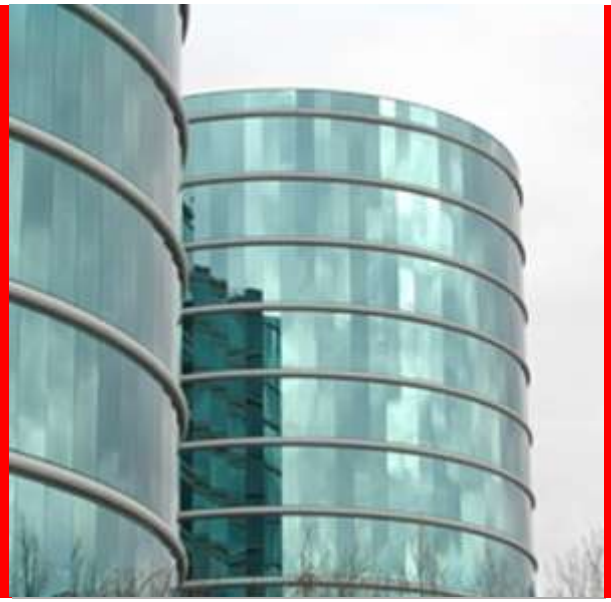


ORACLE

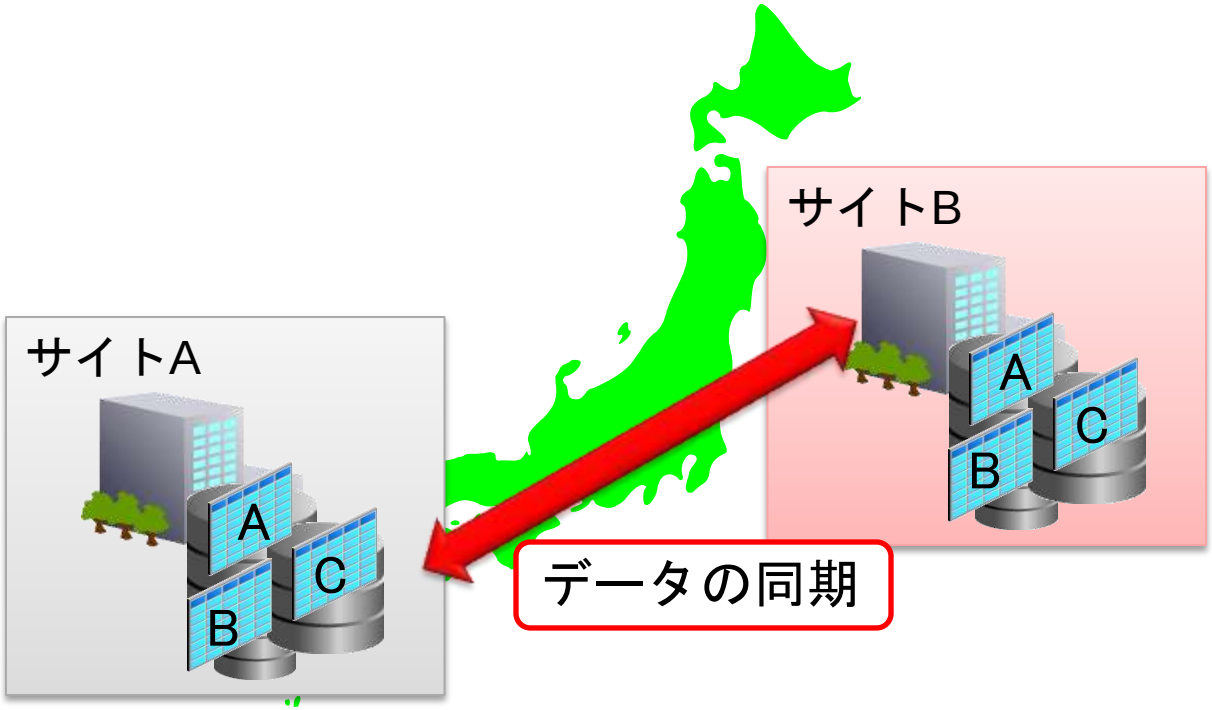
実践!!データベース連携
～レプリケーションの運用管理手引き～

Agenda

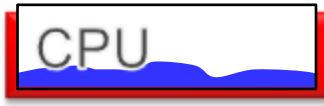
- データ連携ソリューション
- レプリケーションとは
- アドバンスド・レプリケーション環境の構築
- レプリケーション環境の運用管理
  - 競合
  - 障害時のリカバリ
- Appendix

データ連携ソリューションとは？

- 分散DBのテクノロジーを利用して
複数のサイト間でデータを共有する仕組み
  - Replication ：データベースの複製を作成する機能



データ連携利用用途

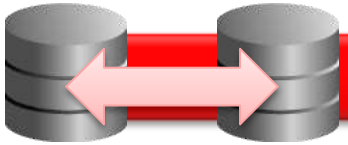


負荷分散

- 複数サイトでの処理の分散化
  - CPU負荷軽減

可用性

- データの2重化
  - 障害時のデータ保護



データ連携

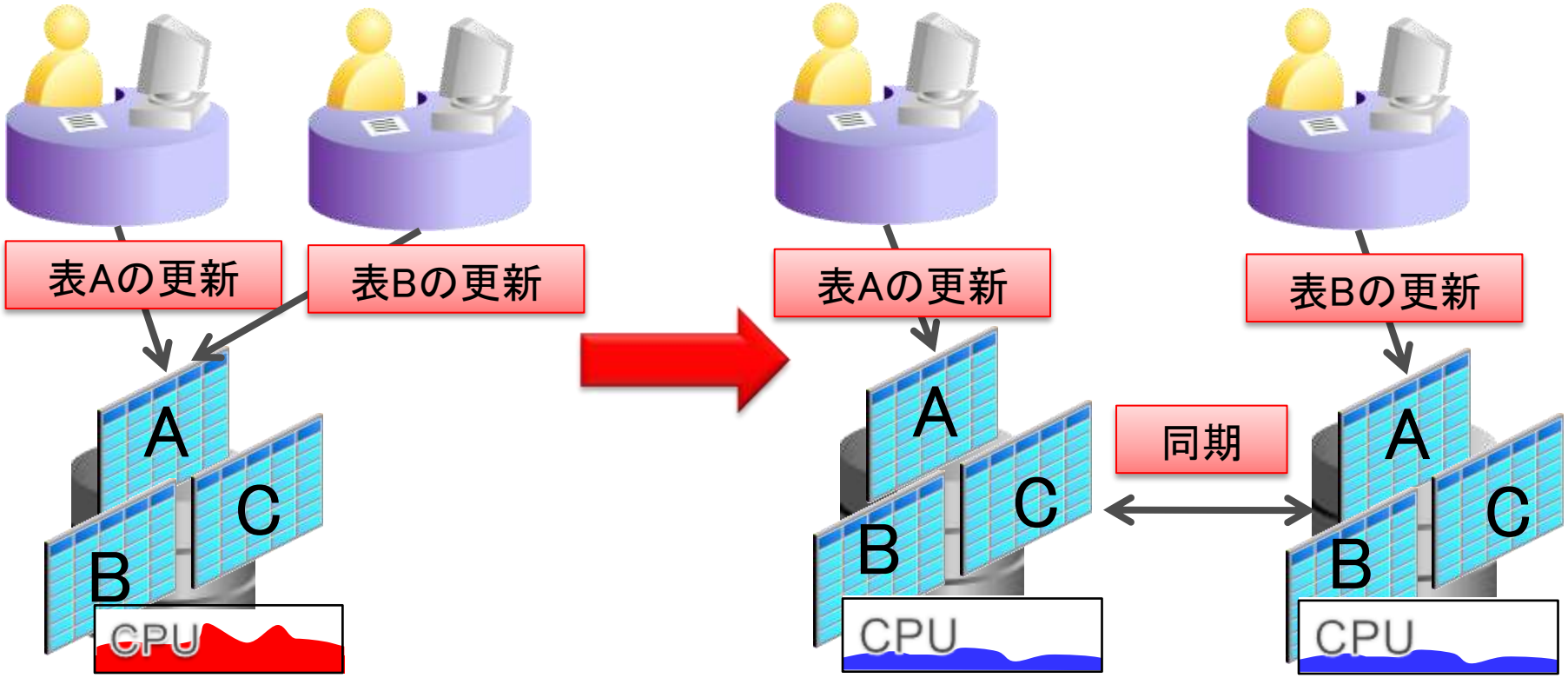
- 異なる環境間でのデータ共有

負荷分散

複数サイトで処理の分散化

CPU 負荷分散

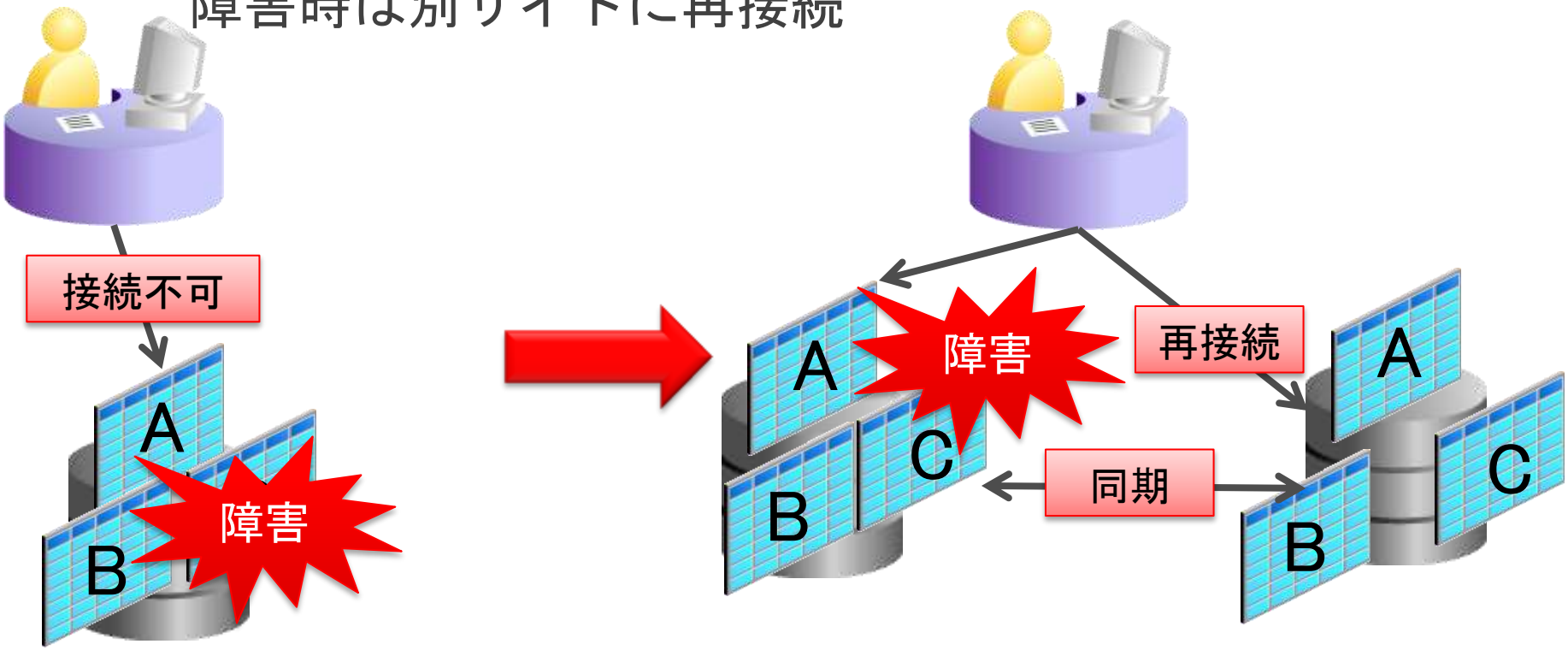
- 負荷が高い処理を別サイトに分散し、処理の負荷を低減



可用性
データの多重化によるデータの保護

可用性

- 別サイトにデータのコピーを作成し、障害時は別サイトに再接続

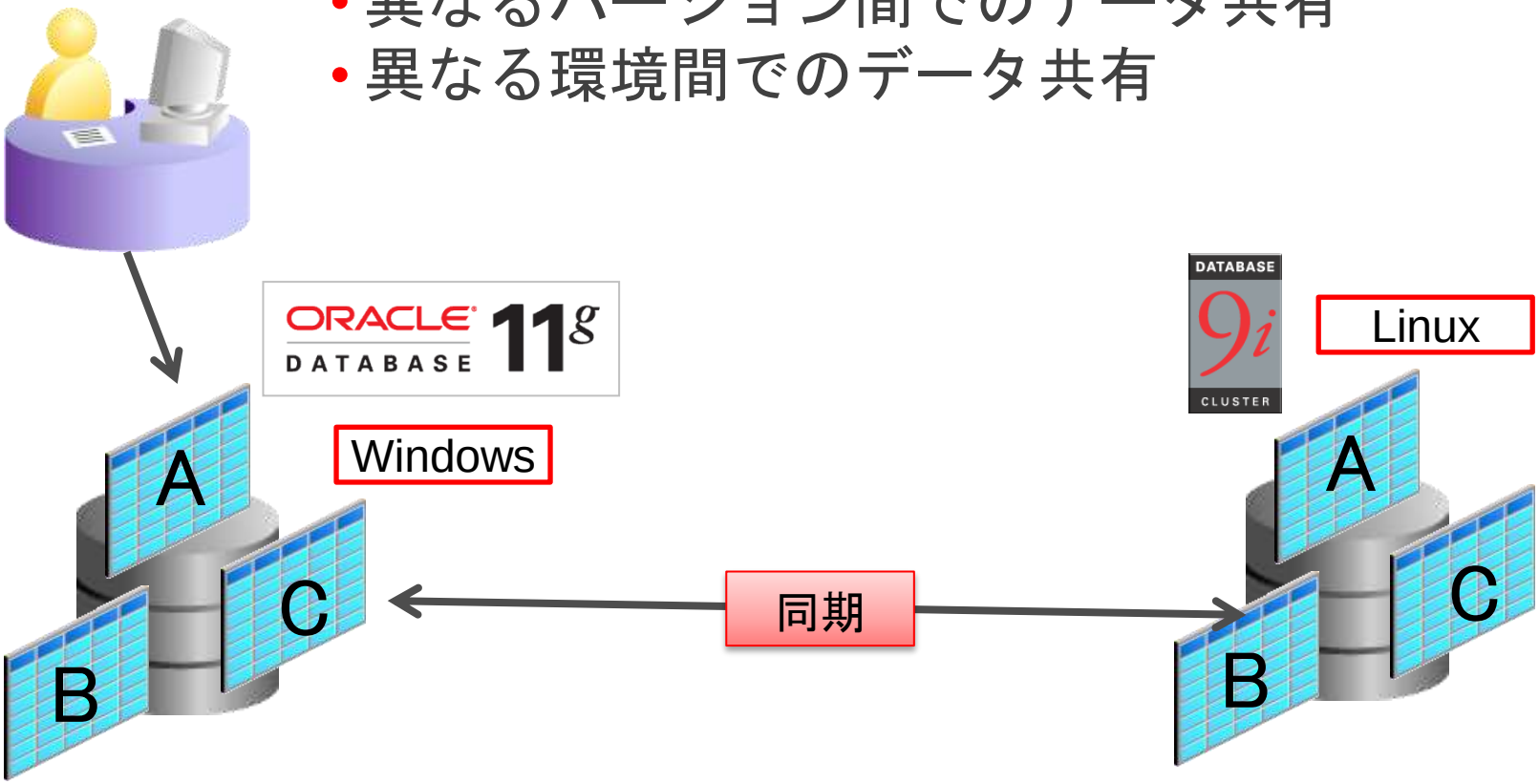


データ連携

異なる環境間でのデータ共有

データ連携

- 異なるバージョン間でのデータ共有
- 異なる環境間でのデータ共有



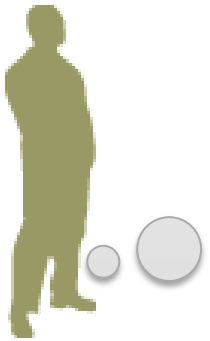
オラクルが提供する連携ソリューション

- オラクルでは、連携基盤構築に際しOracleDatabaseの機能を含む以下のソリューションを提供

本セミナーの範囲
- 基本レプリケーション（概要）
- アドバンスド・レプリケーション

- Oracle DataGuard
- Oracle Data Integrator
- Oracle GoldenGate

連携製品の選定に関しては以下の資料を参照ください
- 実践!!データベース連携製品選定の勘ドコロ
http://www.oracle.com/technetwork/jp/ondemand/db-technique/20100629-integration-kandokoro-dist-255043-ja.pdf



災害対策用に
データの複製を持ちたい

ELT用途として利用したい

負荷分散を行いたい

リアルタイムに同期がとりたい

連携用途・要件によって、利用する機能を選定することが重要

Agenda

- データ連携ソリューション
- レプリケーションとは
- アドバンスド・レプリケーション環境の構築
- レプリケーション環境の運用管理
  - 競合
  - 障害時のリカバリ
- Appendix

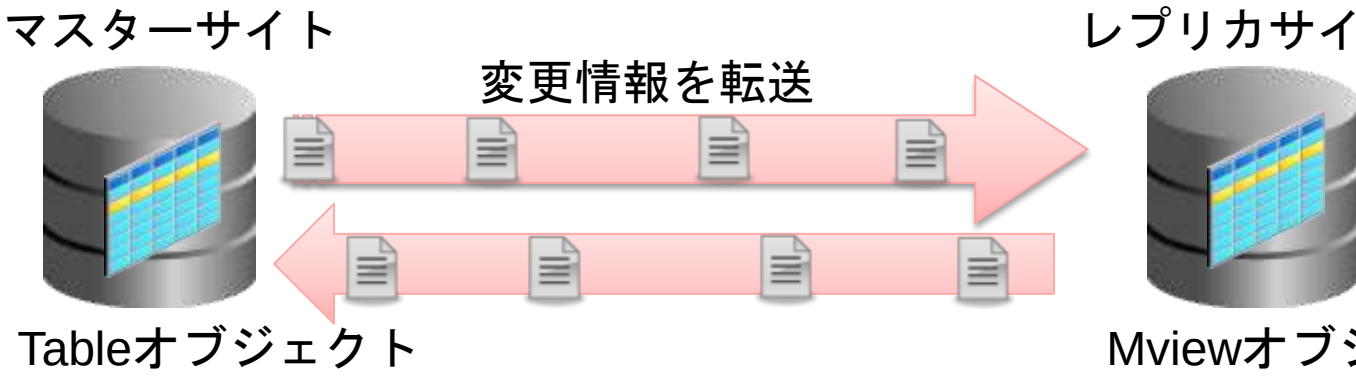
基本レプリケーション

EE

SE

概要

- 基本レプリケーションでは各サイトで対象表の
  マテリアライズド・ビューオブジェクト(MView)を持つ



- 2種類の基本レプリケーション
  - 読み取り専用MView
    - マスターサイトのデータの複製は参照のみ
  - 更新可能MView
    - レプリカ・サイトのマテリアライズド・ビューへの更新が行え、
      その更新内容はマスターに反映されます

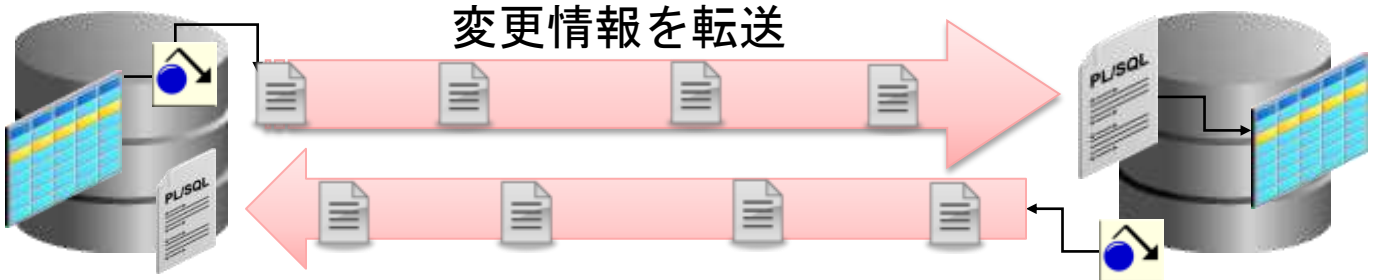
基本レプリケーションの詳細に関しては以下の資料を参照ください
・今さら聞けない!?レプリケーション
http://www.oracle.com/technetwork/jp/ondemand/db-basic/20100922-replication-244733-ja.pdf

アドバンスド・レプリケーション

EE

概要

- アドバンスト・レプリケーション（マルチマスター・レプリケーション）では各サイトで対象表の完全なレプリカを持つ



マスター定義サイト

レプリケーション・グループを定義しているサイト
レプリケーションの管理作業やメンテナンスを実施

マスターサイト

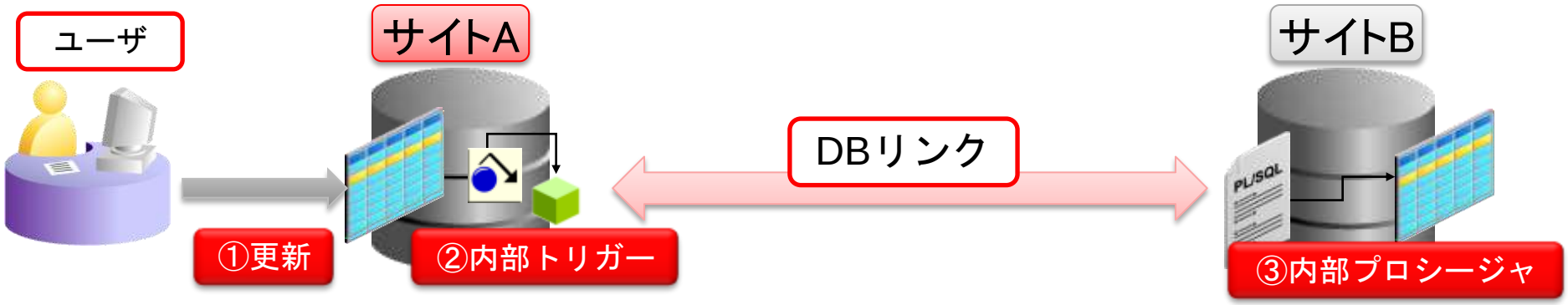
マルチマスター・レプリケーション
環境のノード

- 2種類のデータ同期方法
  - リアルタイム・データレプリケーション（同期）
    - ローカルの変更が即時にリモートサイトに反映
  - 非同期データレプリケーション
    - ローカルの変更は任意の間隔でリモートサイトに伝播・反映

アドバンスド・レプリケーション

アーキテクチャ

- リアルタイムデータレプリケーション（同期）
  - データの更新（DML）がコミットされると、即リモートサイトに反映
  - サイト間のデータ同期を保障
    - リモートサイトで反映が失敗した場合、全てのサイトでロールバック
  - データの競合が発生しない



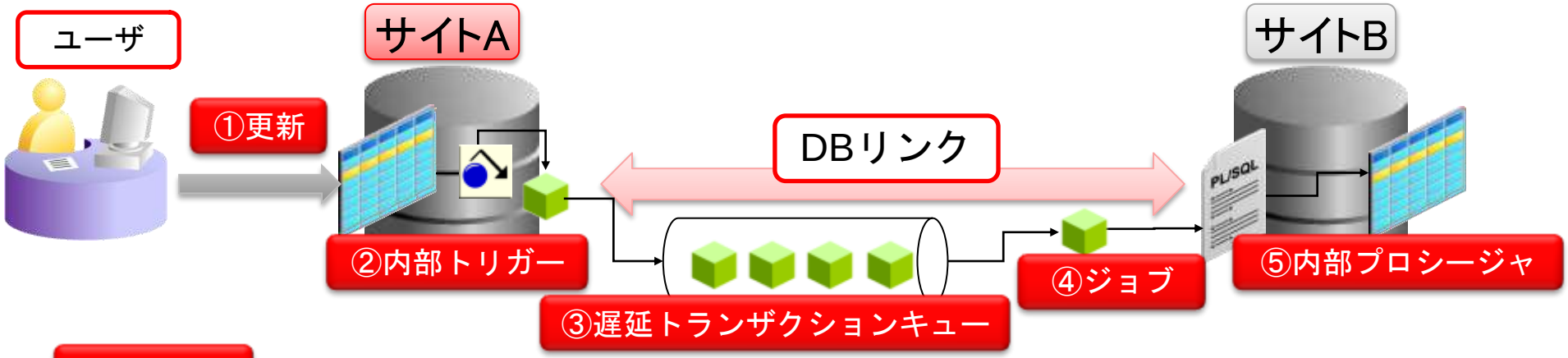
同期の流れ

① 更新処理発生
② 内部トリガーによって更新行、時間、伝播先で実行されるパッケージ名などを生成しリモートサイトに伝播
③リモート・プロシージャ・コール（RPC）を実行し、更新を反映

アドバンスド・レプリケーション

アーキテクチャ

- 非同期レプリケーション
  - 遅延トランザクション・キューを指定した間隔で反映
  - レスポンスタイムへの影響が少ない



同期の流れ

① 更新処理発生

② 内部トリガーによって更新行/時間/伝播先で実行されるパッケージ名を生成

③リモート・プロシージャ・コール（RPC）を遅延トランザクションキューにキューイング

④指定した間隔でJOB（ DBMS_DEFER_SYS.SCHEDULE_PUSH ）を実行しリモート・プロシージャ・コール（RPC）を伝播

⑤リモート・プロシージャ・コール（RPC）を実行し、更新を反映

【参考】遅延トランザクション・キュー

- 遅延トランザクションの数の確認

```
SQL>SELECT DBLINK DEST, COUNT(*) TRANS FROM DEFTRANDEST D GROUP BY DBLINK;
Destination                              Def Trans
 ---------------------------------------- ---------
 ORC2.WORLD                                     1
```

- 伝搬に失敗した遅延トランザクションの確認

```
SQL>SELECT DEFERRED_TRAN_ID ID, ORIGIN_TRAN_DB ORIGIN, DESTINATION,
TO_CHAR(START_TIME, 'DD-Mon-YYYY hh24:mi:ss') TIME_OF_ERROR,
ERROR_NUMBER FROM DEFERROR ORDER BY START_TIME;

ID          ORIGIN           Destination      Time of Error              Error Number
----------  ----------------  ---------------- ---------------------------- -------
1.8.2470 ORC2.WORLD ORC1.WORLD 22-Oct-2003 07:19:14 1403
```

- 次にスケジュールされているジョブの実行時間の確認

```
SQL>SELECT DBLINK,INTERVAL,TO_CHAR(NEXT_DATE, 'YYYY/MM/DD
HH:MM'),TO_CHAR(LAST_DATE,'YYYY/MM/DD HH:MM') FROM DEFSCHEDULE;

DBLink                   INTERVAL              NEXT_DATE       LAST_DATE
-------------------------  ----------------------  ------------------- ----------------------
orcl.jp.oracle.com   SYSDATE + 1/144  2011/08/10 07:00  2011/08/10 06:50
```

Agenda

- データ連携ソリューション
- レプリケーションとは
- アドバンスド・レプリケーション環境の構築
- レプリケーション環境の運用管理
  - 競合
  - 障害時のリカバリ
- Appendix

環境構築の流れ

1. 初期化パラメータの設定(マスター定義・マスター)
   - Global NameをTrueに設定
2. ネットワーク定義(マスター定義・マスター)
   - tnsnames.oraの編集
3. レプリケーション環境の設定(マスター定義・マスター)

Point1:
- レプリケーション管理者の作成
- プロパゲータを登録
- 受信者を登録

Point2:
- DB Linkの作成

Point3:
- スケジュール・プッシュを設定
- パージスケジュールの作成

環境構築の流れ

4. レプリケーショングループの設定（マスター定義サイト）

Point4:
- レプリケーション・グループを作成
- レプリケーション・グループにオブジェクトを追加
- マスター・サイトの追加

Point5:
- レプリケーションサポートの作成

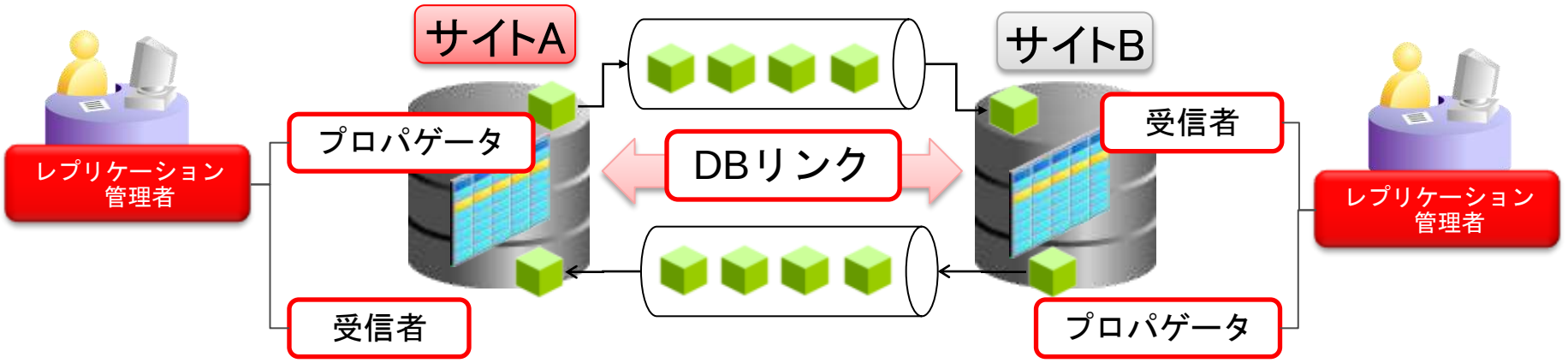
5. レプリケーションの開始

Point6:
レプリケーションの開始

構築手順の詳細に関してはAPPENDIXの項目をご参照ください。

Point1： ロールとユーザ

- リプリケーション環境を構築する上で必要なロールとユーザ
  - レプリケーション管理者（ユーザ）
    - 個々のマスター・レプリケーション・グループの作成と
      メンテナンキューの管理などのレプリケーション環境の管理を行う
    - dbms_repcat_admin.grant_admin_any_schemaを実行
  - プロパゲータ（ロール）： dbms_defer_sys.register_propagatorを実行
    - 遅延トランザクション・キューを伝播する作業を実行
  - 受信者（ロール）:dbms_repcat_admin.register_user_repgroupを実行
    - プロパゲータから遅延トランザクションを受信し、適用

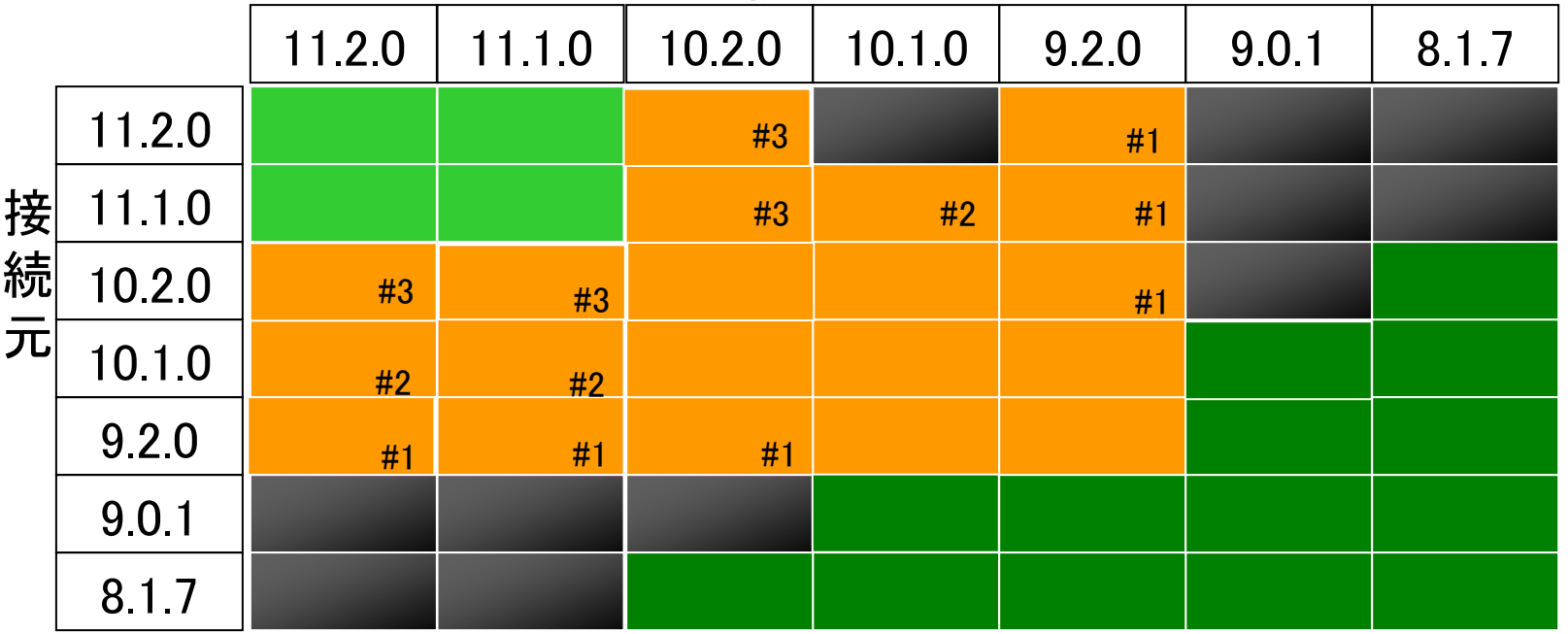


Point2： DB Linkの作成

- DB Link:
  - 別のDBサーバーへの一方向の通信経路を定義するポインタ（1方向）
  - tnsnames.oraに接続先のDB情報を記述
  - CREATE DATABASE LINK文を利用して作成
- 2種類のDB Link:
  - パブリック：すべてのユーザーがそのDBリンクを利用してアクセス可能
  - プライベート：そのDBリンクを作成したユーザーのみがアクセス可能

Point2：DBリンクのサポート状況

※2010年10月20日時点



動作保障

サポート
（Extended Suport(ES)契約を
有する場合のみ不具合の修正が可能）

サポート終了
（以前、この組み合わせはサポートされておりました。
現在は、不具合の修正は行われません。）

サポート外

#1 9.2.0.4（パッチセット・リリース 9.2.0.4）以上の適用が必要です
#2 データベース・リンクを行う場合、PLSQLが使用できるようにするために、10g側は、10.1.0.5以上の必要があります。
#3 データベース・リンクを行う場合、PLSQLが使用できるようにするために、10g側は、10.2.0.2以上の必要があります。

Point3：同期のタイミングの設定

- 非同期レプリケーションを行う場合、遅延トランザクションを伝搬先に伝搬する頻度を設定
  - DBMS_DEFER_SYS.SCHEDULE_PUSHパッケージを利用

- 10分間隔で同期を行う場合

```
execute dbms_defer_sys.schedule_push(
destination => 'DB1.WORLD',
interval => 'sysdate + 1/144',
next_date => sysdate,
stop_on_error => FALSE,
delay_seconds => 0,
parallelism => 1);
```

同期先のデータベースを指定

同期のタイミングを指定（この例では10分間隔を指定）
１＝24時間　　24時間 x 1/144=10分

Point3：同期のタイミングの設定

- 遅延トランザクション・キューから正常に適用されたトランザクションを定期的にパージする頻度を設定
  - DBMS_DEFER_SYS.SCHEDULE_PURGEパッケージを利用

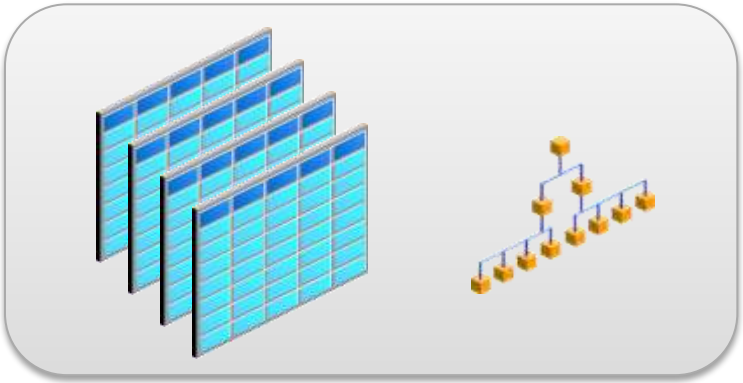
- 1時間間隔でパージを行う場合

```
execute dbms_defer_sys.schedule_purge(
next_date => sysdate,
interval => ' sysdate + 1/24',
delay_seconds => 0,
rollback_segment => '');
```

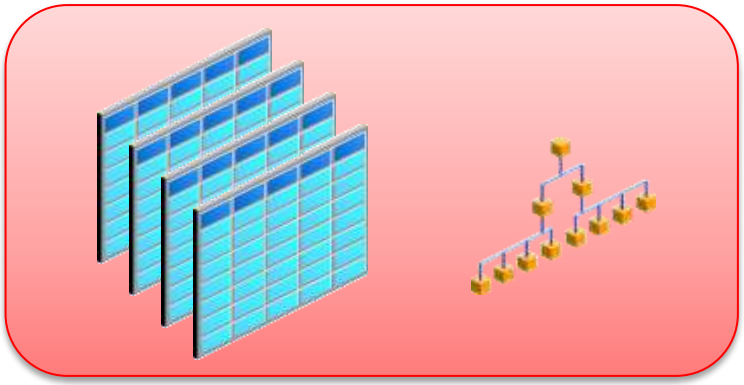
Point4:レプリケーション・グループの設定

- レプリケーション・グループ
  - ➢レプリケーション・オブジェクト(表、索引など)を管理する単位
  - ➢レプリケーションの再開・停止などの管理はレプリケーショングループ単位

レプリケーション・グループ



レプリケーション・グループ



レプリケート可能オブジェクト

- 表、索引、シノニム
- ビュー、オブジェクト・ビュー
- トリガー
- パッケージおよびその本体
- プロシージャ、ファンクション
- ユーザ定義型および型の本体
- 索引タイプ、ユーザ定義演算子

レプリケート不可オブジェクト

- 順序
- クラスタ表
  （事前にリモートサイトで作成されていればサポート）
- LONG型,LONG RAW型を含む表
- BFILEsを含む表
- UROWID列を含む表

Point4:レプリケーション・グループの設定

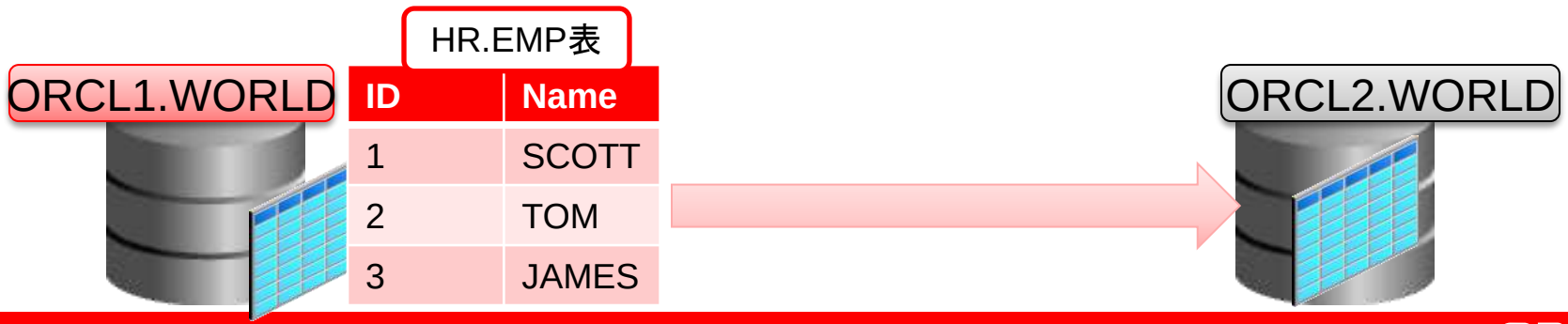
HRスキーマ内のEMP表をレプリケーションする場合

1. 全てのサイトでレプリケーション・オブジェクトを格納するスキーマを作成

```
connect system/oracle@ORCL2.WORLD;
CREATE USER HR IDENTIFIED BY HR;
GRANT CONNECT, RESOURCE TO HR;
```

2. オブジェクトにプライマリー・キー（PK）を作成
   - レプリケーションではPKを利用し、どの列が変更されたかを認識する為、PKがないとエラーが発生

```
ALTER TABLE HR.EMP ADD (CONSTRAINT EMP_PK PRIMARY KEY(ID));
```



Point4:レプリケーション・グループの設定

3. レプリケーション・グループの作成

- レプリケーション管理者でCREATE_MASTER_REPGROUPを実行

```
CONNECT REPADMIN/REPADMIN@ORCL2.WORLD

EXECUTE DBMS_DBMS_REPCAT.CREATE_MASTER_REPGROUP
(GNAME => 'HR_MG');
```

→ レプリケーション・グループ名を指定

4. オブジェクトをレプリケーション・グループに追加

```
BEGIN
DBMS_REPCAT.CREATE_MASTER_REPOBJECT (
gname => 'HR_MG',
sname =>'HR',
use_existing_objects => TRUE,
type=>'TABLE',
oname =>'emp',
copy_rows => TRUE);
END;
```

- gname → レプリケーショングループ名を指定
- sname → スキーマ名を指定
- type → オブジェクトタイプを指定
- oname → オブジェクト名を指定
- use_existing_objects → 既存オブジェクトを再利用するかどうか指定

オブジェクト単位で設定を行う為、DB全体の同期には不向き

Point4:レプリケーション・グループの設定

5. マスターサイトの追加

- DBMS_REPCAT.ADD_MASTER_DATABASEパッケージを実行

```
BEGIN
  DBMS_REPCAT.ADD_MASTER_DATABASE (
    gname => ‘HR_MG',
    master => ‘ORCL2.WORLD');
END;
/
```

- 同期レプリケーション環境を作成するには propagation_modeをsynhchronousに設定
  - ALTER_MASTER_PROPAGATIONを利用して、後で変更することも可能

【参考】レプリケーション・グループ

- レプリケーション・グループの確認

```
SQL>select gname, dblink, masterdef from dba_repsites;
GNAME          DBLINK              MASTERDEF
------------   ------------------  ------------------------------
HR_Rep_Group   ORCL.JP.ORACLE.COM  Y
```

- レプリケーション・グループに属する
オブジェクトの確認

```
SQL>select sname, oname, type, status from dba_repobject;

sname oname         type    status
----- ------------- ------ --------
HR    JOBS          TABLE  VALID
HR    JOB_CONFLICT  TRIGGER VALID
```

【参考】サポートされるData Type

レプリケート可能Data Type

VARCHAR2
NVARCHAR2
CHAR
NCHAR
ROWID
RAW
NUMBER
DATE
TIMESTAMP
TIMESTAMP WITH TIME ZONE
TIMESTAMP LOCAL TIME ZONE
INTERVAL YEAR TO MONTH
INTERVAL DAT TO SECOND
BLOB
CLOB
NCLOB
USER-DEFINED TYPE
NESTED TABLE
VARRAY
REF

レプリケート不可Data Type

LONG
LONG RAW
外部LOB
BFILE

- LONGデータ型は
  LOBに変換することで対応が可能

LONG型からLOB型への返還に関しては
以下のURLを参照ください
「レプリケート表でのLONG列のLOB列への変換」
http://otndnld.oracle.co.jp/document/products/oracle11g/111/doc_dvd/server.111/E05779-02/rarmanage.htm#i31438

Point5 : レプリケーション・サポート

- オブジェクトをレプリケートするには
  レプリケーション・サポートの作成が必要
- レプリケーション・サポート
  - オブジェクトのレプリケーションに必要なトリガーやパッケージを作成
  - レプリケート・オブジェクトの構成を変更する度、
    レプリケーション・サポートの再作成が必要

```
Execute
DBMS_REPCAT.GENERATE_REPLICATION_SUPPORT (
      sname => 'hr',
      oname => 'emp',
      type => 'TABLE');
```

Point6 : 同期の開始

- 同期の開始、停止はマスター定義サイトで実行
- DBMS_REPCAT.RESUME_MASTER_ACTIVITYを実行して同期を開始

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

- 同期の停止にはDBMS_REPCAT.SUSPEND_MASTER_ACTIVITYを実行

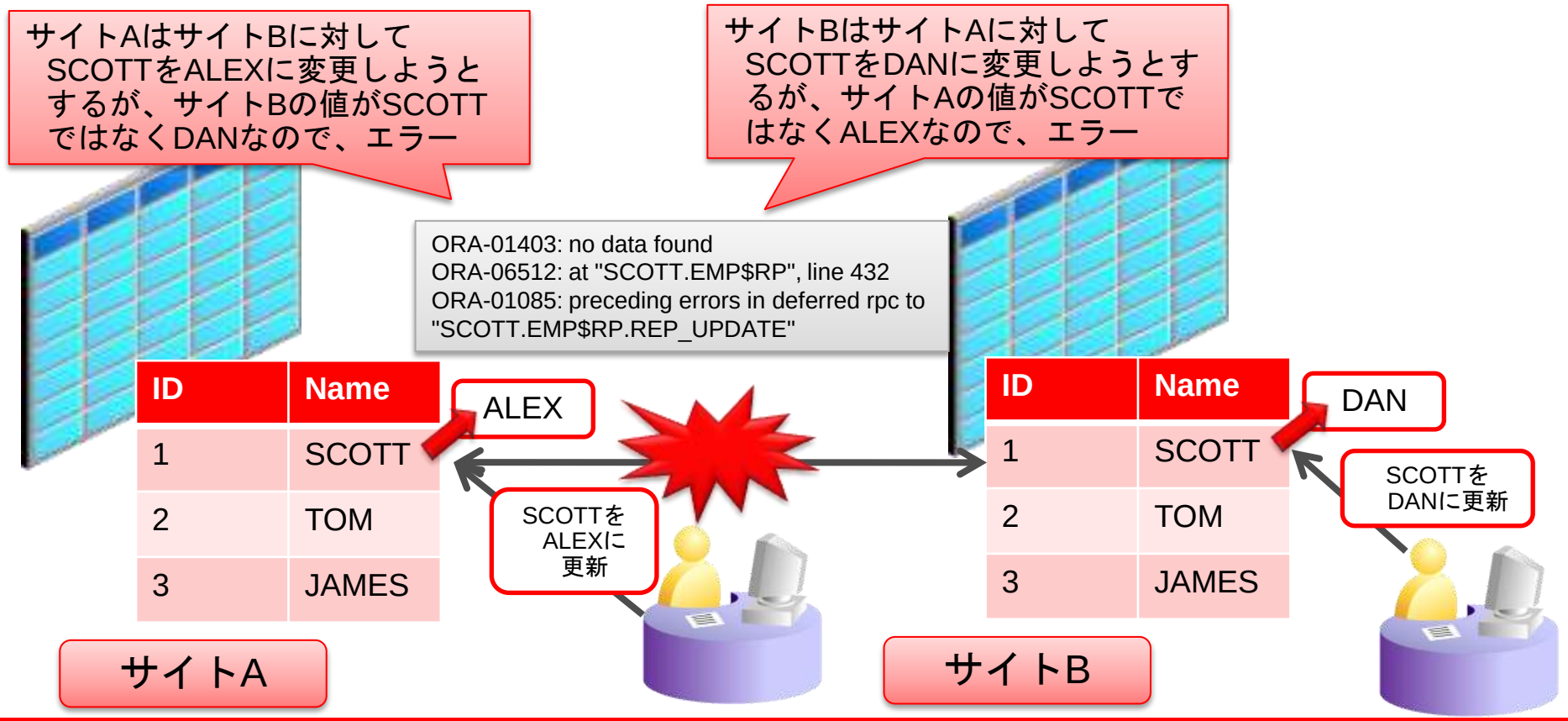
```
execute dbms_repcat.resume_suspend_activity(
gname => 'SCOTT');
```

Agenda

- データベース間連携
- アドバンスド・レプリケーションとは
- レプリケーション環境の運用管理
  - 競合
  - 障害時のリカバリ

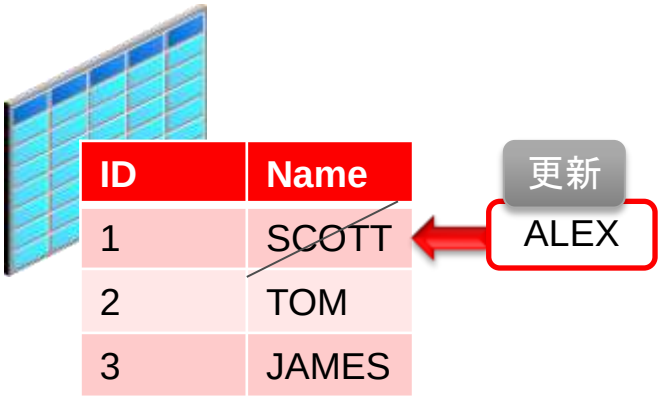
競合

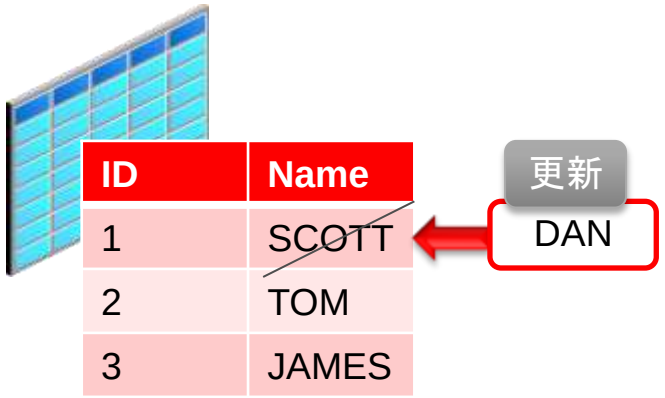
- 同じ行に対して別々のサイトから同時に更新が発生するとデータの競合（不整合）が発生する
  - 同期レプリケーションの場合は競合が発生しない



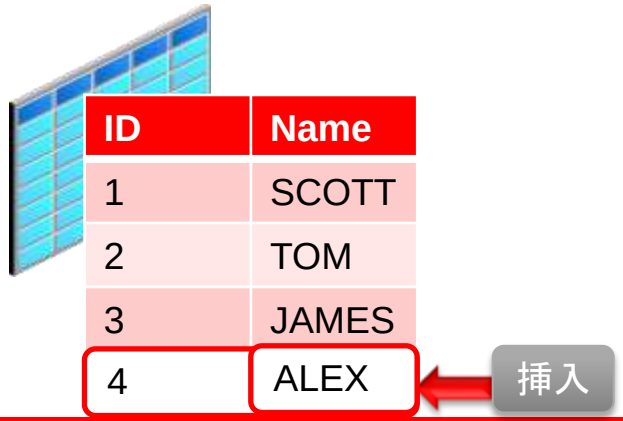
競合の種類

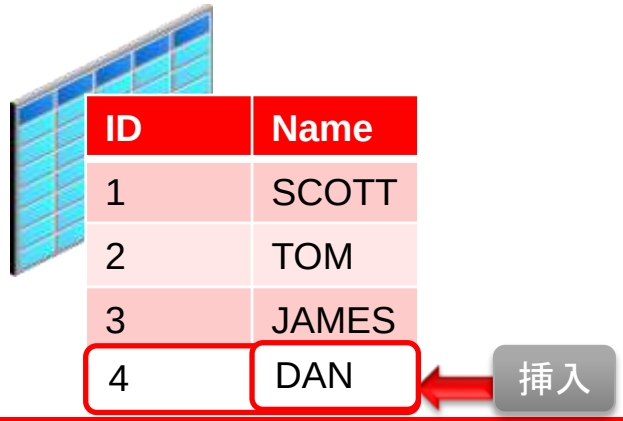
- 更新競合：同じ行を同時に更新





- 一意性競合：エンティティの整合性（主キー制約や一意制約など）が失われる競合

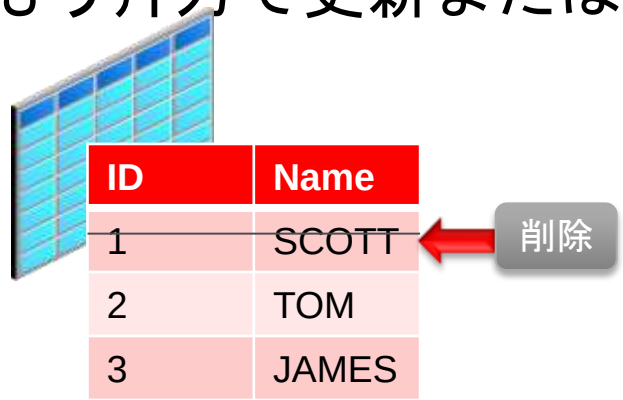


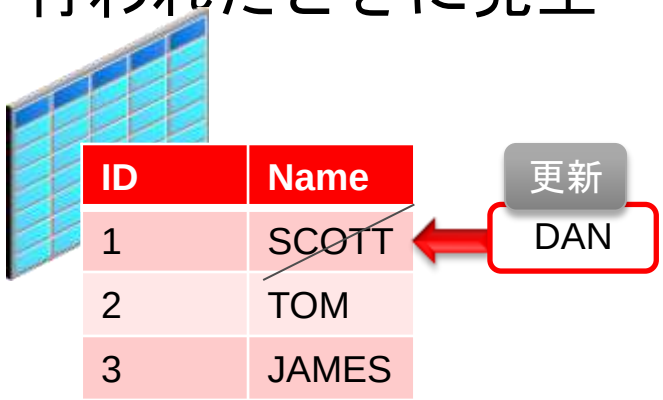


競合の種類

4種類の競合

- 削除競合： 同じ行に対して片方でデータを削除し、もう片方で更新または削除が行われたときに発生





競合の解決方法

- 競合が発生しないレプリケーション環境を設計する
  - 例：サイトAでは表Aのみを更新し、
    サイトBでは表Bのみの更新を行う
  - 列グループを使用する
- 競合解決のルールを設定する
  - Oracleから提供されているビルトインを利用

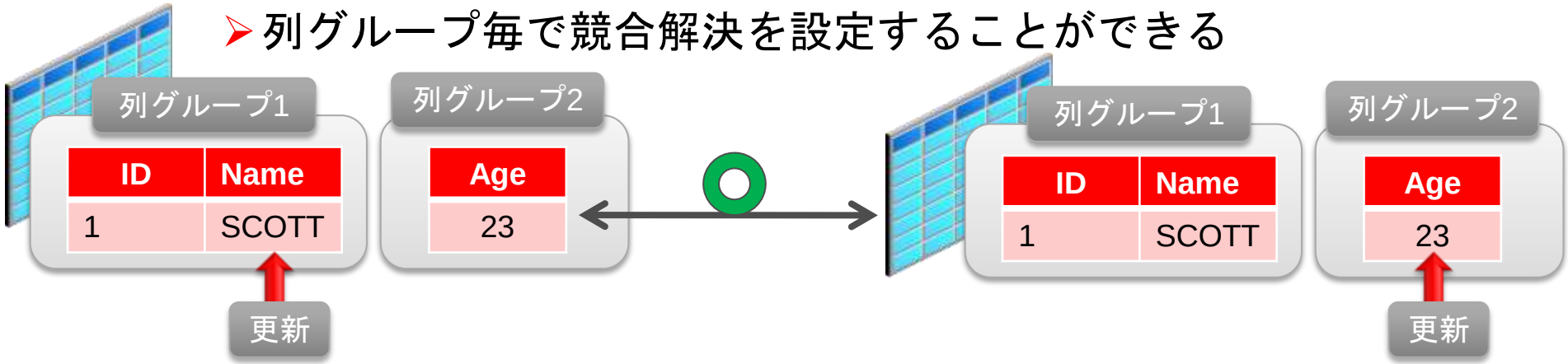
競合の種類によって、競合の解決方法が異なります

列グループの利用

更新競合の回避

- 列を論理的な単位でグループ化

- 競合は列グループレベルで検知
  - 同じ行の更新でも列グループが異なれば競合は発生しない
  - 列グループ毎で競合解決を設定することができる



【参考】列グループの管理

- 列グループ情報はDBA_REPCOLUMN_GROUPビュー、DBA_REPGROUPED_COLUMNビューで確認
- 列グループに含まれていない列はシャドウ列グループに含まれる
- 列グループの管理にはDBMS_REPCATプロシージャを利用
  - MAKE_COLUMN_GROUP
  - DROP_COLUMN_GROUP
  - ADD_GROUPED_COLUMN
  - DROP_GROUPED_COLUMN

列グループ1

ID	Name
1	SCOTT

列グループ2

Age
23

シャドウ列グループ

Gender
MALE

列グループ作成例

```
DBMS_REPCAT.MAKE_COLUMN_GROUP (
  sname=>'scott',
  oname=>'emp',
  column_group=>'column_group_1',
  list_of_column_names =>'ID,Name');

DBMS_REPCAT.MAKE_COLUMN_GROUP (
  sname=>'scott',
  oname=>'emp',
  column_group=>'column_group_2,
  list_of_column_names =>'age');
```

競合の解決

更新競合の解決

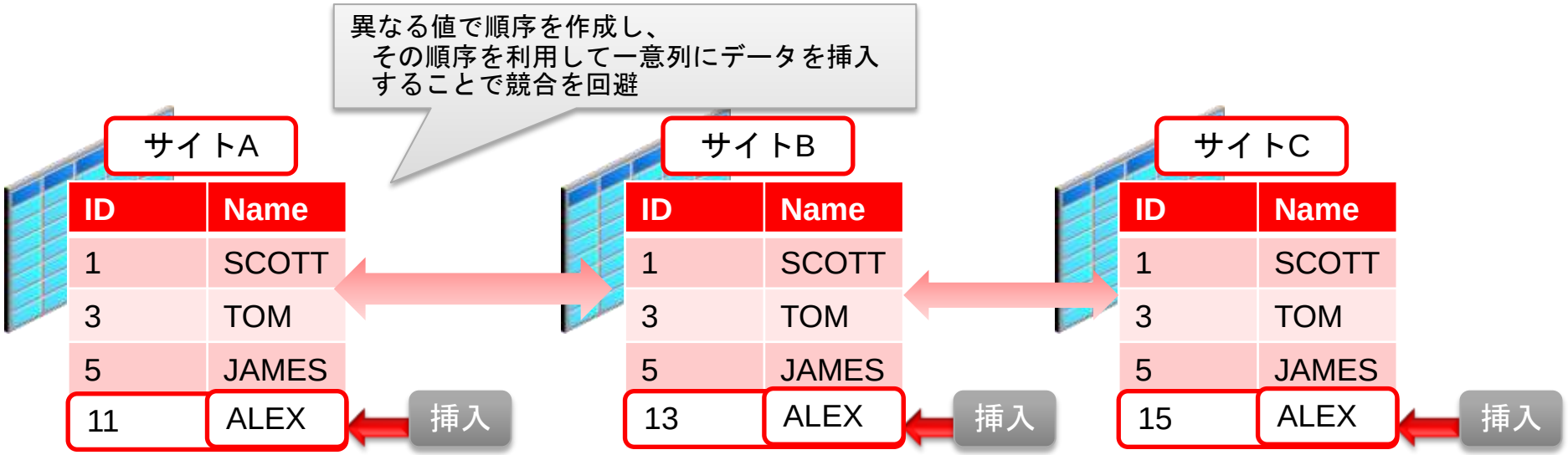
- 競合解消のビルトインを複数用意（データ収束を保障）
  - 一般的に利用されている方法
    - 最新のタイムスタンプ
      - 最も新しいトランザクションの値で行を更新
    - 上書き
      - レプリケーション元の新しい値で、レプリケーション先のカレント値を置き換える
  - その他
    - 加算
    - 平均
    - 廃棄
    - 最も古いタイムスタンプ
    - 最大値
    - 最小値
    - 優先グループ
    - サイト優先順位

各ビルトインの実装方法は以下のURLをご確認ください。
http://otndnld.oracle.co.jp/document/products/oracle11g/111/doc_dvd/server.111/E05779-02/rarconflictres.htm#i14774

順序の利用
一意競合の回避

- 各サイトで順序を作成(CREATE SEQUENCE)

パラメータ	サイトA	サイトB	サイトC
START WITH	1	3	5
INCREMENT BY	10	10	10
作成される値の例	1,11,21,31,41	3,13,23,33,43	5,15,25,35,45

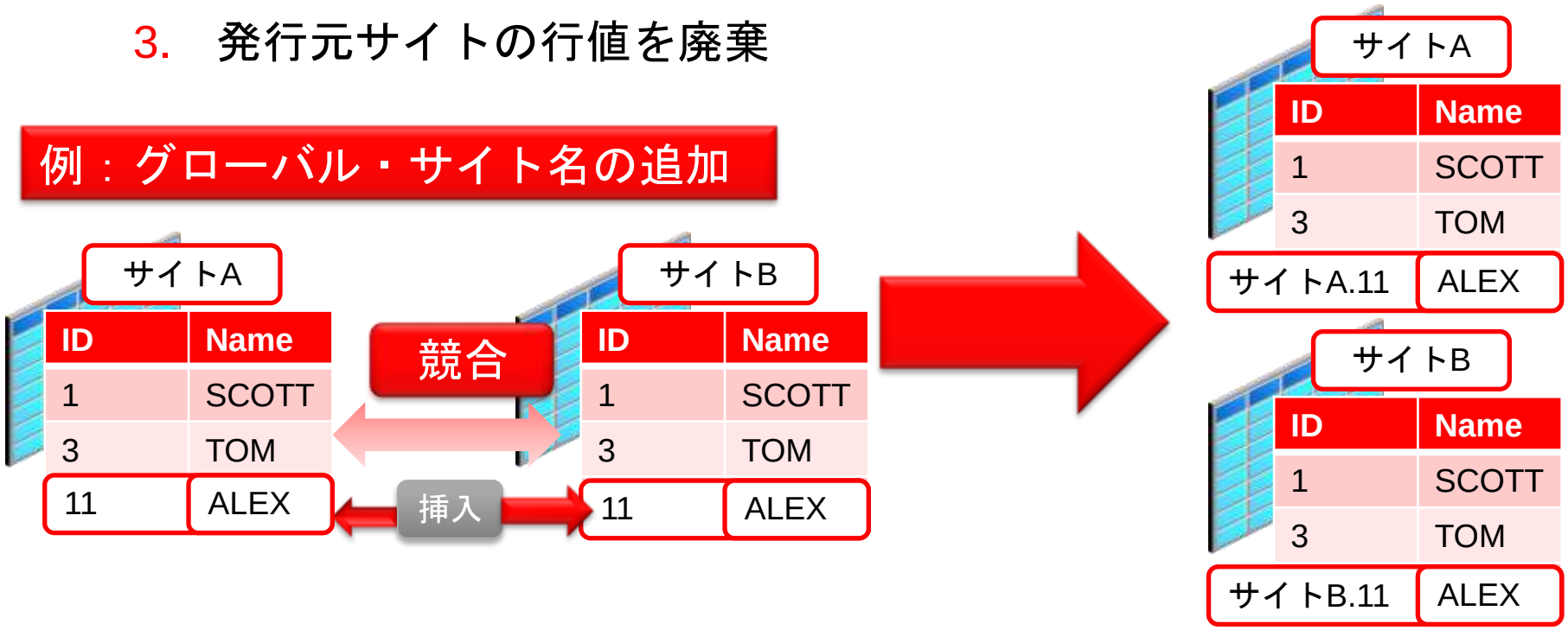


競合の解決

一意競合の解決

- 競合解消のビルトインを3種類用意
  （DBMS_REPCAT.ADD_UPDATE_RESOLUTION を利用）
  1. 発行元サイトのグローバル・サイト名を、発行元サイトの列値に追加
  2. 生成された順序番号を、発行元サイトの列値に追加
  3. 発行元サイトの行値を廃棄

例：グローバル・サイト名の追加



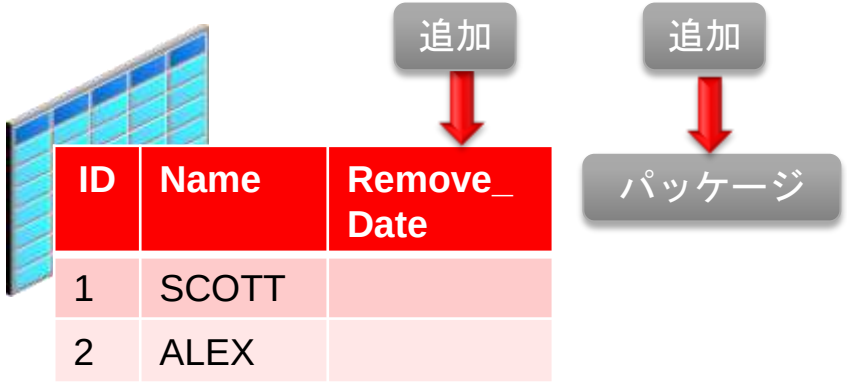
論理的削除の利用
削除競合の回避

- 行の削除を実行しないアプリケーションを作成する
- 行に削除のマークを付け、論理的に削除された行をプロシージャを使用して定期的にパージ

削除競合を解消するビルトインは提供されていない為、上記方法で、削除競合を回避します

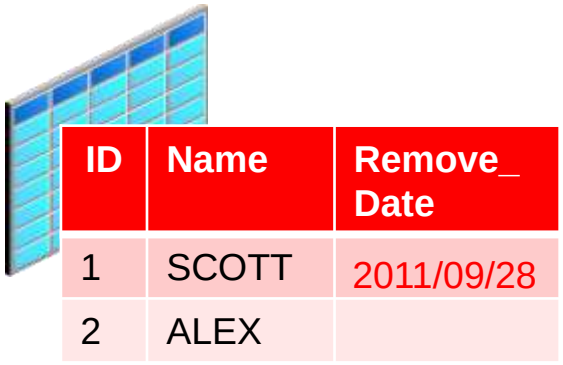
論理的削除の利用
削除競合の回避

1.削除マーク用の列・パッケージを追加



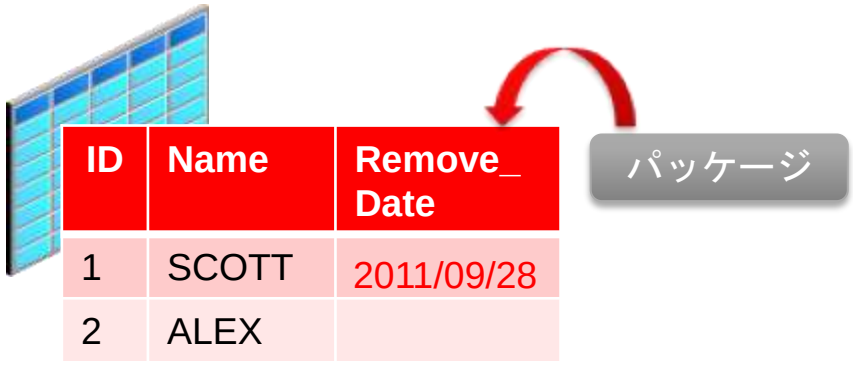
ID	Name	Remove_Date
1	SCOTT	
2	ALEX	

2.行を削除（実際にはデータを削除せずに列にマークを追加）



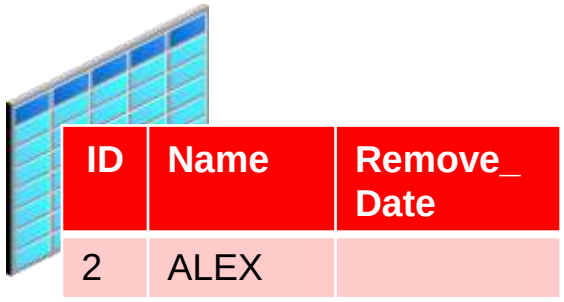
ID	Name	Remove_Date
1	SCOTT	2011/09/28
2	ALEX	

3.パッケージを実行



ID	Name	Remove_Date
1	SCOTT	2011/09/28
2	ALEX	

4.削除マークが付いている列を削除



ID	Name	Remove_Date
2	ALEX	

【まとめ】競合の回避・解決

- 非同期レプリケーションの場合、
  競合が発生する可能性がある
  - そもそも競合が発生しないアプリの設計を行う
- 4種類の競合

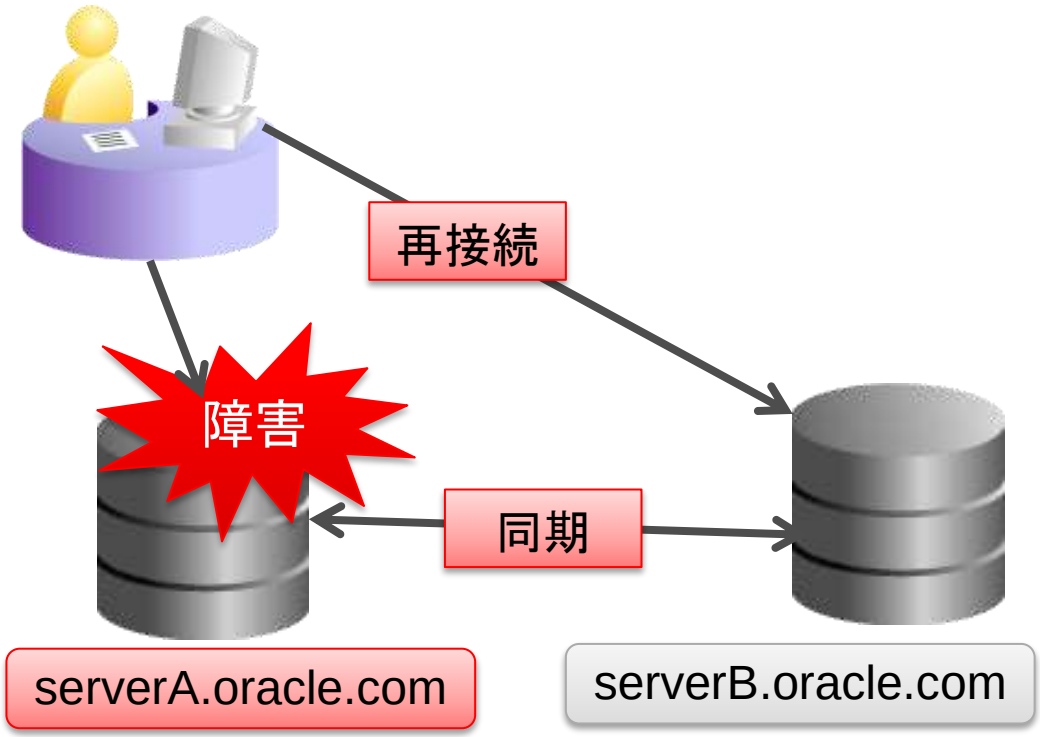
- 更新競合
  - 回避方法：列グループの利用
  - 解決方法：ビルトインの利用
- 一意性競合
  - 回避方法：順序の利用、GUIDの利用
  - 解決方法：ビルトインの利用
- 削除競合
  - 回避方法：論理的削除の実行
  - 解決方法：なし

Agenda

- データベース間連携
- アドバンスド・レプリケーションとは
- レプリケーション環境の運用管理
  - 競合
  - 障害時のリカバリ

障害時の接続サイトの切り替え

- tnsnames.oraに記述をすることで、
  接続時フェイルオーバーを利用できます

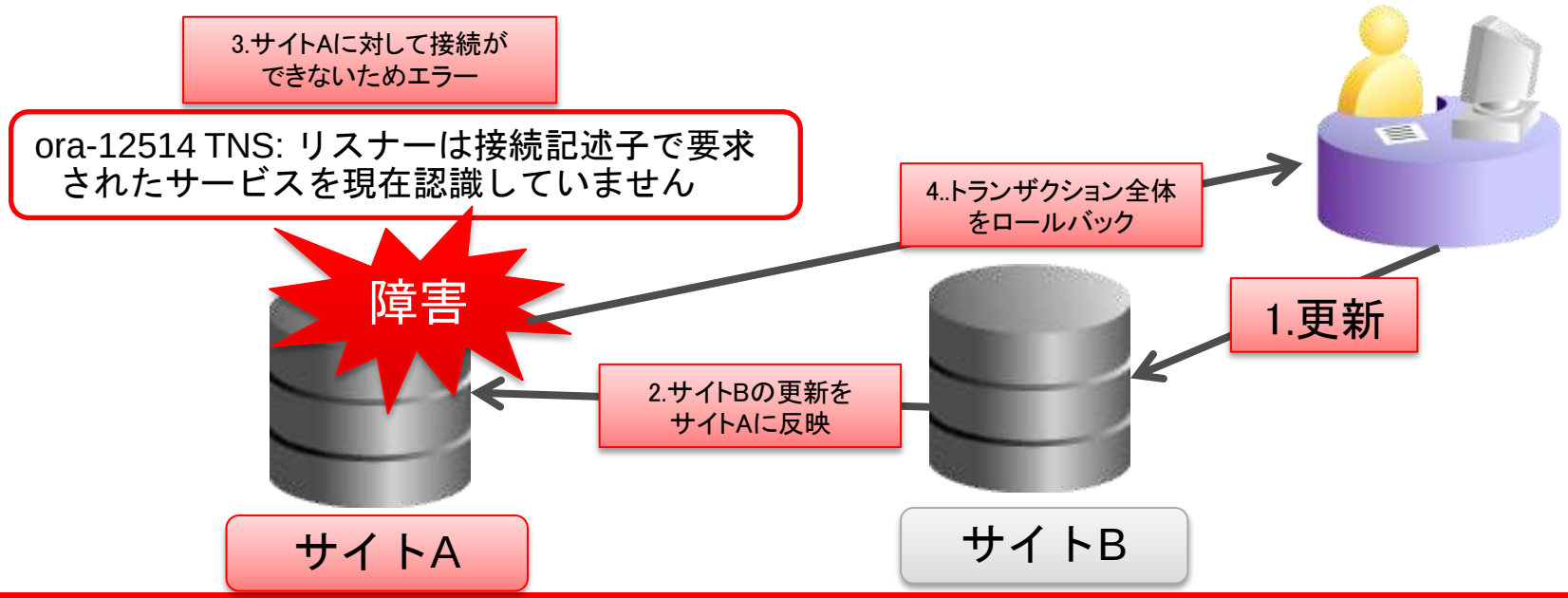


tnsnames.ora

```
ORCL=
 (DESCRIPTION_LIST=
  (FAILOVER=on)(LOAD_BALANCE=off)
  (DESCRIPTION =
    (ADDRESS_LIST =
      (ADDRESS = (PROTOCOL=TCP)(HOST=serverA.oracle.com(PORT=1521))
    )
    (CONNECT_DATA = (service_name = serverA.oracle.com))
  )
  (DESCRIPTION =
    (ADDRESS_LIST =
      (ADDRESS = (PROTOCOL=TCP)(HOST=serverB.oracle.com)(PORT=1521))
    )
    (CONNECT_DATA = (service_name = serverB.oracle.com)
  )
 )
```

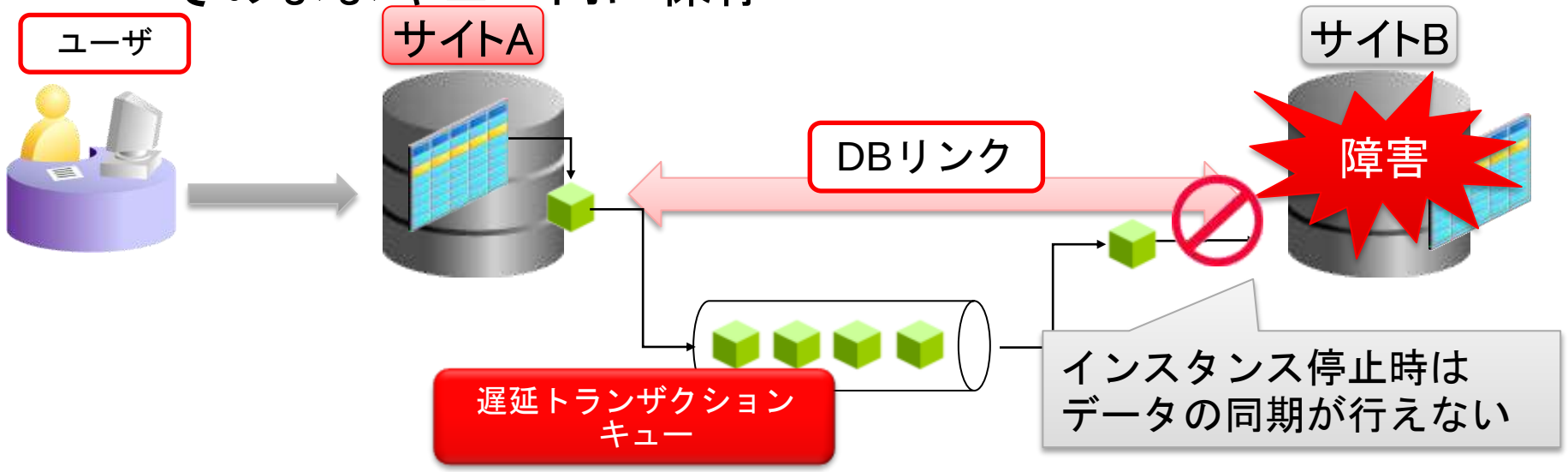
障害時の問題

- 同期レプリケーションの場合、更新が行えなくなる
  - 2フェーズコミットの為、リモートサイトでのコミット完了を待機する
  - 同期レプリケーションを解除することで更新可能（マスター定義サイトに障害が発生した場合、マスター定義サイトを他のマスターサイトへ変更する必要あり）



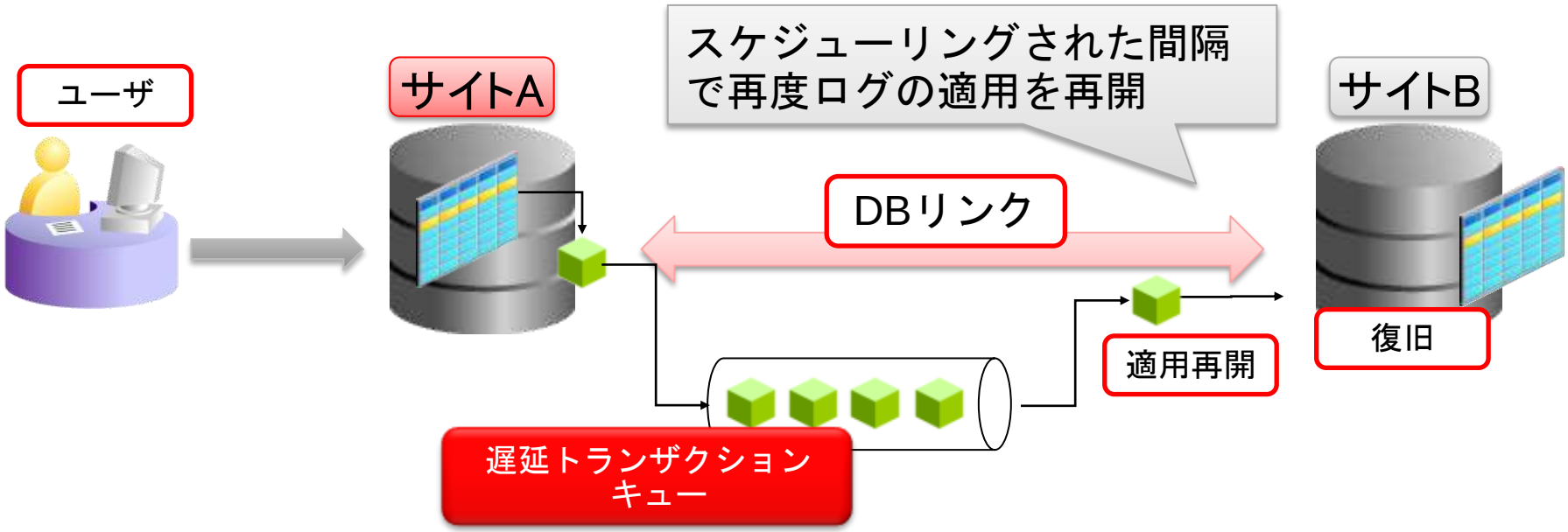
レプリケーションサイトの復旧方法
インスタンス障害の場合

- インスタンス障害
  - OS障害
  - バックグランド・プロセス障害
  - Shutdown Abortの実行
- 障害等で適用できなかった遅延トランザクションキューはそのままキュー内に保存



レプリケーションサイトの復旧方法
インスタンス障害の場合

- Oracleが自動でインスタンスリカバリを実行し、インスタンスを復旧
- インスタンスが復旧した段階で、再度遅延トランザクションキューの適用を再開



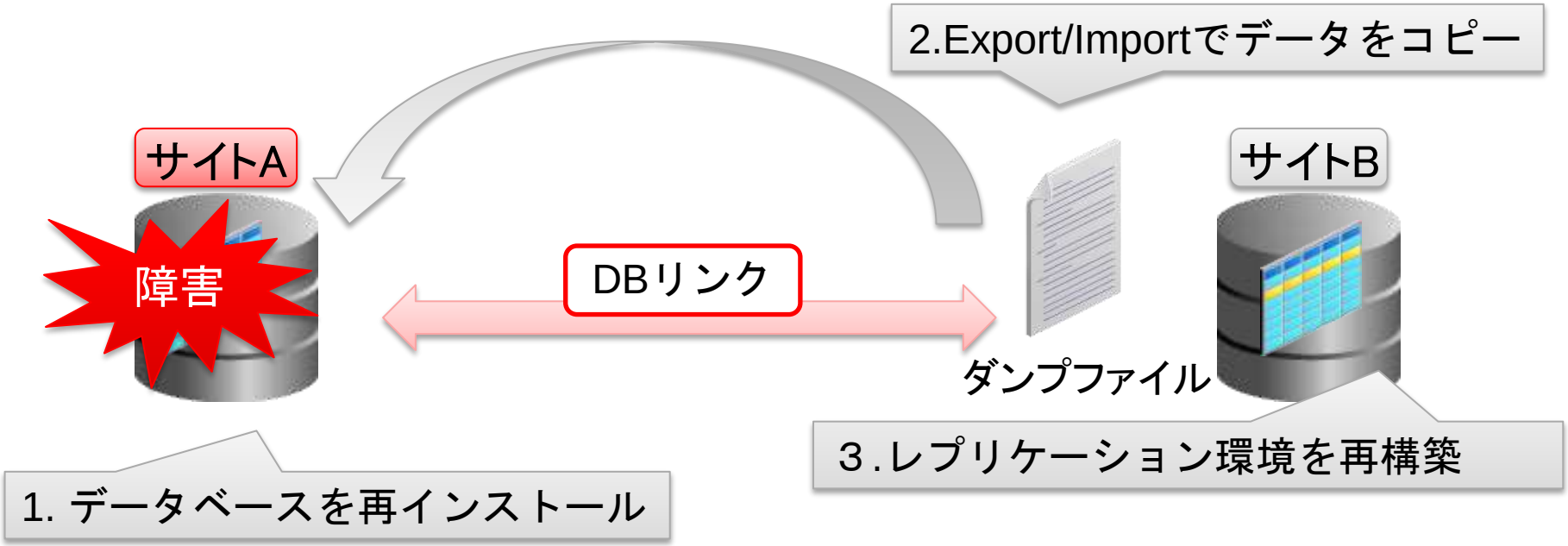
レプリケーションサイトの復旧方法
メディア障害の場合

- メディア障害
  - Oracle Databaseファイルの消失
    （Data file,制御ファイル、Redoログ等）
  - ディスクドライブ、ディスクコントローラの障害

**運用方法によっては、
レプリケーション環境の再構築が必要になる**

レプリケーションサイトの復旧方法

- バックアップを取得していない場合：
  - Export/Importでデータの移行を実行
  - レプリケーション環境を再度作成



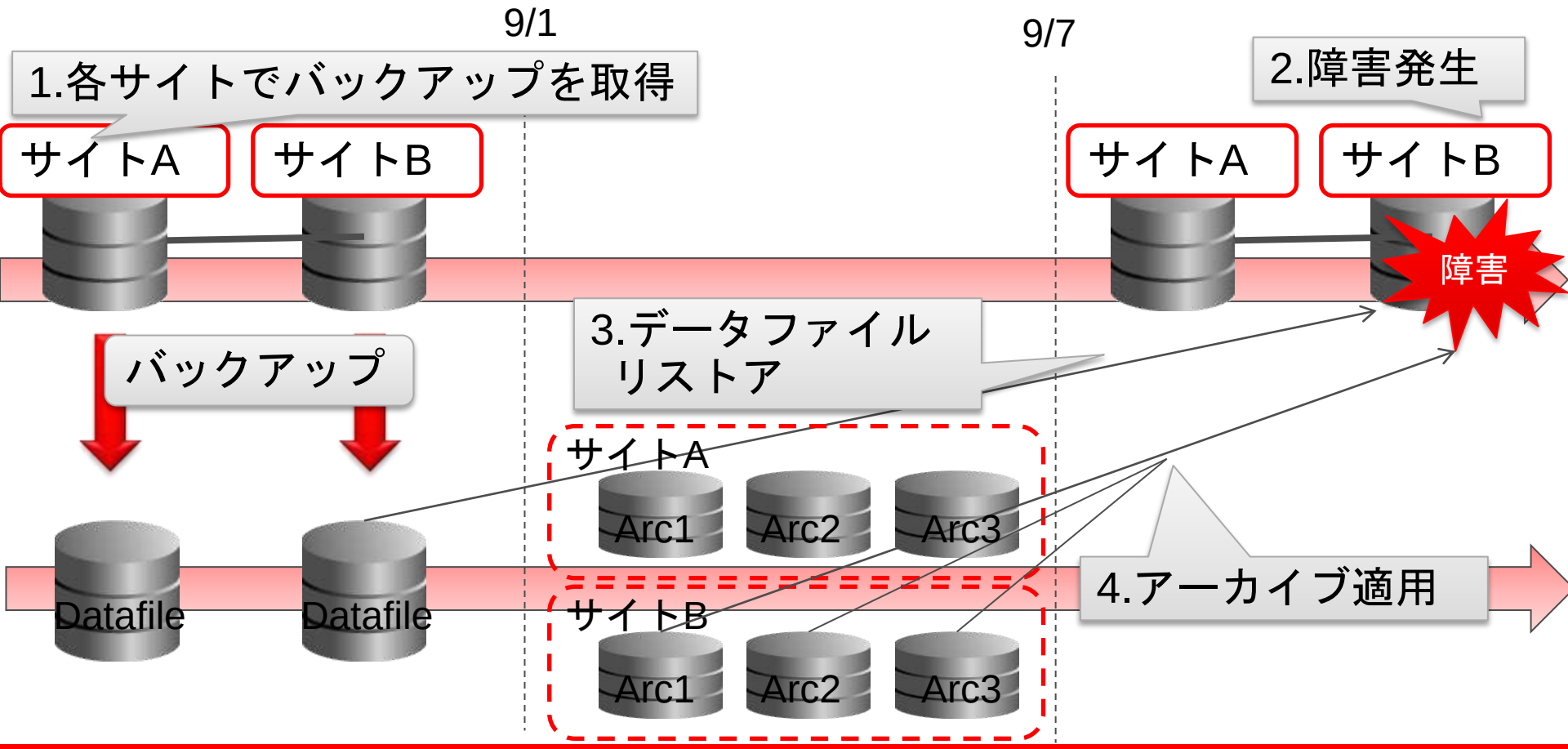
データ量に応じてサイトの復旧に時間がかかります

レプリケーションサイトの復旧方法
メディア障害の場合

- バックアップを取得していた場合：
  - 完全リカバリ
    - 障害直前まで復旧が可能
      - コミットされたデータはリカバリされ、コミットされてないデータはロールバック
  - 不完全リカバリ
    - レプリケーションサイト間で不整合が発生
    - 特定の時間までロールバック

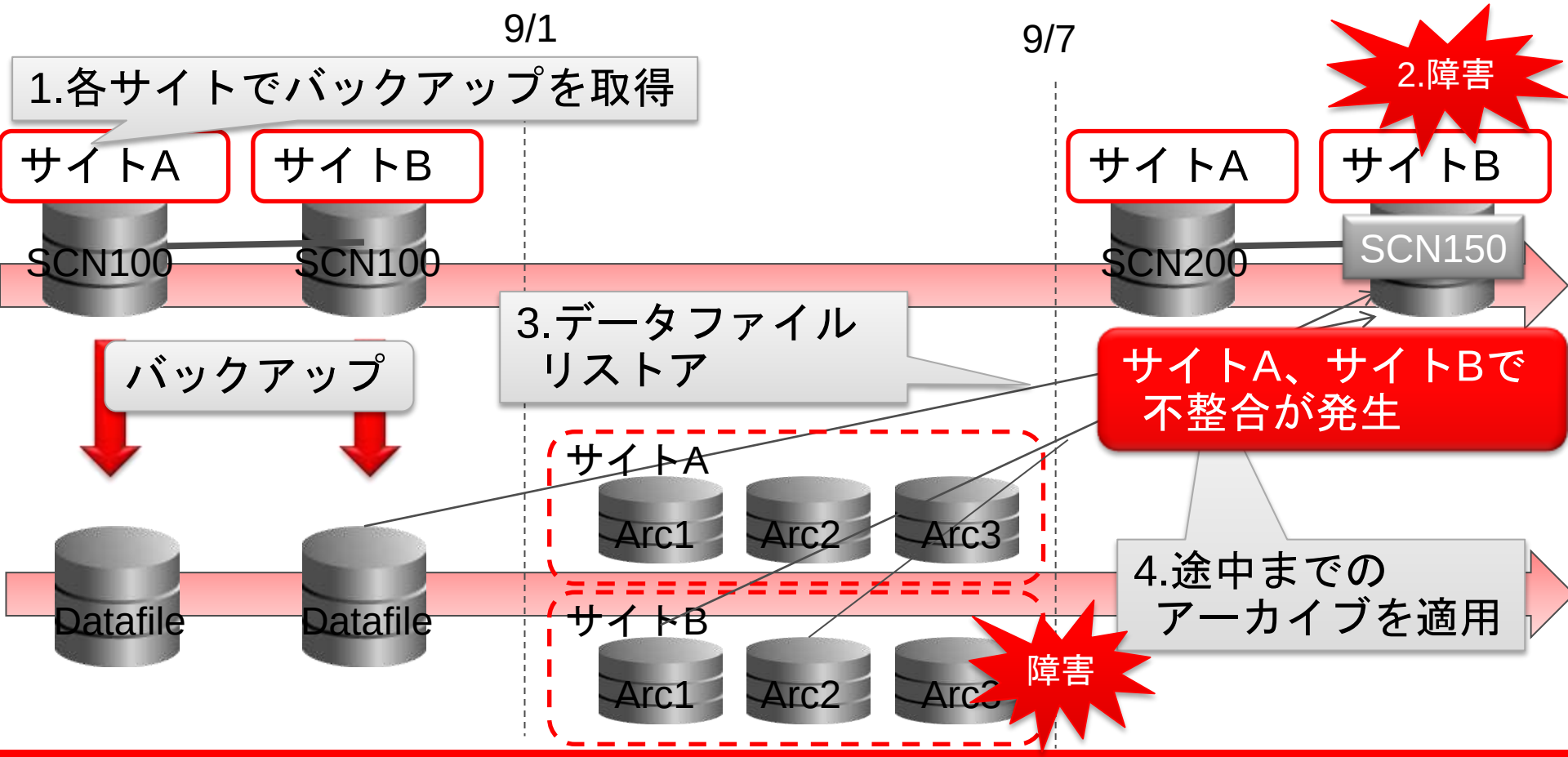
完全リカバリ

- 障害直前まで復旧し、遅延ログの適用を再開
  - コミット済みトランザクションは普及
  - コミットされていないトランザクションはロールバック



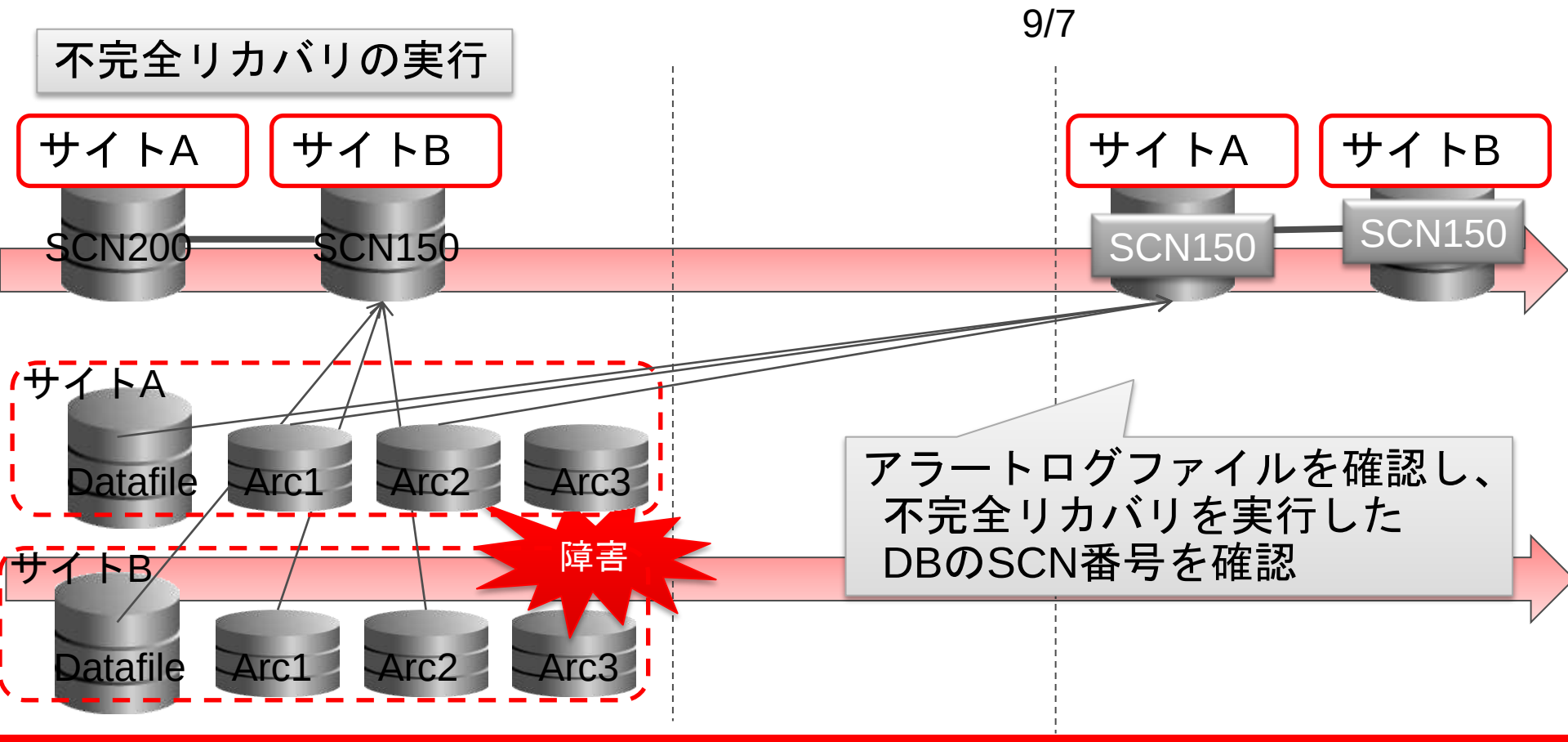
不完全リカバリ

- 不完全リカバリを実行した場合、サイト間で不整合が発生



不完全リカバリ

- 不完全リカバリを実行した場合、障害が発生したサイト以外でも同じ時点までリカバリ



【まとめ】可用性用途としてのアドバンストレプリケーション

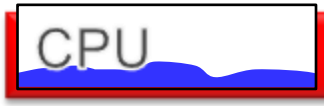
- 障害時の切り替え
  - 接続時フェイルオーバーを利用することで、システムを止めずに別サイトへの切り替えが可能
  - 同期レプリケーションの場合、１つのサイトで障害が起きると、更新が行えなくなる
- 障害からのリカバリ
  - バックアップを取得していない場合は、環境の再構築が必要
  - アーカイブログ・モードで運用することで、障害復旧まで回復が可能で、その状態から再度同期を再開

【まとめ】アドバンスト・レプリケーション

レプリケーション比較

	読み取り専用Mview	更新可能Mview	同期レプリケーション	非同期レプリケーション
参照/更新	× 1方向連携	○ 双方向連携	○ 双方向連携	○ 双方向連携
可用性	× 参照のみ	○ 参照・更新可能	× 参照のみ	○ 参照・更新可能
競合	○ 競合なし	× 競合あり	○ 競合なし	× 競合あり
データの整合性	○ データの整合性を保証	× データの不整合が発生	○ データの整合性を保証	× データの不整合が発生
パフォーマンス	○ レスポンスへの影響が少ない	○ レスポンスへの影響が少ない	× レスポンスへの影響が大きい	○ レスポンスへの影響が少ない
異環境連携	○ DBLinkのサポート範囲内で可能	○ DBLinkのサポート範囲内で可能	○ DBLinkのサポート範囲内で可能	○ DBLinkのサポート範囲内で可能

【まとめ】アドバンスト・レプリケーション

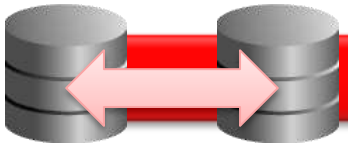


負荷分散

- 複数サイトでの処理の分散化
  - CPU負荷軽減
- 競合の回避・解決の仕組みが必要

可用性

- データの2重化
  - 障害時のデータ保護
- 適切なバックアップの取得が重要



データ連携

- 異なる環境間でのデータ共有
- DB Linkがサポートされる範囲で連携が可能

Agenda

- データベース間連携
- アドバンスド・レプリケーションとは
- レプリケーション環境の運用管理
  - 競合
  - 障害時のリカバリ
- Appendix

アドバンスド・レプリケーション（非同期）設定順序

以下の手順に従って、
マスター定義サイトとマスターサイトの設定を実施します。

1. 初期化パラメータの設定
2. ネットワーク定義
3. マスター定義サイトの設定
4. マスターサイトの設定
5. レプリケーショングループの設定（マスター定義サイト）
6. レプリケーションの開始

１．初期化パラメータの設定

Master定義

Master

GLOBAL_NAMESをTRUEに設定（デフォルトFALSE）

```
ALTER SYSTEM SET GLOBAL_NAMES = TRUE;
```

その他のパラメータはデフォルトのまま

SHOW PARAMETERS コマンドで確認可能

```
JOB_QUEUE_PROCESSES                  10
OPEN_LINKS                           4
REPLICATION_DEPENDENCY_TRACKING      TRUE
```

初期化パラメータの説明

■ GLOBAL_NAMES

データベースリンクの名前を接続先データベースのグローバルデータベース名と同じにするかを決定するパラメータです。デフォルトはFALSEになっているのでTRUEに変更します。FALSEでもレプリケーション環境を作成するのに問題はありませんが運用、管理などを考えた場合にデータベースリンク名と接続先データベースの名前を同一にすることを強く推奨します。

■ JOB_QUEUE_PROCESSES

ジョブ・キュー・プロセスの数を指定するパラメータです。遅延トランザクションの伝播や削除などはジョブとして管理されていて。それらの要求を処理するプロセスがジョブ・キュー・プロセスです。0に指定した場合は、伝播や削除は手動で実行する必要があります。

■ OPEN_LINKS

1つのセッションで同時にオープンするリモートのデータベースに対する接続の最大数を指定します。同期の伝播先がn個ある場合は、nに設定する必要があります。デフォルトでは4なので同期での伝播先が4個以下であれば変更する必要はありません。

■ REPLICATION_DEPENDENCY_TRACKING

データベースに対する読取り/ 書込み操作の依存性の追跡を使用可能または使用禁止にします。依存性の追跡は、レプリケーション環境で変更をパラレル伝播するときは必須です。

＜参考＞グローバルデータベース名

Master定義

Master

※通常、変更の必要はありません

グローバル・データベース名の確認

```
SQL> CONNECT / AS SYSDBA ( sysユーザで接続 )
SQL> SELECT * FROM GLOBAL_NAME;

GLOBAL_NAME
--------------------------------------------------------------------
ADVREPLI. DIRECT21.JP.ORACLE.COM
```

グローバル・データベース名の変更

通常は変更必要なし

```
SQL> ALTER DATABASE RENAME GLOBAL_NAME TO
        ADVREPLI. DIRECT21.JP.ORACLE.COM;
```

＜参考＞グローバルデータベース名の説明

- グローバルデータベース名（GLOBAL_NAME）
  - 複数DBがある環境でDBを一意に識別する名前
  - デフォルトではdb_name.db_domainに設定される
  - ディクショナリで確認できる
  - コマンドで設定できる

２．ネットワーク定義

Master定義

１．マスター定義サイトからマスターサイトへの接続が可能なように、tnsnames.ora に接続文字列を追加します。

```
# tnsnames.ora Network Configuration File:
    /u01/app/oracle/product/10.2.0/db_1/network/admin/tnsnames.ora

ADVREPLI.DIRECT22.JP.ORACLE.COM =
 (DESCRIPTION =
   (ADDRESS = (PROTOCOL = TCP)(HOST = direct22.jp.oracle.com)(PORT = 1521))
   (CONNECT_DATA =
     (SERVER = DEDICATED)
     (SERVICE_NAME = advrepli.direct22.jp.oracle.com)
   )
 )
```

マスターサイトへの接続文字列

２．マスターサイトからマスター定義サイトへの接続が可能なように、tnsnames.ora に接続文字列を追加します。

```
# tnsnames.ora Network Configuration File:
    /u01/app/oracle/product/10.2.0/db_1/network/admin/tnsnames.ora

ADVREPLI.DIRECT21.JP.ORACLE.COM =
 (DESCRIPTION =
   (ADDRESS = (PROTOCOL = TCP)(HOST = direct21.jp.oracle.com)(PORT = 1521))
   (CONNECT_DATA =
     (SERVER = DEDICATED)
     (SERVICE_NAME = advrepli.direct21.jp.oracle.com)
   )
 )
```

マスター定義サイトへの接続文字列

３マスターサイト定義サイトの設定

Master定義

１．レプリケーション管理者を作成し、権限を付与

```
CONNECT system/XXXXX

CREATE USER repadmin IDENTIFIED BY XXXXX;

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

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

GRANT SELECT ANY DICTIONARY TO repadmin;
```

レプリケーション環境を構築、管理するための権限

レプリケーション・マネージメント・ツールの接続権限

２．プロパゲータを登録

レプリケーション管理者を伝播者に設定
（遅延トランザクションをマスターサイトへ伝播する）

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

■**username**
　伝播者の権限を付与するユーザーの名前です。

３．受信者を登録

レプリケーション管理者を受信者に設定
（マスターサイトから伝播された遅延トランザクションを受信する）

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

■**username**
受信者の権限を付与するユーザーの名前です。
■**privilege_type**
割り当てる権限のタイプを指定します。
・受信者 = receiver
■**list_of_gnames**
ユーザーに付与した受信者権限の適用対象にするレプリケーション・グループの、カンマ区切りのリストです。NULLに設定すると、このプロシージャのコール時にはまだ確認されていないレプリケーション・グループを含め、すべてのレプリケーション・グループが受信者権限の適用対象になります。

４．パージプロセス・スケジュールの作成

正常に伝播されたトランザクションエントリを削除

```
CONNECT repadmin/XXXXX

BEGIN
  DBMS_DEFER_SYS.SCHEDULE_PURGE (
    next_date => SYSDATE,
    interval => 'SYSDATE + 1/1440',
    delay_seconds => 0);
END;
/
```

ここでは1分に設定
1／24(H)×60(M)

■**next_date**
次にパージを実行する日付です。デフォルトはSYSDATEです。

■**interval**
次にパージを実行する日付を計算する計算式です。一定間隔でパージを実行するときには必ず指定します。

■**delay_seconds**
delay_secondsの間、遅延トランザクション・キューにパージするトランザクションがなくなると、パージが完全に停止します。

５．パブリック・データベース・リンクを作成

```
CONNECT system/XXXXX

CREATE PUBLIC DATABASE LINK advrepli.direct22.jp.oracle.com
USING ‘advrepli.direct22.jp.oracle.com';
```

６．レプリケーション管理者のデータベース・リンクを作成

マスター定義サイトのレプリケーション管理者から、マスターサイトのレプリケーション管理者へのデータベースリンクを作成

```
CONNECT repadmin/XXXXX

CREATE DATABASE LINK advrepli.direct22.jp.oracle.com
  CONNECT TO repadmin IDENTIFIED BY XXXXX;
```

7．スケジュール・プッシュを設定

遅延トランザクションを伝播するジョブ

```
CONNECT repadmin/XXXXX

BEGIN
  DBMS_DEFER_SYS.SCHEDULE_PUSH (
    destination => ‘advrepli.direct22.jp.oracle.com',
    interval => 'SYSDATE + 1/8640',
    next_date => SYSDATE);
END;
/
```

ここでは10秒に設定
10 ／ 24(H)×60(M) ×60(S)

■destination
　変更を転送するマスターサイトのデータベース名です。
■interval
　次にプッシュを実行する日付を計算する計算式です。一定間隔でプッシュを実行するときには必ず指定します。
■next_date
　次にプッシュを実行する日付です。デフォルトはSYSDATEです。

３．マスターサイトの設定

Master

１．レプリケーション管理者を作成し、権限を付与

```
CONNECT system/XXXXX

CREATE USER repadmin IDENTIFIED BY XXXXX;

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

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

GRANT SELECT ANY DICTIONARY TO repadmin;
```

レプリケーション環境を構築、管理するための権限

レプリケーション・マネージメント・ツールの接続権限

２．プロパゲータを登録

レプリケーション管理者を伝播者に設定
（遅延トランザクションをマスター定義サイトへ伝播する）

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

■**username**
伝播者の権限を付与するユーザーの名前です。

３．受信者を登録

レプリケーション管理者を受信者に設定
（マスター定義サイトから伝播された遅延トランザクションを受信する）

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

■**username**
受信者の権限を付与するユーザーの名前です。
■**privilege_type**
割り当てる権限のタイプを指定します。
・受信者 = receiver
■**list_of_gnames**
ユーザーに付与した受信者権限の適用対象にするレプリケーション・グループの、カンマ区切りのリストです。NULLに設定すると、このプロシージャのコール時にはまだ確認されていないレプリケーション・グループを含め、すべてのレプリケーション・グループが受信者権限の適用対象になります。

４．パージプロセス・スケジュールの作成

正常に伝播されたトランザクションエントリを削除

```
CONNECT repadmin/XXXXX

BEGIN
  DBMS_DEFER_SYS.SCHEDULE_PURGE (
    next_date => SYSDATE,
    interval => 'SYSDATE + 1/1440',
    delay_seconds => 0);
END;
/
```

ここでは1分に設定
1／24(H)×60(M)

■**next_date**
次にパージを実行する日付です。デフォルトはSYSDATEです。

■**interval**
次にパージを実行する日付を計算する計算式です。一定間隔でパージを実行するときには必ず指定します。

■**delay_seconds**
delay_secondsの間、遅延トランザクション・キューにパージするトランザクションがなくなると、パージが完全に停止します。

５．パブリック・データベース・リンクを作成

```
CONNECT system/XXXXX

CREATE PUBLIC DATABASE LINK advrepli.direct21.jp.oracle.com
USING ‘advrepli.direct21.jp.oracle.com';
```

６．レプリケーション管理者のデータベース・リンクを作成

マスターサイトのレプリケーション管理者から、マスター定義サイトのレプリケーション管理者へのデータベースリンクを作成

```
CONNECT repadmin/XXXXX

CREATE DATABASE LINK advrepli.direct21.jp.oracle.com
  CONNECT TO repadmin IDENTIFIED BY XXXXX;
```

7．スケジュール・プッシュを設定

遅延トランザクションを伝播するジョブ

```
CONNECT repadmin/XXXXX

BEGIN
  DBMS_DEFER_SYS.SCHEDULE_PUSH (
    destination => ‘advrepli.direct21.jp.oracle.com',
    interval => 'SYSDATE + 1/8640',
    next_date => SYSDATE);
END;
/
```

ここでは10秒に設定
10 ／ 24(H)×60(M) ×60(S)

■destination
　変更を転送するマスターサイトのデータベース名です。
■interval
　次にプッシュを実行する日付を計算する計算式です。一定間隔でプッシュを実行するときには必ず指定します。
■next_date
　次にプッシュを実行する日付です。デフォルトはSYSDATEです。

４ レプリケーショングループの設定

Master定義

１． レプリケーショングループを作成

```
CONNECT repadmin/XXXXX

BEGIN
  DBMS_REPCAT.CREATE_MASTER_REPGROUP (
    gname => 'rep_group');
END;
/
```

■**gname**
　レプリケーショングループの名前です。

２．レプリケーショングループにオブジェクトを追加

```
BEGIN
  DBMS_REPCAT.CREATE_MASTER_REPOBJECT (
    gname => 'rep_group',
    type => 'TABLE',
    oname => 'emp',
    sname => ‘scott');
END;
/
```

‘TABLE’タイプを指定

今回は、サンプルスキーマである scott の EMP表を使用

■type
　レプリケートするオブジェクトのタイプです。サポートされているタイプは次のとおりです。
　FUNCTION , SYNONYM , INDEX , TABLE , INDEXTYPE , TRIGGER , OPERATOR , TYPE , PACKAGE , TYPE BODY , PACKAGE BODY , VIEW , PROCEDURE

■sname
　オブジェクトが置かれたスキーマの名前です。

■oname
　レプリケーション・サポートを生成するオブジェクトの名前です。

３．マスターサイトの追加

レプリケーション環境に他のマスター・サイトを追加

```
BEGIN
  DBMS_REPCAT.ADD_MASTER_DATABASE (
    gname => 'rep_group',
    master => ‘replica.direct22.jp.oracle.com');
END;
/
```

■gname
　レプリケートされるレプリケーション・グループの名前です。このレプリケーション・グループは、マスター定義サイトに存在している必要があります。
■master
　追加するマスターサイトのデータベース名です。

４．レプリケーション・サポートの作成

レプリケーションに必要なパッケージ等を作成

```
BEGIN
  DBMS_REPCAT.GENERATE_REPLICATION_SUPPORT (
    sname => ‘scott',
    oname => 'emp',
    type => 'TABLE');
END;
/
```

■**sname**
　オブジェクトが置かれたスキーマの名前です。
■**oname**
　レプリケーション・サポートを生成するオブジェクトの名前です。
■**type**
　オブジェクトのタイプです。サポートされているタイプは、TABLE、PACKAGEおよびPACKAGE BODYです。

８．レプリケーションの開始

Master定義

１．指定したマスター・グループのレプリケーションを

```
CONNECT repadmin/XXXXX

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

EXECUTE DBMS_REPCAT.DO_DEFERRED_REPCAT_ADMIN(gname =>
'rep_group');

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

DBA_REPCATLOG ビューにエントリがないことを確認

エントリが残っている場合、マスター定義サイト/マスターサイトの両方でDO_DEFERRED_REPCAT_ADMIN プロシージャを実行

レプリケーションを開始

OTNセミナーオンデマンド とは？

100種類以上の録画セミナーから自分のペースで受講する

ORACLE TECHNOLOGY NETWORK
OTNセミナー オンデマンド

録画されたセミナーの無償ダウンロードサービスです。

- ✓ 毎月旬なトピックの新作コンテンツを追加
- ✓ ダイセミでおなじみの講師陣（オラクルエンジニア）が続々登場
- ✓ MP4形式での提供により、スマートフォンで通勤中にも聴講可能

スマホでもみられる！

毎月新作が登場！

OTNセミナーオンデマンド一覧 はこちら
http://www.oracle.com/technetwork/jp/ondemand/index.html
オススメ＆新作コンテンツ情報 はこちら
http://oracletech.jp/seminar/recommended/

毎月チェック！

OTNオンデマンド　検索

Oracle エンジニアのための技術情報サイト

オラクルエンジニア通信

http://blogs.oracle.com/oracle4engineer/

twitter

最新情報つぶやき中

@oracletechnetjp

- **技術資料が見つかる！**
  - キーワード検索、レベル別、カテゴリ別、製品・機能別
- **コラムでなるほど！！**
  - オラクル製品に関する技術コラムを毎週お届け
  - 決してニッチではなく、誰もが明日から使える技術の「あ、そうだったんだ！」をお届け

オラクルエンジニア通信

以上の事項は、弊社の一般的な製品の方向性に関する概要を説明するものです。また、情報提供を唯一の目的とするものであり、いかなる契約にも組み込むことはできません。以下の事項は、マテリアルやコード、機能を提供することをコミットメント（確約）するものではないため、購買決定を行う際の判断材料になさらないで下さい。オラクル製品に関して記載されている機能の開発、リリースおよび時期については、弊社の裁量により決定されます。

Hardware and Software
Engineered to Work Together


ORACLE®