

ORACLE®

ORACLE®

Oracle Database 12c Release 1

マルチテナント・アーキテクチャ

日本オラクル株式会社
早坂 真由美

ORACLE®
DATABASE 12^c



Plug into the **Cloud**.

以下の事項は、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。

OracleとJavaは、Oracle Corporation 及びその子会社、関連会社の米国及びその他の国における登録商標です。文中の社名、商品名等は各社の商標または登録商標である場合があります。

Program Agenda

- Oracle Database 12c で実現する次世代データベース
- マルチテナント・アーキテクチャの概要
- プラガブル・データベースの基本操作
- プラガブル・データベースの管理

Oracle Database 12c で 実現する次世代データベース

ORACLE[®] 12^c
DATABASE



Plug into the **Cloud.**

ORACLE[®]

データベース統合におけるチャレンジ

サーバー統合による
ITコストの削減

データベース数の
削減

アプリケーション
の独立性は維持、
変更は不要

これまでのデータベースにおけるアーキテクチャ

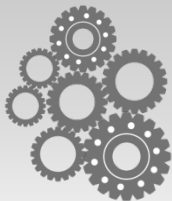
メモリ、プロセス、データファイルによって構成

顧客管理 DB

メモリ
使用率

バックグラウンド
・プロセス

データ
ファイル



これまでのデータベースにおけるアーキテクチャ

データベースごとに専用のメモリやプロセスが必要

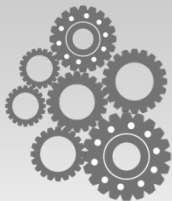
- 企業によっては、数百のデータベースがあるケースも稀ではない

顧客管理 DB

メモリ
使用率

バックグラウンド
・プロセス

データ
ファイル



販売管理 DB

メモリ
使用率

バックグラウンド
・プロセス

データ
ファイル



経理 DB

メモリ
使用率

バックグラウンド
・プロセス

データ
ファイル



マルチテナントを実現する新しいアーキテクチャ

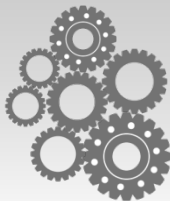
Oracle Database 12c で実現する複数データベースの統合

顧客管理 DB

メモリ
使用率

バックグラウンド
・プロセス

データ
ファイル



販売管理 DB

メモリ
使用率

バックグラウンド
・プロセス

データ
ファイル

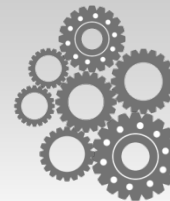


経理 DB

メモリ
使用率

バックグラウンド
・プロセス

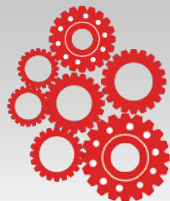
データ
ファイル



マルチテナントを実現する新しいアーキテクチャ

Oracle Database 12c で実現する複数データベースの統合

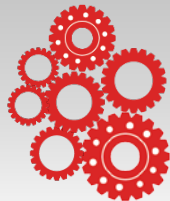
統合 DB



マルチテナントを実現する新しいアーキテクチャ

Oracle Database 12c で実現する複数データベースの統合

マルチテナント・コンテナ・データベース (CDB)



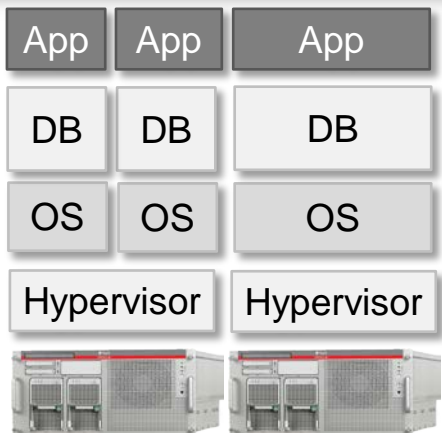
プラグブル・データベース
(PDB)

データベース・クラウドのアーキテクチャ

Oracle Database 12c で実現する新たなアプローチ

インフラストラクチャ・クラウド

データベース・クラウド



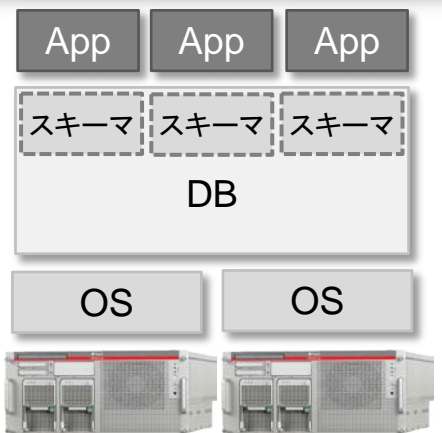
Server

- サーバー仮想化



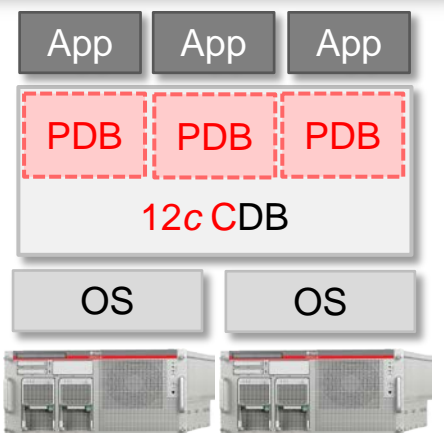
Operating System

- Real Application Clusters (RAC)



Database

- データベース・スキーマ



New
Multitenant
Architecture

マルチテナント・アーキテクチャによる統合

マルチテナントをデータベースのレイヤーで実現

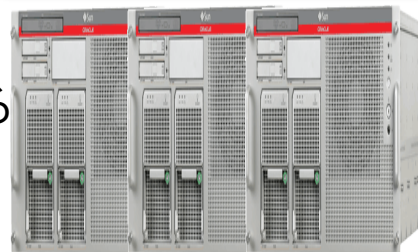
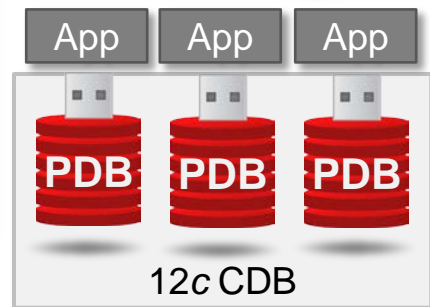
- 複数の顧客データを単一のデータベースへ統合
- サーバーや運用管理にかかるコストを削減
 - クラウド環境を利用したい企業
 - アプリケーションごとにデータベースを分けて運用している企業

プラグブル・データベース (PDB)

- Oracle データベース内における論理的なセット
- ユーザーやアプリケーションからは通常のデータベースと同様に扱える

マルチテナント・コンテナ・データベース (CDB)

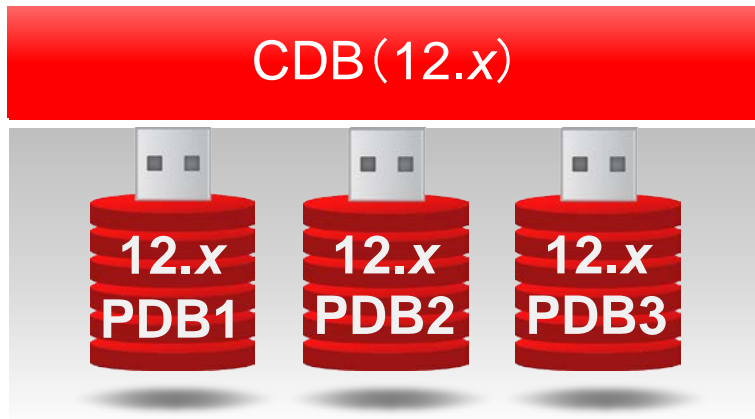
- マルチテナント・アーキテクチャを実現するデータベース



アップグレードあるいはパッチ適用作業の簡略化

1回の適用ですべてのプラガブル・データベースに対応

アップグレード →



必要に応じたデータベースの plug と unplug

データベースのアップグレード、あるいはパッチ適用時にも活用可能

CDB1 (12.1)



CDB2 (12.x)



テスト環境、開発環境のためのクローニング

速攻性と柔軟性に優れたコピーやスナップショット

CDB1 (本番環境)

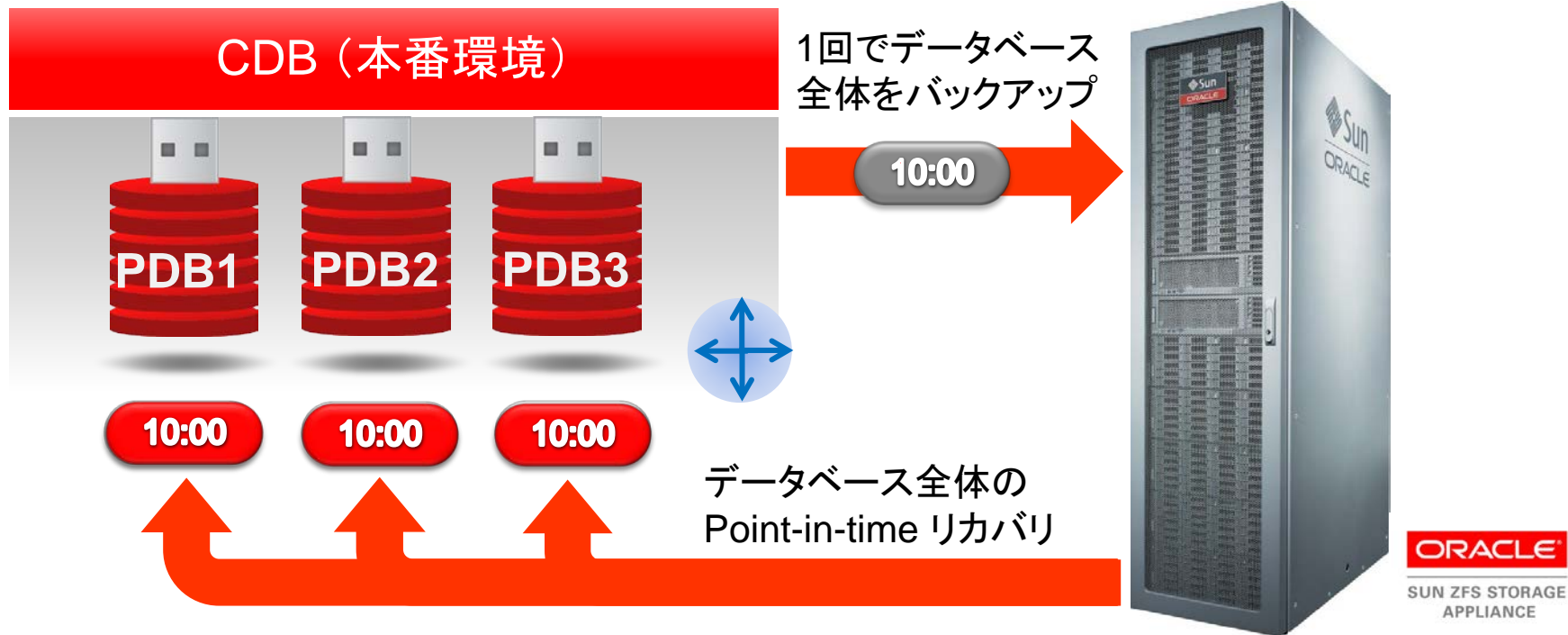


CDB2 (開発環境)



バックアップ・リカバリ時の管理性

容易なバックアップと柔軟なリカバリ



ORACLE
SUN ZFS STORAGE
APPLIANCE

ORACLE

Real Application Clusters (RAC) 構成時の管理性

プラグブル・データベースを含めてデータベース全体の可用性を提供

RAC 構成の CDB



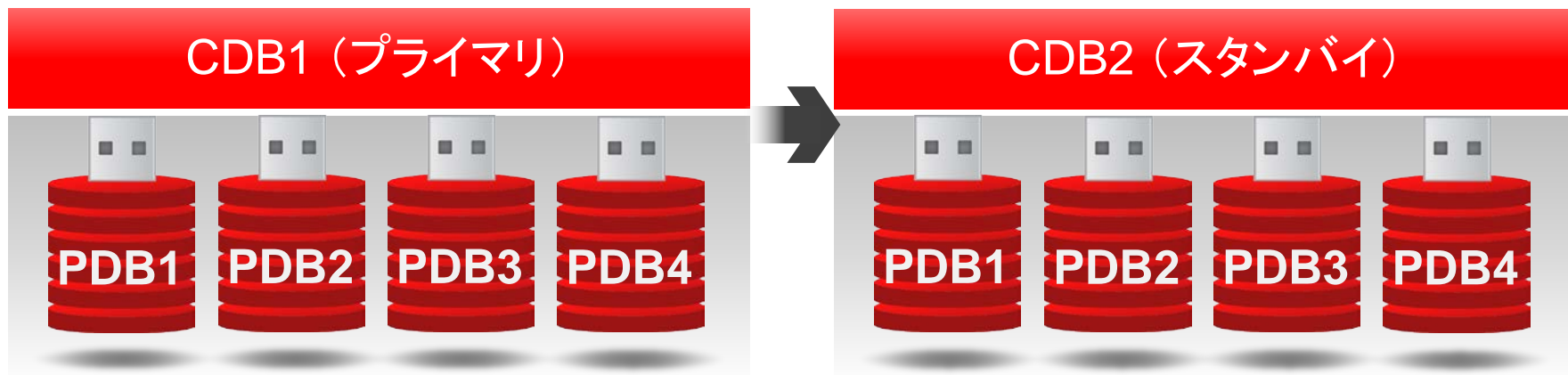
サーバー1



サーバー2

スタンバイ構成時の管理性

データベース単位で構成し、すべてのプラガブル・データベースをカバー



マルチテナント・ アーキテクチャの概要

ORACLE[®] 12^c
DATABASE



Plug into the **Cloud.**

ORACLE[®]

Oracle データベースの拡張

マルチテナント・アーキテクチャへの対応

- Oracle Database 12c では従来のデータベースに加え、マルチテナント・アーキテクチャに対応するデータベースを提供する

Non-CDB

- Oracle Database 11g Release 2 以前から提供されているタイプ
- 従来のアーキテクチャに対応するデータベース

マルチテナント・コンテナ・データベース (CDB) New

- Oracle Database 12c Release 1 より提供されるタイプ
- マルチテナント・アーキテクチャに対応したデータベース
- Oracle データベースとしての基本的な構造は non-CDB と同じ

マルチテナント・アーキテクチャへの対応

データベースの構成タイプに応じた対応状況

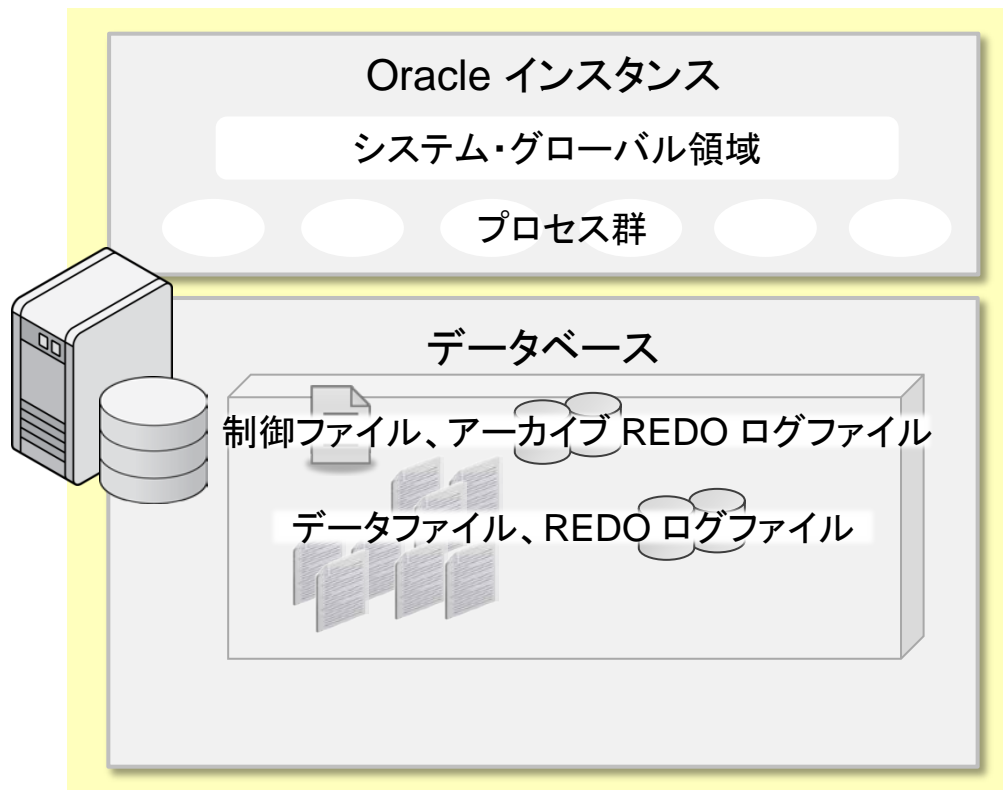
データベースの構成タイプ	Non-CDB	CDB
シングル・インスタンス (Oracle Restart 構成を含む)	○ 対応	○ 対応
Oracle Real Application Clusters (ポリシー管理 / 管理者管理を含む)	○ 対応	○ 対応
Oracle RAC One Node (ポリシー管理 / 管理者管理を含む)	○ 対応	○ 対応
Oracle Data Guard (フィジカル / ロジカル・スタンバイを含む)	○ 対応	○ 対応

 いずれのデータベース構成においても non-CDB や CDB が利用可能

従来からのデータベース

Non-CDB

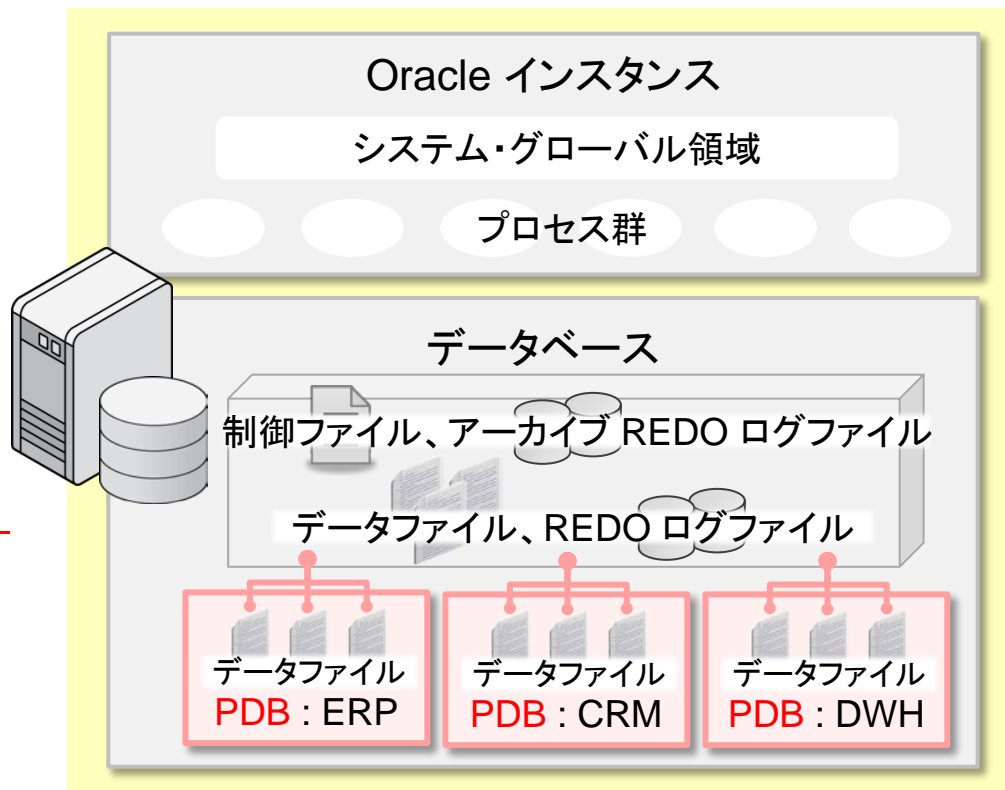
- Oracle データベースは、サーバー上で確保されるメモリ領域とプロセス群から構成される Oracle インスタンスと、ストレージに格納される データベース (ファイル群) により構成される
- この構造を拡張し、Oracle Database 12c では、マルチテナント・アーキテクチャに対応したデータベースを提供



マルチテナント・アーキテクチャ対応のデータベース

CDB

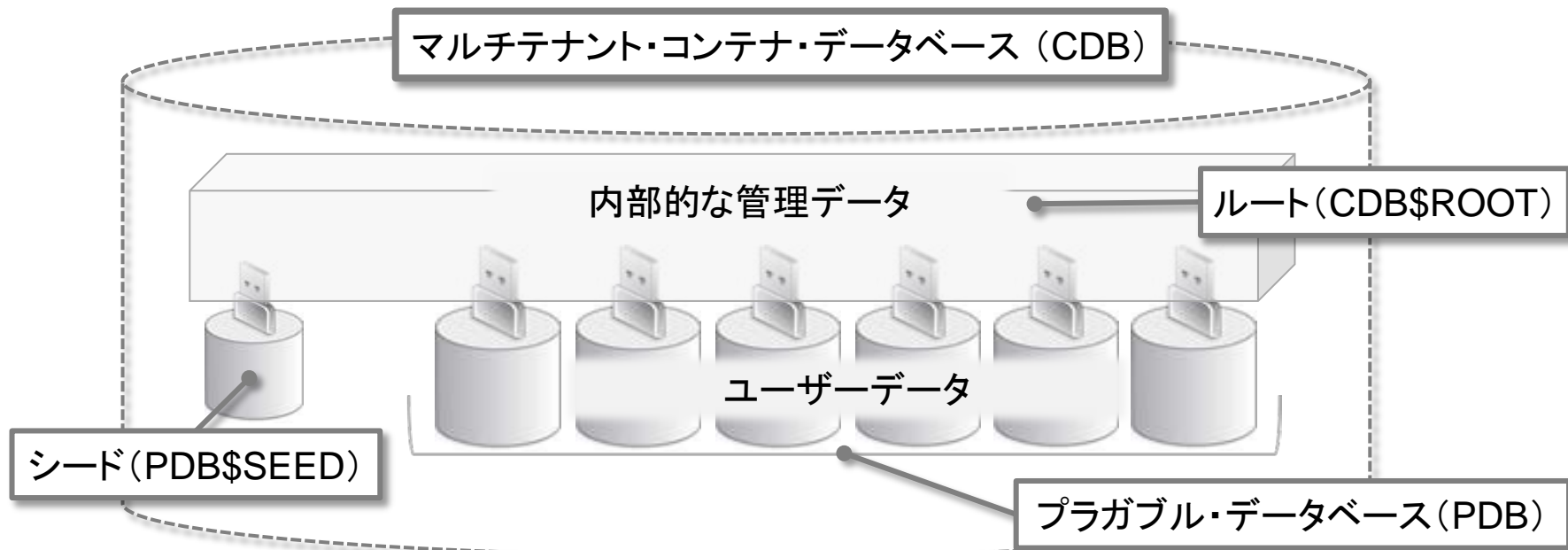
- CDB では、複数の PDB を作成可能
- プロセスやメモリ、メタデータは、すべての PDB 間で共有する
- 各 PDB に固有の部分は、共有化する部分とは分離される
- 同一あるいは異なる CDB 間で、容易に PDB の取り付けや取り外しが可能



マルチテナント・コンテナ・データベース (CDB)

構成要素

- CDB は、コンテナとよばれる要素で構成される

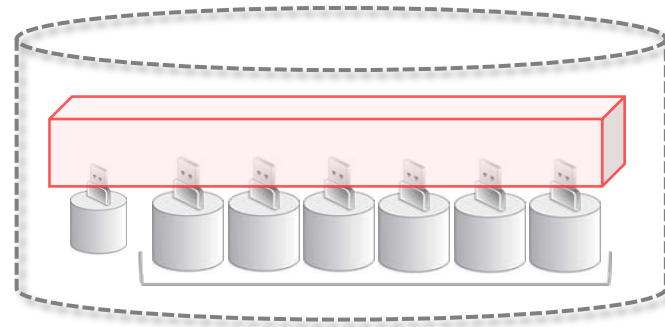


ルート (CDB\$ROOT)

データベースの根幹

ルート (CDB\$ROOT)

- データベース全体で共有するオブジェクトやメタデータを含む
 - Oracle データベースによって提供されるスキーマ
 - ユーザーが作成するスキーマ
- データベース作成時に1つのみ作成される
- オブジェクトの追加や変更は不可
- ルートのデータ・ディクショナリにはデータベース全体で共有する情報として、付属するすべての PDB の情報が含まれる

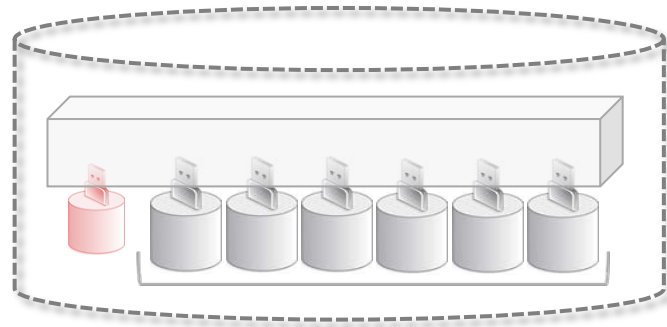


シード (PDB\$SEED)

プラガブル・データベース作成時のテンプレート

シード (PDB\$SEED)

- PDB を新規作成する際に使用するテンプレート
 - Database Configuration Assistant (DBCA) によるデータベース作成時や作成後に、PDB を作成する場合
 - データベース作成後に、手動で PDB を作成する場合
- データベース作成時に1つのみ作成される
 - 読み取り専用
- オブジェクトの追加や変更は不可

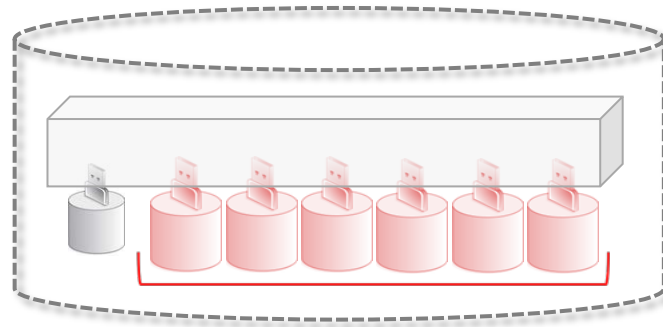


プラグابل・データベース (PDB)

スキーマやオブジェクトの集合

プラグابل・データベース (PDB)

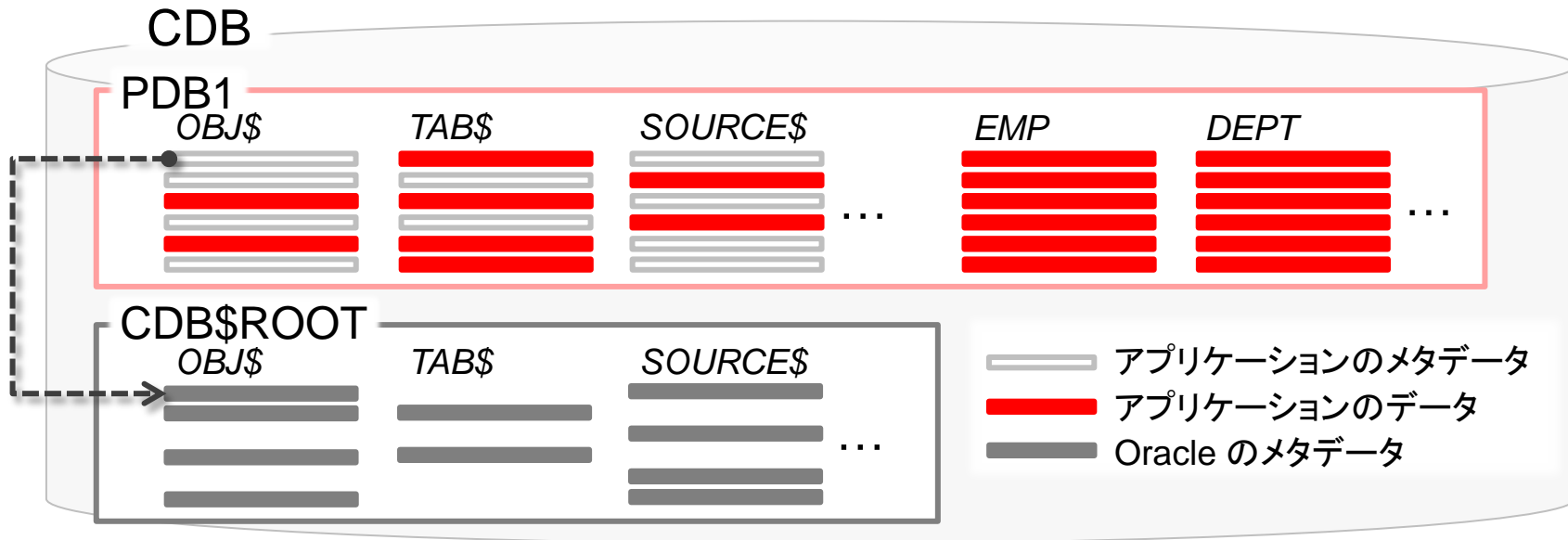
- スキーマや表領域を含む論理的なセット
- 同一 CDB 内に複数作成することが可能 (シードを除いて最大 252 まで)
- PDB 間は排他的な関係にあり、データは論理的に分離される
- CDB から取り外した後、再度取り付けたり、異なる CDB へ取り付けることが可能



データ・ディクショナリ

マルチテナント・アーキテクチャにおけるデータ・ディクショナリの構造

- Oracle に関するメタデータは CDB\$ROOT で保持する
- アプリケーションに関するメタデータは各 PDB で保持する



データ・ディクショナリ

オブジェクトの共有と分離

PDB_SALES

SYSTEM 表領域

SYS.OBJ\$ 表

NAME	TYPE#
...	...
PREF	2
AUDIT_ACTIONS	2

EXAMPLE 表領域

SALES.PREF 表

P_ID	P_NAME
1	北海道
2	...

PDB_HR

SYSTEM 表領域

SYS.OBJ\$ 表

NAME	TYPE#
...	...
EMP	2
AUDIT_ACTIONS	2

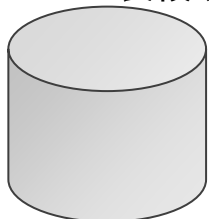
USERS 表領域

HR.EMP 表

ID	L_NAME
...	...
100	早坂
101	...

CDB\$ROOT

SYSTEM 表領域



SYS.OBJ\$ 表

NAME	TYPE#
TAB\$	2
...	...
AUDIT_ACTIONS	2

データ・ディクショナリ

オブジェクトの共有と分離

PDB_SALES

SYSTEM 表領域

SYS.OBJ\$ 表

NAME	TYPE#
...	...
PREF	2
AUDIT_ACTIONS	2

EXAMPLE 表領域

SALES.PREF 表

P_ID	P_NAME
1	北海道
2	...

PDB_HR

SYSTEM 表領域

SYS.OBJ\$ 表

NAME	TYPE#
...	...
EMP	2
AUDIT_ACTIONS	2

USERS 表領域

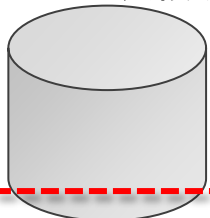
HR.EMP 表

ID	L_NAME
...	...
100	早坂
101	...

CDB\$ROOT

SYSTEM 表領域

SYS.OBJ\$ 表



NAME	TYPE#
TAB\$	2
...	...
AUDIT_ACTIONS	2

- Oracle 固有のオブジェクトの実体は、CDB\$ROOT に格納される
- すべての PDB からアクセスが可能

データ・ディクショナリ

オブジェクトの共有と分離

PDB_SALES

SYSTEM 表領域

SYS.OBJ\$ 表

NAME	TYPE#
...	...
PREF	2
AUDIT_ACTIONS	2

EXAMPLE 表領域

SALES.PREF 表

P_ID	P_NAME
1	北海道
2	...

PDB_HR

SYSTEM 表領域

SYS.OBJ\$ 表

NAME	TYPE#
...	...
EMP	2
AUDIT_ACTIONS	2

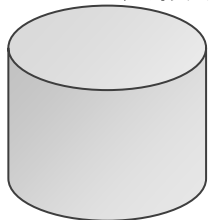
USERS 表領域

HR.EMP 表

ID	L_NAME
100	早坂
101	...

CDB\$ROOT

SYSTEM 表領域



SYS.OBJ\$ 表

NAME	TYPE#
TAB\$	2
...	...
AUDIT_ACTIONS	2

- PDB で作成されたオブジェクトは、そのPDB に固有のオブジェクトとして格納される

データ・ディクショナリ・ビュー

データ・ディクショナリ・ビューの種類

- データ・ディクショナリ・ビューの種類には、次の 4 つがある

CDB_xxx

- ルートおよび全 PDB 内のオブジェクトに関する情報

New

DBA_xxx

- ルートあるいは PDB 内のオブジェクトに関する情報

ALL_xxx

- ユーザーがアクセス可能なオブジェクトに関する情報

USER_xxx

- ユーザーが所有するオブジェクトに関する情報

データ・ディクショナリ・ビューの例 (CDB_XXX)

CDB_TABLESPACES ビュー

- 接頭辞が CDB_ であるビューには、すべてのコンテナの情報が含まれる
 - ルートおよび、すべての PDB の情報を確認することが可能

```
SQL> SELECT TABLESPACE_NAME, STATUS, CON_ID FROM CDB_TABLESPACES;
```

TABLESPACE_NAME	STATUS	CON_ID
-----	-----	-----
SYSTEM	ONLINE	1
SYSAUX	ONLINE	1
UNDOTBS1	ONLINE	1
TEMP	ONLINE	1
USERS	ONLINE	1
SYSTEM	ONLINE	2
SYSAUX	ONLINE	2
TEMP	ONLINE	2

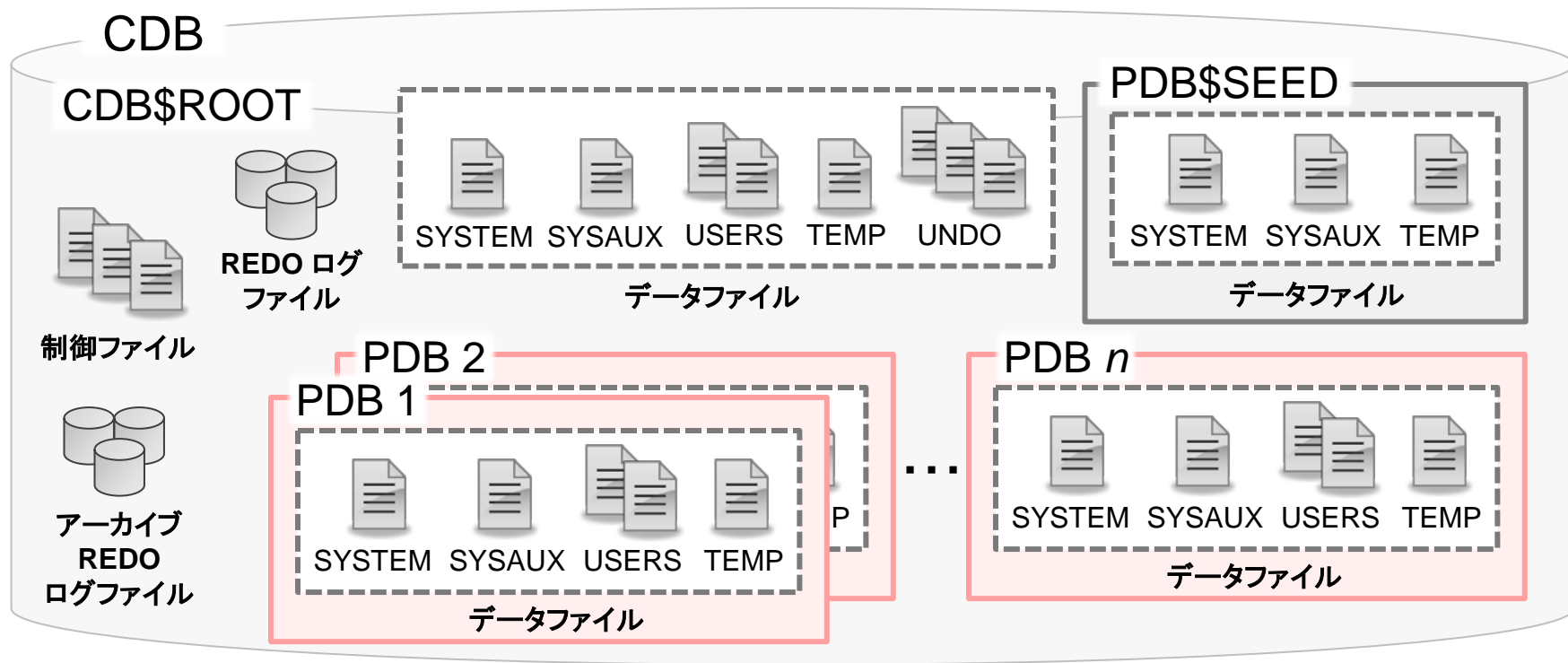
ルートの表領域

PDB の表領域

8行が選択されました。

マルチテナント・コンテナ・データベースの物理構造

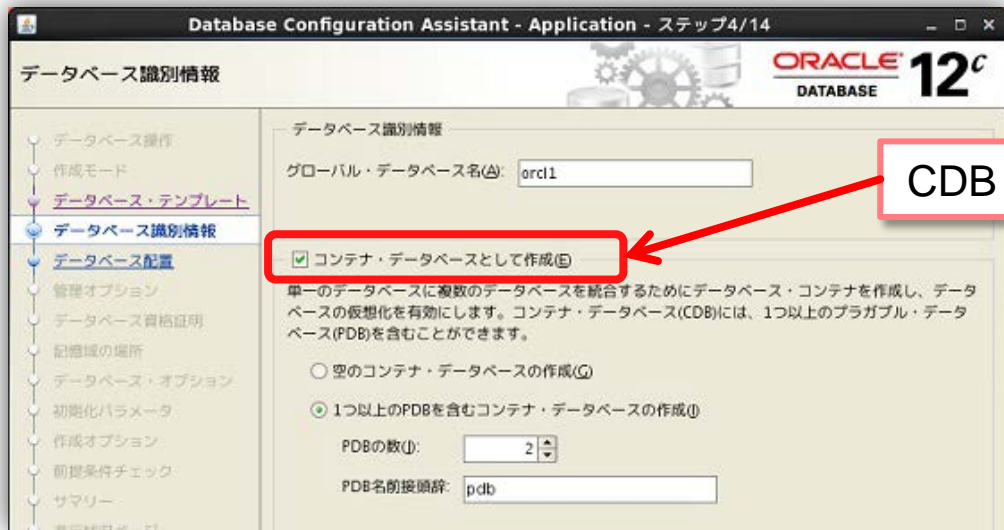
データベース関連ファイル



マルチテナント・コンテナ・データベースの作成

Database Configuration Assistant (DBCA) を使用する場合の例

- 基本的には、non-CDB の作成と手順は同じ
 - 「データベース識別情報」画面の抜粋

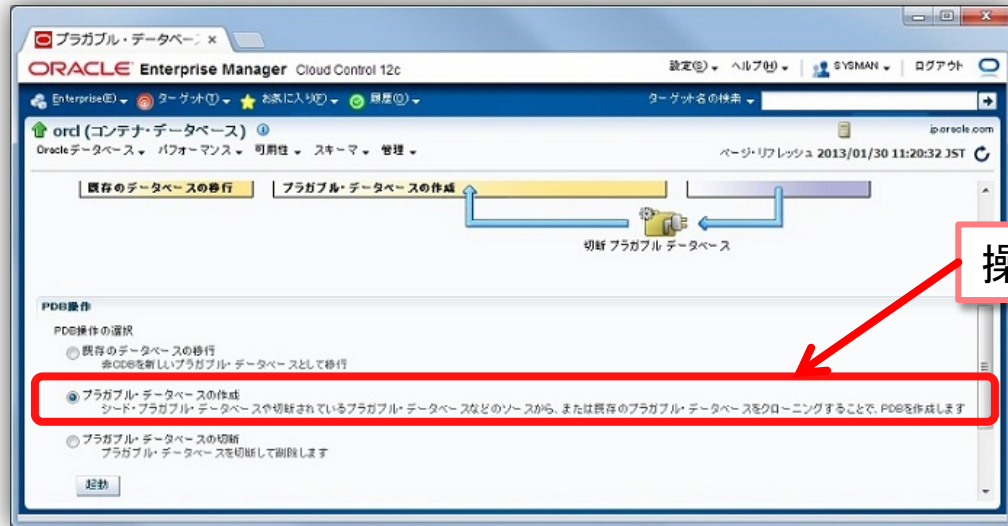


CDB として作成する場合に選択

プラグブル・データベースの作成

Enterprise Manager Cloud Control (EMCC) を使用する場合の例

- DBCA などの GUI ツール、あるいは SQL 文でも作成が可能
 - 「プラグブル・データベースのプロビジョニング」画面の抜粋



操作メニューから作成を選択

Oracle Enterprise Manager による管理

統合管理を可能にする GUI ツールの提供

- Oracle Enterprise Manager によるデータベースの管理

Oracle Enterprise Manager Database Express (EM Database Express)

- 単一のデータベースを管理 (Oracle RAC や RAC One Node も対応)
- データベースと同一サーバー上に構成して使用する

Oracle Enterprise Manager Cloud Control (EM Cloud Control)

- 複数のサーバーを一元的に管理 (統合管理ツール)
- Cloud Control 用に用意したサーバー上に構成して使用する
- 管理対象のサーバーにエージェントを構成して使用する
- データベース管理のすべての機能を提供
- データベース以外にも、オラクル製品を中心に多彩な機能を提供

プラグブル・データベースの 基本操作



データベースにおける接続

SQL*Plus を使用した接続方法

- これまでのデータベースでも提供されていた SQL*Plus を使用した接続方法
 - 接続記述子

```
sqlplus <USERNAME>/<PASSWORD>@<ALIAS>
```

```
<ALIAS>=
(DESCRIPTION =
  (ADDRESS=(PROTOCOL=TCP) (HOST=<HOSTNAME1>) (PORT=<PORT>))
  (ADDRESS=(PROTOCOL=TCP) (HOST=<HOSTNAME2>) (PORT=<PORT>))
  (CONNECT_DATA=(SERVICE_NAME=<SERVICE1>))
)
```

tnsnames.ora

- EZCONNECT (簡易接続ネーミング)

```
sqlplus <USERNAME>/<PASSWORD>@<HOSTNAME>:<PORT>/<SERVICE_NAME>
```

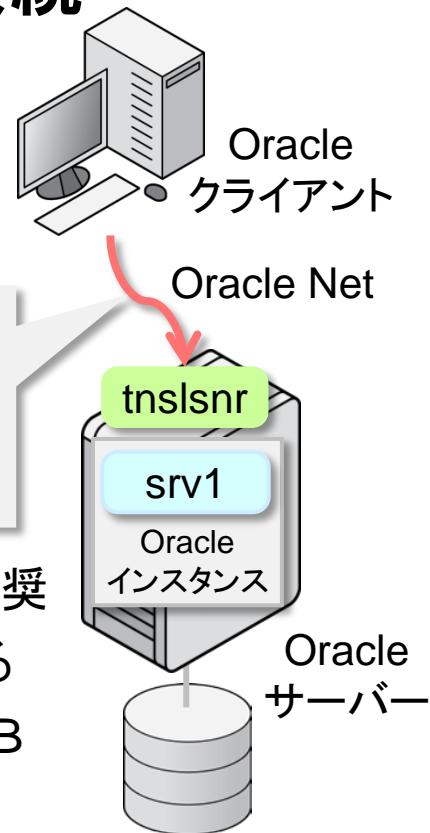

マルチテナント・アーキテクチャにおける接続

サービスを使用した従来と同様の接続方法

- 基本的には、従来のデータベースと同様の方法で接続
 - Oracle クライアントの接続記述子にはサービス名を指定

```
(DESCRIPTION =  
  (ADDRESS=(PROTOCOL=TCP)  
            (HOST=node01.oracle.jp)(PORT=1521))  
  (CONNECT_DATA=(SERVICE_NAME=svr1))  
)
```

- PDB ごとに必要なサービスを作成して接続に利用することを推奨
 - PDB 作成時に PDB 名と同じ名前のサービスが作成される
 - 同一サーバー内に複数の CDB が存在する環境では、PDB 名の重複（同一名のサービス）が生じる場合があるため



LSNRCTL SERVICES

リスナーに登録されたサービスの例

```
$ lsnrctl services
```

```
LSNRCTL for Linux: Version 12.1.0.1.0 - Production on 09-5月 -2013 13:38:28  
Copyright (c) 1991, 2012, Oracle. All rights reserved.
```

```
(DESCRIPTION=(ADDRESS=(PROTOCOL=TCP)(HOST=node01.oracle.jp)(PORT=1521)))に接続中  
サービスのサマリー...
```

```
サービス"cdb1"には、1件のインスタンスがあります。
```

```
インスタンス"cdb1"、状態READYには、このサービスに対する1件のハンドラがあります...
```

```
<中略>
```

```
サービス"hrpdb"には、1件のインスタンスがあります。
```

```
インスタンス"cdb1"、状態READYには、このサービスに対する1件のハンドラがあります...
```

```
ハンドラ:
```

```
"DEDICATED" 確立:30 拒否:0 状態:ready
```

```
LOCAL SERVER30 拒否:0 状態:ready
```

```
LOCAL SERVER
```

```
コマンドは正常に終了しました。
```

PDB のサービスに関する情報

コンテナ間における接続先の切り替え

再接続あるいは ALTER SESSION 文の使用

- コンテナ（ルートあるいは PDB）間における接続先の切り替えには、次の方法がある
 - SQL*Plus による再接続

```
CONNECT <USERNAME>/<PASSWORD>@<HOSTNAME>:<PORT>/<SERVICE_NAME>
```

- 共通ユーザーおよびローカル・ユーザーで使用可能

- ALTER SESSION 文による接続

```
ALTER SESSION SET CONTAINER = <PDB_NAME>
```

- 共通ユーザーのみ使用可能
- SET CONTAINER 権限が必要

接続先の確認方法

SHOW コマンドによる確認

- 接続しているコンテナは SHOW コマンドなどで確認可能
 - ルートに接続している場合の出力例

```
SQL> SHOW CON_NAME
CON_NAME
-----
CDB$ROOT
```

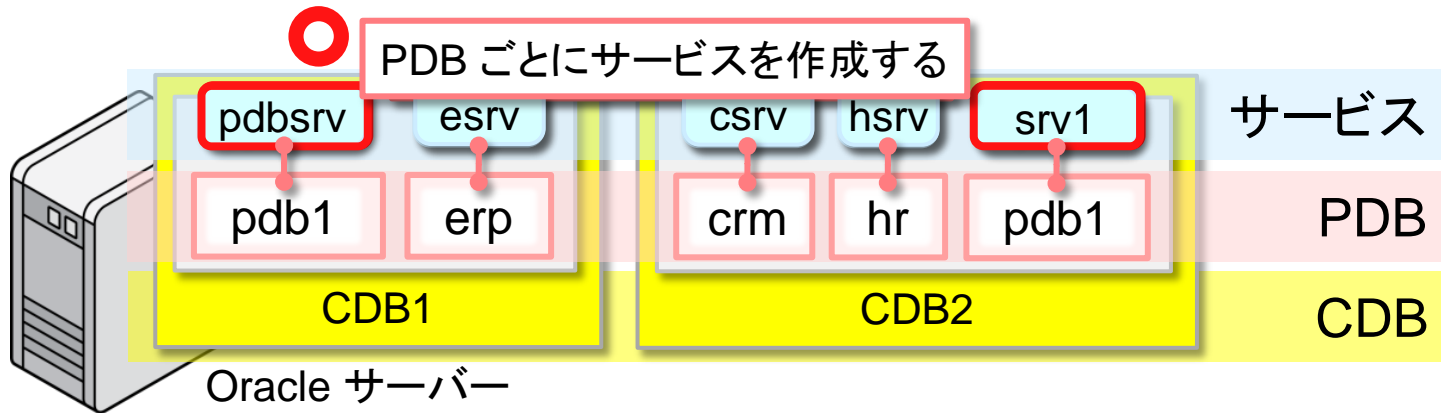
- PDB に接続している場合の出力例

```
SQL> SHOW CON_NAME
CON_NAME
-----
pdb1
```

サービスの使用と管理におけるポイント

一意のサービス名の使用と SRVCTL ユーティリティによる管理

- 統合データベース環境では、PDB ごとに一意のサービスを作成することを推奨
 - デフォルトで作成される PDB 名と同名のサービスは重複する可能性がある



- サービスの管理操作には SRVCTL ユーティリティの使用を推奨
 - DBMS_SERVICE パッケージを使用して管理することも可能

SRVCTL ユーティリティによるサービスの管理

基本操作（作成 / 起動 / 停止）

- サーバー管理（SRVCTL）ユーティリティを使用したサービス管理を推奨

– サービスの作成

```
srvctl add service -db <DBNAME> -service <SERVICE_NAME> -pdb <PDB_NAME>
```

- -PDB でサービスに紐づける PDB を指定する
- ポリシー管理のデータベースでは、-serverpool <SERVERPOOL> の指定が必要

– サービスの起動

```
srvctl start service -db <DBNAME> -service <SERVICE_NAME>
```

– サービスの停止

```
srvctl stop service -db <DBNAME> -service <SERVICE_NAME>
```

SRVCTL ユーティリティによるサービスの管理

サービスの構成情報やリソース属性に関連 PDB の情報を追加

- サービスの構成情報として、関連する PDB の情報が追加
 - SRVCTL ユーティリティによるサービスの構成情報での確認

```
[oracle@node01]$ srvctl config service -db orcl -service srv1
```

```
サービス名: srv1
```

```
サービスは有効です
```

```
サーバー・プール: srvpool1
```

```
... <中略>
```

```
プラグブル・データベース名: erp
```

```
... <省略>
```

- クラスタ管理 (CRSCTL) ユーティリティによるリソースの属性でも確認可能

```
[grid@node01]$ crsctl status resource ora.cdb.srv1.svc -f |grep PLU
```

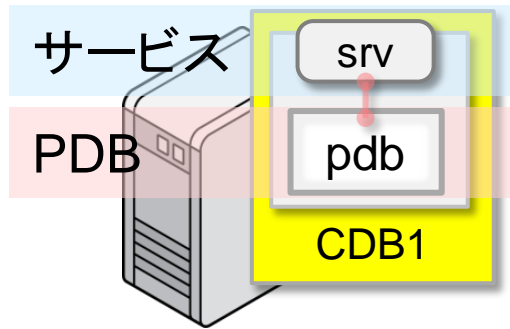
```
PLUGGABLE_DATABASE=pdb1
```

サービスと PDB の依存関係

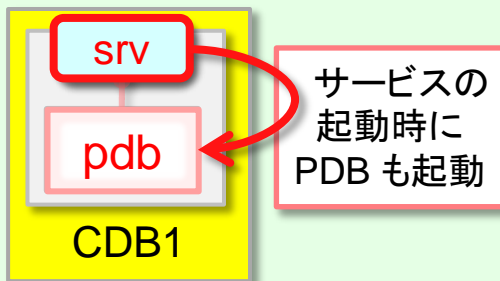
起動・停止における挙動

- サービスを起動する際は、データベースの場合と同様に PDB の起動を試行する
起動時
 - サービスの起動時に PDB が未オープンであればオープンする
 - PDB をオープン時に、停止しているサービスの起動は試行しない

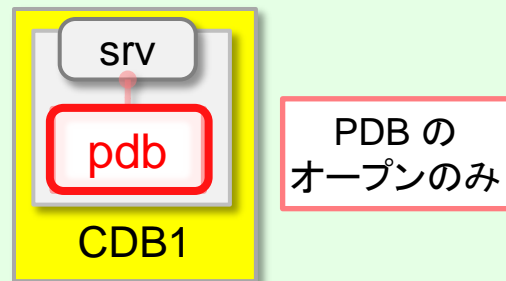
<現在の状態>



A) サービスを起動する場合



B) PDB をオープンする場合



サービスと PDB の依存関係

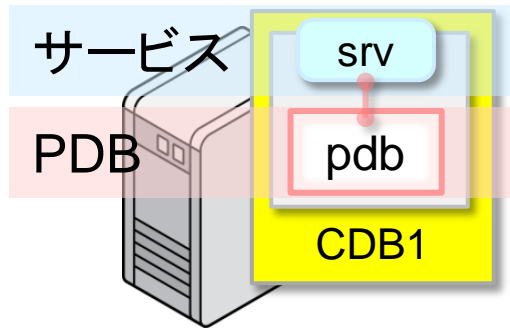
起動・停止における挙動

- サービスを停止する際も、データベースの場合と同様で PDB に影響はない

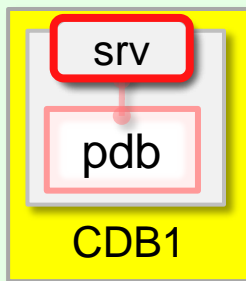
停止時

- サービスの停止時に PDB への操作は行わない
- PDB のクローズ時には、起動しているサービスの停止を試行する

<現在の状態>

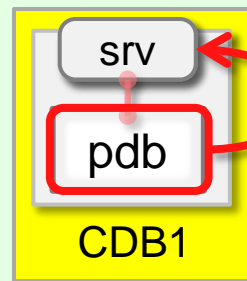


A) サービスを停止する場合



サービスの
停止のみ

B) PDB をクローズする場合



PDB の
クローズ時
にサービスも
停止

DBMS_SERVICE パッケージによるサービスの管理

DBMS_SERVICE パッケージの使用

- DBMS_SERVICE パッケージ (10g R1 ~)
 - シングル・インスタンスに対してサービスの管理を行う
 - サービスの作成、削除、セッションの切断、変更、起動、停止が可能
 - 12c では PDB に対してもサービスの管理を行えるように拡張されている
- PDB のサービス管理では、操作対象の PDB に接続して操作を実行する
 - PDB (pdb1) にサービス (pdb1_srv1) を作成する場合の例

```
SQL> show con_name
```

```
CON_NAME
```

```
-----
```

```
PDB1
```

```
SQL> exec DBMS_SERVICE.CREATE_SERVICE('pdb1_srv1','pdb1_srv1');
```

DBMS_SERVICE パッケージによるサービスの管理

基本操作（作成 / 起動 / 停止 / 削除）

- サービスの作成（PDB に接続して実行する）

```
SQL> exec DBMS_SERVICE.CREATE_SERVICE('pdb1_srv1', 'pdb1_srv1');
```

- サービスの起動

```
SQL> exec DBMS_SERVICE.START_SERVICE('pdb1_srv1');
```

- サービスの停止

```
SQL> exec DBMS_SERVICE.STOP_SERVICE('pdb1_srv1');
```

- サービスの削除

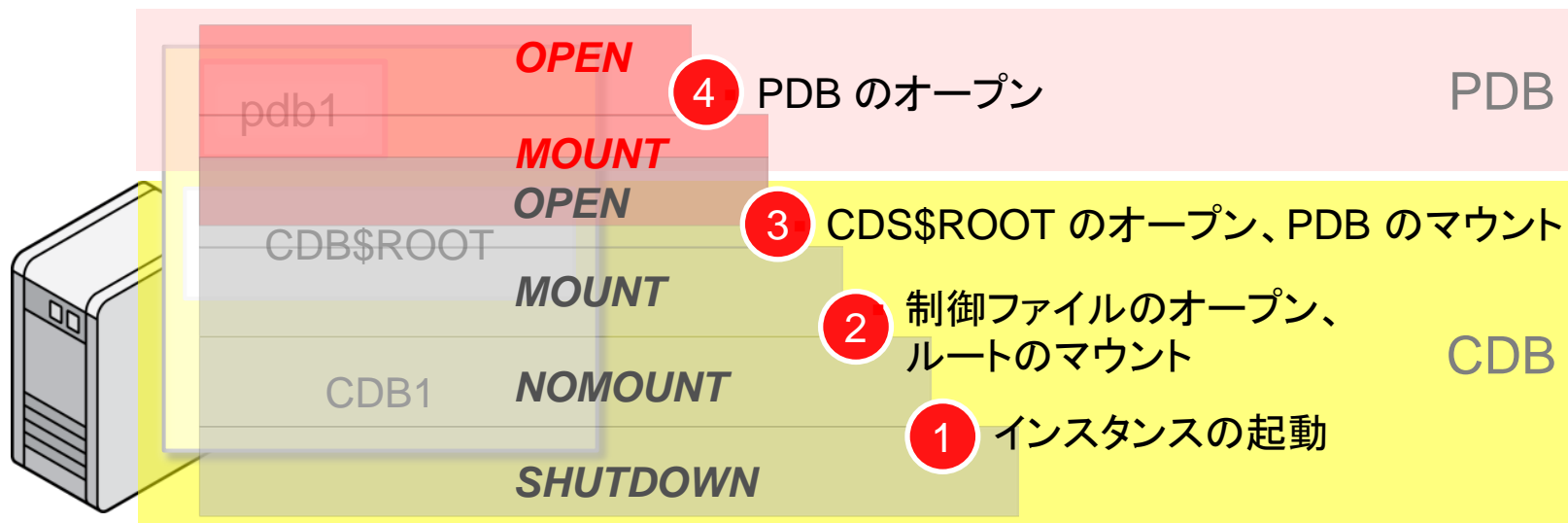
```
SQL> exec DBMS_SERVICE.DELETE_SERVICE('pdb1_srv1');
```

CDB 環境におけるデータベースの起動

オラクルが起動するまでのステップ

- シャットダウンされた状態から PDB のオープンまで、次のフェーズで遷移する

ステータス ステータスの変更ステップ



プラグブル・データベースのステータス管理

PDB のオープン

- PDB のオープン操作は、CDB がオープンしていることが前提
- ルートへの接続時に PDB をオープンする場合の構文

```
ALTER PLUGGABLE DATABASE <PDB_NAME> OPEN [<OPTIONAL_CLAUSE>];
```

– 例

```
ALTER PLUGGABLE DATABASE pdb1 OPEN READ ONLY;
```

- 上記は PDB (*pdb1*) を、読み取り専用でオープンする場合のコマンド例
- *<OPTIONAL_CLAUSE>* には、次の指定が可能
 - オープンにおけるモードの指定
 - 制限付きモードの適用

プラガブル・データベースのオープン

オープンにおけるモードの指定

- PDB のオープン・モード (OPEN MODE) として指定可能なオプション

READ WRITE

- 読み書きが可能
- 明示的に指定しない場合に、デフォルトの設定として適用される

READ ONLY

- 読み取り専用
- DML (INSERT / UPDATE / DELETE / MERGE) 文は実行できない

UPGRADE

- PDB への接続が制限される
- データベースの移行時や、アップグレード時に使用する

プラガブル・データベースのオープン

制限付きモードの適用

- オープン・モードに加えて、RESTRICTED モードをオプションとして指定可能

RESTRICTED

- RESTRICTED SESSION 権限を持つユーザーのみ接続を許可
- オープン・モードにUPGRADE を指定した場合は、暗黙的に適用される
- RESTRICTED モードの適用状況は、次のコマンドでも確認が可能

```
SQL> show pdbs
```

CON_ID	CON_NAME	OPEN MODE	RESTRICTED
2	PDB\$SEED	READ ONLY	NO
3	HR	READ WRITE	YES
4	ERP	MOUNTED	
5	CRM	READ WRITE	NO

プラグブル・データベースのステータス管理

PDB のクローズ

- オープンしている PDB のクローズとは、PDB のステータスをマウントにすることを指す



- ルートへの接続時に PDB をクローズする場合の構文

```
ALTER PLUGGABLE DATABASE <PDB_NAME> CLOSE [<OPTIONAL_CLAUSE>];
```

– 例

```
ALTER PLUGGABLE DATABASE pdb1 CLOSE IMMEDIATE;
```

- 上記は PDB (*pdb1*) を、即時停止する場合のコマンド例

プラガブル・データベースのクローズ

クローズにおけるモードの指定

- PDB のクローズ時には、NORMAL または IMMEDIATE で停止が行われる
- PDB のクローズにおいて指定できるオプション

IMMEDIATE

- 即時停止を実行するモード
- 既に接続しているユーザーがいた場合、接続は切断される
- NORMAL は通常停止を実行するモード
 - 停止を発行後は、新規接続は不可
 - 既に接続しているユーザーがいた場合は、接続が切断されるまで待機する
 - IMMEDIATE を記述しない場合にデフォルトとして適用される

データベース管理者向けの作業効率化

(1) 起動あるいは停止における複数対象の一括指定

- 複数の PDB を操作対象として一括指定することが可能
 - 対象が 1つの場合は PDB 名を記述する
 - 対象が複数の場合は、いくつか記述方法がある
 - PDB 名を複数記述する

```
ALTER PLUGGABLE DATABASE <PDB_NAME1, NAME2, NAME3> <ACTION>;
```

- すべての PDB を対象にする場合は、ALL を記述する

```
ALTER PLUGGABLE DATABASE ALL <ACTION>;
```

- ALL EXCEPT を記述して、対象から除外する PDB 名を 1つ以上記述する

```
ALTER PLUGGABLE DATABASE ALL EXCEPT <PDB_NAME> <ACTION>;
```



データベース管理者向けの作業効率化

(2) トリガーによる PDB の起動

データベース起動時に、特定の PDB も一緒に起動したい

- トリガーの設定により、PDB のオープンを自動化することも可能
 - 通常では CDB のオープンと PDB のオープンは別作業だが、トリガーの設定により作業の効率化が可能
 - トリガーの設定例



データベース
管理者

```
CREATE OR REPLACE TRIGGER <TRIGGER_NAME>
AFTER STARTUP ON DATABASE
BEGIN
  EXECUTE IMMEDIATE 'ALTER PLUGGABLE DATABASE ALL OPEN';
END;
/
```

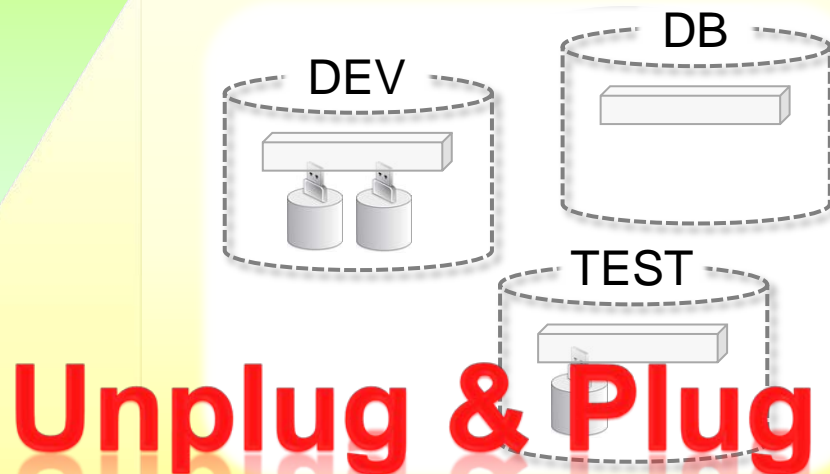
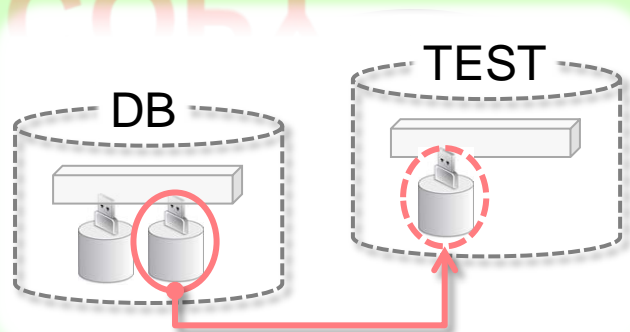
- CDB と PDB で管理者の住み分けに考慮が必要
 - PDB の管理操作には、基本的に PDB 管理者やユーザーを想定

プラグブル・データベースの作成

状況に応じた作成手法

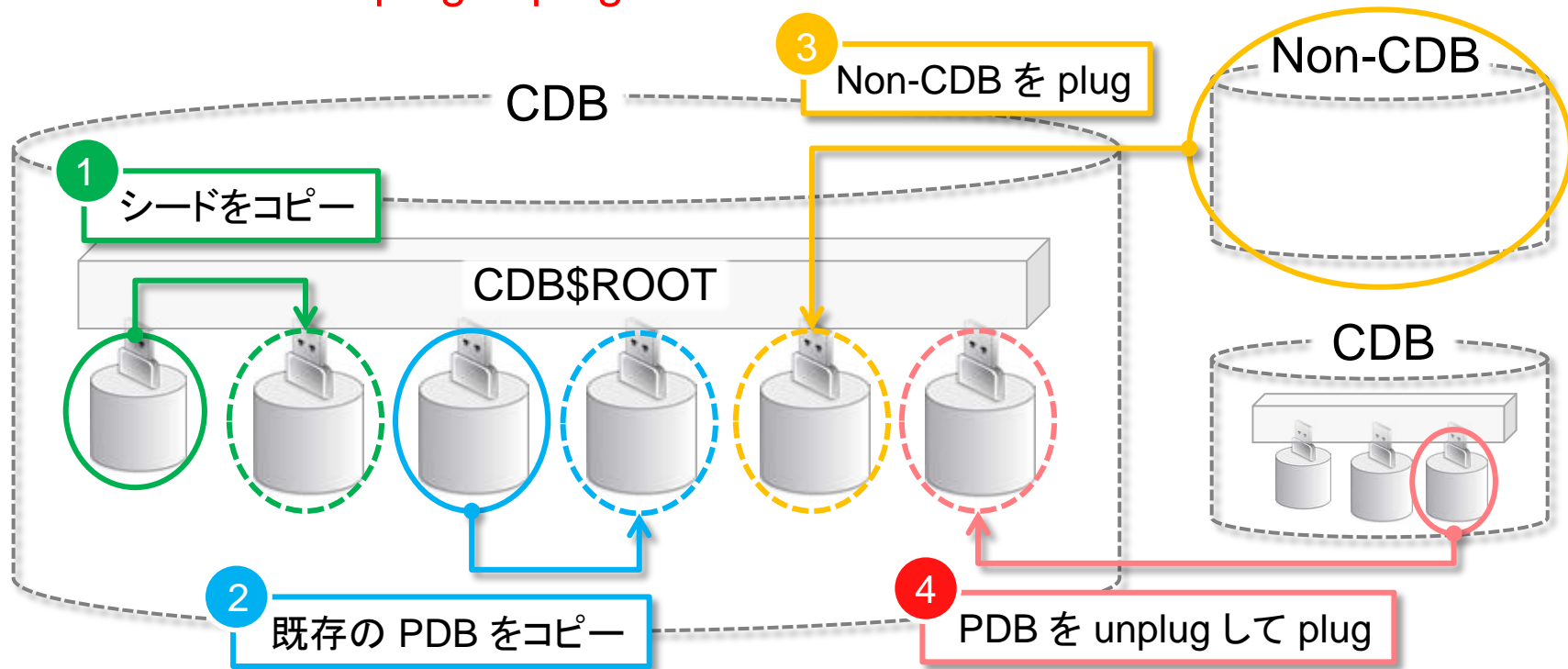
- PDB の作成には、コピーや unplug と plug を使用する
- 手動 (CREATE 文) 以外にも DBCA や EMCC といったツールを使用可能

COPY



状況に応じた作成手法

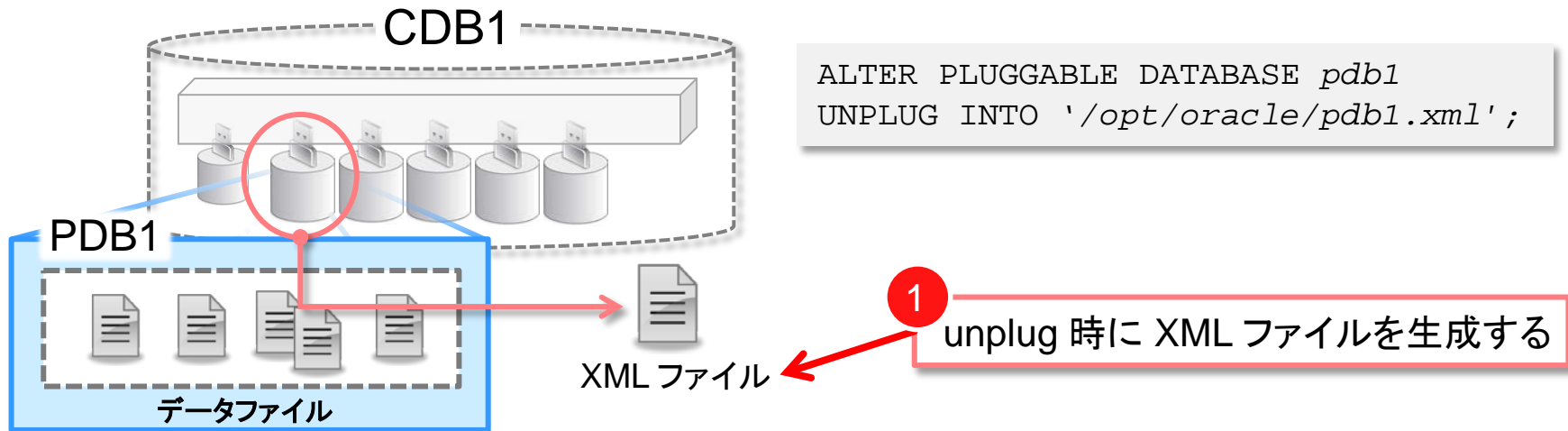
コピーまたは unplug と plug による作成



プラグブル・データベースの unplug

PDB の切断と XML ファイルの作成

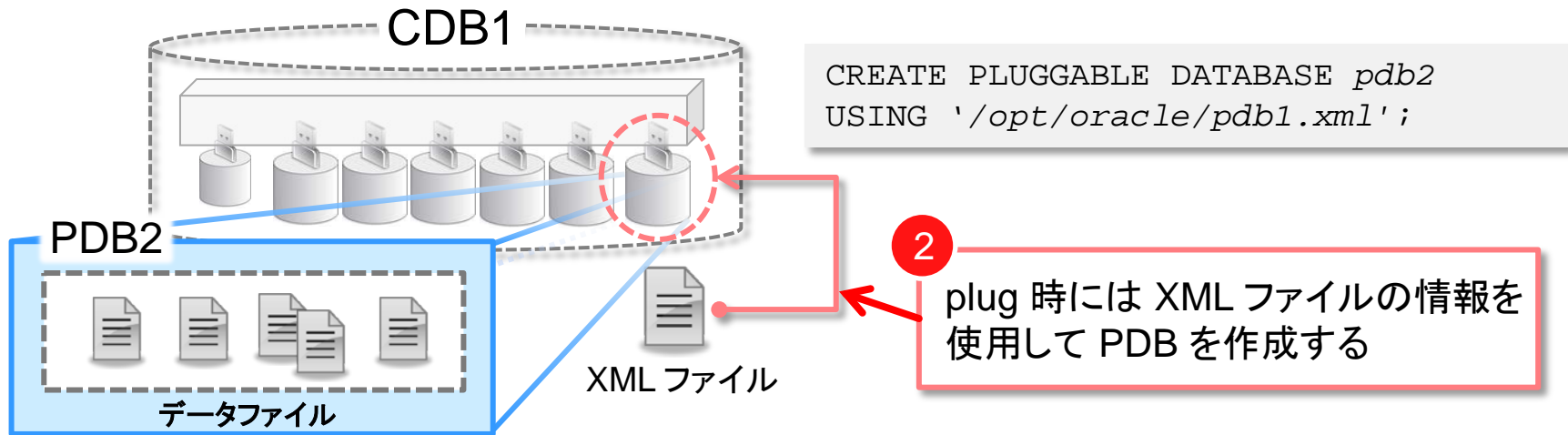
- Unplug 操作では PDB を CDB から切り離し、XML メタデータ・ファイルを作成する
- コマンドラインあるいは DBCA などのツールを使用可能
 - コマンドラインの場合は ALTER PLUGGABLE DATABASE 文を使用する



プラグブル・データベースの plug

XML ファイルを使用した PDB の作成

- Plug 操作では、Unplug 時に作成した XML メタデータ・ファイルを使用する
- コマンドラインあるいは DBCA などのツールを使用可能
 - コマンドラインの場合は CREATE PLUGGABLE DATABASE 文を使用する



初期化パラメータ

PDB ごとに設定可能な初期化パラメータ

- いくつかの初期化パラメータに関しては、PDB ごとに設定が可能
 - インスタンス名やメモリ関連の設定などは除く
- 変更には ALTER SYSTEM SET 文を使用する
 - ルートに接続して変更を行った場合は、すべてのコンテナに適用
 - 個別に設定変更を行った PDB は、変更後、適用対象からは除く
 - 特定の PDB に接続して変更を行った場合は、その PDB のみに適用
- Unplug / plug 時にも PDB ごとに設定変更した値を保持
 - PDB を unplug した際には、XML ファイルに現行の設定値を格納
 - Plug 時には、保持された設定値を基に PDB ごとに設定した値を適用

初期化パラメータに関する留意事項

初期リリースにおける動作と注意点

- 個別に値を設定変更した PDB を異なる CDB 間で unplug / plug する場合、plug 先の CDB の値で設定を上書きする
 - 同一 CDB での unplug / plug は除く
 - 事前に V\$PARAMETER などで PDB の設定値を確認しておくことを推奨
- PDB ごとに設定変更が可能な初期化パラメータの判別方法
 - V\$PARAMETER の ISPDB_MODIFIABLE 列の値により判別が可能
 - 設定変更が可能なパラメータを表示するコマンド例

```
SQL> SELECT NAME FROM V$PARAMETER WHERE ISPDB_MODIFIABLE = 'TRUE'  
ORDER BY NAME;
```

ユーザー

ユーザーの種類

ローカル・ユーザー (local user)

- 特定の PDB のみにユーザーが存在するタイプ
- ローカル・ユーザーを作成する場合は、PDB へ接続して操作を実施
 - ユーザー名は 30 バイト以内

共通ユーザー (common user)

- 各コンテナ（ルートと各 PDB）に同名のユーザーが存在するタイプ
- 共通ユーザーを作成する場合には、ルートへ接続して操作を実施
 - ユーザー名に接頭辞（C## / c##）が必要
 - ユーザー名は 25 バイト以内

異なるユーザー・タイプの活用

ユーザーの作成例と役割

共通

ローカル

データベース全体の管理者は共通ユーザーを使用する



CDB

CDB\$ROOT

PDB\$SEED

アプリケーションごとのユーザーはローカルユーザーを使用する

SYS

SYS

SYSTEM

PDB 2
PDB 1

PDB n

SYS

USER01

SYSTEM

HRADM

SYS

USER01

SYSTEM

ERPADM

他の PDB とユーザー名は重複してもよい



ロールと権限

ロールと権限の種類

ローカル・ロール (local role)

- 特定の PDB のみに存在するロール、共通の権限は含まない

共通ロール (common role)

- ルートと各 PDB で共通のロール、共通およびローカル・ロールを含む
- 共通ロールを作成する場合には、ロール名に接頭辞 (C##) が必要

ローカル権限 (local privilege)

- 特定の PDB のみに限定された権限

共通権限 (common privilege)

- ルートと各 PDB で共通の権限、共通ユーザーによって付与される

プラグブル・データベースの 管理

ORACLE[®] 12^c
DATABASE



Plug into the **Cloud.**

ORACLE[®]

バックアップとリカバリ

マルチテナント・アーキテクチャにおける基本指針

Manage Many As One

- CDB 単位で管理をすることでシンプルな運用を実現
- EM Cloud Control や RMAN による利便性も提供
- 次のようなケースを想定し、PDB 単位での管理も可能
 - バックアップ頻度やスケジュールが PDB ごとに異なるケース
 - リカバリの影響範囲を極小化したいケース

 基本的には CDB 単位でのシンプルな運用を行い、PDB ごとの要件に応じて PDB 単位での運用を検討する

バックアップとリカバリ

Oracle Recovery Manager とフラッシュバック・テクノロジー

- CDB や PDB のバックアップ・リカバリは、Oracle Recovery Manager (RMAN) やフラッシュバック・テクノロジーを使用

Oracle Recovery Manager (RMAN)

- Oracle Database のバックアップ、リストア、リカバリ、バックアップ・ファイルの管理を行うためのクライアント・ユーティリティ

フラッシュバック・テクノロジー

- 人為的ミスなどによるデータのリカバリを目的とした機能
- データを時間軸によって表示する参照系の機能や、データを前後に巻き戻すことが可能なリカバリ系の機能を提供
- 初期リリースでは、CDB にのみ対応

Oracle Recovery Manager によるバックアップ

バックアップ実行時のコマンド例

- RMAN を使用した CDB 全体や PDB ごとのバックアップ実行例
 - バックアップの実行には SYSBACKUP 権限が必要
 - 対象を指定することで、ルートや PDB のバックアップが可能
- CDB 全体を対象とする場合（ルートおよび PDB を含む）

```
RMAN> BACKUP DATABASE
```

- ルートのみを対象とする場合

```
RMAN> BACKUP DATABASE ROOT
```

- 特定の PDB のみを対象とする場合

```
RMAN> BACKUP PLUGGABLE DATABASE
```


Oracle Recovery Manager によるリカバリ

リカバリ実行時のコマンド例

- RMAN を使用した CDB 全体や PDB ごとのリカバリ実行例
 - ルートや PDB のリカバリ時に対象の指定が可能
 - CDB 全体を対象とする場合（ルートおよび PDB を含む）

```
RMAN> RECOVERY DATABASE
```

- ルートのみを対象とする場合

```
RMAN> RECOVERY DATABASE ROOT
```

- 特定の PDB のみを対象とする場合

```
RMAN> RECOVERY PLUGGABLE DATABASE
```

リソースの管理

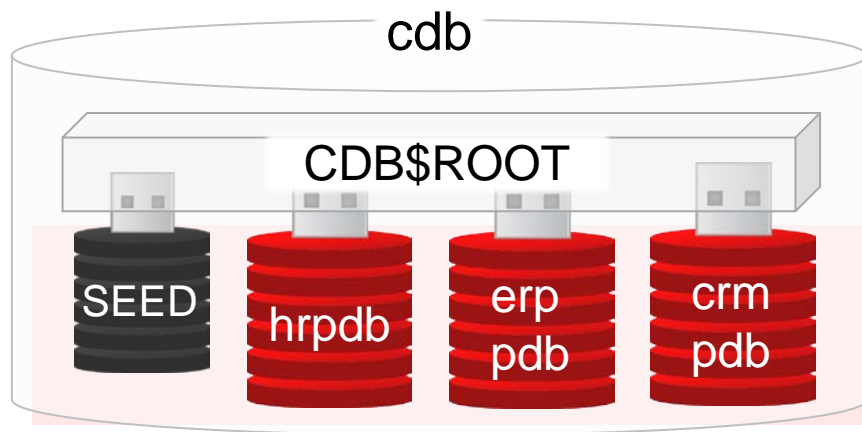
Oracle Database Resource Manager

- CDB や PDB のリソース管理は、Oracle Database Resource Manager を使用
Oracle Database Resource Manager (リソース・マネージャ)
 - サーバーのリソースをデータベース内で管理するための機能
 - データベース内における各アプリケーション間のリソース配分を定義
 - 最低限使用できるリソース量を確保することで、パフォーマンスを安定
- マルチテナント・アーキテクチャにおけるリソース管理
 - 従来提供していたデータベース間でのリソース管理機能に加えて、PDB 間でのリソース管理機能を提供（メモリ、ネットワーク I/O は未対応）

リソース・マネージャによるリソース管理

マルチテナント・アーキテクチャにおけるリソース管理のポイント

- PDB 単位でのリソース管理機能の提供
- PDB 数の増減にも対応する効率的なリソース管理の実現



PDB レベルでリソースを管理
PDB の増減時も効率的に管理

PDB 単位でのリソース管理機能の提供

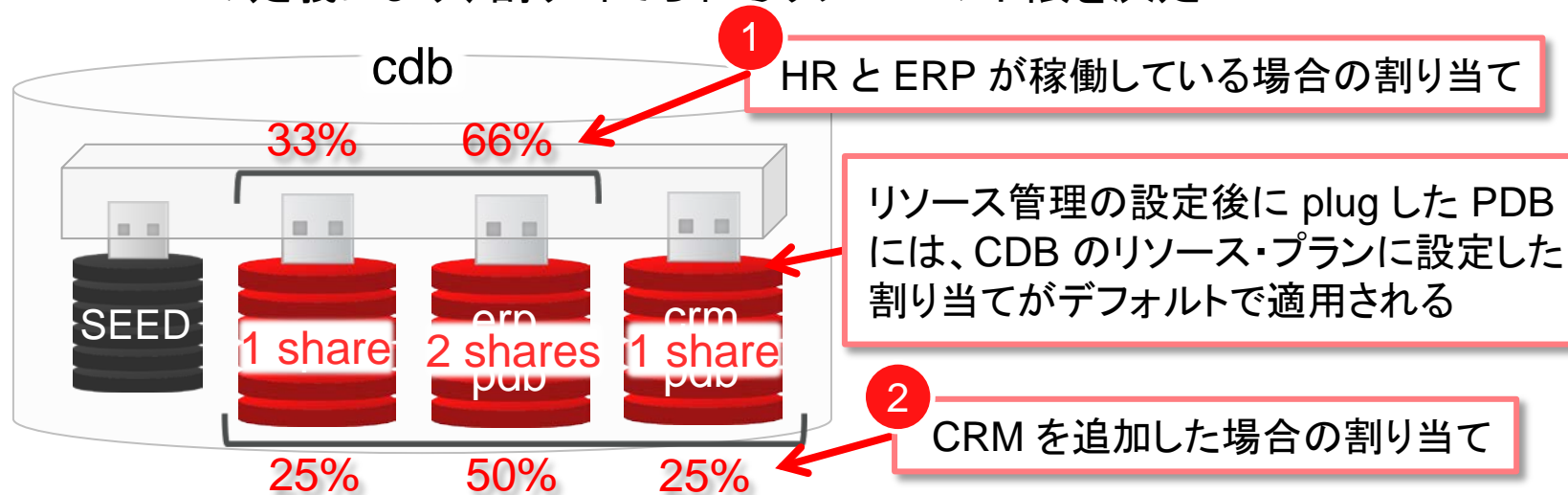
PDB 間でのリソース配分の制御

- CDB 内のリソース配分を PDB ごとに設定して管理
 - CDB レベルでリソース・マネージャが有効化されていることが前提
 - 前提条件に加えて PDB レベルでリソース・マネージャを有効化することで、リソースの制御が可能になる
 - 設定には CDB 単位でのリソース設定と同様に DBMS_RESOURCE_MANAGER パッケージを使用する
- PDB 単位での設定において、CDB 単位での設定と異なる点は以下
 - リソース・プランにサブプランを含むことはできない
 - 設定可能なリソース・コンシューマ・グループは最大 8 つ
 - 複数レベルでのスケジューリング・ポリシーは設定できない

効率的なリソース管理の実現

管理対象である PDB の増減にも最低限の管理工数で対応

- share の設定を使用することにより、リソース割り当てを効率化
 - plug あるいは unplug の際に、リソース割り当ての再計算が不要
 - share の定義により、割り当てられるリソースの下限を決定



まとめ

ORACLE[®] 12^c
DATABASE



Plug into the **Cloud.**

ORACLE[®]

次世代データベースによる統合

マルチテナント・アーキテクチャ



アプリケーションへの変更は不要

独立性とマルチテナントの実現

迅速なプロビジョニング

セキュアかつ高い可用性

ITコストの削減

低い管理コスト

効率のよいリソース活用

性能と拡張性

ライセンス

- すべてのエディションで利用可能なオプション機能として提供
 - 一部のエディションでは、ユーザーが作成できる PDB の数に制限がある
 - Standard Edition One (SE1) および Standard Edition (SE) では 1 個のみ
 - Enterprise Edition (EE) では 252 個まで作成可能

オプション名	Edition	対応状況	作成数の上限
Oracle Multitenant	Standard Edition One	△ 対応(制限あり)	1
	Standard Edition	△ 対応(制限あり)	1
	Enterprise Edition	○ 対応	252

Hardware and Software

ORACLE®

Engineered to Work Together

ORACLE®