

12c New feature - Multitenant

Techdata OTS Team

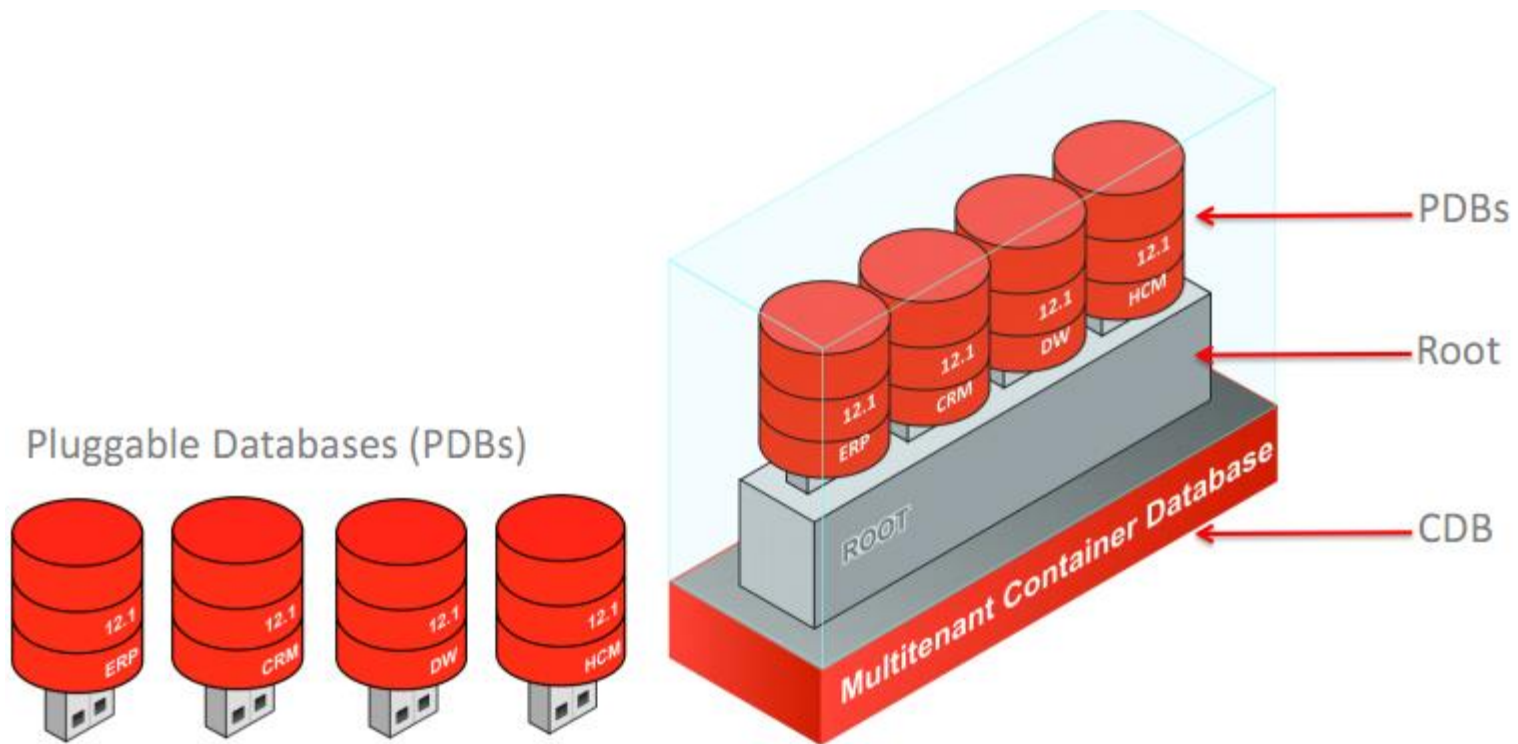
2015.12

Contents

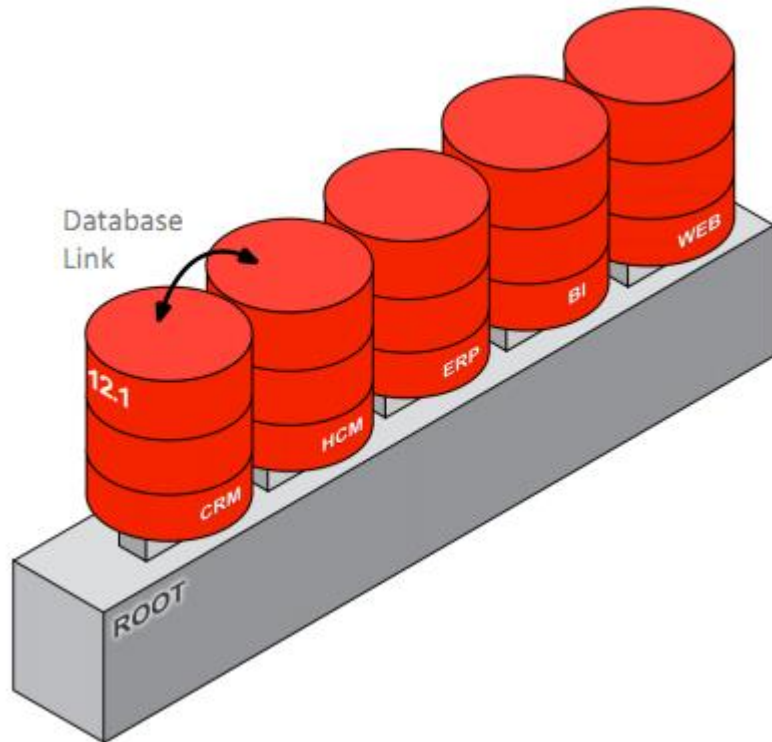
1. Architecture
2. Services and Connections
3. Data Dictionary Architecture
4. Managing CDB
5. Managing PDB

1. Multitenant Architecture

Multitenant Database 구성요소

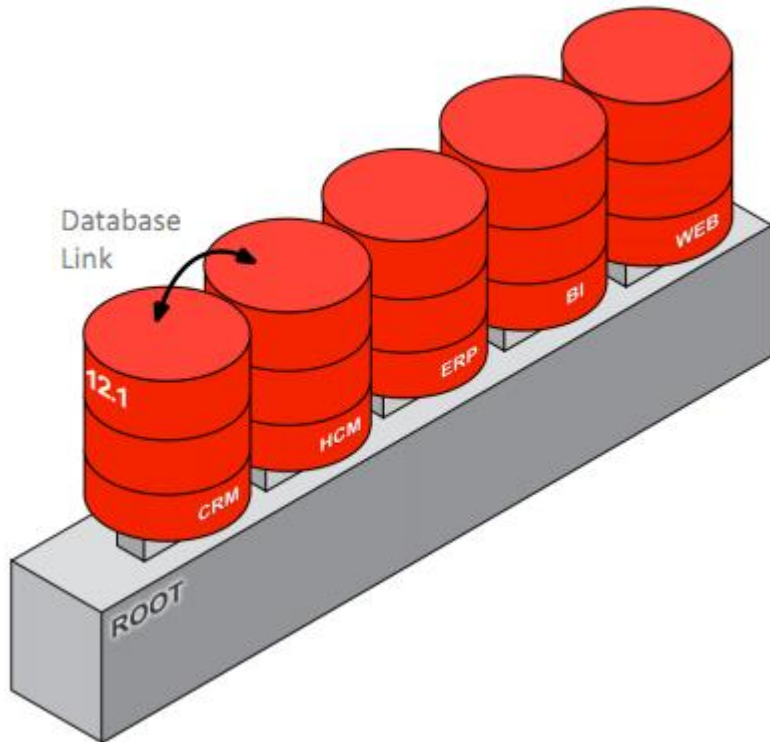


1. Multitenant Architecture



- **Multitenant** 아키텍처는 252개의 **PDB**를 지원
- **PDB**는 **non-CDB**와 동일하게 동작
- 접속하는 입장에서는 **PDB**와 **non-CDB** 구분이 되지 않음

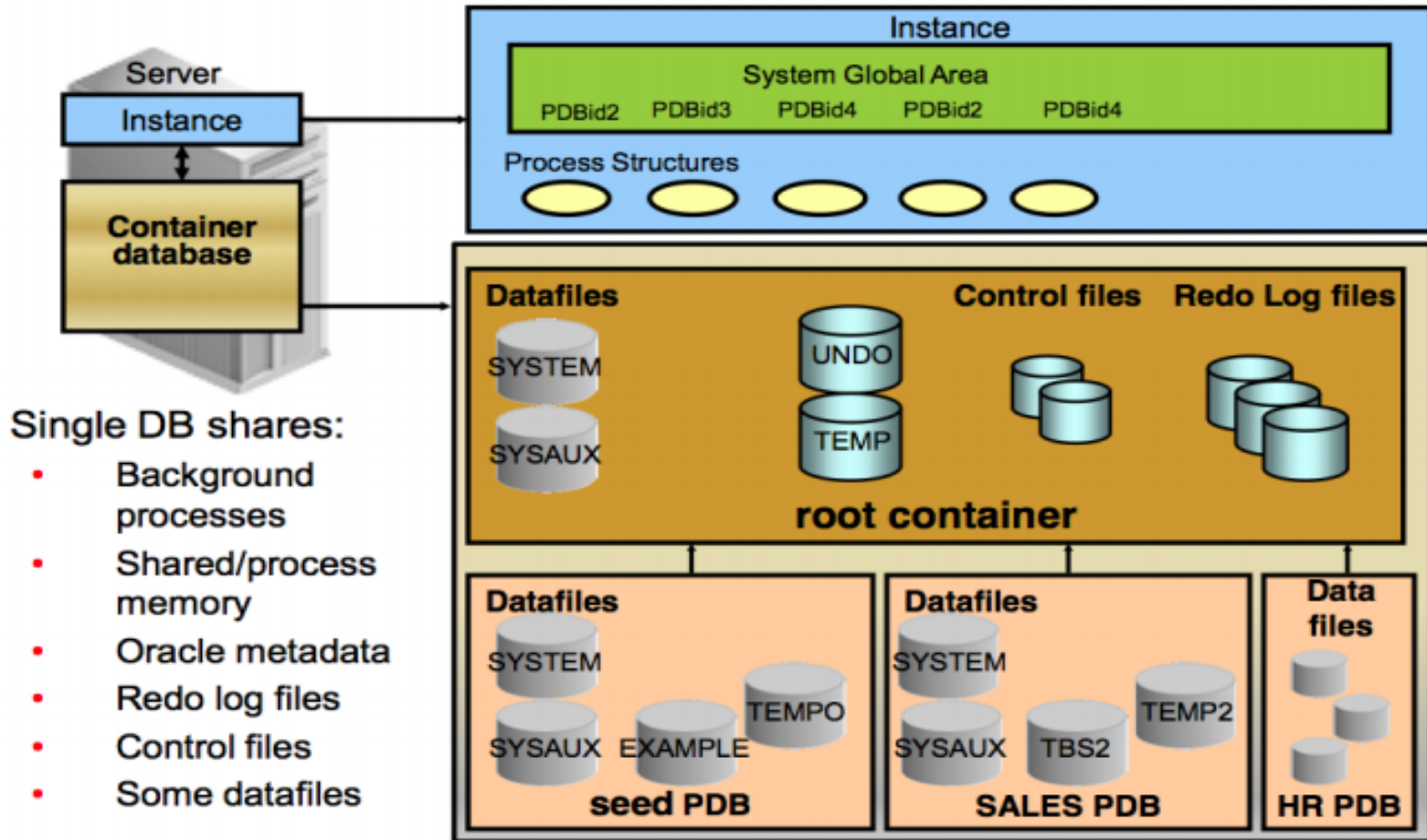
1. Multitenant Architecture



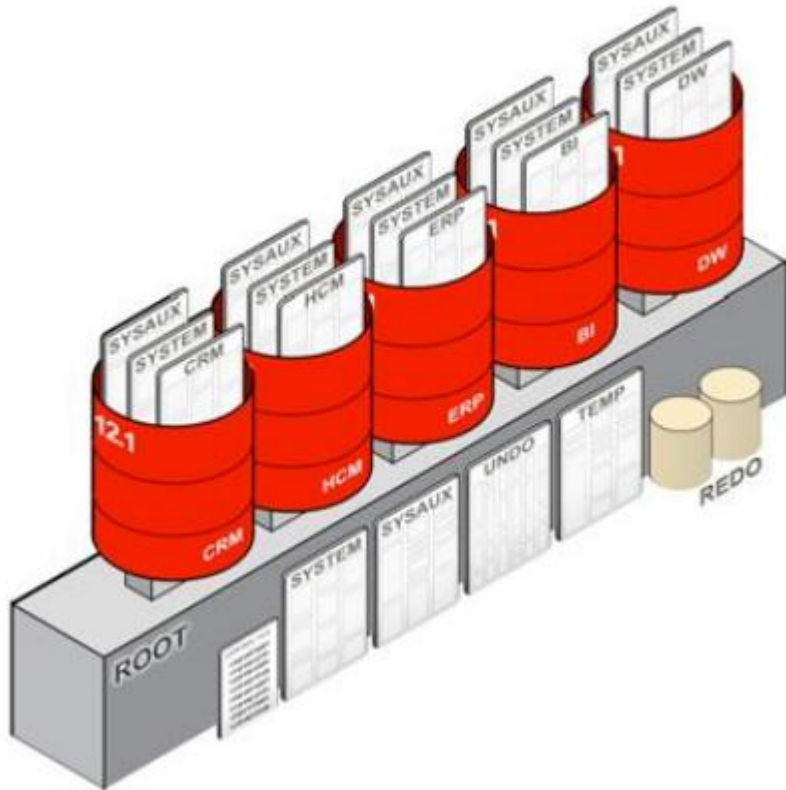
- **Container Database (CDB)**
 - **Root:** CDB\$ROOT라는 이름으로 오라클 메타데이터와 커먼 유저를 포함
 - **Seed (PDB\$SEED):** PDB\$SEED라는 이름으로 새 PDB를 만드는 템플릿
 - **PDBs:** 사용자와 어플리케이션에는 non-CDB와 동일하게 동작하는 데이터베이스
- 모든 **CDB**는 하나의 루트와 **1**개의 씨드, 그리고 **0**개 이상의 **PDB**가 존재

1. Multitenant Architecture

Multitenant Container Database (CDB) 구성요소

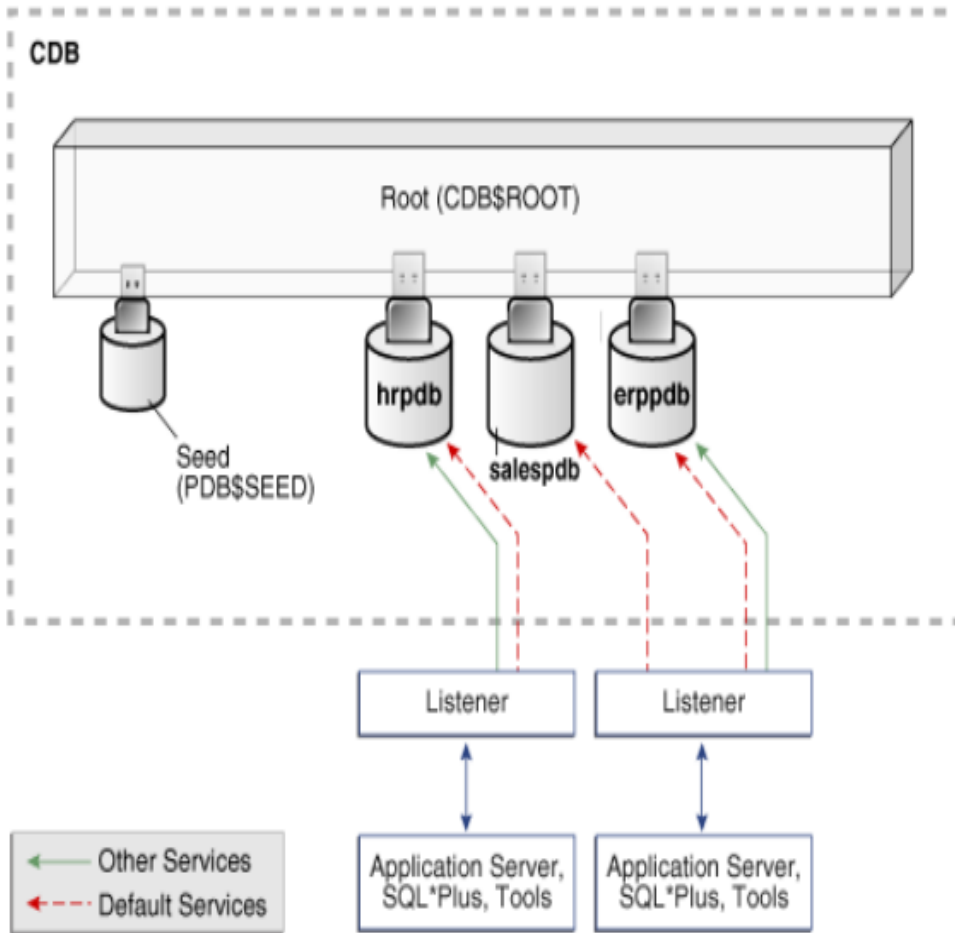


1. Multitenant Architecture



- 각 PDB는 SYSTEM 과 SYSAUX 등의 테이블 스페이스를 소유
- PDB들은 UNDO, REDO 그리고 control files 들을 공유
- CDB는 하나의 TEMP tablespace를 갖고 있지만 PDB는 각자 생성이 가능

2. Services in a CDB



- **PDB**에는 서비스를 통해서만 접속
- “현재” 컨테이너는 접속하여 있는 곳을 의미함
- **CDB**가 생성되면, 동일한 이름의 서비스는 자동적으로 생성됨
(자동적으로 리스너에 등록)
- **PDB**가 생성되면, 서비스가 생성되고 **CDB** 내에 존재하는 것으로 등록됨
(리스너에 서비스 수동 등록)

2. Connecting to CDBs and PDBs

- **CDB Root container (CDB\$ROOT) 접속은 다음과 같이 수행**
 - 서비스 명
@<host>:<port>/<service>
 - sqlplus SYS/oracle@localhost:1521/cdb1 AS SYSDBA
 - OS 인증
 - sqlplus / AS SYSDBA
- **PDB 접속은 다음과 같이 수행:**
 - 서비스 명 혹은 TNS 등록을 통해 수행

2. Connecting to CDBs and PDBs

- 두 개의 유용한 새 커맨드를 이용해 접속 상태를 확인하는 것이 가능:
- **SHOW CON_ID**: 컨테이너 ID를 보여줌
 - 1 은 Root (CDB\$ROOT)를 의미
 - 3 이상은 PDB를 의미
- **SHOW CON_NAME**: 컨테이너 이름을 보여줌
 - CDB\$ROOT (for Root)

2. Connecting to CDBs and PDBs

컨테이너 레벨 접속

```
[oracle@db12cvm1 ~]$ sqlplus SYS/oracle@db12cvm1:1521/orcl AS  
SYSDBA
```

```
SQL> SHOW CON_NAME;
```

```
CON_NAME
```

```
-----  
CDB$ROOT
```

```
SQL> SHOW CON_ID;
```

```
CON_ID
```

```
-----  
1
```

2. Connecting to CDBs and PDBs

PDB 레벨 접속

```
[oracle@db12cvm1 ~]$ sqlplus SYS/oracle@db12cvm1:1521/pdb1 AS  
SYSDBA
```

```
SQL> SHOW CON_NAME;
```

```
CON_NAME
```

```
-----
```

```
PDB1
```

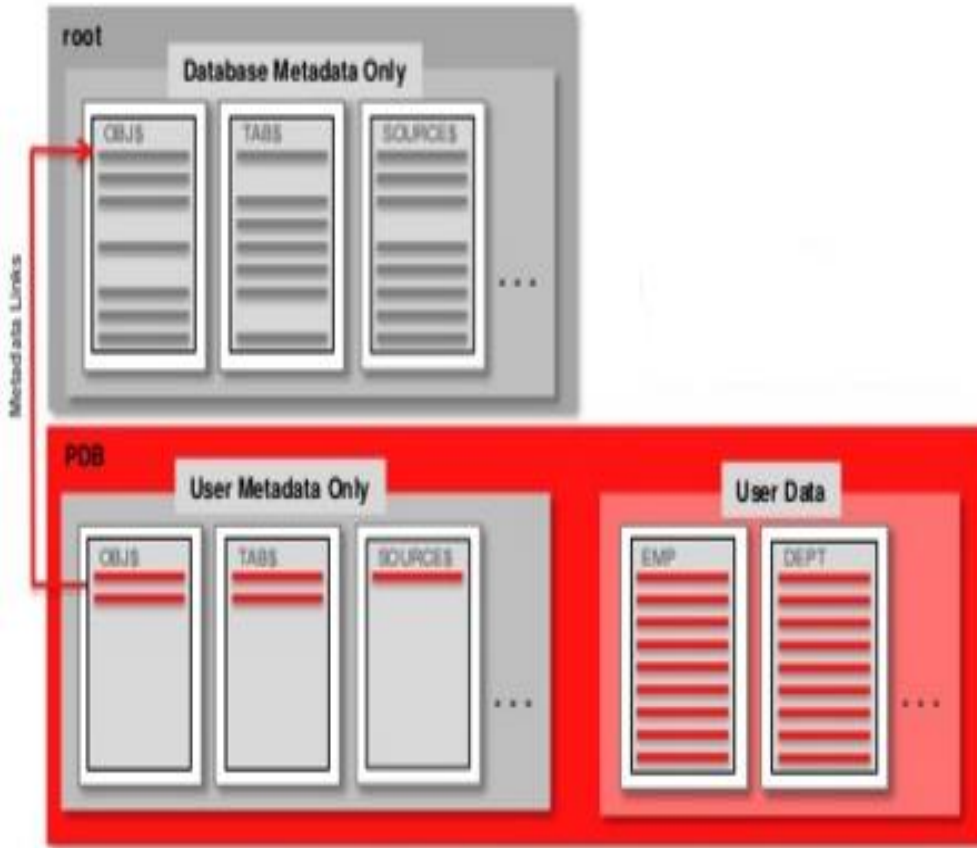
```
SQL> SHOW CON_ID;
```

```
CON_ID
```

```
-----
```

```
3
```

3. Data Dictionary Architecture



- 멀티테넌트에서는 수직적으로 파티셔닝 된 데이터 디렉터리 존재
- 오라클 시스템과 관련된 파라미터만 존재
- 오라클과 사용자 메타데이터는 혼재하게 됨

3. Data Dictionary Architecture

```
[oracle@db12cvm1 ~]$ sqlplus
SYS/oracle@db12cvm1:1521/orcl
AS SYSDBA
```

```
SQL> COL NAME FOR A15
SQL> SELECT NAME, CON_ID
FROM
V$ACTIVE_SERVICES ORDER BY 1;
```

NAME	CON_ID
-----	-----
SYS\$BACKGROUND	1
SYS\$USERS	1
orcl	1
orclXDB	1
pdb1	3
pdb2	4

```
[oracle@db12cvm1 ~]$ sqlplus
SYS/oracle@db12cvm1:1521/pdb1
AS SYSDBA
```

```
SQL> COL NAME FOR A15
SQL> SELECT NAME, CON_ID
FROM
V$ACTIVE_SERVICES ORDER BY 1;
```

NAME	CON_ID
-----	-----
pdb1	3

3. Container Data Objects

- 컨테이너 데이터 오브젝트는 다수의 컨테이너에 일괄 적용되는 테이블 또는 뷰 들을 포함
- 다른 컨테이너에 존재하는 데이터를 보지 못하게 하는 메커니즘이 존재
- 컨테이너 데이터 오브젝트의 오라클 제공 뷰는 **V\$** 혹은 **CDB_**로 시작

Container ID	Rows Pertain To
0	Whole CDB, or non-CDB
1	CDB\$ROOT
2	PDB\$SEED
All Other IDs	User-Created PDBs

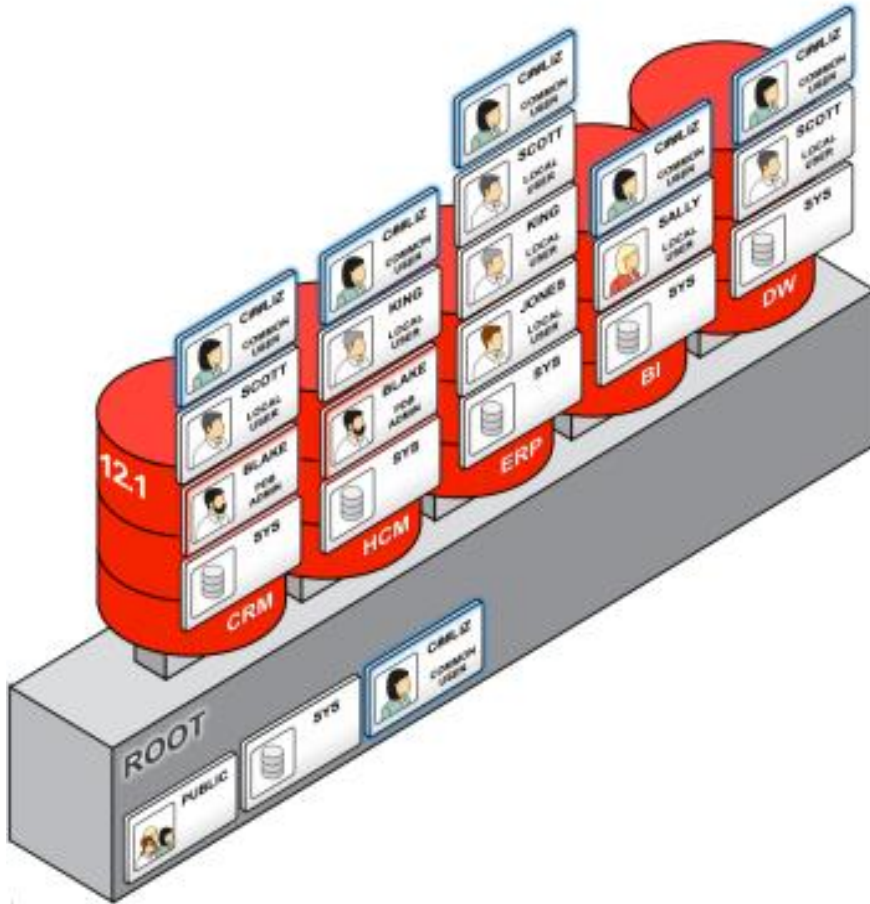
3. Performance Views

- **PDB**에서 조회시, **v\$** 뷰는 **PDB**에 한정된 정보, **CDB** 전반에 걸친 정보를 보여줌
- 루트에서 조회시, **v\$** 와 **CDB_** 뷰는 여러 컨테이너의 정보 조회가 가능
 - 이 뷰들의 정보에는 **con_id** 컬럼이 추가됨
 - **container_data view** 로 불림.
 - 커먼 유저의 경우 이에 대한 조회를 위해 **container_data** 설정 필요

Container ID	Rows Pertain To
0	Whole CDB, or non-CDB
1	CDB\$ROOT
2	PDB\$SEED
All Other IDs	User-Created PDBs

4. Managing CDB

유저(Users)



- **Local users** 가 **non-CDB** 의 유저와 동일
- 로컬 유저는 **PDB** 단위 안에만 생성
- 로컬 유저는 **PDB** 관리가 가능
- 커먼 유저는 루트에 선언하고 모든 **PDB**에 접근이 가능
- 커먼 유저는 어느 **PDB**에도 관리자로 접근이 가능, 단 “**Create Session**” 이 있어야 함
- 오라클 시스템은 커먼 유저에 의해 소유됨

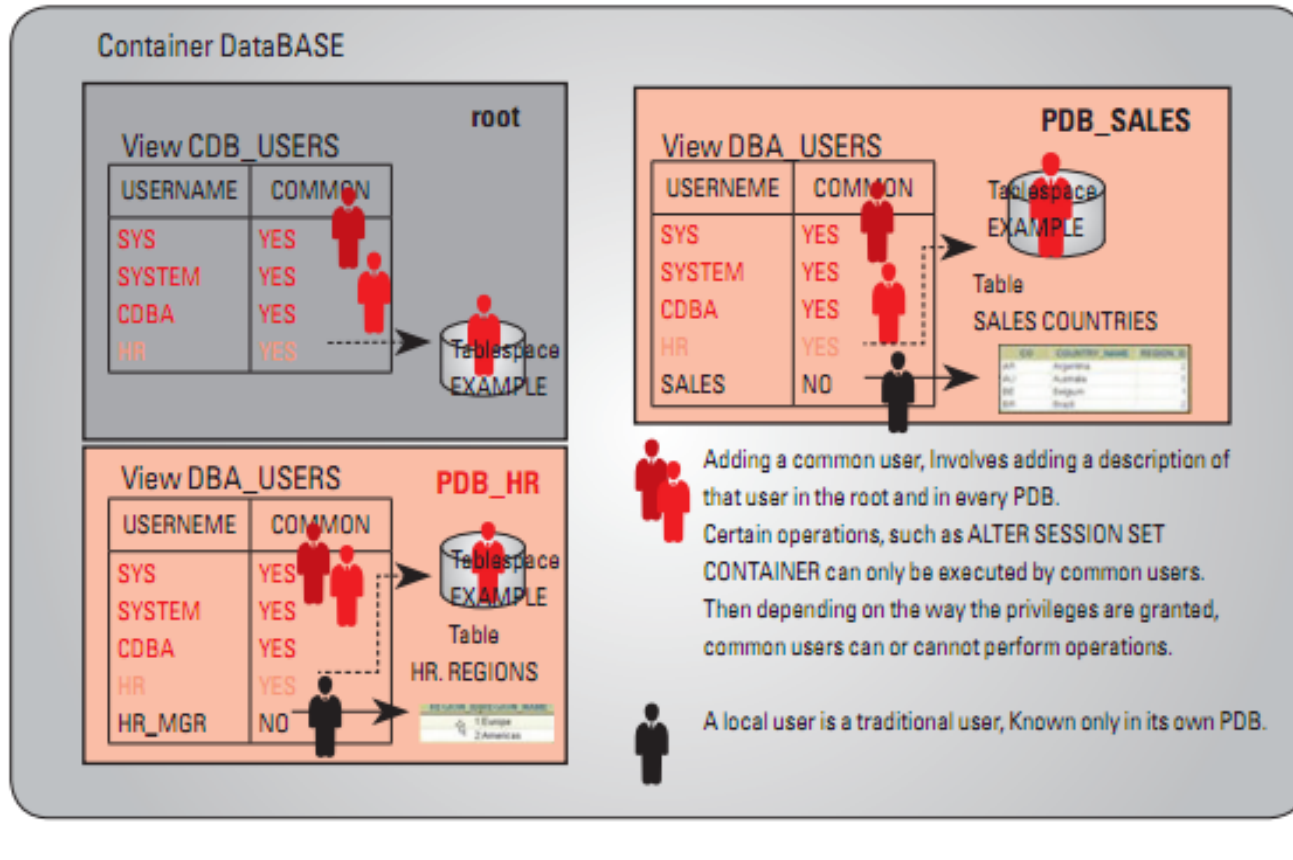
4. Managing CDB

유저의 권한(Users and Privileges)

- 모든 커먼 유저는 **c##** 혹은 **C##**로 시작
 - **SYS** 혹은 **SYSTEM** 같이 오라클이 제공하는 경우는 제외
- 모든 커먼 유저는 모든 컨테이너에 대해 유일성을 가짐
- 생성시 시드 **PDB**를 제외한 모든 **PDB**에 생성
- 커먼 유저의 스키마는 컨테이너마다 다를 수 있음
- **Best practice:** 커먼 유저의 스키마에는 오브젝트를 생성하지 말 것 !

4. Managing CDB

유저의 권한(Users and Privileges)



4. Managing CDB

Local Roles

- 로컬 역할은 **PDB**에 종속되어서 생성이 가능함
- 로컬 역할은 생성된 **PDB** 내에 한해서만 동작이 가능
-다른 **PDB**에서는 조회 불가

4. Managing CDB

Roles and Privileges

- 커먼 롤과 로컬 권한을 부여하는 것이 가능
- 권한 자체는 커먼이거나 로컬의 범위 내에만 있지 않음
 - 부여된 역할(**common, local**)에 한해서 동작
 - **CONTAINER=ALL** 을 통해 전위적인 적용이 가능

4. Managing CDB

Startup/Shutdown the Container Database

- **non-CDB** 인스턴스 종료와 동일함
 - 모든 **PDB**는 자동적으로 종료
 - **PDB** 조회를 위해서는 **CDB** 인스턴스가 마운트 이상이어야 함

- **CDB** 구동 역시 동일함
 - 기본적으로 **PDB**는 자동적으로 실행되지 않음

4. Managing CDB

Startup/Shutdown the Container Database

Shutdown PDB를 close한 후에 전체 인스턴스를 shutdown

```
alter pluggable database all close immediate;
```

```
select con_id, name, open_mode from v$pdb;
```

CON_ID	NAME	OPEN_MODE
2	PDB\$SEED	READ ONLY
3	SALESPDB	MOUNTED
4	HRPDB	MOUNTED

Startup Container Database를 startup 명령을 이용해 open한 후에 PDB를 open

```
alter pluggable database all open;
```

```
select con_id, name, open_mode from v$pdb;
```

CON_ID	NAME	OPEN_MODE
2	PDB\$SEED	READ ONLY
3	SALESPDB	READ WRITE
4	HRPDB	READ WRITE

4. Managing CDB

Using ALTER DATABASE Statement

- **ALTER DATABASE** 구문 역시 **CDB**와 **non-CDB**의 구분이 거의 없음
 - 대다수의 **ALTER DATABASE** 구문은 **CDB** 전체에 영향을 미침
- **ALTER DATABASE** 는
 - 전체 **CDB** 레벨
 - **PDB** 레벨에서 수행 가능
- **ALTER SYSTEM SET** 구문은 컨테이너 레벨에서 사용 가능
- **CDB** 는 기본 파라미터의 상속을 이용하여 운영됨
 - **PDB**는 루트로부터 값을 승계 받음
 - **V\$SYSTEM_PARAMETER** 의 **ISPDB_MODIFIABLE**이 **True**인 경우, 이를 **PDB**에서 수정하는 것이 가능
- 루트에서 운영시, 컨테이너 구절을 넣어 **PDB**를 지정할 수 있음

4. Managing PDB

Creating a PDB from the SEED

- **CREATE PLUGGABLE DATABASE** 구문을 통해 새로운 **PDB**를 생성 가능
 - 시드의 파일을 이용하여 복제를 수행하는데, 새로운 위치 및 변수 변환에 관한 명령어가 필요함
- **PDB**의 관리자를 **ADMIN USER**라는 이름으로 지정하는 것이 필요
 - **PDB_DBA** 역할을 부여하게 됨
- **PDB** 생성 후, 마운트 모드에서 **NEW**상태로 나타나게 됨
 - **READ WRITE** 모드로 열려야 작업이 가능
- **PDB** 와 동일한 이름의 서비스가 생성됨

4. Managing PDB

Creating a PDB from the SEED

```
[oracle@db12cvm1 ~]$ sqlplus SYS/oracle@ORCL AS SYSDBA
```

```
SQL> SHOW PDBS;
```

CON_ID	CON_NAME	OPEN MODE	RESTRICTED
2	PDB\$SEED	READ ONLY	NO
3	PDB1	READ WRITE	NO
4	PDB2	READ WRITE	NO

```
SQL> CREATE PLUGGABLE DATABASE PDB3 ADMIN USER PDB_ADMIN IDENTIFIED BY oracle  
FILE_NAME_CONVERT = ('pdbseed', 'PDB3');
```

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

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

```
SQL> ALTER PLUGGABLE DATABASE PDB3 OPEN;
```

```
Pluggable database altered.
```

4. Managing PDB

Creating a PDB by Cloning

- **CREATE PLUGGABLE DATABASE** 구문으로 복제가 가능
 - 새로운 타겟 PDB에 대한 명시가 필요함
- 소스 PDB 는 CDB에 플러그인 되어 있어야 함
- **FROM** 절에 소스 DB가 들어감
- 소스 PDB 는 로컬 CDB 혹은 원격 CDB에서도 가능
 - DB 링크를 명시하면 가능

```
create pluggable database HCMBI from HCM
```

```
create pluggable database HCMBI from HCM@ORCL;
```

4. Managing PDB

Creating a PDB by Cloning

```
SQL> CONNECT SYS/oracle@ORCL AS SYSDBA
Connected.
```

```
SQL> ALTER PLUGGABLE DATABASE PDB1 CLOSE IMMEDIATE;
Pluggable database altered.
```

```
SQL> ALTER PLUGGABLE DATABASE PDB1 OPEN READ ONLY;
Pluggable database altered.
```

```
SQL> CREATE PLUGGABLE DATABASE PDB1_CLONE FROM PDB1 FILE_NAME_CONVERT = ( 'pdb1',
'pdb1_clone' );
Pluggable database created.
```

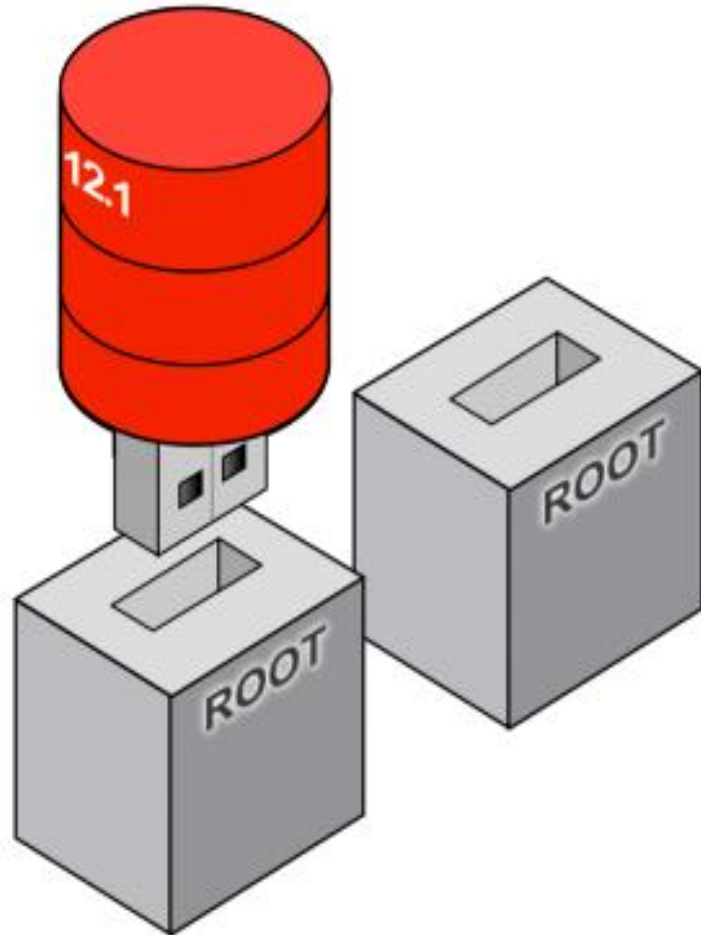
```
SQL> ALTER PLUGGABLE DATABASE PDB1_CLONE OPEN FORCE;
Pluggable database altered.
```

```
SQL> SHOW PDBS;
```

CON_ID	CON_NAME	OPEN MODE	RESTRICTED
2	PDB\$SEED	READ ONLY	NO
3	PDB1	READ ONLY	NO
6	PDB1_CLONE	READ WRITE	NO

4. Managing PDB

Unplug / plug



- PDB의 메타데이터만 있으면 CDB 간에 PDB를 옮기는 작업은 수월하게 진행 가능
- 언플러그 된 PDB는 설정 및 opatch, 암호화 키를 갖고 이동함

4. Managing PDB

Unplug / plug

Unplug

```
alter pluggable database HCM  
unplug into '/u01/app/oracle/oradata/.../hcm.xml'
```

Plug

```
create pluggable database My_PDB  
using '/u01/app/oracle/oradata/.../hcm.xml'
```

4. Managing PDB

Unplug

```
SQL> CONNECT SYS/oracle@ORCL AS SYSDBA
Connected.
```

```
SQL> SHOW PDBS;
```

CON_ID	CON_NAME	OPEN MODE	RESTRICTED
2	PDB\$SEED	READ ONLY	NO
3	PDB1	READ ONLY	NO
4	PDB2	READ WRITE	NO
5	PDB3	READ WRITE	NO
6	PDB1_CLONE	READ WRITE	NO

```
SQL> ALTER PLUGGABLE DATABASE PDB1_CLONE CLOSE IMMEDIATE;
```

```
Pluggable database altered.
```

```
SQL> ! mkdir /u01/app/oracle/oradata/pdb_manifests;
```

```
SQL> ALTER PLUGGABLE DATABASE PDB1_CLONE UNPLUG INTO
'/u01/app/oracle/oradata/pdb_manifests/pdb1_clone.xml';
```

```
Pluggable database altered.
```

4. Managing PDB

Unplug

```
SQL> DROP PLUGGABLE DATABASE PDB1_CLONE KEEP DATAFILES;
```

```
Pluggable database dropped.
```

```
SQL> SHOW PDBS;
```

CON_ID	CON_NAME	OPEN MODE	RESTRICTED
2	PDB\$SEED	READ ONLY	NO
3	PDB1	READ ONLY	NO
4	PDB2	READ WRITE	NO
5	PDB3	READ WRITE	NO

4. Managing PDB

plug

```
SQL> CONNECT SYS/oracle@ORCL AS SYSDBA
Connected.
```

```
SQL> SHOW PDBS;
```

CON_ID	CON_NAME	OPEN MODE	RESTRICTED
2	PDB\$SEED	READ ONLY	NO
3	PDB1	READ ONLY	NO
4	PDB2	READ WRITE	NO
5	PDB3	READ WRITE	NO

```
SQL> CREATE PLUGGABLE DATABASE PDB4 USING
'/u01/app/oracle/oradata/pdb_manifests/pdb1_clone.xml' NOCOPY;
Pluggable database created.
```

```
SQL> SHOW PDBS;
```

CON_ID	CON_NAME	OPEN MODE	RESTRICTED
2	PDB\$SEED	READ ONLY	NO
3	PDB1	READ ONLY	NO
4	PDB2	READ WRITE	NO
5	PDB3	READ WRITE	NO
6	PDB4	MOUNTED	

```
SQL> ALTER PLUGGABLE DATABASE PDB4 OPEN;
Pluggable database altered.
```

Thank you.