



C'sNEXT

第3回どさんこオラクル勉強会

# Oracle Database 12c マルチテナント・アーキテクチャと プラグブル・データベース

Presented By シーズ・ネクスト

2015年12月17日



ORACLE®

Copyright © 2015



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

- マルチテナント・アーキテクチャの利点
- 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

# マルチテナント・アーキテクチャ の利点

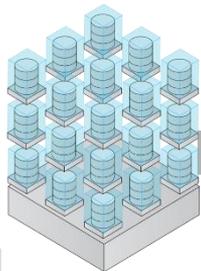
ORACLE<sup>®</sup> DATABASE 12<sup>C</sup>



Plug into the **Cloud**.

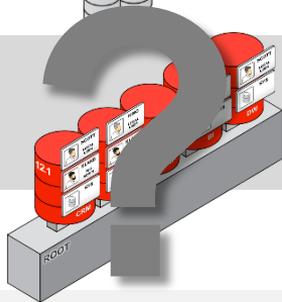
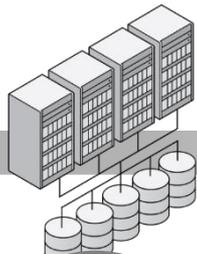
# データベース統合のシステム・アーキテクチャ

仮想マシンによる統合

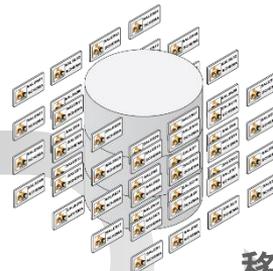


移行の負荷が低い  
システムごとの独立性に優れている  
リソースオーバーヘッドが大きい  
運用コストは減らない

統合基盤にDBを集約



スキーマ統合

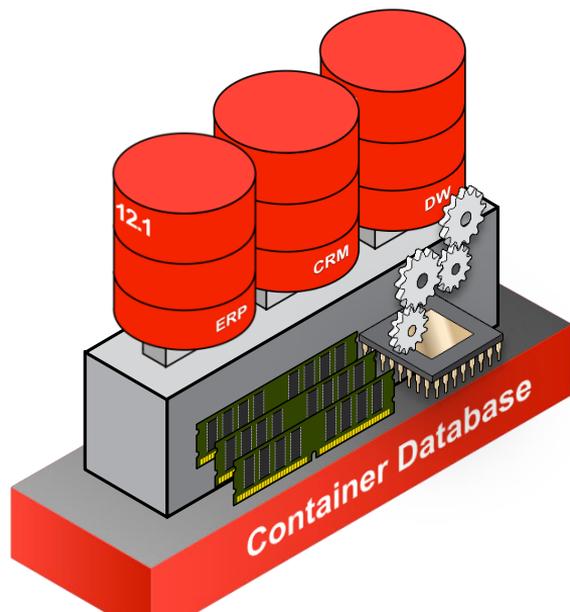


移行の負荷が高い  
システムごとの独立性が弱い  
リソース効率が高い  
運用コストを削減できる

- ✓ これまでの選択肢では移行の負荷とリソース効率のどちらか一方を選択せざるを得なかった
- ✓ ハードウェアを集約しても運用管理の負荷は下がらない

# マルチテナント・アーキテクチャ

運用コストを圧縮してシステムの独立性を保つ、プラグابل・データベースでの統合



## ● Pluggable Database (PDB)

- ✓ 仮想的なデータベース
- ✓ 1つのデータベースに複数作ることができる
- ✓ アプリケーションやデータベースの変更は不要

## ● Multitenant Container Database (CDB)

- ✓ Oracle Database の新しい型(タイプ)
- ✓ メモリ、CPU、制御ファイル、REDO などの管理要素はこのレベルで保持する

- ✓ 運用管理はCDBレベル、アプリケーション開発はPDBレベルに分割
- ✓ 移行作業の負荷を下げ、統合後の運用コストを削減するアーキテクチャ

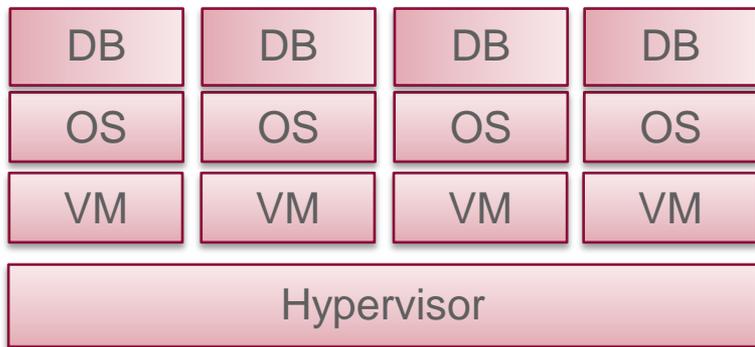
# データベース層でマルチテナントを実装するアドバンテージ

## 1. リソースのオーバーヘッドが少ない

より高密度な統合をシンプルに実装し、リソース効率が向上する

## 2. 運用管理の工数は大幅に削減

複数のシステムを1個のデータベースとして管理する「manage many as one」  
パッチ適用、アップグレード、バックアップなどの運用業務をシンプル化



サーバー仮想化によるマルチテナントの実装



データベースによるマルチテナントの実装

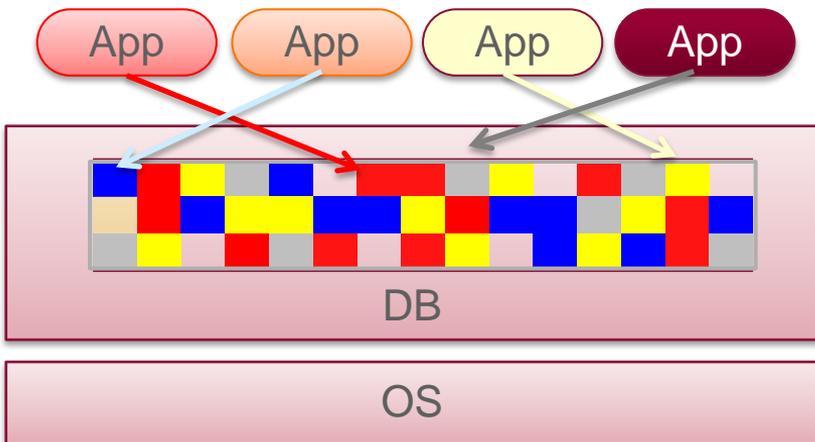
# データベース層でマルチテナントを実装するアドバンテージ

## 3. 統合の負荷が少ない

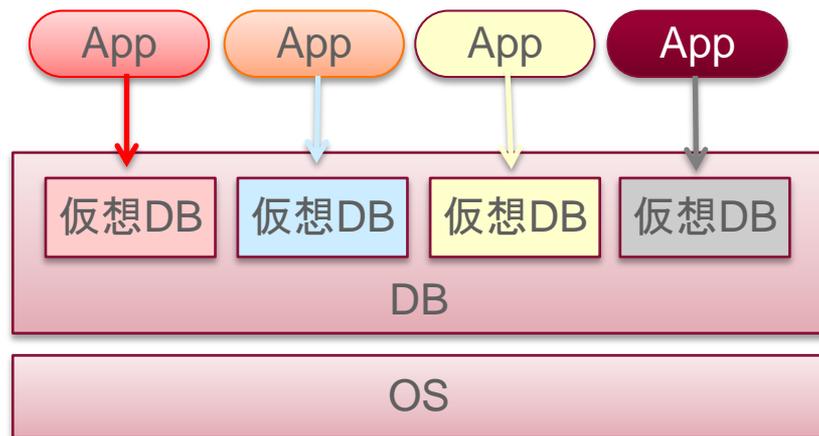
アプリケーションからは統合後のデータベースは統合前と同じに見える

## 4. システム間の独立性を保つ

データやユーザー・権限をシステムごとに分離、パッチ適用・アップグレード等での柔軟な選択肢



アプリケーションによるマルチテナントの実装



データベースによるマルチテナントの実装

# 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

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

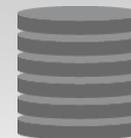


## 経理 DB

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)

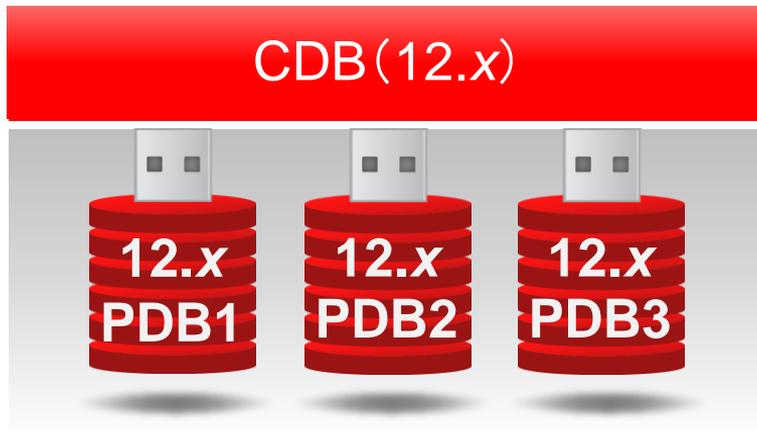


プラグブル・データベース  
(PDB)

# 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 →



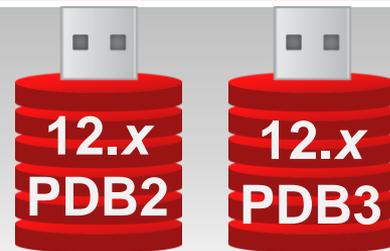
# 必要に応じたプラグブル・データベースの移動

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)

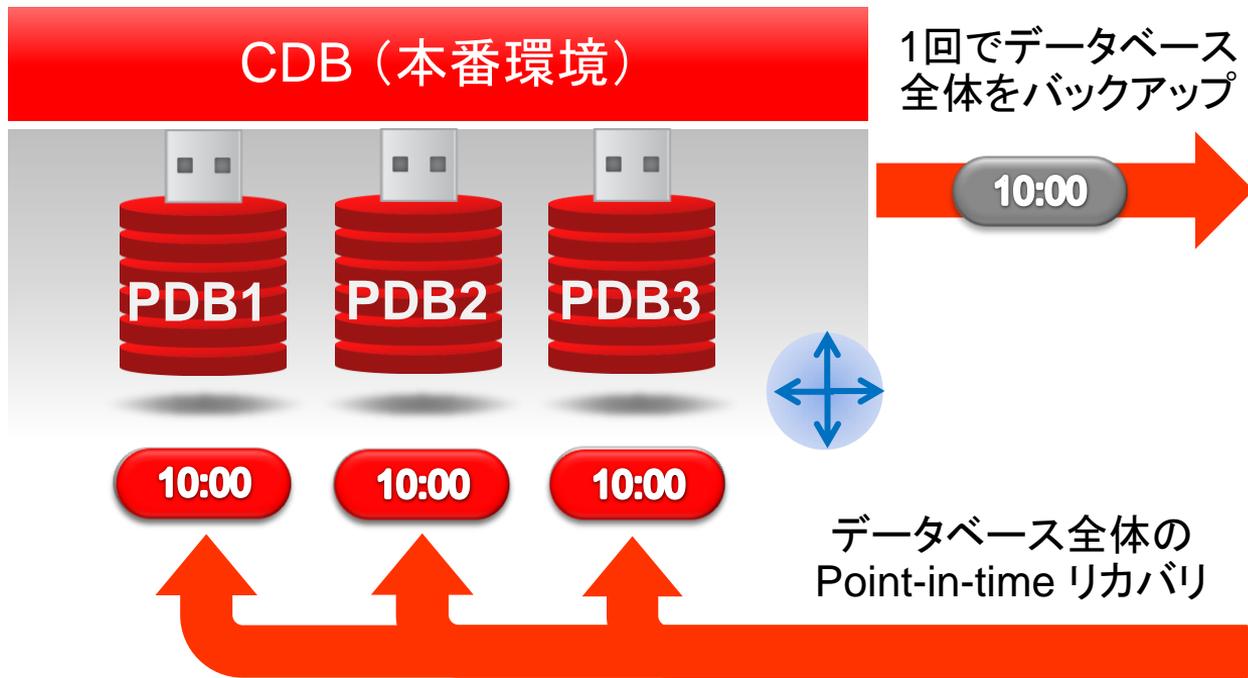


CDB2 (開発環境)



# 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<sup>®</sup>  
SUN ZFS STORAGE  
APPLIANCE

# 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



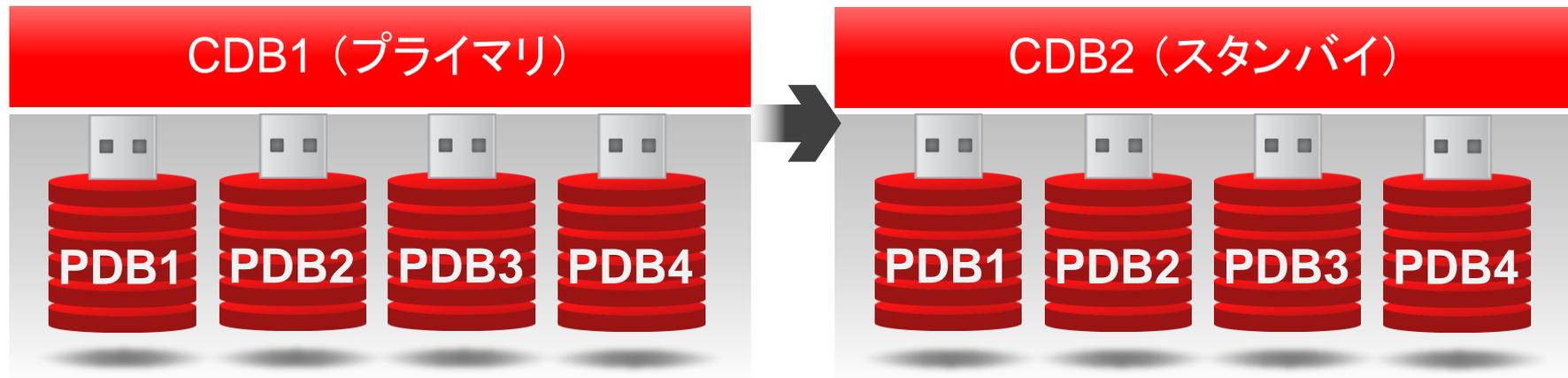
サーバー1



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



# リソースの効率利用

## サーバー・リソースの効率的な利用

いままでのサーバ統合

【低密度】

載せられるデータベースは多くない



マルチテナント・アーキテクチャ

【高密度】

より多くのデータベースを稼働できる



# マルチテナント・ アーキテクチャの概要

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



Plug into the **Cloud.**

# 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

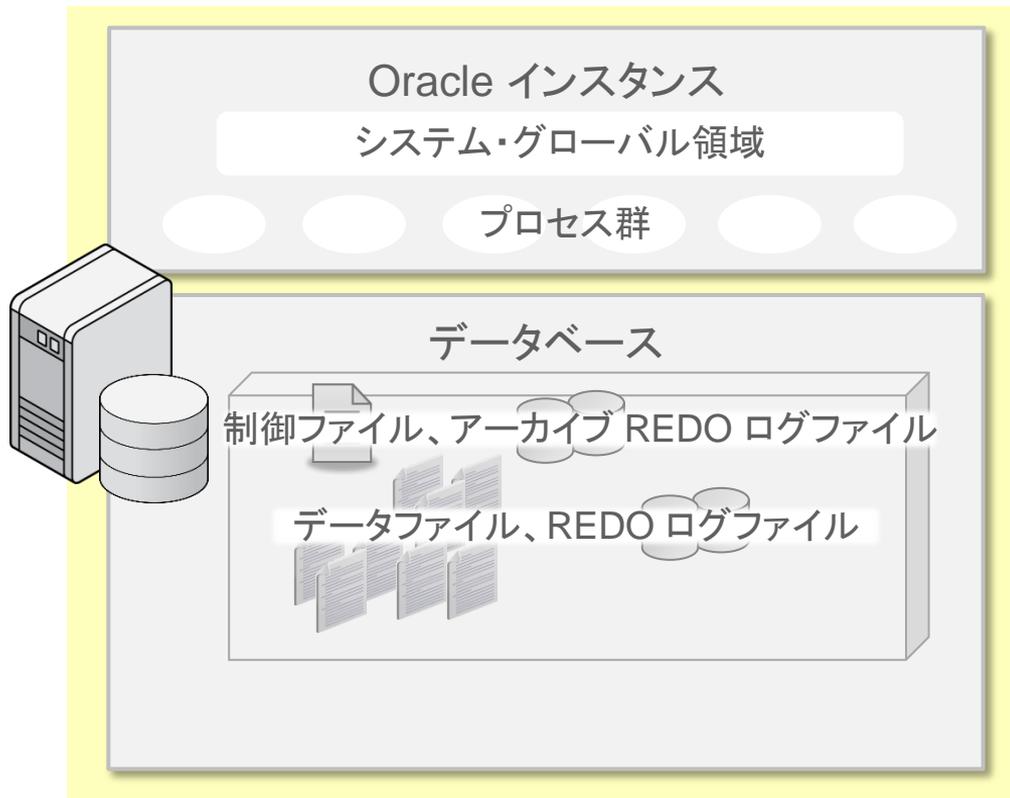
### マルチテナント・コンテナ・データベース (CDB) New

- 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

# 従来からのデータベース

## Non-CDB

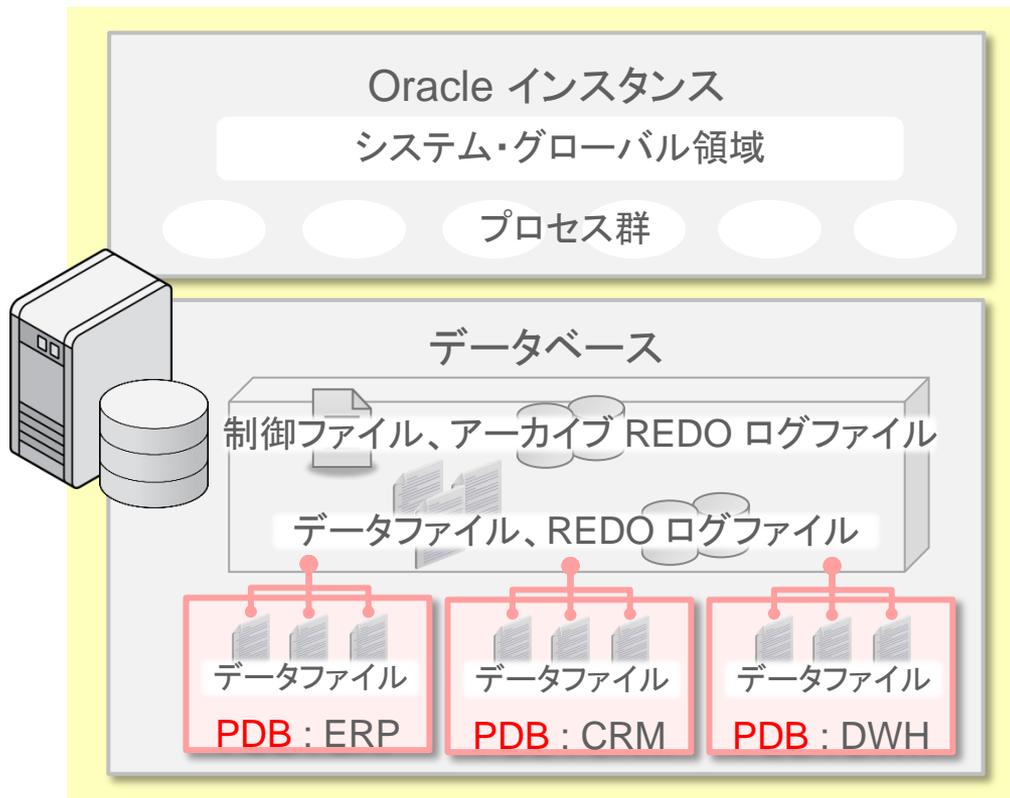
- Oracle データベースは、サーバー上で確保されるメモリ領域とプロセス群から構成される Oracle インスタンス と、ストレージに格納される データベース (ファイル群) により構成される



# 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



# 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
  - 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

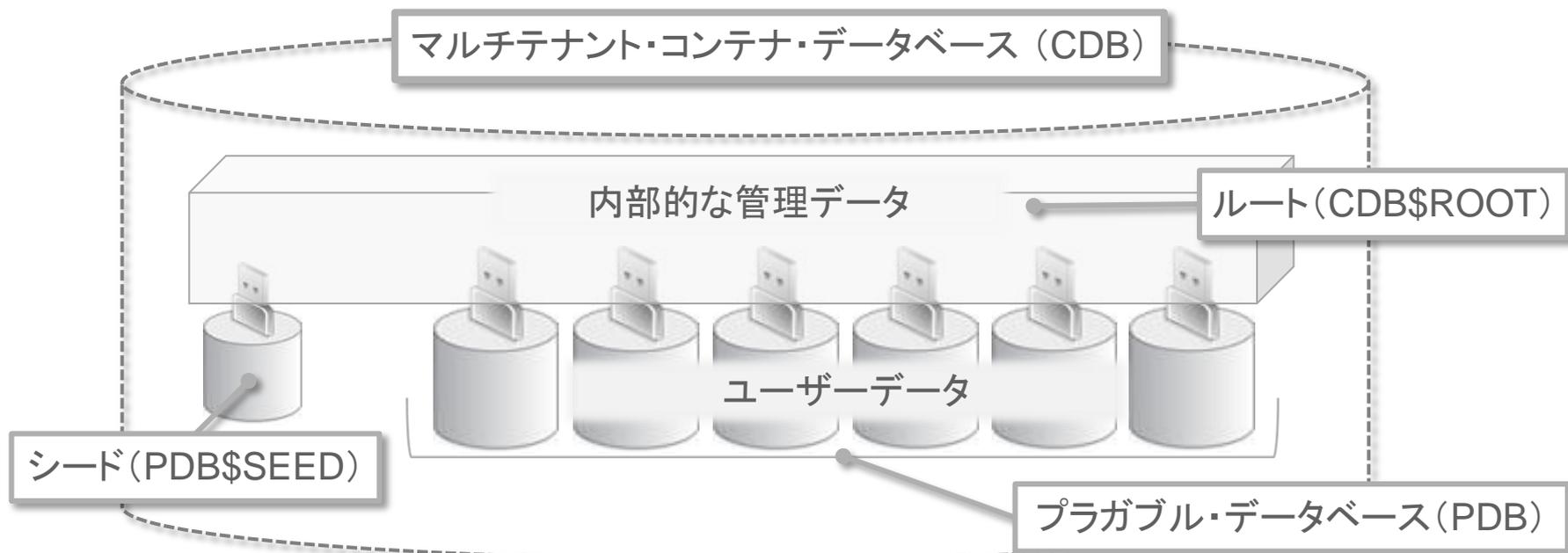
```
SQL> SELECT NAME, VALUE FROM V$PARAMETER WHERE NAME = 'compatible';
```

NAME	VALUE
-----	-----
compatible	12.1.0.0.0

# 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



# 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

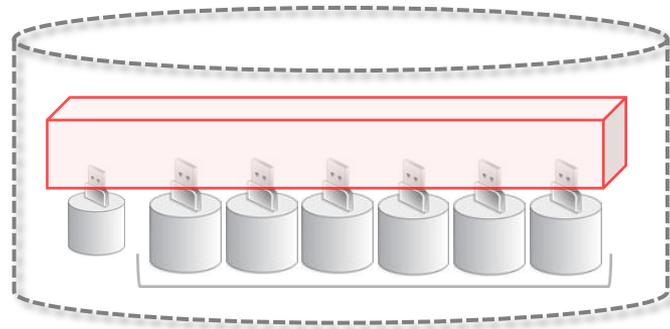
➤ いずれのデータベース構成においても Non-CDB や CDB が利用可能

# ルート (CDB\$ROOT)

## 根幹となるDatabase

### 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
- 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

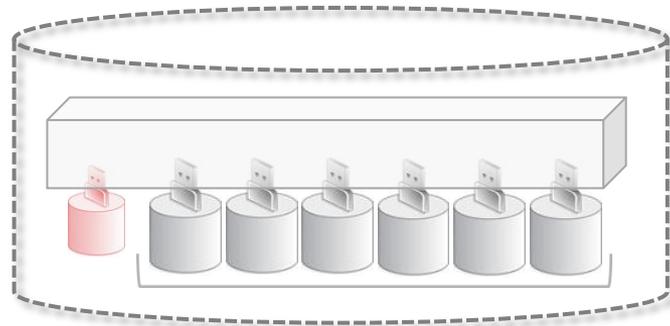


# シード (PDB\$SEED)

## プラグブル・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

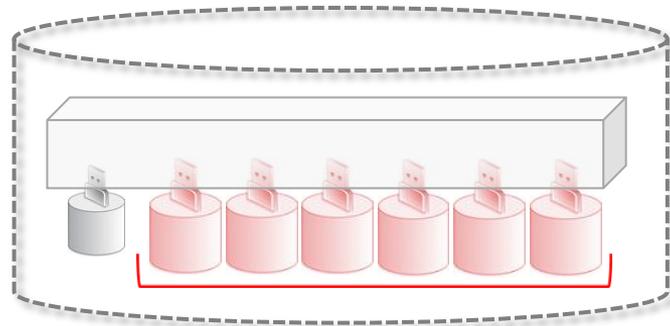


# プラグابل・データベース (PDB)

## 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)

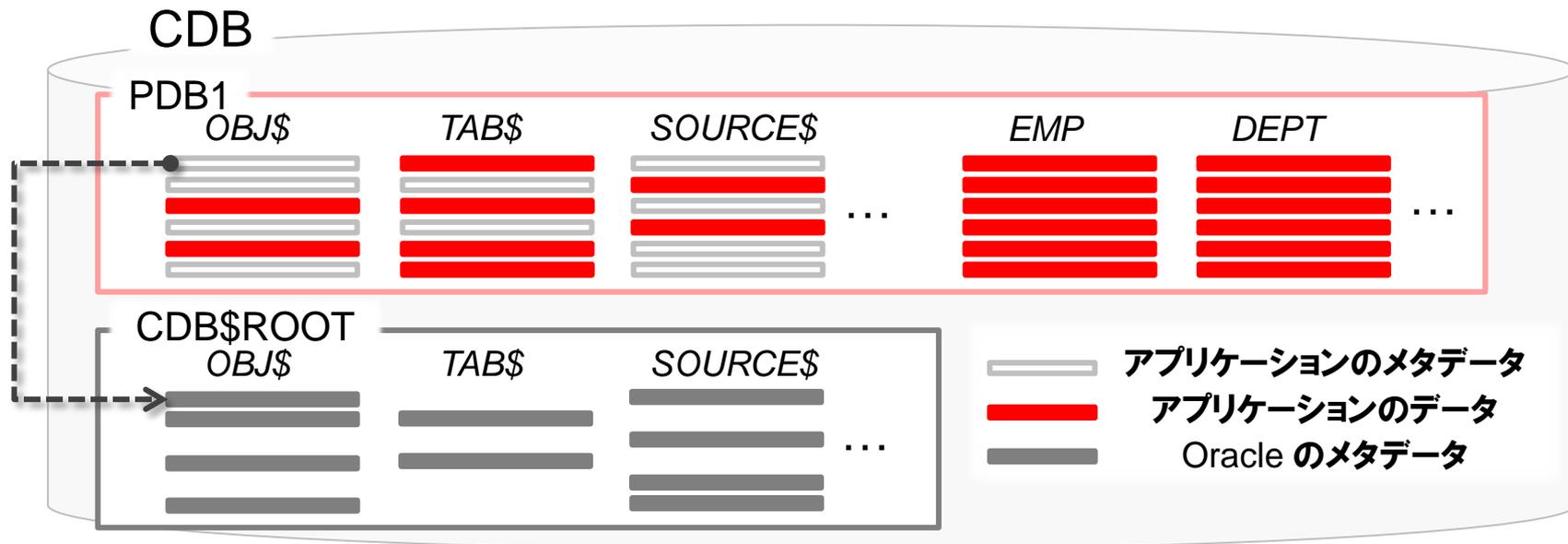
- スキーマや表領域を含む論理的なセット
- 同一 CDB 内に複数作成することが可能 (シードを除いて最大 252 まで)
- PDB 間は排他的な関係にあり、データは論理的に分離される
- CDB から取り外した後、再度取り付けたり、異なる CDB へ取り付けることが可能



# 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・ディクショナリの構造

## 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
<b>AUDIT_ACTIONS</b>	<b>2</b>

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
<b>AUDIT_ACTIONS</b>	<b>2</b>

USERS 表領域

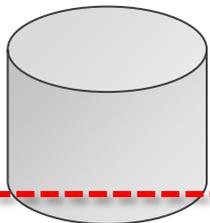
HR.EMP 表

ID	L_NAME
100	早坂
101	...

### CDB\$ROOT

SYSTEM 表領域

**SYS.OBJ\$ 表**



NAME	TYPE#
TAB\$	2
...	...
<b>AUDIT_ACTIONS</b>	<b>2</b>

- 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・ディクショナリの構造

## オブジェクトの共有と分離

### PDB\_SALES

SYSTEM 表領域

EXAMPLE 表領域

SYS.OBJ\$ 表

NAME	TYPE#
...	...
<b>PREF</b>	<b>2</b>
AUDIT_ACTIONS	2

SALES.PREF 表

P_ID	P_NAME
1	北海道
2	...

### PDB\_HR

SYSTEM 表領域

USERS 表領域

SYS.OBJ\$ 表

NAME	TYPE#
...	...
<b>EMP</b>	<b>2</b>
AUDIT_ACTIONS	2

HR.EMP 表

ID	L_NAME
100	早坂
101	...

### CDB\$ROOT

SYSTEM 表領域

SYS.OBJ\$ 表



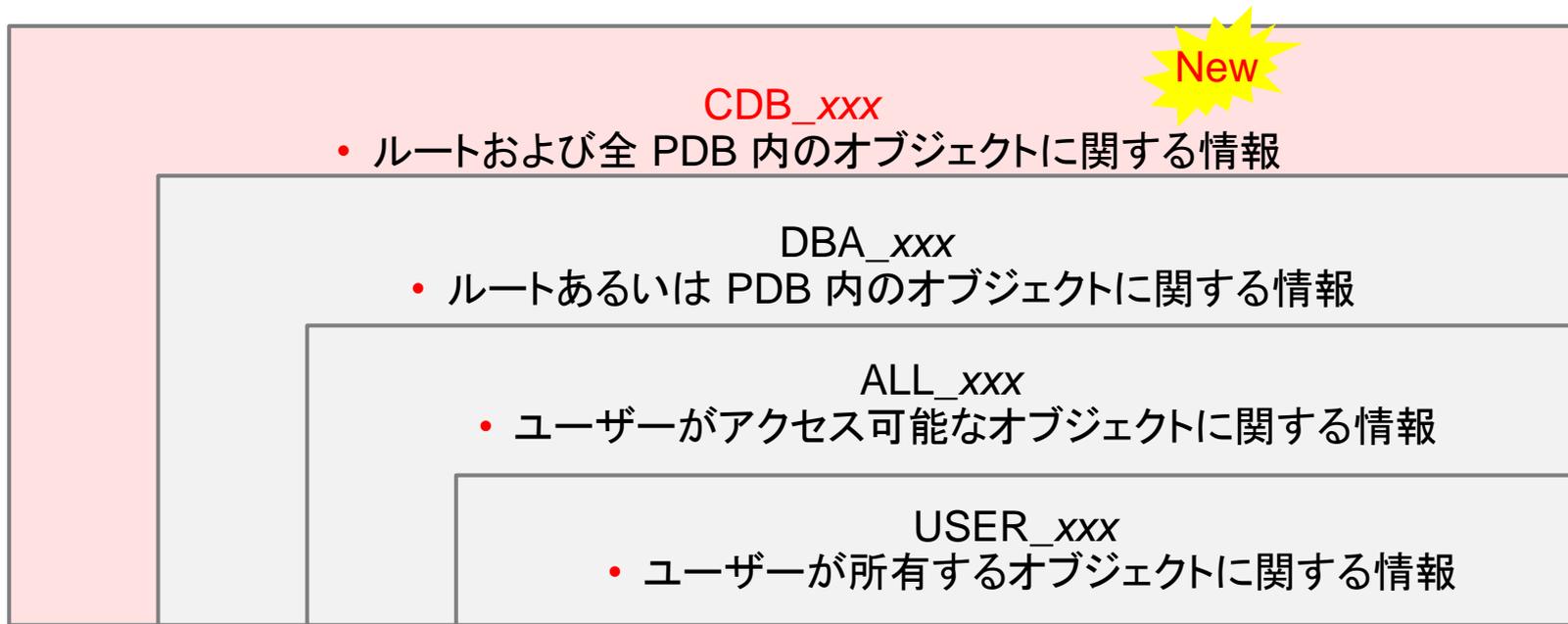
NAME	TYPE#
TAB\$	2
...	...
AUDIT_ACTIONS	2

- 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



# 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

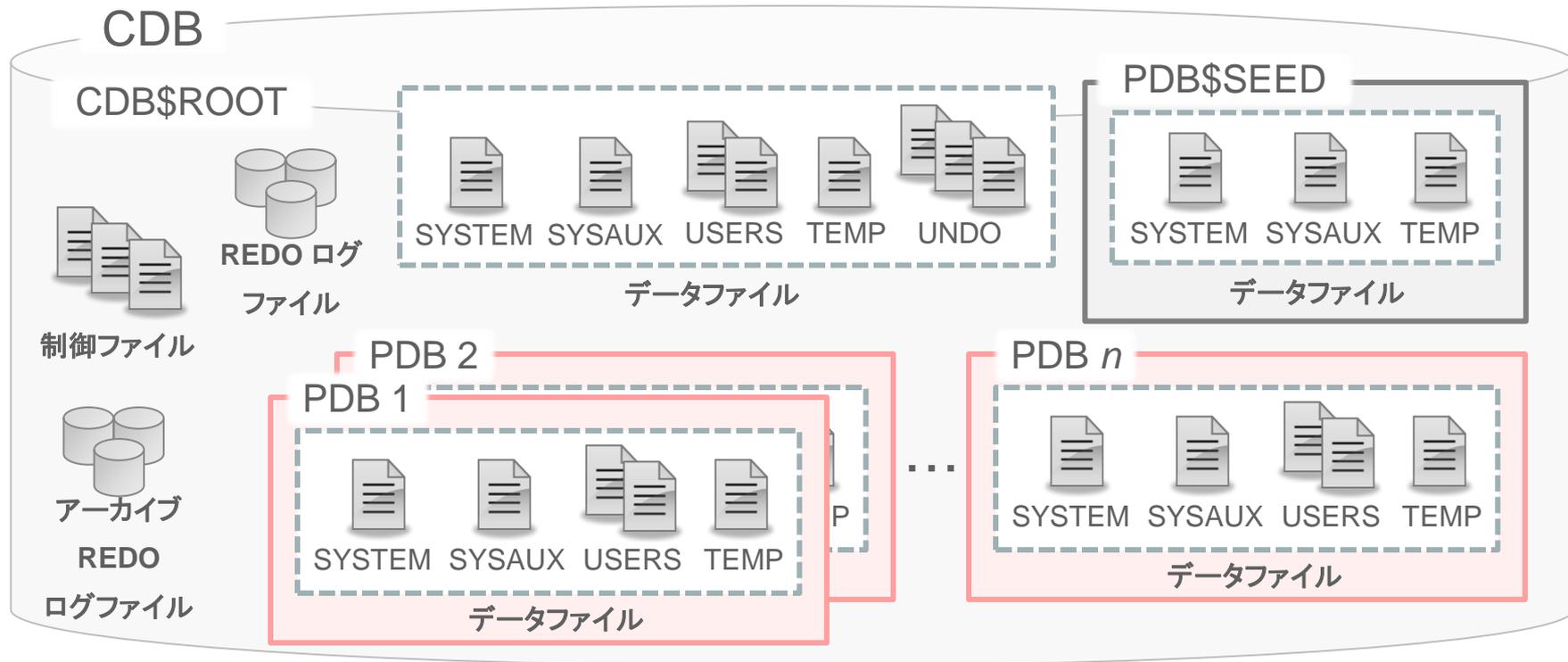
ルートの表領域

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

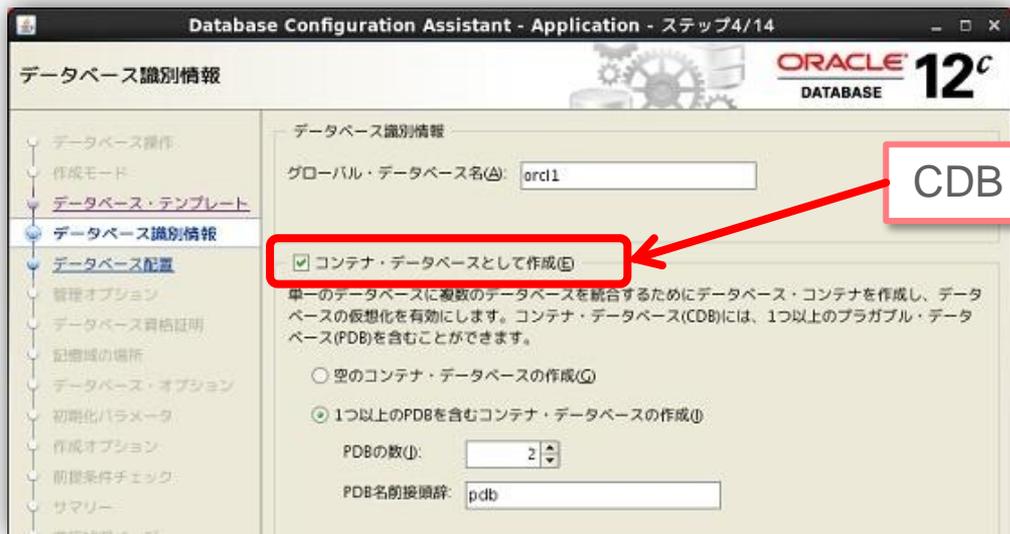
## データベース関連ファイル



# 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ابل・データベースの作成

## シードを使用した作成

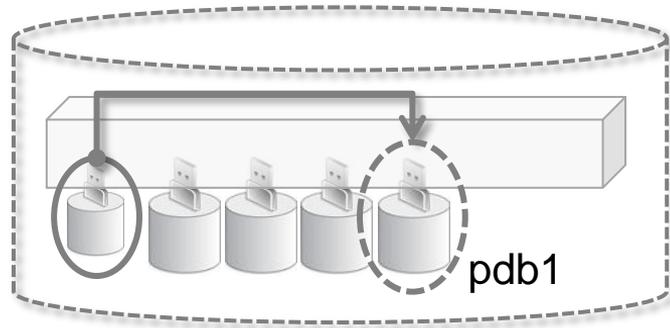
- シード (PDB\$SEED) から PDB を新規作成する場合の構文

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

### –コマンドの実行例

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

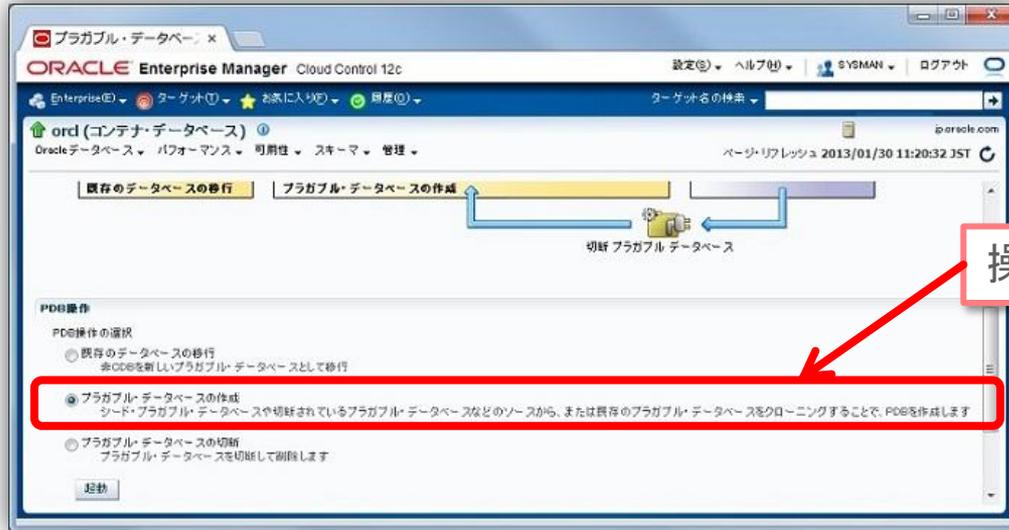
シードをコピーして、新しいプラグابل・データベースである *pdb1* を作成する



# プラグابل・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ابل・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)

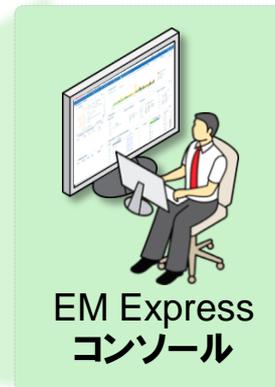
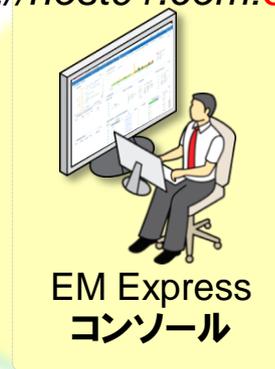
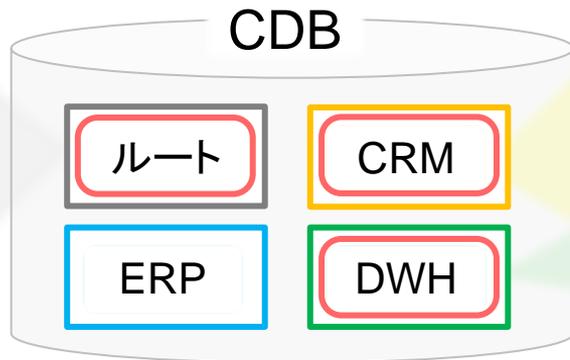
- 複数のサーバーを一元的に管理 (統合管理ツール)
- Cloud Control 用に用意したサーバー上に構成して使用する
- 管理対象のサーバーにエージェントを構成して使用する
- データベース管理のすべての機能を提供
- データベース以外にも、オラクル製品を中心に多彩な機能を提供

# 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、テナントごとに 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、対象のテナントに SYSDBA 権限で接続して行う
  - テナント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/>

# プラグブル・データベースの 基本操作

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



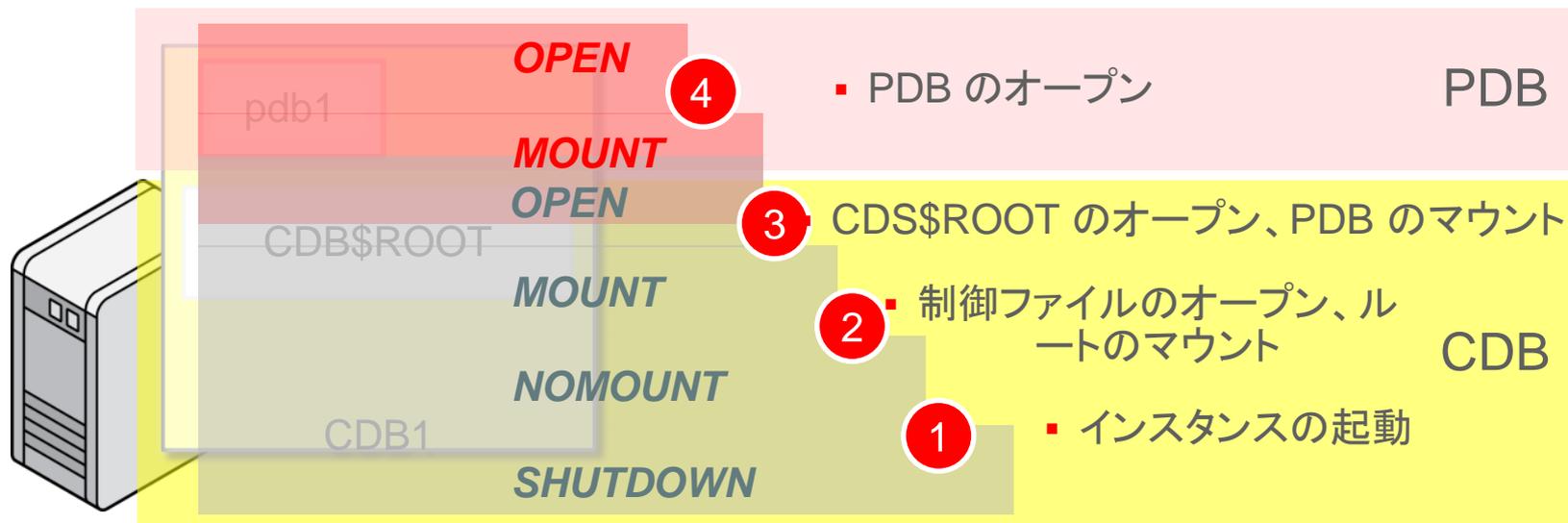
Plug into the **Cloud**.

# CDB 環境におけるデータベースの起動

## Oracle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ابل・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 ( PDB (pdb1) を、読み取り専用でオープンする場合)

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

- <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

# プラグブル・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

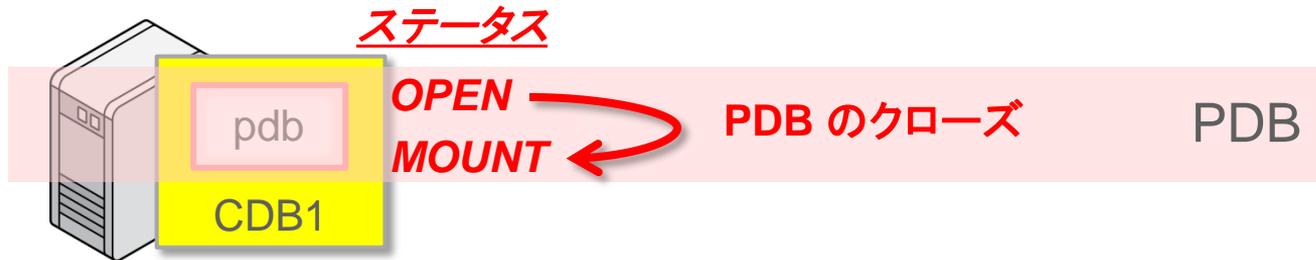
### UPGRADE

- 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>];
```

- コマンド例 (PDB (pdb1) を、即時停止する場合)

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

# プラグブル・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
  - 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
  - PDB のクローズにおける IMMEDIATE オプション
    - 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

# 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) PDB の自動起動

- 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、下記設定方法により作業の効率化が可能
  - トリガーによる設定(例: 12. 1. 0. 1)

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

- SQL文による制御(12. 1. 0. 2)

```
ALTER PLUGGABLE DATABASE ALL OPEN;  
ALTER PLUGGABLE DATABASE ALL SAVE STATE;
```

- 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



データベース起動時に特定の PDB も一緒に起動したい...

データベース  
管理者

# Cloud Controlを使ったCDB / PDB の基本操作

## データベース・ターゲット表示

ORACLE Enterprise Manager Cloud Control 12c

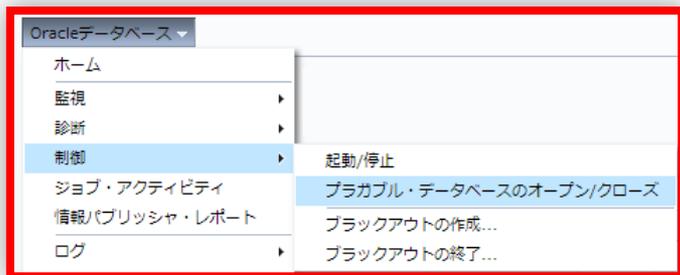
Enterprise(E) ターゲット(T) お気に入り(E) 履歴(Q)

データベース  
データベース・グループ  
システム  
サービス  
ホスト  
データベース  
ミドルウェア  
ビジネス・アプリケーション  
コンポジット・アプリケーション

名前	タイプ	ステータス	ターゲット・バージョン	インシデント			平均スコア
				警告	エラー	重大	
crm.oracle.com	データベース・インスタンス	↓	12.1.0.1.0	1	1	0	
dbcloud.oracle.com	データベース・インスタンス : コンテナ	↑	12.1.0.1.0	0	0	0	
ブラガブル・データベース		N/A		0	0	0	
dbcloud.oracle.com_CRM	ブラガブル・データベース	↑	12.1.0.1.0	0	0	0	
dbcloud.oracle.com_ERP	ブラガブル・データベース	↑	12.1.0.1.0	0	0	0	

# CDB / PDB の基本操作

## プラガブル・データベースのオープン / クローズ



### プラガブル・データベースのオープン / クローズ 画面

Enterprise Manager Cloud Control 12c

ターゲット(T) お気に入り(F) 履歴(O)

dbcloud.oracle.com (コンテナ・データベース)

Oracleデータベース パフォーマンス 可用性 スキーマ 管理

### プラガブル・データベースのオープン/クローズ

プラガブル・データベースを選択し、「アクション」メニューから実行するアクションを選択してください。

ターゲット	名前	状態	結果
dbcloud.oracle.com_CRM	CRM		
dbcloud.oracle.com_ERP	ERP		

アクション

アクション

- 開く
- 読み取り専用でオープン
- 閉じる

### PDB の状態

- オープン または  
 読み取り専用でオープン
- クローズ

- 複数プラガブル・データベースを選択して一括オープン / クローズ可能

# データベースへの接続

## SQL\*Plus を使用した接続方法

- SQL\*Plus を使用した接続方法

### – 接続記述子

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

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

### – EZCONNECT (簡易接続ネーミング)

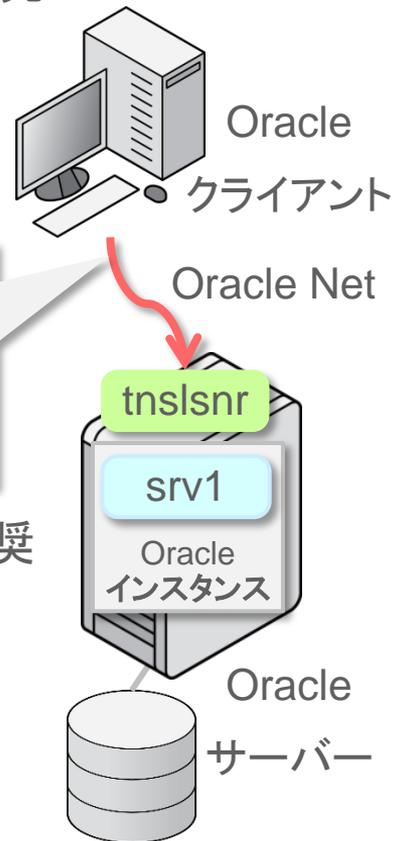
```
sqlplus <USERNAME>/<PASSWORD>@<HOSTNAME>:<PORT>/<SERVICE_NAME>
```

# 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

```
$ 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
```

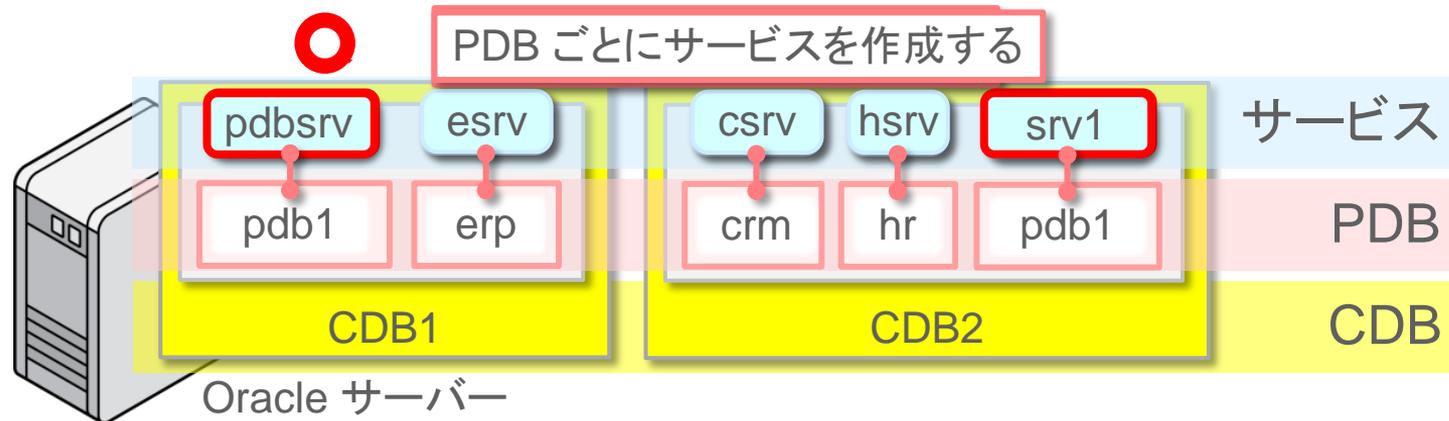
```
-----
```

```
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(Cluster)
  - 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
- ポリシー管理型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

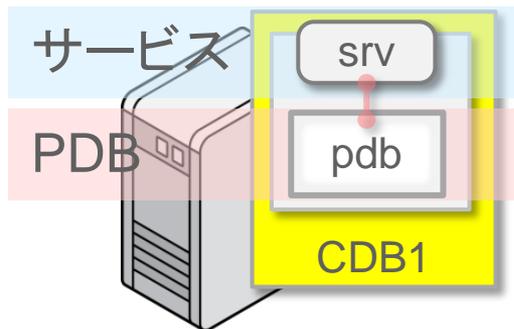
## 起動・停止における挙動

- 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

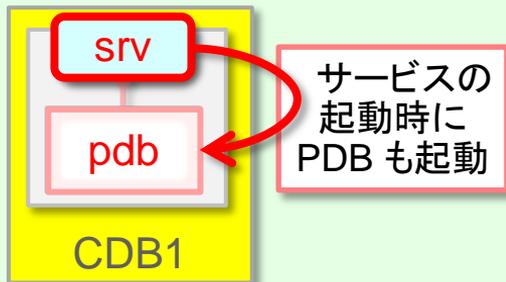
## 起動時

- 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

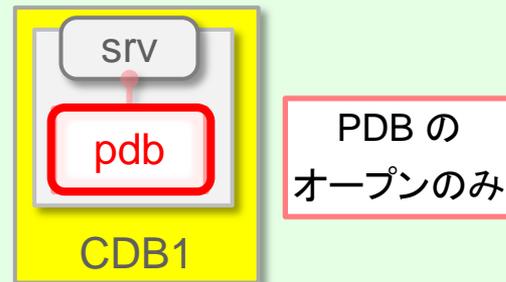
<現在の状態>



### 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

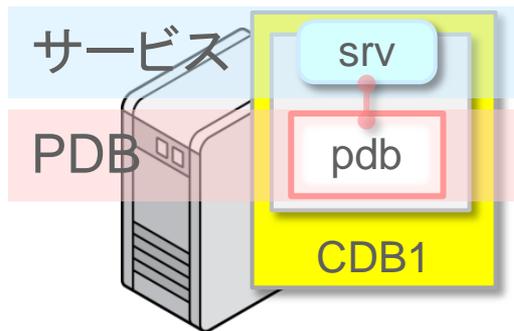
## 起動・停止における挙動

- 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

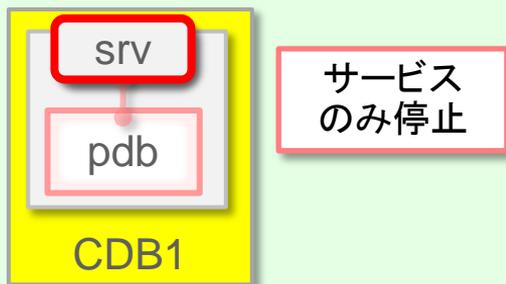
## 停止時

- 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

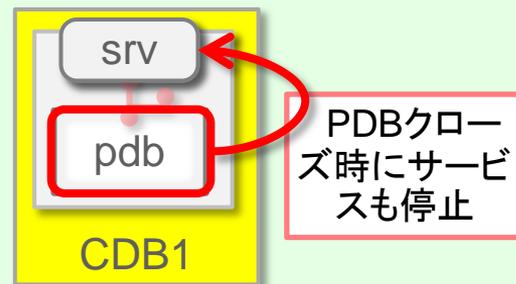
<現在の状態>



### A) サービスを停止する場合



### 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
  - 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
  - 例(PDB (pdb1)にサービス (pdb1\_srv1) を作成する場合)

```
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');
```

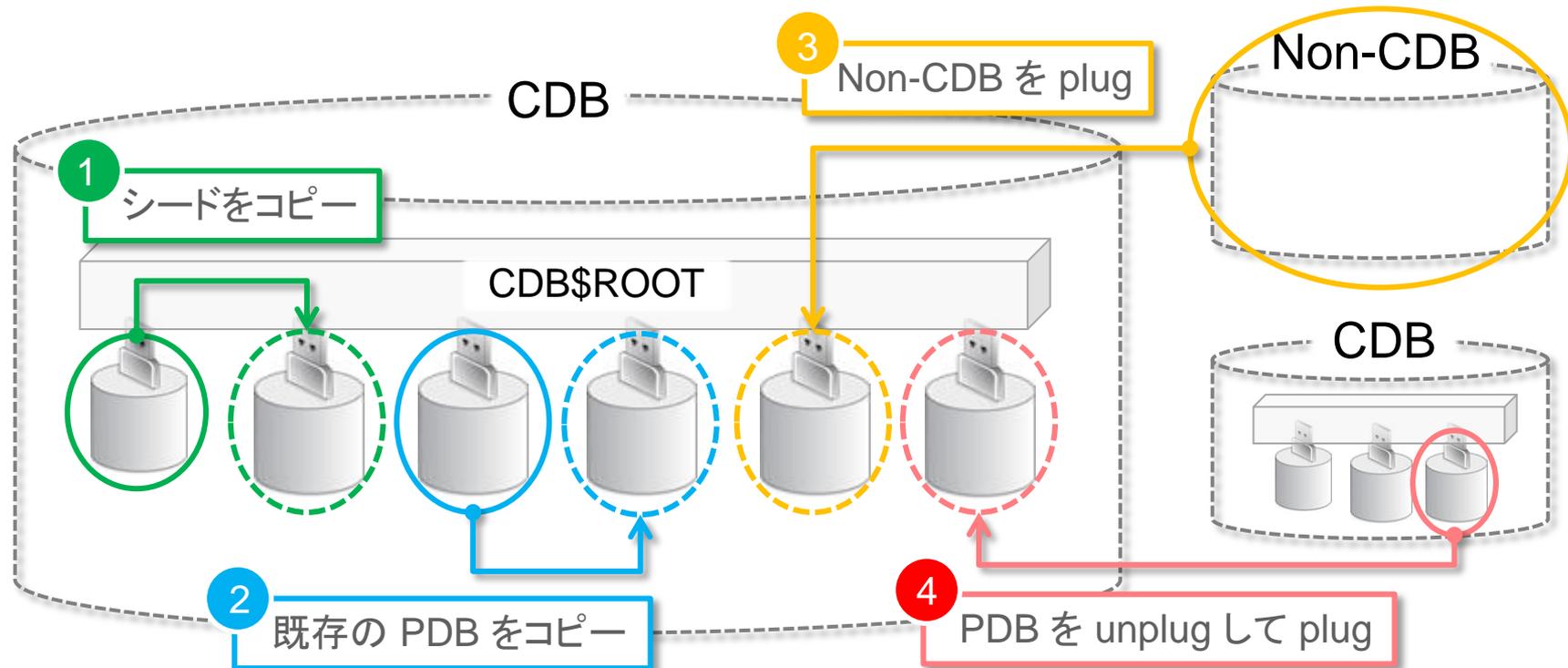
- サービスの削除

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

（詳細は「Oracle® Database PL/SQLパッケージおよびタイプ・リファレンス12c リリース1(12.1)」参照）

# 状況に応じたPDBの作成手法

コピーまたは unplug と plug による作成

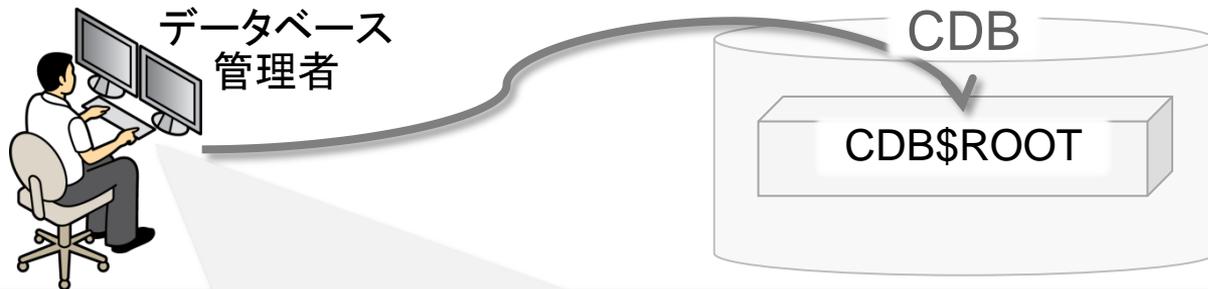


# プラグブル・データベースの作成

## 前提条件

- プラグブル・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
```

# プラグابل・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

# データファイル配置場所の指定

## 指定方法と優先度

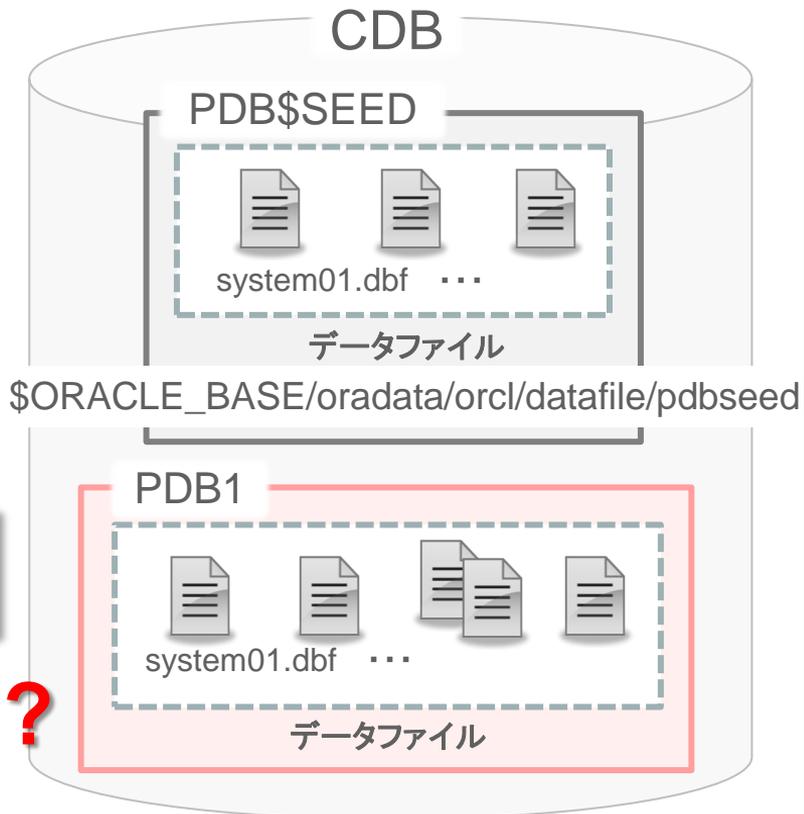
- データファイルの配置場所は、次のいずれかの方法で指定が可能
  1. FILE\_NAME\_CONVERT 句
  2. Oracle Managed Files (OMF)
  3. 初期化パラメータ PDB\_FILE\_NAME\_CONVERT
- 複数の方法を組み合わせた場合は上位の方法による指定が適用される
- PDB 関連のデータファイル管理の例
  - CDB を OMF 構成で作成し、PDB 関連のファイルも基本的には OMF に準拠して配置する
  - 開発用の PDB など例外的に配置場所を変更したい場合には、作成時に FILE\_NAME\_CONVERT 句を使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

PDB\$SEED  
をコピーする



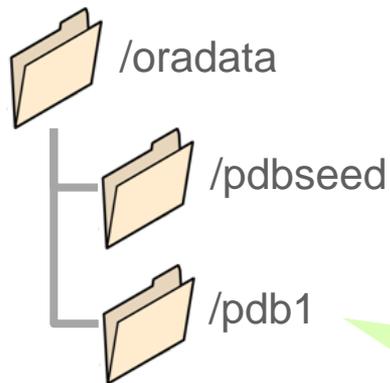
# FILE\_NAME\_CONVERT 句の活用

## 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   pdb1_users01.dbf
```

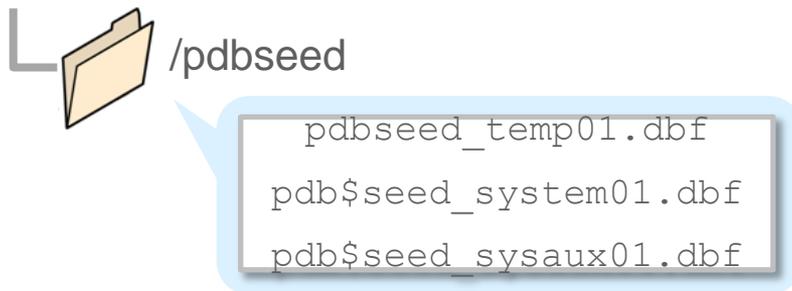
# FILE\_NAME\_CONVERT 句の活用

## ファイル単位での配置場所の指定

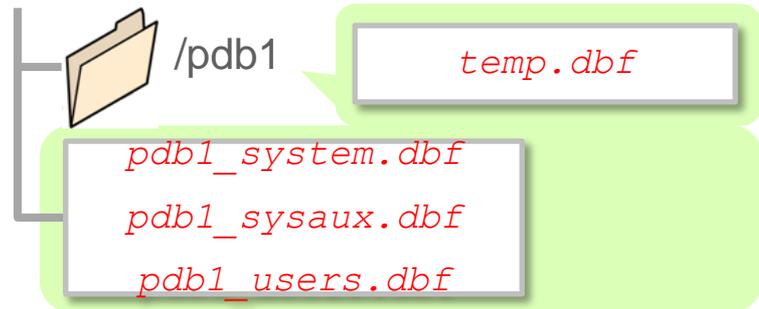
- ファイルごとに配置を指定する場合のコマンド例と配置イメージ

```
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



# 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てすることも可能
- 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');
```

# プラグابل・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>];
```

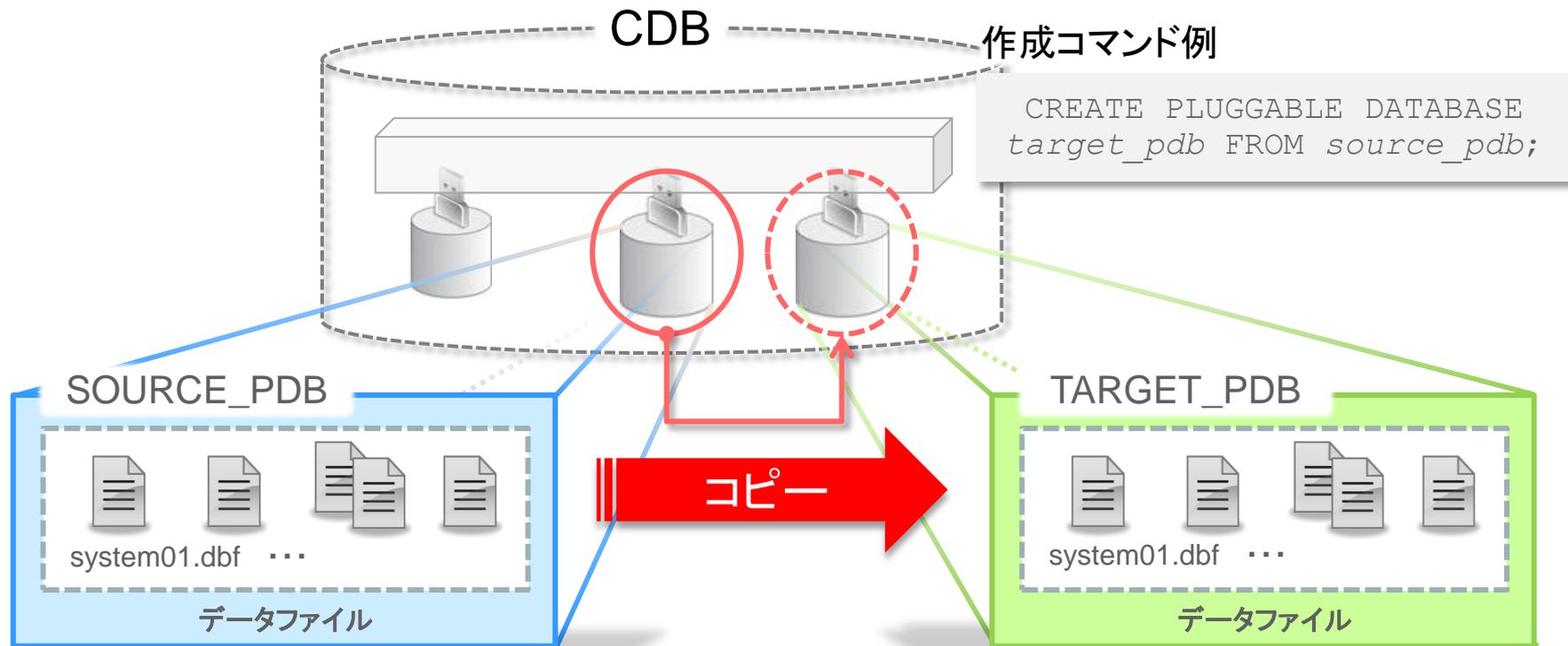
– 例 (hrpdb から testpdb を新規に作成する場合)

```
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

# 既存のプラグابل・データベースを使用した作成

同一 CDB 内 (ローカル) の場合



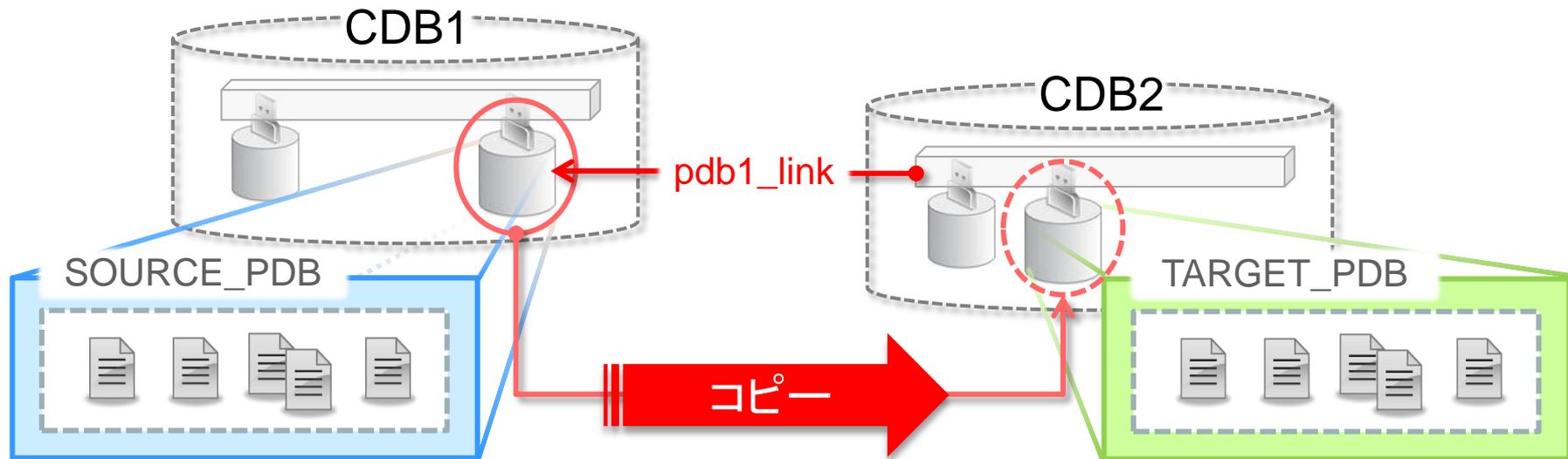
# 既存のプラグابل・データベースを使用した作成

## 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;
```



# 異なる CDB 間でのプラグラブル・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';
```

} データベース・  
リンクを作成

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

# 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 作成の遍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, OPERATION
        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

# プラグابل・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、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

# 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
  - noncdb\_to\_pdb.sql スクリプトを実行し、別途オープンを行う

# Data Pump を使用した作成

## 非CDBからデータをエクスポートし、PDBにインポート

- インポート時に、ユーザー名の後にPDBの接続識別子を指定
  - PDBの接続識別子がhrpdbである場合
    - `impdp user_name@hrpdb ...`
- 非CDBのOracle Databaseが11.2.0.3以上の場合
  - フル・トランスポートابل・エクスポート/インポートを使用して、データを移動可能
  - VERSIONパラメータを12.0.0.0.0以上に設定する必要がある
    - `transportable=always version=12 full=y`
- 非CDBのOracle Database(11.2.0.3)より前の場合、
  - トランスポートابل表領域を使用してデータを移動する
  - または完全なデータベース・エクスポート/インポートを実行
- GoldenGateレプリケーションを使用
  - 非CDBのデータをPDBにレプリケート
  - PDBが非CDBに追いついている場合は、そのPDBにフェイルオーバー可能

# プラグブル・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>;
```

#### – 例

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

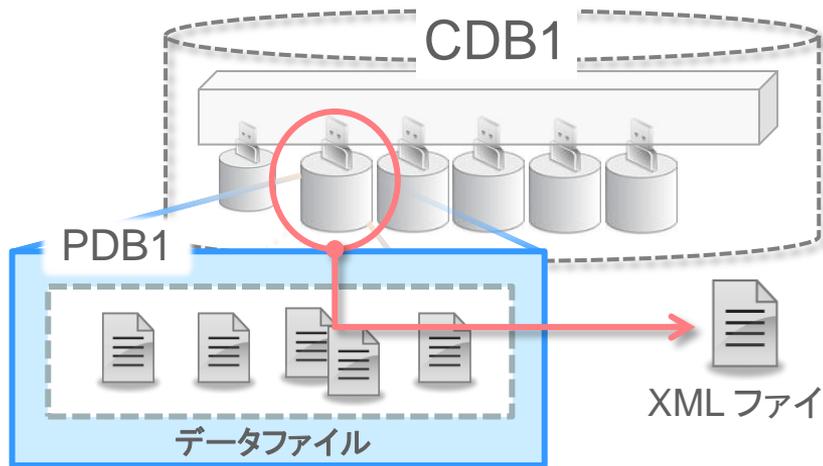
### plug

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

# 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



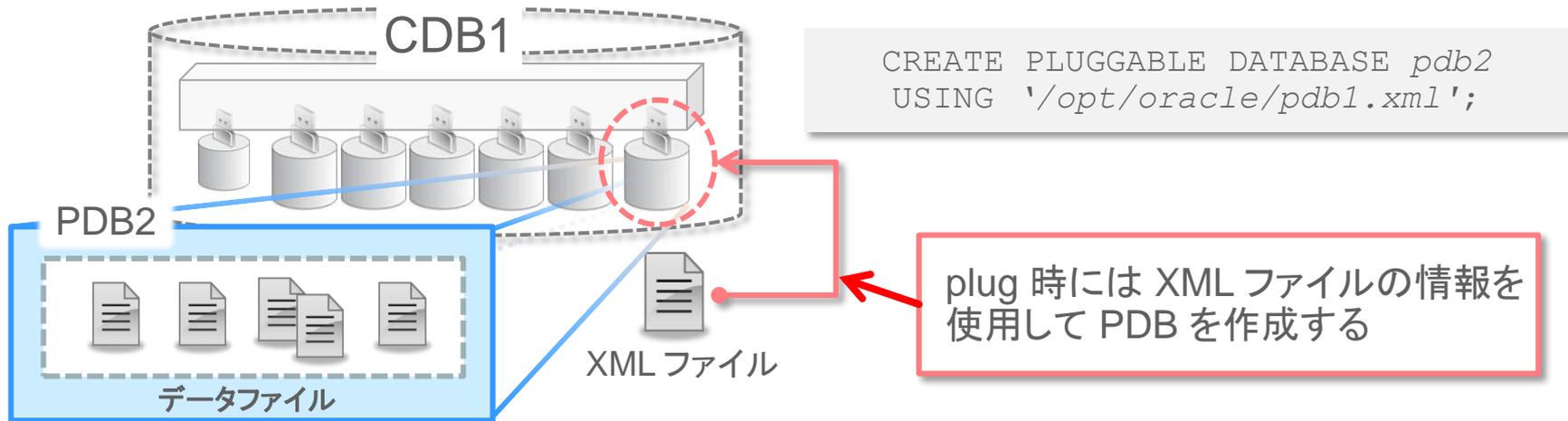
```
ALTER PLUGGABLE DATABASE pdb1  
UNPLUG INTO '/opt/oracle/pdb1.xml';
```

unplug 時に XML ファイルを生成する

# プラグابل・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



# プラグابل・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
    - 例 ( PDB2 を unplug して、PDB3 を plug した場合)

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

```
PDB_NAME                                STATUS
```

```
-----
```

```
PDB$SEED                                NORMAL
```

```
PDB1                                    NORMAL
```

```
PDB2                                    UNPLUGGED
```

```
PDB3                                    NEW
```

# プラグブル・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

# プラグブル・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 (pdb1をデータファイルも含めて削除する場合)

```
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)
- DROP PLUGGABLE DATABASE 文は、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rchive・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

# プラグブル・データベースの 管理

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



Plug into the **Cloud.**

# 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 30 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

SYSTEM

SYS

PDB 2

PDB 1

SYS

USER01

SYSTEM

HRADM

PDB n

SYS

USER01

SYSTEM

ERPADM

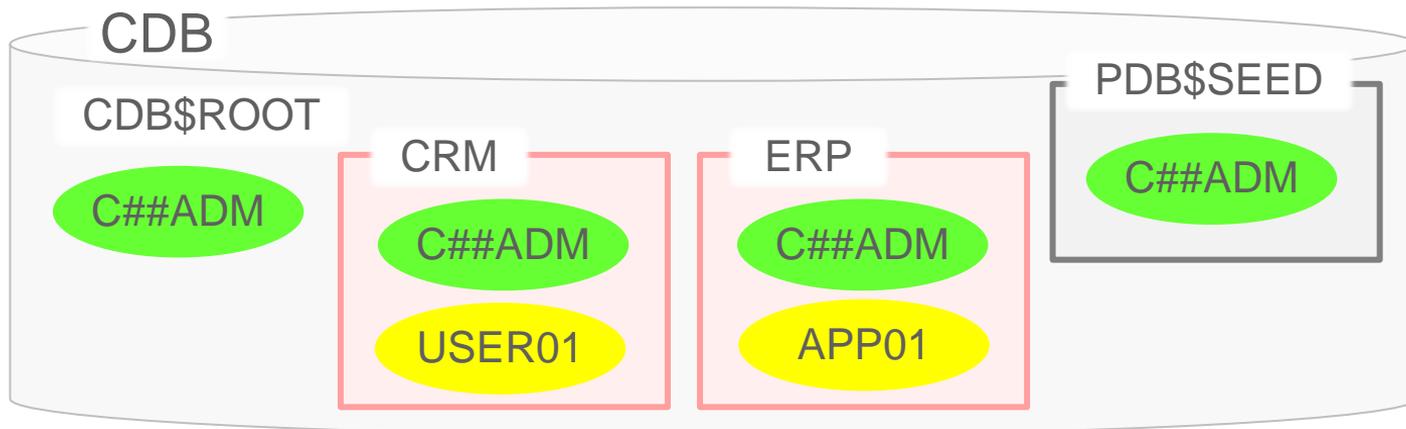
他の PDB とユーザー名は重複してもよい



# 共通ユーザーの管理

## プラグブル・データベースを unplug あるいは plug する場合

- PDB を unplug する場合、ローカル・ユーザーのみ設定を保持する
- PDB を plug する場合には、接続する先の CDB の設定（共通ユーザーの作成状況）が反映される



# 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

# 初期化パラメータ

## PDB ごとに設定が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
- 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
    - 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

# マルチテナント・アーキテクチャ関連のパラメータ (1)

## ENABLE\_PLUGGABLE\_DATABASE

- データベース作成時に CDB であるかどうかを示す設定値

値の範囲	説明
TRUE	CDB であることを示す
FALSE (デフォルト)	Non-CDB であることを示す

- DBCA による CDB 作成時には、自動的に TRUE が設定される
- CREATE DATABASE 文による作成時には、手動で設定が必要
  - CDB 作成前に init.ora ファイルへ TRUE を設定する
  - ENABLE PLUGGABLE DATABASE 句を指定する

# マルチテナント・アーキテクチャ関連のパラメータ (2)

## NONCDB\_COMPATIBLE

- PDB から実行する操作を Non-CDB からと同様の挙動にするための設定値

値の範囲	説明
TRUE	PDB で実行されたコマンドは Non-CDB で実行した場合と同様に動作する
FALSE (デフォルト)	PDB で実行されたコマンドは PDB で実行した場合を想定した動作をする

- シングルテナント (CDB 内に PDB が 1つのみ存在する) 構成を想定
- 通常、ルートでのみ実行可能なコマンドを PDB から実行できるようにする

# プラグブル・データベースの関連情報

## アラート・ログ

- アラート・ログ (alert.log) は、インスタンスごとに存在するファイル
- マルチテナント・アーキテクチャにおいても、単一のファイルを共有して使用する
  - PDB に関する操作やエラーの情報は、単一のアラート・ログに出力される
  - PDB オープン時におけるアラート・ログからの出力例

```
[oracle@node01]$ tail alert_orcl.log
```

```
...
```

```
alter pluggable database erp open
```

```
Wed Jul 11 17:33:12 2013
```

```
Pluggable database ERP opened read write
```

```
Completed: alter pluggable database erp open
```



PDB (ERP)

起動時の出力例

# プラグابل・データベースの関連情報

## Automatic Workload Repository (AWR)

- 自動パフォーマンス統計の利用が可能
- AWR で取得したスナップショットはすべてのコンテナで共有する
- コンテナごとに AWR レポートの生成が可能
  - 一部の情報について、レポートを生成したコンテナの情報のみ表示する

### ルートで生成した場合

- すべてのコンテナ（ルートおよびすべての PDB）の情報を表示する

### 特定の PDB で生成した場合

- そのコンテナ（特定の PDB）の情報のみを表示する

# Automatic Workload Repository (AWR) レポート

## ルートで作成した AWR レポートの出力例

- AWR レポートからの抜粋

[Back to Top](#)

### SQL ordered by Elapsed Time

- Resources reported for PL/SQL code includes the resources used by all SQL statements called by the code.
- % Total DB Time is the Elapsed Time of the SQL statement divided into the Total Database Time multiplied by 100
- %Total - Elapsed Time as a percentage of Total DB time
- %CPU - CPU Time as a percentage of Elapsed Time
- %IO - User I/O Time as a percentage of Elapsed Time
- Captured SQL account for 332.6% of Total DB Time (s): 373
- Captured PL/SQL account for 34.7% of Total DB Time (s): 373

すべてのコンテナ  
の情報を表示する

PDB Name	
ERP	II
CRM	II
HR	II
3) CRM	E
HR	d
CRM	d
ERP	d

Elapsed Time (s)	Executions	Elapsed Time per Exec (s)	%Total	%CPU	%IO	SQL Id	SQL Module	PDB Name	SQL Text
102.59	997,309	0.00	27.48	42.49	0.45	<a href="#">5wru8ckp9r378</a>	SQL*Plus	ERP	INSERT INTO TEST VALUES (:B1 ,...
101.94	998,737	0.00	27.31	42.83	0.26	<a href="#">5wru8ckp9r378</a>	SQL*Plus	CRM	INSERT INTO TEST VALUES (:B1 ,...
100.70	995,102	0.00	26.98	43.31	0.49	<a href="#">5wru8ckp9r378</a>	SQL*Plus	HR	INSERT INTO TEST VALUES (:B1 ,...
11.28	2	5.64	3.02	14.27	4.84	<a href="#">f2fn1pip86yh3</a>	sqlplus@jpintl005.jp.oracle.com (TNS V1-V3)	CRM	BEGIN DBMS_WORKLOAD_REPOSITORY...
9.85	0		2.64	75.64	0.00	<a href="#">4nq6k4fh94quc</a>	SQL*Plus	HR	declare i integer; begin for i...
9.63	0		2.58	60.06	0.00	<a href="#">4nq6k4fh94quc</a>	SQL*Plus	CRM	declare i integer; begin for i...
8.39	0		2.25	55.37	0.00	<a href="#">4nq6k4fh94quc</a>	SQL*Plus	ERP	declare i integer; begin for i...
6.80	2	3.40	1.82	1.57	0.00	<a href="#">6ajkhukk78nsr</a>			begin prvt_hdm.auto_execute( :...
4.96	1	4.96	1.33	0.00	0.00	<a href="#">18u5bnb9bvzvh</a>			SELECT STATUS# FROM SYS.DBA_AD...
3.78	10	0.38	1.01	51.93	22.04	<a href="#">fhf8upax5cxsz</a>	MMON_SLAVE		BEGIN sys.dbms_auto_report_int...

# Automatic Workload Repository (AWR) レポート

## 特定の PDB で作成した AWR レポートの出力例

- AWR レポートからの抜粋

[Back to Top](#)

### SQL ordered by Elapsed Time

- Resources reported for PL/SQL code includes the resources used by all SQL statements called by the code.
- % Total DB Time is the Elapsed Time of the SQL statement divided into the Total Database Time multiplied by 100
- %Total - Elapsed Time as a percentage of Total DB time
- %CPU - CPU Time as a percentage of Elapsed Time
- %IO - User I/O Time as a percentage of Elapsed Time
- Captured SQL account for 27.4% of Total DB Time (s): 373
- Captured PL/SQL account for 5.6% of Total DB Time (s): 373

CRM の情報のみ  
を表示する

PDB Name
CRM

Elapsed Time (s)	Executions	Elapsed Time per Exec (s)	%Total	%CPU	%IO	SQL Id	SQL Module	PDB Name	SQL Text
101.94	998,737	0.00	27.31	42.83	0.26	<a href="#">5wru8ckp9r378</a>	SQL*Plus	CRM	INSERT INTO TEST VALUES (:B1 ,...
11.28	2	5.64	3.02	14.27	4.84	<a href="#">f2fn1pjp86yh3</a>	sqlplus@jint005.jp.oracle.com (TNS V1-V3)	CRM	BEGIN DBMS_WORKLOAD_REPOSITORY...
9.63	0	2.58	60.06	0.00	4nq6k4fh94quc	SQL*Plus	CRM	declare i integer; begin for i...	
0.08	1	0.08	0.02	99.59	0.47	<a href="#">54xn0qpxdqagv</a>	SQL*Plus	CRM	declare TYPE attr_s_cur IS REF ...
0.08	1	0.08	0.02	99.80	0.48	<a href="#">35axt7fjys7f</a>	SQL*Plus	CRM	select sdo_geor_def.listAllGeo...
0.08	1	0.08	0.02	99.34	0.49	<a href="#">96ub5w77tu2ac</a>	SQL*Plus	CRM	SELECT column_name, data_type...
0.01	50	0.00	0.00	7.39	0.00	<a href="#">2au6dvzuwx2n1</a>		CRM	SELECT TS# FROM SYS."TSS" "TSS..."
0.01	30	0.00	0.00	37.77	0.00	<a href="#">5sq7mirpi21z7</a>		CRM	SELECT JOB, LAST_DATE, THIS_DA...
0.01	6	0.00	0.00	75.62	9.03	<a href="#">at1yqf4rd7cvi</a>		CRM	select file#, block#, blocks f...
0.00	6	0.00	0.00	92.45	0.46	<a href="#">0hhmdwwqxwbw0r</a>		CRM	select obj#, type#, flags, rel...

# 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(pdb1) のみを対象とする場合

```
RMAN> BACKUP PLUGGABLE DATABASE pdb1;
```

# 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(pdb1) のみを対象とする場合

```
RMAN> RECOVERY PLUGGABLE DATABASE pdb1;
```

# その他のバックアップ

## OS コマンドを使用したオンラインバックアップ

- PDB 単位のオンラインバックアップ（非一貫性バックアップ）も可能
- PDB に対するオンラインバックアップ実行時のポイント
  - CDB がアーカイブ・ログ・モードで運用されていることを確認
  - バックアップ対象となる PDB の表領域に関連するデータファイルを確認

```
[oracle@node01]$ sqlplus system/Manager1@scan.oracle12c.jp/pdb1  
SQL> SELECT FILE_NAME FROM DBA_DATA_FILES;
```

- 開始と終了のコマンドには ALTER PLUGGABLE DATABASE 文を使用

```
SQL> ALTER PLUGGABLE DATABASE pdb1 BEGIN BACKUP;
```

```
SQL> ALTER PLUGGABLE DATABASE pdb1 END BACKUP;
```

# 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)

# まとめ

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



Plug into the **Cloud.**

# 次世代データベースによる統合

## Multitenant Container Database



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

# マルチテナント・アーキテクチャによるデータベース統合

運用コストを圧縮してシステムの独立性を保つ、プラグブル・データベースでの統合

データベース統合



マルチテナント  
アーキテクチャ

- ✓ ハードウェア・コストの削減
- ✓ データベース運用コストの削減
- ✓ 環境構築にかかる新規開発コストの削減
- ✓ アプリケーションの変更は不要
- ✓ Unplug & Plug による運用
- ✓ アプリ開発とデータベース運用を分離

# Appendix

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



Plug into the **Cloud.**

# Oracle Database 12c New Features

マルチテナント型のデータベース統合、クラウドに必要な可用性、運用の自動化を提供

Cloud &  
Consolidation

## Multitenant Architecture

DB内に仮想DBを複数持つ構成

## Consolidated Testing

複数環境の合計負荷テスト

Information  
Lifecycle  
Management

## Heat Map

データ利用頻度を目に見える形で  
レポートする

## Automatic Data Optimization

ポリシーベースでデータ運用を自動  
化する

## In-Database Archiving

DB内でデータをアーカイブと同等に  
見せる

Security

## Data Redaction

DBユーザ毎に動的マスキング

## Privilege Analysis

特権ユーザの利用状況を分析レポ  
ートとして発行する

Cluster

## Flex ASM

ASMインスタンスとDBインスタンスを  
分離して配置

## Flex Cluster

オラクルの新しいクラスタウェア、  
Hub&Spoke型のクラスタを構成

High  
Availability

## Far Sync (Data Guard)

最小負荷でデータロスのない災対  
サイトを構築

## Global Data Services

複製サイトをまとめてサービス化し、  
障害時の切替に負荷分散を適用

## Application Continuity

DBがダウンした際、成功するまでトラ  
ンザクションをリプレイする

# 参考資料: 主要新機能のライセンス

機能名	必要ライセンス	備考
マルチテナント・アーキテクチャ	Multitenant	マルチテナント構成の場合のみ。シングルテナント構成は各Editionの機能として利用可能。
Heat Map	Advanced Compression	
Automatic Data Optimization (自動データ最適化)	Advanced Compression	
Data Redaction	Advanced Security	
Privilege Analysis	Database Vault	
Unified Auditing	EE/SE/SE1	
Far Sync	Active Data Guard	Far Syncインスタンスは本番環境とは別筐体にインストールする必要がありますが、そのために別途ライセンスを購入する必要はありません
Global Data Service	Active Data Guard	
Application Continuity	Active Data Guard Real Application Clusters	構成に応じていずれかが必要
Consolidated Testing	Real Application Testing	

- Oracle Multitenant
- Oracle Database 12cでは下記3種類のデータベース構成が可能であり、それぞれに必要なライセンスが異なります
- マルチテナント構成
  - ✓ 1つのCDBに複数のPDBを持つ構成（252個まで）
  - ✓ Oracle Database Enterprise Edition および Oracle Multitenant Option が必要になります
- シングルテナント構成
  - ✓ 1つのCDBに1つだけPDBを持つ構成
  - ✓ データベース標準機能であるため、全ての Edition で構成が可能
- Non-CDB構成
  - ✓ 11gまでの従来型のデータベース構成、全てのEdition で構成が可能



# Integrated Cloud

## Applications & Platform Services

ORACLE®