、段階的な移行は次の手順で行うことをお勧めします。

1. すべてのマスター・サイトを Oracle7 リリース 7.3.3 以降にアップグレードし、B-5 ページの「Oracle7 マスター・サイトの段階的移行の準備」で説明されている手順に従って、Oracle7 マスター・サイトを段階的に移行するための準備をする。
2. すべてのスナップショット・サイトを Oracle8 へ段階的に移行する。
3. すべてのマスター・サイトを Oracle8 へ段階的に移行する。

Oracle7 マスター・サイトの段階的移行の準備

Oracle7 のマスター・サイトまたはスナップショット・サイトの段階的移行を始める前に、Oracle7 リリース 7.3.3 以降のマスター・サイトを構成して、すべてのレプリケーション管理機能および伝播が、各サイトで単一のユーザーのセキュリティ・コンテキスト内で実行されるようにする必要があります。さらに、このプライマリ・マスター・レプリケーション管理者は、すべての Oracle7 および Oracle8 サイトで同一のユーザー名とパスワードを持っていなければなりません。Oracle7 マスター・サイトがすでにこの方法で構成されている場合があります。構成されていない場合は、次の手順を実行してください。

1. レプリケーション環境のプライマリ・マスター・レプリケーション管理者を選択する。現行のレプリケーション管理者を選択するか、新規ユーザーを作成します。

2. 各マスター・サイトで、DBMS_REPCAT_ADMIN.GRANT_ADMIN_ANY_REPGROUP と DBMS_REPCAT_AUTH.GRANT_SURROGATE_REPCAT の両方を使用して、必要な権限をプライマリ・マスター・レプリケーション管理者に付与する。
3. 各マスター・サイトからマルチマスター環境への次のリンクがない場合は、リンクを作成する。リンクの合計数は $3N(N - 1)$ です。
 - － SYS として作成され、有効なグローバル・データベース名と、有効な SQL*Net 2.3 TNS 別名を指定した USING 句とを含むパブリック・データベース・リンク。
 - － SYS として作成され、有効なグローバル・データベース名と、プライマリ・マスター・レプリケーション管理者のユーザー名およびパスワードを指定した CONNECT TO 句とを含むプライベート・データベース・リンク。
 - － プライマリ・レプリケーション管理者として作成され、有効なグローバル・データベース名と、プライマリ・マスター・レプリケーション管理者のユーザー名およびパスワードを指定した CONNECT TO 句とを含むプライベート・データベース・リンク。

スナップショット・サイトの段階的な移行

スナップショット・サイトを Oracle8 に移行する前に、対応付けられたマスター・サイトを Oracle7 リリース 7.3.3 以降にアップグレードして、段階的な移行のために完全に準備しておかなければなりません。

Oracle7 スナップショット・サイトを Oracle8 に段階的に移行するには、次のようにします。

1. スナップショット・サイトの更新可能スナップショットへのローカルの更新をすべて停止することによって、そのスナップショット・サイトをレプリケーション環境から分離する（分離されたセッションでは、各スナップショットの実表をロックして、それ以上のトランザクションを防ぐこともできます）。ローカル遅延トランザクション・キューをスナップショットのマスターに送って、そのキューを空にします。詳細は、『Oracle7 Server 分散システム Vol.2: レプリケート・データ』の第4章「DML 変更の非同期伝播」を参照してください。たとえば、スナップショット・サイトでの自動的な伝播およびリフレッシュを制御するジョブ・キューのエントリを一時的に保留または「ブレイク」することによって、スナップショット・サイトからそのマスターへのすべての伝播を停止します。
2. 『Oracle8 Server 移行ガイド』で説明されているように、Oracle7 から Oracle8 への移行ユーティリティと CATREP8M.SQL スクリプトを実行する。
3. Replication Manager のセットアップ・ウィザードを使用するか、または適切なレプリケーション管理 API のコールを実行して、プライマリ・スナップショット・レプリケーション管理者をスナップショット・サイトのレプリケーション管理者および伝播担当者として構成し、対応付けられたマスターの受信者アカウントを構成し、マスターへの適切なリンクを作成する。Oracle7 のマスター・サイトの場合、マスター・サイトの受信者はこれまでの項で指定したプライマリ・マスター・レプリケーション管理者でなければなりません。Replication Manager セットアップ・ウィザードを使用している場合は、カスタマイズ・オプションを選択してこの受信者を指定してください。

4. Replication Manager または適切なレプリケーション管理 API のコールを使用して、スナップショット・レプリケーション・サポートを再生成する。詳細は、4-18 ページの「更新可能スナップショットのレプリケーション・サポートの再生成」を参照してください。他のアクティビティに加えて、レプリケーション・サポートを生成すると、登録済みの伝播担当者が生成されるオブジェクトの所有者として確立されます。
5. Replication Manager または適切なレプリケーション管理 API のコールを使用して、伝播またはリフレッシュ（あるいはその両方）の間隔をマスターで再スケジュールし、適切な場合のローカルの更新を使用可能にする。ステップ 1 でマスター・サイトを分離するために DBMS_JOB.BROKEN プロシージャを使用した場合は、ジョブを「ブレイク解除」してスナップショット・サイトからレプリケーション・アクティビティを再開する必要があります。
6. Oracle7 マルチマスター・レプリケーション環境をメンテナンスするために使用していた、Oracle8 環境では不要な管理用のアカウントとリンクを削除する。不要な権限を取り消してもかまいません。

マスター・サイトの段階的移行

マスター・サイトを Oracle7 から Oracle8 へアップグレードするには、次の条件を満たしていなければなりません。

- マルチマスター環境内のその他のすべてのマスター・サイトが、Oracle7 リリース 7.3.3 以降である。
- B-5 ページの「Oracle7 マスター・サイトの段階的移行の準備」で説明した手順を完了している。
- すべての依存スナップショット・サイトが Oracle8 または Oracle7 リリース 7.3.4 以降である。

Oracle7 マスター・サイトを段階的に移行するには、次のようにします。

1. 移行するマスター・サイトを選択します。最初にマスター定義サイトを移行しなければなりません。
2. プロシージャ・レプリケーションを使用している場合は、既存のプロシージャ Wrapper の構成情報および位置（スキーマ）を記録します。この情報は後で使用されます。
3. レプリケーション環境からマスター・サイトを分離します。これには次のようにしなければなりません。
 - 次のいずれかの方法で、マスター・サイトの更新を停止します。

すべてのマスター・レプリケーション・グループのマスター定義サイトで、`DBMS_REPCAT.SUSPEND_MASTER_ACTIVITY` をコールする。

または各リモート・マスター・サイトおよび各依存スナップショット・サイトで `DBMS_DEFER_SYS.UNSCHEDULE_EXECUTION`（Oracle7 サイトの場合）または `DBMS_DEFER_SYS.UNSCHEDULE_PUSH`（Oracle8 サイトの場合）をコールして、

移行するマスター・サイトでの更新アクティビティを停止する。移行するマスター・サイトに影響を及ぼす可能性があるマスター定義サイトでは、管理操作を実行しないでください。

- ローカル・エラー・キューにエラーがある場合は、空になるまで、エラーを解決して再実行してください。
 - スナップショット・サイトでの自動的な伝播およびリフレッシュを制御するジョブ・キューのエントリを「ブレイク」することにより、依存スナップショット・サイトのリフレッシュを停止します。
 - ステップ 1 のタスクを完了するための詳細は、『Oracle7 Server 分散システム Vol.2: レプリケート・データ』の第 4 章「DML 変更の非同期伝播」および「レプリケーション・アクティビティの中断」、「遅延送信リストからのマスター・サイトの削除」、「遅延トランザクション・キューの送信リストへの強制」を参照してください。また、第 7 章の「エラーの手動による解決」も参照してください。
4. 『Oracle8 Server 移行ガイド』の説明に従い、Oracle7 から『Oracle8 Server 移行ガイド』ユーティリティとレプリケーションCATREP8M.SQLスクリプトを使用してマスター・サイトを移行します。
 5. Replication Managerのセットアップ・ウィザード、またはレプリケーション管理APIを使用して、プライマリ・マスター・レプリケーション管理者に、そのマスター・サイトのOracle8のプライマリ・レプリケーション管理者、伝播担当者、受信者の権限を付与します。詳細は、B-5ページの「Oracle7マスター・サイトの段階的移行の準備」3-3ページの「マルチマスター・レプリケーションの設定」を参照してください。B-5ページの「Oracle7マスター・サイトの段階的移行の準備」で説明した手順に従ってOracle7のマスター・サイトにOracle8との互換性を持たせた場合は、その他のすべてのOracle7およびOracle8のマスター・サイトにプライマリ・レプリケーション管理者からプライマリ・マスター・レプリケーション管理者へのリンクが存在しなければなりません。
 6. まだ休止状態になっていない場合は、Replication Managerまたはレプリケーション管理APIを使用してすべてのマスター・グループのレプリケーション・アクティビティを中断してください。詳細は、3-15ページの「マスター・グループのレプリケーション・アクティビティの中断」を参照してください。
 7. Replication Managerまたはレプリケーション管理APIを使用して、各レプリケート・オブジェクトのレプリケーション・サポートを再作成する。詳細は、3-23ページの「マスター・グループ・オブジェクトのレプリケーション・サポートの生成」を参照してください。レプリケーション環境にあるサイトがまだOracle7で実行されている場合は、min_communicationパラメータをFALSEに設定する必要があります。いったんすべてのサイトをOracle8に移行したら、min_communicationパラメータはTRUE（デフォルト）にしか設定できません。詳細は、5-34ページの「更新の競合解消のためのデータ伝播の最小化」を参照してください。他のアクティビティに加えて、レプリケーション・サポートを生成すると、登録済みの伝播担当者が、生成されるオブジェクトの所有者として確立されます。
 8. プロシージャ・レプリケーションを使用している場合は、残っているOracle7マスター・サイトをチェックし、Wrapperが移動しているかどうかを確認する（ステップ2で

ストが作成されている)。Wrapper が移動されている場合、前の位置（そのサイトで以前にシステムベース・モデルが使われていたか、ユーザーベース・モデルが使われていたかにより、レプリケーション管理者のスキーマまたは表所有者のスキーマのどちらか）に別名を作成して、プライマリ・レプリケーション管理者のスキーマ内の新しい位置を指すようにします。新しい所有者および位置にアクセスするのに必要なオブジェクト権限が付与されていることを確認します。スナップショット・サイトでプロシージャ・レプリケーションを使用している場合は、プロシージャ・レプリケーションで使用するすべてのパッケージとその本体でスナップショット・サポートを再生成します。

9. Replication Manager またはレプリケーション管理 API を使用して、レプリケーション・アクティビティを再開し各マスター・グループ用の環境を非休止する。詳細は、3-16 ページの「マスター・グループのレプリケーション・アクティビティの再開」を参照してください。他のマスターへの伝播、または他のマスターからの伝播をスケジュール解除することによってマスターを分離していた場合は、すべてのマスター・サイトについて DBMS_DEFER_SYS.SCHEDULE_EXECUTION（Oracle7 サイトの場合）を実行するか、または 3-10 ページの「スケジュール・リンクの編集」（Oracle8 サイトの場合）の指示に従って、伝播を再スケジュールしてください。
10. Oracle7 および Oracle8 両方のスナップショット・サイトのすべてのスナップショットは、それらのマスター・サイトの Oracle8 への移行が終わったら、完全リフレッシュを行う必要があります。Oracle8 の新しい ROWID フォーマットのため、Oracle7 から Oracle8 への移行ユーティリティによってすべてのマスター・スナップショット・ログが切り捨てられます。ステップ 3 でマスター・サイトを分離するために DBMS_JOB.BROKEN プロシージャを使った場合は、ジョブを「ブレイク解除」してスナップショット・サイトからレプリケーション・アクティビティを再開します。

スナップショットがリフレッシュの "FORCE" オプションを使用して定義されている場合は、次にリフレッシュを試みると自動的に完全リフレッシュが実行されます。リフレッシュの "FAST" オプションを使用して定義されているスナップショットは、dbms_refresh.refresh または他のリフレッシュ用のプロシージャを使用して手動でリフレッシュする必要があります。

注意：マスター・サイトを Oracle8 に移行する際にマスターのすべてのスナップショット・サイトを Oracle8 に移行できる（つまり、スナップショット・サイトを段階的に移行する必要がない）場合は、マスターのスナップショット・ログを削除して、それらを主キー・スナップショット・ログとして再生成することができます。各スナップショット・サイトのスナップショットを変更して、それらを主キー・スナップショットに変換する必要があります。その後で、各主キー・スナップショットについて完全リフレッシュを実行できます。詳細は、B-11 ページの「主キー・スナップショットへのアップグレード」を参照してください。

11. Oracle7 マルチマスター・レプリケーション環境をメンテナンスするために使用していた、Oracle8 環境では不要な管理用のアカウントとリンクを削除してください。不要な権限を取り消してもかまいません。ただし、Oracle7 のスナップショット・サイトまたはマスター・サイトのメンテナンスに必要なアカウントを削除しないように注意してください。

エクスポート / インポートを使用する移行

Oracle7 リリース 7.3.3 以降からの全データベースのエクスポートと Oracle8 へのインポートは、マスターとスナップショットの両方でサポートされています。上記の手順での Oracle7 から Oracle8 への移行ユーティリティと CATREP8M.SQL スクリプトのかわりに、エクスポート / インポートを使用できます。ただし、上記の手順を実行する場合は、実際に Oracle7 から Oracle8 に移行する前と後に、すべてのステップに正しく従っていることを確認してください。

Oracle7 リリース 7.3.3 以降から全データベースをエクスポートするには、次のステップに従います。

1. FULL=y を指定した SYSTEM スキーマ下でリリース 7.3 の Export ユーティリティを使用し、Oracle7リリース7.3.3以降のデータベースをダンプ・ファイルにエクスポートする。
2. インストール済のアドバンスド・レプリケーションで、Oracle8 データベースを起動する。SYSTEM として接続し、DBMS_REPCAT_MIG.PRE_IMPORT を実行します。この手順により、レプリケーション・データ・ディクショナリ表の参照制約が一時的に使用できなくなります。
3. FULL=y を指定した SYSTEM スキーマ下で Oracle8 の Import ユーティリティを使用し、ダンプ・ファイルを Oracle8 データベースにインポートする。レプリケーション・データ・ディクショナリ表の参照制約は、自動的に使用可能になります。

個々の Oracle7 の表からデータをエクスポートして、Oracle8 の表へインポートし、標準のアドバンスド・レプリケーション・プロシージャを使用して Oracle8 のレプリケーション環境でそれらの表をマスターとして構成することもできます。

詳細は、『Oracle8 Server ユーティリティ』を参照してください。

主キー・スナップショットへのアップグレード

スナップショット・サイトおよびそのマスターを Oracle8 へ移行したら、ROWID スナップショットを Oracle8 の主キー・スナップショットにアップグレードできます。これを行うには、最初に各マスター表のスナップショット・ログを変更して、マスターの行が更新されるときに、主キー情報と ROWID 情報を両方ともロギングする必要があります。マスター・サイトでこれが完了すると、スナップショットを変更してそれらを主キー・スナップショットに変換することによって、Oracle8 のスナップショット・サイトを段階的に変換できます。ROWID 情報だけでなく主キー情報もロギングするように変更されている Oracle8 マスターでは、Oracle7 の ROWID スナップショットも Oracle8 の ROWID および主キー・スナップショットも同時にサポートできるので、段階的な移行が可能です。

注意：主キー・スナップショットは ROWID スナップショットに変換またはダウングレードできません。

マスター・サイトでの主キー・スナップショットの変換

主キー・スナップショットをサポートするには、Oracle8 マスター・サイトで次の手順を実行します。

1. 主キー制約が使用可能になっていない各マスター表において、主キー制約を定義して使用可能にします。
2. ALTER SNAPSHOT LOG コマンドを使用して、高速リフレッシュをサポートしている各マスター表のスナップショット・ログを変更して、主キー情報を含むようにします。詳細は、『Oracle8 Server SQL リファレンス』の ALTER SNAPSHOT LOG の項を参照してください。

注意：上記の条件が満たされない場合、スナップショット・サイトで主キー・スナップショットに変換するために ALTER SNAPSHOT コマンドを実行すると、エラーが発生します。

スナップショット・サイトでの主キー・スナップショットの変換

Oracle8 マスター・サイトを構成して主キー・スナップショットをサポートするようにしたら、Oracle8 スナップショット・サイトで次の手順を実行します。

1. スナップショット・サイトの更新可能スナップショットに対するローカルの更新のすべてを停止することによって、そのスナップショット・サイトをレプリケーション環境から分離します。
2. 主キー・スナップショットに変換中の読み専用 ROWID スナップショットにその主キーのすべての列が含まれていない場合、スナップショットを削除し、すべての主キー列を使用して再作成します。詳細は、2-10 ページの「単純スナップショットの作成」を参照してください。

注意：ROWID スナップショットには制約を定義しないでください。

3. すべてのスナップショットの高速リフレッシュを実行して、マスター・スナップショット・ログに残っている ROWID の参照性を取り除きます。

4. ALTER SNAPSHOT コマンドを使用して、ROWID スナップショットを主キー・スナップショットに変換します。構文の全情報は、『Oracle8 Server SQL リファレンス』を参照してください。
5. 伝播またはスナップショット・リフレッシュ（あるいはその両方）をマスターで再スケジュールすることによって、レプリケーションを再開し、適切な場合のローカルの更新を使用可能にします。ステップ 1 でマスター・サイトを分離するために DBMS_JOB.BROKEN プロシージャを使用した場合は、ジョブを「ブレイク解除」してスナップショット・サイトからレプリケーション・アクティビティを再開する必要があります。

Oracle8 への移行を必要とする機能

次に示す機能を使用する場合、関連するすべてのサイトが Oracle8 に正常に移行されている必要があります。

1. LOB データ型のレプリケーション
2. データ伝播の縮小
 - － min_communication パラメータの使用
 - － send_and_compare_old_values プロシージャの使用
3. 遅延トランザクションの平行伝播
4. グローバルな認証および権限付きデータベース・リンク
5. 妥当性検査プロシージャ

追加情報：付録 A 「新機能」を参照してください。

次に示す機能では、関連するすべてのサイトが Oracle8 および主キー・スナップショットに正常に移行されている必要があります。

1. 副問合せを使用する単純スナップショット
2. マスター表再編成プロシージャ

次の機能は、Oracle7 と Oracle8 が混在する環境で自動的に動作しますが、Oracle8 サイトだけに影響します。

1. きめ細かな休止
2. スナップショットの登録

注意：Oracle7 サイトのマスター・グループのいずれかが休止すると、そのサイトのすべてのマスター・グループが休止します。

注意：Oracle7 スナップショットは Oracle8 サイトに自動的に登録されませんが、マスター・サイトで DBMS_SNAPSHOT.REGISTER_SNAPSHOT および DBMS_SNAPSHOT.UNREGISTER_SNAPSHOT プロシージャを使用して手動で登録できま

す。詳細は、2-34 ページの「マスター・サイトでのスナップショットの登録」を参照してください。

旧プロシージャ

Oracle8 で廃止されたプロシージャは次のとおりです。

DBMS_REPCAT_ADMIN.GRANT_ADMIN_REPGROUP

DBMS_REPCAT_ADMIN.GRANT_ADMIN_ANY_REPGROUP

DBMS_REPCAT_ADMIN.REVOKE_ADMIN_REPGROUP

DBMS_REPCAT_ADMIN.REVOKE_ADMIN_ANY_REPGROUP

DBMS_REPCAT_AUTH.GRANT_SURROGATE_REPCAT

DBMS_REPCAT_AUTH.REVOKE_SURROGATE_REPCAT

DBMS_DEFER_SYS.EXECUTE

旧プロシージャ

索引

A

ADD_MASTER_DATABASE プロシージャ
 DBMS_REPCAT パッケージ 3-22
ADD プロシージャ
 DBMS_REFRESH パッケージ 2-37
ALTER SNAPSHOT LOG コマンド 2-28
ALTER SNAPSHOT コマンド
 REFRESH 句 2-43
ALTER_MASTER_REPOBJECT プロシージャ
 DBMS_REPCAT パッケージ 3-20
ALTER_SNAPSHOT_PROPAGATION プロシージャ
 DBMS_REPCAT パッケージ 4-15
AND 式
 単純副問合せスナップショット 2-24

B

BEGIN_INSTANTIATION プロシージャ
 DBMS_OFFLINE_OG パッケージ 7-15
BEGIN_LOAD プロシージャ
 DBMS_OFFLINE_OG パッケージ 7-16
BEGIN_TABLE_REORGANIZATION プロシージャ
 DBMS スナップショット・パッケージ 2-30
BLOB のサポート A-3

C

CHANGE プロシージャ
 DBMS_REFRESH パッケージ 2-37
CLOB のサポート A-3
CREATE SNAPSHOT LOG コマンド 2-17
CREATE SNAPSHOT コマンド 2-7, 2-10, 2-19–2-21
 REFRESH 句 2-43

CREATE TABLE コマンド 2-16
CREATE_MASTER_REPGROUP プロシージャ
 DBMS_REPCAT パッケージ 3-14

D

DBA_REFRESH_CHILDREN ビュー 2-43
DBA_REFRESH ビュー 2-43
DBA_REGISTERED_SNAPSHOTS ビュー 2-34, 2-43
DBA_RGROUP ビュー 2-43
DBA_SNAPSHOT_LOGS ビュー 2-35, 2-43
DBA_SNAPSHOTS ビュー 2-43
DBMS_DEFER_QUERY パッケージ
 GET_ARG_TYPE プロシージャ 9-9
 GET_CALL_ARGS プロシージャ 9-10
 GET_CHAR_ARG プロシージャ 9-11
 GET_DATE_ARG プロシージャ 9-11
 GET_NUMBER_ARG プロシージャ 9-11
 GET_RAW_ARG プロシージャ 9-11
 GET_ROWID_ARG プロシージャ 9-11
 GET_VARCHAR2_ARG プロシージャ 9-11
DBMS_DEFER_SYS パッケージ
 ADD_DEFAULT_DEST プロシージャ 9-14
 DELETE_DEF_DESTINATION プロシージャ 9-15
 DELETE_DEFAULT_DEST プロシージャ 8-6, 9-14
 DELETE_ERROR プロシージャ 9-15
 DELETE_TRAN プロシージャ 9-16
 DISABLED ファンクション 9-16
 DISABLED プロシージャ 3-11
 EXCLUDE_PUSH ファンクション 9-17
 EXECUTE_ERROR プロシージャ 9-18
 PURGE プロシージャ 9-20

PUSH プロシージャ 9-22
 REGISTER PROPAGATOR プロシージャ 9-24
 SCHEDULE_EXECUTION プロシージャ 3-9, 3-10,
 3-12, 3-13, 9-26
 SCHEDULE_PURGE プロシージャ 9-25
 SCHEDULE_PUSH プロシージャ 3-9, 3-10, 3-12,
 3-13, 9-26
 SET_DISABLED プロシージャ 3-10, 3-12, 3-13,
 9-28
 UNSCHEDULE_PURGE プロシージャ 9-30
 UNSCHEDULE_PUSH プロシージャ 3-11, 9-30
 DBMS_DEFER パッケージ
 ADD_DEFAULT_DEST プロシージャ 8-6
 CALL プロシージャ 9-4
 CHAR_ARG プロシージャ 8-5, 9-6
 COMMIT_WORK プロシージャ 9-5
 datatype_ARG プロシージャ 9-6
 DATE_ARG プロシージャ 8-5, 9-6
 NUMBER_ARG プロシージャ 8-5, 9-6
 RAW_ARG プロシージャ 8-5, 9-6
 ROWID_ARG プロシージャ 8-5, 9-6
 TRANSACTION プロシージャ 8-4, 9-7
 VARCHAR2_ARG プロシージャ 8-5, 9-6
 DBMS_OFFLINE_OG パッケージ
 BEGIN_INSTANTIATION プロシージャ 7-15, 9-31
 BEGIN_LOAD プロシージャ 7-16, 9-32
 END_INSTANTIATION プロシージャ 7-16, 9-33
 END_LOAD プロシージャ 7-16, 9-34
 RESUME_SUBSET_OF_MASTERS プロシージャ
 7-15, 9-35
 DBMS_OFFLINE_SNAPSHOT パッケージ
 BEGIN_LOAD プロシージャ 9-36
 END_LOAD プロシージャ 9-38
 DBMS_RECTIFIER_DIFF パッケージ 3-23, 6-16
 DIFFERENCES プロシージャ 9-39
 RECTIFY プロシージャ 9-42
 DBMS_REFRESH パッケージ
 ADD プロシージャ 2-37, 9-44
 CHANGE プロシージャ 2-37, 9-45
 DESTROY プロシージャ 2-38, 9-47
 MAKE プロシージャ 4-21, 9-48
 REFRESH プロシージャ 2-38, 9-50
 SUBTRACT プロシージャ 2-37, 9-51
 DBMS_REPCAT_ADMIN パッケージ
 GRANT_ADMIN_ANY_SCHEMA プロシージャ
 9-130
 GRANT_ADMIN_SCHEMA プロシージャ 9-130
 REVOKE_ADMIN_ANY_SCHEMA プロシージャ
 9-131
 REVOKE_ADMIN_SCHEMA プロシージャ 9-132
 DBMS_REPCAT_AUTH パッケージ
 GRANT_SURROGATE_REPCAT プロシージャ
 9-132
 REVOKE_SURROGATE_REPCAT プロシージャ
 9-133
 DBMS_REPCAT パッケージ 6-6
 ADD_DELETE_RESOLUTION プロシージャ 9-60
 ADD_GROUPED_COLUMN プロシージャ 9-54
 ADD_MASTER_DATABASE プロシージャ 3-22,
 9-55
 ADD_PRIORITY_CHAR プロシージャ 5-20, 9-57
 ADD_PRIORITY_DATE プロシージャ 5-20, 9-57
 ADD_PRIORITY_NUMBER プロシージャ 5-20,
 9-57
 ADD_PRIORITY_RAW プロシージャ 5-20, 9-57
 ADD_PRIORITY_VARCHAR2 プロシージャ 5-20,
 9-57
 ADD_SITE_PRIORITY_SITE プロシージャ 5-24,
 9-58
 ADD_UNIQUE_RESOLUTION プロシージャ 9-60
 ADD_UPDATE_RESOLUTION プロシージャ 9-60
 ALTER_MASTER_PROPAGATION プロシージャ
 9-63
 ALTER_MASTER_REPOBJECT プロシージャ
 3-20, 9-64
 ALTER_PRIORITY_CHAR プロシージャ 5-20, 9-67
 ALTER_PRIORITY_DATE プロシージャ 5-20, 9-67
 ALTER_PRIORITY_NUMBER プロシージャ 5-20,
 9-67
 ALTER_PRIORITY_RAW プロシージャ 5-20, 9-67
 ALTER_PRIORITY_VARCHAR2 プロシージャ
 5-20, 9-67
 ALTER_PRIORITY プロシージャ 5-21, 9-66
 ALTER_SITE_PRIORITY_SITE プロシージャ 5-25,
 9-70
 ALTER_SITE_PRIORITY プロシージャ 5-24, 9-68
 ALTER_SNAPSHOT_PROPAGATION プロシージャ
 9-71
 CANCEL_STATISTICS プロシージャ 6-15, 9-72
 COMMENT_ON_COLUMN_GROUP プロシージャ
 6-25, 9-72
 COMMENT_ON_DELETE_RESOLUTION プロシージャ

ジャ 9-77
 COMMENT_ON_DELETE 解消パッケージ 6-25
 COMMENT_ON_PRIORITY_GROUP プロシージャ 6-25, 9-73
 COMMENT_ON_REPCAT プロシージャ 6-25
 COMMENT_ON_REPGROUP プロシージャ 9-74
 COMMENT_ON_REPOBJECT プロシージャ 6-25, 9-76
 COMMENT_ON_REPSchema プロシージャ 6-25
 COMMENT_ON_REPSITES プロシージャ 9-75
 COMMENT_ON_SITE_PRIORITY プロシージャ 9-73
 COMMENT_ON_UNIQUE_RESOLUTION プロシージャ 9-77
 COMMENT_ON_UNIQUE 解消パッケージ 6-25
 COMMENT_ON_UPDATE_RESOLUTION プロシージャ 9-77
 COMMENT_ON_UPDATE 解消パッケージ 6-25
 CREATE_MASTER_REPGROUP プロシージャ 3-14, 9-79
 CREATE_MASTER_REPOBJECT プロシージャ 9-80
 CREATE_SNAPSHOT_REPGROUP プロシージャ 9-83
 CREATE_SNAPSHOT_REPOBJECT プロシージャ 9-84
 DEFINE_COLUMN_GROUP プロシージャ 9-87
 DEFINE_PRIORITY_GROUP プロシージャ 5-19, 9-88
 DEFINE_SITE_PRIORITY プロシージャ 5-24, 9-89
 DO_DEFERRED_REPCAT_ADMIN プロシージャ 6-5, 6-5-6-6, 9-90
 DROP_COLUMN_GROUP プロシージャ 9-91
 DROP_GROUPED_COLUMN プロシージャ 9-92
 DROP_MASTER_REPGROUP プロシージャ 3-15, 9-93
 DROP_MASTER_REPOBJECT プロシージャ 3-21, 9-94
 DROP_PRIORITY_CHAR プロシージャ 5-22, 9-97
 DROP_PRIORITY_DATE プロシージャ 5-22, 9-97
 DROP_PRIORITY_GROUP プロシージャ 5-23, 9-96
 DROP_PRIORITY_NUMBER プロシージャ 5-22, 9-97
 DROP_PRIORITY_RAW プロシージャ 5-22, 9-97
 DROP_PRIORITY_VARCHAR2 プロシージャ 5-22, 9-97
 DROP_PRIORITY プロシージャ 5-22, 9-95
 DROP_SITE_PRIORITY_SITE プロシージャ 5-25, 9-99
 DROP_SITE_PRIORITY プロシージャ 5-25, 9-98
 DROP_SNAPSHOT_REPGROUP プロシージャ 9-100
 DROP_SNAPSHOT_REPOBJECT プロシージャ 4-20, 9-101
 EXECUTE_DDL プロシージャ 6-2, 9-104
 GENERATE_REPLICATION_PACKAGE プロシージャ 9-105
 GENERATE_REPLICATION_SUPPORT プロシージャ 3-25, 9-106
 GENERATE_REPLICATION_TRIGGER プロシージャ 9-108
 GENERATE_SNAPSHOT_SUPPORT 9-110
 MAKE_COLUMN_GROUP プロシージャ 9-112
 PURGE_MASTER_LOG プロシージャ 9-113
 PURGE_STATISTICS プロシージャ 6-15, 9-113
 REFRESH_SNAPSHOT_REPGROUP プロシージャ 9-114
 REGISTER_STATISTICS プロシージャ 6-15, 9-117
 RELOCATE_MASTERDEF プロシージャ 6-3, 9-117
 REMOVE_MASTER_DATABASES プロシージャ 3-22, 9-119
 REPCAT_IMPORT_CHECK プロシージャ 6-14, 9-120
 RESUME_MASTER_ACTIVITY プロシージャ 3-17, 9-121
 SET_COLUMNS プロシージャ 3-18, 9-123
 SUSPEND_MASTER_ACTIVITY プロシージャ 3-16, 9-124
 SWITCH_SNAPSHOT_MASTER プロシージャ 4-15, 6-3, 9-125
 VALIDATE プロシージャ 9-126
 WAIT_MASTER_LOG プロシージャ 9-129
 DBMS_REPUTIL パッケージ
 FROM_REMOTE 変数 9-143
 GLOBAL_NAME 変数 9-143
 REPLICATION_IS_ON 変数 9-143
 REPLICATION_OFF プロシージャ 7-31, 9-134

REPLICATION_ON プロシージャ 7-32, 9-134
DBMS_SNAPSHOT パッケージ
 BEGIN_TABLE_REORGANIZATION プロシージャ 2-30, 9-135
 END_TABLE_REORGANIZATION プロシージャ 2-30, 9-135
 I_AM_A_REFRESH ファンクション 9-136
 PURGE_LOG プロシージャ 2-29, 9-136
 REFRESH プロシージャ 9-137
 REGISTER_SNAPSHOT プロシージャ 2-35, 9-139
 SET_I_AM_A_REFRESH プロシージャ 9-141
 UNREGISTER_SNAPSHOT 9-141
 UNREGISTER_SNAPSHOT プロシージャ 2-35
「DB 接続の編集」プロパティ・シート
 Replication Manager 3-12, 3-13
 セットアップ・ウィザード 3-12
DDL
 マスター・グループ内での実行 6-2
 マスター・サイトに伝播されない変更 6-20
 マスター・サイトへの文の伝播 6-2
「DDL の実行」ダイアログ
 Replication Manager 6-2
DEFCALL ビュー 10-18
DEFDEFAULTDEST ビュー 10-18
DefDefaultDest 表
 接続先の追加 8-6, 9-14
 接続先の取消し 8-6, 9-14, 9-15
DEFERRCOUNT ビュー 10-19
DEFERROR ビュー 10-19, 10-20
DefError 表
 トランザクションの削除 9-15
DEFLOB 10-20
DEFSCHEDULE 10-21
DEFTRAN 10-22
DEFTRANDEST ビュー 10-22
DESTROY プロシージャ
 DBMS_REFRESH パッケージ 2-38
DIFFERENCES
 プロシージャ 6-16
DISABLED プロシージャ
 DBMS_DEFER_SYS パッケージ 3-11
DML
 他のサイトに伝播されない変更 6-20
DO_DEFERRED_REPCAT_ADMIN パッケージ 6-6
DO_DEFERRED_REPCAT_ADMIN プロシージャ

 DBMS_REPCAT パッケージ 6-5, 6-5-6-6
DROP SNAPSHOT LOG コマンド 2-33
DROP SNAPSHOT コマンド 2-35
DROP_MASTER_REPGROUP プロシージャ
 DBMS_REPCAT パッケージ 3-15
DROP_MASTER_REPOBJECT プロシージャ
 DBMS_REPCAT パッケージ 3-21
DROP_SNAPSHOT_REPOBJECT プロシージャ
 DBMS_REPCAT パッケージ 4-20

E

END_INSTANTIATION プロシージャ
 DBMS_OFFLINE_OG パッケージ 7-16
END_LOAD プロシージャ
 DBMS_OFFLINE_OG パッケージ 7-16
END_TABLE_REORGANIZATION プロシージャ
 DBMS_SNAPSHOT パッケージ 2-30
EXECUTE_DDL プロシージャ
 DBMS_REPCAT パッケージ 6-2
EXIST 句
 単純副問合せスナップショット 2-23
Export ユーティリティ
 マスター・サイトのオフライン・インスタンス
 ーション 7-15

G

gen_rep_pack 5-18
GENERATE_REPLICATION_SUPPORT プロシージャ
 DBMS_REPCAT パッケージ 3-25

I

import_check 6-14
INIT.ORA 3-15

J

JOB_QUEUE_INTERVAL 初期化パラメータ 2-6, 4-9, 6-6
JOB_QUEUE_PROCESSES 初期化パラメータ 2-6, 4-9, 6-6

L

LOB
 サポートする A-3
 古い値の省略 5-34

M

MAKE プロシージャ
 DBMS_REFRESH パッケージ 2-26, 4-21
 NEXT_DATE パラメータ 2-26
min_communication
 例 5-34

N

NCLOB のサポート A-3
NEXT_DATE パラメータ
 DBMS_REFRESH パッケージ
 MAKE プロシージャ 2-26

O

OPEN_LINKS 初期化パラメータ
 初期化パラメータ
 OPEN_LINK 3-15
Oracle Replication Manager 1-15
Oracle7
 min_communication 4-18
Oracle8
 機能および移行 B-12

P

PARALLEL_MAX_SERVERS 初期化パラメータ 3-28
PARALLEL_MIN_SERVERS 初期化パラメータ 3-28
PARALLEL_SERVER_IDLE_TIME 初期化パラメータ 3-28
PCTFREE
 スナップショット・ログの値 2-9
 複合スナップショットの値 2-42
PCTUSED
 スナップショット・ログの値 2-9
 複合スナップショットの値 2-42
pgroup 5-18
PL/SQL トリガー
 スナップショット実表での使用 4-19
PRESERVE SNAPSHOT LOG オプション
 TRUNCATE TABLE コマンド 2-31
PRIMARY KEY 制約
 スナップショット要件 2-12
 単純副問合せスナップショットと 2-23
priority_groups 5-17
PURGE SNAPSHOT LOG オプション

TRUNCATE TABLE コマンド 2-31

PURGE_LOG プロシージャ
 DBMS_SNAPSHOT パッケージ 2-29

R

RECTIFY
 プロシージャ 6-16
REFRESH_CHILDREN ビュー 10-28
REFRESH 句
 ALTER SNAPSHOT コマンド 2-43
 CREATE SNAPSHOT コマンド 2-43
REFRESH ビュー 10-26
REFRESH プロシージャ
 DBMS_REFRESH パッケージ 2-38
REGISTER_SNAPSHOT プロシージャ
 DBMS_SNAPSHOT パッケージ 2-35
REGISTERED_SNAPSHOTS 10-24
RELOCATE_MASTERDEF プロシージャ
 DBMS_REPCAT パッケージ 6-3
REMOVE_MASTER_DATABASES プロシージャ
 DBMS_REPCAT パッケージ 3-22
REPCATLOG ビュー 3-27, 6-6–6-7, 10-5
RepCatLog ビュー
 削除 9-113
RepCat 表
 更新 6-25
REPCOLUMN_GROUP ビュー 10-6
RepColumn_Group 表
 更新 6-25, 9-72
REPCOLUMN ビュー 10-6
REPCONFLICT ビュー 10-7
REPDDL ビュー 10-7
REPGENOBJECTS ビュー 10-16
REPERATED ビュー 10-8
REPGROUPED_COLUMN ビュー 10-8
REPGROUP ビュー 3-27, 10-3
RepGroup ビュー
 更新 9-74
REPKEY_COLUMNS ビュー 10-9
Replication Manager 1-15
 「DDL の実行」ダイアログ 6-2
 「間隔の設定」ダイアログ 3-8, 3-11
管理要求 6-4–6-7
 「グループへのオブジェクトの追加」ダイアログ
 3-17

「終了」ページ 3-6
「新規スケジュール・リンクの作成」プロパティ・シート 3-8
「新規マスター・グループの作成」プロパティ・シート 3-13
「スケジュール」フォルダ 3-10
「スケジュール・リンク」フォルダ 3-10
「スケジュール・リンクの設定」ダイアログ 3-8
「スナップショット・ログの作成」プロパティ・シート 4-10
「スナップショット・ログの編集」プロパティ・シート 4-10
セットアップ・ウィザード 3-3
「代替キー列の設定」ダイアログ 3-14, 3-18
「データベース接続先の編集」プロパティ・シート 3-26
「日付の設定」ダイアログ 3-8, 3-11
「マスター・グループの編集」プロパティ・シート 3-16, 3-17
「リフレッシュ・グループの作成」プロパティ・シート 4-3
「レプリケーション・オブジェクトの変更」ダイアログ 3-20
「レプリケーション・オブジェクトの編集」プロパティ・シート 3-20
REPOBJECT ビュー 3-27, 10-9
RepObject 表
更新 6-25, 9-76
REPPARAMETER_COLUMN ビュー 10-10
REPPRIORITY_GROUP ビュー 10-12
RepPriority_Group 表
更新 6-25, 9-73
REPPRIORITY ビュー 10-11
REPPROP ビュー 10-12
REPRESOL_STATS_CONTROL ビュー 10-14
REPRESOLUTION_METHOD ビュー 10-14
REPRESOLUTION_STATISTICS ビュー 10-15
RepResolution_Statistics ビュー
統計収集 6-15
RepResolution_Statistics 表
削除 6-15, 9-113
REPRESOLUTION ビュー 10-13
RepResolution 表
更新 6-25, 9-77
RepSchema 表

更新 6-25
REPSITES ビュー 3-27, 10-16
RepSite ビュー
更新 9-75
RESUME_MASTER_ACTIVITY プロシージャ
DBMS_REPCAT パッケージ 3-17
RESUME_SUBSET_OF_MASTERS プロシージャ
DBMS_OFFLINE_OG パッケージ 7-15
ROWID スナップショット 1-11, 2-42
RPC 1-19

S

SCHEDULE_EXECUTION プロシージャ
DBMS_DEFER_SYS パッケージ 3-9, 3-10, 3-12, 3-13
SCHEDULE_PUSH プロシージャ
DBMS_DEFER_SYS パッケージ 3-9, 3-10, 3-12, 3-13
Schema Manager
「スナップショット・ログ」プロパティ・シート 2-34
スナップショット・ログの作成 2-7
「スナップショット・ログの作成」プロパティ・シート 2-8
スナップショットの作成 2-7, 2-10
「スナップショットの作成」プロパティ・シート 2-13, 2-14
「リフレッシュ・グループの作成」プロパティ・シート 2-26
SET_COLUMNS プロシージャ
DBMS_REPCAT パッケージ 3-18
set_disabled 9-28
SET_DISABLED プロシージャ
DBMS_DEFER_SYS パッケージ 3-10, 3-12, 3-13
SNAPSHOTS_LOGS ビュー 10-25
SNAPSHOTS ビュー 10-23
SNP バックグラウンド・プロセス 3-7, 4-9
基本レプリケーション
環境 2-6
自動スナップショット・リフレッシュ 1-10
スナップショット・サイト 4-9
SQL
コマンド
スナップショットの作成 2-16–2-21
単純スナップショットに関する制限事項 2-13

SUBTRACT プロシージャ
 DBMS_REFRESH パッケージ 2-37
SUSPEND_MASTER_ACTIVITY プロシージャ
 DBMS_REPCAT パッケージ 3-16
SWITCH_SNAPSHOT_MASTER プロシージャ
 DBMS_REPCAT パッケージ
 ALTER_SNAPSHOT_PROPAGATION プロ
 シージャ 4-15

T

TRUNCATE TABLE コマンド
 PRESERVE SNAPSHOT LOG オプション 2-31
 PURGE SNAPSHOT LOG オプション 2-31
TRUNCATE コマンド 2-29

U

UNIQUE 制約
 単純副問合せスナップショットと 2-23
UNREGISTER_SNAPSHOT プロシージャ
 DBMS_SNAPSHOT パッケージ 2-35
UNSCHEDULE_PUSH プロシージャ
 DBMS_DEFER_SYS パッケージ 3-11
USER_REFRESH_CHILDREN ビュー 2-39
USER_REFRESH ビュー 2-39

V

VALIDATE プロシージャ A-4

W

WHERE 句
 更新可能スナップショットの作成
 更新可能スナップショット
 WHERE 句を使用して作成する 4-17

Wrapper
 プロシージャ・レプリケーション 1-25

あ

アカウント
 スナップショット用の作成 2-4
アドバンスト・レプリケーション 3-1-7-32
 RPC 1-19
 概要(「基本レプリケーション」も参照) 1-11-
 1-27
 行レベル・レプリケーション 1-17

更新可能スナップショット 1-13
使用禁止にする 7-31
使用方法 1-11
ジョブ・キュー 1-20
スナップショット・サイト
 レプリケーション 4-1-4-22
 オフライン・インスタンスエーション 7-16
 セキュリティ設定 7-23
遅延トランザクション 1-20
テクニック 7-1-7-32
同期伝播 1-26, 7-6
トランザクション伝播保護 3-29-3-30
ハイブリッド構成 1-14
非同期伝播 1-18
プロシージャ・レプリケーション 1-25
マルチマスター 3-1-3-30
マルチマスター構成 1-12
アドバンスト・レプリケーション・システム
 監視 6-4
アプリケーション負荷の分配
 アドバンスト・レプリケーション 1-12

い

移行
 インポートおよびエクスポートを使用する B-10
 概要 B-2
 旧プロシージャ B-13
 主キー・スナップショットへのアップグレード
 B-11
 スナップショット・サイト B-6
 段階的な B-5
 必要とする機能 B-12
 マスター・サイト B-3
移行を必要とする機能 B-12
意思決定支援アプリケーション
 基本レプリケーション 1-5
依存性
 順序
 レプリケート・トランザクション 3-29
 追跡
 パラレル伝播 3-30
一意性の競合 1-22, 5-3
 解消 5-7, 5-40
 構成 5-28
 解消方法 5-8

- 削除 5-28
- 割当て 5-28
- 防止 5-4
- 「一般」ページ
 - 「新規マスター・グループの作成」プロパティ・シート 3-13
 - 「スナップショット・ログ」プロパティ・シート 2-34
 - 「スナップショット・ログの作成」プロパティ・シート 4-10
 - 「スナップショット・ログの編集」プロパティ・シート 4-10
 - 「レプリケーション・オブジェクトの編集」プロパティ・シート 3-24, 4-19
- インスタンスエーション
 - オフライン 7-13
 - アドバンスド・レプリケーション 7-16
 - アドバンスド・レプリケーションのマスター・サイト 7-15
- インスタンスエーション、オフラインの A-4
- インポート・ユーティリティ
 - マスター・サイトのオフライン・インスタンスエーション 7-16
- インポートされたデータのチェック 6-14
- インポートする
 - オブジェクト・グループ
 - オフライン・インスタンスエーション 9-32, 9-34
 - 状態チェック 9-120
 - スナップショット
 - オフライン・インスタンスエーション 9-36, 9-38

う

- 上書き
 - 競合解消方法 5-16
 - 使用 5-32

え

- エラー
 - VALIDATE による回避 A-4
 - トランザクション
 - 解決 6-10
 - 管理 6-9
 - 表示 6-10

- 「エラー時に停止」設定
 - 「新規スケジュール・リンクの作成」プロパティ・シート 3-9

お

- オブジェクト
 - 定義
 - マスター・サイトへのレプリケーション 3-18
 - マスター・グループ
 - サポートの生成 3-24
 - マスター・グループから削除 3-21
 - マスター・グループでの変更 3-20
 - レプリケート
 - 再作成 6-21
- 「オブジェクト」ページ
 - 「新規マスター・グループの作成」プロパティ・シート 3-13
- 「オプション」ページ 2-28
 - 「スナップショットの作成」プロパティ・シート 2-13
- オフライン・インスタンスエーション 3-20
 - サポートする A-4
 - スナップショット 9-36, 9-38
 - マスター・サイト 7-15–7-16
 - レプリケート・オブジェクト・グループ 9-31, 9-32, 9-33, 9-34, 9-35

か

- 解消の統計
 - 収集 6-15
 - 表示 6-15
- 解消方法
 - 上書き 5-16
 - 加算、ビルトイン 5-14
 - 管理 5-13
 - 競合
 - 削除 5-13
 - 最古のタイムスタンプ 5-15
 - 最小値による方法 5-14
 - 最新のタイムスタンプ 5-15
 - 最大値による方法 5-15
 - 廃棄 5-16
 - 平均、ビルトイン 5-14
 - 列グループ

順序付け 5-13
 回復 6-14
 レプリケーション用 6-14
 拡張
 新機能 A-2
 加算によるビルトインの競合解消方法 5-14
 カタログ、レプリケーション 1-17
 「間隔の式」設定
 「新規スケジュール・リンクの作成」プロパティ
 ・シート 3-8
 セットアップ・ウィザード 3-11
 「間隔の設定」ダイアログ
 Replication Manager 3-8, 3-11
 監査 6-15
 競合解消 6-15
 監視
 基本レプリケーション環境 2-43
 完全リフレッシュ 1-8
 管理
 基本レプリケーション環境 2-27-2-43
 管理者
 アカウント
 スナップショット・サイト 4-5, 4-6
 管理者アカウント
 マスター・サイト 3-4
 「管理」フォルダ 6-8, 6-10
 管理要求 1-17, 6-4-6-7
 削除 6-5
 適用 6-5
 表示 6-4
 問題の診断方法 6-7
 「管理要求」フォルダ 6-4

 き
 記憶領域オプション
 PCTFREE 2-9
 PCTUSED 2-9
 記憶領域設定
 スナップショット用に編集 4-19
 記憶領域パラメータ
 スナップショット 2-13
 スナップショット・ログ 2-8
 変更 2-28
 記憶領域ページ
 「スナップショット・ログ」プロパティ・シート

2-28
 「スナップショットの作成」プロパティ・シート
 2-13
 基本レプリケーション 1-2, 1-4-1-11, 2-1-2-43
 SNP バックグラウンド・プロセス 2-6
 環境
 読み込み専用スナップショット 2-33-2-36
 環境の監視 2-43
 環境の管理 2-27-2-43
 環境の構成 2-2-2-27
 例 2-2
 環境の設計 2-2
 権限 2-5
 使用方法 1-4
 基本レプリケーションにおける 1-5
 きめ細かな休止 A-3
 キュー
 遅延トランザクション 1-20
 休止
 ~ときめ細かな休止 A-3
 マスター・グループ 3-15
 レプリケート・スキーマ 9-124
 旧プロシージャ B-13
 行
 競合検出時の識別 5-6
 競合
 Oracle での検出方法 5-5
 一意性 1-22
 一意性の 5-3
 防止 5-4
 解消 1-24
 RepResolution_Statistics 表、削除 6-15
 一意性の 5-7
 構成 5-28
 監査 6-15
 競合の検出 5-5
 更新の
 列グループ 5-6
 構成
 概要 5-10
 削除
 方法 7-24
 削除の 5-7
 削除の、構成 5-30

- 自動と手動 5-6
- 情報
 - 表示 5-46
- 情報のビュー 5-46
- 設計および準備のガイドライン 5-10
- 宣言方法、更新の競合 5-13
- 同期伝播での 7-7
- 統計 9-72, 9-117
 - 取消し 6-15
- 統計収集 6-15
- プロシージャ・レプリケーション 7-3
- 方法 1-24
- 方法の追加 9-60
- 最も高い優先順位 5-25
- ユーザー定義ルーチン
 - パラメータ 5-39
- ユーザー定義ルーチン、制限 5-41
- ユーザー定義ルーチン、例 5-41
- 解消および通知方法
 - ユーザー定義 5-9
- 解消方法 5-7
 - 一意性の、削除 5-28
 - 一意性の、割当て 5-28
 - 上書き 5-16
 - 管理 5-13
- 更新の
 - ビルトイン 5-7
 - 削除 5-13
 - 削除の、割当て 5-30
 - 削除の、削除 5-30
 - 使用、通知 5-10
 - 廃棄 5-16
 - ビルトイン、制限 5-8
 - 複数の
 - 使用方法 5-9
 - 平均、ビルトイン 5-14
 - ユーザー定義 5-39
- 加算による解消方法 5-14
- 行レベル・レプリケーション 1-24
- 検出 1-24, 5-5, 7-8
 - 行の識別 5-6
 - マスター・サイト、防止 5-5
- 更新 1-22
- 更新の 5-3

- 防止 5-5
- 更新の～の解消
 - サイト優先順位の使用 5-23
- 最小値による競合解消方法 5-14
- 最大値による競合解消方法 5-15
- 削除 1-22
- 削除の 5-3
 - 防止 5-4
- 通知
 - パッケージ、サンプル 5-44
- 通知パッケージ
 - 作成 5-43
- 通知方法
 - ユーザー定義 5-43
- 通知ログ
 - 作成 5-43
- 通知ログ表
 - サンプル 5-43
- データ・モデル 1-22
- プロシージャ・レプリケーション 1-25
- 防止 5-3
 - 動的所有権 7-26
- 列グループ 1-24
- 列グループの～解消方法
 - 順序付け 5-13
- レプリケーション 1-22
- 競合の検出 5-5
- 共有所有権 1-23
- 行レベル・レプリケーション 1-17
 - 競合の検出 1-24, 5-5

く

- クラスタ・ページ
 - 「スナップショットの作成」プロパティ・シート 2-14
- クラスタ化スナップショット 2-14
- 「グループ化されていないスナップショット」フォルダ 4-21
- 「グループへのオブジェクトの追加」ダイアログ
 - Replication Manager 3-17

け

- 結合
 - 単純副問合せスナップショット 2-24
- 権限

- 基本レプリケーション環境 2-3, 2-5
- スナップショット・ログの行の削除 2-30
- スナップショット・ログの作成 2-8
- スナップショットの削除 2-36
- スナップショットの作成 2-11
- スナップショットの変更 2-35

こ

- 交差部表 2-20—2-23
- 更新
 - min_communication 5-35
 - コメント 6-25
 - 通信の最小化 5-34
 - レプリケーション表 6-25
- 更新可能スナップショット 1-13
 - 伝播メカニズム 1-20
 - プロパティ 1-13
 - レプリケーション・サポートの生成 4-19
- 更新の競合 1-22, 5-3
 - 解消 5-39
 - サイト優先順位の使用 5-23
 - データ伝播の最小化 5-34
 - 列グループ 5-6
 - 解消方法
 - ビルトイン 5-7
 - 割当て 5-13
 - 防止 5-5
- 更新の競合解消の管理 5-13
- 構成
 - 基本レプリケーション環境 2-2—2-27
 - 例 2-2
 - 「構成」フォルダ 3-25, 6-4
- 高速リフレッシュ 1-8
 - スナップショットの要件 2-17
 - 表の再編成 A-4
- コマンド、SQL
 - ALTER SNAPSHOT LOG 2-28
 - CREATE SNAPSHOT 2-7, 2-10, 2-19—2-21
 - CREATE SNAPSHOT LOG 2-7, 2-17
 - CREATE TABLE 2-16
 - DROP SNAPSHOT 2-35
 - DROP SNAPSHOT LOG 2-33
 - TRUNCATE 2-29
 - TRUNCATE TABLE 2-31
- コメント

- 更新 6-25
- コメント・フィールド
 - ビュー内の更新 6-25

さ

- 最古および最新のタイムスタンプ
 - 競合解消方法 5-15
- 最小化
 - 更新と min_communication 5-35
- 最小値による競合解消方法 5-14
- 最新のタイムスタンプ
 - 競合解消方法 5-15
- 最大値による競合解消方法 5-15
- サイト
 - 優先順位レベルによる削除 5-25
- サイト所有権
 - 動的 5-4
 - プライマリ 5-3
- サイトの優先順位レベル
 - 変更 5-24
- サイト名の追加
 - 競合解消方法 5-29
- サイト優先順位
 - 管理 5-23
 - タイムスタンプによる競合解消方法の予備方法 5-15
 - 変更 9-68
- サイト優先順位グループ 5-23
 - 削除 5-25, 9-98
 - 作成 5-24
 - 構文 9-89
 - 使用方法 5-23
 - メンバーの削除 5-25, 9-99
 - メンバーの追加 5-24, 9-58
 - メンバーの変更
 - 値 5-25
 - 優先順位 5-24

索引

- スナップショット 2-39
- パーティション表 A-4

削除

- RepCatLog 表 9-113
- 競合解消方法 5-13
- サイト

- 優先順位レベルによる 5-25
- サイト優先順位グループ 5-25, 9-98
 - メンバー 5-25
- スケジュール・リンク 3-11
- スナップショット 2-35, 4-19
- スナップショット・サイト 9-100
- スナップショット・ログ 2-29, 2-33, 4-11
- 遅延トランザクション・キュー 3-11
- 統計 9-113
- マスター・グループ 3-15
- マスター・サイト 9-119
- 優先順位グループ 5-23, 9-96
- 優先順位グループのメンバー
 - 値による ~ 5-22
 - 優先順位による ~ 5-22
- 列グループ 5-12
 - 構文 9-91
- 列グループの列 5-12
- レプリケート・オブジェクト
 - グループ 9-93
 - スナップショット・サイトから
 - 構文 9-101
 - マスター・サイトから
 - 構文 9-94
- 削除の競合 1-22, 5-3
 - 解消 5-7, 5-40
 - 構成 5-30
 - 方法、割当て 5-30
 - 解消方法
 - 削除 5-30
 - 防止 5-4, 7-24
- 作成
 - 基本レプリケーション環境 2-2-2-27
 - 例 2-2
 - サイト優先順位グループ 5-24
 - 構文 9-89
 - スナップショット 2-3, 4-16
 - 単純 2-10
 - スナップショット・サイト
 - 構文 9-83
 - スナップショット・リフレッシュ・グループ 2-3
 - スナップショット・ログ 2-3, 2-6
 - スナップショット用のデータベース・リンク 2-2
 - 遅延トランザクション 8-2

- マスター・サイト 3-4
- 優先順位グループ 5-19, 9-88
- リフレッシュ・グループ 9-48
- 列グループ
 - 構文 9-87, 9-112
- レプリケート・オブジェクト
 - 構文 9-80
 - サポートの生成 9-105, 9-106, 9-108
 - スナップショット・サイト 9-84
- レプリケート・オブジェクト・グループ
 - 構文 9-79

し

- 「次回日付」設定
 - 「新規スケジュール・リンクの作成」プロパティ・シート 3-8
- セットアップ・ウィザード 3-11
- 次回日付フィールド
 - 「リフレッシュ・グループの作成」プロパティ・シート 2-26

識別列

- 競合の検出 1-24
- システム・ベースのセキュリティ A-5
- 自動競合解消
 - 手動に対する 5-6
- 自動リフレッシュ
 - リフレッシュ・グループ 1-9
 - リフレッシュ間隔 1-9
- シノニム
 - 作成、スナップショットに基づく 2-34
- 支部自動化 A-3
- シャドウ列グループ 1-24, 5-7
 - 「終了」ページ
 - Replication Manager 3-6
 - セットアップ・ウィザード 3-6
- 主キー
 - スナップショットのアップグレード B-11
 - レプリケート表から削除 3-14
- 主キー・スナップショット 1-10, A-3
 - ログ作成の要件 2-8
- 受信者 1-21
 - アカウント
 - マスター・サイト 3-4
- 手動競合解消
 - 自動に対する 5-6

- 手動リフレッシュ 1-10
- 順序付け
 - 列グループの競合解消方法 5-13
- 順序の競合
 - 防止 5-32
- 順序の追加
 - 競合解消方法 5-29
- 障害、媒体
 - 読み込み専用スナップショットの回復 2-36
 - 「使用可能」設定
 - 「新規スケジュール・リンクの作成」プロパティ・シート 3-8
- 使用禁止にする
 - 伝播 9-28
- 状態
 - 伝播 9-16
- 情報
 - 移送 1-5
 - オフロード
 - 基本レプリケーションにおける 1-5
 - 統合
 - アドバンスド・レプリケーション 1-13
 - 配布
 - 基本レプリケーションにおける 1-4
- 情報の配布
 - 基本レプリケーション 1-4
- 初期化パラメータ
 - JOB_QUEUE_INTERVAL 2-6, 4-9, 6-6
 - JOB_QUEUE_PROCESSES 2-6, 4-9, 6-6
 - PARALLEL_MAX_SERVERS 3-28
 - PARALLEL_MIN_SERVERS 3-28
 - PARALLEL_SERVER_IDLE_TIME 3-28
- ジョブ
 - キュー
 - ジョブの削除 9-30
 - 手動実行 6-13
 - 使用可能にする 6-13
 - スケジュール・リンク用のチェック 6-22
 - ブレイク 6-13
 - 問題の診断方法 6-13
 - ローカル
 - 表示 6-11
 - プロパティの編集 6-12
- ジョブ・キュー 1-20
- ジョブ・キュー・プロセス 3-7, 4-9

- シリアル伝播 1-20
 - スケジュール・リンク 3-8
 - 「新規スケジュール・リンクの作成」プロパティ・シート 3-8
 - 「一般」ページ 3-8
 - 「オプション」ページ 3-8
 - 「新規スナップショット・サイト」ダイアログ
 - セットアップ・ウィザード 4-6
- 新機能
 - LOB のサポート A-3
 - 管理および使用 A-3
 - スナップショットのための副問合せ A-3
 - セキュリティの拡張 A-5
 - データのサブセット化と副問合せ A-3
 - パフォーマンス A-2
 - 副問合せのスナップショット A-3
- 新機能の管理と使用 A-3
 - 「新規マスター・グループの作成」プロパティ・シート
 - Replication Manager 3-13
 - 「一般」ページ 3-13
 - 「オブジェクト」ページ 3-13
 - 「接続先」ページ 3-13
 - 「新規マスター・サイト」ダイアログ
 - セットアップ・ウィザード 3-4
 - 「新規レプリケート・オブジェクト・スキーマ」ダイアログ
 - セットアップ・ウィザード 3-5
- す
- スキーマ
 - スナップショット・サイト用 2-2, 2-4
 - スナップショット・サイト用の作成 4-6
 - スナップショット用の作成 2-4
 - マルチマスター・レプリケーションの作成 3-5
- スケジュール・パージ
 - ガイドライン 3-12
 - 定期的なパージ 3-12
 - 編集 3-12
 - 「スケジュール」フォルダ
 - Replication Manager 3-10
- スケジュール・リンク
 - ガイドライン 3-9
 - 管理 3-7
 - 削除 3-11

- 作成 3-8
 - スナップショット・サイト 4-7
- 使用するデータベース・リンク 3-8
- 使用するバックグラウンド・プロセスの数 3-9
- 状態の表示 3-10
- シリアル伝播 3-8, 3-27
- 定期的な送信 3-10
- パラレル伝播 3-8, 3-27–3-28
- 編集 3-10
- 連続的な送信 3-10, 3-12
- 「スケジュール・リンク」フォルダ
 - Replication Manager 3-10
- 「スケジュール・リンクの作成」プロパティ・シート
 - セットアップ・ウィザード 3-10, 3-12
- 「スケジュール・リンクの設定」ダイアログ
 - Replication Manager 3-8
- 「スケジュール・リンクの編集」プロパティ・シート
 - Replication Manager 3-10
 - セットアップ・ウィザード 3-10
- 「スケジュール作成」ページ
 - 「リフレッシュ・グループの作成」プロパティ・シート 4-20
- スナップショット 2-5
 - ROWID 1-11, 2-42
 - SNAPSHOTS ビュー 10-23
 - SQL
 - 単純 ~ に関する制限事項 2-13
 - アクセス権限の付与 2-3, 4-4
 - オフライン・インスタンスエーション 9-36, 9-38
 - 管理 4-15–4-20
 - 記憶領域オプション 2-13
 - 記憶領域設定の編集 4-19
 - 基礎オブジェクト 1-7
 - 基本レプリケーションのクローニング 7-14
 - クラスタ化 2-14
 - グループ 1-16
 - 上級管理テクニック 6-2
 - クローニング
 - オフライン・インスタンスエーション 7-13
 - 更新 1-7
 - 更新可能 1-13
 - 遅延制約 A-4
 - 伝播メカニズム 1-20
 - 高速リフレッシュの要件 2-17
 - サイト 1-16
 - スキーマ 2-4
 - マスターの変更 6-3
- 索引作成 2-39
- 削除
 - 必要な権限 2-36
- 作成 2-3, 4-16
 - WHERE 句を使用する 4-17
 - 必要な権限 2-11
 - 副問合せを持つ 4-18
- サポートされているデータ型 4-16
- 実表で PL/SQL トリガーを使用する 4-19
- 実表の操作 4-19
- シノニムの作成 2-34
- 主キー 1-10, 2-12, A-3
 - ログ作成の要件 2-8
- 主キー ~ へのアップグレード B-11
- 情報の表示 2-34
- スキーマの作成 2-4
- スナップショット・グループから削除 4-19
- 単純
 - 構造体 1-6
- 単純副問合せ
 - AND 式 2-24
 - EXISTS 句 2-23
 - 結合 2-24
 - 高度なサブセット化 2-16
 - フィルタ列要件 2-9, 2-23
 - マスター表の列数 2-24
 - 要件 2-9
 - 例 2-16–2-21
- チューニング 2-40
- 定義問合せ 1-6, 2-12
- 定義の変更 4-19
- データ・ディクショナリ・ビュー 10-22–10-28
- データ・ロード・オプション 2-15
- トラブルシューティング 6-23
- トレース・ファイル 2-39
- パフォーマンスのチューニング 2-39
- ビューの作成 2-34
- 複合 1-10, 2-40
 - PCTFREE の値 2-42
 - PCTUSED の値 2-42
- 副問合せ 1-7, 2-15
 - 複数対 1 参照 2-19
 - 複数対複数参照 2-19

- 副問合せスナップショット、制限事項 2-23
- 副問合せで作成 2-15
- 変更
 - 必要な権限 2-35
- 編集 4-19
- マスター表の削除 2-36
- 命名 2-11
- 読み込み専用 1-5
 - 管理 2-33–2-36
 - 索引 1-7
 - 削除 2-35
 - 実表 1-7
 - 使用方法 2-33
 - 問合せに必要な権限 2-34
 - 登録 2-35
 - 登録解除 2-35
 - ビュー 1-7
 - 副問合せを持つ単純な A-3
 - 変更 2-35
- リフレッシュ 1-7, 1-10, 9-137
 - 再試行 2-38
 - 最終リフレッシュ時刻の問合せ 2-35
 - 障害 2-38
 - スナップショットの個別リフレッシュ 2-42
 - トラブルシューティング 2-38, 2-39
 - 問題 6-23
 - ロールバック・セグメント 2-27
- リフレッシュ・グループ 4-20
 - 削除 2-38
 - 作成 2-24
 - 手動リフレッシュ 2-38
 - 設計 2-5
 - 設定の変更 2-37
 - データ・ディクショナリ・ビュー 10-22
 - メンバーの削除 2-37
 - メンバーの追加 2-37
- リフレッシュ・グループの割当て 2-3
- リフレッシュ・タイプ 2-27
- リフレッシュの設定 2-12
- リモート・データの参照 2-13
- レプリケーション
 - セキュリティ 7-21
 - レプリケーションを使用可能 / 使用禁止にする 7-32
- スナップショット・グループ

- 管理 4-11–4-15
- マスター・グループの選択 4-4
- マスター・サイトの選択 4-4
- レプリケートするオブジェクトの選択 4-4
- スナップショット・グループ・ウィザード
 - Replication Manager 4-3
- スナップショット・サイト
 - SNP バックグラウンド・プロセス 4-9
 - アカウント構成 4-6
 - アドバンスト
 - 代替セキュリティの設定 7-23
 - アドバンスト・レプリケーション環境に追加する 4-2–4-9
 - オフライン・インスタンスエーション
 - アドバンスト・レプリケーション 7-16
- 構成 4-3
- 削除 9-100
- 作成
 - 構文 9-83
- スキーマ 2-2
- スキーマ作成 4-6
- スケジュール・ページ
 - ガイドライン 3-12
- スケジュール・リンク 3-7
 - ガイドライン 3-9
- 設計 4-2
- 設定のカスタマイズ 4-7
- 設定を調べる 4-8
- データベース・リンク 2-2, 2-5
- マスター・サイトの指定 4-5
- マスターの変更 9-125
- マスターへの変更の伝播 9-26
- リフレッシュ
 - 構文 9-114
- リフレッシュ・グループの作成 4-20
- レプリケーション伝播担当者 4-5
- レプリケーション管理者 4-5
- レプリケーションの設定 4-4
- スナップショット・サイト・レプリケーション 4-1–4-22
- スナップショット・サイトおよび移行 B-6
- 「スナップショット・サイトのカスタマイズ」プロパティ・シート
 - セットアップ・ウィザード 4-7
- 「スナップショット」ページ

- 「リフレッシュ・グループの作成」プロパティ・シート 4-20
- スナップショット・リフレッシュ 4-9
 - 完全 1-8
 - グループ 1-9
 - リフレッシュ間隔 1-9
 - 高速 1-8
 - スナップショット・ログ 1-8
- スナップショット・リフレッシュ・グループ管理 2-5-2-39
 - 削除 2-38
 - 作成 2-3
 - 手動リフレッシュ 2-38
 - 設定の変更 2-37
 - メンバーの削除 2-37
 - メンバーの追加 2-37
- スナップショット・ログ
 - 記憶領域パラメータ 2-8
 - 基礎表 1-8
 - 行の削除
 - 必要な権限 2-30
 - 切捨て 2-29
 - 必要な権限 2-30
 - 削除 2-29, 2-33, 4-11
 - 手動 2-29
 - 必要な権限 2-29, 2-33
 - 作成 2-3, 2-6, 4-3, 4-10
 - 必要な権限 2-8
 - 作成時期 2-8
 - 内部操作 2-10
 - 内容 1-8
 - フィルタ列
 - 追加 2-28
 - 変更 2-27, 4-10
 - 必要な権限 2-27
 - マスター表
 - 削除 2-29, 2-33, 9-136
 - マスター表の切捨て 2-30
 - マスター表の再編成 2-30
 - 命名 2-8
 - 領域の管理 2-28
 - 領域割当ての削減 2-29
- 「スナップショット・ログ」プロパティ・シート 2-28
 - 「一般」ページ 2-34

- 「オプション」ページ 2-28
- 記憶領域ページ 2-28
- 「スナップショット・ログの作成」プロパティ・シート
 - Replication Manager 4-3, 4-10
 - 「一般」ページ 4-10
 - 「オプション」ページ 2-8
- 記憶領域ページ 2-8
 - 「表領域およびエクステント」ページ 4-10
 - 「フィルタ列」ページ 4-10
- 「スナップショット・ログの編集」プロパティ・シート
 - Replication Manager 4-10
 - 「一般」ページ 4-10
 - 「表領域およびエクステント」ページ 4-10
 - 「フィルタ列」ページ 4-10
- 「スナップショットの作成」プロパティ・シート
 - 「オプション」ページ 2-13
- 記憶領域ページ 2-13
- クラスタ・ページ 2-14
- スナップショットのための副問合せ A-3
- 「スナップショットの編集」プロパティ・シート
 - 「表領域およびエクステント」ページ 4-19

せ

制限事項

- ブローージャ・レプリケーション 7-2

生成

- スナップショット・サポート 9-110
- レプリケーション・サポート
 - ブローージャ・レプリケーション 7-3
- レプリケーション・サポートおよび移行 B-3

生成オブジェクト

- マスター・グループ用の表示 3-26
- 「生成オブジェクト」フォルダ 3-26
- 生成レプリケーション・オブジェクト
 - Wrapper パッケージ 1-18

- 静的所有権 1-22

- 制約の違反 6-21

セキュリティ A-5

- スナップショット・レプリケーション用 7-21
- 設定

- 代替 7-23

- マルチマスター・レプリケーション用 7-17

- 「接続先」ページ

- 「新規マスター・グループの作成」プロパティ・シート 3-13
- 「マスター・グループの編集」プロパティ・シート 3-22, 6-3
- 「接続先別遅延トランザクション」フォルダ 6-8
- 接続先マスター・サイト
 - マスター・グループへの追加 3-14, 3-22
- 接続先マップ
 - マスター・グループ用の表示 3-26
- 「接続先マップ」フォルダ 3-26
- 接続修飾子
 - マスター・グループ 3-14
- 設定 2-4
- セットアップ・ウィザード
 - 「DB 接続の編集」プロパティ・シート 3-12
- Replication Manager 3-3
- 作成
 - 管理者アカウント 3-4, 4-6
 - 受信者アカウント 3-4
 - 伝播担当者アカウント 3-4, 4-6
 - 「終了」ページ 3-6
 - 「新規マスター・サイト」ダイアログ 3-4
 - 「新規レプリケート・オブジェクト・スキーマ」ダイアログ 3-5
 - 「スケジュール・リンクの作成」プロパティ・シート 3-10, 3-12
 - 「スケジュール・リンクの編集」プロパティ・シート 3-10
- スナップショット・サイトを構成する 4-3
- 「スナップショットのカスタマイズ」プロパティ・シート 4-7
- 「マスター・グループの編集」プロパティ・シート 3-22
- 「マスター・サイトのカスタマイズ」プロパティ・シート 3-6
- 「マスター・サイトの作成」ページ 3-21
- マスター・スキーマ・リストから選択する 4-7

そ

相違

- 表の間の 9-39
- 調整 9-42
- 「操作」ページ
 - 「マスター・グループの編集」プロパティ・シート 3-16, 3-25, 6-2

送信

定期的

- スケジュール 3-10

連続的

- スケジュール 3-10, 3-12

即時

- データ・レプリケーション 1-26

- レプリケーション 1-26, 7-6

た

代替キー

- 競合の検出 1-24

- レプリケート表 3-18

- 「代替キー列の設定」ダイアログ

- Replication Manager 3-14, 3-18

- タイム・スタンプの競合解消方法

- サンプル・トリガー 5-26

- 大量配置 A-3

- 段階的な移行 B-5

単純スナップショット

- SQL 制限事項 2-13

- 構造体 1-6

- リフレッシュ・グループ

- 作成 2-24

単純副問合せスナップショット

- チューニング 2-40

- 複数対 1 参照 2-19, 2-23

- 複数対複数参照 2-19, 2-23

- 要件 2-9

ち

- 遅延 RPC 8-2

- 遅延制約および更新可能スナップショット A-4

- 遅延トランザクション 1-20

- DEFCALLDEST ビュー 10-18

- DefCallDest 表 8-4

- DefCall 表 8-4

- DEFDEFAULTDEST ビュー 10-18

- DefDefaultDest 表

- 接続先の追加 8-6, 9-14

- 接続先の取消し 8-6, 9-14, 9-15

- DEFERROR ビュー 10-19, 10-20

- DEFTRANDEST ビュー 10-22

- DEFTRAN ビュー 10-22

- DefTran 表 8-4

開始 8-4, 9-7
管理 6-8
キュー
 削除 1-20, 3-13
 送信 1-20
 マスター・サイト削除後の消去 3-23
キューからの削除 9-16
再実行 9-18
削除 3-11, 6-9
作成 8-2
実行 6-9
実行のスケジューリング 9-26
遅延リモート・プロシージャ・コール (RPC)
 作成 8-4, 9-4
 即時実行 9-22
 引数 8-5, 9-6
 引数型 9-9
 引数値 9-11
ビュー 10-17-10-22
表示 6-8
問題の診断方法 6-22
遅延秒数
 セットアップ・ウィザード 3-11
「遅延秒数」設定
 「新規スケジュール・リンクの作成」プロパティ
 ・シート 3-9
蓄積交換レプリケーション 1-18
調整
 表 6-16, 9-42
直列化
 トランザクションの 7-3

つ
追加
 サイト
 アドバンスド・レプリケーション環境への
 7-8
 サイト優先順位グループのメンバー 5-24
 優先順位グループのメンバー 5-20
 列グループの列 5-12
通信設定の最小化
 Replication Manager 4-18
 「レプリケーション・オブジェクトの編集」プロ
 パティ・シート 3-25
通知

複数の競合解消方法の使用 5-10
方法
 ユーザー定義、競合解消の 5-9
通知パッケージ
 競合
 作成 5-43
通知方法
 ユーザー定義 5-43
通知ログ
 競合
 作成 5-43
通知ログ表
 競合
 サンプル 5-43

て
定義サイト
 マスター・グループの再配置 6-3
定期的な送信
 スケジュール 3-10
定期的なバージ
 スケジュール 3-12
定義問合せ
 スナップショット 1-6, 2-12
データ
 収束 5-6
 保証 5-31
 整合性
 シリアル伝播 3-30
 パラレル伝播 3-30
 複数の列グループでの保証 5-7
 非一貫性
 マスター・サイト削除後の除去 3-23
データ・ウェアハウス
 基本レプリケーション 1-5
データ・クラスタ
 一部としてスナップショットを作成する 2-14
データ・ディクショナリ
 DBA_REGISTERED_SNAPSHOTS ビュー 2-34
データ・ディクショナリ・ビュー 10-2
 DEFCALL 10-18
 DEFCALLDEST 10-18
 DEFDEFAULTDEST 10-18
 DEFERRCOUNT 10-19
 DEFERROR 10-19, 10-20

- DEFLOB 10-20
- DEFPROPAGATOR 10-20
- DEFSCHEDULE 10-21
- DEFTRAN 10-22
- DEFTRANDEST 10-22
- REFRESH 10-26
- REFRESH_CHILDREN 10-28
- REGISTERED_SNAPSHOTS 10-24
- REPCATALOG 10-5
- REPCOLUMN 10-6
- REPCOLUMN_GROUP 10-6
- REPCONFLICT 10-7
- REPDDL 10-7
- REPGENERATED 10-8
- REPGENOBJECTS 10-16
- REPGROUP 10-3
- REPGROUPED_COLUMN 10-8
- REPKEY_COLUMNS 10-9
- REPOBJECT 10-9
- REPPARAMETER_COLUMN 10-10
- REPPRIORITY 10-11
- REPPRIORITY_GROUP 10-12
- REPPROP 10-12
- REPRESOL_STATS_CONTROL 10-14
- REPRESOLUTION 10-13
- REPRESOLUTION_METHOD 10-14
- REPRESOLUTION_STATISTICS 10-15
- REPSITES 10-16
- SNAPSHOT_REFRESH_TIMES 10-26
- SNAPSHOTS 10-23
- SNAPSHOTS_LOGS 10-25
- スナップショット 10-22
- 遅延トランザクション 10-17
- データ・レプリケーション
 - 即時 7-6
- データ型
 - サポートする A-3
 - スナップショットで可能な 2-12, 4-16
 - レプリケート表で許可 3-18
- データ所有権モデル 1-22
 - 共有所有権 1-23
 - 静的所有権 1-22
 - 動的所有権 1-23
 - プライマリ所有権 1-22
- データ操作言語 (DML)
 - 伝播の最小化 5-34
- データ定義言語 (DDL)
 - 非同期 ~ の供給 9-104
- データ伝播
 - および依存性の維持 3-29
 - 更新の競合解消のための最小化 5-34
 - 最小化 3-25
 - 削減 A-2
 - 同期 7-6
 - モード
 - マスター・サイトでの変更 7-10
- データのサブセット化 A-3
- データベース
 - マルチマスター・レプリケーションの設定 3-3
 - リンク
 - 異なるバスの使用方法 3-14
- データベース・オブジェクト
 - 読み込み専用スナップショット用 1-7
 - 「データベース・オブジェクト」フォルダ
 - マスター定義サイト 3-17
- データベース・リンク 6-19
 - スナップショット・サイト 2-5
 - スナップショット・サイト用の作成 2-2
 - 「データベース・リンク」設定
 - 「新規スケジュール・リンクの作成」プロパティ・シート 3-8
- データベース・リンクの指定
 - 不備 6-22
 - 「データベース接続先の表示」プロパティ・シート 6-9
 - 「データベース接続先の編集」プロパティ・シート Replication Manager 3-26
- データ要件
 - 評価および理解 5-2
- デッドロック
 - 解消
 - 同期伝播での 7-7
- 伝播
 - 使用禁止にする 9-28
 - 状態 9-16
 - シリアル 1-20
 - データの最小化 5-34
 - デフォルト特性の設定 3-5
 - 伝播担当者のセキュリティ・コンテキスト 7-6
 - パラレル 1-20, A-2

- 変更の
 - 伝播方法の変更 9-63
- モード 7-8
 - マスター・サイトでの変更 7-10
- 伝播担当者 1-21
 - アカウント
 - スナップショット・サイト 4-5, 4-6
 - 登録 9-24
- 伝播担当者アカウント
 - マスター・サイト 3-4
- 伝播の削減 A-2
- 伝播モード 7-8

と

- 問合せ、定義
 - スナップショット用 2-12
- 同期データ伝播 1-26, 7-6
- 同期をとってレプリケートされるトランザクション
 - 接続先 7-7
- 統計
 - 競合解消 6-15
 - 競合解消の監査 6-15
 - 削除 6-15
 - 収集する 9-117
- 動作の保証 7-10
 - Parallel Server 7-11
 - 設計上の考慮事項 7-12
- 動的サイト
 - 所有権 5-4
- 動的所有権 1-23
 - 競合防止 7-26
 - ワークフロー・パーティション分割 7-26
- 登録
 - スナップショット A-3
 - ローカル・データベースの伝播担当者 9-24
- トークン渡し 7-27
 - 実現例 7-26
- トラブルシューティング 6-19
- トランザクション
 - エラー
 - 表示 6-10
 - キュー
 - 遅延、問題の診断方法 6-22
 - 遅延
 - 削除 6-9

- 実行 6-9
- 直列化 7-3
- 伝播
 - 保護メカニズム 3-29–3-30
- トリガー
 - 内部 A-2
 - レプリケート 7-32
- トレース・ファイル
 - スナップショット 2-39

な

- 内部スナップショット・オブジェクト 1-7
- 内部トリガー A-2

は

- ページ
 - 手動 3-13
 - 定期的
 - スケジュール 3-12, 4-7
- パーティション
 - レプリケーションおよび 3-19
- パーティション表
 - 索引 A-4
- 廃棄
 - 競合解消方法 5-16
 - 使用 5-32
- 廃棄の一意性
 - 競合解消方法 5-30
- 媒体の障害
 - 回復
 - 読み込み専用スナップショット 2-36
- ハイブリッド構成
 - アドバンスド・レプリケーション 1-14
- バックアップ
 - レプリケーション用 6-14
- パッケージ変数
 - from_remote 9-143
 - global_name 9-143
 - i_am_a_refresh 9-136
 - i_am_a_snapshot 9-143
 - replication_is_on 9-143
- 「バッチ・サイズ」設定
 - 「新規スケジュール・リンクの作成」プロパティ・シート 3-9
- パフォーマンスの拡張 A-2

- パフォーマンスのチューニング
 - スナップショット 2-39
 - 単純副問合せスナップショット 2-40
- パラレル・サーバー・プロセス
 - アドバンスト・レプリケーション環境用の構成 3-28
- パラレル伝播 1-20, A-2
 - アドバンスト・レプリケーション環境 3-27–3-28
 - 依存性
 - 追跡 3-30
 - 計画 3-27
- 「パラレル伝播」設定
 - 「新規スケジュール・リンクの作成」プロパティ・シート 3-8

ひ

比較

- 表 9-39

非接続環境

- アドバンスト・レプリケーション 1-11

「日付の設定」ダイアログ

- Replication Manager 3-8, 3-11

非同期

- DDL 9-104

- RPC 8-2

- レプリケーション 1-18

ビュー

- DBA_REFRESH 2-43

- DBA_REFRESH_CHILDREN 2-43

- DBA_REGISTERED_SNAPSHOTS 2-34, 2-43

- DBA_RGROUPE 2-43

- DBA_SNAPSHOT_LOGS 2-35, 2-43

- DBA_SNAPSHOTS 2-43

- DEFCALL 10-18

- DEFCALLDEST 10-18

- DEFDEFAULTDEST 10-18

- DEFERRCOUNT 10-19

- DEFERROR 10-19, 10-20

- DEFLOB 10-20

- DEFPROPAGATOR 10-20

- DEFSCHEDULE 10-21

- DEFTRAN 10-22

- DEFTRANDEST 10-22

- REFRESH 10-26

- REFRESH_CHILDREN 10-28

- REGISTERED_SNAPSHOTS 10-24

- REPCATALOG 3-27, 6-6–6-7, 10-5

- REPCOLUMN 10-6

- REPCOLUMN_GROUP 10-6

- REPCONFLICT 10-7

- REPDDL 10-7

- REPGENERATED 10-8

- REPGENOBJECTS 10-16

- REPGROUP 3-27, 10-3

- REPGROUPED_COLUMN 10-8

- REPKEY_COLUMNS 10-9

- REPOBJECT 3-27, 10-9

- REPPARAMETER_COLUMN 10-10

- REPPRIORITY 10-11

- REPPRIORITY_GROUP 10-12

- REPPROP 10-12

- REPRESOL_STATS_CONTROL 10-14

- REPRESOLUTION 10-13

- REPRESOLUTION_METHOD 10-14

- REPRESOLUTION_STATISTICS 10-15

- REPSITES 3-27, 10-16

- SNAPSHOT_REFRESH_TIMES

- SNAPSHOT_REFRESH_TIMES ビュー 10-26

- SNAPSHOTS 10-23

- SNAPSHOTS_LOGS 10-25

- USER_REFRESH 2-39

- USER_REFRESH_CHILDREN 2-39

- 作成、スナップショットに基づく 2-34

- レプリケーション・カタログ 10-2

表

- 交差部 2-20–2-23

- 再編成 A-4

- 調整 6-16, 9-42

- パーティション

- レプリケート 3-19

- パーティション～と索引 A-4

- 比較 9-39

- 変更

- 変更をレプリケートしない 7-30

- マスター・グループ

- レプリケーション・サポートの生成 3-24

- レプリケーション・サポートの生成の問題 6-21

表データ

- マスター・サイトへのレプリケート 3-19

表の自己参照型制約

- マスター・グループへのサイトの追加 3-22
- 表の循環型依存
 - マスター・グループへのサイトの追加の制限事項 3-22
- 「表領域およびエクステント」ページ
 - 「スナップショット・ログの作成」プロパティ・シート 4-10
 - 「スナップショット・ログの編集」プロパティ・シート 4-10
 - 「スナップショットの編集」プロパティ・シート 4-19
- ビルトインの一意性の競合
 - 解消方法 5-8
- ビルトインの競合
 - 解消方法
 - 制限事項 5-8
- ビルトインの更新の競合
 - 解消方法 5-7
- ビルトインの更新の競合解消方法 5-13

ふ

- フィルタ列
 - 単純副問合せスナップショット 2-9, 2-23
- 「フィルタ列」ページ
 - 「スナップショット・ログの作成」プロパティ・シート 4-10
 - 「スナップショット・ログの編集」プロパティ・シート 4-10
- フェイルオーバー・サイト
 - アドバンスド・レプリケーション 1-11
 - インプリメント 7-13
- 複合スナップショット 1-10, 2-40
 - PCTFREE の値 2-42
 - PCTUSED の値 2-42
- 複数更新 6-21
- 複数対 1 参照
 - 単純副問合せスナップショット 2-23
- 複数対複数参照
 - 単純副問合せスナップショット 2-23
- 複数の競合
 - 解消方法
 - 使用方法 5-9
 - 通知に使用 5-10
 - 予備の 5-10
- 複数の列グループ

- データ整合性の保証 5-7
- 副問合せ
 - スナップショットの作成 4-18
- 副問合せのスナップショット 1-7, 2-15
 - AND 式 2-24
 - EXIST 句 2-23
 - 結合 2-24
 - 高度なサブセット化 2-16
 - 作成 4-18
 - チューニング 2-40
 - フィルタ列要件 2-9, 2-23
 - 複数対 1 参照 2-19, 2-23
 - 複数対複数参照 2-19, 2-23
 - マスター表の列数 2-24
 - 要件 2-9
- 再び使用可能にする
 - アドバンスド・レプリケーション 7-32
- プライマリ・サイト
 - 所有権 5-3
- プライマリ所有権 1-22
- プロシージャ・レプリケーション 1-25
 - Wrapper 1-25
 - 競合 7-3
 - 競合の検出 1-25
 - 削除 7-24
 - サポートの生成 7-3
 - 使用方法 7-2
 - 制限事項 7-2
- 「プロセス」設定
 - 「新規スケジュール・リンクの作成」プロパティ・シート 3-9
- 分散トランザクション
 - 問題 6-22

へ

- 平均
 - 競合解消方法
 - 使用 5-32
- 平均によるビルトインの競合解消方法 5-14
- 変更
 - サイト優先順位グループのメンバー値 5-25
 - 優先順位 5-24
 - スナップショット
 - 必要な権限 2-35

スナップショット定義 4-19
 伝播方法 9-63, 9-71
 表
 変更をレプリケートしない 7-30
 優先順位グループのメンバー
 値 5-20
 優先順位 5-21
 優先レベル 9-66
 レプリケート・オブジェクト 9-64
 変更の伝播
 伝播方法の変更 9-71

ま
 マスター・グループ 1-16
 DDL の実行 6-2
 オブジェクト
 サポートの生成 3-24
 オブジェクト定義の変更 3-20
 オブジェクトの表示 3-25
 管理 3-13–3-26
 管理要求
 削除 6-5
 管理要求の適用 6-5
 検査 6-2
 サイトへの DDL の伝播 6-2
 削除 3-15
 作成 3-13
 自己参照型制約がある表を持つサイトの追加 3-22
 循環型依存がある表を持つサイトの追加 3-22
 情報の表示 3-25
 生成オブジェクトの表示 3-26
 接続先マスター・サイト 3-14
 接続先マスター・サイトの追加 3-21, 3-22
 接続先マップの表示 3-26
 接続修飾子 3-14
 選択、スナップショット・グループ 4-4
 定義サイト
 再配置 6-3
 変更に応じたレプリケーション・サポートの生成 3-23
 マスター・サイトの削除 3-22
 マスター定義サイトの再配置 6-3
 リスト 3-25
 レプリケーション・アクティビティの再開 3-16

レプリケーション・アクティビティの中断 3-15
 レプリケーション・オブジェクトの削除 3-21
 レプリケーション・オブジェクトの追加 3-14, 3-17
 「マスター・グループ」フォルダ 3-25, 6-4
 マスター・グループおよびスナップショット・グループ
 上級管理テクニック 6-2
 マスター・グループおよびスナップショット・グループの管理
 上級 6-2
 「マスター・グループの編集」プロパティ・シート
 Replication Manager 3-16, 3-17
 「接続先」ページ 3-22, 6-3
 セットアップ・ウィザード 3-22
 「操作」ページ 3-16, 3-25
 マスター・サイト 1-16
 オブジェクト定義のレプリケート 3-18
 手動 3-19
 オフライン・インスタンスエーション 7-15–7-16
 管理者アカウント 3-4
 きめ細かな休止 A-3
 削除 9-119
 作成 9-55
 作成、マルチマスター・レプリケーション 3-4
 受信者アカウント 3-4
 スキーマ作成 3-5
 スケジュール・ページ
 ガイドライン 3-12
 スケジュール・リンク 3-7
 ガイドライン 3-9
 スナップショット・サイト用の指定 4-5
 スナップショット登録 A-3
 設定のカスタマイズ 3-6
 選択、スナップショット・グループ 4-4
 相違の判別 6-16
 段階的な移行 B-7
 データ伝播の最小化 3-25
 データのレプリケート 3-19
 デフォルトの伝播特性 3-5
 伝播担当者アカウント 3-4
 表データの手動ロード 3-19
 変更の伝播 9-26
 マスター・グループ
 管理 3-13–3-26

- マスター・グループから削除 3-22
- マスター・グループへの追加 3-21
- 問題の診断方法 6-19
- マスター・サイトおよび移行 B-3
- マスター・サイトでのスナップショット登録 A-3
- 「マスター・サイトのカスタマイズ」プロパティ・シート
 - セットアップ・ウィザード 3-6
- 「マスター・サイトの作成」ダイアログ
 - セットアップ・ウィザード 3-21
- マスター・スキーマ・リストから選択する
 - セットアップ・ウィザード 4-7
- マスター定義サイト 1-16
 - 再配置 6-3, 9-117
- マスター表
 - 許可されるデータ型 2-12
 - 切捨て 2-30
 - 再編成 2-30
 - 方法 2-31–2-33
 - 削除 2-36
 - 主キー・スナップショットの特別要件 2-8
 - スナップショット・ログ 1-8, 2-6
 - スナップショットの PRIMARY KEY 制約 2-12
 - 列
 - 単純副問合せスナップショットに関する数値制限 2-24
- マルチマスター・レプリケーション 1-12
 - アドバンスド・オプション 3-27–3-30
 - 環境の作成 3-1–3-3
 - 例 3-2
 - 構成の設定の確認 3-6
 - 作成
 - マスター・グループ 3-13
 - システムの監視 3-25–3-27, 6-4–6-13
 - スキーマ作成 3-5
 - スケジュール・リンク 3-7
 - セキュリティ 7-17
 - セットアップ・ウィザード 3-3
 - データベースの設定 3-3
 - トランザクション伝播保護 3-29–3-30
 - マスター・サイトの作成 3-4
 - マスター・サイトの設定のカスタマイズ 3-6
 - ローカル・ジョブ・キュー 6-11

め

命名

- スナップショット 2-11

メンバー

- 優先順位の変更 5-21

も

- 問題の診断方法 6-19

ゆ

ユーザー定義

- 通知方法 5-43

- ユーザー定義の競合解消方法 5-39

優先順位グループ 5-18

- 競合解消に使用 5-18

- サイト優先順位 5-17

- サイト優先順位グループ

- メンバーの追加 9-58

- 削除 5-23, 9-96

- 作成 5-19, 9-88

- メンバーの削除 9-95, 9-97

- 値による ~ 5-22

- 優先順位による ~ 5-22

- メンバーの追加 5-20, 9-57

- メンバーの変更

- 値 5-20, 9-67

- 優先順位 5-21, 9-66

優先順位グループのメンバー

- 値による削除 5-22

- 削除 5-22

よ

予備

- 複数の競合解消方法の使用 5-10

読み専用 1-2

- 使用方法 1-4

読み専用スナップショット 1-5

- 管理 2-33–2-36

- 索引 1-7

- 削除 2-35

- 実表 1-7

- 使用方法 2-33

- 問合せに必要な権限 2-34

- 登録 2-35

- 手動 2-35
- 登録解除 2-35
- 媒体の障害からの回復 2-36
- ビュー 1-7
- 変更
 - 必要な権限 2-35
- リフレッシュ・タイプ 1-9
- 読み込み専用レプリケーション 1-4-1-7

ら

- ラージ・オブジェクト型
- レプリケート表で許可 3-18

り

- リフレッシュ
 - 高速～と表の再編成 A-4
 - 再試行 2-38
 - 障害 2-38
 - スナップショット 4-9, 9-137
 - スナップショット・サイト
 - 構文 9-114
- リフレッシュ、計画 2-5
- リフレッシュ・グループ 1-9
 - REFRESH CHILDREN ビュー 10-28
 - REFRESH ビュー 10-26
 - 管理 2-5-2-39
 - グループ・メンバーのリスト 10-28
 - 削除 2-38, 9-47
 - 作成 2-3, 2-24, 4-3
 - スナップショット・サイト用 4-20
 - 自動リフレッシュ 1-9
 - 手動リフレッシュ 1-10, 2-38
 - 新規作成 9-48
 - スナップショット 4-20
 - スナップショットの追加 4-21
 - 設計 2-5
 - 設定
 - 次回日付 2-26
 - 設定の変更 2-37
 - データ・ディクショナリ・ビュー 10-22
 - トラブルシューティング 2-38
 - メンバーの削除 2-37, 9-51
 - メンバーの追加 2-37, 9-44
- リフレッシュ
 - 手動で 9-50

- リフレッシュ間隔 2-26
 - 変更 9-45
- 「リフレッシュ・グループの作成」プロパティ・シート
 - Replication Manager 4-3
 - 「一般」ページ 4-20
 - 次回日付フィールド 2-26
 - 「スケジュール作成」ページ 4-20
 - 「スナップショット」ページ 4-20
- リフレッシュ・タイプ
 - 手動 1-10
 - スナップショット 2-27
 - デフォルト 1-9
 - 読み込み専用スナップショット 1-9
- リフレッシュ間隔
 - REFRESH ビュー 10-26
 - スナップショット・リフレッシュ・グループ 1-9
 - パラメータ制約 1-9
 - リスト 10-26
- リフレッシュの設定
 - スナップショット 2-12
- リモート・データ
 - スナップショットの参照 2-13
- リモート・プロシージャ・コール 1-19
 - 遅延 8-4
- 領域
 - スナップショット・ログの削減 2-29
- リンクの指定
 - 不備 6-22

れ

例

- 最小化
 - 通信 5-34

列

- 列グループの～の追加および削除 5-12

列グループ 1-24

- 更新の競合解消 5-6
- 削除 5-12
 - 構文 9-91
- 作成
 - 構文 9-87, 9-112
- シャドウ 1-24, 5-7
- メンバーの削除
 - 構文 9-92

- メンバーの追加
 - 構文 9-54
- レプリケーション
 - アドバンスト
 - 競合の概要 5-2
 - 再び使用可能にする 7-32
 - アドバンスト、使用方法 1-11
 - オフ / オンが現行のセッションに影響する 7-31
 - オブジェクト
 - 生成 1-18
 - オブジェクト定義をマスター・サイトに 3-18
 - 手動 3-19
 - カタログ 1-17
 - カタログ・ビュー 10-2
 - 環境
 - アドバンスト
 - 新規サイトの追加 7-8
 - 管理者 1-21
 - 基本 1-2, 1-4—1-11, 2-1—2-43
 - 競合 1-22
 - 解消 1-24
 - 解消方法 1-24
 - 行レベル・レプリケーション 1-24
 - 検出 1-24
 - データ・モデル 1-22
 - プロシージャ・レプリケーション 1-25
 - 防止 1-24
 - 列グループ 1-24
 - グループ 1-16
 - サイト 1-16
 - 受信者 1-21
 - 使用可能にする 7-30
 - 上級テクニック 7-1—7-32
 - 使用禁止にする 7-30, 7-31
 - 即時 1-26, 7-6
 - 定義 1-2
 - 伝播担当者 1-21
 - トリガー 7-32
 - バックアップおよび回復 6-14
 - プロシージャ 1-25
 - マスター・グループの再開 3-16
 - マスター・グループの中断 3-15
 - マルチマスター 3-1—3-30
 - アドバンスト・オプション 3-27—3-30
 - 環境の作成

- 例 3-2
 - 監視 3-25—3-27, 6-4—6-13
- 読み込み専用、使用方法 1-4
- レプリケーション・アクティビティの再開 9-121
- レプリケーション・オブジェクト 1-16
 - 定義の変更 3-20
 - マスター・グループから削除 3-21
 - マスター・グループへの追加 3-14, 3-17
- 「レプリケーション・オブジェクトの変更」ダイアログ
 - Replication Manager 3-20
- 「レプリケーション・オブジェクトの編集」プロパティ・シート
 - 「一般」ページ 3-20, 3-24, 4-19
- 通信設定の最小化 3-25
- レプリケーション・カタログ・ビュー
 - 不適切な 6-22
- レプリケーション・グループ
 - マスター・サイト
 - 管理 3-13—3-26
- レプリケーション・サポート
 - オブジェクト用の生成 3-24
- 表の
 - 生成できない 6-21
 - マスター・グループのすべての表に対する生成 3-24
 - マスター・グループ用の生成 3-23
- レプリケーション・サポートの生成
 - 移行および B-8
- レプリケーション・トリガー A-2
- レプリケーション管理 API 1-17, 9-1
- レプリケーション表
 - コメントの更新 6-25
- レプリケーションを使用可能にする 7-30
- レプリケーションを使用禁止にする 7-30
- レプリケート・オブジェクト
 - DROP_MASTER_REPOBJECT 9-94
 - グループ
 - 作成
 - マスター・サイト 9-79
 - 再作成 6-21
 - 削除
 - スナップショット・サイト 9-101
 - 作成

- スナップショット・サイト 9-84
- マスター・サイト 9-80
- サポートの生成 9-105, 9-106, 9-108
- スナップショット・サイト
 - 作成に関する問題 6-23
- 変更 9-64
- レプリケート・オブジェクト・グループ
 - オフライン・インスタンスエーション 9-31, 9-32, 9-33, 9-34, 9-35
- 削除 9-93
- レプリケート・トランザクション
 - 依存性の順序 3-29
- レプリケート環境
 - 表相互間の相違の判別 6-16
- レプリケートされたプロシージャ
 - サポートの生成 7-3
- レプリケート表
 - DML の非互換性 6-21
 - 許可されているデータ型 3-18
 - 手動移入 3-19
 - 相違 6-16
 - 代替キー 3-18
- 連続的な送信
 - スケジュール 3-10, 3-12

ろ

- 「ローカル・エラー」フォルダ 6-10
- ローカル・ジョブ
 - 表示 6-11
 - プロパティの編集 6-12
- ロールバック・セグメント
 - スナップショットの設定 2-27
 - セットアップ・ウィザード 3-11

わ

- ワークフロー 7-26
- 割当て
 - 更新の競合解消方法 5-13

