

TECHFEST

1-4 DECEMBER 2019 // THE GRAND BRIGHTON // UK

#UKOUG
#TECHFEST2019
#PASSTHEKNOWLEDGE

Baku, Azerbaijan



Oracle Multitenant 19c - All About Pluggable Databases

Mahir M. Quluzade
Central Bank of Azerbaijan



- Oracle ACE 
- Board Member of Azerbaijan Oracle User Group (@AzerbaijanOUG)
- Board Member of International Data Guard Oracle User Group (@idgoug)
- Oracle Database Architect at Central Bank of Azerbaijan
- Blogger since 2011 : <http://www.mahir-quluzade.com>

Azerbaijan Oracle User Group

@AzerbaijanOUG

- **Azerbaijan Oracle Users Group**
- Founded **June 2012**
- Organized many international conferences, also local events

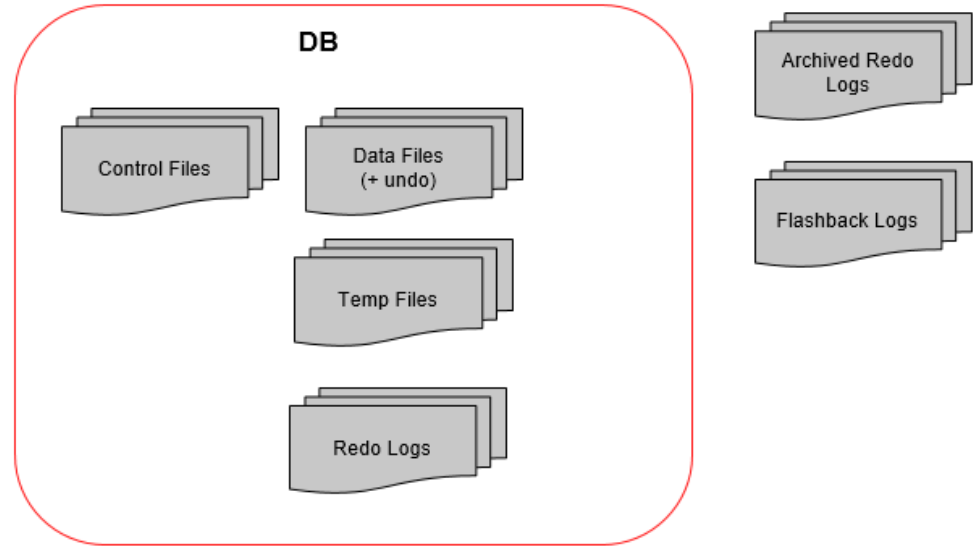


Traditional Oracle Database Architecture

Oracle Database Architecture

Pre-12c Oracle Database

- Oracle Instance
 - SGA & PGA
 - Background Processes
- Oracle Database
 - Control Files
 - Data Files
 - Redo Logs (Online and Archived)
 - Parameter and Password file



Container Databases

```
SQL> CREATE DATABASE ... ENABLE PLUGGABLE DATABASE
```

Create as Container Database

```
CREATE DATABASE cdb1
```

```
DATAFILE '/u01/app/oracle/oradata/cdb1/system01.dbf' SIZE  
700M REUSE AUTOEXTEND ON NEXT 10240K MAXSIZE UNLIMITED  
EXTENT MANAGEMENT LOCAL
```

```
SYSAUX DATAFILE
```

```
'/u01/app/oracle/oradata/cdb1/sysaux01.dbf' SIZE 550M  
REUSE AUTOEXTEND ON NEXT 10240K MAXSIZE UNLIMITED
```

```
SMALLFILE DEFAULT TEMPORARY TABLESPACE TEMP TEMPFILE  
'/u01/app/oracle/oradata/cdb1/temp01.dbf' SIZE 20M REUSE  
AUTOEXTEND ON NEXT 640K MAXSIZE UNLIMITED
```

```
SMALLFILE UNDO TABLESPACE "UNDOTBS1" DATAFILE  
'/u01/app/oracle/oradata/cdb1/undotbs01.dbf' SIZE 200M  
REUSE AUTOEXTEND ON NEXT 5120K MAXSIZE UNLIMITED
```

```
CHARACTER SET AL32UTF8  
NATIONAL CHARACTER SET AL16UTF16  
LOGFILE GROUP 1  
('/u01/app/oracle/oradata/cdb1/redo01.log') SIZE 50M,  
GROUP 2 ('/u01/app/oracle/oradata/cdb1/redo02.log') SIZE  
50M,  
GROUP 3 ('/u01/app/oracle/oradata/cdb1/redo03.log') SIZE  
50M
```

```
USER SYS IDENTIFIED BY "&&sysPassword" USER SYSTEM  
IDENTIFIED BY "&&systemPassword"
```

```
ENABLE PLUGGABLE DATABASE ...;
```

Database Configuration Assistant - Create a database - Step 4 of 14

Specify Database Identification Details

19c ORACLE Database

Database Operation
Creation Mode
Deployment Type
Database Identification
Storage Option
Fast Recovery Option
Network Configuration
Configuration Options
Management Options
User Credentials
Creation Option
Summary
Progress Page
Finish

Provide a unique database identifier information. An Oracle database is uniquely identified by a Global database name, typically of the form "name.domain".

Global database name:

SID:

Service name:

Create as Container database

A Container database can be used for consolidating multiple databases into a single database, and it enables database virtualization. A Container database (CDB) can have zero or more pluggable databases (PDB).

Use Local Undo tablespace for PDBs

Create an empty Container database

Create a Container database with one or more PDBs

Number of PDBs:

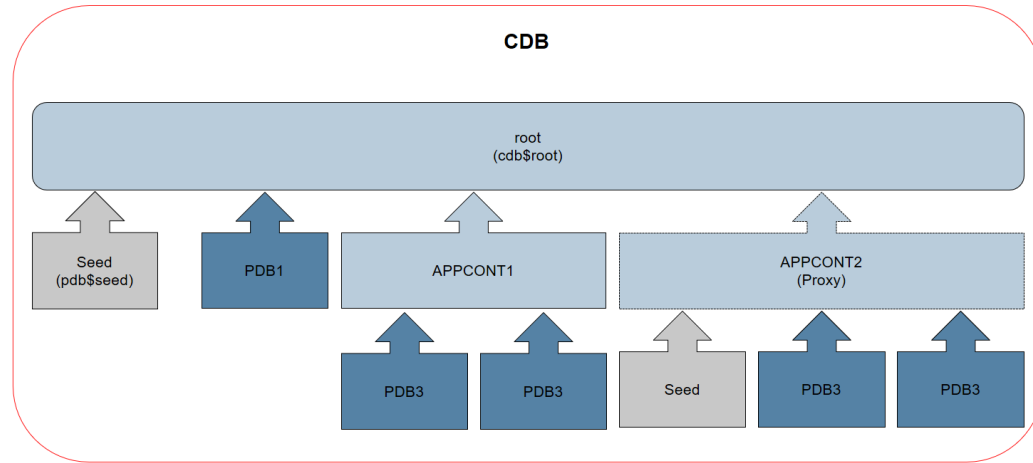
PDB name:

Help < Back Next > Finish Cancel

Oracle Database Architecture

Multitenant Architecture – 19c

- **Oracle Multitenant Option**
 - Root Container (CDB\$ROOT)
 - Seed PDB (PDB\$SEED)
 - Application Root
 - Application Seed
 - Application PDBS
 - Pluggable Database (PDBs)

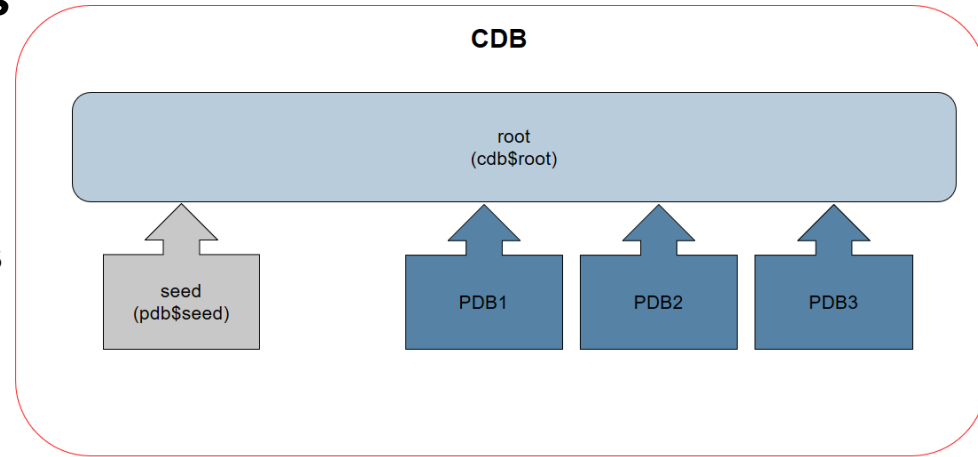


Oracle Database Architecture

Multitenant Architecture – 19c

▪ ***CDB with No Application Containers***

- Each PDB has its own dedicated application
- A different PDB administrator manages each PDB
- At the physical level, this CDB has a database instance and database files, just as a non-CDB does

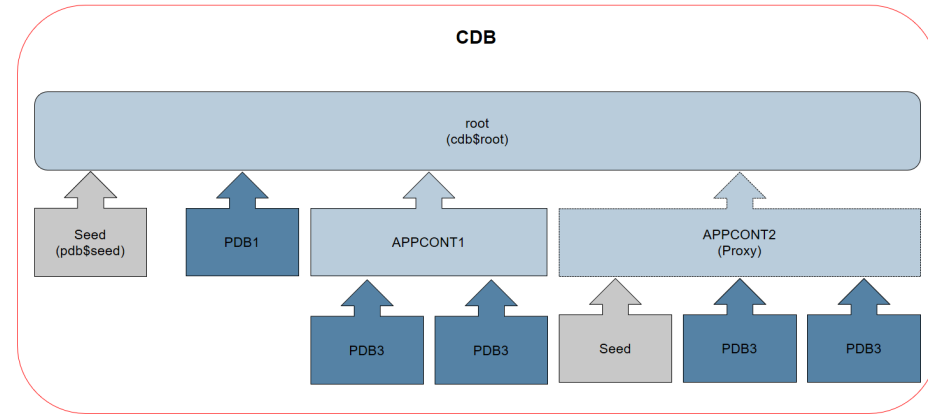


Oracle Database Architecture

Multitenant Architecture – 19c

▪ *CDB with an Application Container*

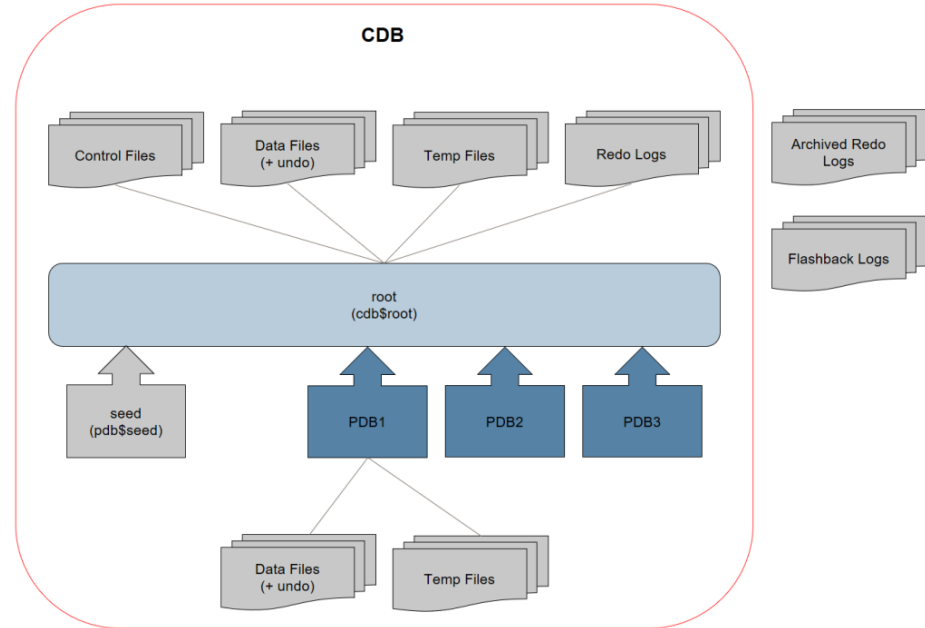
- A CDB administrator manages the CDB itself.
- An application container administrator manages the **application** container, including application installation and upgrades.
- A PDB administrator manages **hrpdb**.



Container Databases

CDB Root - CDB\$ROOT

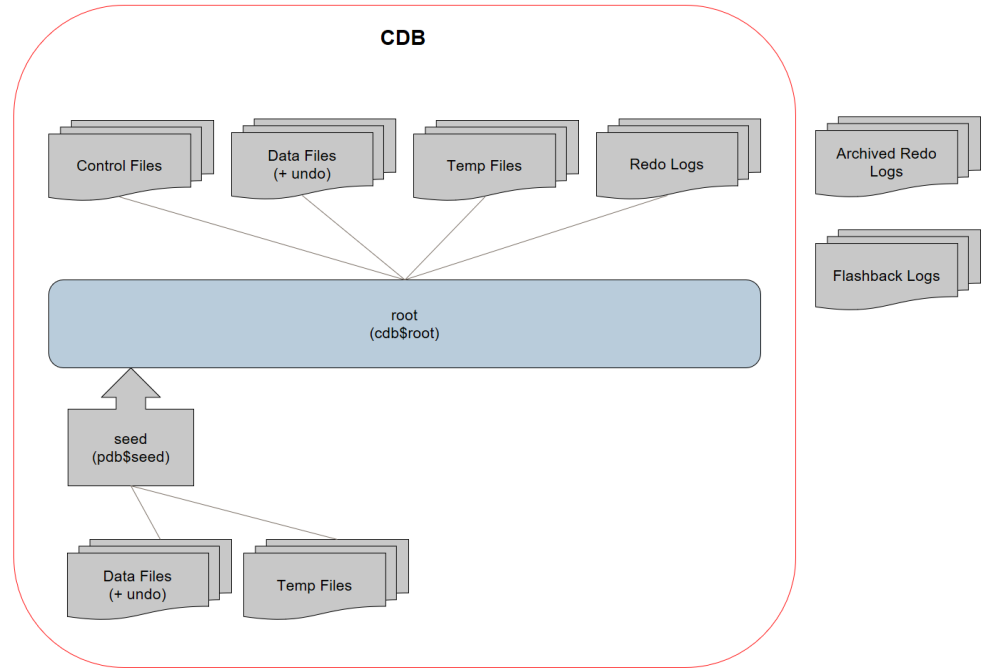
- System metadata required to manage PDBs
- Does not store user data
- Only for common users
- Automatically created tablespaces
 - SYSTEM, SYSAUX, TEMP, UNDO
- Oracle recommends AL32UTF8 for the root character set



Container Databases

Seed PDB - PDB\$SEED

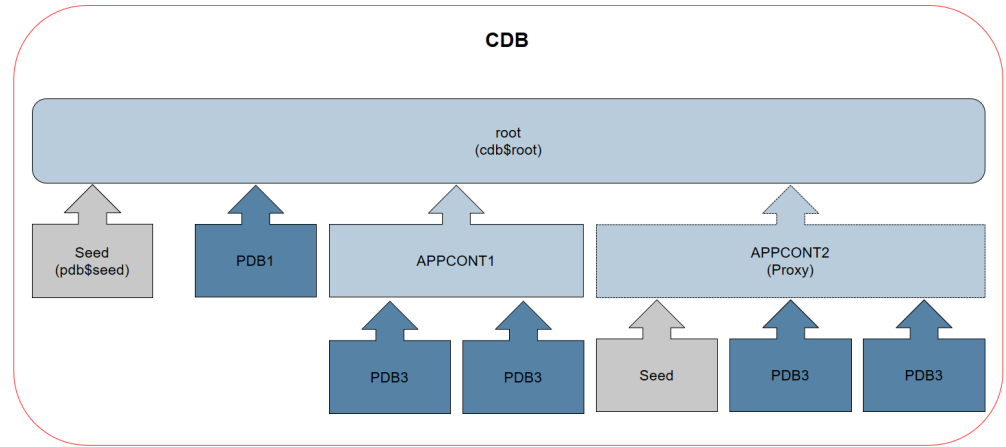
- Template for the creation of PDBs
- Created in all CDB
- Automatically created
 - SYSTEM
 - SYSAUX
 - TEMP
 - UNDO
- PDBs with different character sets can reside in the same CDB without requiring character set conversion



Container Databases

Application Container

- Is an optional user-created CDB component

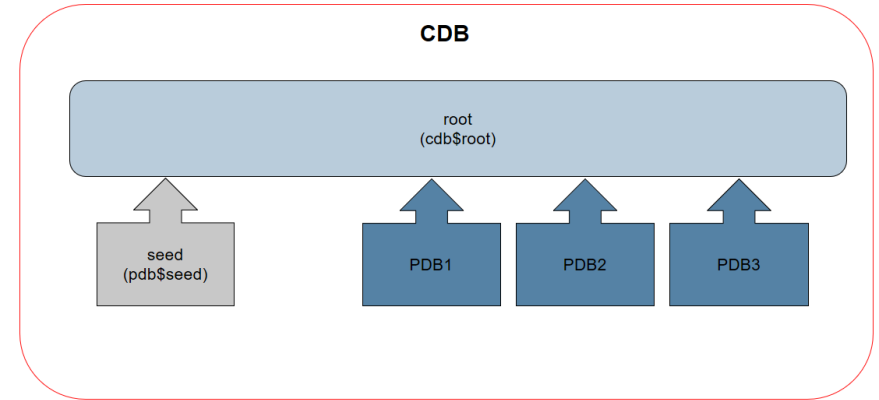


- Stores data and metadata for application
- A CDB includes zero or more application containers..
- The application root stores metadata and data that all application PDBs can share.
- CREATE PLUGGABLE DATABASE statement with the AS APPLICATION CONTAINER clause

Container Databases

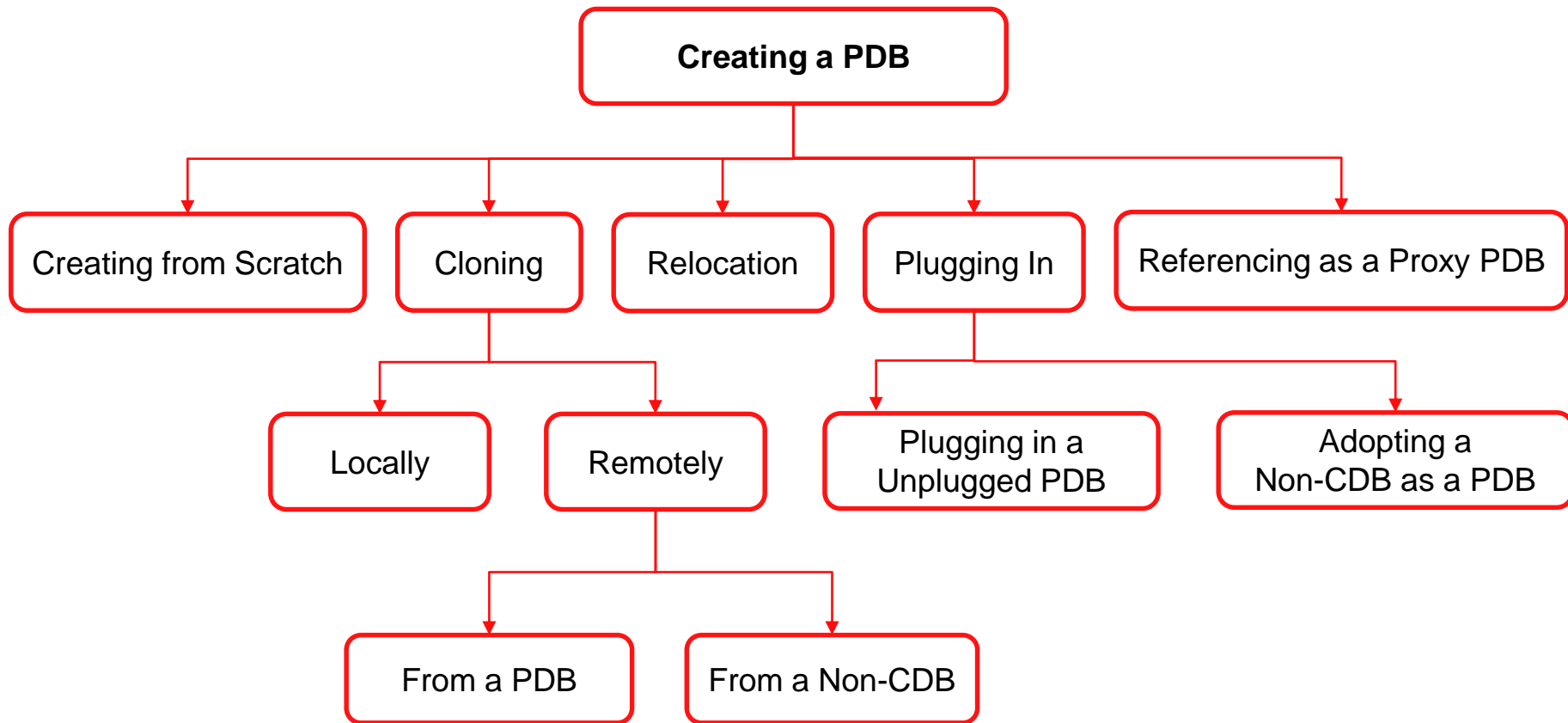
Standard PDB

- PDB is a user-created
 - set of schemas, objects
 - related structures that appears logically to a client application as a separate database.
- PDB is
 - owned by SYS regardless of which user created the PDB
SYS is a common user in the CDB.



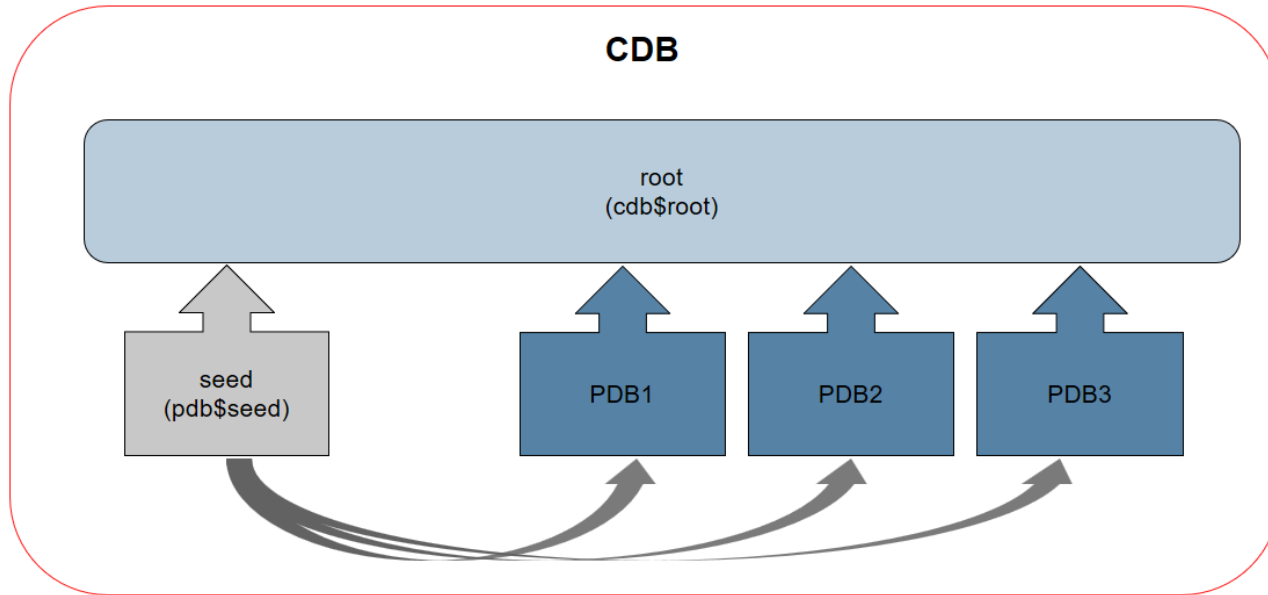
Pluggable Databases

```
SQL> CREATE PLUGGABLE DATABASE ...
```



Pluggable Databases

Creating a PDB from Scratch



Pluggable Databases

Creating a PDB from Scratch

- **CDB1**

```
$ sqlplus sys@cdb1 sys as sysdba
```

```
SQL> ALTER SYSTEM SET db_create_file_dest = '/u01/app/oracle/oradata';
```

```
SQL> CREATE PLUGGABLE DATABASE pdb1 ADMIN USER pdbladm IDENTIFIED BY *****;
```

OR

```
SQL> CREATE PLUGGABLE DATABASE pdb1 ADMIN USER pdbladm IDENTIFIED BY *****  
2 CREATE_FILE_DEST='/u01/app/oracle/oradata';
```

OR

```
SQL> CREATE PLUGGABLE DATABASE repotestpdb ADMIN USER pdbadm IDENTIFIED BY *****  
FILE_NAME_CONVERT=('/u01/app/oracle/oradata/cdb1/pdbseed/', '/u01/app/oracle/oradata/cdb1/pdb2/');
```

OR

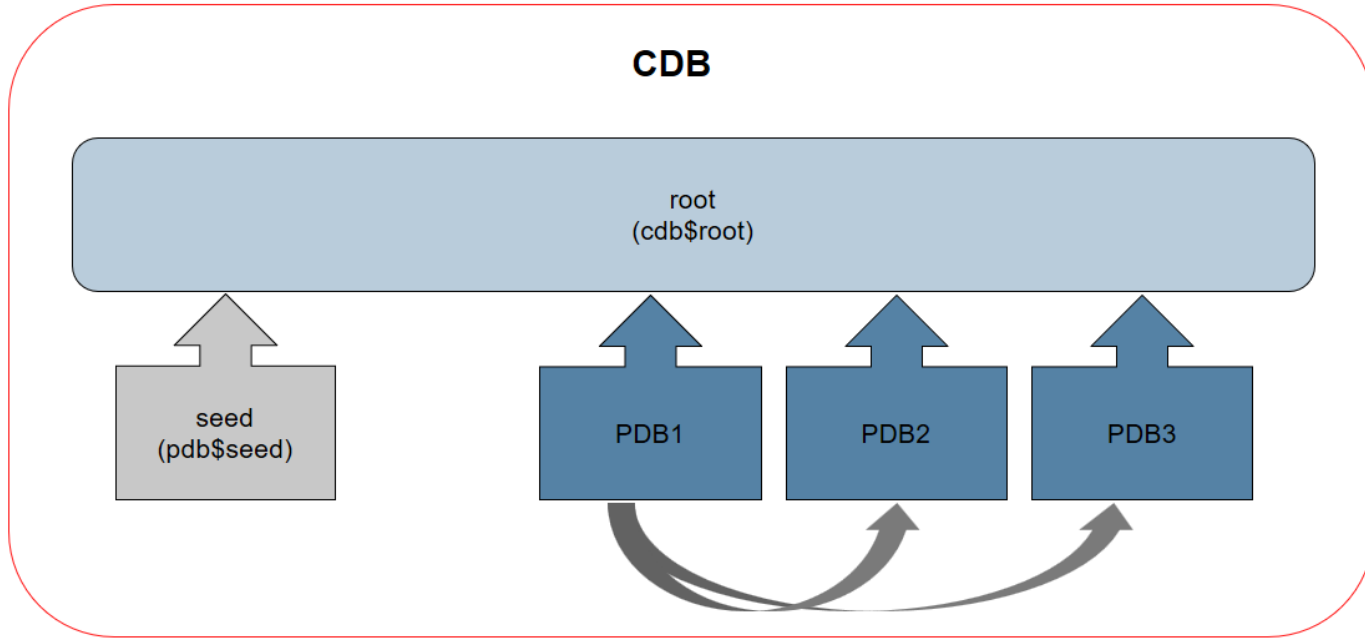
```
SQL> ALTER SESSION SET PDB_FILE_NAME_CONVERT='/u01/app/oracle/oradata/cdb1/pdbseed/'  
        , '/u01/app/oracle/oradata/cdb1/pdb3/';
```

```
SQL> CREATE PLUGGABLE DATABASE pdb3 ADMIN USER pdb_adm IDENTIFIED BY Password1 ROLES= (DBA) ;
```

You must open the new PDB in read/write mode for Oracle Database to complete the integration of the new PDB into the CDB.

Pluggable Databases

Locally Cloning a PDB



Pluggable Databases

Locally Cloning a PDB

- **CDB1**

```
$ sqlplus sys@cdb1 sys as sysdba
```

```
SQL> ALTER PLUGGABLE DATABASE PDB1 CLOSE;
```

```
SQL> ALTER PLUGGABLE DATABASE PDB1 OPEN READ ONLY;
```

```
SQL> CREATE PLUGGABLE DATABASE PDB2 FROM PDB1
```

```
2 STORAGE UNLIMITED TEMPFILE REUSE
```

```
3 FILE_NAME_CONVERT=('/u01/app/oracle/oradata/cdb1/pdb1/', '/u01/app/oracle/oradata/cdb1/pdb2');
```

- **PDB is in LOCAL UNDO MODE - HOT CLONING**

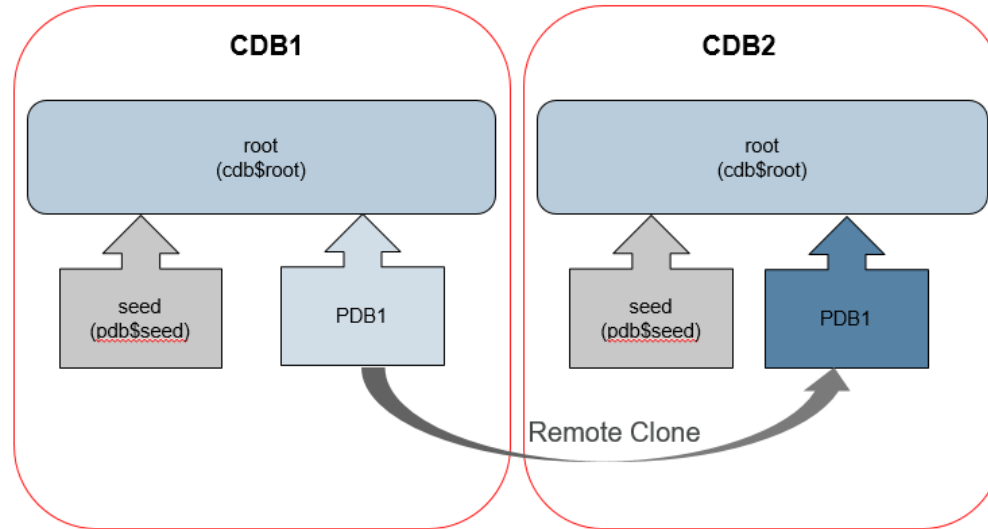
```
SQL> CREATE PLUGGABLE DATABASE PDB3 FROM PDB1
```

```
2 STORAGE UNLIMITED TEMPFILE REUSE
```

```
3 FILE_NAME_CONVERT=('/u01/app/oracle/oradata/cdb1/pdb1/', '/u01/app/oracle/oradata/cdb1/pdb3');
```

Pluggable Databases

Remotely Cloning a PDB



Pluggable Databases

Remotely Cloning a PDB

- **CDB1**

```
$ sqlplus sys@cdb1 sys as sysdba
```

```
SQL> CREATE USER c##remote_clone_user IDENTIFIED BY remote_clone_user CONTAINER=ALL;
```

```
SQL> GRANT CREATE SESSION, CREATE PLUGGABLE DATABASE TO c##remote_clone_user CONTAINER=ALL;
```

```
SQL> SELECT property_name, property_value FROM database_properties WHERE property_name = 'LOCAL_UNDO_ENABLED';
```

PROPERTY_NAME	PROPERTY_VALUE
LOCAL_UNDO_ENABLED	TRUE

- **CDB2**

```
$ sqlplus sys@cdb2 sys as sysdba
```

```
SQL> CREATE DATABASE LINK clone_link CONNECT TO c##remote_clone_user IDENTIFIED BY remote_clone_user USING 'CDB1';
```

```
SQL> CREATE PLUGGABLE DATABASE PDB1CLONE FROM PDB1@clone_link;
```

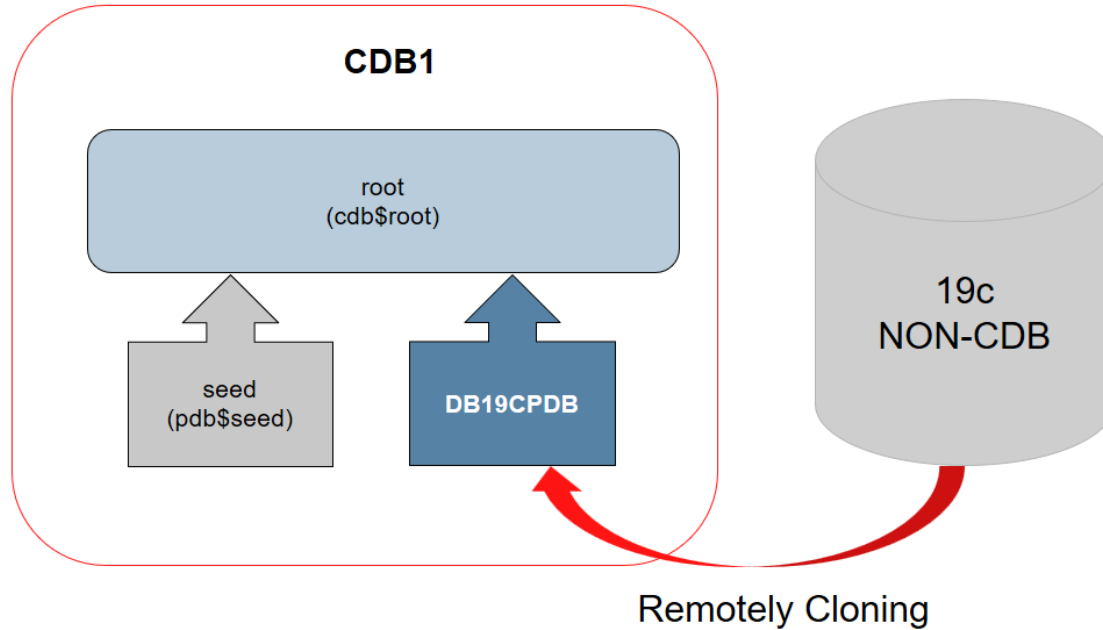
```
SQL> SELECT name, open_mode FROM v$pdbs WHERE name = 'PDB1CLONE';
```

NAME	OPEN_MODE
PDB1CLONE	MOUNTED

- In local undo mode and archivelog mode, we don't need to turn the remote database into read-only mode

Pluggable Databases

Remotely Cloning a PDB - From a Non-CDB



Pluggable Databases

Remotely Cloning a PDB - From a Non-CDB

- **DB19C NON-CDB**

```
$ sqlplus sys@db19c sys as sysdba
```

```
SQL> CREATE USER c##remote_clone_user IDENTIFIED BY remote_clone_user;
SQL> GRANT CREATE SESSION, CREATE PLUGGABLE DATABASE TO c##remote_clone_user;
SQL> SELECT log_mode FROM v$database;

LOG_MODE
-----
ARCHIVELOG
```

- **CDB1**

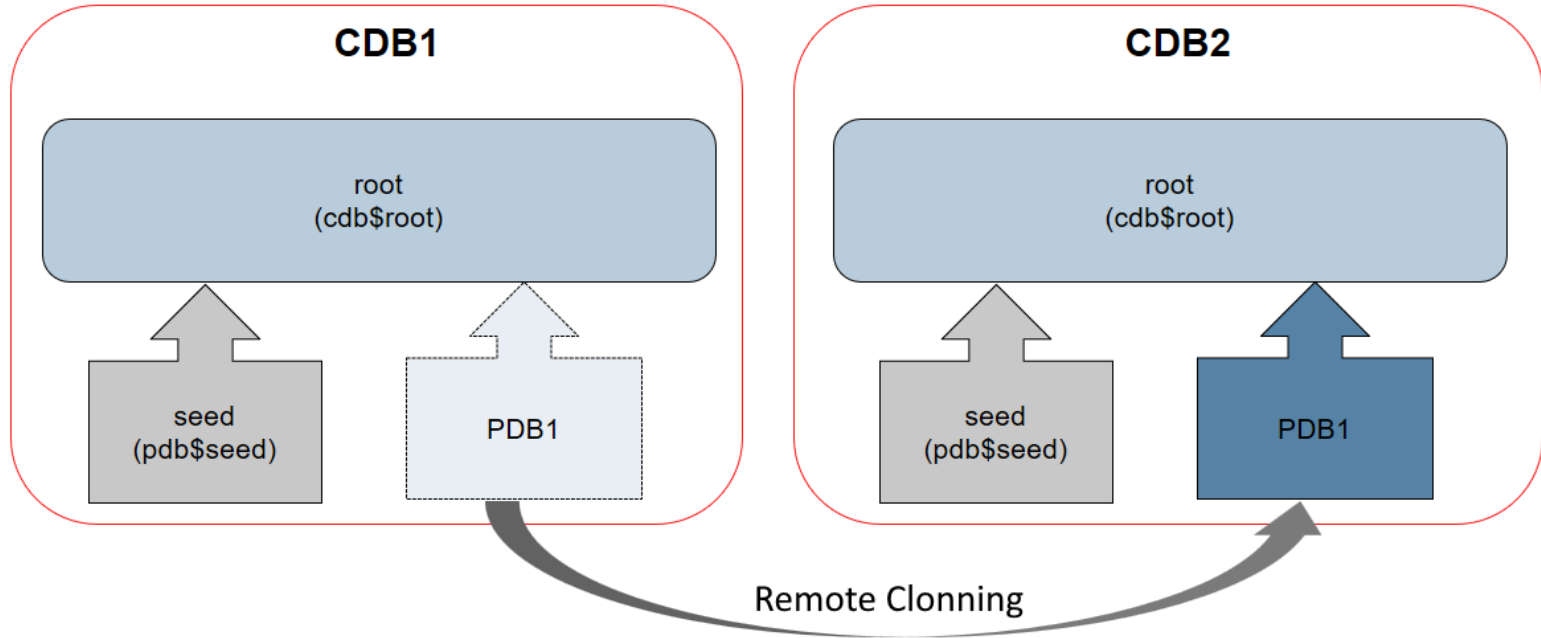
```
$ sqlplus sys@cdb1 sys as sysdba
```

```
SQL> CREATE DATABASE LINK clone_link CONNECT TO c##remote_clone_user IDENTIFIED BY remote_clone_user USING 'DB19C';
SQL> CREATE PLUGGABLE DATABASE DB19CPDB FROM NON$CDB@clone_link;
SQL> SELECT name, open_mode FROM v$pdbs WHERE name = 'DB19CPDB';
```

NAME	OPEN_MODE
DB19CPDB	MOUNTED

Pluggable Databases

Relocating a PDB



Pluggable Databases

Remotely Cloning a PDB

- **CDB1**

```
SQL> CREATE USER c##remote_clone_user IDENTIFIED BY remote_clone_user CONTAINER=ALL;
SQL> GRANT CREATE SESSION, CREATE PLUGGABLE DATABASE TO c##remote_clone_user CONTAINER=ALL;
```

```
SQL> SELECT property_name, property_value FROM database_properties
2  WHERE property_name = 'LOCAL_UNDO_ENABLED'
```

PROPERTY_NAME	PROPERTY_VALUE
LOCAL_UNDO_ENABLED	TRUE

- **CDB2**

```
SQL> CREATE DATABASE LINK clone_link CONNECT TO c##remote_clone_user IDENTIFIED BY remote_clone_user USING 'CDB1';
SQL> CREATE PLUGGABLE DATABASE PDB1CLONE FROM PDB1@clone_link RELOCATE;
```

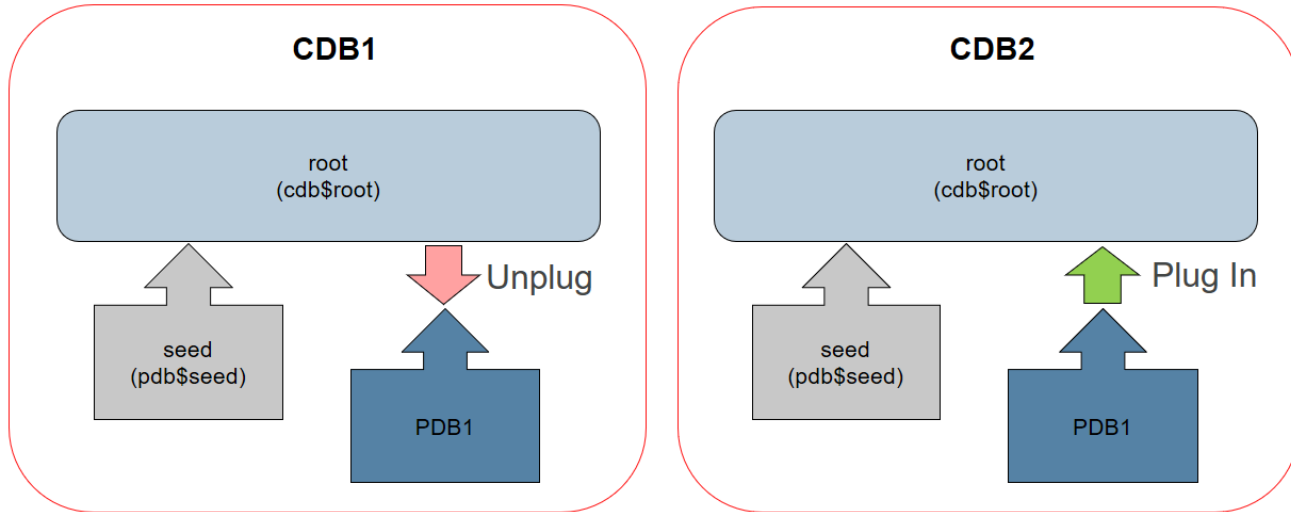
```
SQL> SELECT name, open_mode FROM v$pdb WHERE name = 'PDB1CLONE';
```

NAME	OPEN_MODE
PDB1CLONE	MOUNTED

- In local undo mode and archivelog mode, we don't need to turn the remote database into read-only mode

Pluggable Databases

Plugging In an Unplugged PDB



Pluggable Databases

Plugging In an Unplugged PDB

UNPLUG

```
SQL> CONNECT SYS@CDB1 AS SYSDBA
SQL> ALTER PLUGGABLE DATABASE pdb1 CLOSE;
SQL> ALTER PLUGGABLE DATABASE pdb1 UNPLUG INTO '/u01/app/oracle/oradata/cdb1/pdb1/pdb1.xml';
```

OR

```
SQL> ALTER PLUGGABLE DATABASE pdb1 UNPLUG INTO '/u01/app/oracle/oradata/cdb1/pdb1/pdb1.pdb';
```

PLUG-IN

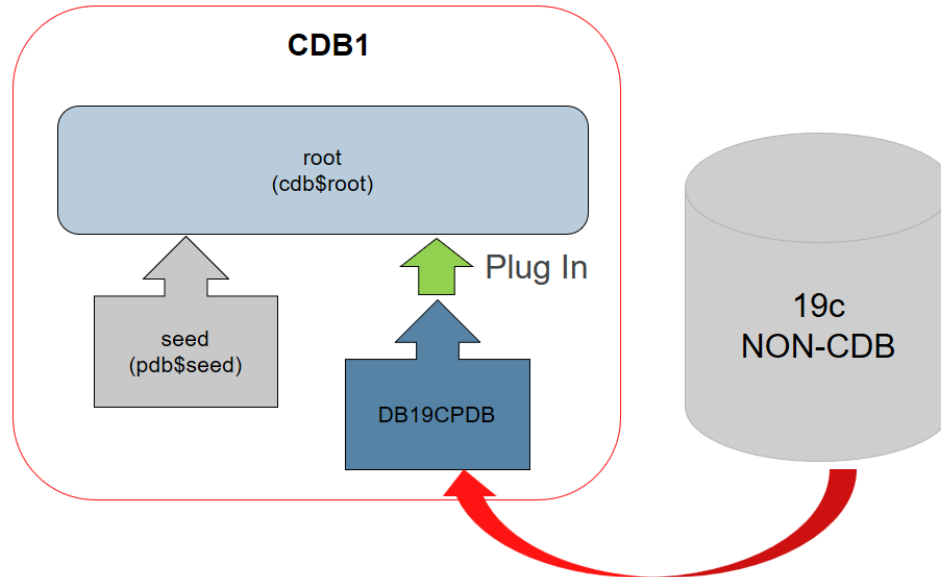
```
SQL> CONNECT SYS@CDB2 as sysdba
SQL> CREATE PLUGGABLE DATABASE pdb2 USING '/u01/app/oracle/oradata/cdb2/pdb2/pdb1.xml'
  2 FILE_NAME_CONVERT=('/u01/app/oracle/oradata/cdb1/pdb1/', '/u01/app/oracle/oradata/cdb2/pdb2/');
```

OR

```
SQL> CREATE PLUGGABLE DATABASE pdb2 USING '/u01/app/oracle/oradata/cdb2/pdb2/pdb1.pdb'
  2 FILE_NAME_CONVERT=('/u01/app/oracle/oradata/cdb1/pdb1/', '/u01/app/oracle/oradata/cdb2/pdb2/');
SQL> ALTER PLUGGABLE DATABASE pdb2 OPEN READ WRITE;
```

Pluggable Databases

Adopting a Non-CDB as a PDB



Adopting a Non-CDB as a PDB

Pluggable Databases

Adopting a Non-CDB as a PDB

- **DB19C NON-CDB**

```
sqlplus / as sysdba
```

- SQL> shutdown immediate
- SQL> startup mount exclusive
- SQL> alter database open read only;
- SQL> exec dbms_pdb.describe(pdb_descr_file=>'/u01/app/oracle/oradata/db19cnoncdb/noncdb.xml');
- SQL> shutdown immediate

- **CDB1**

```
sqlplus / as sysdba
```

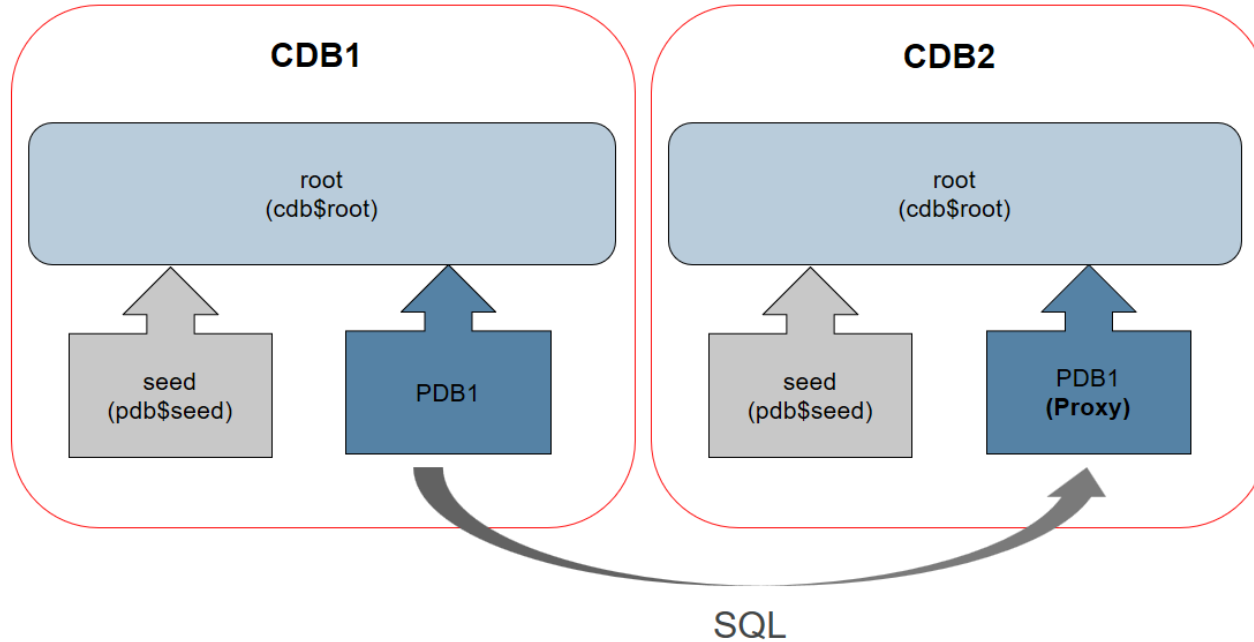
- SQL> create pluggable database **db19cpdb** as clone using '/ u01/app/oracle/oradata/db19cnoncdb/noncdb.xml'
2 file_name_convert=('/u01/app/oracle/oradata/db19cnoncdb','/u01/app/oracle/oradata/db19cpdb') copy;
- SQL> alter pluggable database db19cpdb open;

```
sqlplus sys@db19cpdb as sysdba
```

- SQL> @\$ORACLE_HOME/rdbms/admin/**noncdb_to_pdb.sql**

Pluggable Databases

Creating a PDB as a Proxy PDB



Pluggable Databases

Creating a PDB as a Proxy PDB

▪ CDB1

- SQL> CREATE USER c##remote_clone_user IDENTIFIED BY remote_clone_user CONTAINER=ALL;
- SQL> GRANT CREATE SESSION, CREATE PLUGGABLE DATABASE TO c##remote_clone_user CONTAINER=ALL;

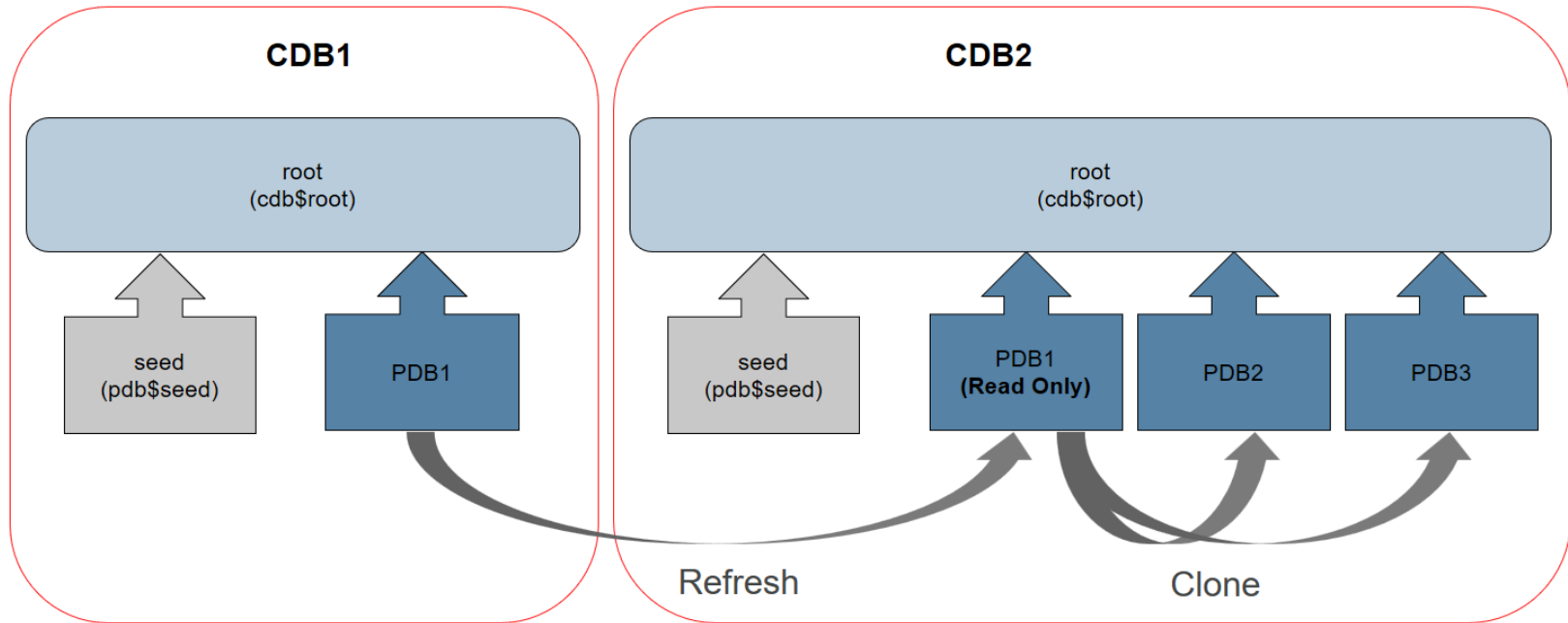
▪ CDB2

- SQL> CREATE DATABASE LINK clone_link CONNECT TO c##remote_clone_user IDENTIFIED BY remote_clone_user USING 'CDB1';
- SQL> CREATE PLUGGABLE DATABASE PDB1PROXY **AS PROXY** FROM pdb1@clone_link;
- SQL> ALTER PLUGGABLE DATABASE PDB1PROXY OPEN;
- SQL> SELECT name, open_mode FROM v\$pdb WHERE name = 'PDB1PROXY';

NAME	OPEN_MODE
PDB1PROXY	READ WRITE

Pluggable Databases

Refreshable PDB



Pluggable Databases

Refreshable PDB

▪ CDB1

```
- SQL> CREATE USER c##remote_clone_user IDENTIFIED BY remote_clone_user CONTAINER=ALL;
- SQL> GRANT CREATE SESSION, CREATE PLUGGABLE DATABASE TO c##remote_clone_user CONTAINER=ALL;
```

```
- SQL> SELECT property_name, property_value FROM database_properties
2 WHERE property_name = 'LOCAL_UNDO_ENABLED';
```

PROPERTY_NAME	PROPERTY_VALUE
LOCAL_UNDO_ENABLED	TRUE

▪ CDB2

```
- SQL> CREATE DATABASE LINK clone_link CONNECT TO c##remote_clone_user IDENTIFIED BY remote_clone_user USING 'CDB1';
- SQL> CREATE PLUGGABLE DATABASE PDB1RO FROM pdb1@clone_link REFRESH MODE MANUAL;;
```

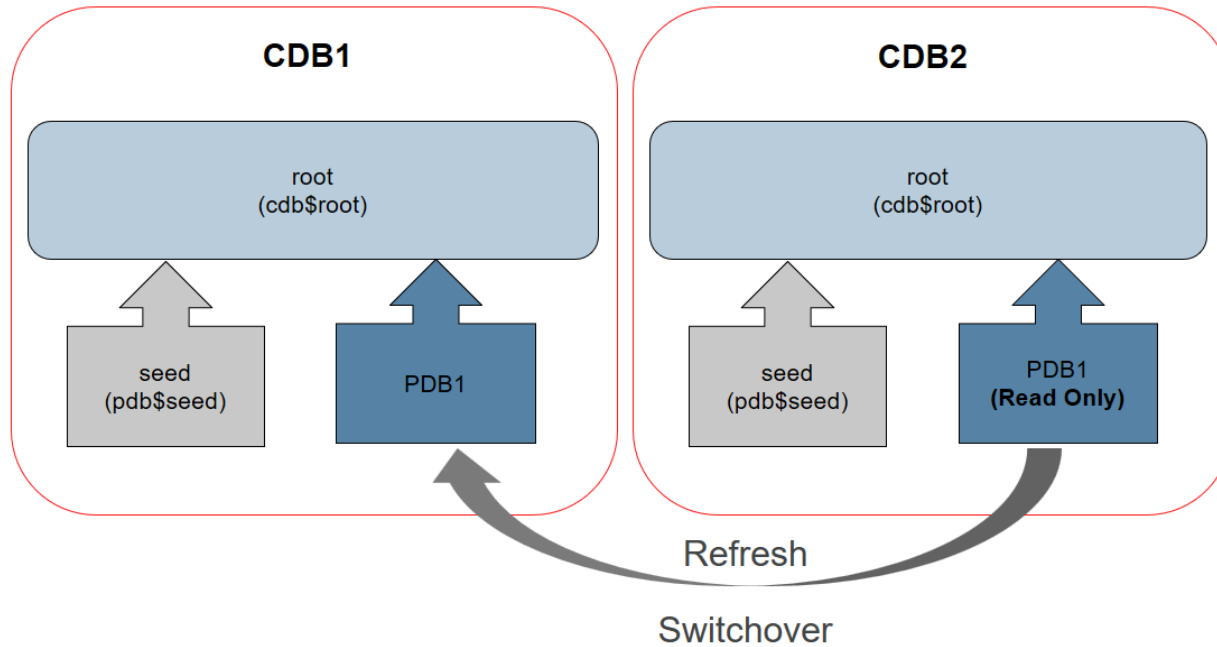
```
- SQL> SELECT name, open_mode FROM v$pdb WHERE name = 'PDB1RO';
```

NAME	OPEN_MODE
PDB1RO	READ ONLY

```
- SQL> ALTER PLUGGABLE DATABASE PDB1RO REFRESH MODE EVERY 60 MINUTES;
- SQL> ALTER PLUGGABLE DATABASE PDB1RO REFRESH MODE EVERY 120 MINUTES;
```

Pluggable Databases

Refreshable PDB Switchover



Pluggable Databases

Refreshable PDB Switchover

▪ CDB1

```
- SQL> CREATE DATABASE LINK clone_link CONNECT TO c##remote_clone_user IDENTIFIED BY remote_clone_user USING 'CDB2';

- SQL> ALTER SESSION SET CONTAINER = PDB1;

- SQL> ALTER PLUGGABLE DATABASE REFRESH MODE MANUAL
  2 FROM PDB1RO@clone_link
  3 SWITCHOVER;
```

▪ CDB2

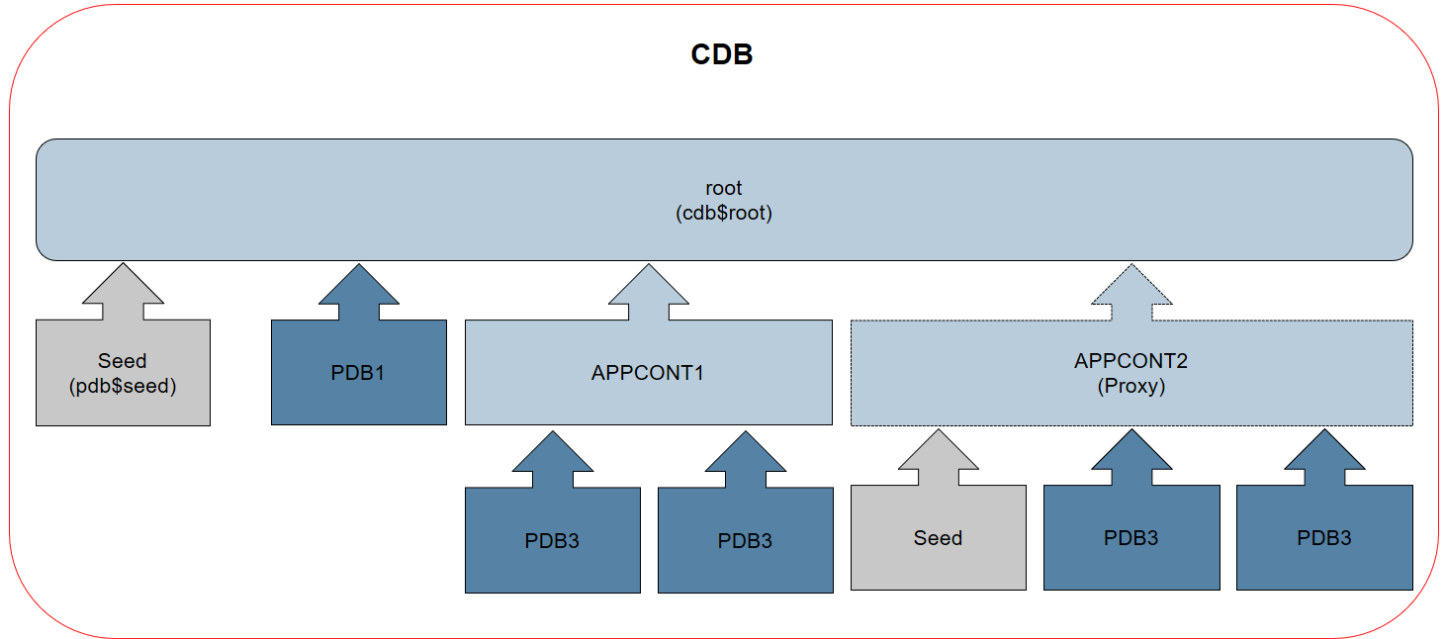
```
- SQL> CREATE DATABASE LINK clone_link CONNECT TO c##remote_clone_user IDENTIFIED BY remote_clone_user USING 'CDB1';

- SQL> ALTER SESSION SET CONTAINER = PDB1RO;

- SQL> ALTER PLUGGABLE DATABASE REFRESH MODE MANUAL
  2 FROM PDB1@clone_link
  3 SWITCHOVER;
```

Pluggable Databases

Creating Application Containers



Pluggable Databases

Creating Application Containers

- **CDB1**

```
$ sqlplus sys@cdb1 sys as sysdba
```

```
SQL> ALTER SYSTEM SET db_create_file_dest = '/u01/app/oracle/oradata';
```

```
SQL> CREATE PLUGGABLE DATABASE appcon1 AS APPLICATION CONTAINER ADMIN USER adm IDENTIFIED BY *****;
```

```
SQL> ALTER PLUGGABLE DATABASE appcon1 OPEN;
```

```
SQL> ALTER SESSION SET CONTAINER = appcon1;
```

```
SQL> CREATE PLUGGABLE DATABASE apppdb1 ADMIN USER adm IDENTIFIED BY *****
```

```
SQL> ALTER PLUGGABLE DATABASE apppdb1 OPEN;
```

Pluggable Databases

Creating Application Containers

- **CDB1**

```
$ sqlplus sys@cdb1 sys as sysdba
```

```
SQL> ALTER SYSTEM SET db_create_file_dest = '/u01/app/oracle/oradata';
```

```
SQL> CREATE PLUGGABLE DATABASE appcon1 AS APPLICATION CONTAINER ADMIN USER adm IDENTIFIED BY *****;
```

```
SQL> ALTER PLUGGABLE DATABASE appcon1 OPEN;
```

```
SQL> ALTER SESSION SET CONTAINER = appcon1;
```

```
SQL> CREATE PLUGGABLE DATABASE apppdb1 ADMIN USER adm IDENTIFIED BY *****
```

```
SQL> ALTER PLUGGABLE DATABASE apppdb1 OPEN;
```

Pluggable Databases

Installing Applications

- **CDB1**

```
$ sqlplus sys@cdb1 sys as sysdba
SQL> ALTER SESSION SET CONTAINER = appcon1;
SQL> ALTER PLUGGABLE DATABASE APPLICATION app1 BEGIN INSTALL '1.0';

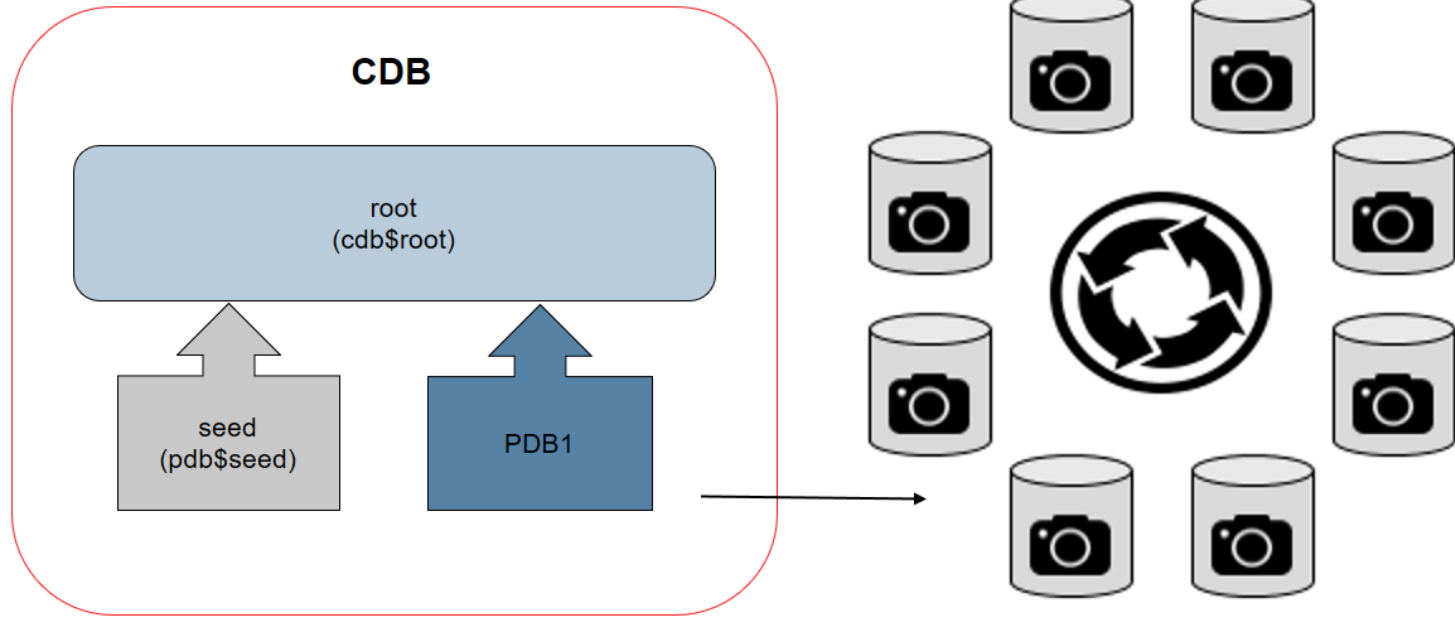
SQL> CREATE TABLESPACE ...
SQL> CREATE USER ...
SQL> GRANT CREATE SESSION, CREATE TABLE TO ...
SQL> CREATE TABLE ...
SQL> INSERT INTO ...

...
SQL> ALTER PLUGGABLE DATABASE APPLICATION app1 END INSTALL;

SQL> ALTER SESSION SET CONTAINER = apppdb1;
SQL> ALTER PLUGGABLE DATABASE APPLICATION app1 SYNC;
```


Pluggable Databases

Snapshot Carousel PDB



Pluggable Databases

Snapshot Carousel PDB

- Library of PDB snapshots
 - **PDB snapshot** is a point-in-time copy of a PDB
 - Maximum number of snapshots `MAX_PDB_SNAPSHOTS = 8` (default)
 - `NONE` : The PDB does not support snapshots.
 - `MANUAL` : The PDB supports snapshots, but they are only created manually requested.
 - `EVERY n HOURS` : A snapshot is automatically created every "n" hours. Where "n" is between 1 and 1999.
 - `EVERY n MINUTES` : A snapshot is automatically created every "n" minutes. Where "n" is between 1 and 2999.

```
SQL> CONNECT / AS SYSDBA
```

```
SQL> CREATE PLUGGABLE DATABASE PDB2 ADMIN USER PDBADMIN IDENTIFIED BY *****
```

```
FILE_NAME_CONVERT=('pdbseed', 'pdb2')
```

```
SNAPSHOT MODE EVERY 24 HOURS;
```

```
SQL> ALTER PLUGGABLE DATABASE PDB2 OPEN;
```

```
SQL> ALTER PLUGGABLE DATABASE PDB2 SAVE STATE;
```

Pluggable Databases

Snapshot Carousel PDB

```
SQL> SELECT con_id, con_name, snapshot_name,  
           snapshot_scn, full_snapshot_path  
2  FROM cdb_pdb_snapshots  
3  ORDER BY con_id, snapshot_scn;
```

CON_ID	CON_NAME	SNAPSHOT_NAME	SNAPSHOT_SCN	FULL_SNAPSHOT_PATH
4	PDB2	SNAP_688979926_996491289	1764864	/u02/oradata/snap_688979926_1764864.pdb

■ Recovering From a PDB Snapshot

```
SQL> CREATE PLUGGABLE DATABASE PDB2COPY FROM pdb2 USING SNAPSHOT SNAP_688979926_996491631;
```

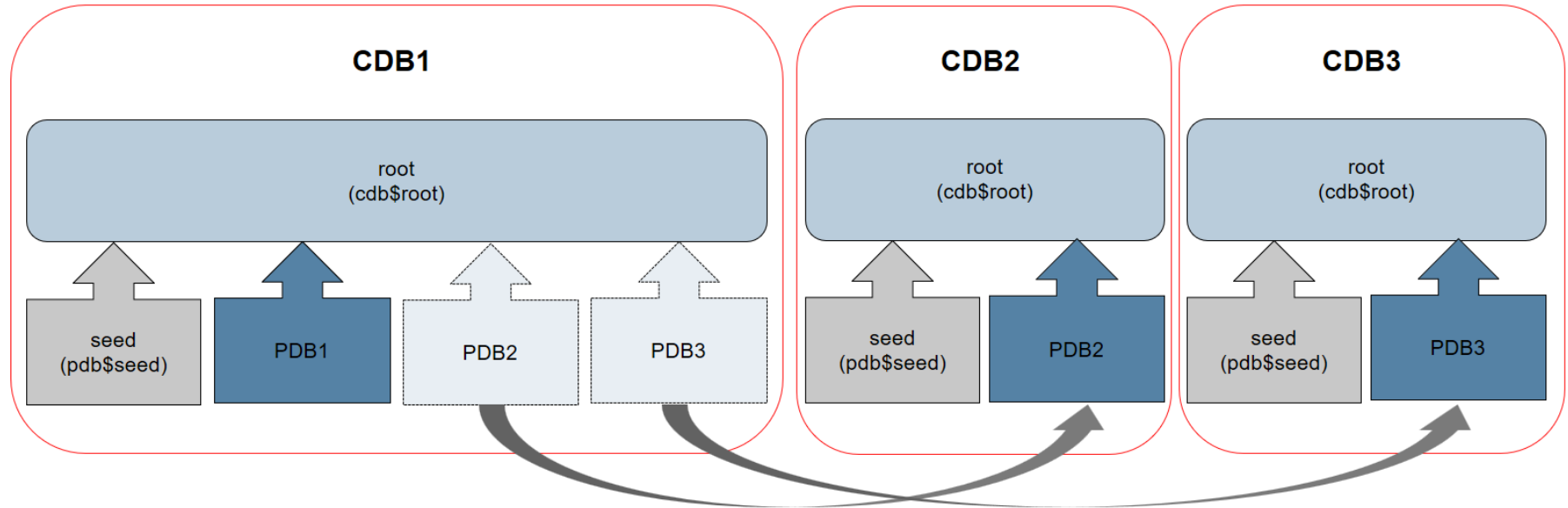
OR

```
SQL> CREATE PLUGGABLE DATABASE PDB2COPY FROM pdb2 USING SNAPSHOT SNAP_688979926_996491631  
      SNAPSHOT MODE EVERY 24 HOURS;
```

```
SQL> ALTER PLUGGABLE DATABASE DROP SNAPSHOT SNAP_688979926_996491289;
```

Pluggable Databases

Container Database (CDB) Fleet Management



Pluggable Databases

Container Database (CDB) Fleet Management – Lead CDB

- **CDB1 - LEAD**

```
SQL> CONN sys/SysPassword1@cdb1 AS SYSDBA
```

```
SQL> ALTER DATABASE SET lead_cdb = TRUE;
```

```
SQL> SELECT property_name, property_value FROM database_properties  
WHERE property_name = 'LEAD_CDB';
```

PROPERTY_NAME	PROPERTY_VALUE
LEAD_CDB	TRUE

```
SQL> CREATE USER c##cdb_fleet_link_user IDENTIFIED BY cdb_fleet_link_user;
```

```
SQL> GRANT CREATE SESSION TO c##cdb_fleet_link_user;
```

Pluggable Databases

Container Database (CDB) Fleet Management – Members CDB

- CDB2 - MEMBER

```
SQL> CONN sys/SysPassword1@cdb2 AS SYSDBA
SQL> CREATE PUBLIC DATABASE LINK lead_cdb_link
      CONNECT TO c##cdb_fleet_link_user IDENTIFIED BY
      cdb_fleet_link_user USING 'cdb1';
```

```
SQL> ALTER DATABASE SET lead_cdb_uri =
      'dblink:LEAD_CDB_LINK';
```

```
SQL> SELECT property_value FROM database_properties
      WHERE property_name = 'LEAD_CDB_URI';
```

```
PROPERTY_VALUE
-----
dblink:LEAD_CDB_LINK
```

- CDB3 - MEMBER

```
SQL> CONN sys/SysPassword1@cdb2 AS SYSDBA
SQL> CREATE PUBLIC DATABASE LINK lead_cdb_link
      CONNECT TO c##cdb_fleet_link_user IDENTIFIED BY
      cdb_fleet_link_user USING 'cdb1';
```

```
SQL> ALTER DATABASE SET lead_cdb_uri =
      'dblink:LEAD_CDB_LINK';
```

```
SQL> SELECT property_value FROM database_properties
      WHERE property_name = 'LEAD_CDB_URI';
```

```
PROPERTY_VALUE
-----
dblink:LEAD_CDB_LINK
```

Pluggable Databases

Container Database (CDB) Fleet Management – LEAD CDB

- LEAD CDB1

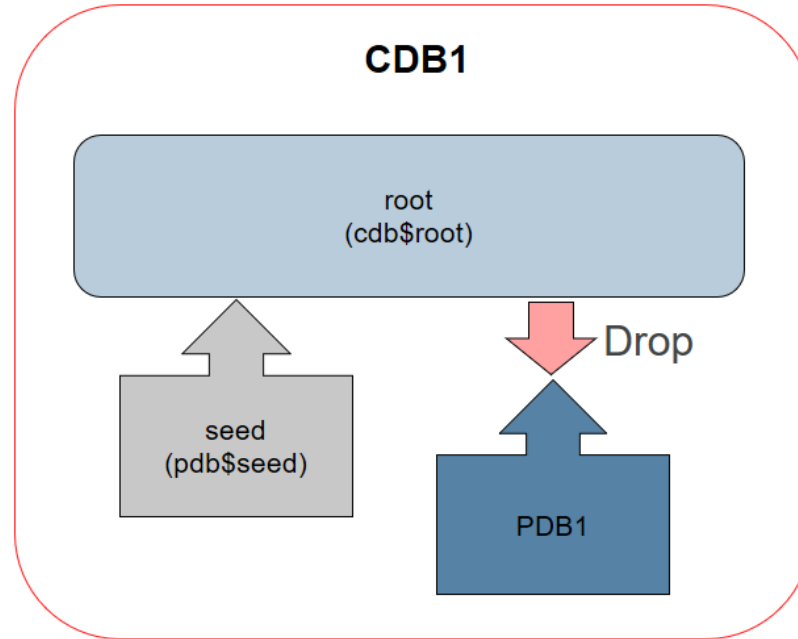
```
SQL> CONN sys/SysPassword1@cdb1 AS SYSDBA
```

```
SQL> SELECT con_id,  
2      name,  
3      open_mode,  
4      proxy_pdb  
5 FROM v$pdb  
6 ORDER BY name;
```

CON_ID	NAME	OPEN_MODE	PROXY_PDB
2	PDB\$SEED	READ ONLY	NO
3	PDB1	READ WRITE	NO
5	PDB4	MOUNTED	YES
7	PDB5	MOUNTED	YES

Pluggable Databases

Removing a PDB



Pluggable Databases

Removing a PDB

- SQL> CONNECT sys@**CDB1** AS SYSDBA
- SQL> ALTER PLUGGABLE DATABASE pdb1 CLOSE;
- SQL> DROP PLUGGABLE DATABASE PDB1 INCLUDING DATAFILES;

OR

- SQL> DROP PLUGGABLE DATABASE PDB1 KEEP DATAFILES;

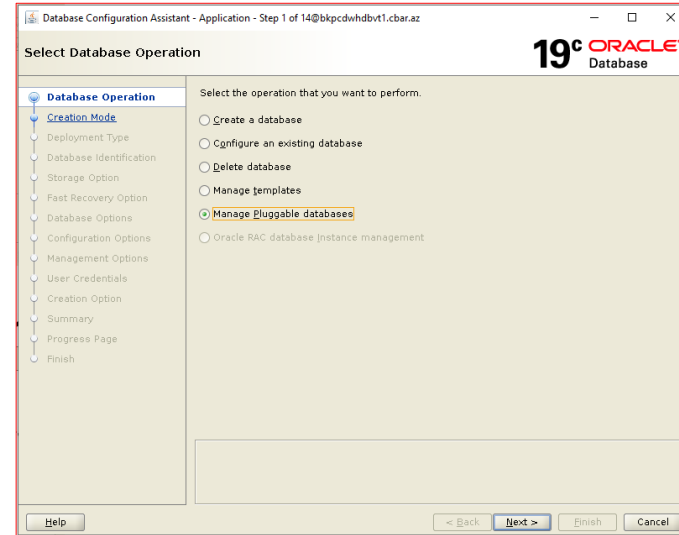
- If a PDB was created with the SNAPSHOT COPY clause, then you must specify INCLUDING DATAFILES when you drop the PDB

Administering a Multitenant Environment

Administering a Multitenant Environment

Using DBCA

- DBCA supporting manage container database
 - Manage Pluggable Database
 - Create a Pluggable Database
 - Delete a Pluggable Database
 - Unplug a Pluggable Database
 - Configure a Pluggable Database



- It also enables you to create, relocate, clone, plug in, and unplug PDBs
 - Silent mode

Administering a Multitenant Environment

Users in CDB

- **Common User** : The user is present in all containers (root and all PDBs).

```
SQL> CONNECT SYS@CDB1 AS SYSDBA
```

- **Create the common user using the CONTAINER clause.**

```
SQL> CREATE USER c##test_user1 IDENTIFIED BY password1 CONTAINER=ALL;  
SQL> GRANT CREATE SESSION TO c##test_user1 CONTAINER=ALL;
```

- **Create the common user using the default CONTAINER setting.**

```
SQL> CREATE USER c##test_user2 IDENTIFIED BY password1;  
SQL> GRANT CREATE SESSION TO c##test_user2;
```

Administering a Multitenant Environment

Users in PDB

- **Local User** : The user is only present in a specific PDB
 - The same username can be present in multiple PDBs, but they are unrelated

```
SQL> CONNECT SYS@CDB1 AS SYSDBA
SQL> ALTER SESSION SET CONTAINER = pdb1;
```

- Create the local user using the CONTAINER clause.

```
SQL> CREATE USER test_user3 IDENTIFIED BY password1 CONTAINER=CURRENT;
SQL> GRANT CREATE SESSION TO test_user3 CONTAINER=CURRENT;
```

- Create the local user using the default CONTAINER setting

```
SQL> CONNECT system/password@pdb1
SQL> CREATE USER test_user4 IDENTIFIED BY password1;
SQL> GRANT CREATE SESSION TO test_user4;
```

Administering a Multitenant Environment

Roles in CDB

- **Common Role** : The role is present in all containers (root and all PDBs)

```
SQL> CONN / AS SYSDBA
```

- Create the common role

```
SQL> CREATE ROLE c##test_role1;
```

```
SQL> GRANT CREATE SESSION TO c##test_role1;
```

- Grant it to a common user.

```
GRANT c##test_role1 TO c##test_user1 CONTAINER=ALL;
```

- Grant it to a local user.

```
SQL> ALTER SESSION SET CONTAINER = pdb1;
```

```
SQL> GRANT c##test_role1 TO test_user3;
```

Administering a Multitenant Environment

Roles in PDB

- **Local Role** : The role is only present in a specific PDB.
 - The same role name can be used in multiple PDBs, but they are unrelated.

```
SQL> CONN / AS SYSDBA
```

- Switch container.

```
SQL> ALTER SESSION SET CONTAINER = pdb1;
```

- Alternatively, connect to a local or common user with the PDB service

```
SQL> CONN system/password@pdb1
```

- Create the common role

```
SQL> CREATE ROLE test_role1;
```

```
SQL> GRANT CREATE SESSION TO test_role1;
```

- Grant it to a common and local user

```
SQL> GRANT test_role1 TO c##test_user1;
```

```
SQL> GRANT test_role1 TO test_user3;
```

Administering a Multitenant Environment

Connecting to CDB

- Using SID
- Using TNSNAMES and EZCONNECT

```
[oracle@bkpcdwhdbvt1 admin]$ export ORACLE_SID=cbacdbt1
[oracle@bkpcdwhdbvt1 admin]$ sqlplus / as sysdba

SQL*Plus: Release 19.0.0.0.0 - Production on Wed Oct 30 12:01:10 2019
Version 19.3.0.0.0

Copyright (c) 1982, 2019, Oracle. All rights reserved.

Connected to:
Oracle Database 19c Enterprise Edition Release 19.0.0.0.0 - Production
Version 19.3.0.0.0

SQL> select name from v$pdbs;

NAME
-----
PDB$SEED
WSSTESTPDB
DWHTESTPDB
REPOTESTPDB
```

```
CBACDBT1 =
  (DESCRIPTION =
    (ADDRESS = (PROTOCOL = TCP)(HOST = bkpcdwhdbvt1)(PORT = 1521))
    (CONNECT_DATA =
      (SERVER = DEDICATED)
      (SERVICE_NAME = cbacdbt1)
    )
  )
```

```
SQL>
SQL> connect sys/***** @//bkpcdwhdbvt1:1521/cbacdbt1 as sysdba
Connected.
SQL>
SQL> connect sys/***** @cbacdbt1 as sysdba
Connected.
SQL>
```


Administering a Multitenant Environment

Connecting to PDB

- Using TNSNAMES and EZCONNECT

```
REPOTESTPDB =
  (DESCRIPTION =
    (ADDRESS_LIST =
      (ADDRESS = (PROTOCOL = TCP)(HOST = bkpcdwhdbvt1)(PORT = 1521))
    )
    (CONNECT_DATA =
      (SERVICE_NAME = repotestpdb)
    )
  )
```

```
SQL>
SQL> connect sys/***** @repotestpdb as sysdba
Connected.
SQL>
SQL> connect sys/***** @//bkpcdwhdbvt1:1521/repotestpdb as sysdba
Connected.
SQL>
```

- Can't use SID for connecting to PDB

- Using ALTER SESSION SET CONTAINER

```
SQL> show con_name
CON_NAME
-----
CDB$ROOT
SQL>
SQL> alter session set container=repotestpdb;
Session altered.
SQL>
SQL> show con_name
CON_NAME
-----
REPOTESTPDB
SQL>
SQL> alter session set container=cdb$root;
Session altered.
SQL> show con_name
CON_NAME
-----
CDB$ROOT
```

Using Oracle Features in a Multitenant Environment

Using Oracle Features in a Multitenant Environment

Recovery Manager - RMAN

- Backup CDB

```
RMAN> CONNECT TARGET SYS@CDB1  
RMAN> BACKUP DATABASE PLUS ARCHIVELOG;
```

- Backup Pluggable Database

```
RMAN> CONNECT TARGET SYS@CDB1  
RMAN> BACKUP PLUGGABLE DATABASE pdb1;  
RMAN> BACKUP PLUGGABLE DATABASE pdb1, pdb2, pdb3;
```

- Point In Time Recovery

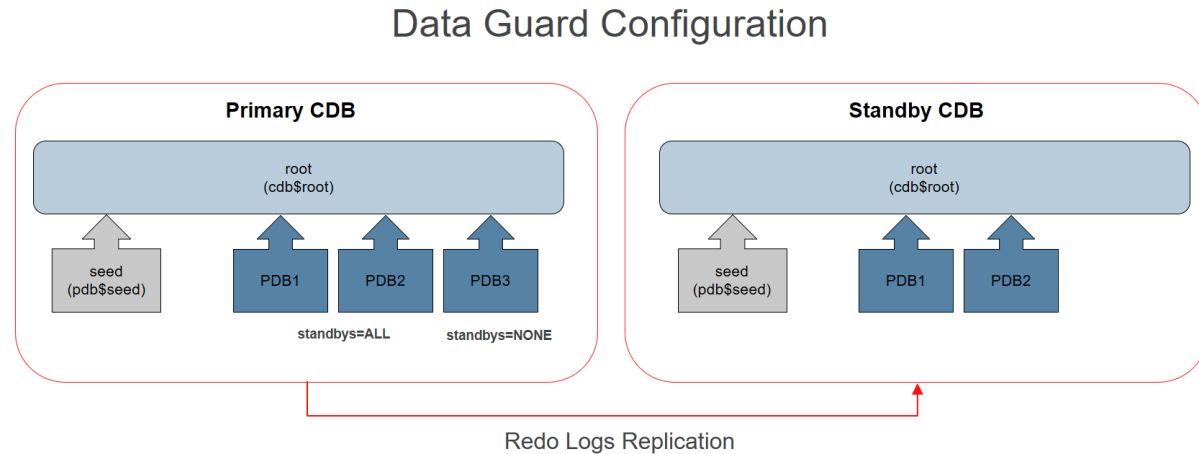
```
RMAN> ALTER PLUGGABLE DATABASE pdb5 CLOSE;  
RMAN>run  
    {  
        SET UNTIL SCN 1066;  
        RESTORE PLUGGABLE DATABASE pdb5;  
        RECOVER PLUGGABLE DATABASE pdb5;  
    }  
RMAN> ALTER PLUGGABLE DATABASE pdb5  
OPEN RESETLOGS;
```

Using Oracle Features in a Multitenant Environment

Data Guard Configurations

- Data Guard Configuring on CDB

- Control Files
- Redo Logs
- Standby Redo Logs
- Archive Log Mode
- Flashback Logs

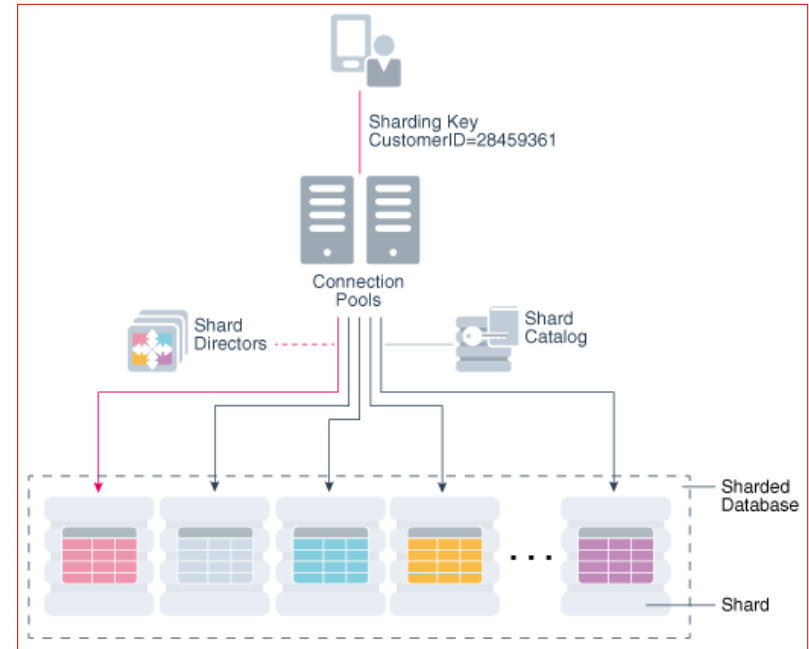


```
SQL> CREATE PLUGGABLE DATABASE ...  
      STANDBYS= NONE | ALL [EXCEPT ('pdb_name', 'pdb_name', ...)] }
```

Using Oracle Features in a Multitenant Environment

Sharding

- Horizontal Partitioned Database
 - Pluggable databases
 - As Shards
 - Shard catalogs



Oracle Multitenant 20c

Oracle Multitenant



Oracle Database 20c

- Non-CDB Architecture Depreciated/Desupported
 - From Oracle database 12c version 12.1.0.2, the non-CDB architecture is **DEPRECATED**
- In **Oracle Database 20c** the non-CDB architecture will be **DESUPPORTED**
 - *“For all offerings, if you are not licensed for Oracle Multitenant, then you may have up to 3 PDBs in a given container database at any time.”*

Q&A

THANK YOU

#UKOUG
#TECHFEST2019
#PASSTHEKNOWLEDGE

Mahir M. Quluzade

@marzade

mahir.quluzade@gmail.com

<http://www.mahir-quluzade.com>

Baku, Azerbaijan

