

ORACLE®



# Oracle Database 12c Release 1

## マルチテナント・アーキテクチャ (1)

日本オラクル株式会社

ORACLE<sup>®</sup>  
DATABASE 12<sup>c</sup>



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。

ORACLE®

# 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 (1)
- プラガブル・データベースの基本操作 (2)

# Oracle Database 12c で 実現する次世代データベース



Plug into the **Cloud**.

# 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 データ  
プロセス ファイル



ORACLE®

# 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有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

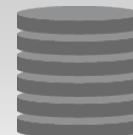
## 顧客管理 DB

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



ORACLE®

# 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

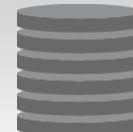
## 顧客管理 DB

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



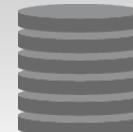
## 販売管理 DB

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



ORACLE®

# 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



ORACLE®

# 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)

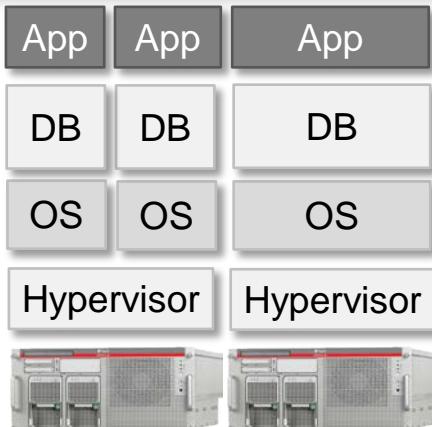


ORACLE®

# 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



*Server*

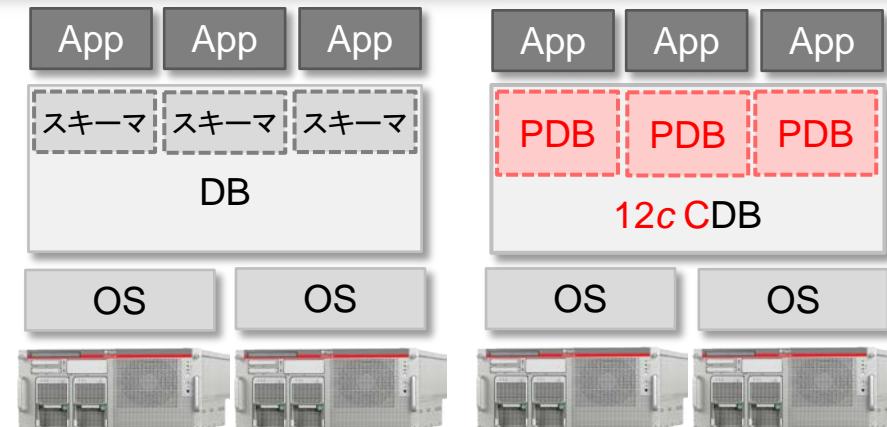
- サーバー仮想化



*Operating System*

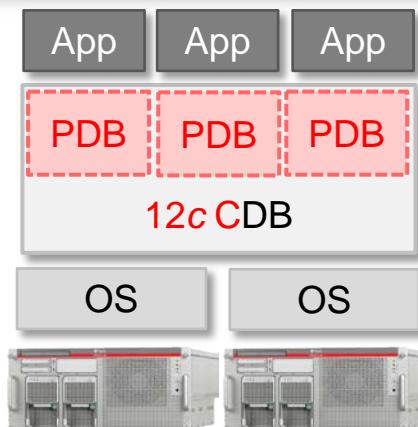
- Real Application Clusters (RAC)

データベース・クラウド



*Database*

- データベース・スキーマ



*New*  
**Multitenant  
Architecture**

ORACLE

# 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

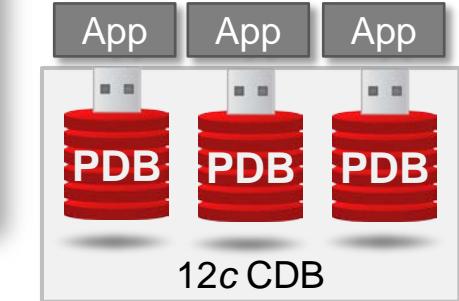
- 複数の顧客データを単一のデータベースへ統合
- サーバーや運用管理にかかるコストを削減
  - クラウド環境を利用したい企業
  - アプリケーションごとにデータベースを分けて運用している企業

## プラガブル・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

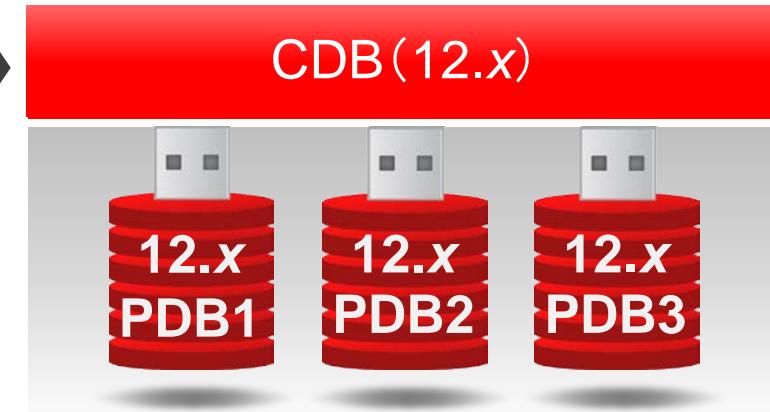


ORACLE

# 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



ORACLE®

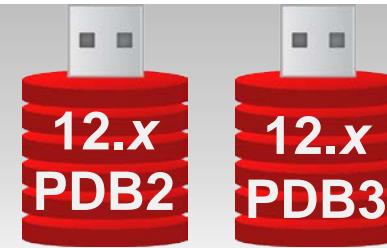
# 必要に応じたプラガブル・データベースの移動

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)



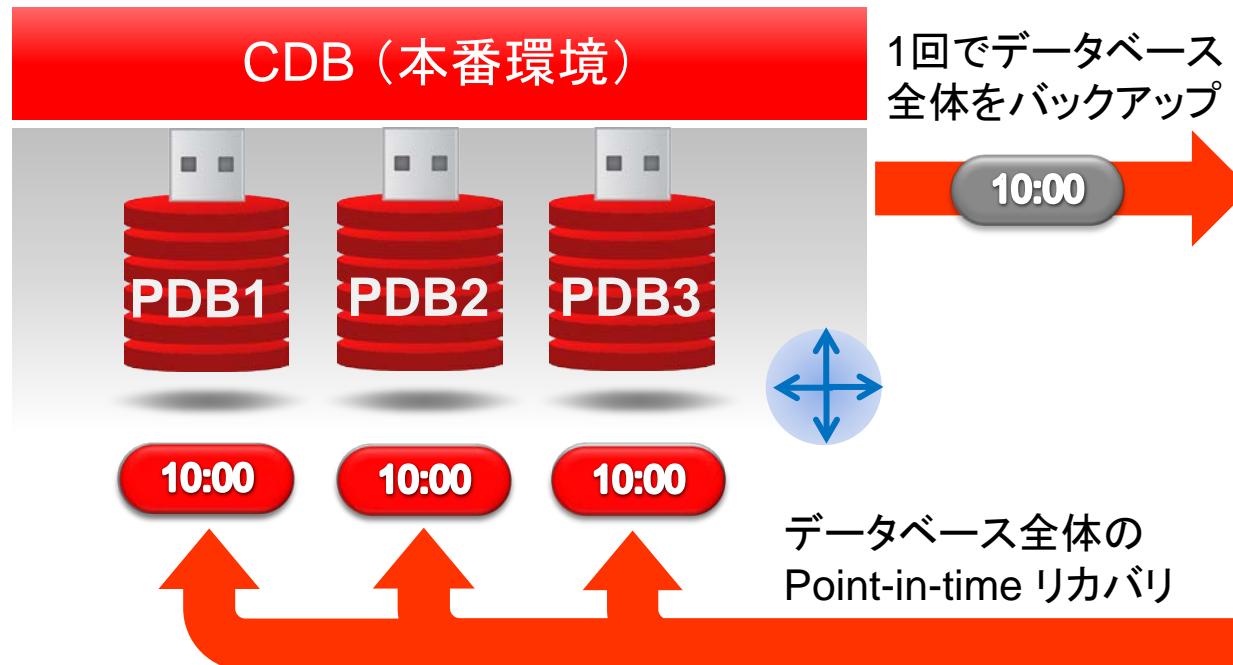
CDB2 (開発環境)



ORACLE®

# 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

プラガブル・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

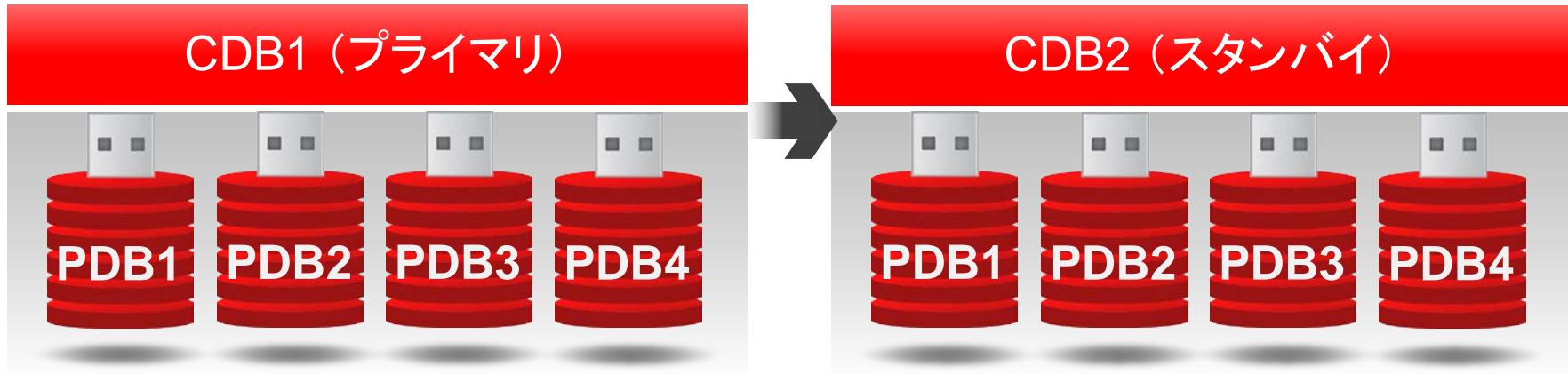
RAC 構成の CDB



ORACLE®

# 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



ORACLE®

# マルチテナント・ アーキテクチャの概要



# 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

ORACLE

# マルチテナント・アーキテクチャの利用

## 構成および使用のための条件

- マルチテナント・アーキテクチャを構成および使用するための条件は以下
  - Oracle Database 12c をインストールしていること
  - データベースの互換性を 12.0.0 以上に設定していること
    - 初期化パラメータ COMPATIBLE の値で確認が可能
    - 以下は、OUI および DBCA による初期構成における出力例

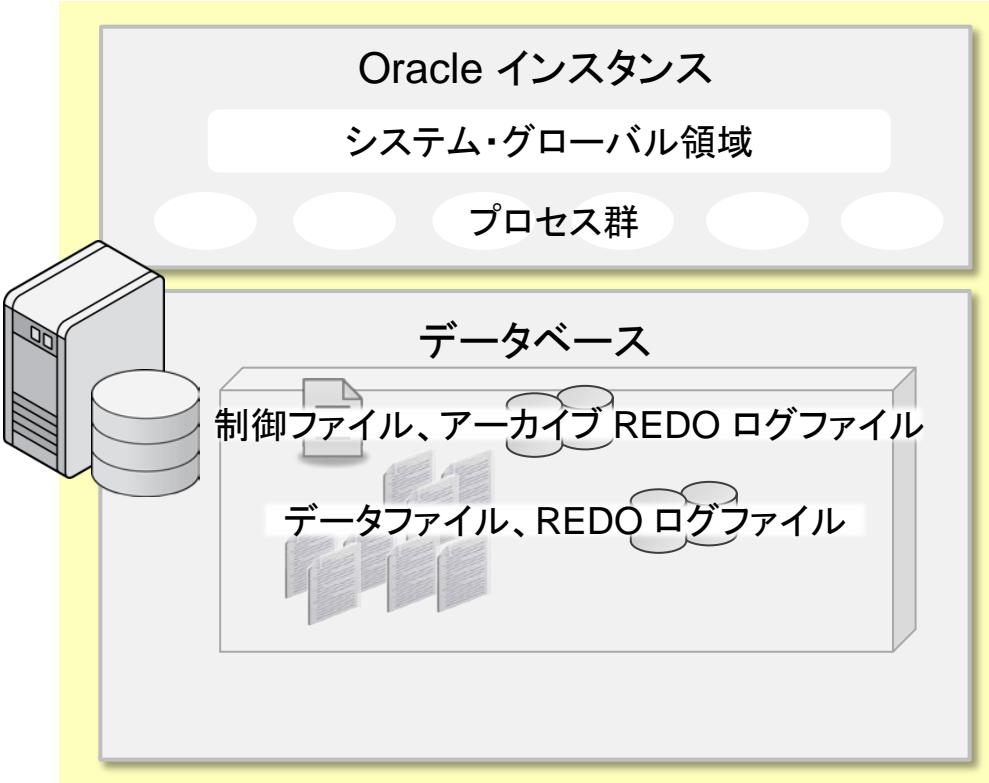
```
SQL> SELECT NAME, VALUE FROM V$PARAMETER WHERE NAME = 'compatible';  
NAME          VALUE  
-----  
compatible    12.1.0.0.0
```

ORACLE®

# 従来からのデータベース

## Non-CDB

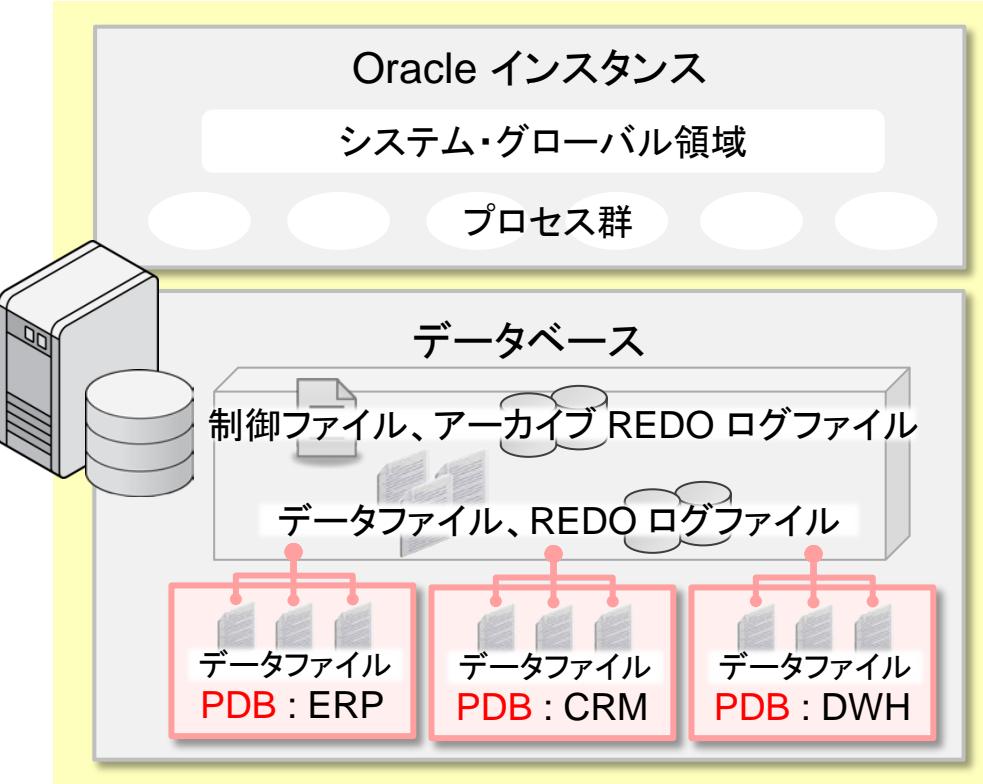
- Oracle データベースは、サーバー上で確保されるメモリ領域とプロセス群から構成される  
Oracle インスタンスと、ストレージに格納されるデータベース（ファイル群）により構成される
- この構造を拡張し、Oracle Database 12c では、マルチテナント・アーキテクチャに対応したデータベースを提供



# マルチテナント・アーキテクチャ対応のデータベース

## CDB

- 従来のデータベースと基本的な構造は同じ
- 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

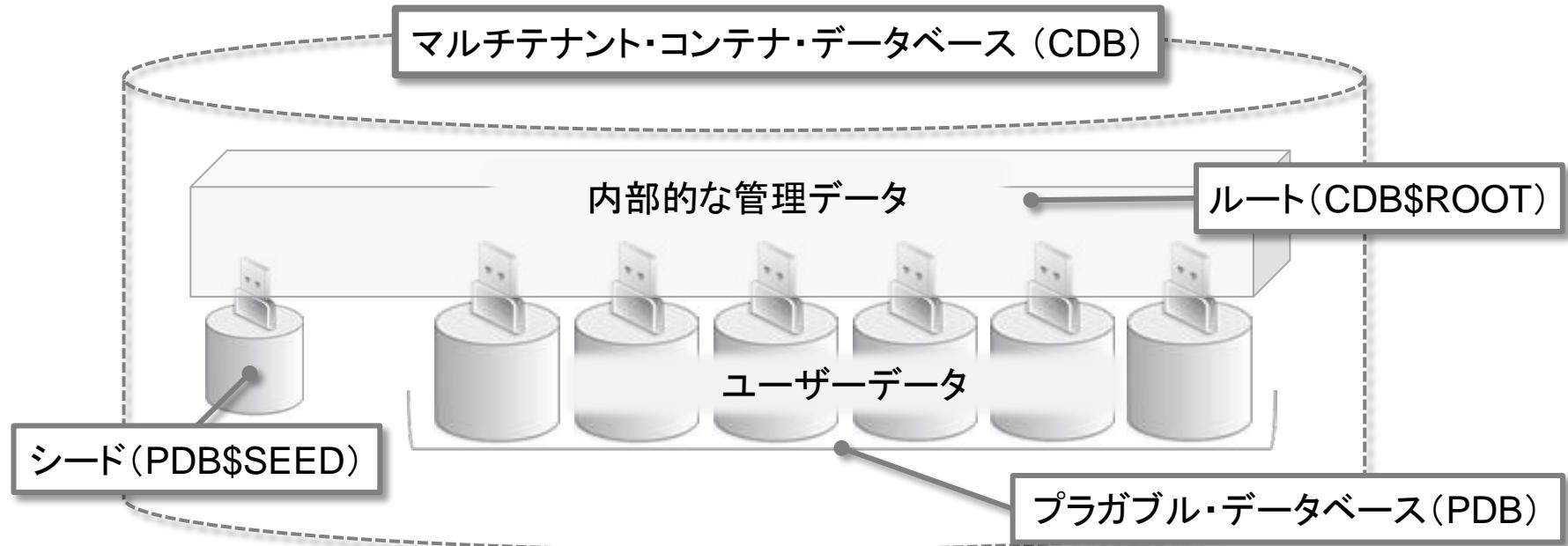


ORACLE

# 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



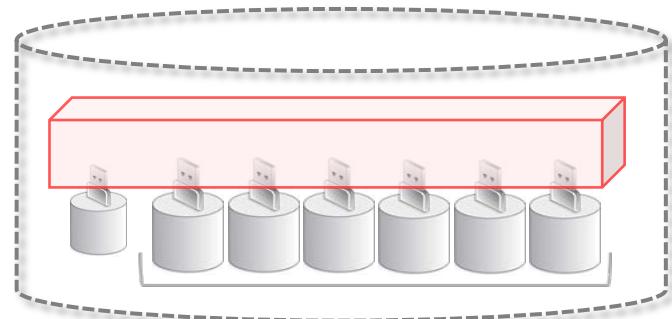
ORACLE

# 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



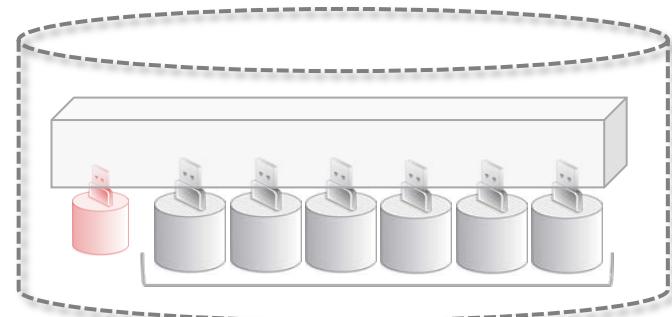
ORACLE

# 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



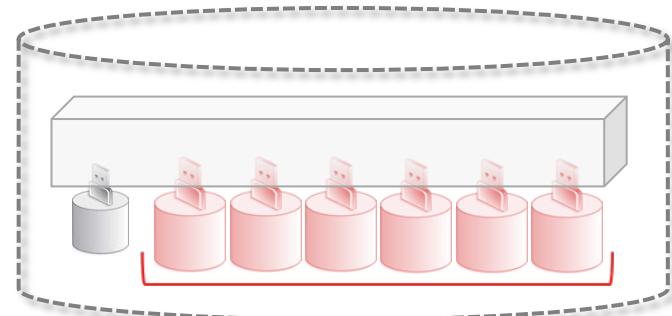
ORACLE

# プラガブル・データベース (PDB)

スキーマやオブジェクトの集合

## プラガブル・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

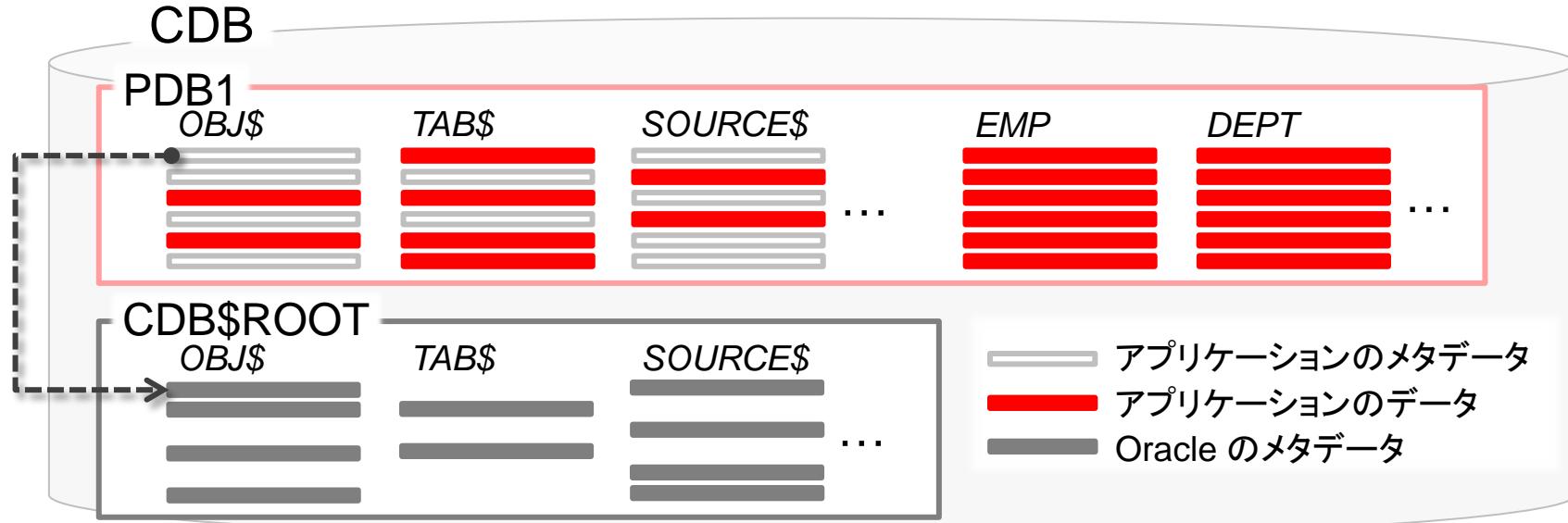


ORACLE

# 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



ORACLE

# 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

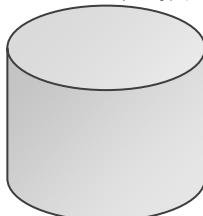
USERS 表領域

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®

# 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

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

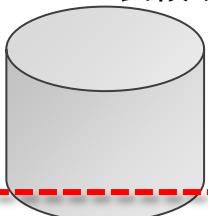
USERS 表領域

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

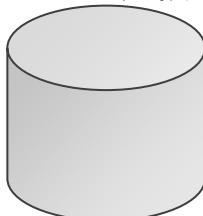
USERS 表領域

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



### 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

ORACLE®

# 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

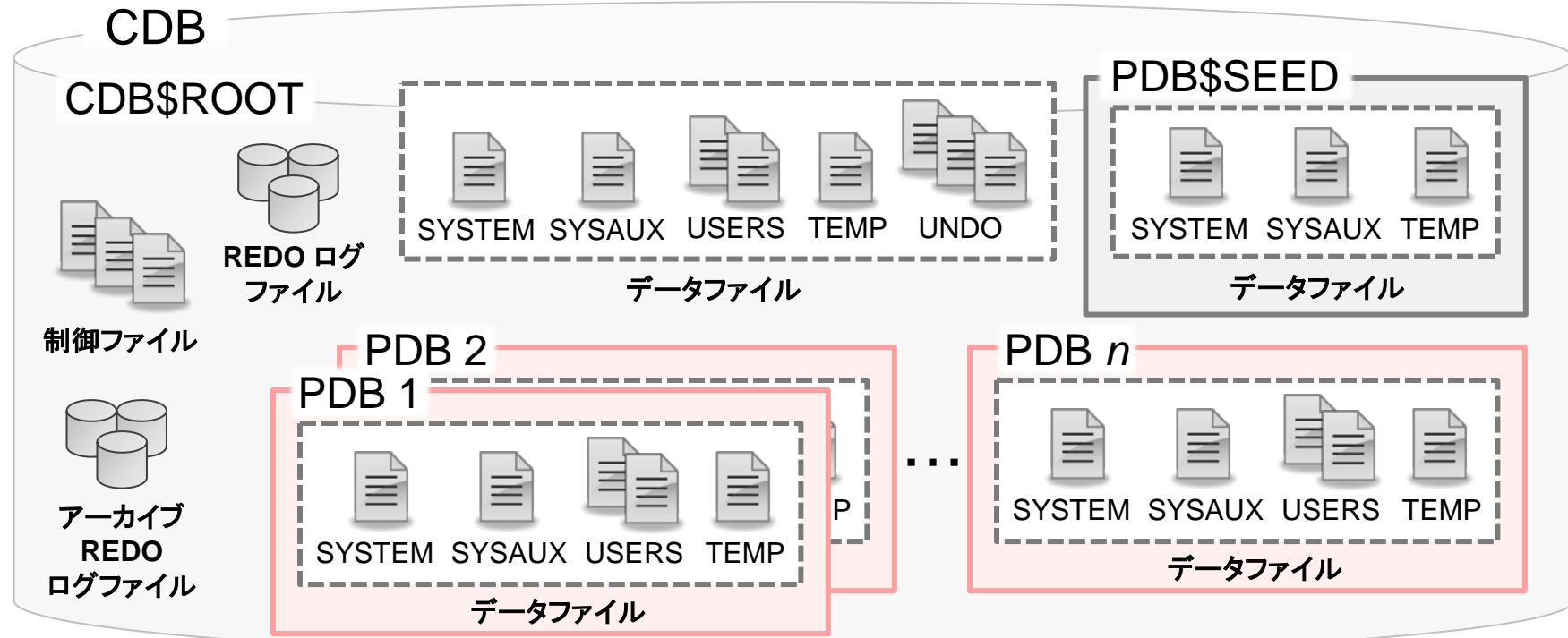
8行が選択されました。

ルートの表領域

PDB の表領域

# 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



ORACLE

# 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

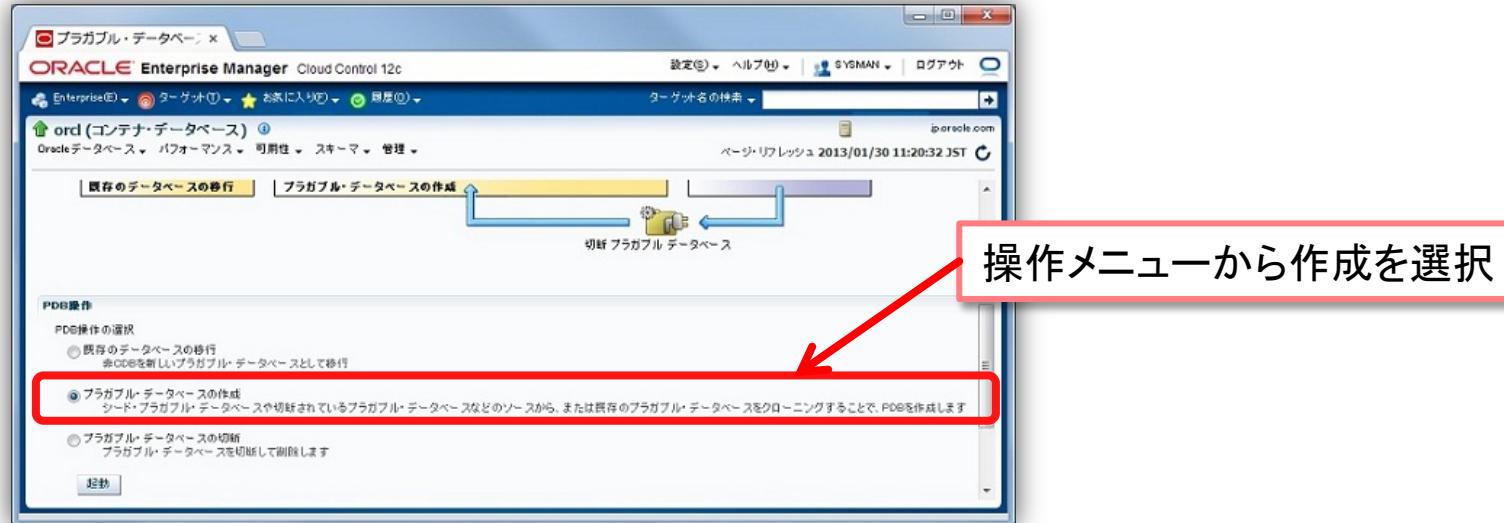
- 基本的には、non-CDB の作成と手順は同じ
  - 「データベース識別情報」画面の抜粋



# プラガブル・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
    - 「プラガブル・データベースのプロビジョニング」画面の抜粋



38 | Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

ORACLE

# 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)

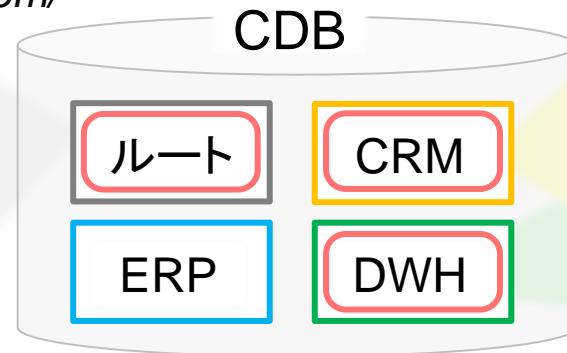
- 複数のサーバーを一元的に管理 (統合管理ツール)
- Cloud Control 用に用意したサーバー上に構成して使用する
- 管理対象のサーバーにエージェントを構成して使用する
- データベース管理のすべての機能を提供
- データベース以外にも、オラクル製品を中心に多彩な機能を提供

# Oracle Enterprise Manager Database Express

## マルチテナント・アーキテクチャにおける構成

- マルチテナント・アーキテクチャでは、コンテナごとに EM Express を構成する
  - DBCA を使用した CDB 作成時には、ルートに対しての構成を指定可能
  - 手動での構成は、対象のコンテナに SYSDBA 権限で接続して行う
  - コンテナごとに異なるポート番号を使用する <https://host01.com:5501/em/>

<https://host01.com:5500/em/>



<https://host01.com:5502/em/>

ORACLE

# プラガブル・データベースの 基本操作 (1)



# 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>
```

ORACLE

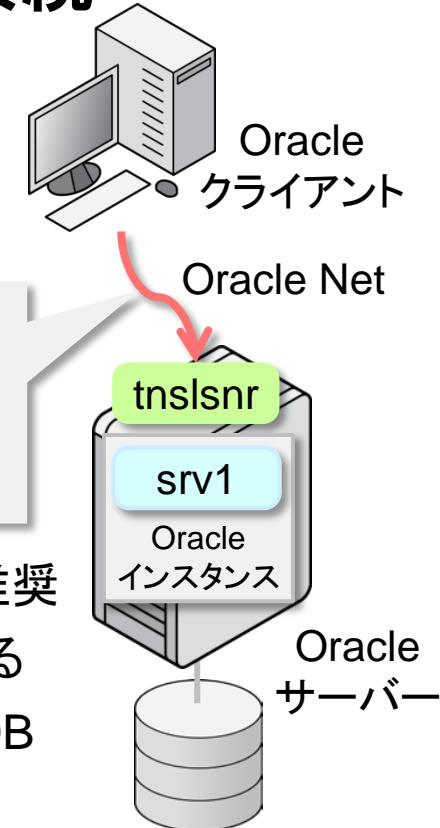
# マルチテナント・アーキテクチャにおける接続

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=srv1))
)
```

- 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



ORACLE

# 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

ORACLE

# 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

ORACLE®

# 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
```

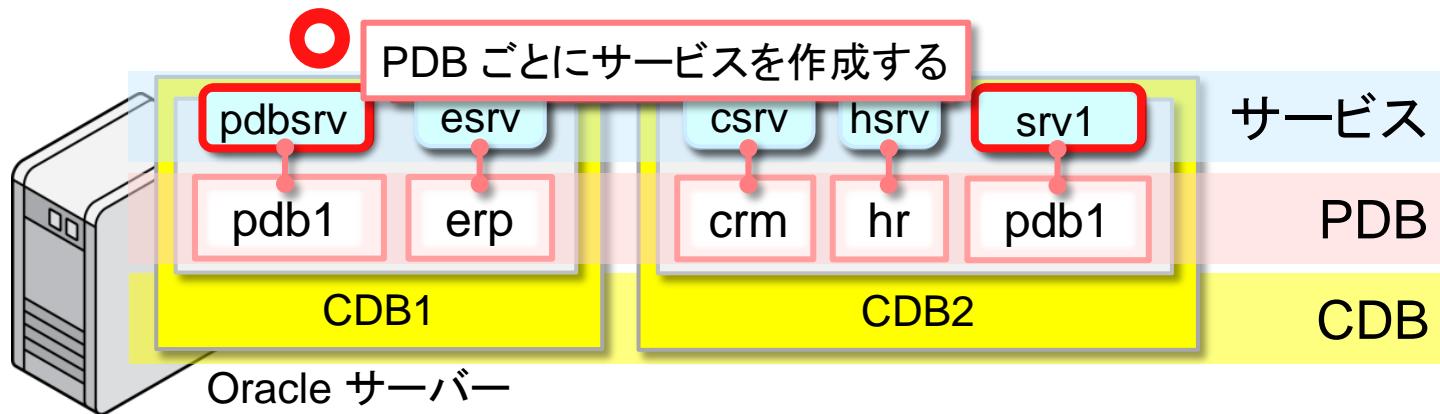
- PDB に接続している場合の出力例

```
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

ORACLE

# 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>
```

ORACLE®

# 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する PDB の情報を追加

- サービスの構成情報として、関連する PDB の情報を追加
  - 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>
プラガブル・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
```

ORACLE

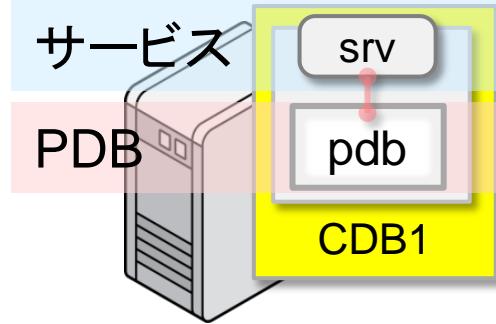
# サービスと PDB の依存関係

## 起動・停止における挙動

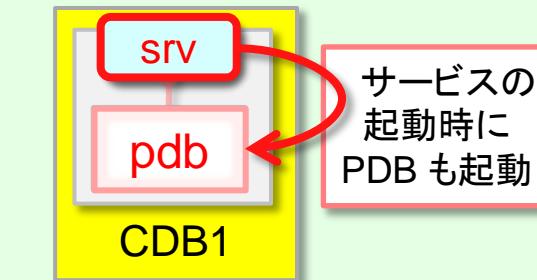
- サービスを起動する際は、データベースの場合と同様に PDB の起動を試行する
- 起動時

- 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

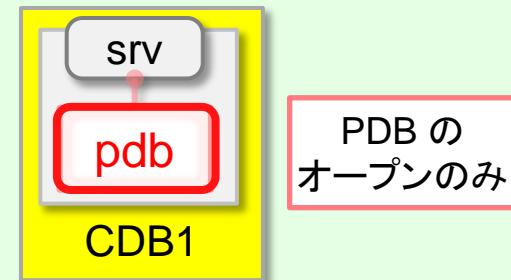
＜現在の状態＞



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

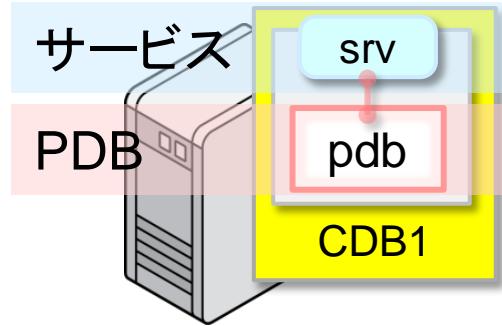
## 起動・停止における挙動

- 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

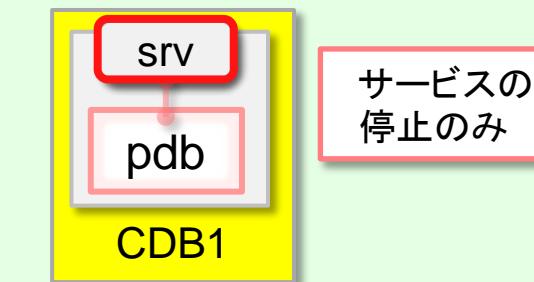
### 停止時

- 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

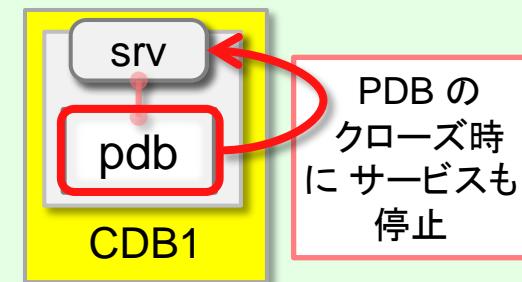
#### ＜現在の状態＞



#### A) サービスを停止する場合



#### B) PDB をクローズする場合



# 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');
```

ORACLE

# 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');
```

ORACLE

# 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



ORACLE®

# プラガブル・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

ORACLE®

# 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

ORACLE®

# プラガブル・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

ORACLE

# 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
  - IMMEDIATE を記述しない場合にデフォルトとして適用される
  - 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

# 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>;
```

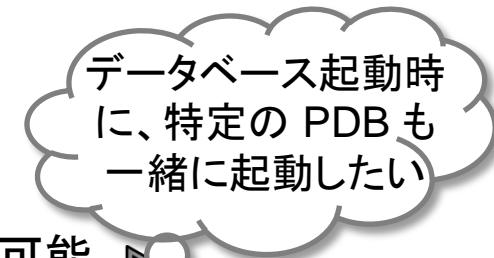
- ALL EXCEPT を記述して、対象から除外する PDB 名を 1つ以上記述する

```
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



- 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



```
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

ORACLE®

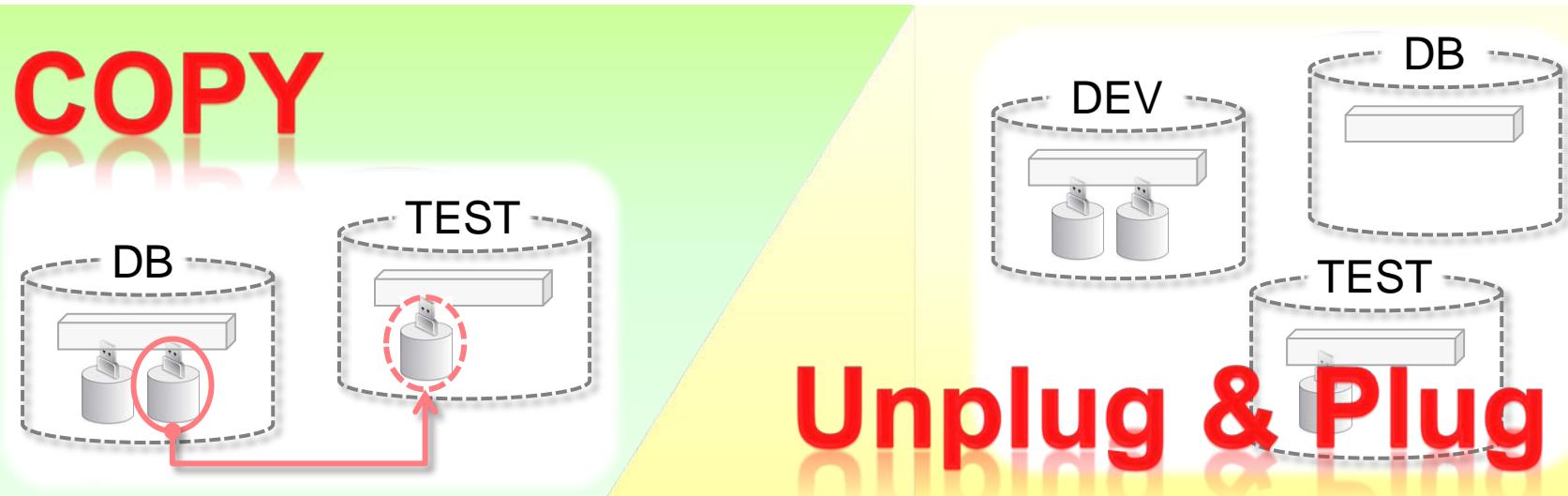
# プラガブル・データベースの 基本操作 (2)



# プラガブル・データベースの作成

## 状況に応じた作成手法

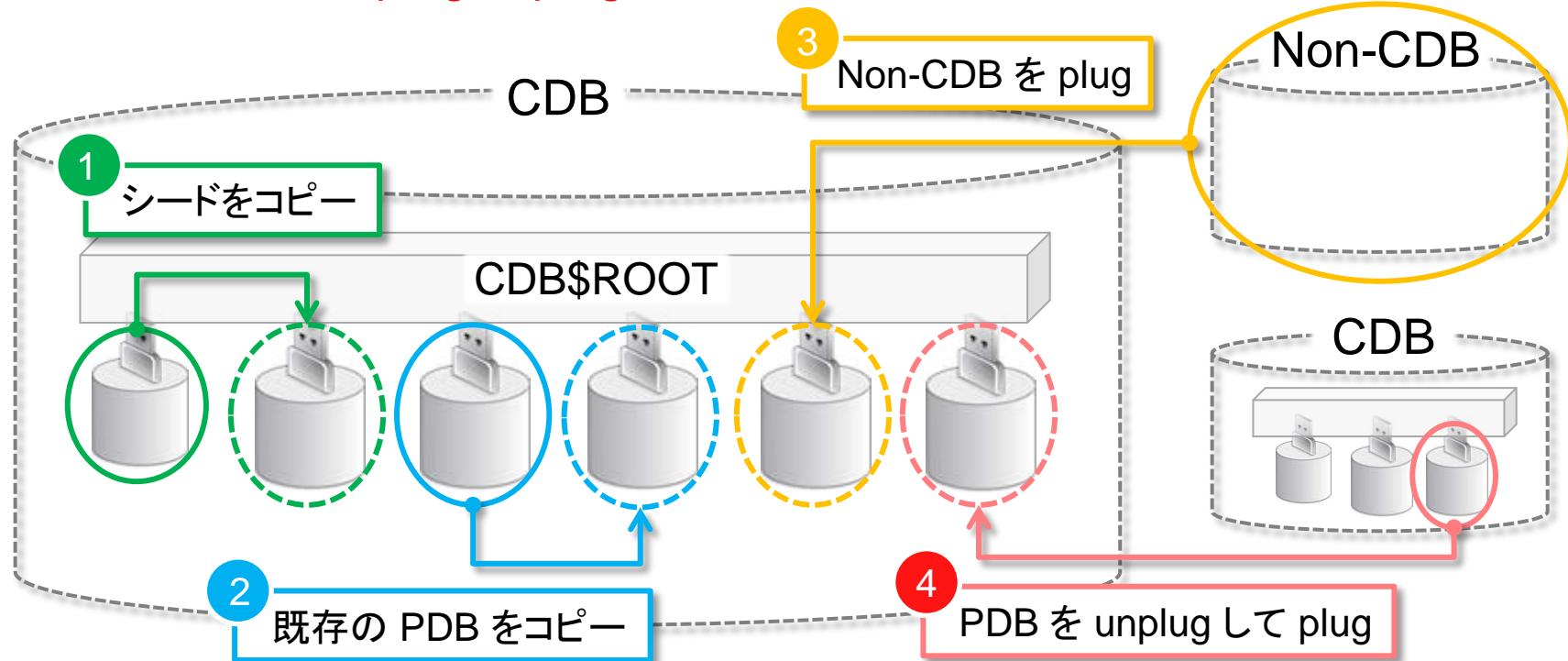
- PDB の作成には、コピーや unplug と plug を使用する
- 手動 (CREATE 文) 以外にも DBCA や EMCC といったツールを使用可能



ORACLE

# 状況に応じた作成手法

コピーまたは unplug と plug による作成



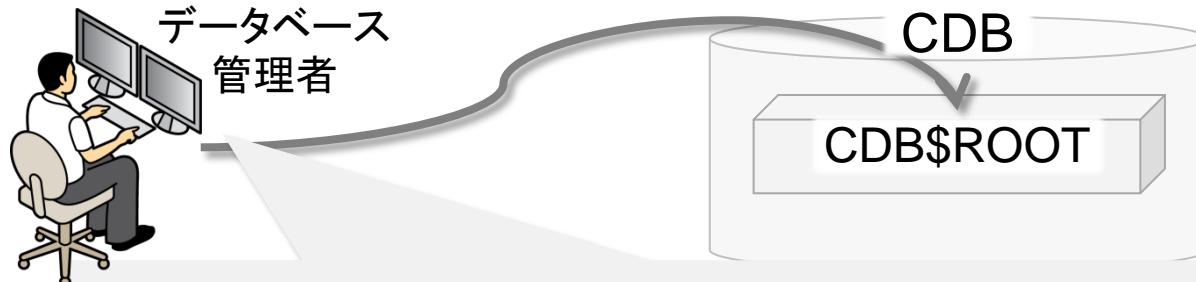
ORACLE

# プラガブル・データベースの作成

## 前提条件

- プラガブル・データベースの作成にあたり、次の条件を満たしている必要がある
  - CDB が作成され、READ WRITE モードで起動している
  - 共通ユーザーでルートに接続している
  - 接続ユーザーが CREATE PLUGGABLE DATABASE 権限を有している

例



```
$ sqlplus sys/Welcome1@scan.oracle12c.jp:1521/cdb1 as sysdba
```

ORACLE

# プラガブル・データベースの作成

## (1) PDB\$SEED を使用した作成

- PDB\$SEED から PDB を作成する
- 構文

```
CREATE PLUGGABLE DATABASE <PDB_NAME> ADMIN USER <USER_NAME> IDENTIFIED  
BY <PASSWORD> [<OPTIONAL_CLAUSE>];
```

### - 例

```
CREATE PLUGGABLE DATABASE pdb1 ADMIN USER admin IDENTIFIED BY Pwd;
```

- 作成する PDB のデータファイル配置場所は、Oracle Managed Files (OMF) や初期化パラメータ PDB\_FILE\_NAME\_CONVERT の設定により異なる
- FILE\_NAME\_CONVERT 句を用いて、明示的に指定することも可能

ORACLE®

# データファイル配置場所の指定

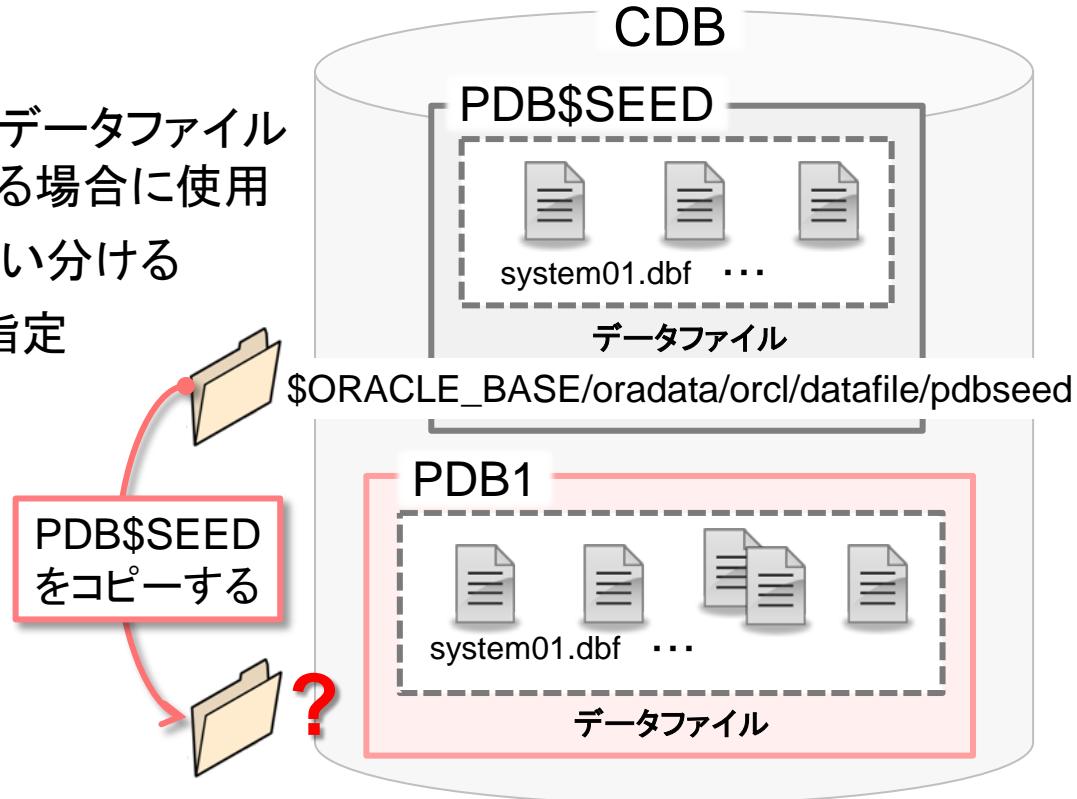
## 指定方法と優先度

- データファイルの配置場所は、次のいずれかの方法で指定が可能
  1. FILE\_NAME\_CONVERT 句
  2. Oracle Managed Files (OMF)
  3. 初期化パラメータ PDB\_FILE\_NAME\_CONVERT
- 複数の方法を組み合わせた場合は、上位の方法による指定が適用される
- PDB 関連のデータファイル管理の例
  - CDB を OMF 構成で作成し、PDB 関連のファイルも基本的には OMF に準拠して配置する
  - 開発用の PDB など例外的に配置場所を変更したい場合には、作成時に FILE\_NAME\_CONVERT 句を使用して配置先を変更する

# FILE\_NAME\_CONVERT 句

## ファイルの配置場所を指定

- 新規作成する PDB について、データファイルの配置場所を明示的に指定する場合に使用
- ケースに応じて、指定方法を使い分ける
  - ディレクトリ単位での一括指定
  - ファイル単位での指定
  - ディレクトリ単位とファイル単位を組み合わせた指定



ORACLE

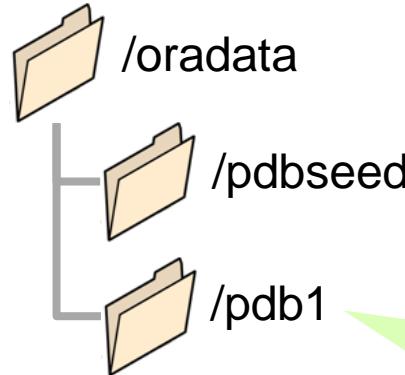
# FILE\_NAME\_CONVERT 句の活用

## ディレクトリ単位での配置場所の指定

- ファイルを配置するディレクトリを指定する場合のコマンド例と配置イメージ

```
CREATE PLUGGABLE DATABASE pdb1 ADMIN USER admin IDENTIFIED BY Admin
FILE_NAME_CONVERT=( '/oradata/pdbseed' , '/oradata/pdb1' );
```

例



PDB\$SEED のファイル群

temp01.dbf	system01.dbf
sysaux01.dbf	

pdb1 のファイル群

temp01.dbf	system01.dbf
sysaux01.dbf	<i>pdb1</i> _users01.dbf

ORACLE®

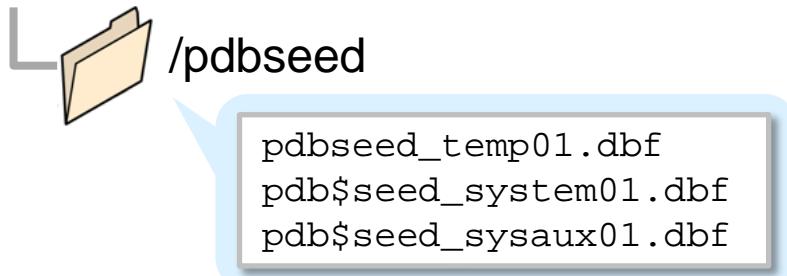
# FILE\_NAME\_CONVERT 句の活用

## ファイル単位での配置場所の指定

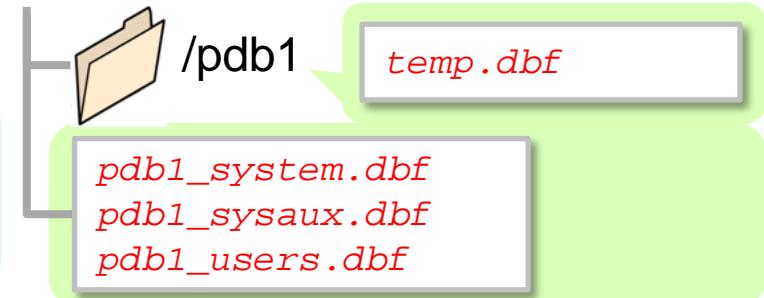
- ファイルごとに配置を指定する場合のコマンド例と配置イメージ

```
CREATE PLUGGABLE DATABASE pdb1 ADMIN USER admin IDENTIFIED BY Admin
FILE_NAME_CONVERT=(
  '+DG1/pdbseed/pdbseed_temp01.dbf', '+DG2/pdb1/temp.dbf',
  '+DG1/pdbseed/system01.dbf', '+DG2/pdb1_system.dbf',
  '+DG1/pdbseed/sysaux01.dbf', '+DG2/pdb1_sysaux.dbf');
```

例 +DG1



+DG2



ORACLE

# FILE\_NAME\_CONVERT 句の活用

## ディレクトリおよびファイル単位での配置場所の指定

- ディレクトリ単位とファイル単位の指定を組み合わせることも可能
- 例：system01.dbf のみ別ディレクトリに配置したい場合

```
CREATE PLUGGABLE DATABASE pdb1 ADMIN USER admin IDENTIFIED BY Admin
FILE_NAME_CONVERT=( '/oracle/pdbseed' , '/oracle/pdb1' ,
' /oracle/pdbseed/system01.dbf' , '/oracle/pdb1_system01.dbf' );
```

- 特定のディレクトリあるいはファイルへの指定を複数回記述した場合には、後に記述した指定内容が適用されるので注意が必要
  - 先に記述したファイル単位での指定が、後のディレクトリ単位の指定により上書きされてしまう



```
FILE_NAME_CONVERT=( '/oracle/pdbseed/system01.dbf' ,
' /oracle/pdb1_system01.dbf' , '/oracle/pdbseed' , '/oracle/pdb1' );
```

# プラガブル・データベースの作成

## (2) 既存の PDB を使用した作成

- 既存の PDB から、新しい PDB を作成する
- 構文

```
CREATE PLUGGABLE DATABASE <TARGET_PDB_NAME> FROM <SOURCE_PDB_NAME>
[<OPTIONAL_CLAUSE>];
```

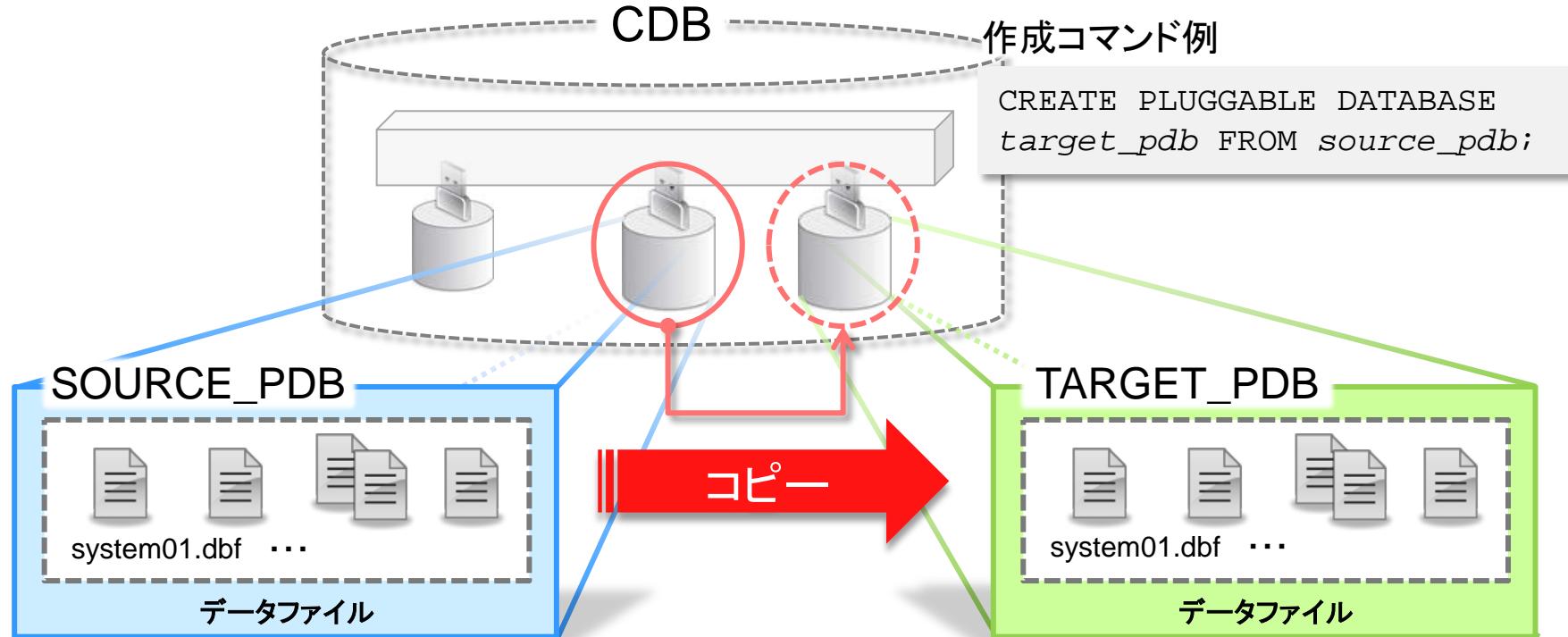
### – 例

```
CREATE PLUGGABLE DATABASE testpdb FROM hrpdb;
```

- 同一 CDB 内（ローカル）、あるいは異なる CDB 間（リモート）での作成が可能
- ソースとする PDB は、読み取り専用（READ ONLY モード）でオープンされている必要がある

# 既存のプラガブル・データベースを使用した作成

同一 CDB 内（ローカル）の場合



ORACLE

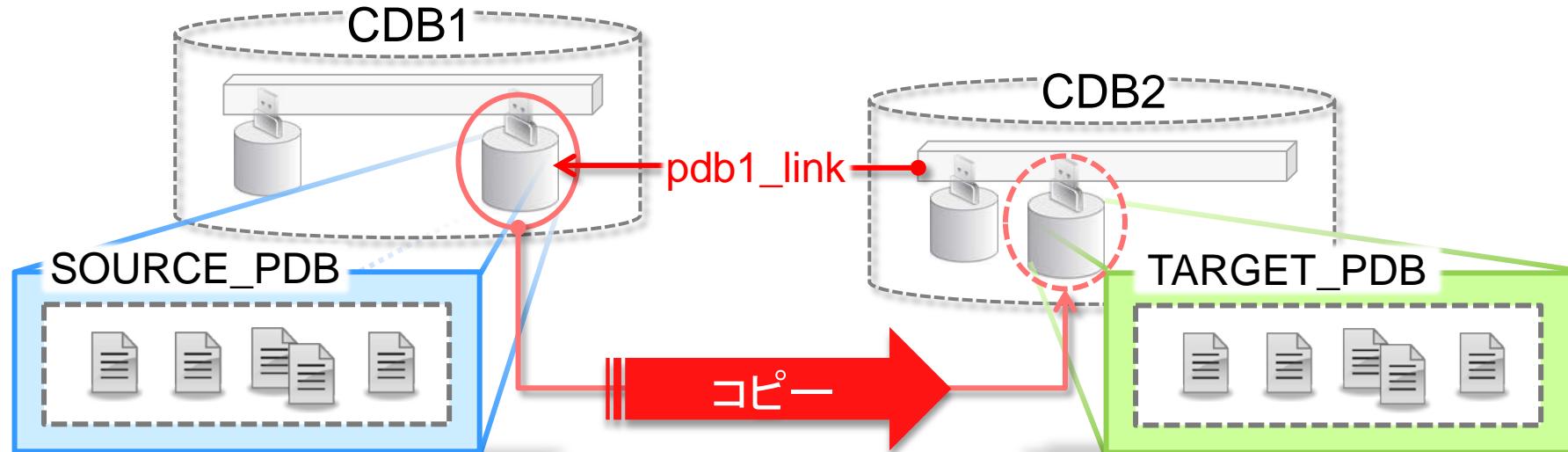
# 既存のプラガブル・データベースを使用した作成

## 異なる CDB 間（リモート）の場合

- 異なる CDB 間での作成する場合は、データベース・リンクを使用する

作成コマンド例

```
CREATE PLUGGABLE DATABASE target_pdb FROM source_pdb@pdb1_link;
```



ORACLE

# 異なる CDB 間でのプラガブル・データベースの作成

## データベース・リンクの作成例

- 異なる CDB 間で既存の PDB を使用して新規 PDB を作成する場合の例

### CDB1 の PDB1 (ソース側)

```
CREATE USER dblink IDENTIFIED BY dblink;  
GRANT CREATE SESSION, DBA TO dblink;
```

```
SHUTDOWN IMMEDIATE;  
STARTUP OPEN READ ONLY;
```

} データベース・リンク用の  
ユーザーを作成

} READ ONLY モードにしておく

### CDB2 のルート (ターゲット側)

```
CREATE DATABASE LINK pdb1_link CONNECT TO  
dblink IDENTIFIED BY dblink USING 'PDB1_LINK';
```

```
SELECT * FROM SYS.TEST@PDB1_LINK; データベース・リンク作成後の確認
```

} データベース・リンクを作成

# 既存のプラガブル・データベースを使用した作成

## コピー元の確認

- CDB\_PDB\_HISTORY ビューにより、PDB 作成の履歴を確認することが可能
  - 作成元や作成した時間を確認することが可能
  - HRPDBTEST の作成元となった PDB を確認する場合の実行例

```
SQL> SELECT PDB_NAME, CLONED_FROM_PDB_NAME FROM CDB_PDB_HISTORY  
      WHERE PDB_NAME = 'HRPDBTEST' ;
```

PDB_NAME	CLONED_FROM_PDB_NAME	OPERATION
HRPDBTEST	HRPDB	CLONE

- 既存の HRPDB を元に、クローンとして作成されていることがわかる
- OPERATION 列により、新規作成や unplug / plug といった操作も判別が可能

# プラガブル・データベースの作成

## (3) 既存の non-CDB からの作成

- 既存の non-CDB を PDB として plug することで作成する
  - あらかじめ CDB を作成しておき、PDB として plug する
- 作成方法は、次のいずれかを使用可能（対応バージョンに制限あり）
  - DBMS\_PDB パッケージ DB 12c ~
    - DBMS\_PDB パッケージを使用して XML ファイルを生成し、PDB を作成
  - Oracle Data Pump
    - トランスポータブル・データベースを使用 DB PSR 11.2.0.3 ~
    - トランスポータブル表領域、または export / import を実行 ~ DB PSR 11.2.0.2

# プラガブル・データベースの作成

## 既存の non-CDB からの作成に使用できる方法の一覧

- 使用できる方法は、non-CDB のバージョンによって異なる

作成方法 バージョン	DBMS_PDB パッケージ	Oracle Data Pump		
		トランス ポータブル・ データベース	トランス ポータブル 表領域	Export / Import
12c	○ 対応	○ 対応	○ 対応	○ 対応
11.2.0.3 以降	N/A	○ 対応	○ 対応	○ 対応
11.2.0.3 以前	N/A	N/A	○ 対応	○ 対応



Non-CDB を最新のバージョンすることで移行の選択肢が広がる

ORACLE®

# DBMS\_PDB パッケージを使用した作成

## 作成手順

- PDB を作成する CDB を作成し、non-CDB を READ ONLY モードで起動する
- DBMS\_PDB.DESCRIBE プロシージャを使用して XML ファイルを作成する
  - non-CDB に対して XML ファイルを生成する場合の実行例

```
BEGIN
  DBMS_PDB.DESCRIBE(
    pdb_descr_file => '/home/oracle/nonCDBtoPDB1.xml');
END;
/
```

- 生成した XML ファイルを使用して、PDB を作成する
- USING 句を含む CREATE PLUGGABLE DATABASE 文で作成
  - PDB を使用する際には作成後に別途オープンを行う

# プラガブル・データベースの作成

## (4) 既存の PDB の unplug / plug による作成

- 既存の PDB を unplug (取り外し) と plug (取り付け) することによる作成
  - 関連ファイル群の位置情報を含む XML ファイルを生成して作成に使用する
- 構文

### unplug

```
ALTER PLUGGABLE DATABASE <PDB_NAME> UNPLUG INTO <FILE_LOCATION>;
```

### plug

```
CREATE PLUGGABLE DATABASE <PDB_NAME> [ AS CLONE ] USING <FILE_LOCATION>
[ <OPTIONAL_CLAUSE> ];
```

- 例 (unplug の場合)

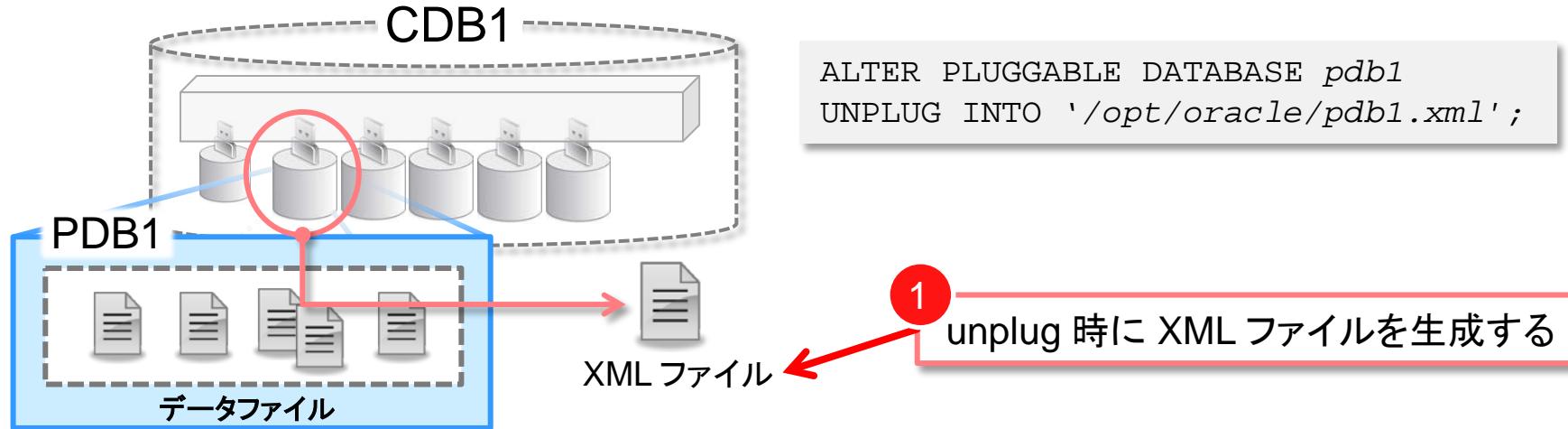
```
ALTER PLUGGABLE DATABASE pdb1 UNPLUG INTO '/opt/oracle/pdb1.xml';
```

ORACLE

# プラガブル・データベースの unplug

## PDB の切断と XML ファイルの作成

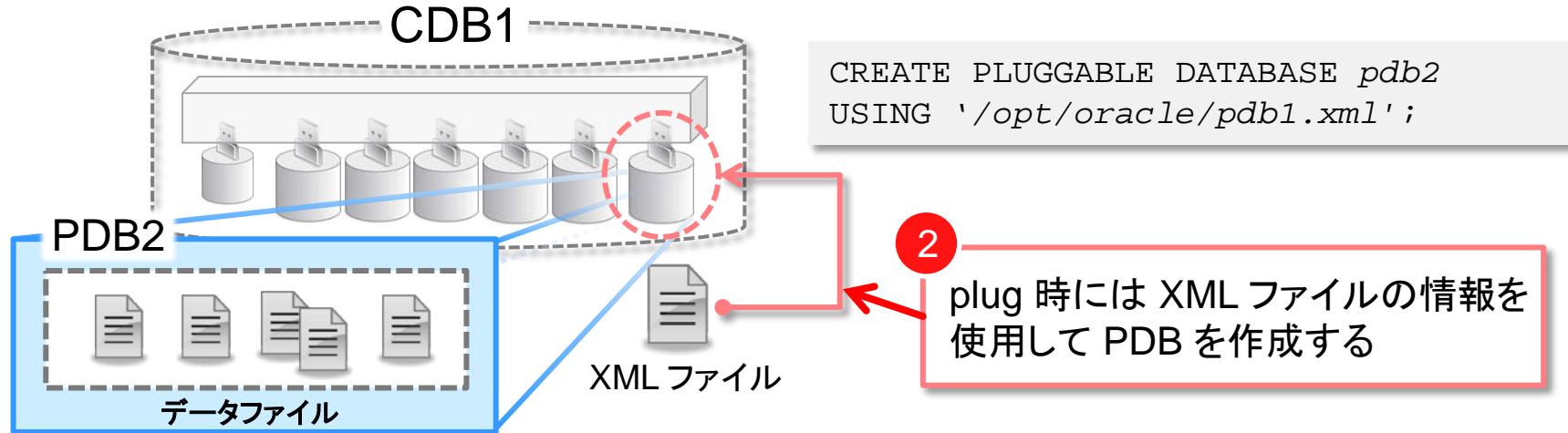
- Unplug 操作では PDB を CDB から切り離し、XML メタデータ・ファイルを作成する
- コマンドラインあるいは DBCA などのツールを使用可能
  - コマンドラインの場合は ALTER PLUGGABLE DATABASE 文を使用する



# プラガブル・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



# プラガブル・データベースの unplug と plug

## ステータスの確認

- CDB\_PDBS 表からステータスを確認可能
  - unplug した PDB は **UNPLUGGED** として表示する
  - plug した直後の PDB は **NEW** として表示する
    - 一度でもオープンした PDB のステータスは NORMAL と表示される
    - 下記は PDB2 を unplug して、PDB3 を plug した場合の表示例

```
SQL> SELECT PDB_NAME, STATUS FROM CDB_PDBS;
```

PDB_NAME	STATUS
PDB\$SEED	NORMAL
PDB1	NORMAL
PDB2	UNPLUGGED
PDB3	NEW

ORACLE®

# プラガブル・データベースの削除

## DROP PLUGGABLE DATABASE 文による削除

- 既存の PDB をデータベースから削除する
- 構文

```
DROP PLUGGABLE DATABASE <PDB_NAME> [<OPTIONAL_CLAUSE>];
```

### – 例

```
DROP PLUGGABLE DATABASE pdb1 INCLUDING DATAFILES;
```

- コマンドでの削除は PDB をクローズしておく（オープン中の削除操作は不可）
- 削除としては、制御ファイルにリストされているデータファイルの削除を実行
  - アーカイブ・ログおよびバックアップは削除の対象とはしない
    - Recovery Manager (RMAN) を使用して削除が可能

# プラガブル・データベースの削除

## 削除方法

- DBCA を使用する方法
  - 削除時のオプションは選択することはできない
    - データファイルを含めて削除を実行
  - ステータスがオープンの PDB に対しても削除を実行可能
    - DBCA による削除では、内部的にクローズしてから削除を実行しているため
- DROP PLUGGABLE DATABASE 文を使用する方法
  - 削除時のオプションを選択することが可能
    - データファイルを含めて削除
    - データファイルを含めずに削除
  - ステータスがオープンの PDB に対しては削除ができない

ORACLE®



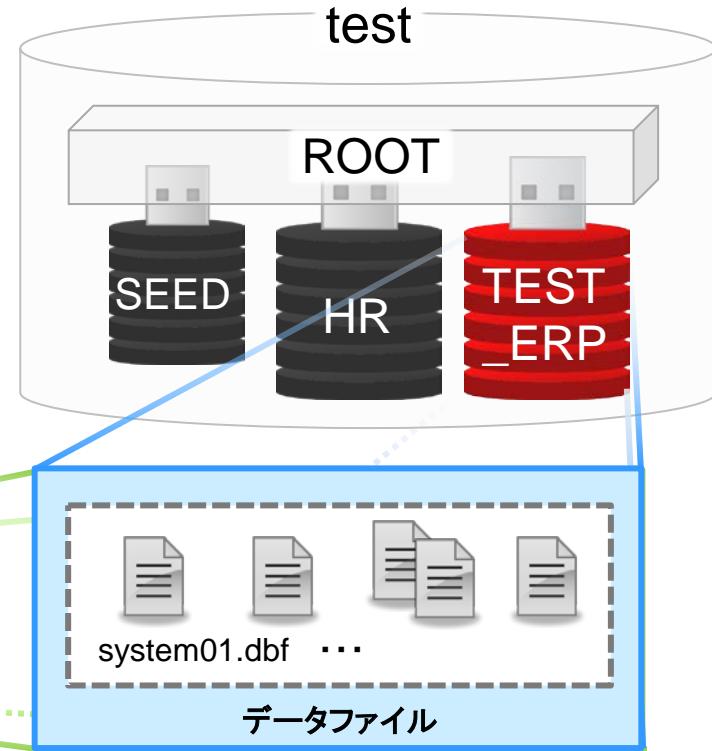
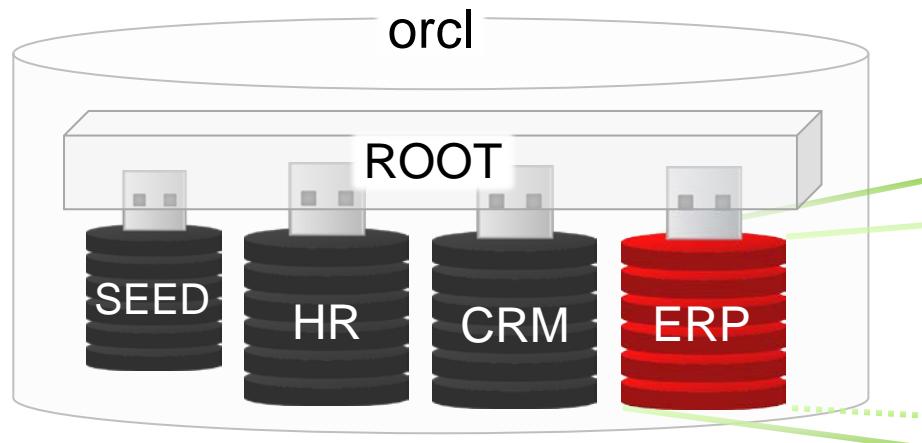
DEMONSTRATION

ORACLE®

# プラガブル・データベースの unplug と plug

## デモンストレーションの概要

- 異なる CDB 間で unplug / plug を実行する
  - plug 時に PDB 名を変更する
    - TEST\_ERP から ERP へ



# プラガブル・データベースの unplug

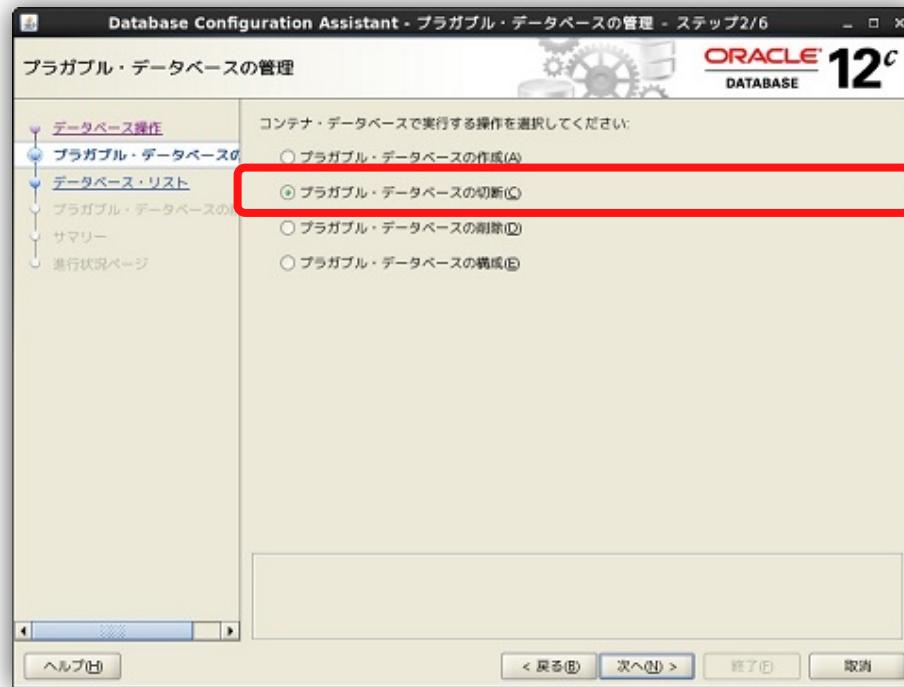
## 1. データベース操作



DBCA を起動して、「**プラガブル・データベースの管理**」を選択

# プラガブル・データベースの unplug

## 2. プラガブル・データベースの管理



「プラガブル・データベースの  
切断」を選択

ORACLE

# プラガブル・データベースの unplug

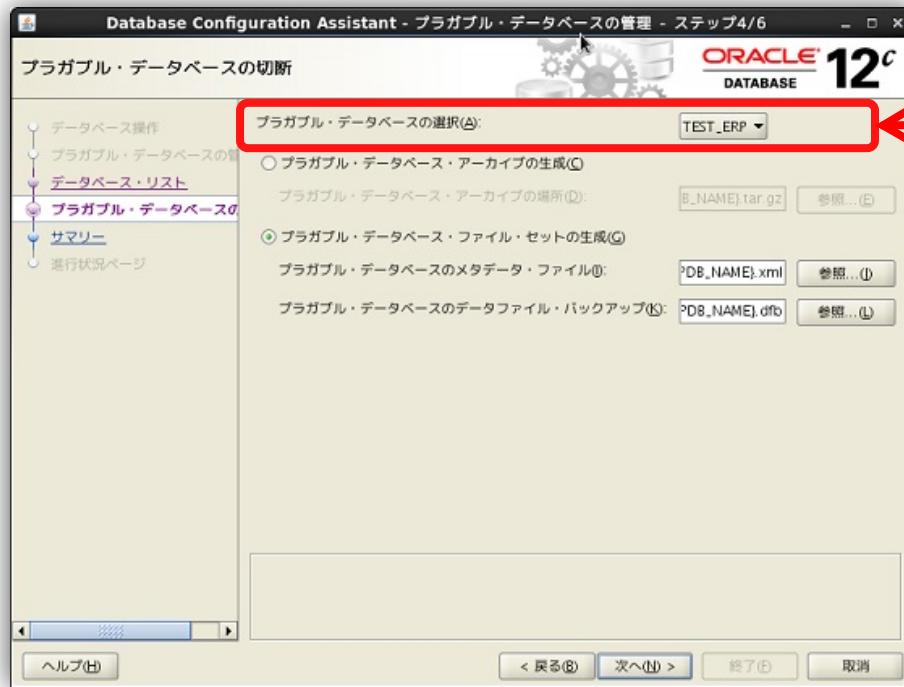
## 3. データベース・リスト



表示された中から、操作対象  
のデータベースを選択

# プラガブル・データベースの unplug

## 4. プラガブル・データベースの切断



プルダウンから、切断するプラガブル・データベースを選択

- XML メタデータ・ファイルとデータファイルのバックアップを作成

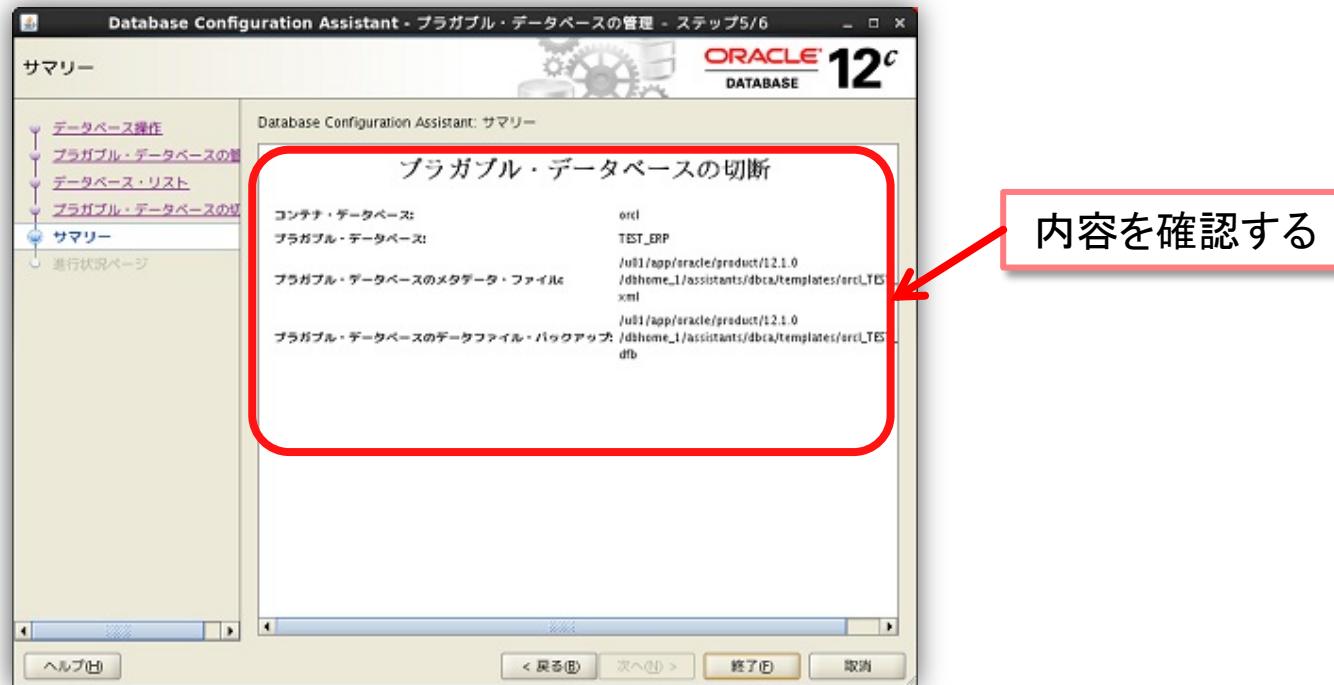
### 配置場所

`$ORACLE_HOME/assistants/dbca/templates`

ORACLE

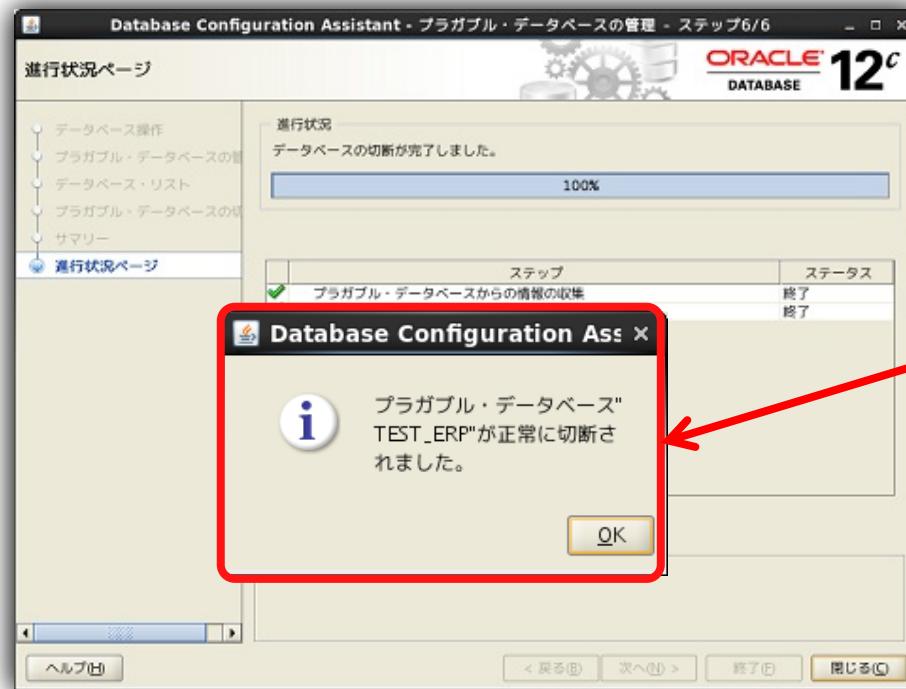
# プラガブル・データベースの unplug

## 5. サマリー



# プラガブル・データベースの unplug

## 6. 進行状況ページ



正常に切断されたことを確認

- 選択したプラガブル・データベースの切断と削除まで実行する

# プラガブル・データベースの unplug

## 確認

- DBCA による unplug 実行後の確認
  - プラガブル・データベースの削除まで実行されていることを確認

```
SQL> SELECT PDB_NAME, STATUS FROM CDB_PDBS;
```

PDB_NAME	STATUS
PDB\$SEED	NORMAL
HR	NORMAL

TEST\_ERP のエントリがないことを確認

- 生成されたファイルを確認

```
[oracle@node01]$ ll $ORACLE_HOME/assistants/dbca/templates/ |grep ERP
-rw-r----- 1 oracle oinstall 215015424 7月 10 22:20 2013 orcl_TEST_ERP.dfb
-rw-r----- 1 oracle oinstall 3696 7月 10 22:19 2013 orcl_TEST_ERP.xml
```

# プラガブル・データベースの plug

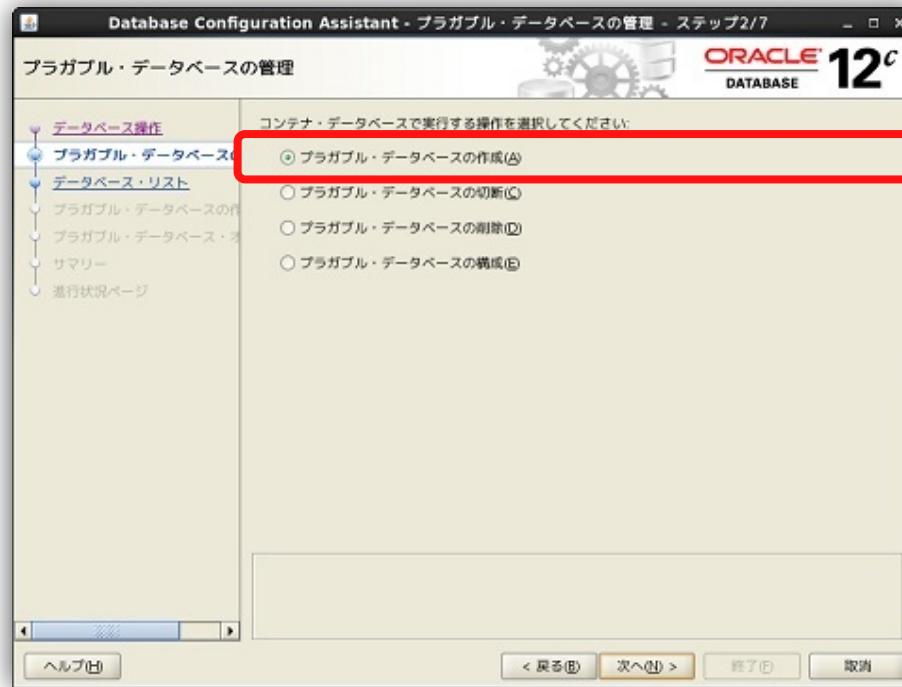
## 1. データベース操作



DBCA を起動して、「プラガブル・データベースの管理」を選択

# プラガブル・データベースの plug

## 2. プラガブル・データベースの管理



「プラガブル・データベースの作成」を選択

ORACLE

# プラガブル・データベースの plug

## 3. データベース・リスト

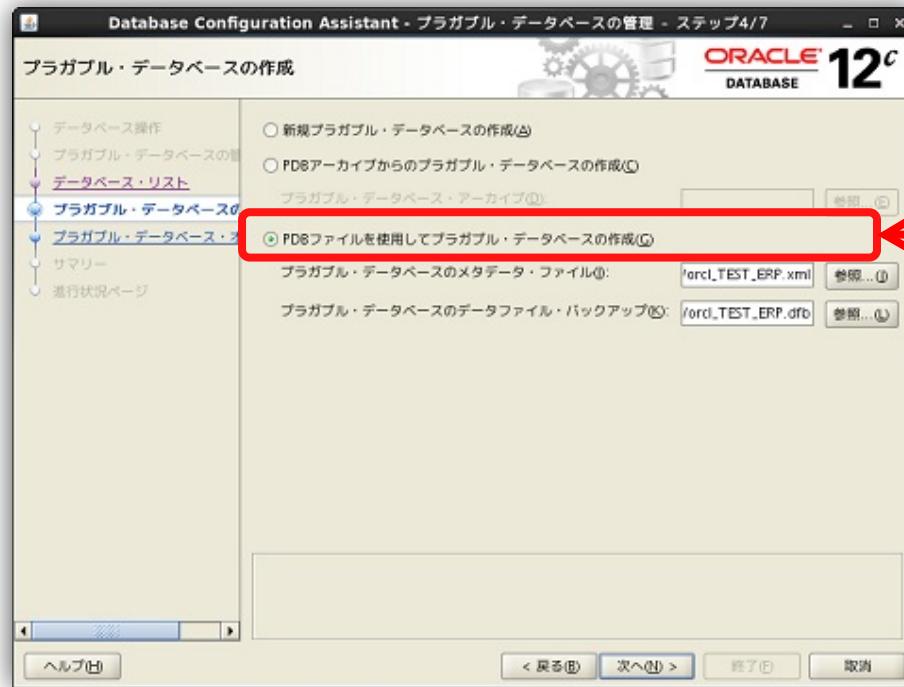


表示された中から、操作対象  
のデータベースを選択

ORACLE

# プラガブル・データベースの plug

## 4. プラガブル・データベースの作成

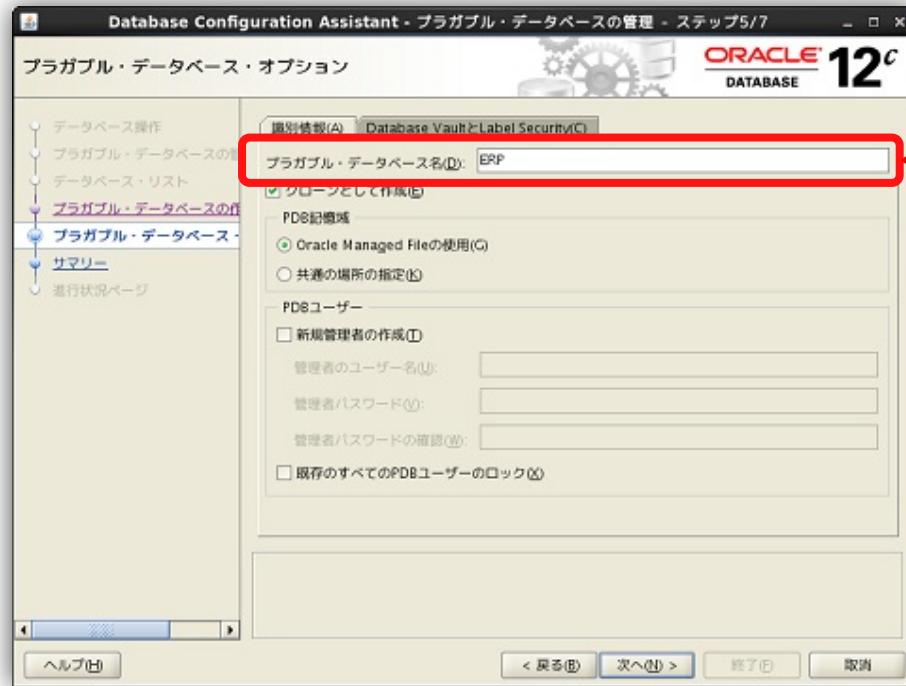


「PDB ファイルを使用してプラガブル・データベースの作成」を選択

- Unplug 時に作成した PDB の関連ファイルを選択する

# プラガブル・データベースの plug

## 5. プラガブル・データベース・オプション

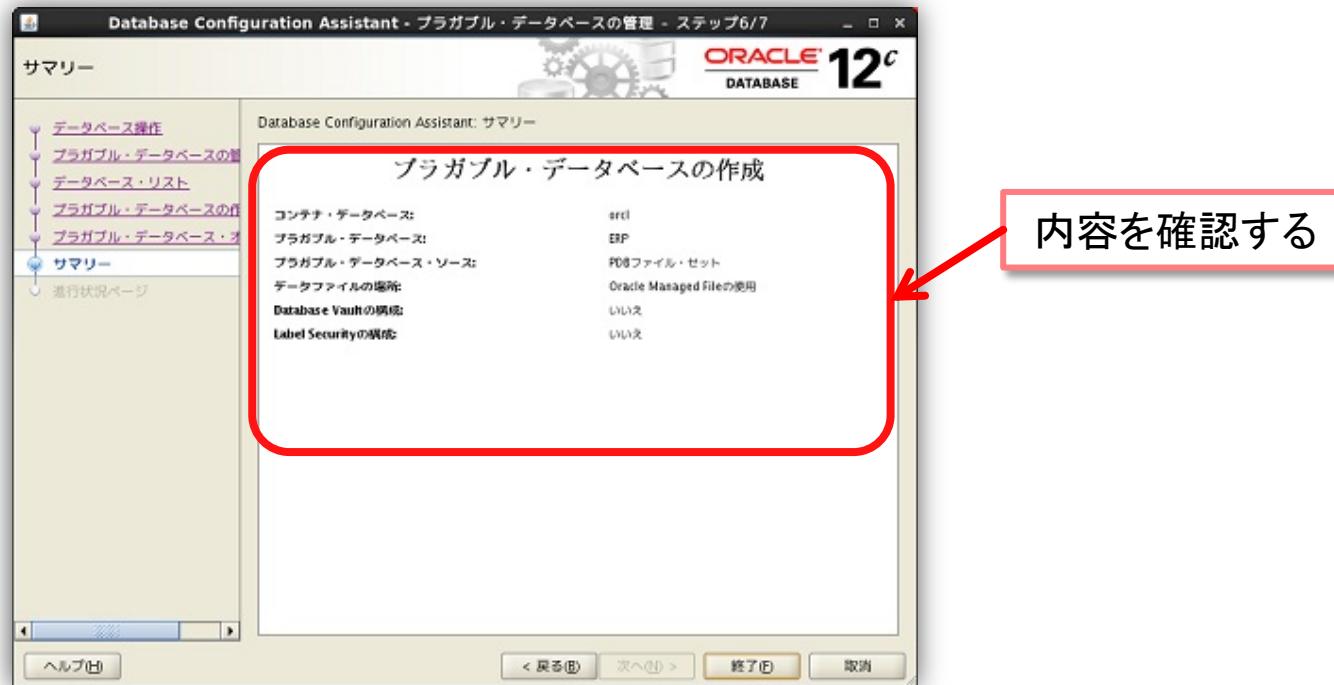


作成するプラガブル・データベースの名前を入力

- 今回は unplug 前と異なる PDB 名を入力する
- 必要に応じて管理ユーザーの再作成や既存のユーザーのロックが可能

# プラガブル・データベースの plug

## 6. サマリー



# プラガブル・データベースの plug

## 7. 進行状況ページ



正常に作成されたことを確認

- 作成したプラガブル・データベースはオープンされている状態

# プラガブル・データベースの plug

## 確認

- DBCA による plug 実行後の確認
  - プラガブル・データベースのオープンまで実行されていることを確認

```
SQL> SELECT PDB_NAME, STATUS FROM CDB_PDBS;
```

PDB_NAME	STATUS
PDB\$SEED	NORMAL
ERP	NORMAL
HR	NORMAL

- 同様の手順で、何度でもプラガブル・データベースの作成が可能
    - ユーザーの管理は DBCA を使用した作成時にカスタマイズが可能
-  ユーザー独自のテンプレートとして活用することも可能

# **Hardware and Software**

**ORACLE®**

# **Engineered to Work Together**

ORACLE®