



レプリケーション超入門

日本オラクル株式会社

¹
ORACLE®



目的

このセミナーは、
レプリケーションのことを知らない人が
簡単なレプリケーション環境が作れるようになるための
基礎知識と手順を説明するものです。



内容

1. レプリケーションとは？利点は？
2. レプリケーションの種類と比較
3. 基本レプリケーションの構築手順
読み専用スナップショットを使ったレプリケーション
4. マルチマスターレプリケーションの構築手順
5. スナップショットサイトレプリケーションの構築手順
更新可能スナップショットを使ったレプリケーション



内容

1. レプリケーションとは？利点は？

2. レプリケーションの種類と比較
3. 基本レプリケーションの構築手順
読み専用スナップショットを使ったレプリケーション
4. マルチマスターレプリケーションの構築手順
5. スナップショットサイトレプリケーションの構築手順
更新可能スナップショットを使ったレプリケーション

【1. レプリケーションとは？利点は？】



分散システムとは？

- レプリケーションはOracleの分散ソリューションのひとつ
- 集中型 vs 分散型
 - 集中型システム
 - システムの一元管理が可能
 - 障害によってはシステム全体がダウンする可能性あり
 - 分散型システム
 - システム毎に役割を持たせる
 - 1マシンがダウンしても全体としては稼働可能
 - 複数のマシンで処理を分散できる

5
ORACLE®

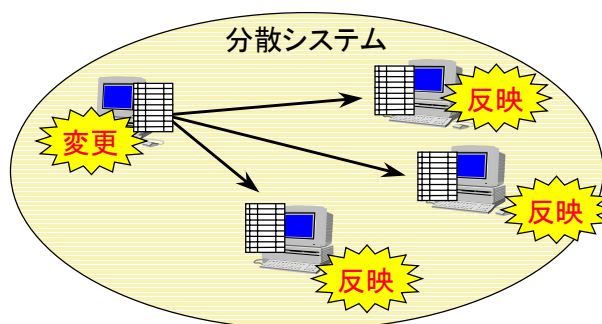
「分散システム」とは、データ処理及びトランザクション処理がネットワーク上の1つ又は複数のコンピュータの間に分配され、各コンピュータがシステム内で特定の役割を果たしているようなシステムです。

【1. レプリケーションとは？利点は？】



レプリケーションとは？

- Replication → 複製(Replica)を作ること
 - オブジェクトのコピーを作って分散させる。
 - システムのどこかで変更が行われると、分散システム内の全てのサイトのデータにその変更が反映される。



6
ORACLE®

レプリケーションとは？

簡単に言うと、レプリケーションとはデータベース・オブジェクトのコピーを作ることです。

【1. レプリケーションとは？利点は？】



レプリケーションの利点は？

- **なぜコピーするのか？**
 - 代替データにアクセス
 - 利点① パフォーマンスの向上
 - 利点② 可用性の向上
 - 利点③ データの切り出し
 - 利点④ データマート

⁷
ORACLE®

何のためにコピーを作るのか？

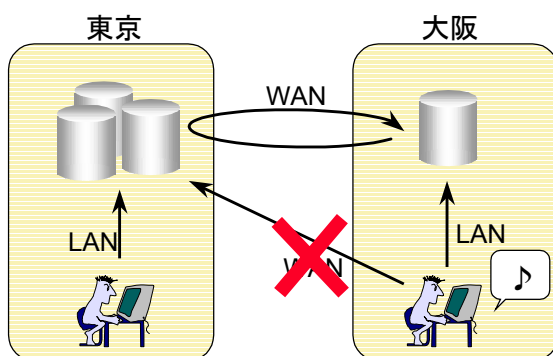
コピーを作成することで得られるメリットは4つあります。

【1. レプリケーションとは？利点は？】



利点①：パフォーマンスの向上

- システムの負荷分散
- ローカルデータへの高速アクセス
- ネットワークトラフィックの減少



8
ORACLE®

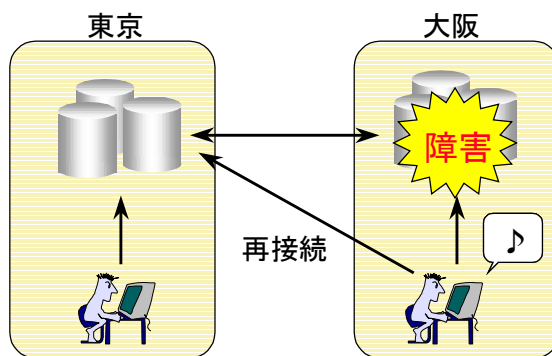
1つめは**パフォーマンス**です。リモートデータのコピーをローカルに置けば、ネットワーク・トラフィックは最小限に抑えられ、応答時間も速くなります。また大きなデータの必要な部分だけコピーをしておくことでもパフォーマンス向上、負荷分散に役立ちます。

【1. レプリケーションとは？利点は？】



利点②：可用性の向上

- ノード障害が発生しても業務が止まらない



9
ORACLE®

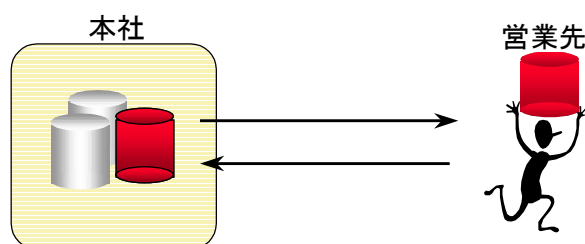
2つめは**可用性**です。マスターを複数持つことができるマルチマスター構成では、ノード傷害が発生してもアプリケーションを自動的に他のノードのマスターデータに再接続してくれるので、処理を継続することができます。

【1. レプリケーションとは？利点は？】

Oracle
8

利点③：データの切り出し

- 営業担当者はノートPCで必要なデータを持っていくことができる。
- マスターデータへの反映、最新データの取得も容易



10
ORACLE®

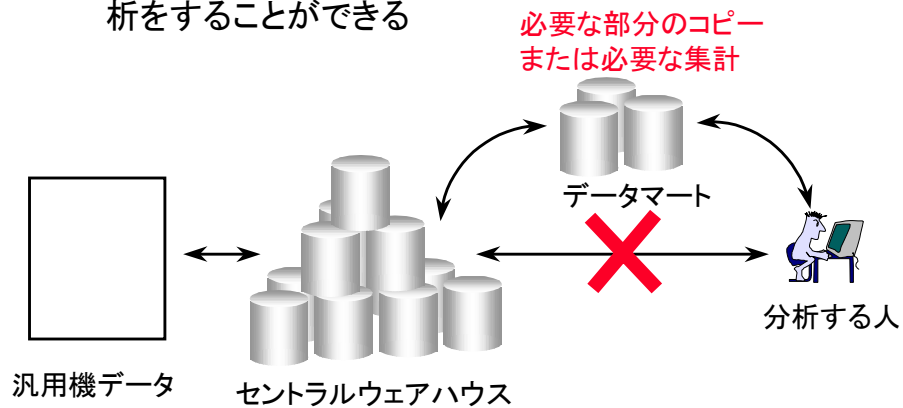
3つめは**データの切り出し**です。営業先などに必要なデータのコピーを持ち出すことが可能です。共にマスターへの反映や最新データの取得を容易に行うことができます。

【1. レプリケーションとは？利点は？】



利点④：データマート

- 膨大なデータから必要な集計を持つことができる
- ユーザはデータマートにアクセスしてオンライン分析をすることができる



11
ORACLE®

4つめはデータマートです。膨大なデータから必要な集計を持つことが可能です。また、ユーザはデータマートにアクセスし素早く分析を行えます。



内容

1. レプリケーションとは？利点は？

2. レプリケーションの種類と比較

3. 基本レプリケーションの構築手順

読み込み専用スナップショットを使ったレプリケーション

4. マルチマスターレプリケーションの構築手順

5. スナップショットサイトレプリケーションの構築手順

更新可能スナップショットを使ったレプリケーション

【2. レプリケーションの種類と比較】



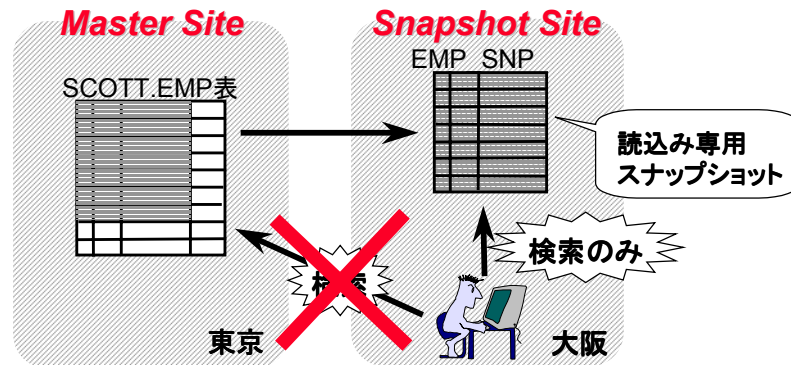
レプリケーションの種類

- 基本レプリケーション
 - 種類①: 読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
 - 種類②: マルチマスター・レプリケーション
 - 種類③: 更新可能スナップショット
 - ハイブリッド構成
- レプリケーション機能の選択

Oracle8のレプリケーション機能は上記の様に分類されます。

レプリケーション環境を構築する際には、どのレベルのレプリケーション機能が必要とされているか判断する必要があります。

【2. レプリケーションの種類と比較】 種類①：読み専用スナップショット



リモートへのアクセスがなくなります。

リモート表への検索を行う必要がある場合、主に2つの問題点があります。

- ・パフォーマンス：リモートへのアクセスはローカルへのアクセスに比べて遅い。又、リモートへの接続が多くなると、ネットワークトラフィックが増加する。
- ・ネットワークの可用性：ネットワーク障害が起こる可能性がある。

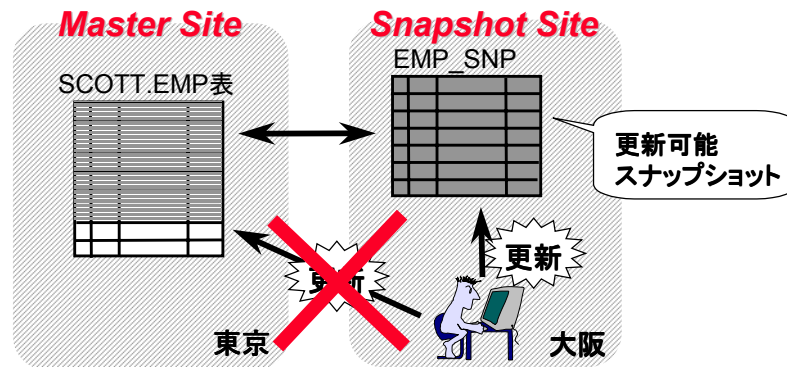
そこで、使用されるのが**読み専用スナップショット**です。

スナップショットとは、マスター表を反映する表のコピー、またはサブセットのことです。スナップショットによって、ローカルのコピーデータを参照できるのでリモートデータにアクセスするよりも高速にアクセスすることができます。また、ネットワーク障害などが原因で、マスターサイトが使用できなくなった場合でも、ローカルのコピーデータを使用して、問合せ処理を続けることができます。

スナップショットはリモートのマスター表への問合せで定義されます。

```
CREATE SNAPSHOT xxx AS SELECT * FROM xxx@xxx;
```

種類②：更新可能スナップショット



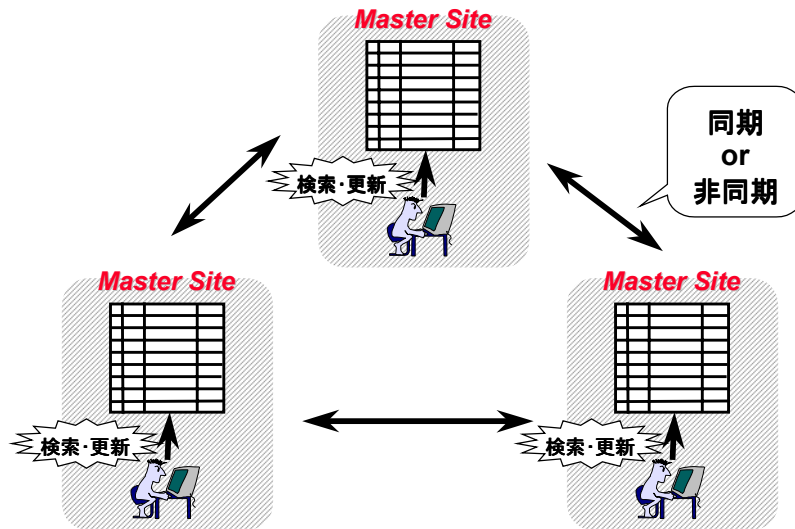
読み専用スナップショットは検索のみが可能でしたが、それを更新可能にしたのが、**更新可能スナップショット**です。読み専用スナップショットと同様、マスターサイトにアクセスできなくなっても、処理を続行することができます。

更新可能スナップショットもマスター表への問合せで定義されます。

```
CREATE SNAPSHOT xxx FOR UPDATE AS SELECT * FROM xxx@xxx;
```

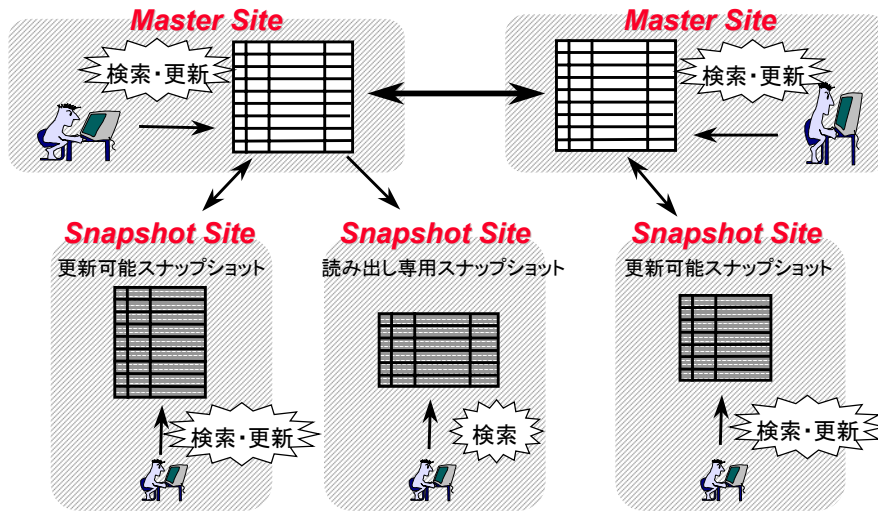
【2. レプリケーションの種類と比較】

種類③：マルチマスター・レプリケーション



マルチマスター・レプリケーションでは表の内容全体を、指定した全てのサイトでレプリケートします。マスター表のどれかに適用した変更が伝播されて、他のサイトの全てのマスター表に反映されます。

【2. レプリケーションの種類と比較】
ハイブリッド構成



以上で述べた読み専用スナップショット、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を組み合わせる構成することが可能です。
これによって、各レプリケーション機能の利点をうまく生かすことができます。

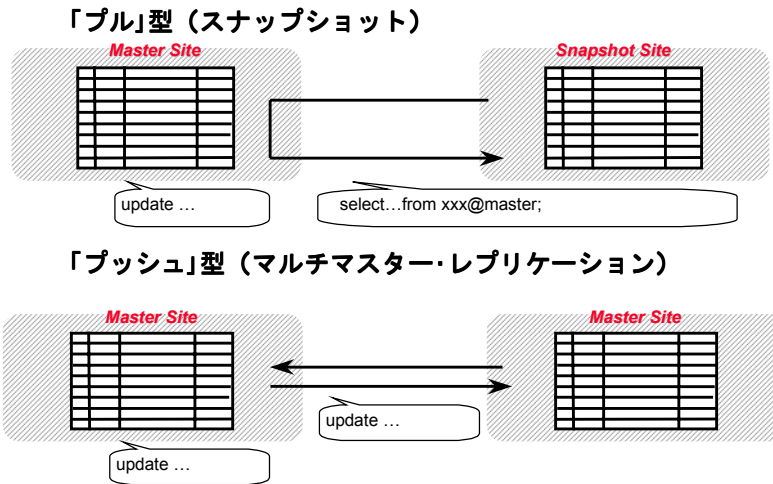
【2. レプリケーションの種類と比較】

比較①：非同期レプリケーションと同期レプリケーション



	非同期 レプリケーション	同期 レプリケーション
あるサイトがダウンした場合に他のサイトでは？	トランザクション 実行可能	トランザクション 実行不可能
競合の発生する可能性	ある	ない
マスター・サイト間でのデータの整合性	一時的な不整合	完全なデータの 整合性を保証
パフォーマンス	良い	あまり良くない (2フェーズコミットを 使用するため)

【2. レプリケーションの種類と比較】
比較②：スナップショットとマルチマスター



マルチマスター・レプリケーションでは表の内容全体を、指定した全てのサイトでレプリケートします。マスター表のどれかに適用した変更が伝播されて、他のサイトの全てのマスター表に反映されます。

スナップショットでは、変更が「プル」の形で行われるのに対し、マルチマスター・レプリケーションでは「プッシュ」型で変更が行われます。

【2. レプリケーションの種類と比較】
比較③：レプリケーションの利点による比較

	マルチマスター レプリケーション	更新可能 スナップショット	読み込み専用 スナップショット
パフォーマンス	○	○	○ 更新:△
可用性	◎ 非同期のみ	○ 非同期のみ	○ 更新:△
持ち運び	—	◎ 非同期のみ	○ 更新:△
データマート	—	—	◎

読取専用スナップショットは「非同期」のみ

レプリケーション環境を構築するためには、サイトをどのレプリケーション形態にするのかを判断する必要があります。

その為の指針を3つ紹介します。

1. レプリケーション機能の利点

レプリケーションによる利点から実装に当てはめて考えてみます。

【2. レプリケーションの種類と比較】

比較③：レプリケーションの利点による比較



	マルチマスター	更新可能	読み専用
パフォーマンス	○	○	○ 更新: △

- 全てローカルにデータを持ち、ネットワーク上をデータが流れないため、どれもパフォーマンス向上する。
- 読み専用スナップショットが更新する場合は、リモートのマスターサイトに更新に行くのでパフォーマンスは良くない

①パフォーマンス

全てローカルにデータを持つことになるので、パフォーマンスでは差がありません。読み専用スナップショットが更新しに行く場合は、リモートのマスターサイトに更新に行くのでパフォーマンスはでなくなります。

【2. レプリケーションの種類と比較】

比較③：レプリケーションの利点による比較



	マルチマスター	更新可能	読み専用
可用性	◎ 非同期のみ	○ 非同期のみ	○ 更新:△

- マルチマスターは、他のサイトから復旧可能なので、可用性が高い。
- スナップショットは、マスターの復旧は不可能だが、一時的な業務の継続を保証する。

②可用性

マルチマスター・レプリケーションであれば、1つのマスターサイトに障害が起きても、他のサイトから復旧可能なので可用性は高くなります。そのため、マスターサイトは複数位置すべきです。

スナップショットは、システムの完全な復旧はできませんが、マスターサイトが回復するまでの業務の一時的な継続を保証します。但し、読み専用スナップショットの場合は、完全に更新が不可能となります。

【2. レプリケーションの種類と比較】

比較③：レプリケーションの利点による比較

	マルチマスター	更新可能	読み専用
持ち運び	—	◎ 非同期のみ	○ 更新:△

- スナップショットは、一時的にマスター表からデータを持ち出すことが可能である。
- 更新可能スナップショットは、持ち運び先で変更したデータを、その後マスターに反映することができる。
- マルチマスターは、論理的には持ち運び可能だが、一般的に利用しない。

③持ち運び

スナップショットは一時的にマスター表からデータを持ち出すことが可能です。更新可能スナップショットの場合は、持ち運び先でデータを更新し、その後データをマスターに反映させることができます。

【2. レプリケーションの種類と比較】

比較③：レプリケーションの利点による比較



	マルチマスター	更新可能	読み専用
データマート	—	—	◎

- 更新はしないでデータのサブセットや集計を持つという意味のデータマートでは、読み専用スナップショットを使用することが考えられる。

④データマート

更新はしないでデータのサブセットや集計を持つという意味のデータマートでは、読み専用スナップショットを使用することが考えられます。

比較④：レプリケーション範囲による比較

表全体をレプリケート？サブセットをレプリケート？

	読み専用 スナップショット	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
垂直分割 (列単位)	可	不可	不可
水平分割 (行単位)	可	可	不可

表全体のみ

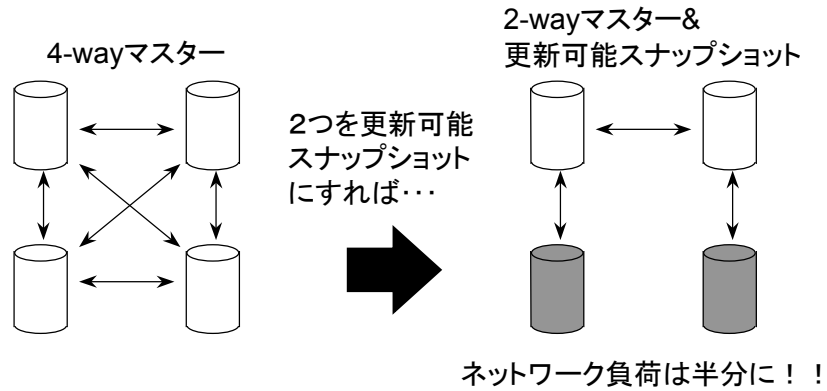
2. レプリケートデータは表全体か、表のサブセットか

レプリケートしたい表のうち、どの部分のデータを使いたいかによって判断することもできます。

マルチマスター・レプリケーションは表全体をレプリケートする必要がありますが、更新可能スナップショットは表のサブセットのみをレプリケートすることができます。よって、サイト毎に必要なデータのみをレプリケートしたい場合は、更新可能スナップショットを使用したデータの水平分散を選択します。これにより使用ディスク量を節約できます。読み専用スナップショットは垂直分散、水平分散をサポートします。

【2. レプリケーションの種類と比較】

比較⑤：伝播のネットワーク負荷



マスターの数が増えれば増えるほどネットワーク負荷が増大する。

マスターの数は必要最小限にする!!

3.伝播のネットワーク負荷

更新可能スナップショットでは、バッチ処理の様に変更をまとめて行いますが、マルチマスターレプリケーションはトランザクション毎に変更が全てのマスターサイトにレプリケートされます。また、マスター同士が変更を伝播しあうために、マスターが1つ増えるたびにネットワーク負荷が増大していきます。

- 2-wayマスター:1
- 3-wayマスター:3
- 4-wayマスター:6
- 5-wayマスター:10

⋮

そのため、マスターサイトは必要最小限にした方が良いでしょう。



内容

1. レプリケーションとは？利点は？
2. レプリケーションの種類と比較

3. 基本レプリケーションの構築手順

読み込み専用スナップショットを使ったレプリケーション

4. マルチマスターレプリケーションの構築手順
5. スナップショットサイトレプリケーションの構築手順
更新可能スナップショットを使ったレプリケーション

【3. 基本レプリケーションの構築手順】

基本レプリケーションの前提知識



- 初期化パラメータ
- 主キー・スナップショット
- リフレッシュ
 - 完全リフレッシュと高速リフレッシュ
 - リフレッシュのタイミング
- 単純スナップショットと複合スナップショット
- リフレッシュ・グループ



【3. 基本レプリケーションの構築手順- 前提知識】
初期化パラメータの設定

- GLOBAL_NAMES = TRUE
- SHARED_POOL_SIZE: 最低15～20M
- SNPプロセスの調整
 - JOB_QUEUE_PROCESSES \geq 1
 - JOB_QUEUE_INTERVAL etc



主キースナップショット

- マスター表の行とスナップショットの行を対応づける必要がある。
 - Oracle7 : ROWIDスナップショット
 - マスター表の再編成が使用できない (SQL*Loader, Importユーティリティなど)
 - Oracle8 : **主キースナップショット** (デフォルト)
- マスター表にPrimary Keyがついている必要がある。

主キースナップショット

スナップショットでは、マスター表とローカルスナップショットの行と行を対応づけてデータの整合性を保証する必要があります。

Oracle7では、ROWIDを元にしてデータ整合性を保証していました。このため、ROWIDを元にするSQL*LoaderやImportユーティリティを再編成のために使用することができませんでした。

再編成とは、

- ① Unload ② Truncate table ③ SQL*loader
- ① Export ② Truncate table ④ Import

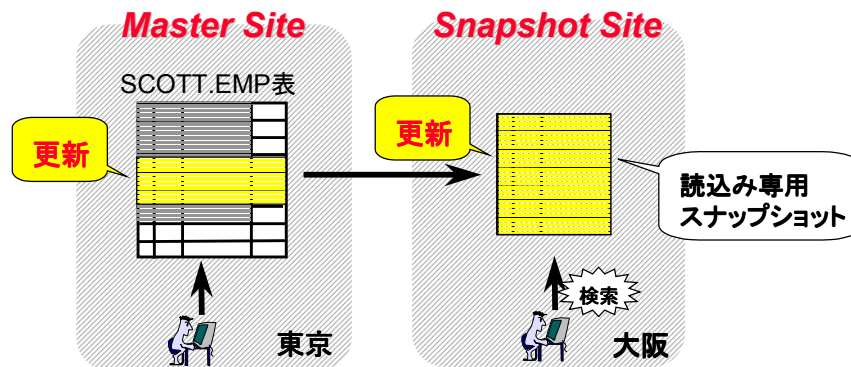
そこで、Oracle8からは、主キーで整合性を保証する**主キー・スナップショット**がデフォルトとして作成されるようになりました。ROWIDスナップショットも作成できますが、Oracle7との互換性のためだけに存在しています。

そのため、マスター表に必ずPrimary Key制約が存在する必要があります。

リフレッシュの方法(1/2) : 完全リフレッシュ

スナップショットを定期的にはリフレッシュしなければなりません。

- 完全リフレッシュ



マスター表が更新されれば、スナップショットも定期的にはリフレッシュする必要があります。
読み込み専用スナップショットは、**手動**または**自動**でリフレッシュすることができます。
また、リフレッシュの種類も「**完全リフレッシュ**」と「**高速リフレッシュ**」の2種類が用意されています。

完全リフレッシュ

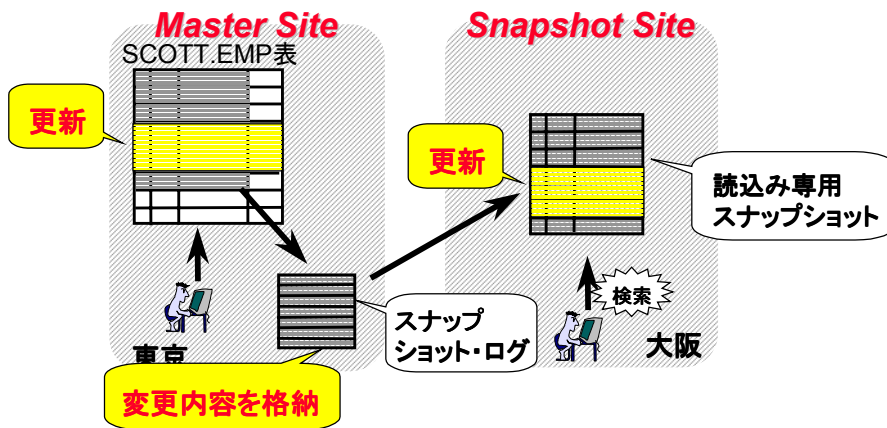
スナップショット全体をマスター表からコピーし直します。

リフレッシュの方法(2/2) : 高速リフレッシュ



スナップショットを定期的にはリフレッシュしなければなりません。

- 高速リフレッシュ(単純スナップショットの場合のみ)



高速リフレッシュ

最後にリフレッシュされた時点から、マスターに加えられた変更だけ差分リフレッシュします。高速リフレッシュはスナップショット・ログというマスター表への変更を格納するための表を作成する必要があります。ただし、高速リフレッシュが行われるのは単純スナップショットと呼ばれる形態のものだけです。

【3. 基本レプリケーションの構築手順- 前提知識】



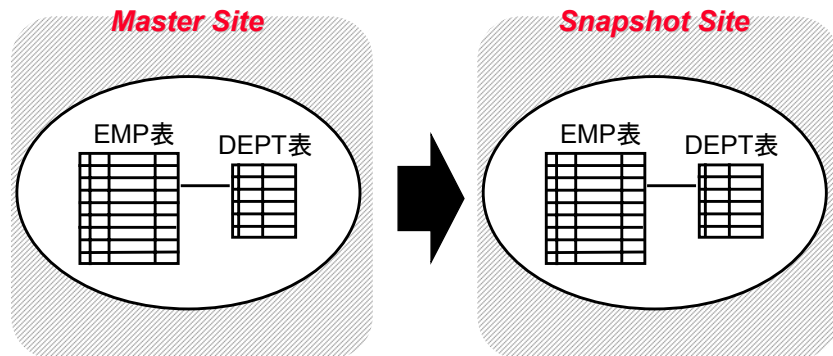
単純スナップショットと複合スナップショット

- 単純スナップショット
- 複合スナップショット
 - 違い: 単純スナップショットに以下のものは含まれません。
 - GROUP BY句
 - CONNECT BY句
 - 固有の関数又は集約関数
 - 結合(使用可能なタイプの副問合せを除く)
 - 集合演算子

複合スナップショットは高速リフレッシュできません。

リフレッシュ・グループ

- EMP表とDEPT表は同時にレプリケートしたい



レプリケーショングループ内に関連するレプリケーションオブジェクトを編成することで、複数のオブジェクトをまとめて管理できます。複数のマスター表の間で、参照整合性、トランザクション一貫性を保つために、レプリケートグループを作成します。但し、1つのオブジェクトは1つのレプリケートグループにしか存在できません。

基本レプリケーション環境の作成手順



① 前提条件

- @マスター表にPrimary Keyがついていること
- * マスターサイトへのデータベースリンクがあること

* ② (高速リフレッシュの場合)

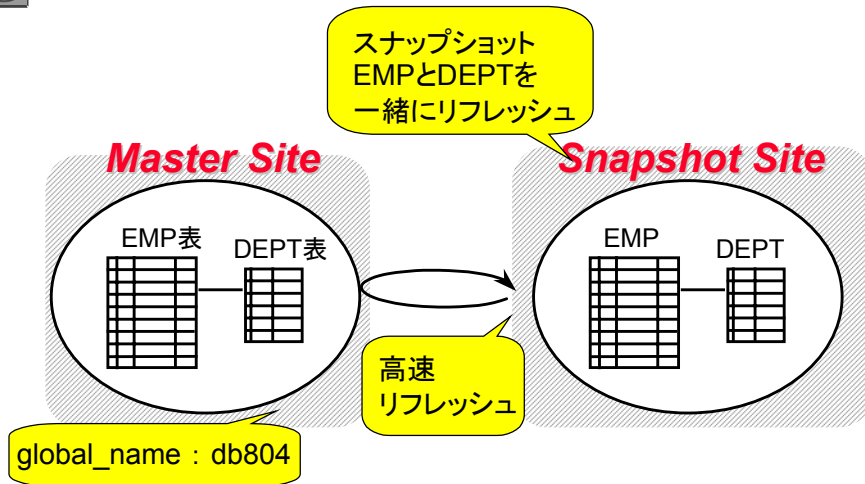
スナップショットログを作成する。

@③ スナップショットを作成する。

@④ (必要なら)スナップショットサイト・リフレッシュ・グループを作成する。

* 青字の部分はマスターサイトでのみ実行します。
@赤字の部分はスナップショットサイトでのみ実行します。

【3. 基本レプリケーションの構築手順】
例：EMP表とDEPT表の読み専用スナップショット



このようなスナップショットを作ってみましょう。

【3. 基本レプリケーションの構築手順】

例：EMP表とDEPT表の読み専用スナップショットを作る



。

① 前提条件

- マスター表にPrimary Keyがついていること
- マスターサイトへのデータベースリンクがあること
 - データベースリンクの名前はグローバル・ネームと同じでなければならない。

```
select * from global_name;(マスターサイトで)  
alter database rename global_name to xxx;
```

② マスター表に対してスナップショットログを作成する。

```
CREATE SNAPSHOT LOG ON emp;  
CREATE SNAPSHOT LOG ON dept;
```

*このフェーズはマスターサイトで実行します。



【3. 基本レプリケーションの構築手順】

例：EMP表とDEPT表の読込み専用スナップショット
を作る。

③ スナップショットを作成する。

```
CREATE SNAPSHOT emp  
REFRESH FAST  
AS SELECT * from scott.emp@db804;
```

リフレッシュ方法の指定
[FAST | COMPLETE | FORCE]

自動リフレッシュタイミングの指定もできるが、
リフレッシュグループで指定するので指定しない。

スナップショットの定義問合せ

*同様にDEPTもスナップショットを作成する。

@このフェーズはスナップショットサイトで実行します。

38
ORACLE®

START WITH、NEXT両方とも指定しなかった場合は、スナップショットは自動的にリフレッシュされません。

SNAPSHOT_REFRESH_PROCESSES

SNAPSHOT_REFRESH_INTERVAL

SNAPSHOT_REFRESH_KEEP_CONNECTIONS

【3. 基本レプリケーションの構築手順】

例：EMP表とDEPT表の読み込み専用スナップショットを作る



④スナップショットサイト・リフレッシュ・グループを作成する。

```
execute DBMS_REFRESH.MAKE(  
NAME      => 'scott.refgrp1',-  
LIST      => 'scott.dept,scott.emp',-  
NEXT_DATE => SYSDATE,-  
INTERVAL  => 'SYSDATE+1/24');
```

リフレッシュグループの名前

リフレッシュ対象のスナップショット

リフレッシュ
タイミングの指定

@このフェーズはスナップショットサイトで実行します。



内容

1. レプリケーションとは？利点は？
2. レプリケーションの種類と比較
3. 基本レプリケーションの構築手順
読み込み専用スナップショットを使ったレプリケーション
- 4. マルチマスターレプリケーションの構築手順**
5. スナップショットサイトレプリケーションの構築手順
更新可能スナップショットを使ったレプリケーション

【4. マルチマスターレプリケーションの構築手順】

マルチマスターレプリケーションの前提知識



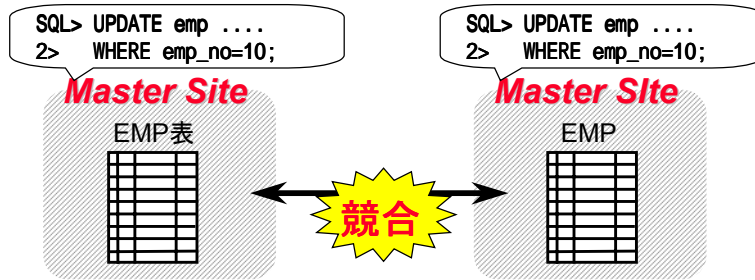
- 競合
- レプリケートサイト
- マスターサイトの構造
- マルチマスターレプリケーションの仕組み

【4. マルチマスターレプリケーションの構築手順- 前提知識】

競合

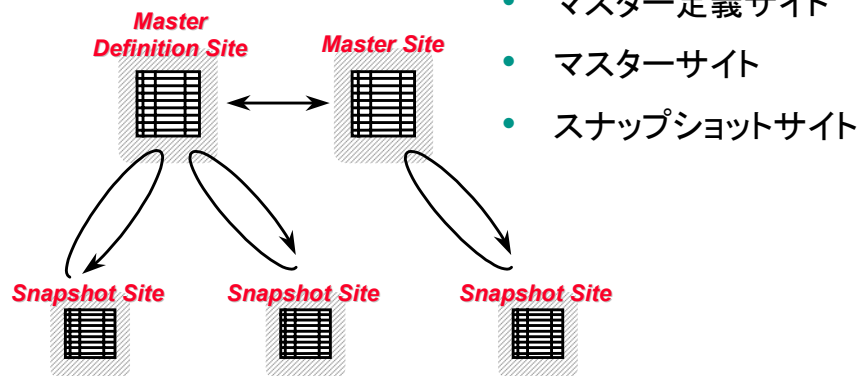


各マスターで同じ行を更新



競合回避ルーチンを設定する必要がある。
(最新のタイムスタンプ、最大値、平均、上書き etc)

レプリケートサイト



- マスター定義サイト
- マスターサイト
- スナップショットサイト

レプリケート環境は、1つのマスター定義サイトと、0個以上のマスターサイト、各マスターサイトごとに0個以上のスナップショットサイトで構成されます。

マスター定義サイト

マスター定義サイトは、管理作業を行うためのサイトです。データレベルの変更はどのサイトでも行えますが、スキーマレベルの変更はマスター定義サイトで行わなくてはなりません。

マスターサイト

マスター定義サイト以外のマスターサイトは、マスター定義サイトから作成します。

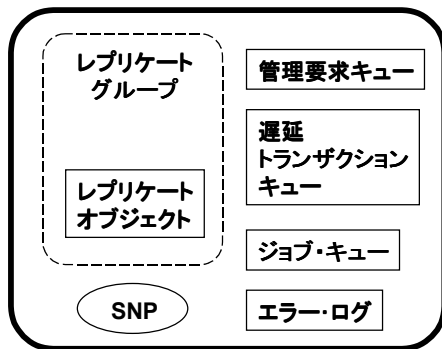
スナップショットサイト

スナップショットサイトは、任意のマスターサイトから作成できます。

マスターサイトの構造



マスター定義サイト、マスターサイトの構造



管理要求キュー(RepCatLogビュー)

非同期の管理要求、エラーを格納するビュー

遅延トランザクションキュー

ローカルで行われた変更が一時格納されるキュー

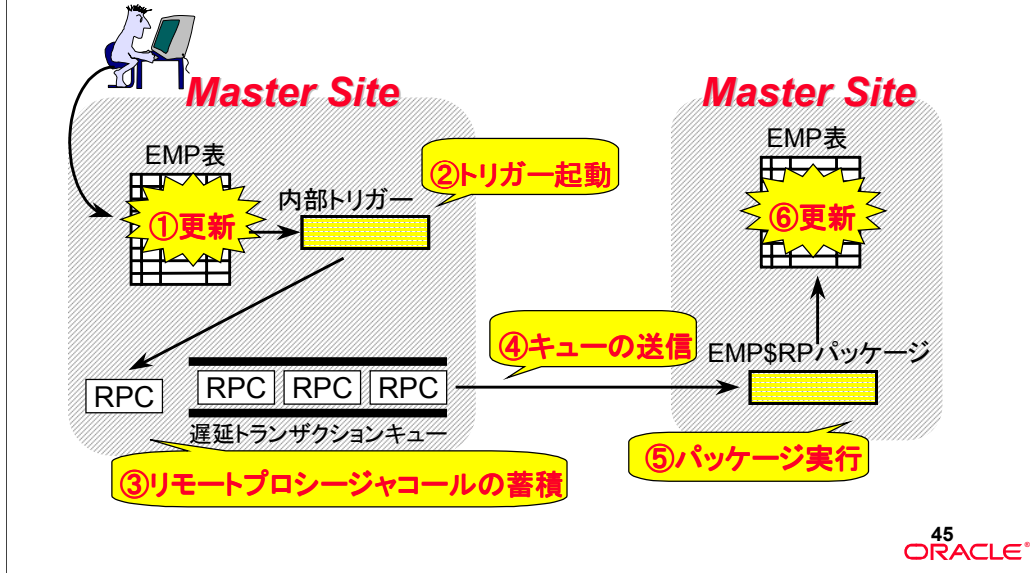
ジョブキュー

遅延トランザクションキューに格納された変更を定期的によりモートサイトに適応するジョブのためのキュー

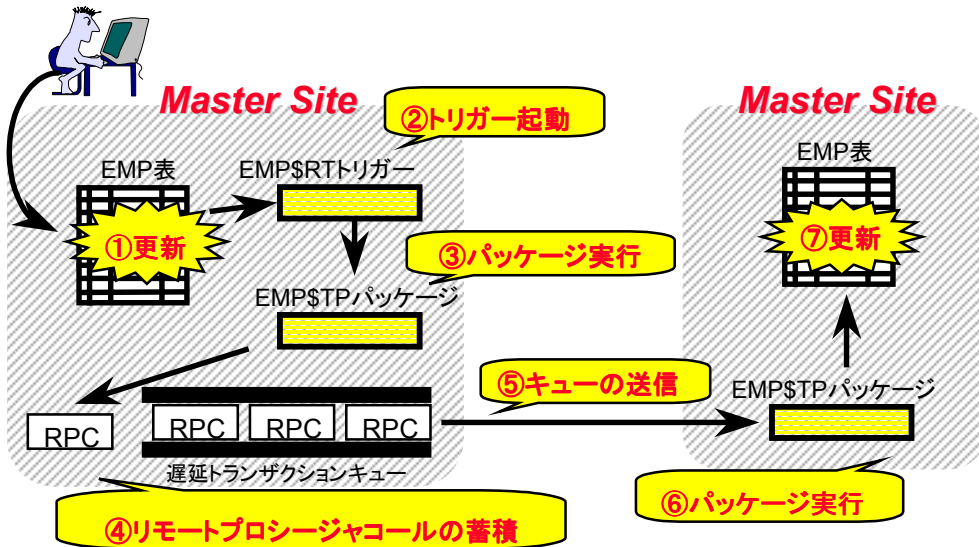
【4. マルチマスターレプリケーションの構築手順 - 前提知識】

マルチマスター・レプリケーションの仕組み

Oracle
8



上図はOracle8の場合です。Oracle7の場合は下図のようになります。



【4. マルチマスターレプリケーションの構築手順】

マルチマスターレプリケーション環境の 作成手順



- ① ユーザ作成・権限の付与・DBリンクの作成
- ② 自動伝播のデフォルトスケジュールを設定する。
 - * ③ レプリケートグループを作成する。
 - * ④ レプリケートグループにレプリケートオブジェクトを追加する。
 - * ⑤ (競合解消ルーチンが必要な場合)競合解消ルーチンを設定する。
 - * ⑥ 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等を作成する。
 - * ⑦ レプリケート環境に新しいマスターサイトを追加する。
 - * ⑧ レプリケーションの動作を開始する。

* 青字の部分はマスター定義サイトでのみ実行します。



【4. マルチマスターレプリケーションの構築手順】

① ユーザ作成・権限の付与・DBリンクの作成

- レプリケーション環境の管理者 兼 伝播担当者 兼 受信者であるREPADMINを作成し、必要な権限を与える。(各マスターサイトに一人)
 - CREATE USER repadmin...
 - execute dbms_repcat_admin.grant_admin_any_schema(-
username => 'REPADMIN');
 - execute dbms_defer_sys.register_propagator(-
username => 'REPADMIN');
 - grant comment any table to REPADMIN;
 - grant lock any table to REPADMIN;
 - grant execute any procedure to REPADMIN;
- レプリケーションオブジェクトを持つユーザを各マスターサイトに作成する。
 - CREATE USER repuser ...

【4. マルチマスターレプリケーションの構築手順】

① ユーザ作成・権限の付与・DBリンクの作成

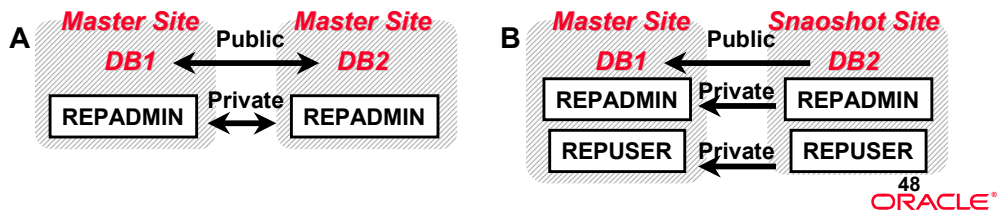


A. マルチマスターレプリケーションの場合 * 各マスターサイトで実行します。

- マスターサイト間でお互いにパブリックDBリンクを作る。
 - CREATE PUBLIC DATABASE LINK global_name USING 'db1.world';
- 管理者同士でプライベートDBリンクを作る。
 - CREATE DATABASE LINK global_name
CONNECT TO repadmin IDENTIFIED BY repadmin;

B. スナップショットサイトレプリケーションの場合 * 各スナップショットサイトで実行します。

- マスターサイトへパブリックDBリンクを作る。
- 管理者へプライベートDBリンクを作る。
- ユーザーからプライベートDBリンクを作る。



注)

ユーザー同士のデータベースリンクは、マルチマスターの場合は特に必要ありませんが、更新可能スナップショットの時は作成する必要があります。

【4. マルチマスターレプリケーションの構築手順】



② 自動伝播のデフォルトスケジュールを設定する。

- 自動伝播スケジュールを設定します。

設定方法

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

```
execute dbms_defer_sys.schedule_push( -  
    destination => 'DB1.world',-  
    interval     => 'sysdate + 1/24',-  
    next_date    => sysdate);
```

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

```
execute dbms_defer_sys.schedule_purge( -  
    next_date    => sysdate,-  
    interval     => 'sysdate + 1');
```

確認方法: select * from defschedule;

設定解除API: dbms_defer_sys.unschedule_push
dbms_defer_sys.unschedule_purge

* このフェーズは各マスターサイトで実行します。

49
ORACLE®

Oracle7では、この2つ機能はdbms_defer_sys.schedule_executionというパッケージにまとめられていましたが、Oracle8から、遅延トランザクションキューの中身を送信するためのdbms_defer_sys.schedule_pushと、遅延トランザクションキューの中身を削除するためのdbms_defer_sys.schedule_purgeというプロシージャに分けられました。

なぜかという、purge(削除)は負荷が高く、また送信する度に行う必要がないためです。2つの機能を分けることにより、昼間定期的に送信を行い、夜間にまとめてキューの中身を削除する、といった使い方が可能になりました。



【4. マルチマスターレプリケーションの構築手順】

③ レプリケートグループを作成する。

- マスター定義サイトに空のレプリケートグループを作成する。
- レプリケートグループとは、レプリケートオブジェクトを管理する単位です。

作成方法

```
execute dbms_repcat.create_master_repgroup(-  
      gname      => 'GROUP1');
```

確認方法: select * from dba_repgroup;

削除API: dbms_repcat.drop_master_repgroup

* このフェーズはマスター定義サイトでのみ実行します。

【4. マルチマスターレプリケーションの構築手順】



④ レプリケートオブジェクトを追加する。

- マスター定義サイトのレプリケートグループにレプリケートしたいオブジェクトを追加する。

作成方法

```
execute dbms_repcat.create_master_repobject(-  
  sname      => 'REPUSER1',-  
  oname      => 'EMP',-  
  type       => 'TABLE',-  
  gname      => 'GROUP1');
```

確認方法 : select * from dba_repobject;

削除API : dbms_repcat.drop_master_repobject

* このフェーズはマスター定義サイトでのみ実行します。

【4. マルチマスターレプリケーションの構築手順】



⑥ レプリケーションサポートオブジェクトを作成する。

- レプリケーションをサポートするのに必要なトリガー、パッケージを作成する。

作成方法

```
execute dbms_repcat.generate_replication_support( -  
  sname      => 'REPUSER1',-  
  oname      => 'EMP',-  
  type       => 'TABLE');
```

確認方法 : select * from dba_repobject;

削除API : dbms_repcat.drop_master_repobjectで同時に消える。

* このフェーズはマスター定義サイトでのみ実行します。

【4. マルチマスターレプリケーションの構築手順】



⑦ 新しいマスターサイトを追加する。

- レプリケート環境に新しいマスターサイトを追加する。

作成方法

```
execute dbms_repcat.add_master_database(-  
  gname      => 'GROUP1',-  
  master     => 'db1.world');
```

確認方法 : select * from dba_repsites;

削除API : dbms_repcat.remove_master_databases

各マスターサイト : dbms_repcat.drop_master_repgroup

* このフェーズはマスター定義サイトでのみ実行します。

【4. マルチマスターレプリケーションの構築手順】

⑧ レプリケーションの動作を開始する。



- レプリケート動作を開始する。

作成方法

```
execute dbms_repcat.resume_master_activity( -  
      gname      => 'GROUP1');
```

確認方法 : select id,request,status from dba_repcatlog;

停止API : dbms_repcat.suspend_master_activity

* このフェーズはマスター定義サイトでのみ実行します。



内容

1. レプリケーションとは？利点は？
2. レプリケーションの種類と比較
3. 基本レプリケーションの構築手順
読み込み専用スナップショットを使ったレプリケーション
4. マルチマスターレプリケーションの構築手順
- 5. スナップショットサイトレプリケーションの構築手順**
更新可能スナップショットを使ったレプリケーション

【5. スナップショットサイトレプリケーションの構築手順】
スナップショットサイトレプリケーション
環境の作成手順



- ① ユーザ作成・権限の付与
- * ② レプリケートグループを作成する。
- * ③ レプリケートグループにレプリケートオブジェクトを追加する。
- * ④ (競合解消ルーチンが必要な場合)競合解消ルーチンを設定する。
- * ⑤ 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等を作成する。
- * ⑥ (高速リフレッシュをする場合)スナップショットログを作成する。
- @ ⑦ マスターサイトへのデータベースリンクを作成する。
- @ ⑧ 更新可能スナップショットを作成する。
- NEW @ ⑨ 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る。
- @ ⑩ 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する。
- @ ⑪ リフレッシュ・グループを作成する。

* 青字の部分はマスターサイトでのみ実行します。

@赤字の部分はスナップショットサイトでのみ実行します。ORACLE®

56

【5. スナップショットサイトレプリケーションの構築手順】

⑧ 更新可能スナップショットを作成する。



```
CREATE SNAPSHOT emp
```

リフレッシュ方法の指定
[FAST | COMPLETE | FORCE]

```
REFRESH FAST
```

```
FOR UPDATE
```

更新可能の指定

```
AS SELECT * from scott.emp@db804;
```

スナップショットの定義問合せ

自動リフレッシュタイミングの指定もできるが、リフレッシュグループで指定するので指定しない。

@このフェーズはスナップショットサイトでのみ実行します。

【5. スナップショットサイトレプリケーションの構築手順】



⑨ スナップショットレプリケーショングループの作成

- 空のスナップショットレプリケーショングループを作成する。
- マスターレプリケートグループと同じ名前にする。

作成方法

```
execute dbms_repcat.create_snapshot_repgroup(-  
    gname      => 'GROUP1',  
    master     => 'db1.world');
```

確認方法: select * from dba_repgroup;

削除API: dbms_repcat.drop_snapshot_repgroup

@このフェーズはスナップショットサイトでのみ実行します。

【5. スナップショットサイトレプリケーションの構築手順】



⑩ スナップショットレプリケーションオブジェクトの作成

- スナップショットサイトのスナップショットレプリケートグループにレプリケートしたいオブジェクトを追加する。

作成方法

```
execute dbms_repcat.create_snapshot_repobject(-  
  sname      => 'REPUUSER1',-  
  oname      => 'EMP',-  
  type       => 'TABLE',-  
  gname      => 'GROUP1');
```

確認方法 : select * from dba_repobject;

削除API : dbms_repcat.drop_snapshot_repobject

@このフェーズはスナップショットサイトでのみ実行します。

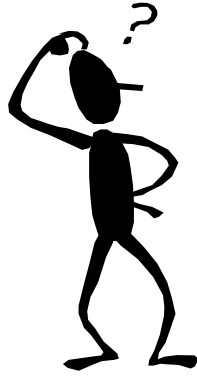


今回のセミナーの内容は・・・

1. レプリケーションとは？利点は？
2. レプリケーションの種類と比較
3. 基本レプリケーションの構築手順
読み込み専用スナップショットを使ったレプリケーション
4. マルチマスターレプリケーションの構築手順
5. スナップショットサイトレプリケーションの構築手順
更新可能スナップショットを使ったレプリケーション



Q & A





この文書はあくまでも参考資料であり、掲載されている情報は予告なしに変更される事があります。この文書に関連して不都合が生じた場合も、米国オラクル社及び日本オラクル株式会社は一切保証せず、特に責任は負いかねますのでご容赦ください。

また許可なく改編、引用する事を禁じます。