# Feature of our Oracle Databases - Why should we migrate to Multitenant architecture

*Radosław Cisz*
*rcisz@dataconsulting.pl*

## Multitenant New Terminology

Basic terminology connected with multitenant architecture

- Non-CDB database
- CDB database
- ROOT container
- PDB
- UnPlug database
- Plug Database

# Multitenant New Terminology

- Non CDB database
  - Traditional database not plugged into container for databases
- CDB database
  - Database that is part of container with other databases.
  - Other databases can be ROOT container (CDB$ROOT), SEED (CDB$SEED) database, other pluggable databases (PDBs)

## Multitenant New Terminology

- ROOT container
  - In other words ROOT database
  - Main database in Container
  - Keeps among others
    - Shared objects
    - Common users
    - Common roles
    - PDB specific settings
    - One common instance

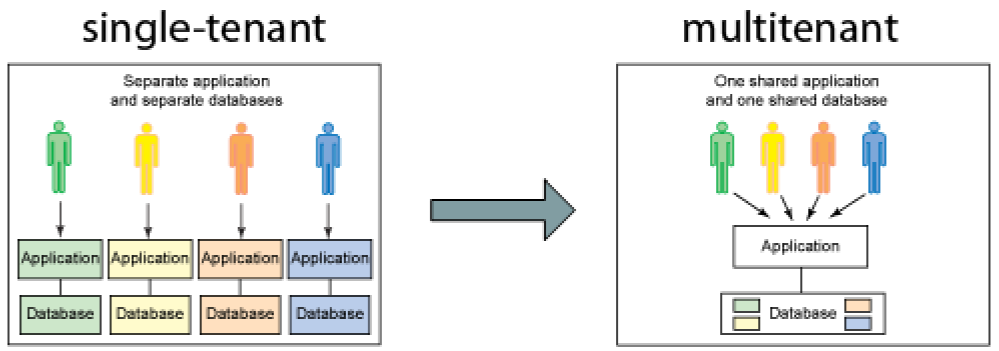
## Multitenant New Terminology

- PDB
  - Database that is part of container
  - Was created in specific container or
    - Was plugged from another container
    - Was plugged from non-CDB database

## Multitenant New Terminology

- UnPlug database
  - Process of describing database architecture in XML format,
  - Packageing it into tar format, ready to transfer and plug
- Plug database
  - Process of attaching/including database into container
    - From unplugged database
    - From backup using RMAN
    - From non-CDB database

# Description of Multitenant Architecture


single-tenant
Separate application and separate databases
Application
Database
multitenant
One shared application and one shared database
Application
Database

- Multitenant is mainly consolidation option
  - In opposite
    - to schemas for applications
    - Multiple instances on one server
- Multiple databases in centrally managed platform
- Provides isolation

## Schema based consolidation

- Cons
  - Name collision might prevent schema-based consolidation
  - Schema-based consolidation brings weak security
  - Per application backend point-in-time recovery is prohibitively difficult
  - Resource management between application backends is difficult
  - Patching the Oracle version for a single application backend is not possible
  - Cloning a single application backend is difficult. Data Pump is the only option

## Multiple instances consolidation

- Cons
  - Increased memory and CPU utilization
  - High cost of ownership

# Benefits of Multitenant DBs

- Benefits
  - Because of single instance for whole container
    - Less background processes
    - Better CPU utilization
    - Better memory utilization
  - Because of single dictionary for Oracle Supplied objects and packages
    - Better storage utilization

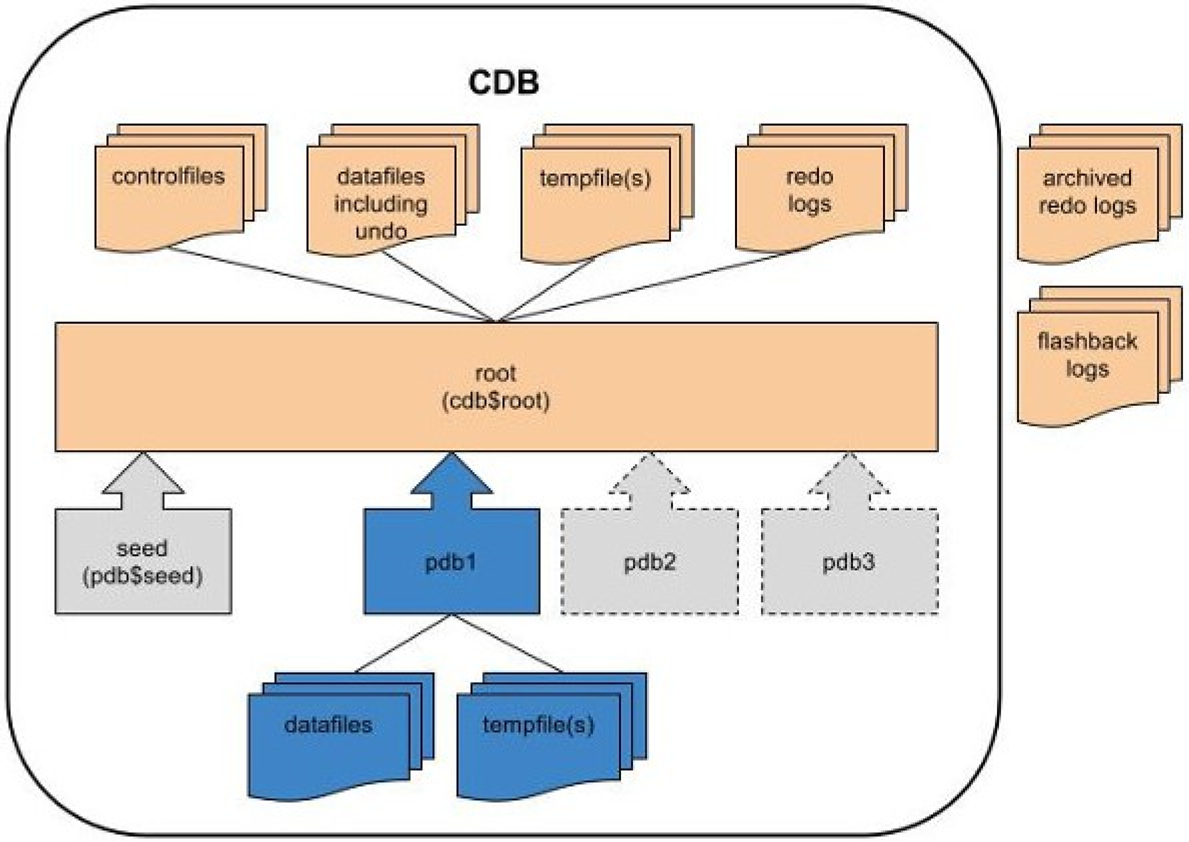
# Benefits of Multitenant DBs

- Reduced DBA cost
  - No demands for application changes
  - Unified and simplified patching and upgrade'ing
  - Separation of duties
    - DBA
    - CDBA
    - PDBA
- Backward compatibility with Non-CDB databases
- Integrated with Resource Manager (Inter PDBs plans)
- Simplified backup procedures

## Description of Multitenant Architecture

- Multitenant configuration
  - Can act as RAC
  - Can act as single instance
- Singletenant configuration
  - No fee for additional license
  - One pluggable database in single Container EE/SE2

# Description of Multitenant Architecture



- Root database (CDB$ROOT) is like management database
- Cannot open PDB without opened Root database

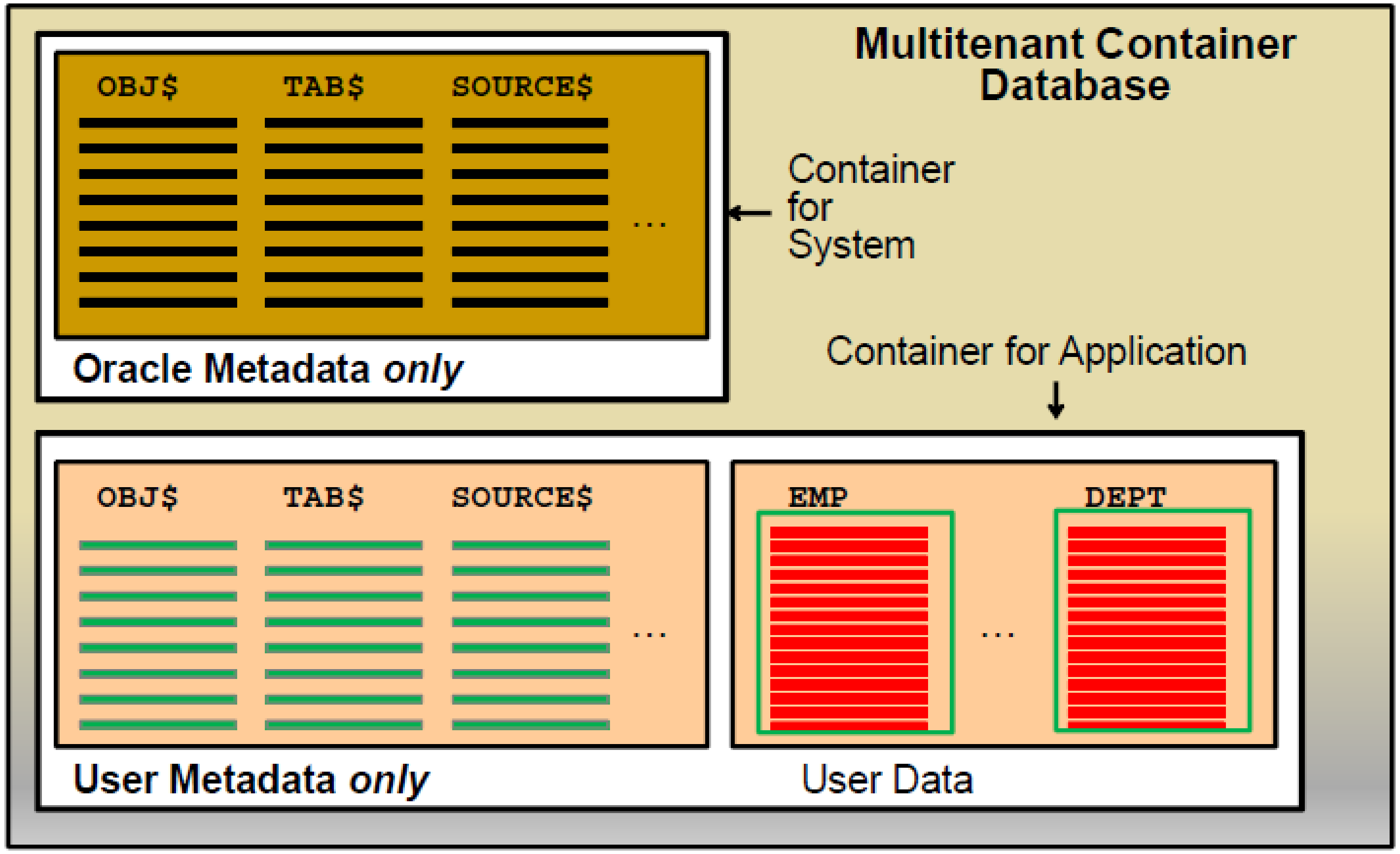
## Description of Multitenant Architecture

- Each DB Shares
  - Backgroud processes
  - Shared memory
  - Process memory
  - Oracle metadata
  - Redo Logs/Archive Logs
  - Undo Tablespace
  - Control Files
  - Optionally – Temporary Tablespace
  - Each PDB has corresponding SERVICE, by default PDB name + domain Name

# Types of containers

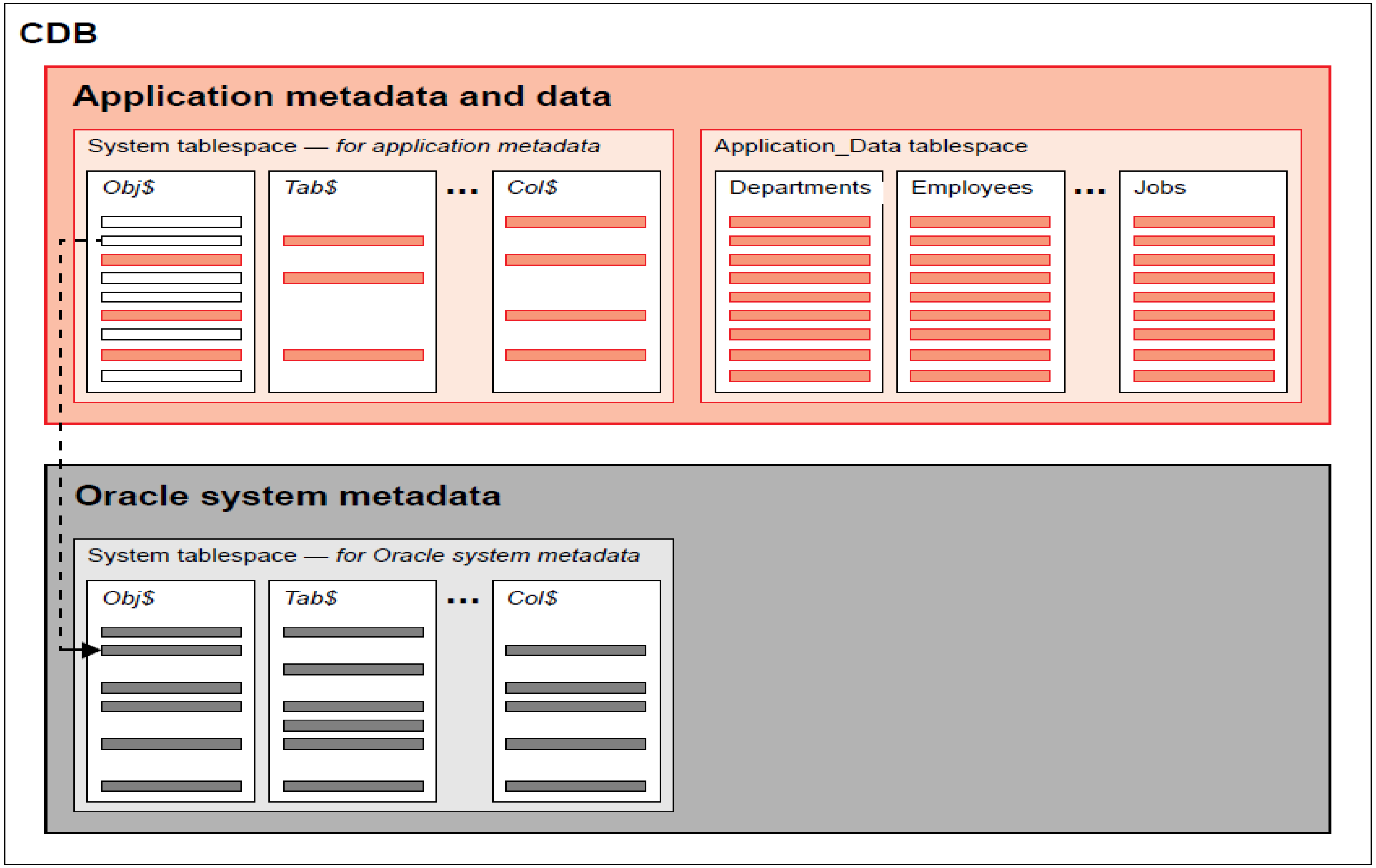
- Root Container
- Pluggable database containers
  - Application Tablespaces
  - Local Users/Roles
  - Local Objects/Schemas/Privileges
  - Local Non-Shared Application Metadata
  - PDB Resource Manager Plan
- PDB$SEED
- 252 PDBs not counting PDB$SEED
- 512 Services in CDB
- V$CONTAINERS

## Separation of System and User Data



- Pristine installation
- Non - Mixed Metadata for User and System

# Horizontal Partitioning of Data Dictionary


CDB
Application metadata and data
System tablespace — for application metadata
Obj$
Tab$
...
Col$
Application_Data tablespace
Departments
Employees
...
Jobs
Oracle system metadata
System tablespace — for Oracle system metadata
Obj$
Tab$
...
Col$

## Separation of System and User Data

- Metadata for system objects (Oracle Supplied Objects) visible in each PDB by „links”, without duplicate'ing them
- CDB$ROOT is the name for root database
- CDB$SEED is the name for always Read-Only database, used to create empty PDBs
- PDBs communicates with each other using extremely fast Intra-CDB database links

## Common vs Local Users/Privileges/Roles

- Common Users
  - CDB_USERS (COMMON) / DBA_USERS on PDB and ROOT scope
  - Are created only in CDB$ROOT replicated into each PDB
  - Can be granted COMMON (In CDB$ROOT) and LOCAL privileges/roles (In PDB)
- Local Users
  - Defined in a specific container
  - Can connect/manage , be granted only local privileges

## Common vs Local Users/Privileges/Roles

- Common Roles
  - Created only in ROOT
  - Replicated in each PDB
  - Can be granted to common users/roles
  - All Oracle supplied roles are common roles
- Local Roles
  - Created in specific container
  - Cannot contain common privileges/roles
  - Can be granted to common and local users

# Common vs Local Users/Privileges/Roles

- Common Privileges
  - Granted in ROOT
  - Can be granted to  common users/roles
  - Can refer to common and local objects
- Local Privileges
  - Granted in specific PDB
  - Can be granted to common / local users/roles
-

## Metadata linked / Object Linked Objects

- Metadata Linked Dictionary Objects
  - Store metadata about shared objects only in root
  - Each PDB has own copy of data pointing to ROOT metadata
- Object Linked Dictionary Objects
  - Metadata and data exists only in ROOT dictionary
- New column SHARING in DBA_OBJECTS

# New Multitenant Dictionary Views

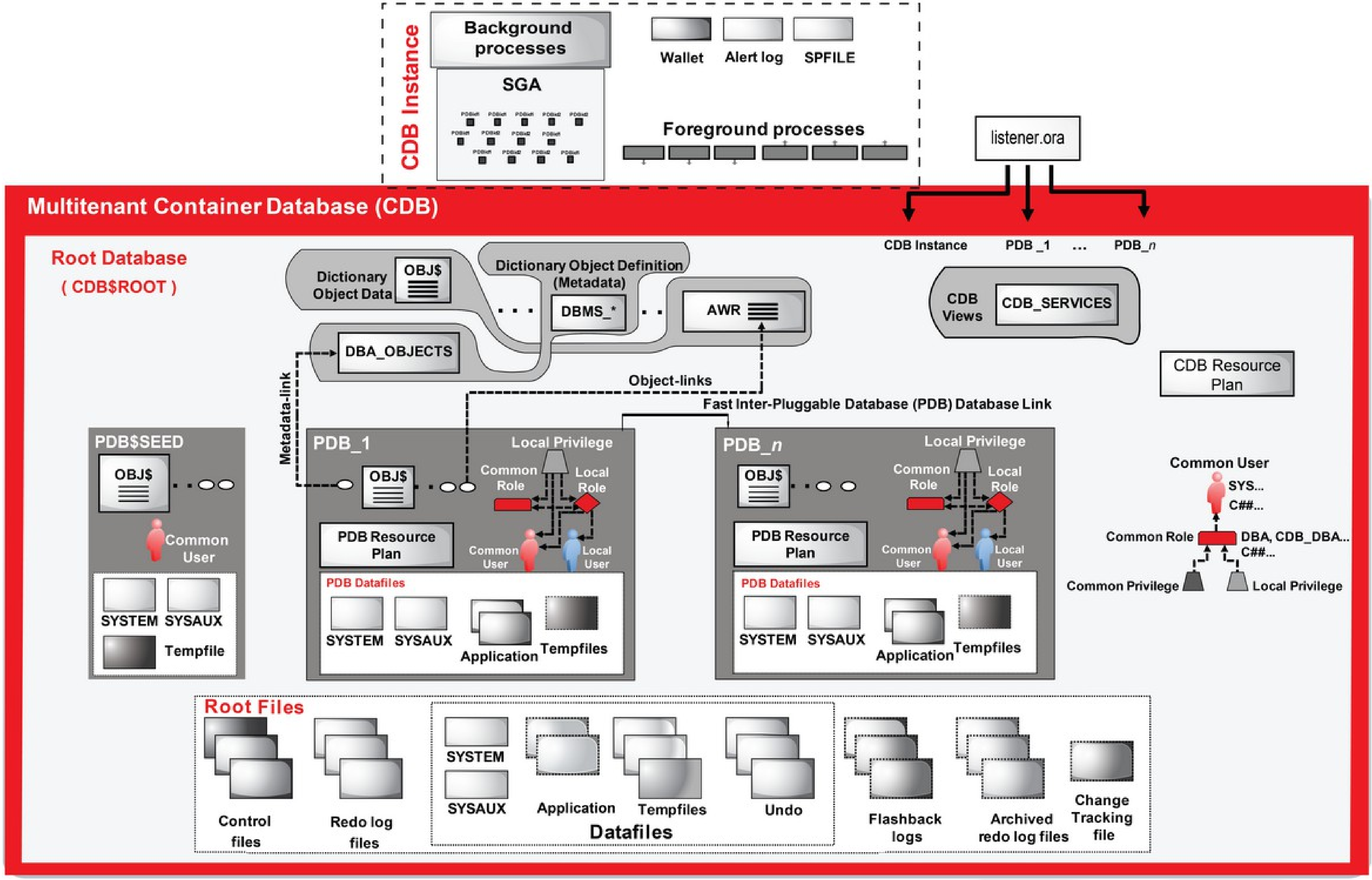
- CDB_XXX
  - DBA_XXX
    - ALL_XXX
      - USER_XXX
- DBA_XXX views maintained for backward compatibility
- CDB_XXX
  - In Common Scope (ROOT) – artifacts from all currently open PDBs and ROOT
  - New CON_ID column
  - In Local Scope

# New Multitenant Dictionary Views

- Export ORACLE_SID=CDB1
  - Select role,common,con_id from cdb_roles;
  - Select role,common,con_id from dba_roles;
  - Select name,open_mode from v$pdbs;
- Connect sys@PDB1 as sysdba
  - Select role,common,con_id from cdb_roles;
    - Displays common users and local PDB1 users
- V$_ VIEWS
  - Select distinct status,con_id from v$bh order by con_id;
  - Select object_id,oracle_username,locked_mode,con_id from v$locked_object;

## General Considerations about Multitenant

- One SPFILE
- Data Guard at CDB level
- Flashback database at CDB level
- One characterset for all containers



CDB Instance
Background processes
SGA
Wallet
Alert log
SPFILE
Foreground processes
listener.ora
Multitenant Container Database (CDB)
CDB Instance
PDB_1
...
PDB_n
Root Database
( CDB$ROOT )
Dictionary Object Data
OBJ$
Dictionary Object Definition (Metadata)
DBMS_*
AWR
DBA_OBJECTS
CDB Views
CDB_SERVICES
CDB Resource Plan
Object-links
Metadata-link
Fast Inter-Pluggable Database (PDB) Database Link
PDB$SEED
OBJ$
Common User
SYSTEM
SYSAUX
Tempfile
PDB_1
Local Privilege
Common Role
Local Role
OBJ$
PDB Resource Plan
Common User
Local User
PDB Datafiles
SYSTEM
SYSAUX
Application
Tempfiles
PDB_n
Local Privilege
Common Role
Local Role
OBJ$
PDB Resource Plan
Common User
Local User
PDB Datafiles
SYSTEM
SYSAUX
Application
Tempfiles
Common User
SYS...
C##...
Common Role
DBA, CDB_DBA...
C##...
Common Privilege
Local Privilege
Root Files
Control files
Redo log files
SYSTEM
SYSAUX
Application
Tempfiles
Undo
Datafiles
Flashback logs
Archived redo log files
Change Tracking file

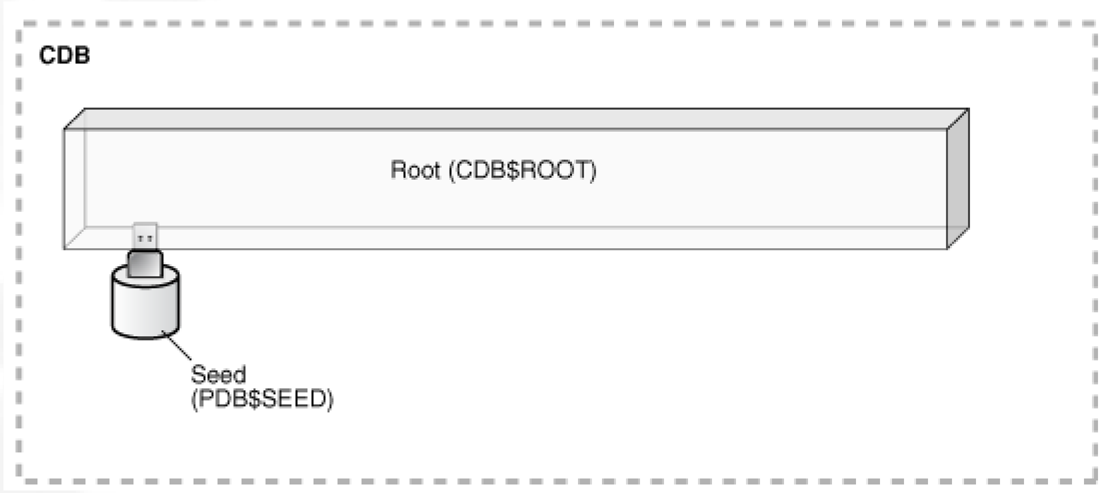
## Creation of Container Databases

- Supported Tools
  - OUI
  - DBCA
  - SQL Developer
  - EM CC
- New parameter and statement
  - enable_pluggable_database
  - CREATE DATABASE .. ENABLE PLUGGABLE DATABASE

# Creation of Container Databases

- Export ORACLE_SID
- Set enable_pluggable_database parameter to true
- Startup nomount
- CREATE DATABASE CDB ENABLE PLUGGABLE DATABASE SEED FILE_NAME_CONVERT('/u01/oradata','/u01/oradata/seed');
- Alter session set "_oracle_scripts"=true;
- Alter pluggable database pdb$seed close;
- Alter pluggable database pdb$seed open;
- @catalog.sql
- @catblock.sql
- @catproc.sql
- @catoctk.sql
- @owminst.sql
- @pupbld.sql

# Creation of Container Databases



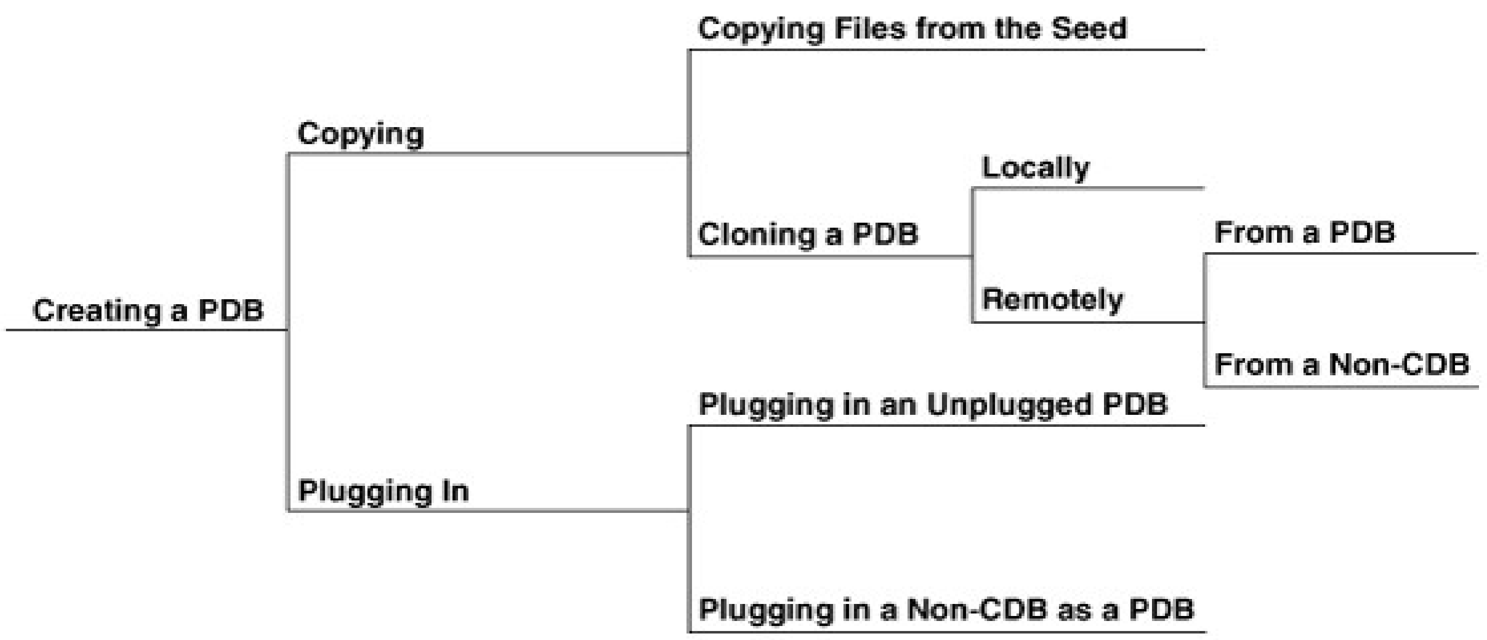
- SELECT NAME, CDB, CON_ID FROM V$DATABASE;

```
NAME     CDB    CON_ID
CDB1     YES         0
```

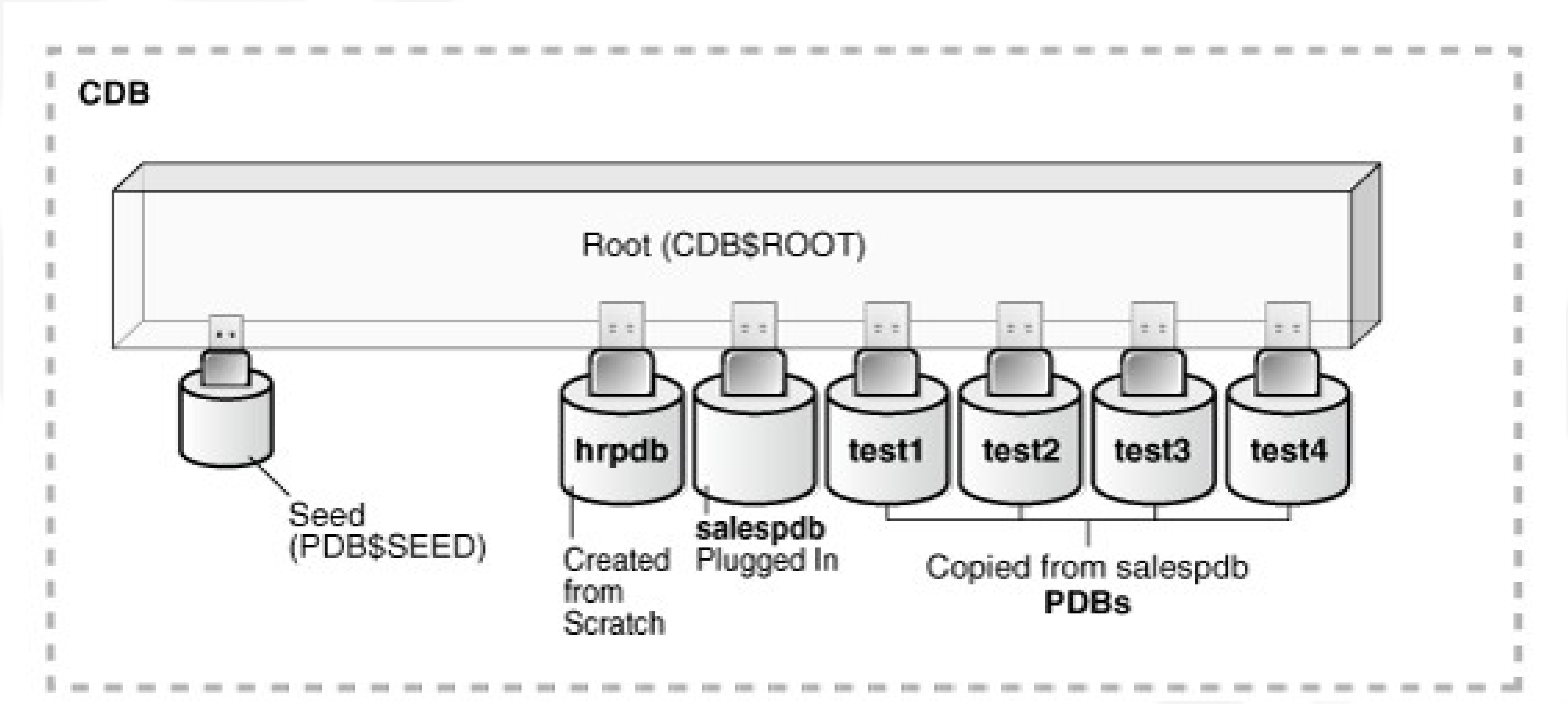
- File location parameters
  - Using OMF – DB_CREATE_FILE_DEST
  - Without OMF - PDB_FILE_NAME_CONVERT

## Creation of PDB

- CREATE PLUGGABLE DATABASE ...

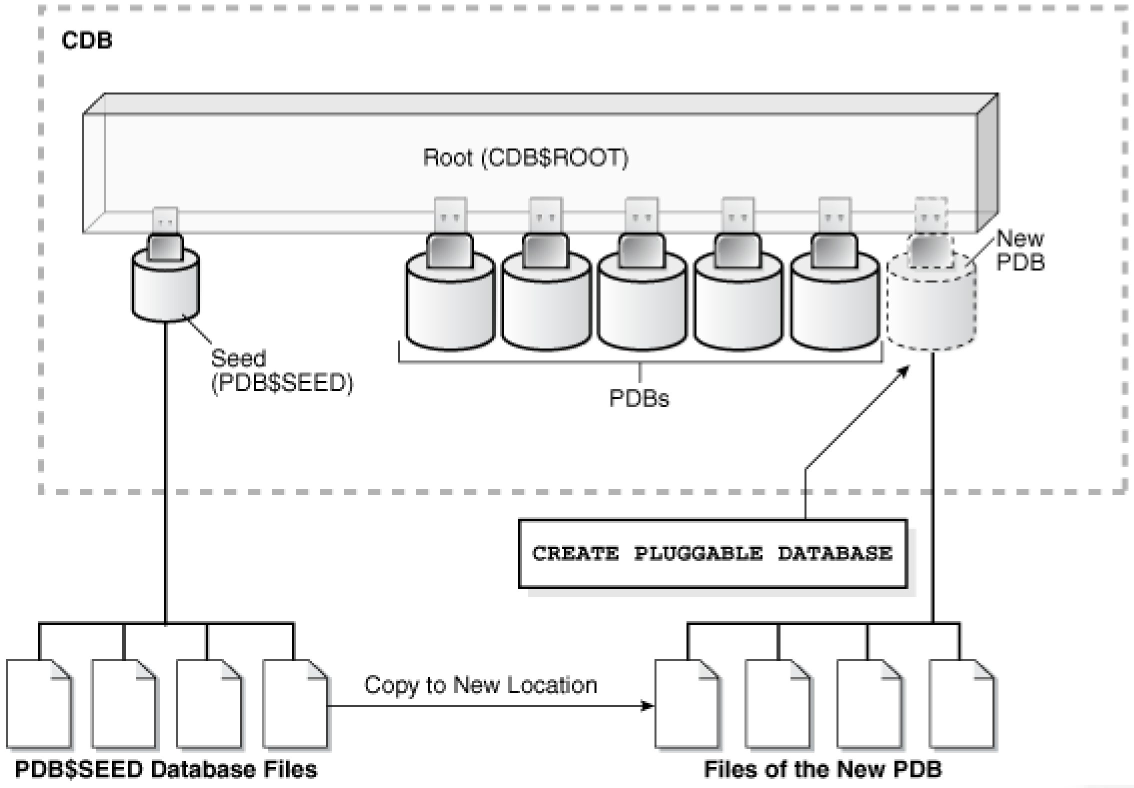

Creating a PDB
Copying
Copying Files from the Seed
Cloning a PDB
Locally
Remotely
From a PDB
From a Non-CDB
Plugging In
Plugging in an Unplugged PDB
Plugging in a Non-CDB as a PDB

# Creation of PDB



# Create from SEED

- CREATE PLUGGABLE DATABASE pdb1 ADMIN USER pdba1 IDENTIFIED BY oracle;



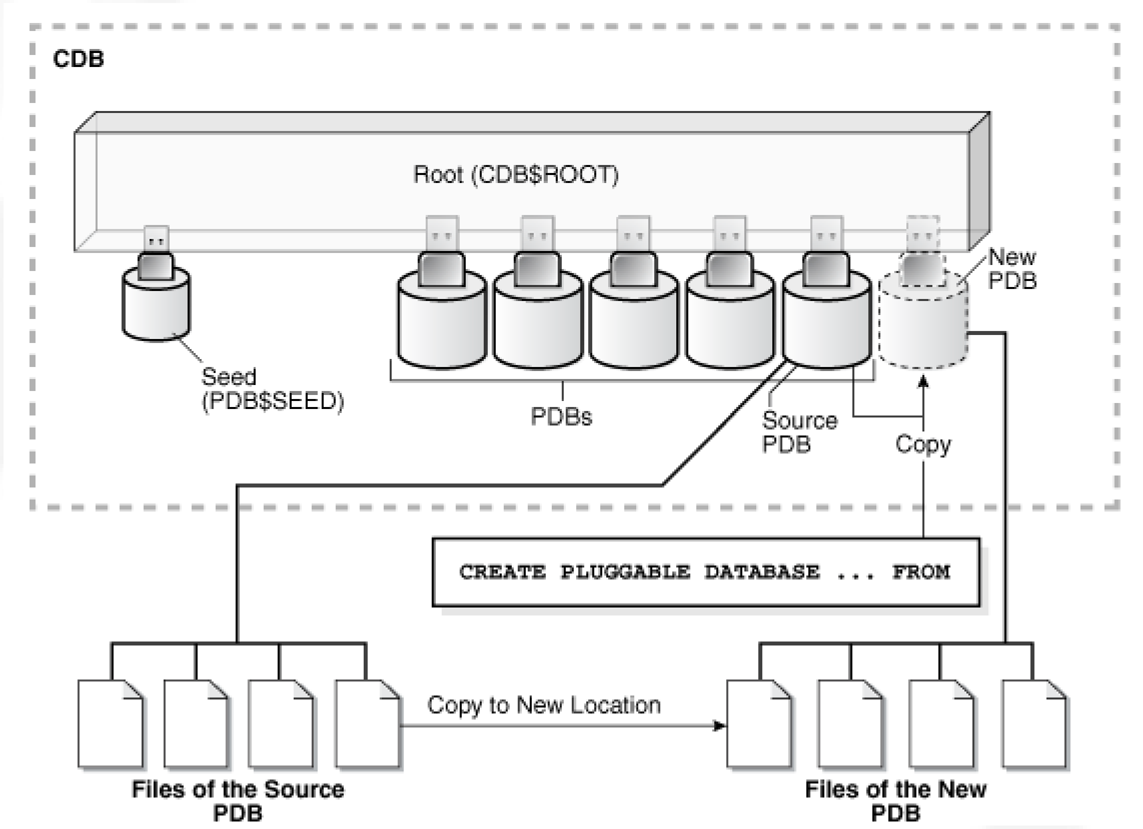
## Create from SEED

- Copied datafiles of user tablespaces
- Created SYSTEM,SYSAUX tablespaces
- Creates full catalog including metadata pointing to Oracle-Supplied objects
- Creates common users
  - SYSTEM, SYS
- Creates a local dba user PDBA granted PDB_DBA role
- Creates default service

## Creation of a PDB by Cloning a PDB or a Non-CDB

- CREATE PLUGGABLE DATABASE salespdb FROM hrpdb
    - You can use the CREATE PLUGGABLE DATABASE statement to clone a source PDB or non-CDB and plug the clone into the CDB
    - Source can be a PDB in a local or remote CDB, or starting in Oracle Database 12c Release 1 (12.1.0.2), it can also be a remote non-CDB
    - Great benefit of storage savings when filesystem support Snapshots copies
-

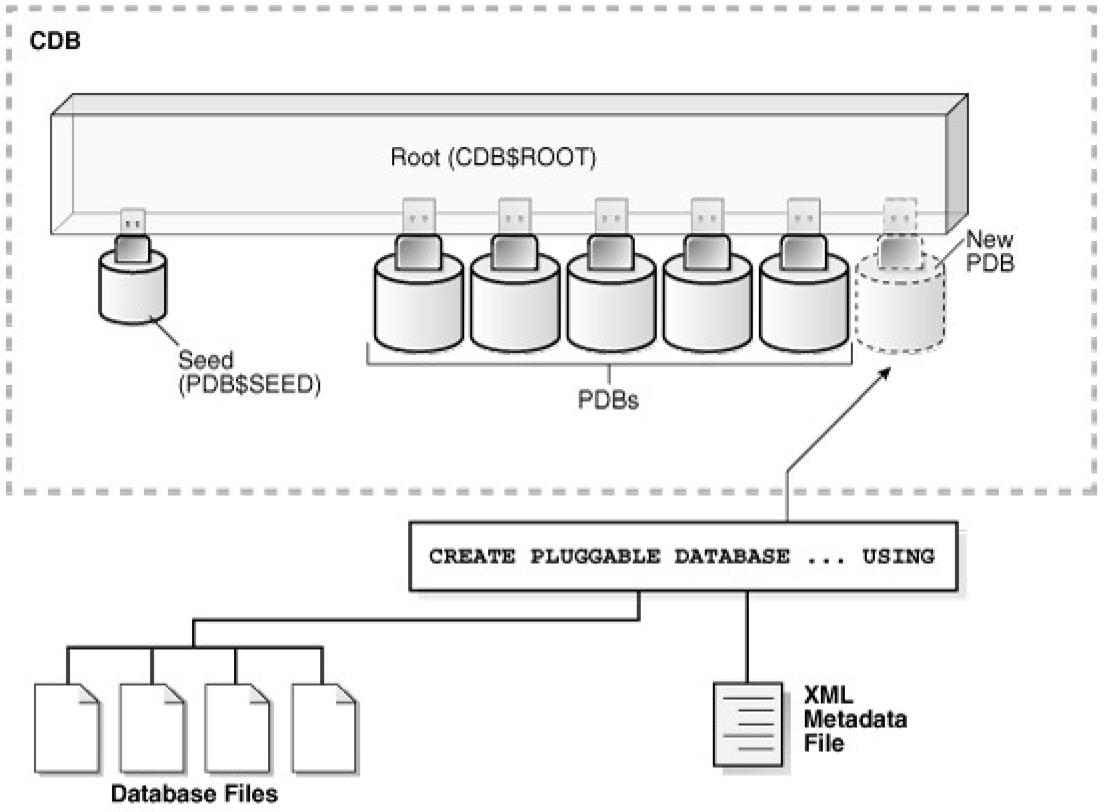
# Creation of a PDB by Cloning a PDB or a Non-CDB



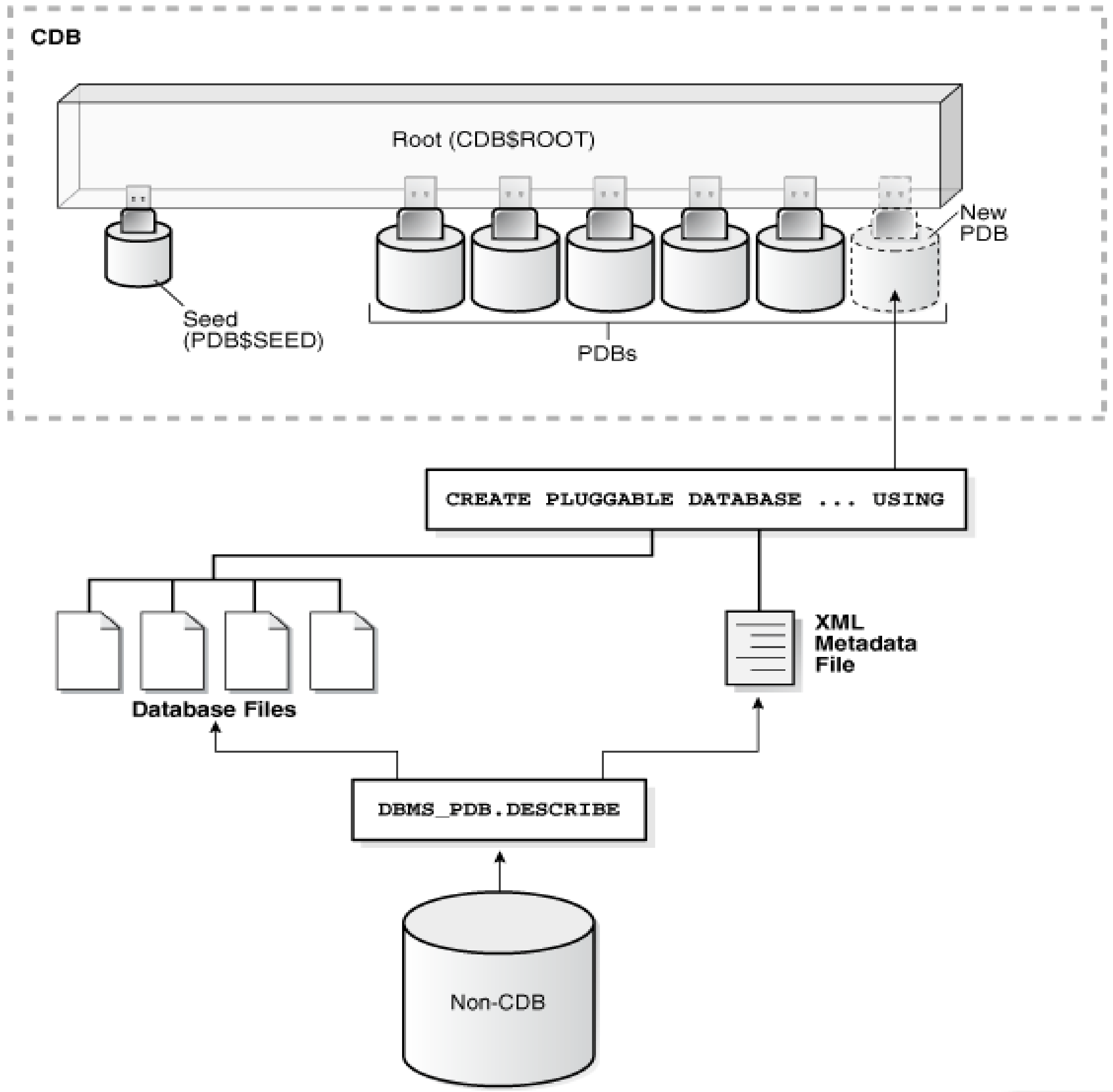
## Creation of a PDB by Plugging In an Unplugged PDB

- CREATE PLUGGABLE DATABASE pdb2 USING '/u01/oradata/pdb1.xml' NOCOPY;
  - In its unplugged state, a PDB is a self-contained set of data files and an XML metadata file
  - XML metadata file describes PDBs
  - Use NOCOPY option if target datafiles are in desired location
  - Default is COPY

# Creation of a PDB by Plugging In an Unplugged PDB



# Creation of a PDB from a Non-CDB



# Other options

- Create empty PDB from SEED and USE
  - DataPump (transportable tablespaces or full exp/imp)
  - Golden Gate to replicate database

## Unplug/plug across different endiannesses

- When a customer-created tablespace is converted from one endianness to the other, using the RMAN convert command, it is only block headers that are changed (because these are encoded in an endianness-sensitive way)
- unplugged PDB contains data dictionary tables, and some of the columns in these encode information in an endianness-sensitive way

- Conclusions:
  - There is no supported way to handle the conversion of such columns automatically. This means, quite simply, that an unplugged PDB cannot be moved across an endianness difference.

## Unplug/plug across different operating systems, chipsets

  - PL/SQL native compilation might have been used
  - Particular data dictionary table will hold machine code
  - That is sensitive to the chipset of the platform where it was compiled
- Conclusions
  - All natively compiled PL/SQL units in the incoming PDB must be recompiled before the PDB can be made available for general use

# Unplug/plug for patching the Oracle Patch version

  - Soft Patch - changes only binaries
  - Hard Patches - changes bianries and data dictionary (body not specification)
- Conclusions
  - Since data dictionary is in ROOT PDB does not need to be changed

## Working with CDBs and PDBs

- Connection to PDBs
  - By default there is still one listener process
  - Each PDB has registered default unique accross CDB service
    - SQL> Show CON_NAME
    - SQL> Select name,pdb from cdb_services;
  - Use DBMS_SERVICE.CREATE_SERVICE in non-Oracle Clusterware/Oracle Restart configuration
  - srvctl add service -d cdb1 -service pdb1_srv -pdb pdb1

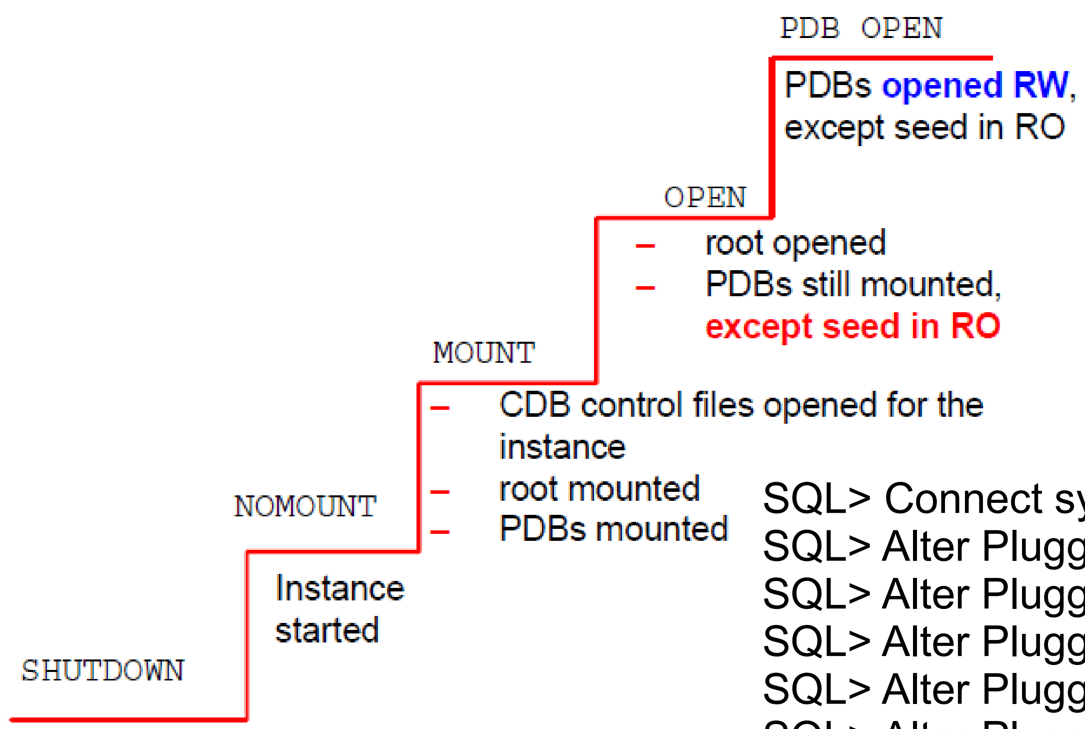
## Working with CDBs and PDBs

- User with SET CONTAINER privilege
  - Alter session set container=pdbapp1;
    - Do not fire AFLTER LOGON TRIGGERS
  - Show con_name;
  - Alter session set container=CDB$ROOT;

## Working with CDBs and PDBs

- Command with a scope of pluggable database
  - SQL> Alter System Flush Shared_Pool;
  - SQL> Alter System Flush Buffer_Cache;
  - SQL> Alter System Enable Restricted Session;
  - SQL> Alter System Kill Session ..
- Command that affect whole CDB
  - SQL> Alter System Checkpoint;
  - SQL> Alter System Switch Logfile;

# Starting CDB


PDB OPEN
PDBs opened RW, except seed in RO
OPEN
– root opened
– PDBs still mounted, except seed in RO
MOUNT
– CDB control files opened for the instance
– root mounted
– PDBs mounted
NOMOUNT
Instance started
SHUTDOWN

```
SQL> Connect sys@CDB1 as sysdba
SQL> Alter Pluggable Database PDB1 OPEN;
SQL> Alter Pluggable Database ALL OPEN;
SQL> Alter Pluggable Database ALL EXCEPT PDB1 OPEN;
SQL> Alter Pluggable Database PDB1 CLOSE IMMEDIATE;
SQL> Alter Pluggable Database ALL EXCEPT PDB1 CLOSE;
SQL> Alter Pluggable Database ALL CLOSE;
SQL> Connect sys@pdb1 as sysdba
        SQL> Alter Pluggable Database Close;
        or
        SQL> Shutdown Immediate
        SQL> Alter Pluggable Database Open Restricted;
```

# Modify PDB Settings

- connect sys@PDB1 as sysdba
  - SQL> alter pluggable database datafile '/u01/oradata/pdb1/user01.dbf' ONLINE;
  - SQL> Alter pluggable database default temporary tablespace TEMP1;
  - SQL> Alter pluggable database STORAGE (MAXSIZE 15G);

## Instance parameters for PDB

- One common spfile;
- New column ISPDB_MODIFYABLE in v$parameter
- Persistent between restarts
- Stored in CDB$ROOT dictionary
  - Alter System set job_queue=4 ;
  - Select DB_UNIQ_NAME,PDB_UID,NAME,VALUE$ from pdb_spfile$;

## Users / Privileges / Roles Security

- Common User
  - SQL> Create User c##user1 identified by oracle container=ALL;
  - SQL> Grant create session to c##user1 container=ALL;
- Local User
  - SQL> Create User user2 identified by oracle container=CURRENT;
  - SQL> Grant create table to user2 (container=current);

## Users / Privileges / Roles Security

- Common role
  - SQL> CREATE ROLE c##r1 CONTAINER=ALL;
    - Can be granted common and local privileges
    - Can be granted to local and common users
- Local role
  - SQL> CREATE ROLE lr1 CONTAINER=CURRENT;
    - Can be granted only local privileges
    - Can be granted to common and local users in current container

## Differences in Backup and Recovery

- Export ORACLE_SID=CDB1
- rman target=/
- RMAN> BACKUP DATABASE;
  - Above sequence of command backups all datafiles of ROOT and PDB containers
- RMAN> BACKUP PLUGGABLE DATABASE PDB1,PDB2;
  - Backups datafiles of PDB1,PDB2 containers

## Differences in Backup and Recovery

- RMAN> BACKUP PLUGGABLE DATABASE „CDB$ROOT”;
  - Backups only  ROOT container datafiles
- RMAN> BACKUP TABLESPACE SYSTEM;
  - Backups system tablespace of ROOT container
- RMAN> BACKUP TABLESPACE pdb1:SYSAUX;
  - Backups SYSAUX tablespace of PDB1

## Differences in Backup and Recovery

- Recovery
  - Instance recovery pertains to whole CDB
  - Flashback of database pertains to whole CDB
    - All Containers are flashed-back
    - If on one of PDBs Point In Time Recovery was perfomed, it is impossible to flashback to point earlier then PDBPITR was done
  - During restart of CDB tempfiles are recreated
  - Restart of PDB does not implicate local temporary tablespace recreation

## Differences in Backup and Recovery

- Recovery contd
  - Lost CONTROLFILE/ROOT SYSTEM tablespace implicates downtime for all PDBs, for the time of controlfile restoration
    - At CDB Level
      - STARTUP NOMOUNT
      - RESTORE CONTROLFILE FROM ..
      - ALTER DATABASE MOUNT
      - RECOVER DATABASE
      - ALTER DATABASE OPEN RESETLOGS
      - ALTER PLUGGABLE DATABASE ALL OPEN

# Differences in Backup and Recovery

- Recovery contd
  - Media failure of single PDB (SYSTEM TABLESPACE of PDB), when PDB is not closed, can affect whole CDB
  - When PDB is open, on CDB level
    - SHUTDOWN IMMEDIATE
    - STARTUP MOUNT
    - RESTORE PLUGGABLE DATABASE PDB1
    - RECOVER PLUGGABLE DATABASE PDB1
    - ALTER DATABASE OPEN
    - ALTER PLUGGABLE DATABASE PDB1 OPEN

## Differences in Backup and Recovery

- Features like DATA RECOVERY ADVISOR works for CDB$ROOT, and all PDBs
- Block Corruption Validation
  - VALIDATE DATABASE
    - All containers affected
  - VALIDATE DATABASE ROOT
  - VALIDATE PLUGGABLE DATABASE PDB1,PDB2

## Differences in Backup and Recovery

- Database Duplication
  - Create Auxiliary instance with enable_pluggable_database set to TRUE
  - CDB$ROOT and CDB$SEED are created during duplication
    - RMAN> DUPLICATE DATABASE TO CDB1 PLUGGABLE DATABASE PDB1,PDB2

## AWR reports

- You can create AWR Reports for the CDB as a whole
- You can also create AWR Reports for a particular PDB

## Upgrade in Multitenant configuration

- Everything at Once
- One at a Time
  - via unplug/plug
- Everything at Once
  - Using DBUA against CDB will upgrade CDB$ROOT and all PDBs in container
  - Pros
    - One step operation
  - Cons
    - All pdbs are inaccessible for users

## Upgrade in Multitenant configuration

- One at a Time
  - Connect to PDB being upgraded
    - connect sys@PDB1
  - Execute preupgrade
    - @/u01/app/oracle/product/12.1.0.2/omh1/rdbms/admin/preupgrd.sql
  - If any errors met, it generates preupgrade_fixups.sql
    - /u01/app/oracle/cfgtoollogs/cdb1/preupgrade/preupgrade_fixups.sql
  - Execute fixups
    - @/u01/app/oracle/cfgtoollogs/cdb1/preupgrade/preupgrade_fixups.sql
  - Gather dictionary stats
    - EXECUTE dbms_stats.gather_dictionary_stats;

## Upgrade in Multitenant configuration

- One at a Time contd
  - Unplug PDB1
    - connect sys@cdb1
    - alter pluggable database pdb1 close;
    - alter pluggable database pdb1 unplug into '/u01/app/oracle/manifest/pdb1.xml'
    - drop pluggable database pdb1 keep datafiles;
  - Connect to CDB in newer version (CDB2), and check compatibility to plug in
    - connect sys@cdb2
    - Select DBMS_PDB.CHECK_PLUG_COMPATIBILITY(pdb_descr_file=>'/u01/app/oracle/manifest/pdb1.xml'mpdb_name=>'PDB1') from dual;
    - select message, status from pdb_plug_in_violations;
  -

## Upgrade in Multitenant configuration

- One at a Time contd
  - Plug pdb to new container CDB2
    - create pluggable database PDB1 using '/u01/app/oracle/manifest/pdb1.xml' file_name_convert('/u01/oradata/cdb1/pdb1','/u01/oradata/cdb2/pdb1);
    - alter pluggable database pdb1 open upgrade;
  - Run catupgrd.sql script
    - cd $ORACLE_HOME/rdbms/admin
    - $ORACLE_HOME/perl/bin/perl catctl.pl -c "pdb1" -l /tmp catupgrd.sql
  - PDB will be closed, startup it and recompile, run postupgrade_fixups
    - @?/rdbms/admin/utlrp.sql
    - @/u01/app/oracle/cfgtoollogs/CDB1/preupgrade/postupgrade_fixups.sql
    - EXECUTE DBMS_STATS.gather_fixed_objects_stats;

# Running Scripts Against Container and Pluggable databases

- Using SET CONTAINER

```
cat run_on_pdb.sh
sqlplus / as sysdba <<EOF
ALTER SESSION SET CONTAINER = $1;
-- Perform smoe actions as before...
SHOW CON_NAME;
EXIT;
EOF
./run_on_pdb.sh PDB1
```

# Running Scripts Against Container and Pluggable databases

- Using TWO_TASK variable
  - ! Does not work with connection as SYSDBA !

```
export TWO_TASK=PDB1
sqlplus l_user1/pass
```

- Oracle Scheduler
  - New types of JOBs
    - SQL_SCRIPT
    - BACKUP_SCRIPT

# Running Scripts Against Container and Pluggable databases

- Secure External Password Store

Configure Oracel Net in sqlnet.ora

```
WALLET_LOCATION =
   (SOURCE =
     (METHOD = FILE)
     (METHOD_DATA =
       (DIRECTORY = /u01/app/oracle/wallet)
     )
   )

SQLNET.WALLET_OVERRIDE = TRUE
SSL_CLIENT_AUTHENTICATION = FALSE
SSL_VERSION = 0
```

## Running Scripts Against Container and Pluggable databases

- Secure External Password Store contd

Create Wallet

```
mkdir -p /u01/app/oracle/wallet
orapki wallet create -wallet "/u01/app/oracle/wallet" -pwd "wallet_password" -auto_login_local
```

Store PDB Credentials in Wallet

```
mkstore -wrl "/u01/app/oracle/wallet" -createCredential PDB1_ALIAS l_user1 user_password
```

Connect using Wallet

```
sqlplus /@PDB1_ALIAS
```

# Running scripts on multiple PDB/CDB in correct order

- Some Oracle supplied scripts must be applied in the correct order, starting with the CDB$ROOT container.
- catcon.pl
  - container specific logs

```
perl catcon.pl -d $ORACLE_HOME/rdbms/admin -b shrink_temp_log -c
shrink_temp.sql

$ ls shrink_temp_log*

shrink_temp_log0.log shrink_temp_log1.log shrink_temp_log2.log shrink_temp_log3.log

      - shrink_temp_log0.log - output from CDB$ROOT and PDB$SEED
      - shrink_temp_log3.log - general messages from task
      - shrink_temp_log1.log, shrink_temp_log2.log - output messages from PDBs
```

# Running scripts on multiple PDB/CDB in correct order

On all CDB/PDBs

```
perl catcon.pl -e -b sql_output -- --x"SELECT SYS_CONTEXT('USERENV', 'CON_NAME') FROM dual"
```

On ALL databases except CDB$ROOT and PDB$SEED

```
perl catcon.pl -e -C 'CDB$ROOT PDB$SEED' -b sql_output -- --x"SELECT SYS_CONTEXT('USERENV', 'CON_NAME') FROM dual"
```

On specifed databases

```
perl catcon.pl -e -c 'PDB1 PDB2' -b sql_output -- --x"SELECT SYS_CONTEXT('USERENV', 'CON_NAME') FROM dual"
```

## Running scripts on multiple PDB/CDB in correct order

```
$ORACLE_HOME/perl/bin/perl $ORACLE_HOME/rdbms/admin/catcon.pl
[-u username[/password]] [-U username[/password]] [-d directory]
[-l directory] [{-c|-C} container] [-p parallelism] [-e] [-s]
[-E { ON | errorlogging-table-other-than-SPERRORLOG } ] [-I] [-g] [-f]
-b log_file_name_base -- { SQL_script [arguments] | --x'SQL_statement' }
```

## Follow-up Courses

- In order of importance:
  - Oracle Database 12c: Managing Multitenant Architecture (D79128GC10)
  - Oracle Database 12c: New Features for Administrators Ed 2 (D77758GC20)
  - Oracle Database 12c: High Availability New Features (D79794GC10)
  - Oracle Database 12c: Administration Workshop Ed 2 (D78846GC20)
  - Oracle Database 12c: Backup and Recovery Workshop Ed 2 (D78850GC20)
  - Oracle Database 12c: Admin, Install and Upgrade Accelerated (D79027GC10)
  - Oracle Database 12c: Data Guard Administration (D79232GC10)