multisoft

Inside Oracle® 12c Multitenant Environnent

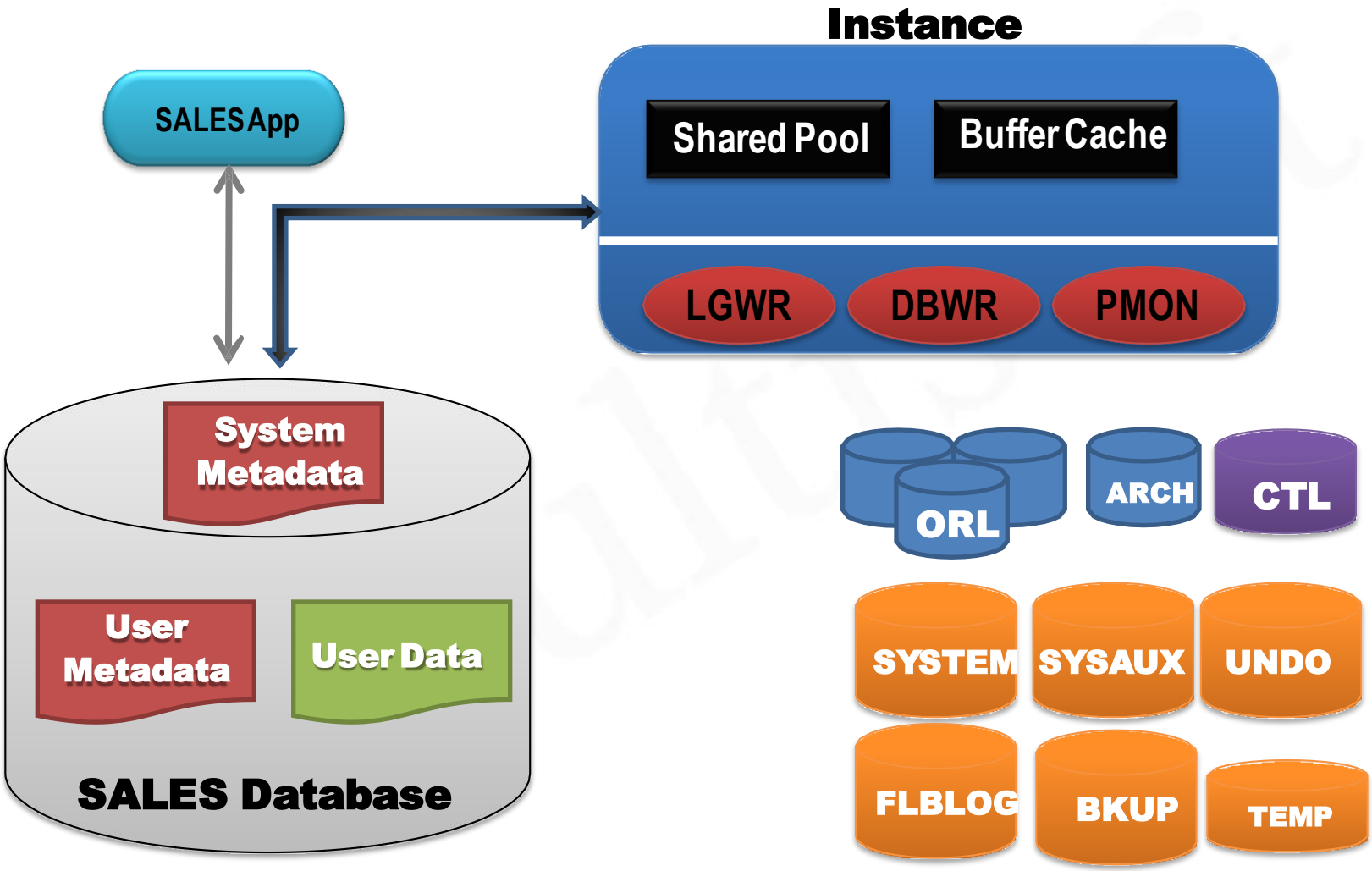
Based on

ORACLE® 12cR1 And 12cR2

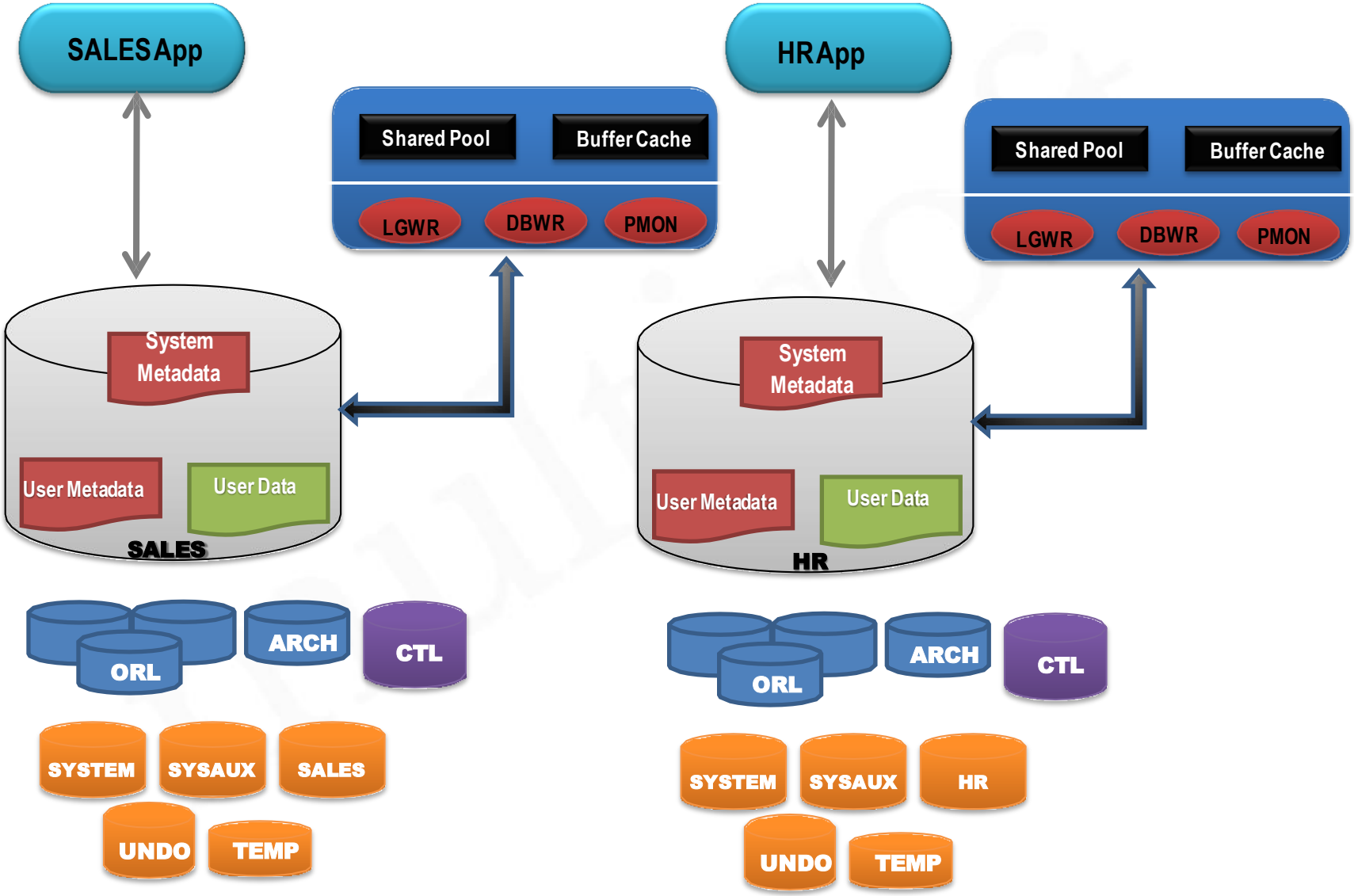
Agenda of today's interaction

- Introduction to pre 12.1 Databases
- Over view of Oracle 12c Multitenant Architecture Environment.
- Benefits of the Multitenant Architecture
- Prerequisites for a Multitenant Environment.
- Separation of Duties in CDB and PDB Administration
- Data Dictionary Architecture in a CDB
- Overview of Common and Local Users in a CDB/PDB
- Creation of CDB and PDB
- How to start ,Stop, Manage CDB and PDBs.
- Q & A

NON CDB Architecture :



NON CDB Architecture



Not really!!

- *Ideally*, ***one*** server should host just ***one*** database

– ***But*** world **!=** ideal

- In ***real-world***-server to database ratio **=1:N**
- Individual databases need individual resource allocations
- Resource requirements are never same
- **Consolidation**, with 1:N-becomes a major challenge
- Un-necessary system metadata duplication
- **Provisioning** remains a challenge
- **Patching** remains a challenge


11g


12c
©PA

- Newest release-came out in **March, 2017**
- About **500** new features
- Introduced the concept of ***Multi-tenancy***
- Forget ***g*** as ***c*** is in the house
- Brings ***clouds*** a little more closer ☺

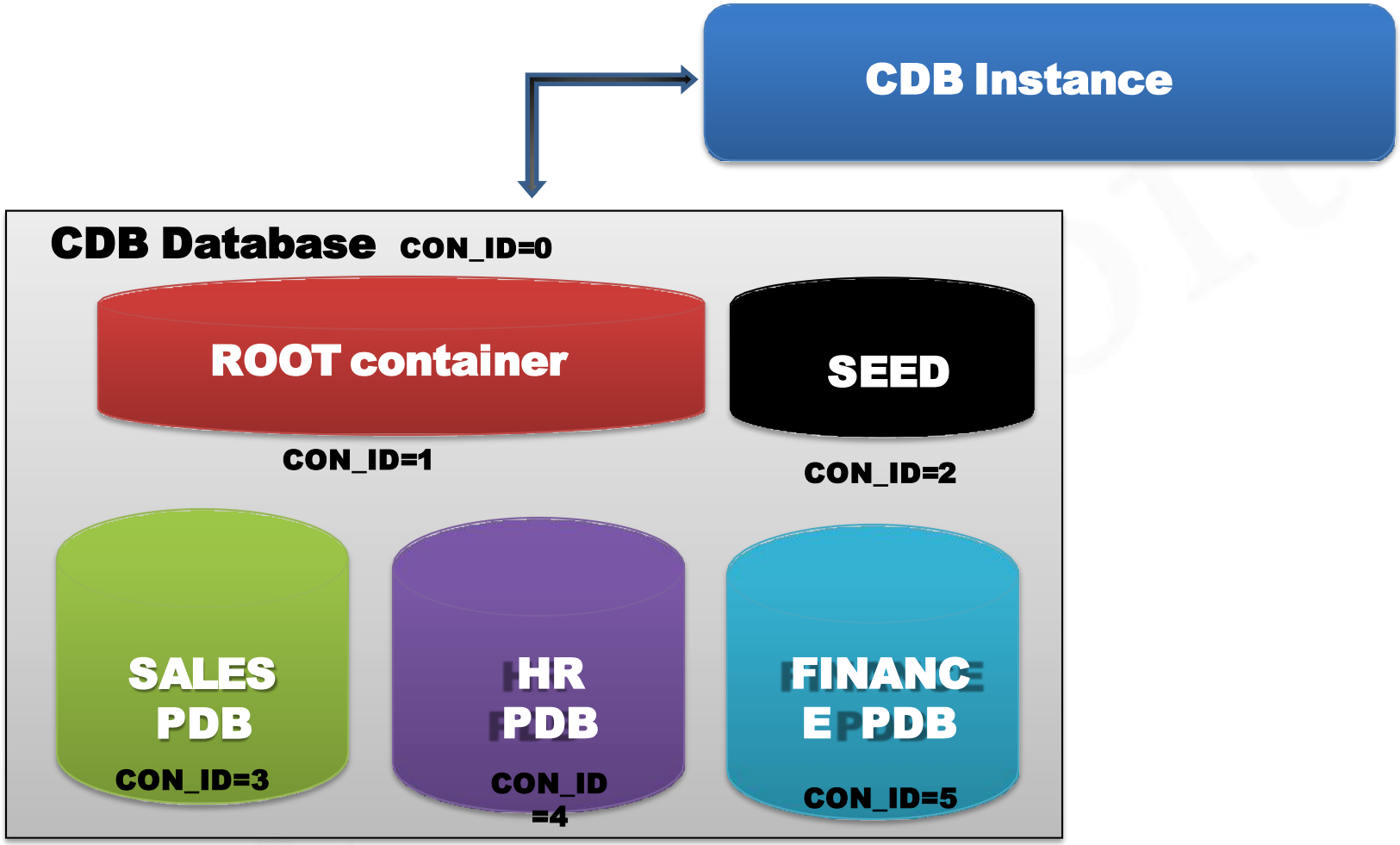
Multi-Tenant Oracle Database 12c

- ***First*** Oracle database release for ***cloud*** and ***consolidation***
- Allows multiple ***tenant*** databases under one main ***container***
- Tenants *share* the SGA, background processes & data-dictionary
- Reduced resource sharing
  - Less storage space
  - Less RAM
    - Lesser **cost**
- Rapid
  - Provisioning
  - Migration
- Much faster Patching and Upgrading
- Database privacy remains intact

- Full support for RAC & Data Guard features
- Compatible with non-CDB databases working
- Easy GUI management using
  - **Cloud Control 12c**
  - **EM Express**
  - **SQL Developer**
- Ease in Database Administration
- **DBaaS**
- Resource Manager for *both* CDB and PDB
- **Manage many as one!!**

Multitenant Database Architecture(*simplified*)

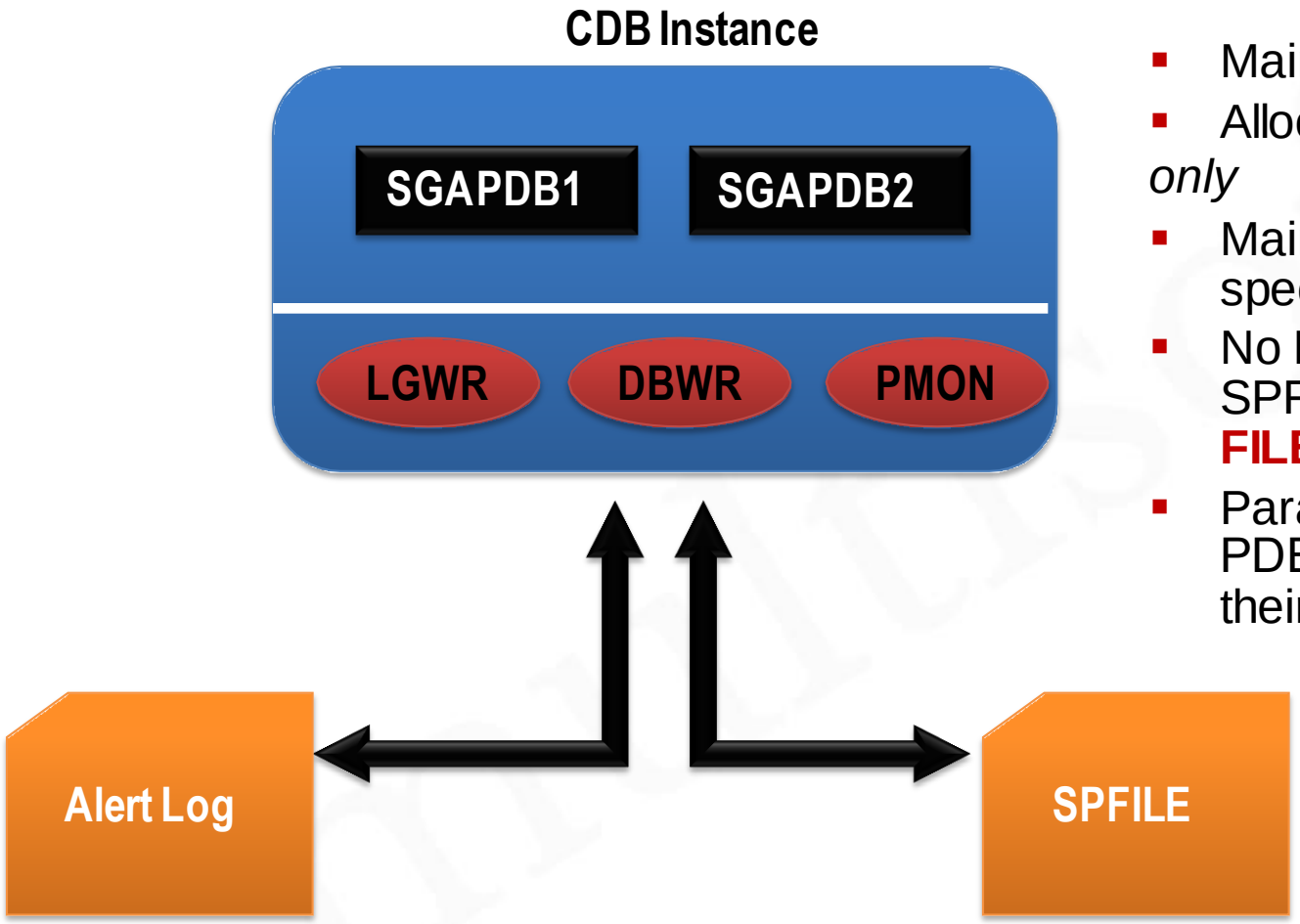
multisoft



Containers in a Multitenant Database

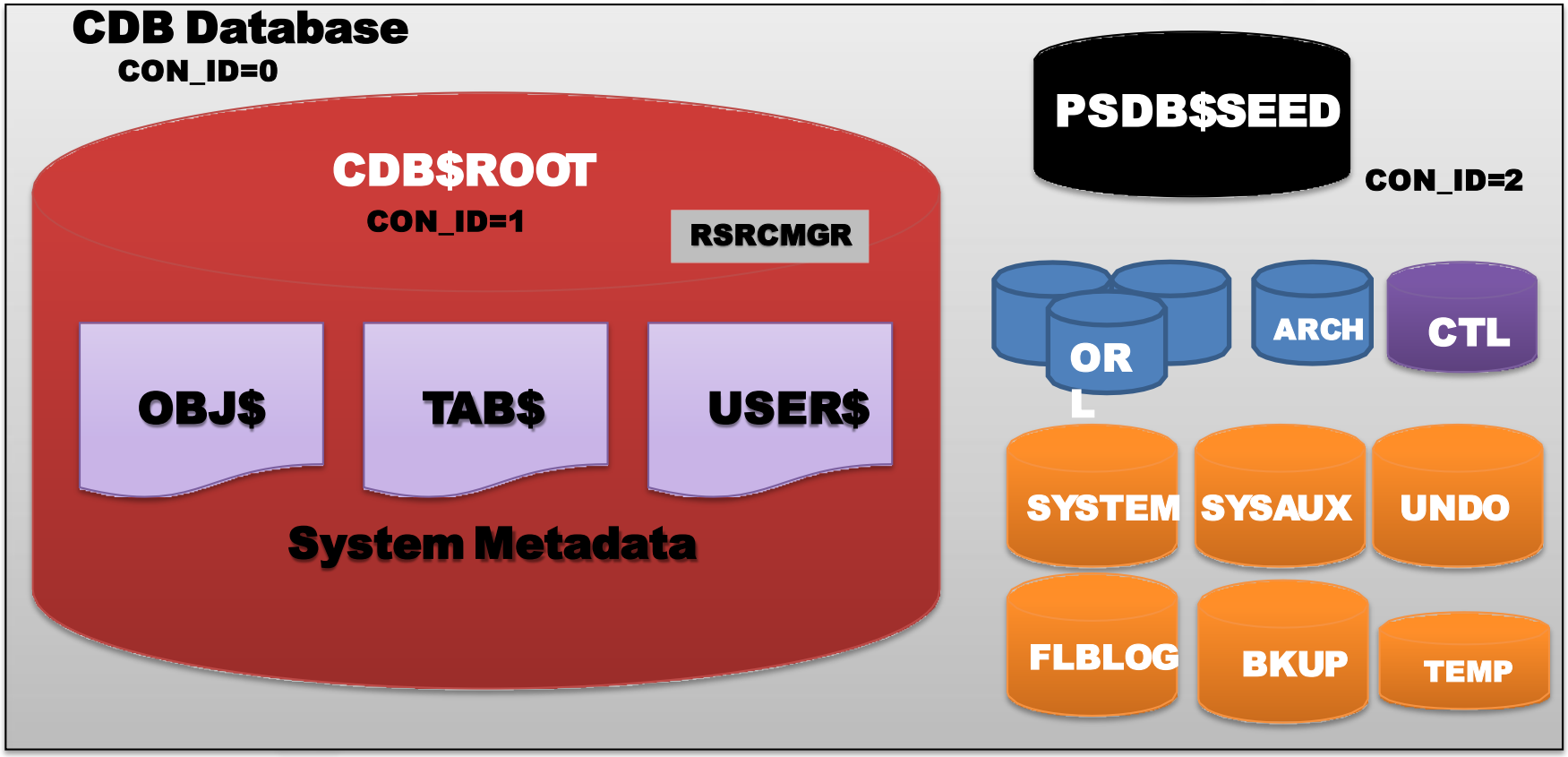
- A multitenant CDB database always contains
  - **ROOT container**
  - **SEED container**
- A CDB *may* also contain
  - **PDB container**
- An Instance is allocated for the whole CDB database
- **V$CONTAINERS**

CDB Instance



- Maintained by **SYS** user
- Allocated for the CDB *only*
- Maintained by CDB specific **SPfile/PFILE**
- No PDB specific SPFILE/PFILE(**PDB_SPFILE$)**
- Parameter(s) changed at PDB level don't go up to their CDB database

Root Container CDB$ROOT



Oracle 12c Single Tenant:

A CDB is a container

- CON_ID=0 is the CDB

that contains containers

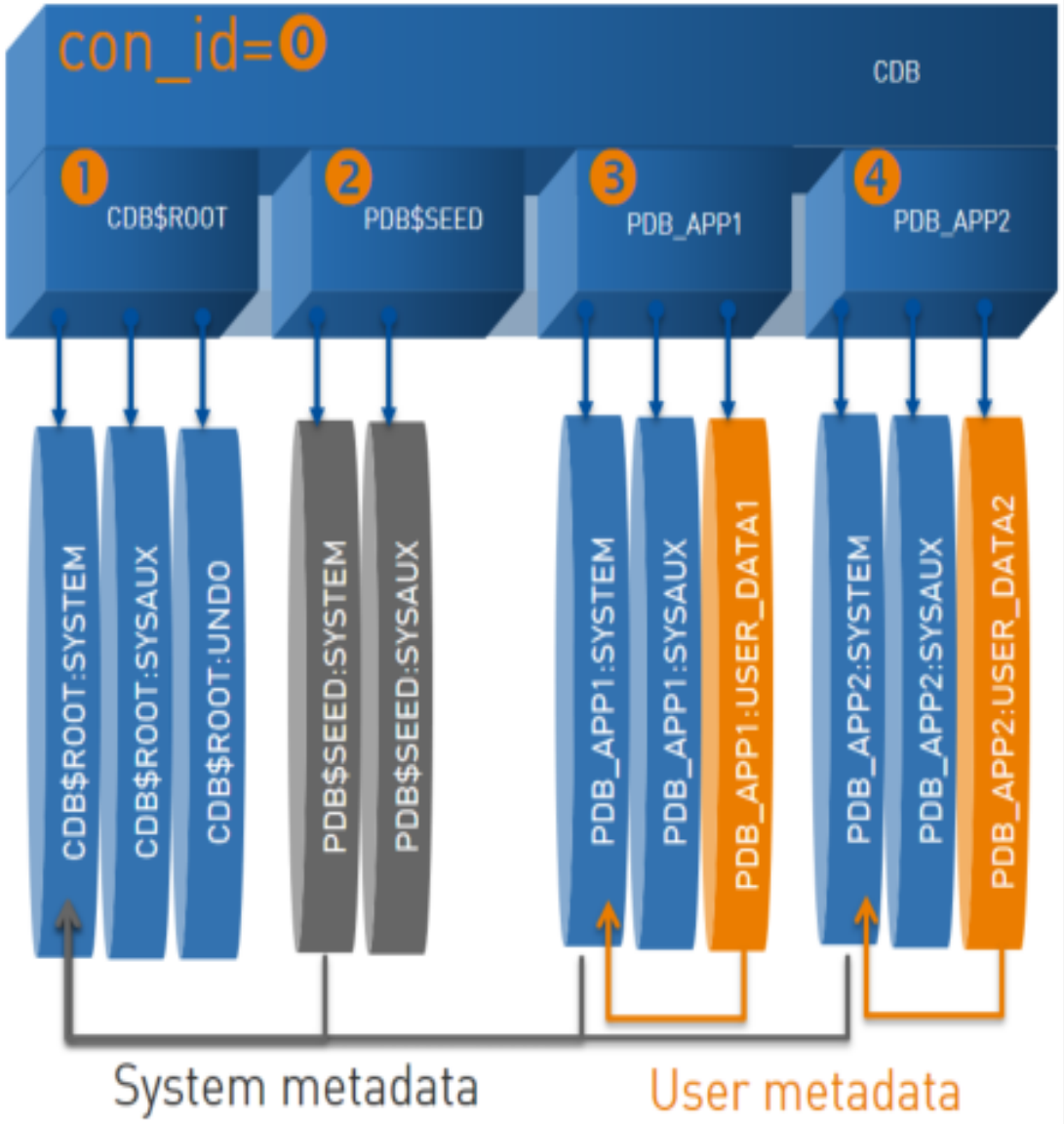
- CON_ID=1 is CDB$ROOT
- CON_ID=2 is PDB$SEED
- CON_ID=3 is the first PDB
- ...

Containers where con_id ⩾ 2

- are PDBs (pluggable databases)

CDB having max(con_id)= 3

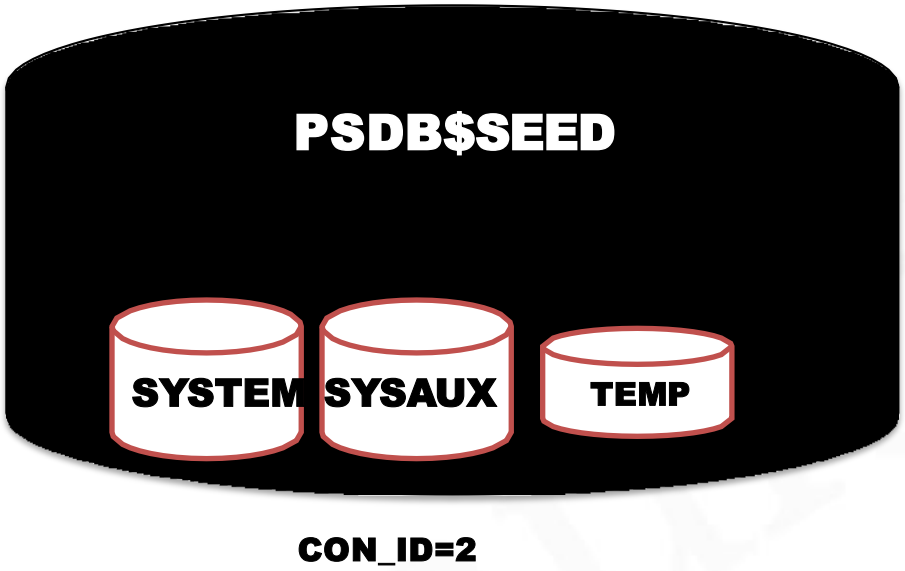
- is Single-Tenant (no option)
- Controlled by max_pdbs=1



Root Container CDB$ROOT

- The very first tenant within a multitenant database CDB$ROOT
- Contains SGA & PGA-*shared* among all the PDB's
- Contains data files, online redo logs, control files, Undo
- Contains ***system*** metadata for Oracle supplied objects
- Contains ***system*** **schemas** & **roles**
- Contains information about underlying PDB's
- Contains *common users*
- Should ***not*** contain user data
- Backup support by **RMAN**
- Managed by a Container DBA(CDBA)

Seed Container PDB$SEED



- *Default* Pluggable database
- Created at the time of creation of the multitenant database
- Named as **PDB$SEED**
- Acts as a template for creating pluggable databases in the future
- Users/DBA's can't create and modify objects in the seed database
- Can't be deleted
- One CDB contains only *one* SEED container
- Backup support by RMAN

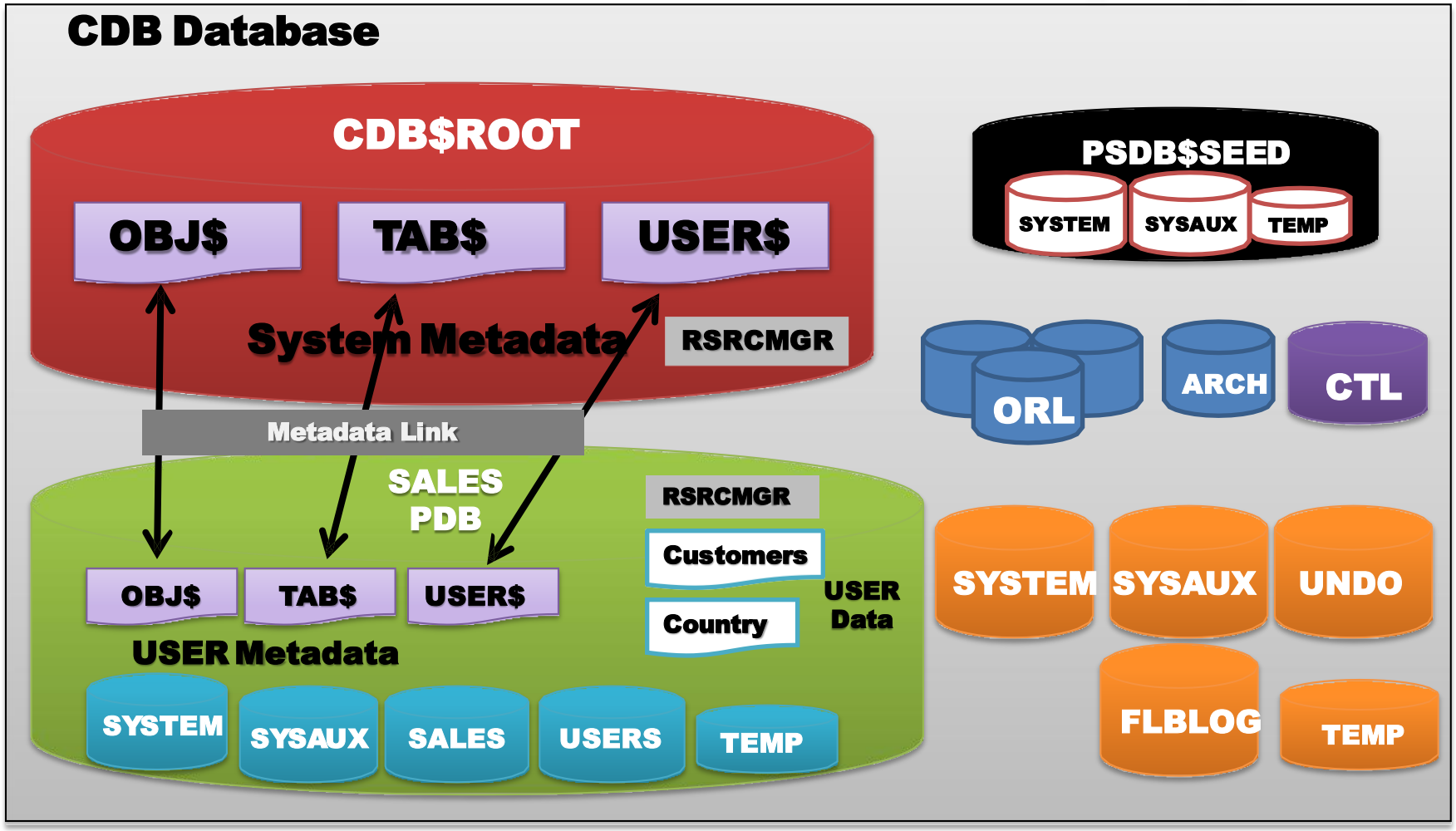
Meet The Tenants via SQL

multisoft

```
SQL> select con_id, dbid, name from V$database;

    CON_ID       DBID       NAME
---------- ---------- ---------
         0 4244090852 PAWCDB

SQL> select con_id, dbid,
name from  v$containers;

    CON_ID       DBID       NAME
---------- ---------- ----------------
         1 4244090852 CDB$ROOT
         2 4063311135 PDB$SEED
         3 2577910186 PAWPDB
```

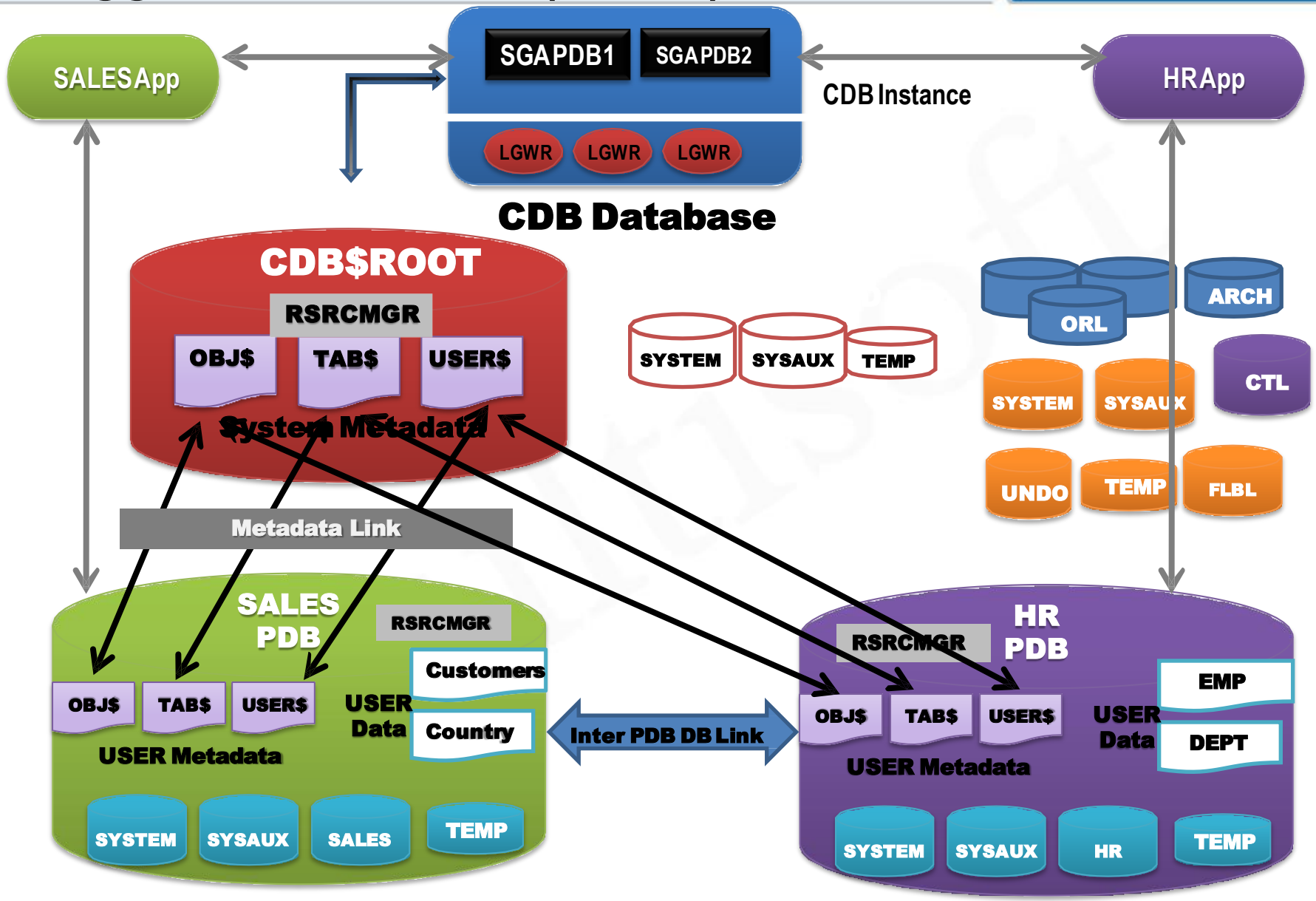
Pluggable Containers(PDB's)

multisoft



Pluggable Containers(PDB's) Contd.....

multisoft

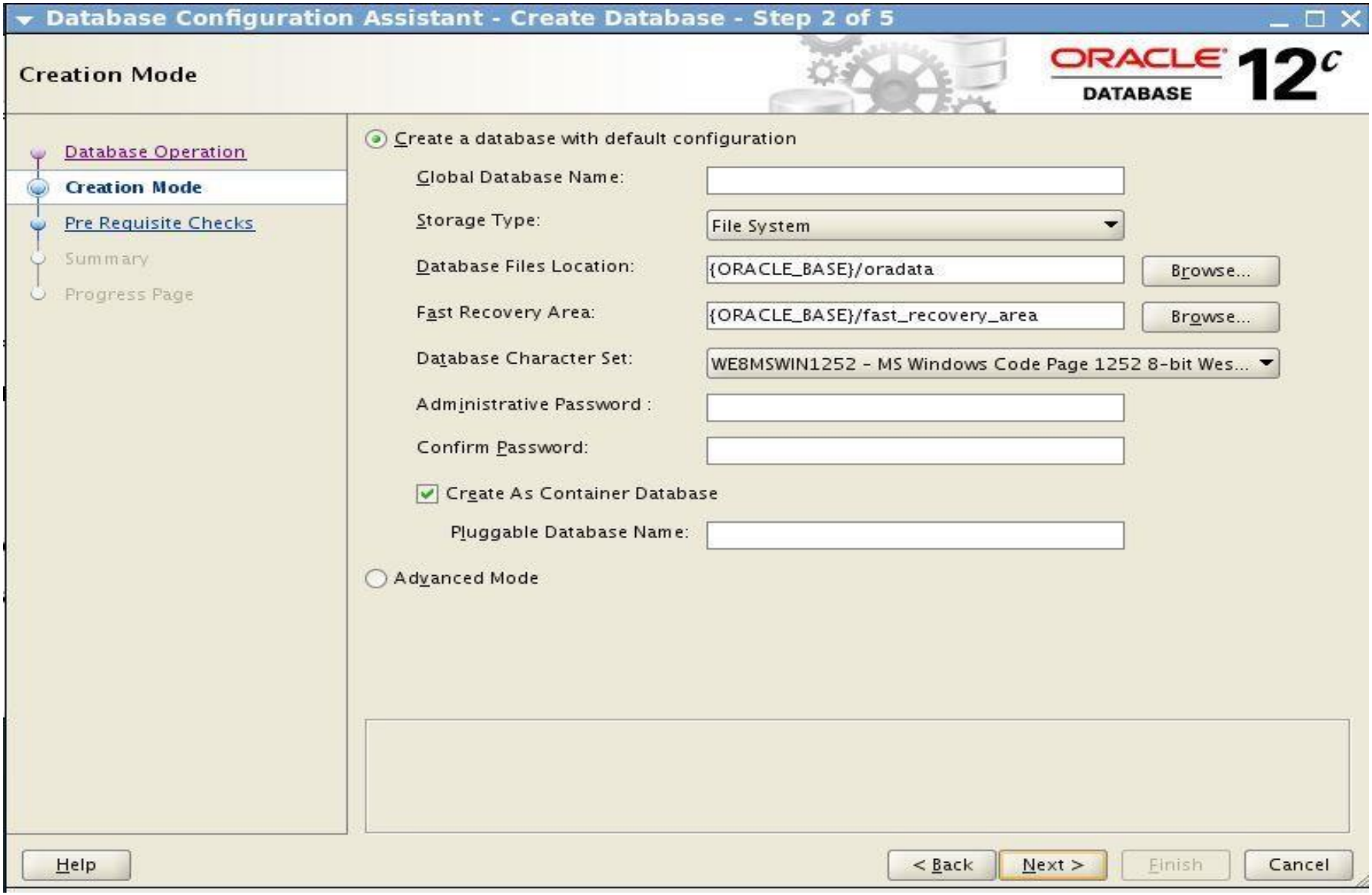


Pluggable Containers(PDB's)

- User defined and created
- Explicitly named
- Stores user defined objects e.g. tables, indexes etc
- Contains data dictionary holding information about **user data**
- ***Pointers*** are maintained to the system metadata container
- **Service Naming** is used for connections
- Contains ***local users***-specific to exactly one PDB
- PDB specific Resource Manager
- In Oracle **12.1.0.2 ,253** PDB's(including **SEED**) maximum are allowed in one CDB, but in 12.2.0.1 , **4096** PDB's(including **SEED**) maximum are allowed in one CDB
- Managed by PDB DBA(PDBA)
- Linked with other PDB's using *inter-PDB* **DB links**
- **V$PDBS , DBA_PDBS**

Creating Your 1st Multi-Tenant Database

- There are several ways
- ✓Using **OUI**(at the time of the installation)
- ✓Using **DBCA**
- ✓Using **CREATE DATABASE** command


Database Configuration Assistant - Create Database - Step 2 of 5
Creation Mode
ORACLE DATABASE 12c
Database Operation
Creation Mode
Pre Requisite Checks
Summary
Progress Page
Create a database with default configuration
Global Database Name:
Storage Type:
File System
Database Files Location:
{ORACLE_BASE}/oradata
Browse...
Fast Recovery Area:
{ORACLE_BASE}/fast_recovery_area
Browse...
Database Character Set:
WE8MSWIN1252 - MS Windows Code Page 1252 8-bit Wes...
Administrative Password :
Confirm Password:
Create As Container Database
Pluggable Database Name:
Advanced Mode
Help
< Back
Next >
Finish
Cancel

Creating CDB using CREATE DATABASE

- The **CREATE DATABASE** command is *almost* the same
- **ENABLE PLUGGABLE DATABASE** clause must be used
- **SEED FILE_NAME_CONVERT** clause (only if *not* using OMF)
- **DB_CREATE_FILE_DEST** initialization parameter if using OMF

OR

- **PDB_FILE_NAME_CONVERT** initialization parameter

CREATE DATABASE *with* OMF

multisoft

```
SQL>
    CREATE DATABASE pawcdb
   USER SYS IDENTIFIED BY ORCL
   USER SYSTEM IDENTIFIED BY ORCL
   EXTENT MANAGEMENT LOCAL
   DEFAULT TABLESPACE users
   DEFAULT TEMPORARY TABLESPACE temp
   UNDO TABLESPACE undotbs1
   ENABLE PLUGGABLE DATABASE
   SEED
   SYSTEM DATAFILES SIZE 300M AUTOEXTEND ON NEXT
   10M MAXSIZE UNLIMITED SYSAUX DATAFILES SIZE
   200M;

   SQL> @?/rdbms/admin/catcdb.sql*
```

*~~17033183.8~~

CREATE DATABASE *without* OMF


```
SQL>CREATE DATABASE pawcdb
USER SYS IDENTIFIED BY ORACLE  USER SYSTEM
IDENTIFIED BY ORACLE
....
ENABLE PLUGGABLE DATABASE SEED FILE_NAME_CONVERT =
('/u01/app/oracle/oradata/acdb/',
'/u01/app/oracle/oradata/pdbseed/') SYSTEM
DATAFILES SIZE 300M AUTOEXTEND ON NEXT 10M MAXSIZE
UNLIMITED SYSAUX DATAFILES SIZE 200M USER_DATA
TABLESPACE usertbs DATAFILE
'/u01/app/oracle/oradata/pdbseed/usertbs01.dbf'
SIZE 200M REUSE AUTOEXTEND ON MAXSIZE UNLIMITED;

SQL>@?/rdbms/admin/catcdb.sql
```

- One **ROOT** container, one **SEED** container
- One database service per container
- **Local** & **Common** Users/Roles
- ***New*** Data Dictionary views
- Tablespaces

Common vs Local

multisoft

- **Common user**
  - *Must* start from **C##** or **c##**
  - Are available in *all* the containers
  - Is defined within the
- **CDB$ROOT**
  - Works for *both* Root and
- underlying PDB's
  - Default schemas SYS, SYSTEM are common to all containers
  - CREATE USER….CONTAINER=**ALL**

- **Local user**
  - Doesn't need any prefix
  - Are available only in the PDB in which they are created
  - Is defined within the
- **specific PDB** itself only
  - Can *connect* and *work* for only their specific PDB
  - CREATE USER….CONTAINER=<PDB>
- (from Root)
  - CREATE USER(from **PDB**)

Data Dictionary Changes

- **CDB_***
- – Dictionary views showing data from *Root* and all
- *Pluggable* databases
- **DBA_***
- – Dictionary views showing data from a *container* or from
- *Pluggable* databases
- **ALL_***
- – Dictionary views showing data *accessible* by a users
- **USER_***
- – Dictionary views showing data *owned* by a user

Which Container We Are Into Now?

multisoft

```
SQL> show con_id

CON_ID
------------------------------
1
SQL> show con_name
CON_NAME
------------------------------
PAWPDB

SQL> conn sys/oracle@pawpdb as sysdba
Connected.

SQL>  show con_id

CON_ID
------------------------------
3
SQL> conn sys/oracle@pawcdb as sysdba
Connected.

SQL> alter session set container = pawpdb;

Session altered.

SQL> show con_id

CON_ID
------------------------------
3
```

Look What You Ask For

- **SQL> conn / as sysdba**

```
Connected.
```

- **SQL> show con_id**

```
CON_ID
------------------------------
1
```

- **SQL> select count(username) from dba_users;**

```
COUNT(USERNAME)
---------------
             36
```

- **SQL> select count(username) from cdb_users;**

```
COUNT(USERNAME)
---------------
            119
```

contd.

Look What You Ask For

- **SQL> connect sys/oracle@pawpdb as sysdba**
  Connected.

- **SQL> show con_id**

```
CON_ID
------------------------------
3
```

- **SQL> select count(username) from dba_users;**

```
COUNT(USERNAME)
---------------
             48
```

- **SQL> select count(username) from cdb_users;**

```
COUNT(USERNAME)
---------------
             48
```

CDB_SERVICES


```
SQL> select name from cdb_services;

NAME
---------------
orclpdb1.airydba
SYS$BACKGROUND
SYS$USERS
orclXDB
orcl.airydba
orclpdb.airydba

6 rows selected.
```

512 maximum services

CDB_TABLESPACES


```
SQL> select b.con_id, a.pdb_id, a.pdb_name,b.TABLESPACE_NAME
       from dba_pdbs a
       full outer join cdb_tablespaces b
       on a.pdb_id=b.con_id
       order by b.con_id;
```

CON_ID	PDB_ID	PDB_NAME	TABLESPACE_NAME
1			SYSTEM
1			SYSAUX
1			UNDOTBS1
1			TEMP
1			USERS
	2	PDB$SEED	
	4	ORCLPDB	
	3	ORCLPDB1	

8 rows selected.

@detail.sql

Managing & Provisioning PDB's

- ✓Using SQL commands
- ✓Using SQL Developer
- ✓Using Multi-tenant Self Service Provisioning
- application(Beta)
- ✓Using Database Configuration Assistant(DBCA)

Managing PDB's using SQL

multisoft

```
SQL> alter pluggable database all open;
   Pluggable database altered.
SQL> alter pluggable database all close;
   Pluggable database altered.

SQL> select name, open_mode from v$pdbs;
    NAME             OPEN_MODE
    -------------- ----------
    PDB$SEED  READ ONLY
    ORCLPDB1  READ WRITE
    ORCLPDB   READ WRITE
SQL> alter pluggable database orclpdb close;
   Pluggable database altered.

SQL> alter pluggable database orclpdb open read only;

   Pluggable database altered.

SQL> select name, open_mode from v$pdbs;
    NAME             OPEN_MODE
    -------------- ----------
    PDB$SEED  READ ONLY
    ORCLPDB1  READ WRITE
    ORCLPDB   READ ONLY
```

Querying Info About The PDB's

multisoft

```
SQL> select con_id,con_uid, name from V$pdbs;

    CON_ID    CON_UID NAME
---------- ---------- ----------------
         2 4063311135 PDB$SEED
         3 2577910186 ORCLPDB
SQL> column pdb_name format a20

SQL> select pdb_id, pdb_name, con_uid from pda_pdbs;

    PDB_ID     PDB_NAME          CON_UID
---------- ------------ -----------------
         3        ORCLPDB        2577910186
         2        PDB$SEED       4063311135

SQL> show pdbs

    CON_ID CON_NAME                       OPEN MODE  RESTRICTED
---------- ------------------------------ ---------- ----------
         2 PDB$SEED                       READ ONLY  NO
         3 ORCLPDB1                       READ WRITE NO
         4 ORCLPDB                        READ ONLY  NO
```

Manually Provisioning PDB's

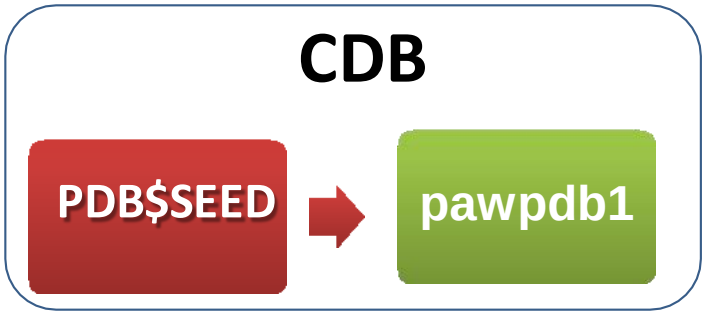
I. By using **SEED** container
II. By ***cloning*** an existing PDB
III. By ***plugging-in*** an unplugged PDB
IV. By plugging-in a ***non-CDB*** into a ***CDB***

Creating PDB using SEED container

multisoft

```
SQL> create pluggable database  pawpdb1 admin user pawan
       identified by  airy
       file_name_convert = ('pdbseed', 'pawpdb1');

Pluggable database created.
```

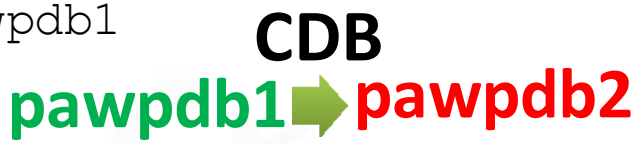
```
SQL> select con_id, name from V$pdbs;

    CON_ID NAME
---------- ---------------
         2 PDB$SEED
         3 ORCLPDB1
         4 ORCLPDB
         6 PAWPDB1
```



Creating PDB using existing PDB

multisoft

```
SQL> create pluggable database pawpdb2 from pawpdb1;
create pluggable database pawpdb2 from pawpdb1
*
ERROR at line 1:
ORA-65081: database or pluggable database is not open in
read only mode
```




```
SQL> show pdbs
    CON_ID CON_NAME                       OPEN MODE  RESTRICTED
---------- ------------------------------ ---------- ----------
         2 PDB$SEED                       READ ONLY  NO
         3 ORCLPDB1                       READ WRITE NO
         4 ORCLPDB                        READ ONLY  NO
         6 PAWPDB1                        READ WRITE NO

SQL> alter pluggable database pawpdb1 close;
Pluggable database altered.
SQL> alter pluggable database pawpdb1 open read only;
Pluggable database altered.
SQL> create pluggable database pawpdb2 from pawpdb1;
Pluggable database created.
```

Creating PDB using existing PDB Contd…

multisoft

```
SQL> create pluggable database pawpdb2 from pawpdb1
file_name_convert = ('/u01/app/oracle/oradata/orcl/pawpdb1',
'/u01/app/oracle/oradata/orcl/pawpdb2');
Pluggable database created.
SQL> show pdbs
    CON_ID CON_NAME                       OPEN MODE  RESTRICTED
---------- ------------------------------ ---------- ---------
         2 PDB$SEED                       READ ONLY  NO
         3 ORCLPDB1                       READ WRITE NO
         4 ORCLPDB                        READ ONLY  NO
         5 PAWPDB2                        MOUNTED
         6 PAWPDB1                        READ WRITE NO
SQL> alter pluggable database pawpdb2 open;
Pluggable database altered.
SQL> show pdbs
    CON_ID CON_NAME                       OPEN MODE  RESTRICTED
---------- ------------------------------ ---------- ---------
         2 PDB$SEED                       READ ONLY  NO
         3 ORCLPDB1                       READ WRITE NO
         4 ORCLPDB                        READ ONLY  NO
         5 PAWPDB2                        READ WRITE NO
         6 PAWPDB1                        READ WRITE NO
```

Creating PDB using an Unplugged PDB

multisoft

```
SQL> alter pluggable database PAWPDB1 close;
Pluggable database altered.

SQL> alter pluggable database pawpdb1 unplug into
'/home/oracle/pawpdb1.xml';
Pluggable database altered.

SQL> drop pluggable database pawpdb1 keep datafiles;
Pluggable database dropped.

SQL> show pdbs
    CON_ID CON_NAME                       OPEN MODE  RESTRICTED
---------- ------------------------------ ---------- ----------
         2 PDB$SEED                       READ ONLY  NO
         3 ORCLPDB1                       READ WRITE NO
         4 ORCLPDB                        READ ONLY  NO
         5 PAWPDB2                        READ WRITE NO

SQL> create pluggable database pawpdb3 using '/home/oracle/pawpdb1.xml'
file_name_convert=('pawpdb1','pawpdb3');
Pluggable database created.
```

“OR”

```
SQL> create pluggable database pawpdb3 using '/home/oracle/pawpdb1.xml'
NOCOPY TEMPFILE REUSE;
```

Creating PDB using an non-CDB

In non-CDB(12c) database:

Pre-12.1

- Data-Pump
- Replication

```
SQL> alter database open read only;
Database altered.

SQL>exec dbms_pdb.describe('orcl12.xml');
PL/SQL procedure successfully completed.
```

In 12c CDB database:

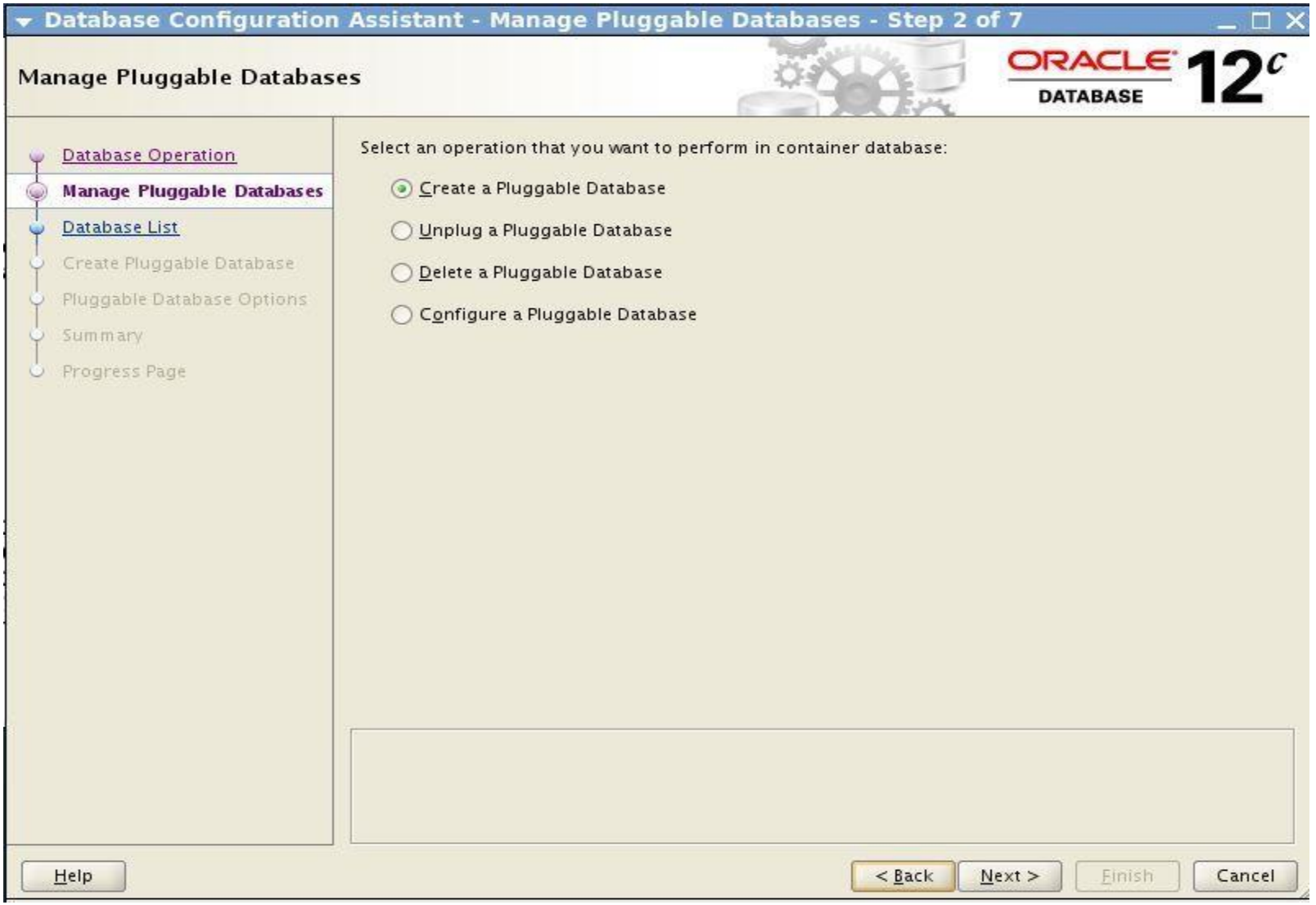
```
SQL> create pluggable database orcl12 using
'/u01/app/oracle/product/12.1.0/dbhome_1/dbs/orcl12.xml';

Pluggable database created.

SQL>alter session set container=orcl12;
Session altered.
SQL>@?/rdbms/admin/noncdb_to_pdb.sql
```

Dropping a PDB


```
SQL> drop pluggable database pawpdb2 including datafiles;
drop pluggable database pawpdb2 including datafiles
*
ERROR at line 1:
ORA-65025: Pluggable database PAWPDB2 is not closed on
all instances.

SQL> alter pluggable database pawpdb2 close;
Pluggable database altered.

SQL> drop pluggable database pawpdb2 including datafiles;
Pluggable database dropped.

SQL> show pdbs
    CON_ID CON_NAME                       OPEN MODE  RESTRICTED
---------- ------------------------------ ---------- ----
         2 PDB$SEED                       READ ONLY  NO
         3 ORCLPDB1                       READ WRITE NO
         4 ORCLPDB                        READ ONLY  NO
         6 PAWPDB3                        MOUNTED
```

PDB's Administration Using DBCA

multisoft



What's New for Multitenants in *12.1.0.2*

- **PDB *CONTAINERS*** clause
- **PDB File Placement in OMF** using ***CREATE_FILE_DEST***
- **PDB Tablespace Logging**
  - PDB must be opened in **Restricted** mode
- **PDB Metadata Clone** using ***NO DATA*** clause
  - Source PDB must NOT contain
    - Index-Organized tables
    - Advanced Queue tables
    - Clustered tables
    - Table clusters
- **PDB Subset Cloning** Using ***USER_TABLESPACES*** clause
- **PDB State Management** across CDB restart using ***SAVE/DISCARD STATE*** clause
  - New view to check the state-***DBA_PDB_SAVED_STATES***
- **PDB Remote Cloning**
  - A non-CDB database is added to a CDB database as a PDB container usind DB links

What's New for Multitenants in 12.2.0.1

- Maximum Number of PDBs can be **4096**.
- Different character sets for PDBs.
- Proxy PDB.
- Reallocate a PDB from one CDB to another CDB.
- Memory Management at PDB level.
- Disk I/O management at PDB level.
- Local UNDO at PDB level.
- Flashback at PDB level.
- Faster creation of PDBs using parallelism .
- Prioritize PDB Upgrade.
- PDB performance profile.

MOS Note: 742060.1

Wrap-Up

- Oracle database release 12c has brought up very major changes in almost all the aspects of the database administration
- Multitenant database has been created to solve a lot of issues faced by the DBA's in the day-to-day management of large data centers
- This was just an ***introduction*** to 12c Multi-tenant databases

Good luck!!

THANK YOU FOR JOINING

multisoft

Website: www.multisoftvirtualacademy.com
Email: info@multisoftvirtualacademy.com
Contact No: +918130666206/209

multisoft
Virtual Academy

Thank You

FOR MORE DETAILS, CONTACT UNDERSIGNED

info@multisoftvirtualacademy.com

(+91) 8130666206 / 209

www.multisoftvirtualacademy.com