

ORACLE®

ORACLE®

# Oracle Database 12c Release 1

## マルチテナント・アーキテクチャ (2)

日本オラクル株式会社

ORACLE®  
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

- プラガブル・データベースの管理 (1)
- プラガブル・データベースの管理 (2)
- まとめ

# プラグブル・データベースの 管理 (1)



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

SYS

SYSTEM

PDB 2  
PDB 1

PDB n

SYS

USER01

SYSTEM

HRADM

SYS

USER01

SYSTEM

ERPADM

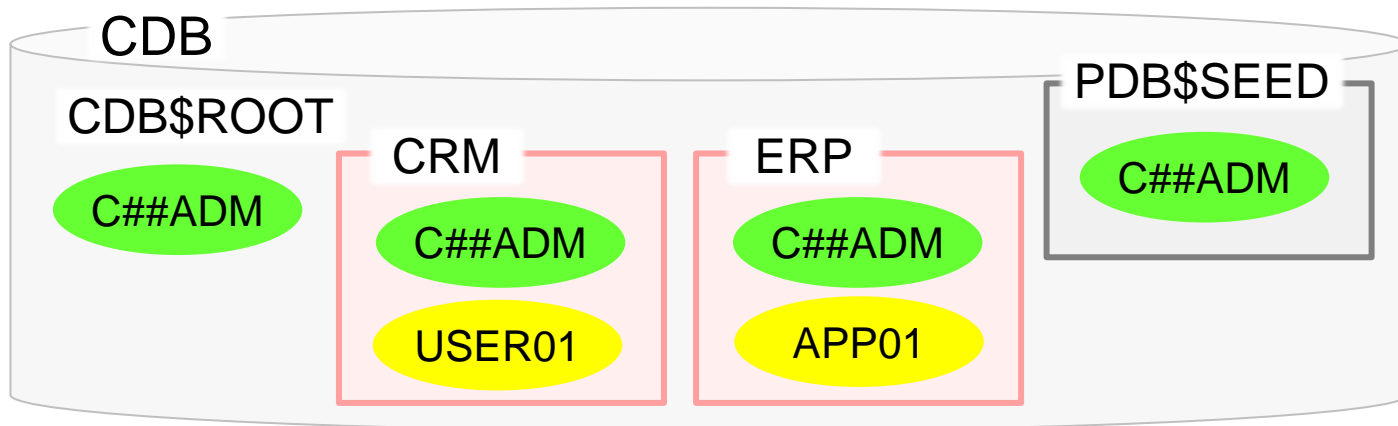
他の PDB とユーザー名は重複してもよい



# 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

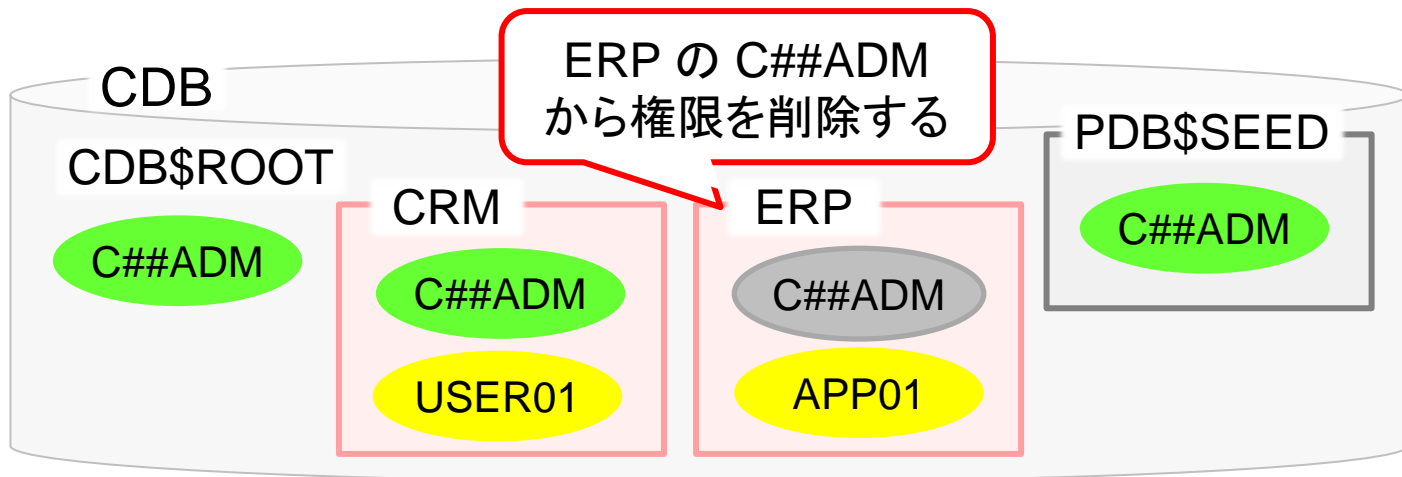




# 共通ユーザーの管理

## テナントごとにロールや権限を個別に設定する場合

- PDB ごとのユーザーであれば、ローカル・ユーザーで対応する
- 既存の共通ユーザーに対して、PDB への操作を制御したい場合にはテナントごとにロールや権限の設定をすることが可能



# 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
- 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;
```

# 初期化パラメータ

## マルチテナント・アーキテクチャに関連する初期化パラメータ

- Oracle Database 12c で追加された初期化パラメータの一覧
  - 赤字部分はマルチテナント・アーキテクチャに関連する初期化パラメータ

CONNECTION\_BROKERS

DNFS\_BATCH\_SIZE

ENABLE\_PLUGGABLE\_DATABASE

HEAT\_MAP

MAX\_STRING\_SIZE

PARALLEL\_DEGREE\_LEVEL

PDB\_FILE\_NAME\_CONVERT

OPTIMIZER\_ADAPTIVE\_FEATURES

OPTIMIZER\_ADAPTIVE\_REPORTING\_ONLY

PGA\_AGGREGATE\_LIMIT

PROCESSOR\_GROUP\_NAME

NONCDB\_COMPATIBLE

SPATIAL\_VECTOR\_ACCELERATION

TEMP\_UNDO\_ENABLED

THREADED\_EXECUTION

UNIFIED\_AUDIT\_SGA\_QUEUE\_SIZE

USE\_DEDICATED\_BROKER

# 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

# CREATE DATABASE 文による CDB の作成

## コマンドによる手動作成時の例

- CREATE DATABASE 文の実行時に CDB としての作成に必要な句を含める
  - OMF を使用する場合のコマンド例

```
CREATE DATABASE cdb1
USER SYS IDENTIFIED BY Welcome1
USER SYSTEM IDENTIFIED BY Welcome1
EXTENT MANAGEMENT LOCAL
DEFAULT TABLESPACE users
DEFAULT TEMPORARY TABLESPACE temp
UNDO TABLESPACE undotbs1
ENABLE PLUGGABLE DATABASE
SEED
SYSTEM DATAFILES SIZE 125M AUTOEXTEND ON NEXT 10M MAXSIZE UNLIMITED
SYSAUX DATAFILES SIZE 100M;
```

# マルチテナント・アーキテクチャ関連のパラメータ (2)

## PDB\_FILE\_NAME\_CONVERT

- PDB のデータファイル配置場所を示す設定値
- 既存のデータファイル配置場所と、新規に作成するデータファイル配置場所をマッピングする
- デフォルト値はなし
- 構文

```
PDB_FILE_NAME_CONVERT = 'string1', 'string2', 'string3', 'string4', ...
```

### – 例

```
PDB_FILE_NAME_CONVERT = 'dbs/t1/', 'dbs/t1/s_', 'dbs/t2/', 'dbs/t2/s_'
```



# マルチテナント・アーキテクチャ関連のパラメータ (3)

## 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

# プラガブル・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

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="#">18u5bnb9bvzha</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...

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

# 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
/3) CRM
CRM
CRM
CRM
CRM
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@pintl005.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="#">54xn0gpxdqayv</a>	SQL*Plus	CRM	declare TYPE attr_s_cur IS REF ...
0.08	1	0.08	0.02	99.80	0.48	<a href="#">35axt7fjjs7f</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="#">at1yqf4rd7cvj</a>		CRM	select file#, block#, blocks f...
0.00	6	0.00	0.00	92.45	0.46	<a href="#">0hhmdwvwxqbw0r</a>		CRM	select obj#, type#, flags, rel...

# プラグブル・データベースの 管理 (2)

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



Plug into the **Cloud.**

ORACLE<sup>®</sup>

# 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
```

# 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

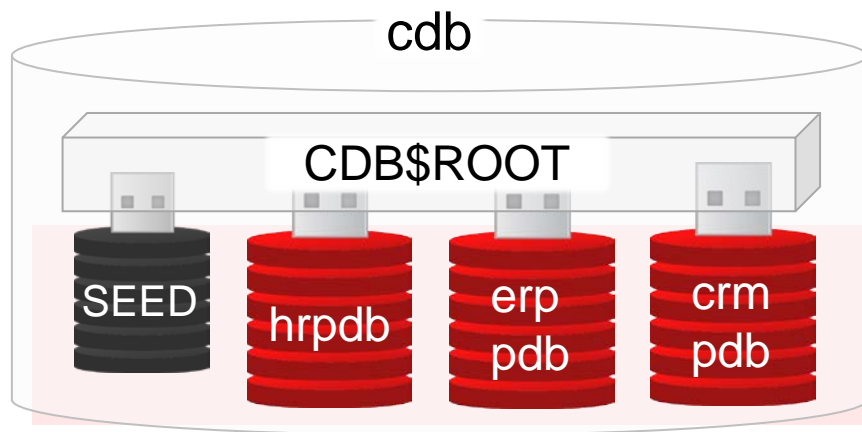
## Oracle Database Resource Manager

- 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）

# 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

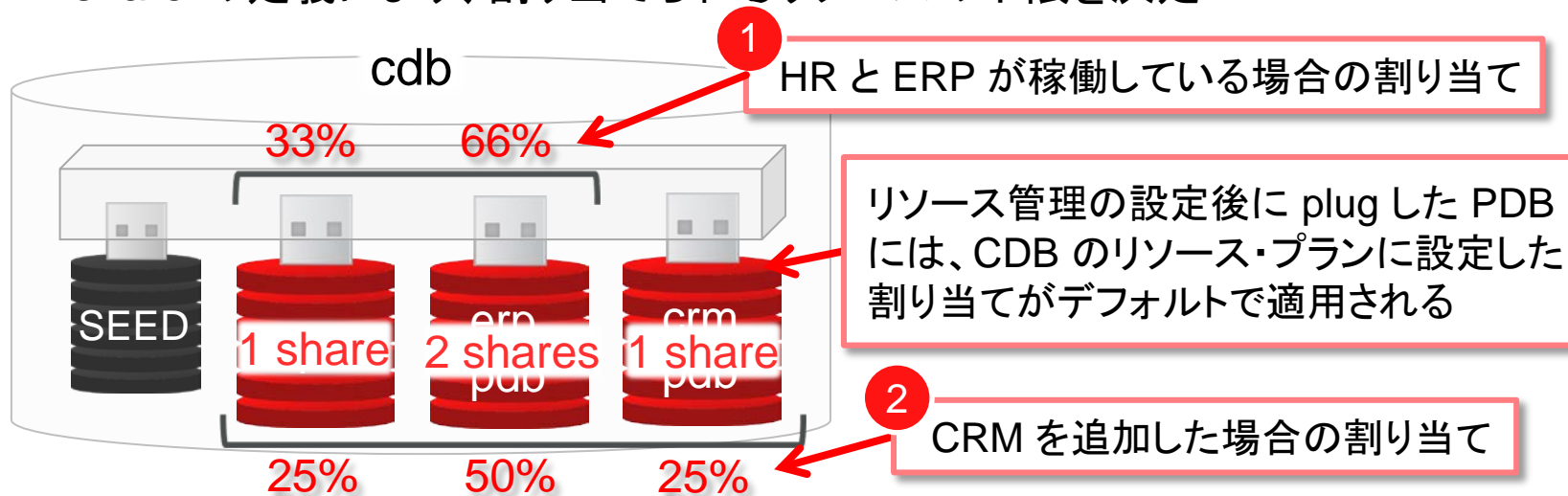


PDB レベルでリソースを管理  
PDB の増減時も効率的に管理

# 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



# Resource Manager によるリソース管理

## CDB レベルに加えて PDB レベルでもリソース制御を実現

- 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 レベル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 レベルでの設定において、CDB レベル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

# Resource Manager によるリソース管理

## CDB レベルでのリソース管理の例

- CDB 内の各 PDB のリソースを制御する
- リソース・プラン (NEWCDB\_PLAN) における PDB (erp) のリソース設定例

```
SQL> exec DBMS_RESOURCE_MANAGER.CREATE_PENDING_AREA();
```

PL/SQLプロシージャが正常に完了しました。

```
SQL> BEGIN
```

```
2     DBMS_RESOURCE_MANAGER.CREATE_CDB_PLAN(
```

```
3     plan      => 'newcdb_plan',
```

```
4     comment => 'New CDB resource plan for weekend');
```

```
5 END;
```

```
6 /
```

PL/SQLプロシージャが正常に完了しました。

<次ページへ続く>



# Resource Manager によるリソース管理

## CDB レベルでのリソース管理の例

```
SQL> BEGIN
  2   DBMS_RESOURCE_MANAGER.CREATE_CDB_PLAN_DIRECTIVE(
  3     plan                => 'newcdb_plan',
  4     pluggable_database  => 'erp',
  5     shares              => 3,
  6     utilization_limit   => 100,
  7     parallel_server_limit => 100);
  8 END;
  9 /
```

PDB (ERP) に対する  
リソース制御の設定

PL/SQL プロシージャが正常に完了しました。

```
SQL> exec DBMS_RESOURCE_MANAGER.VALIDATE_PENDING_AREA();
```

PL/SQL プロシージャが正常に完了しました。

```
SQL> exec DBMS_RESOURCE_MANAGER.SUBMIT_PENDING_AREA();
```

PL/SQL プロシージャが正常に完了しました。

# Resource Manager によるリソース管理

## 設定内容の確認とポイント

```
SQL> SELECT PLAN, PLUGGABLE_DATABASE, SHARES, UTILIZATION_LIMIT,  
PARALLEL_SERVER_LIMIT FROM DBA_CDB_RSRC_PLAN_DIRECTIVES ORDER BY PLAN WHERE PLAN;
```

Plan	Pluggable Database	Shares	Utilization Limit	Parallel Server Limit
DEFAULT_CDB_PLAN	ORA\$AUTOTASK			
DEFAULT_CDB_PLAN	ORA\$DEFAULT_PDB_DIRECTIVE			
DEFAULT_MAINTENANCE_PLAN	ORA\$DEFAULT_PDB_DIRECTIVE			
DEFAULT_MAINTENANCE_PLAN	ORA\$AUTOTASK		90	100
NEWCDB_PLAN	ORA\$DEFAULT_PDB_DIRECTIVE	1	100	100
NEWCDB_PLAN	ERP	3	100	100
NEWCDB_PLAN	ORA\$AUTOTASK		90	100
ORA\$INTERNAL_CDB_PLAN	ORA\$DEFAULT_PDB_DIRECTIVE			
ORA\$INTERNAL_CDB_PLAN	ORA\$AUTOTASK			
ORA\$QOS_CDB_PLAN	ORA\$DEFAULT_PDB_DIRECTIVE	1	100	100
ORA\$QOS_CDB_PLAN	ORA\$AUTOTASK		90	100

11行が選択されました。

1 新規の PDB に対するデフォルトの設定を必要に応じて変更しておく

2 個別に設定が必要な PDB は設定を行う

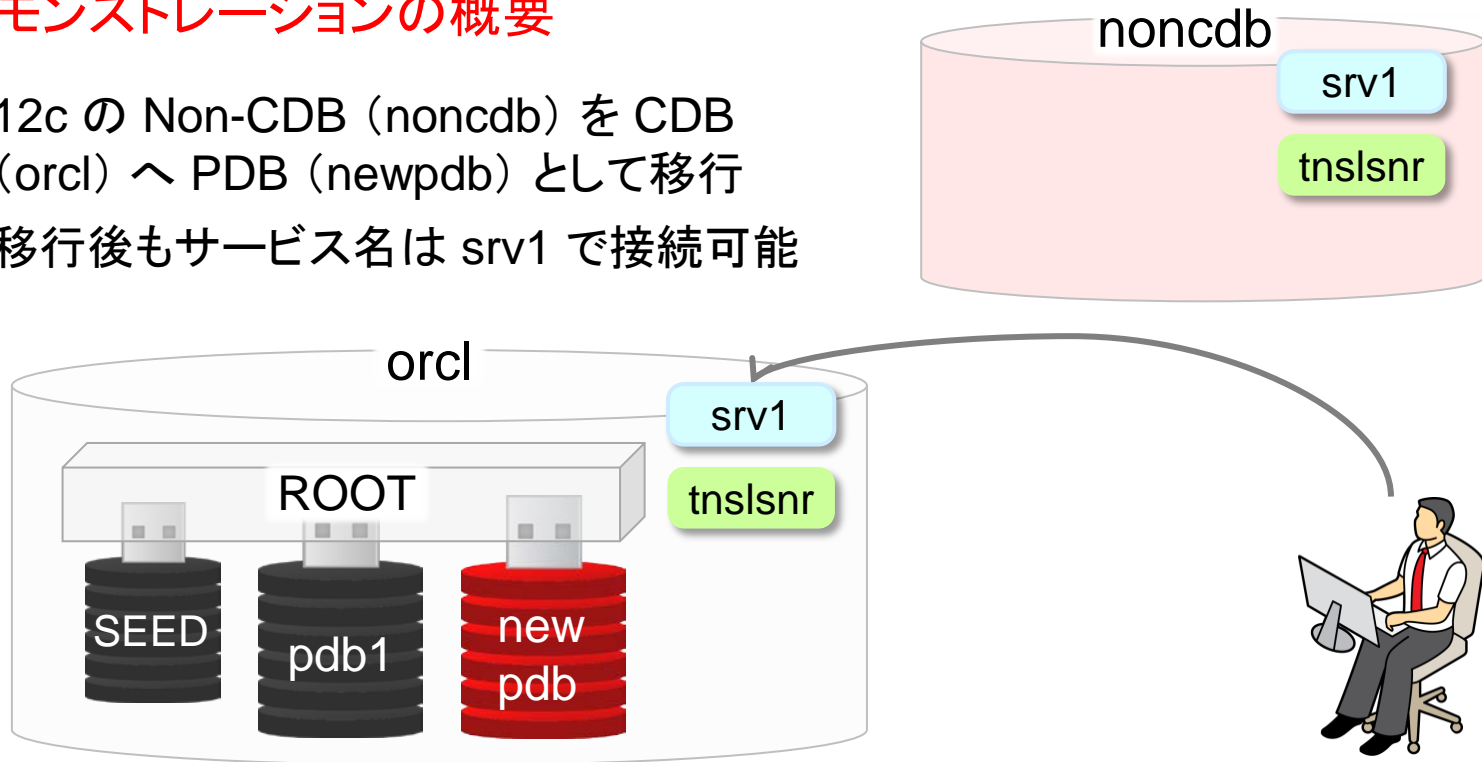


# ***DEMONSTRATION***

# Non-CDB から PDB への移行

## デモンストレーションの概要

- 12c の Non-CDB (noncdb) を CDB (orcl) へ PDB (newpdb) として移行
- 移行後もサービス名は srv1 で接続可能



# デモンストレーション

## 1. 事前準備（共通ユーザーの作成）

- orcl へ接続して、共通ユーザーを作成

```
[oracle@node01]$ export ORACLE_SID=orcl  
sqlplus /nolog @c_user
```

### – スクリプト中で実行しているコマンド

```
SQL> SHOW PDBS  
  
SQL> ALTER PLUGGABLE DATABASE pdb1 OPEN;  
  
SQL> CREATE USER c##user01 IDENTIFIED BY Welcome1;  
  
SQL> ALTER PLUGGABLE DATABASE pdb1 CLOSE IMMEDIATE;  
  
SQL> SELECT username, con_id FROM CDB_USERS WHERE username = 'C##USER01';
```

既存の PDB には、共通ユーザー（C##USER01）が作成されている状態にしておく

# デモンストレーション

## 2. 事前準備（サービスの作成）

- noncdb へ接続して、サービス (srv1) を作成

```
[oracle@node01]$ export ORACLE_SID=noncdb  
sqlplus /nolog @n_srv
```

### – スクリプト中で実行しているコマンド

```
SQL> SELECT NAME, NETWORK_NAME FROM V$SERVICES;  
SQL> exec DBMS_SERVICE.CREATE_SERVICE('srv1','srv1');  
SQL> exec DBMS_SERVICE.START_SERVICE('srv1');  
SQL> SELECT NAME, NETWORK_NAME FROM V$SERVICES;
```

# デモンストレーション

## 3. 事前準備（サンプル・データの作成）

- ユーザーを作成して、サンプル・データを用意する

```
sqlplus /nolog @n_tab
```

– スクリプト中で実行しているコマンド

```
SQL> CREATE USER hr IDENTIFIED BY hr;
```

```
SQL> GRANT DBA TO hr;
```

```
SQL> conn hr/hr@localhost:1521/srv1
```

< 表 HR の作成と、サンプル・データ 5行の挿入 >

```
SQL> SHUTDOWN IMMEDIATE
```

# デモンストレーション

## 4. noncdb の unplug (1)

- noncdb を unplug (切断) する

```
sqlplus /nolog @n_xml
```

– スクリプト中で実行しているコマンド

```
SQL> STARTUP MOUNT
```

```
SQL> ALTER DATABASE OPEN READ ONLY;
```

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

noncdb の互換性が 12.1.0.0.0 以上であることを確認



# デモンストレーション

## 4. noncdb の unplug (2)

- noncdb を unplug (切断)する作業の続き

```
SQL> BEGIN
  2   DBMS_PDB.DESCRIBE(
  3     pdb_descr_file => '/home/oracle/noncdb.xml');
  4   END;
  5   /
SQL> SHUTDOWN IMMEDIATE
```

noncdb の構成 (データファイルの情報など) に関する XML メタデータ・ファイルを作成

# デモンストレーション

## 5. PDB としての plug

- orcl へ接続して、PDB を作成

```
[oracle@node01]$ export ORACLE_SID=orcl  
sqlplus /nolog @c_plug
```

### – スクリプト中で実行しているコマンド

```
SQL> SELECT PDB_NAME, STATUS FROM CDB_PDBS;  
  
SQL> CREATE PLUGGABLE DATABASE newpdb USING '/home/oracle/noncdb.xml' COPY;  
  
SQL> SELECT PDB_NAME, STATUS FROM CDB_PDBS;
```

unplug 時に作成した XML メタデータ・ファイルを使用して PDB を作成する

# デモンストレーション

## 6. plug 後の作業

- plug 後に、データディクショナリの再作成やルートとの同期を実施する

```
sqlplus /nolog @c_script
```

### – スクリプト中で実行しているコマンド

```
SQL> ALTER SESSION SET CONTAINER = newpdb;  
SQL> @?/rdbms/admin/noncdb_to_pdb.sql  
SQL> ALTER PLUGGABLE DATABASE newpdb OPEN READ WRITE RESTRICTED;  
SQL> ALTER SESSION SET CONTAINER = newpdb;  
SQL> exec DBMS_PDB.SYNC_PDB();  
SQL> SELECT username, con_id FROM CDB_USERS WHERE username = 'C##USER01';  
SQL> SHUTDOWN IMMEDIATE  
SQL> STARTUP
```

# デモンストレーション

## 7. サービスの再作成

- PDB (newpdb) のサービス (srv1) を作成

```
[oracle@node01]$ export ORACLE_SID=orcl  
sqlplus /nolog @c_srv
```

– スクリプト中で実行しているコマンド

```
SQL> exec DBMS_SERVICE.CREATE_SERVICE('srv1','srv1');  
SQL> exec DBMS_SERVICE.START_SERVICE('srv1');  
SQL> SELECT NAME, NETWORK_NAME, PDB, CON_ID FROM V$SERVICES;
```

PDB (newpdb) のサービスとして srv1 と newpdb が表示されることを確認

# デモンストレーション

## 8. 表 HR への SELECT を実行

- ユーザー HR で、表 HR へアクセスできることを確認

```
[oracle@node01]$ export ORACLE_SID=orcl  
sqlplus /nolog @c_check
```

– スクリプト中で実行しているコマンド

```
SQL> conn hr/hr@localhost:1521/srv1  
SQL> SELECT * FROM hr;
```



unplug / plug 前後で、サービス名やスキーマ・オブジェクト名に変更がないことを確認

# まとめ

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



Plug into the **Cloud.**

ORACLE<sup>®</sup>

# 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

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

ORACLE

# マルチテナント・アーキテクチャ

## データベースの構成パターン

- 12c におけるデータベースは、次のいずれかの構成となる

### Non-CDB 構成

- 従来型のデータベース構成

### シングルテナント構成

- CDB 内に PDB を 1つだけ有する構成

### マルチテナント構成

- CDB 内に PDB を 1つ以上有する構成

## Non-CDB 構成

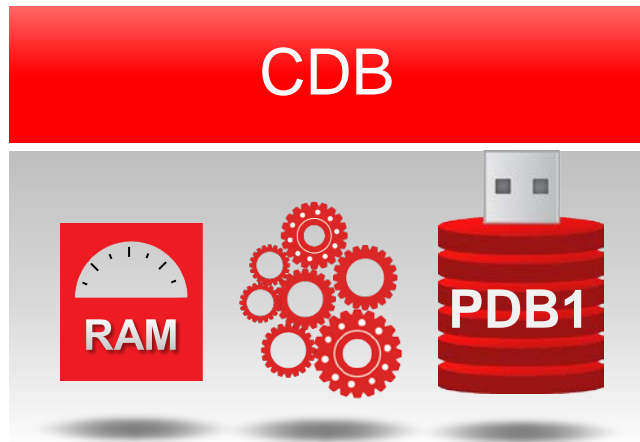




# マルチテナント・アーキテクチャ

## データベースの構成パターン

シングルテナント構成



マルチテナント構成



➤ **マルチテナント・アーキテクチャを最大限に活用できる構成**

## ライセンス

- Enterprise Edition (EE) で利用可能なオプション機能として提供
  - EE 以外のエディションでは CDB を作成および利用できるが、PDB の作成数に制限がある
- 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

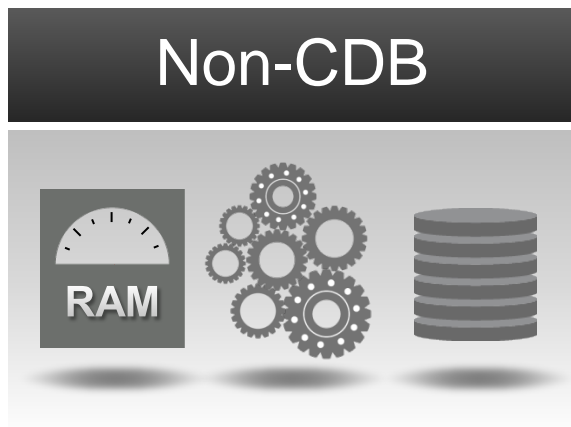
Edition	対応状況	PDB 作成数の上限
Standard Edition One	△ 対応 (制限あり)	1
Standard Edition	△ 対応 (制限あり)	1
Enterprise Edition	○ 対応	252

# エディションごとに利用可能な構成パターン

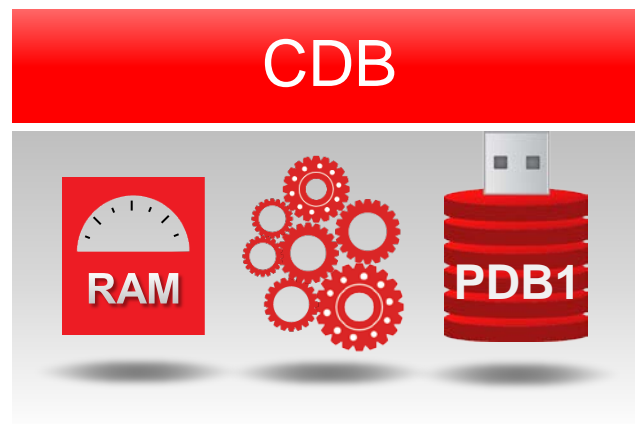
## Standard Edition / Standard Edition One

- Non-CDB 構成、あるいはシングルテナント構成が利用可能
- オプションは不要

### Non-CDB 構成



### シングルテナント構成

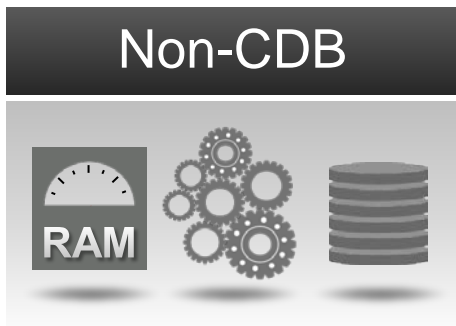


# エディションごとに利用可能な構成パターン

## Enterprise Edition

- Non-CDB 構成、シングルテナント構成、マルチテナント構成が利用可能
- PDB を複数作成する場合（マルチテナント構成）のみオプションが必要

### Non-CDB 構成



### シングルテナント構成



### マルチテナント構成



Oracle Multitenant が必要

# マルチテナント・アーキテクチャ

## 機能レベルでの主な制限事項

- 次の機能と組み合わせて使用することはできない
  - Heat map
  - Automatic Data Optimization
  
- 次の機能の使用には一部制限がある（PDB では使用することができない）
  - フラッシュバック・テクノロジー

**Hardware and Software**

**ORACLE®**

**Engineered to Work Together**

ORACLE®