Oracle Database 12*c*: Managing Multitenant Architecture

Activity Guide

D79128GC10
Edition 1.0
July 2013
D82585

ORACLE

Authors

Dominique Jeunot, Jean-François Verrier

Technical Contributors and Reviewers

Bill Millar, Branislav Valny, Gerlinde Frenzen, Joel Goodman, Harald Van Breederode, Maria Billings, Randy Urbano

This book was published using: Oracle Tutor

Table of Contents

Practices for Lesson 1: Enterprise Manager Cloud Control and Other Tools

Chapter 1

Practices for Lesson 1: Overview

Practices Overview

Your system currently has Oracle Database 12*c* software installed, as well as three precreated databases called `orcl`, `orcl2`, and `cdb1`.

You will act as an Enterprise Manager administrator. You will access Oracle Enterprise Manager Cloud Control 12*c* as the `sysman` user with the `Oracle123` password and select **Summary** as your home page. You will explore some of the Oracle Enterprise Manager Cloud Control 12*c* functionalities through the different menus and options. And lastly, you will add the `cdb1` database as a monitored target.

Practice 1-1: Accessing Enterprise Manager

Overview

In this practice, you will access Oracle Enterprise Manager Cloud Control 12*c* as the `sysman` user with the `Oracle123` password and select **Summary** as your home page.

Assumptions

You reviewed the Oracle Enterprise Manager 12*c*: Console Overview and Customization demonstration or have the equivalent navigation knowledge.

Tasks

1. Click the Firefox icon on the top panel (toolbar region) above the desktop to open a browser to access the Enterprise Manager Cloud Control console.
2. Enter the URL for Cloud Control: https://localhost:7802/em.
   If the OMS is not started, start it as follows:

```
$ export OMS_HOME=/u01/app/oracle/product/middleware/oms
$ $OMS_HOME/bin/emctl start oms
Oracle Enterprise Manager Cloud Control 12c Release 2
Copyright (c) 1996, 2012 Oracle Corporation.  All rights
reserved.
Starting Oracle Management Server...
Starting WebTier...
WebTier Successfully Started
Oracle Management Server Successfully Started
Oracle Management Server is Up
WARNING: Limit of open file descriptors is found to be 1024.
The OMS has been started but it may run out of descriptors under
heavy usage.
For proper functioning of OMS, please set "ulimit -n" to be at
least 4096.
$
```

3. Most likely, you will receive a Secure Connection Failed message and you would need to add a security exception.
   a. At the end of the alert box, click **I Understand the Risks**.
   b. At the bottom of the page, click **Add Exception**.
   c. In the Add Security Exception pop-up window, click **Get Certificate**.
   d. Confirm that “Permanently store this exception” is selected in your training environment and click **Confirm Security Exception**.
4. The Enterprise Manager Cloud Control console appears.
5. Enter **`sysman`** in the User Name field and **`Oracle123`** in the Password field. Then click **Login**.

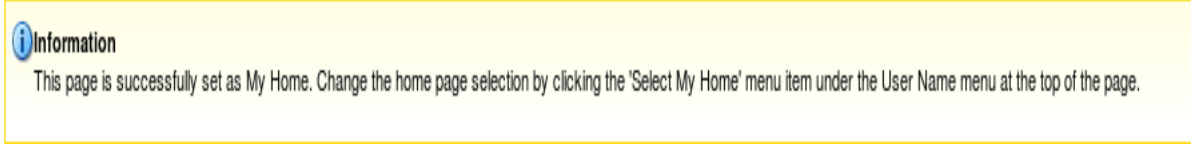
6. The first time a new user logs in to Enterprise Manager, a the user is prompted to accept the license agreement. You have to accept only once. During subsequent logins, the license agreement page will not appear.

7. The “Select Enterprise Manager Home Page” page appears with choices, such as the following:
   - Summary
   - Databases
   - Incidents
   - SOA
   - Middleware
   - Composite Application
   - Service Request
   - Services
   - Business Applications
   - Compliance Dashboard

   Each choice has a Preview and a Select As My Home button.

   The page also has global menus with the following choices: Enterprise, Targets, Favorites, History, and Search Target Name (next to the search entry field). Each of the menu items has drop-down menus with further choices.

   Preview any images that interest you.
8. Click the “Select As My Home” next to the **Summary** choice. After being successfully set, it informs you how to change it.



Practice 1-2: Adding a Multitenant Container Database Instance as a New Target Monitored by EM Cloud Control

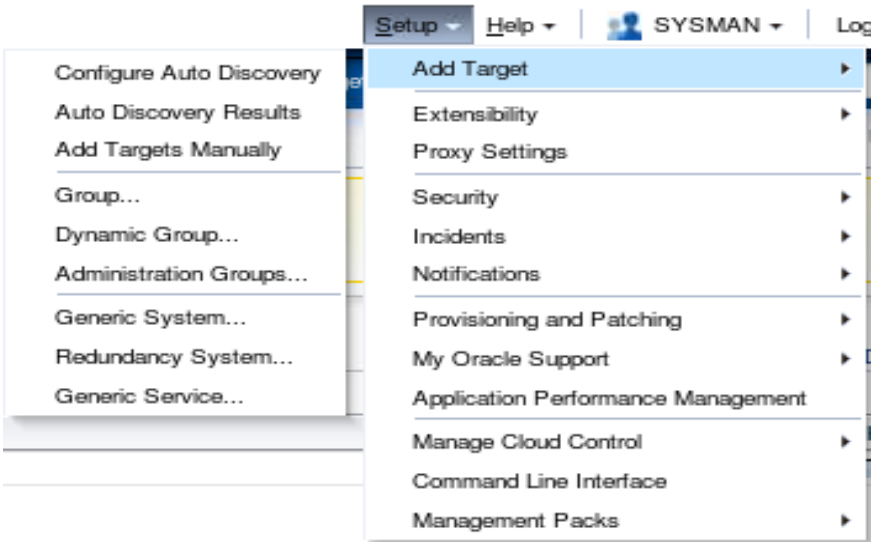
Assumptions

As a prerequisite task, you must log in to Enterprise Manager Cloud Control as the `sysman` user with `Oracle123` as the password.

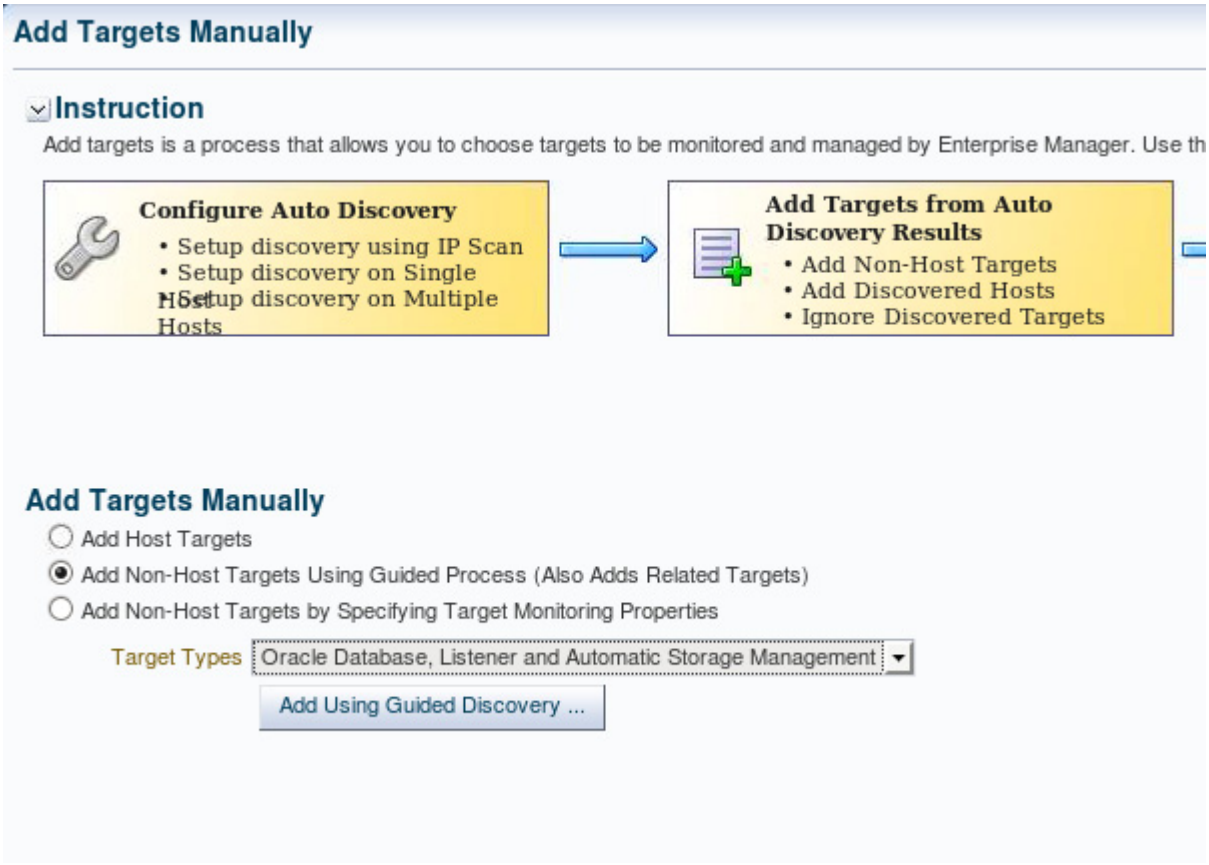
Tasks

First add the multitenant container database instance `cdb1` as a new target monitored by Oracle Enterprise Manager Cloud Control.

1. Add the `cdb1` Database Instance as a new target in Enterprise Manager Cloud Control.
   a. On the top right corner of Enterprise Manager, select "**Setup**" > "**Add Target**" > "**Add Targets Manually**."



   b. Under "**Add Targets Manually**," select "**Add Non-Host Targets Using Guided Process** (Also Adds Related Targets)." Then under "**Target Types**," select "**Oracle Database, Listener and Automatic Storage Management**". Click the "**Add Using Guided Discovery …**" button.



c. In "**Add Database Instance target: Specify Host**," click the magnifying glass to find your host. Select your host, and then click "**Continue**."

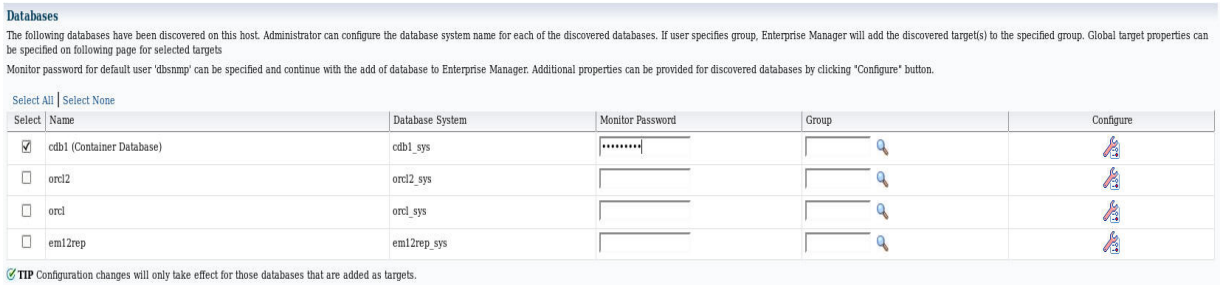
d. In the "**Databases**" list, uncheck all databases except `cdb1`. Uncheck the listener.

1) Unlock the `DBSNMP` user. This user is the monitoring user used to test the connection once the target is being added. Open a terminal window.

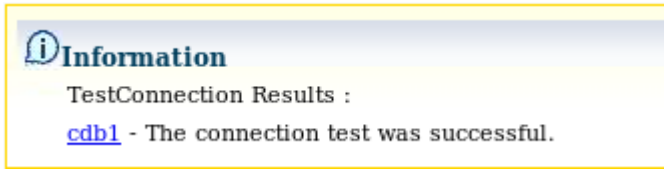
```
$ . oraenv
ORACLE_SID = [oracle] ? cdb1
The Oracle base for
ORACLE_HOME=/u01/app/oracle/product/12.1.0/dbhome_1 is
/u01/app/oracle
$ sqlplus / as sysdba
```

```
SQL*Plus: Release 12.1.0.1.0 Production on Thu Oct 25 05:12:01
2012
Copyright (c) 1982, 2012, Oracle.  All rights reserved.


Connected to:
Oracle Database 12c Enterprise Edition Release 12.1.0.1.0 -
64bit Production
With the Partitioning, OLAP, Advanced Analytics and Real
Application Testing options

SQL> alter user dbsnmp identified by oracle_4U account unlock;

User altered.

SQL> EXIT
Disconnected from Oracle Database 12c Enterprise Edition Release
12.1.0.1.0 - 64bit Production
With the Partitioning, OLAP, Advanced Analytics and Real
Application Testing options
$
```

2) Back to the Enterprise Manager page, enter `oracle_4U` for the "Monitor Password."



e. Click the "**Test Connection**" button. You should receive the following message:



f. Click "**Finish**" and then click "**Save**" to complete the operation, and finally click "**OK**."

Practice 1-3: Creating New Named Credentials

Overview

In this practice, you create the `credcdb1` credential used for any connection as `SYS` user sharable in the multitenant container database instance `cdb1`.

Assumptions

You completed the practice 1-2 to add the `cdb1` multitenant container database instance as a new target monitored by Enterprise Manager Cloud Control.

Tasks

1. Navigate to **Setup** > **Security** > **Named Credentials**.
2. Click **Create**.
   a. Enter the following values, then complete the **Access Control** section:

Field	Choice or Value
General Properties	
Credential Name	**credcdb1**
Credential description	**Credentials for Database**
Authenticating Target Type	**Database Instance**
Credential type	**Database Credentials**
Scope	**Target**
Target type	**Database Instance**
Target Name	**cdb1 (Click the magnifying glass to find cdb1 and select)**
Credential Properties	
Username	**SYS**
Password	**oracle_4U**
Confirm Password	**oracle_4U**
Role	**SYSDBA**

   b. Specify who can share, edit, or delete this shared credential using one of the three privileges (Full, Edit, View).
      - `SYS` user with Full privilege will be able to use, edit, and delete the credential.
      - `SYSTEM` user with Edit privilege will be able to use and edit the credential.
   c. Click “Add Grant” and then select the user `SYS` to be added in the Access Control list.
   d. Repeat this operation to add the user `SYSTEM`.
      By default, the selected users are granted only View privilege.
   e. To grant Full privilege to `SYS`, select the SYS user and click “Change Privilege.”
      Choose Full and click OK.

f. To grant Edit privilege to `SYSTEM`, select the `SYSTEM` user and click "Change Privilege." Choose Edit and click OK.

3. Test against the `cdb1` multitenant container database instance, click "**Test and Save**" until you get the following message: "**Confirmation Credential Operation Successful**." This means that the credential was successful and saved.

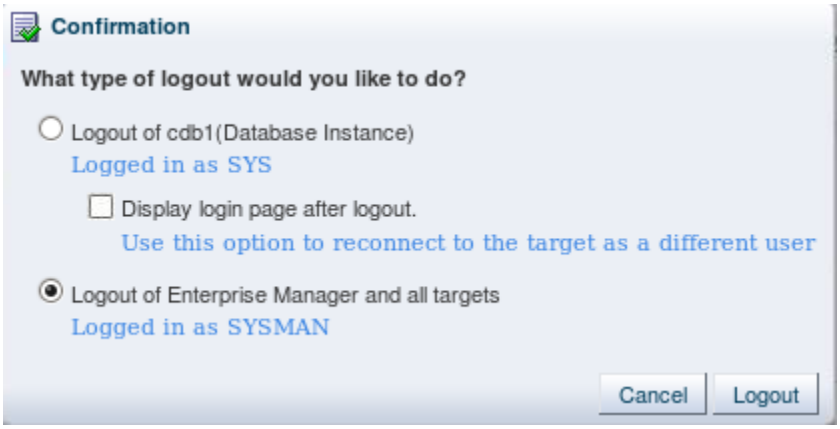
Practice 1-4: Testing the Named Credential

Overview

In this practice, you test the `credcdb1` named credential to connect to `cdb1` multitenant container database.

Tasks

1. Test if the named credential works when you connect to the `cdb1` target.
   Click **Targets** and then select **Databases**.
2. Choose `cdb1`. Click on `cdb1`.
3. Click **Administration**, then **Storage**, and then **Tablespaces.** The named credential `credcdb1` is displayed.
4. Click **Login** if you accept this named credential to log in to the `cdb1` multitenant container database; otherwise choose **New** to define new login username and password.
5. When you click Logout, the following screenshot gets displayed.



Choose "**Logout of Enterprise Manager and all targets**" and click **Logout**.

Practices for Lesson 2: Basics of Multitenant Container Database and Pluggable Databases

Chapter 2

Practices for Lesson 2: Overview

Practices Overview

In this practice, you will explore and get familiar with the architecture and structures of CDBs and PDBs.

Practice 2-1: Exploring CDB Architecture and Structures

Overview

In this practice, you will explore the architecture and structures of `cdb1` and its pluggable databases.

Tasks

1. Explore the `cdb1` instance, the background processes, and the multitenant container database.
   a. Use the **`ps -ef|grep`** Unix command.

```
$ ps -ef|grep cdb1
oracle   19997     1  0 03:43 ?        00:00:02 ora_pmon_cdb1
oracle   19999     1  0 03:43 ?        00:00:01 ora_psp0_cdb1
oracle   20003     1  0 03:43 ?        00:01:34 ora_vktm_cdb1
oracle   20007     1  0 03:43 ?        00:00:05 ora_gen0_cdb1
oracle   20009     1  0 03:43 ?        00:00:00 ora_mman_cdb1
oracle   20013     1  0 03:43 ?        00:00:00 ora_diag_cdb1
oracle   20015     1  0 03:43 ?        00:00:00 ora_ofsd_cdb1
oracle   20017     1  0 03:43 ?        00:00:00 ora_dbrm_cdb1
oracle   20019     1  0 03:43 ?        00:00:11 ora_dia0_cdb1
oracle   20021     1  0 03:43 ?        00:00:03 ora_dbw0_cdb1
oracle   20023     1  0 03:43 ?        00:00:01 ora_lgwr_cdb1
oracle   20025     1  0 03:43 ?        00:00:03 ora_ckpt_cdb1
oracle   20027     1  0 03:43 ?        00:00:02 ora_lg00_cdb1
oracle   20029     1  0 03:43 ?        00:00:00 ora_lg01_cdb1
oracle   20031     1  0 03:43 ?        00:00:00 ora_smon_cdb1
oracle   20033     1  0 03:43 ?        00:00:00 ora_reco_cdb1
oracle   20035     1  0 03:43 ?        00:00:00 ora_lreg_cdb1
oracle   20037     1  0 03:43 ?        00:00:12 ora_mmon_cdb1
oracle   20040     1  0 03:43 ?        00:00:06 ora_mmnl_cdb1
oracle   20042     1  0 03:43 ?        00:00:00 ora_d000_cdb1
oracle   20044     1  0 03:43 ?        00:00:05 ora_s000_cdb1
oracle   20071     1  0 03:44 ?        00:00:00 ora_tmon_cdb1
oracle   20074     1  0 03:44 ?        00:00:00 ora_tt00_cdb1
oracle   20076     1  0 03:44 ?        00:00:00 ora_smco_cdb1
oracle   20078     1  0 03:44 ?        00:00:00 ora_fbda_cdb1
oracle   20080     1  0 03:44 ?        00:00:03 ora_w000_cdb1
oracle   20082     1  0 03:44 ?        00:00:00 ora_aqpc_cdb1
oracle   20094     1  0 03:44 ?        00:00:34 ora_p000_cdb1
oracle   20096     1  0 03:44 ?        00:00:37 ora_p001_cdb1
oracle   20098     1  0 03:44 ?        00:00:36 ora_p002_cdb1
oracle   20100     1  0 03:44 ?        00:02:40 ora_p003_cdb1
oracle   20102     1  0 03:44 ?        00:00:01 ora_p004_cdb1
```

```
oracle   20104     1  0 03:44 ?        00:01:04 ora_p005_cdb1
oracle   20106     1  0 03:44 ?        00:00:02 ora_p006_cdb1
oracle   20108     1  0 03:44 ?        00:00:02 ora_p007_cdb1
oracle   20110     1  0 03:44 ?        00:00:00 ora_qm02_cdb1
oracle   20114     1  0 03:44 ?        00:00:00 ora_q002_cdb1
oracle   20116     1  0 03:44 ?        00:00:00 ora_q003_cdb1
oracle   20172     1  0 03:44 ?        00:00:09 ora_cjq0_cdb1
oracle   21060     1  0 03:51 ?        00:00:00 ora_w001_cdb1
oracle   21095     1  0 03:51 ?        00:00:00 ora_w002_cdb1
oracle   21173     1  0 03:52 ?        00:00:00 ora_w003_cdb1
oracle   21207     1  0 03:53 ?        00:00:00 ora_w004_cdb1
oracle   21212     1  0 03:53 ?        00:00:00 ora_w005_cdb1
oracle   21216     1  0 03:53 ?        00:00:00 ora_w006_cdb1
oracle   21350     1  0 03:54 ?        00:00:00 ora_w007_cdb1
oracle   21632     1  0 03:56 ?        00:00:00 ora_w008_cdb1
oracle   21651     1  0 03:56 ?        00:00:00 ora_w009_cdb1
oracle   23483     1  0 08:39 ?        00:00:00 ora_p008_cdb1
oracle   23485     1  0 08:39 ?        00:00:00 ora_p009_cdb1
oracle   23487     1  0 08:39 ?        00:00:00 ora_p00a_cdb1
oracle   23489     1  0 08:39 ?        00:00:00 ora_p00b_cdb1
$
```

b. Connect to the multitenant container database `cdb1`.

```
$ . oraenv
ORACLE_SID = [oracle] ? cdb1
The Oracle base has been set to /u01/app/oracle

$ sqlplus / as sysdba

SQL*Plus: Release 12.1.0.1.0 Production on Thu Oct 25 08:44:45
2012

Copyright (c) 1982, 2012, Oracle.  All rights reserved.



Connected to:
Oracle Database 12c Enterprise Edition Release 12.1.0.1.0 -
64bit Production
With the Partitioning, OLAP, Advanced Analytics and Real
Application Testing options


SQL>
```

c. Check if the database is a multitenant container database.

```
SQL> select name, cdb, con_id from v$database;

NAME      CDB     CON_ID
--------- --- ----------
CDB1      YES          0

SQL>
```

d. Check the instance name.

```
SQL> select INSTANCE_NAME, STATUS, CON_ID from v$instance;

INSTANCE_NAME    STATUS           CON_ID
---------------- ------------ ----------
cdb1             OPEN                  0

SQL> EXIT
$
```

2. Explore the services.
   a. Check if the listener is started. If it is not yet started, use the following command to start the listener:

```
$ lsnrctl status

LSNRCTL for Linux: Version 12.1.0.1.0 - Production on 25-OCT-
2012 09:37:00

Copyright (c) 1991, 2012, Oracle.  All rights reserved.

Connecting to
(DESCRIPTION=(ADDRESS=(PROTOCOL=IPC)(KEY=EXTPROC1521)))
STATUS of the LISTENER
------------------------
Alias                     LISTENER
Version                   TNSLSNR for Linux: Version 12.1.0.1.0
- Production
Start Date                24-OCT-2012 10:52:22
Uptime                    0 days 2 hr. 42 min. 19 sec
Trace Level               off
Security                  ON: Local OS Authentication
SNMP                      OFF
Listener Parameter File
/u01/app/oracle/product/12.1.0/dbhome_1/network/admin/listener.o
ra
Listener Log File
/u01/app/oracle/diag/tnslsnr/yourserver/listener/alert/log.xml
```

```
Listening Endpoints Summary...
  (DESCRIPTION=(ADDRESS=(PROTOCOL=ipc)(KEY=EXTPROC1521)))
(DESCRIPTION=(ADDRESS=(PROTOCOL=tcp)(HOST=yourserver)(PORT=1521)
))
(DESCRIPTION=(ADDRESS=(PROTOCOL=tcp)(HOST=yourserver)(PORT=5500)
)(Presentation=HTTP)(Session=RAW))
(DESCRIPTION=(ADDRESS=(PROTOCOL=tcp)(HOST=yourserver)(PORT=5501)
)(Presentation=HTTP)(Session=RAW))
(DESCRIPTION=(ADDRESS=(PROTOCOL=tcp)(HOST=yourserver)(PORT=5502)
)(Presentation=HTTP)(Session=RAW))
Services Summary...
Service "cdb1" has 1 instance(s).
  Instance "cdb1", status READY, has 1 handler(s) for this
service...
Service "cdb1XDB" has 1 instance(s).
  Instance "cdb1", status READY, has 1 handler(s) for this
service...
Service "em12rep" has 1 instance(s).
  Instance "em12rep", status READY, has 1 handler(s) for this
service...
Service "em12repXDB" has 1 instance(s).
  Instance "em12rep", status READY, has 1 handler(s) for this
service...
Service "orcl" has 1 instance(s).
  Instance "orcl", status READY, has 1 handler(s) for this
service...
Service "orcl2" has 1 instance(s).
  Instance "orcl2", status READY, has 1 handler(s) for this
service...
Service "orcl2XDB" has 1 instance(s).
  Instance "orcl2", status READY, has 1 handler(s) for this
service...
Service "orclXDB" has 1 instance(s).
  Instance "orcl", status READY, has 1 handler(s) for this
service...
Service "pdb1_1" has 1 instance(s).
  Instance "cdb1", status READY, has 1 handler(s) for this
service...
The command completed successfully
$
```

The listener is already started. If it were not started, you would use the following command to start the listener:

```
$ lsnrctl start

LSNRCTL for Linux: Version 12.1.0.1.0 - Production on 25-OCT-
2012 09:37:00
```

```
Copyright (c) 1991, 2012, Oracle.  All rights reserved.

Starting /u01/app/oracle/product/12.1.0/dbhome_1/bin/tnslsnr:
please wait...

LSNRCTL for Linux: Version 12.1.0.1.0 - Production on 25-OCT-
2012 09:37:00
System parameter file is
/u01/app/oracle/product/12.1.0/dbhome_1/network/admin/listener.o
ra
Log messages written to
/u01/app/oracle/diag/tnslsnr/yourserver/listener/alert/log.xml
Listening on:
(DESCRIPTION=(ADDRESS=(PROTOCOL=ipc)(KEY=EXTPROC1521)))
Listening on:
(DESCRIPTION=(ADDRESS=(PROTOCOL=tcp)(HOST=yourserver)(PORT=1521)
))

Connecting to
(DESCRIPTION=(ADDRESS=(PROTOCOL=IPC)(KEY=EXTPROC1521)))
STATUS of the LISTENER
------------------------
Alias                     LISTENER
Version                   TNSLSNR for Linux: Version 12.1.0.1.0
- Production
Start Date                24-OCT-2012 10:54:22
Uptime                    0 days 0 hr. 0 min. 0 sec
Trace Level               off
Security                  ON: Local OS Authentication
SNMP                      OFF
Listener Parameter File
/u01/app/oracle/product/12.1.0/dbhome_1/network/admin/listener.o
ra
Listener Log File
/u01/app/oracle/diag/tnslsnr/yourserver/listener/alert/log.xml
Listening Endpoints Summary...
  (DESCRIPTION=(ADDRESS=(PROTOCOL=ipc)(KEY=EXTPROC1521)))

(DESCRIPTION=(ADDRESS=(PROTOCOL=tcp)(HOST=yourserver)(PORT=1521)
))
The listener supports no services
The command completed successfully
$
```

b. Check services.

```
$ lsnrctl services

LSNRCTL for Linux: Version 12.1.0.1.0 - Production on 25-OCT-
2012 08:47:20

Copyright (c) 1991, 2012, Oracle.  All rights reserved.

Connecting to
(DESCRIPTION=(ADDRESS=(PROTOCOL=IPC)(KEY=EXTPROC1521)))
Services Summary...
Service "cdb1" has 1 instance(s).
  Instance "cdb1", status READY, has 1 handler(s) for this
service...
    Handler(s):
      "DEDICATED" established:2 refused:0 state:ready
         LOCAL SERVER
Service "cdb1XDB" has 1 instance(s).
  Instance "cdb1", status READY, has 1 handler(s) for this
service...
    Handler(s):
      "D000" established:0 refused:0 current:0 max:1022
state:ready
         DISPATCHER <machine: yourserver, pid: 27840>
         (ADDRESS=(PROTOCOL=tcp)(HOST=yourserver)(PORT=29863))
Service "em12rep" has 1 instance(s).
  Instance "em12rep", status READY, has 1 handler(s) for this
service...
    Handler(s):
      "DEDICATED" established:2748 refused:0 state:ready
         LOCAL SERVER
Service "em12repXDB" has 1 instance(s).
  Instance "em12rep", status READY, has 1 handler(s) for this
service...
    Handler(s):
      "D000" established:0 refused:0 current:0 max:1022
state:ready
         DISPATCHER <machine: yourserver, pid: 18255>
         (ADDRESS=(PROTOCOL=tcp)(HOST=yourserver)(PORT=59402))
Service "orcl" has 1 instance(s).
  Instance "orcl", status READY, has 1 handler(s) for this
service...
    Handler(s):
      "DEDICATED" established:251 refused:0 state:ready
         LOCAL SERVER
```

```
Service "orcl2" has 1 instance(s).
  Instance "orcl2", status READY, has 1 handler(s) for this
service...
    Handler(s):
      "DEDICATED" established:4 refused:0 state:ready
         LOCAL SERVER
Service "orcl2XDB" has 1 instance(s).
  Instance "orcl2", status READY, has 1 handler(s) for this
service...
    Handler(s):
      "D000" established:6 refused:0 current:0 max:1022
state:ready
         DISPATCHER <machine: yourserver, pid: 23615>
         (ADDRESS=(PROTOCOL=tcp)(HOST=yourserver)(PORT=50200))
Service "orclXDB" has 1 instance(s).
  Instance "orcl", status READY, has 1 handler(s) for this
service...
    Handler(s):
      "D000" established:0 refused:0 current:0 max:1022
state:ready
         DISPATCHER <machine: yourserver, pid: 30821>
         (ADDRESS=(PROTOCOL=tcp)(HOST=yourserver)(PORT=27384))
Service "pdb1_1" has 1 instance(s).
  Instance "cdb1", status READY, has 1 handler(s) for this
service...
    Handler(s):
      "DEDICATED" established:2 refused:0 state:ready
         LOCAL SERVER
The command completed successfully
$
```

c. List the services automatically created for each container.

```
$ sqlplus / as sysdba

SQL*Plus: Release 12.1.0.1.0 Production on Thu Oct 25 09:39:00
2012

Copyright (c) 1982, 2012, Oracle.  All rights reserved.



Connected to:
Oracle Database 12c Enterprise Edition Release 12.1.0.1.0 -
64bit Production
With the Partitioning, OLAP, Advanced Analytics and Real
Application Testing options
```

```
SQL> col name format A20
SQL> select name, con_id from v$services;

NAME                      CON_ID
-------------------- ----------
pdb1_1                         3
cdb1XDB                        1
cdb1                           1
SYS$BACKGROUND                 1
SYS$USERS                      1

SQL>
```

Notice that the PDB$SEED service is not listed. The user should not connect to this service as there should not be any operation performed on this container, which is reserved as a template to create other PDBs.

3. Display the pluggable databases.
   a. Use the new view V$PDBS.

```
SQL> select CON_ID, NAME, OPEN_MODE from v$pdbs;

    CON_ID NAME                           OPEN_MODE
---------- ------------------------------ ----------
         2 PDB$SEED                       READ ONLY
         3 PDB1_1                         READ WRITE

SQL>
```

Notice that the seed PDB is in READ ONLY open mode.

   b. Use the new command SHOW CON_NAME and CON_ID to know which container you are connected to.

```
SQL> show con_name

CON_NAME
------------------------------
CDB$ROOT

SQL> show con_id

CON_ID
------------------------------
1

SQL>
```

You can also use the SYS_CONTEXT function to view the CON_NAME and CON_ID attributes of your session context.

```
SELECT sys_context('userenv','CON_NAME') from dual;
SELECT sys_context('userenv','CON_ID') from dual;
```

4. View some of the new family of views CDB_*xxx*:

```
SQL> col PDB_NAME format a8
SQL> col CON_ID format 99
SQL> select PDB_ID, PDB_NAME, DBID, GUID, CON_ID
  2  from cdb_pdbs;

PDB_ID PDB_NAME       DBID GUID
CON_ID
------ -------- ---------- --------------------------------    -
-----
     3 PDB1_1   3624951709 C39AE2177B2E530EE043160200C043C3
     1
     2 PDB$SEED 4029862422 C2CBF2921BD1161FE04388AAE80AB141
     1


SQL>
```

The PDB_ID number 2 is always assigned to the seed PDB because it is the second container to be created after the root container (CON_ID 1).

5. Check all the files of the CDB.
   a. View the redo log files of the CDB.

```
SQL> col MEMBER format A40
SQL> select GROUP#, CON_ID, MEMBER from v$logfile;


    GROUP# CON_ID MEMBER
---------- ------ ----------------------------------------
         3      0 /u01/app/oracle/oradata/cdb1/redo03.log
         2      0 /u01/app/oracle/oradata/cdb1/redo02.log
         1      0 /u01/app/oracle/oradata/cdb1/redo01.log


SQL>
```

The CON_ID value 0 refers to the whole multitenant container database.

   b. View the control files of the CDB.

```
SQL> col NAME format A60
SQL> select NAME , CON_ID from v$controlfile;


NAME                                                  CON_ID
----------------------------------------------------- ------
/u01/app/oracle/oradata/cdb1/control01.ctl                 0
```

```
/u01/app/oracle/fast_recovery_area/cdb1/control02.ctl       0

SQL>
```

c. View all the data files of the CDB, including those of the root and all PDBs.

1) With CDB_DATA_FILES view:

```
SQL> col file_name format A50
SQL> col tablespace_name format A8
SQL> col file_id format 9999
SQL> col con_id  format 999
SQL> select FILE_NAME, TABLESPACE_NAME, FILE_ID, con_id
  2  from   cdb_data_files  order  by con_id ;

FILE_NAME                                          TABLESPA
FILE_ID CON_ID
-------------------------------------------------- -------- ----
--- ------
/u01/app/oracle/oradata/cdb1/users01.dbf           USERS
6      1
/u01/app/oracle/oradata/cdb1/undotbs01.dbf         UNDOTBS1
4      1
/u01/app/oracle/oradata/cdb1/sysaux01.dbf          SYSAUX
3      1
/u01/app/oracle/oradata/cdb1/system01.dbf          SYSTEM
1      1
/u01/app/oracle/oradata/cdb1/pdbseed/system01.dbf  SYSTEM
5      2
/u01/app/oracle/oradata/cdb1/pdbseed/sysaux01.dbf  SYSAUX
7      2
/u01/app/oracle/oradata/pdb1_1/system01.dbf   SYSTEM
8      3
/u01/app/oracle/oradata/pdb1_1/sysaux01.dbf   SYSAUX
9      3
/u01/app/oracle/oradata/pdb1_1/SAMPLE_SCHEMA_ USERS
10     3
users01.dbf
/u01/app/oracle/oradata/pdb1_1/example01.dbf  EXAMPLE
11     3

10 rows selected.

SQL>
```

2) With the **ls** Unix command:

```
SQL> !ls -l $ORACLE_BASE/oradata/cdb1
```

```
total 2575988
-rw-r----- 1 oracle oinstall  17874944 Oct 25 23:38
control01.ctl
drwxr-xr-x 2 oracle oinstall      4096 Oct 25 10:54 pdb1_1
drwxr-x--- 2 oracle oinstall      4096 Oct 25 10:37 pdbseed
-rw-r----- 1 oracle oinstall  52429312 Oct 25 23:38 redo01.log
-rw-r----- 1 oracle oinstall  52429312 Oct 25 20:01 redo02.log
-rw-r----- 1 oracle oinstall  52429312 Oct 25 22:23 redo03.log
-rw-r----- 1 oracle oinstall 849354752 Oct 25 23:35 sysaux01.dbf
-rw-r----- 1 oracle oinstall 828383232 Oct 25 23:35 system01.dbf
-rw-r----- 1 oracle oinstall 571482112 Oct 25 23:18 temp01.dbf
-rw-r----- 1 oracle oinstall 246423552 Oct 25 23:36
undotbs01.dbf
-rw-r----- 1 oracle oinstall   5251072 Oct 25 22:29 users01.dbf

SQL>
```

```
SQL> !ls -l $ORACLE_BASE/oradata/cdb1/pdbseed
total 985064
-rw-r----- 1 oracle oinstall  88088576 Oct 25 10:50
pdbseed_temp01.dbf
-rw-r----- 1 oracle oinstall 671096832 Oct 25 10:50 sysaux01.dbf
-rw-r----- 1 oracle oinstall 262152192 Oct 25 10:50 system01.dbf

SQL>
```

There are only the SYSTEM and SYSAUX datafiles and a tempfile for the seed PDB.

d. Ensure that you are connected to the root; then use the DBA_DATA_FILES view.

```
SQL> col file_name format A42
SQL> select FILE_NAME, TABLESPACE_NAME, FILE_ID
  2  from   dba_data_files;

FILE_NAME                                  TABLESPA FILE_ID
------------------------------------------ -------- -------
/u01/app/oracle/oradata/cdb1/users01.dbf   USERS           6
/u01/app/oracle/oradata/cdb1/undotbs01.dbf UNDOTBS1        4
/u01/app/oracle/oradata/cdb1/sysaux01.dbf  SYSAUX          3
/u01/app/oracle/oradata/cdb1/system01.dbf  SYSTEM          1

SQL>
```

Notice that only the root datafiles are listed.

e. Now use the V$TABLESPACE and V$DATAFILE view.

```
SQL> col NAME format A12
SQL> select FILE#, ts.name, ts.ts#, ts.con_id
```

```
  2  from v$datafile d, v$tablespace  ts
  3  where d.ts#=ts.ts#
  4  and   d.con_id=ts.con_id
  5  order by 4,3;

     FILE# NAME                 TS# CON_ID
---------- ------------ ---------- ------
         1 SYSTEM                0      1
         3 SYSAUX                1      1
         4 UNDOTBS1              2      1
         6 USERS                 4      1
         5 SYSTEM                0      2
         7 SYSAUX                1      2
         8 SYSTEM                0      3
         9 SYSAUX                1      3
        10 USERS                 3      3
        11 EXAMPLE               4      3

10 rows selected.

SQL>
```

f. List the tempfiles of the CDB.

```
SQL> col file_name format A57
SQL> select FILE_NAME, TABLESPACE_NAME, FILE_ID
  2  from cdb_temp_files;

FILE_NAME                                                 TABLESPA
FILE_ID
---------------------------------------------------------
-------- -------
/u01/app/oracle/oradata/cdb1/temp01.dbf
TEMP       1
/u01/app/oracle/oradata/pdb1_1/pdb1_1_temp01.dbf
TEMP            3
/u01/app/oracle/oradata/cdb1/pdbseed/pdbseed_temp01.dbf
TEMP       2

SQL>
```

6. List all the users created.
   a. Verify that the `SYSTEM` user is created.

```
SQL> col username format A22
```

```
SQL> select username, common, con_id from cdb_users
  2  where username ='SYSTEM';

USERNAME                       COM CON_ID
------------------------------ --- ------
SYSTEM                         YES      1
SYSTEM                         YES      3
SYSTEM                         YES      2

SQL>
```

Notice that the user `SYSTEM` exists in all containers as a common user.

b. List all the common users of the CDB.

```
SQL> select distinct username from cdb_users
  2  where common ='YES'
  3  ORDER BY 1;

USERNAME
--------------------
ANONYMOUS
APEX_040200
APEX_PUBLIC_USER
APPQOSSYS
AUDSYS
CTXSYS
DBSNMP
DIP
DVF
DVSYS
FLOWS_FILES
GSMADMIN_INTERNAL
GSMCATUSER
GSMUSER
LBACSYS
MDDATA
MDSYS
OJVMSYS
OLAPSYS
ORACLE_OCM
ORDDATA
ORDPLUGINS
ORDSYS
OUTLN
```

```
SI_INFORMTN_SCHEMA
SPATIAL_CSW_ADMIN_USR
SPATIAL_WFS_ADMIN_USR
SYS
SYSBACKUP
SYSDG
SYSKM
SYSTEM
WMSYS
XDB
XS$NULL

35 rows selected.

SQL>
```

c. List all the local users of the CDB.

```
SQL> select distinct username, con_id from cdb_users
  2  where common ='NO';

USERNAME             CON_ID
-------------------- ------
SH                        3
BI                        3
IX                        3
PDBADMIN                  3
HR                        3
OE                        3
SCOTT                     3
PM                        3

8 rows selected.

SQL>
```

d. List the local users in the root.

```
SQL> select username, con_id from cdb_users
  2  where common ='NO';

USERNAME             CON_ID
-------------------- ------
PDBADMIN                  3
HR                        3
OE                        3
```

```
SH                             3
IX                             3
PM                             3
BI                             3
SCOTT                          3

8 rows selected.

SQL>
```

Notice that there is no local user in the root container because it is impossible to create any local user in the root.

7. List all the roles and privileges of the CDB.
   a. List all the roles of the CDB.

```
SQL> col role format A30
SQL> select role, common, con_id from cdb_roles order by 3;

ROLE                           COM CON_ID
------------------------------ --- ------
CONNECT                         YES      1
DV_REALM_OWNER                  YES      1
DV_REALM_RESOURCE               YES      1
DV_DATAPUMP_NETWORK_LINK        YES      1
DV_AUDIT_CLEANUP                YES      1
DV_GOLDENGATE_REDO_ACCESS        YES     1
DV_XSTREAM_ADMIN                YES      1
DV_GOLDENGATE_ADMIN             YES      1
DV_STREAMS_ADMIN                YES      1
DV_PATCH_ADMIN                  YES      1
…
PDB_DBA                         YES      1
CDB_DBA                         YES      1
IMP_FULL_DATABASE               YES      1
EXP_FULL_DATABASE               YES      1
…
DV_XSTREAM_ADMIN                YES      2
DV_GOLDENGATE_REDO_ACCESS       Yes      2
DV_AUDIT_CLEANUP                YES      2
DV_REALM_OWNER                  YES      2
…
CDB_DBA                         YES      2
PDB_DBA                         YES      2
```

```
…
DV_GOLDENGATE_ADMIN              YES      3
DV_XSTREAM_ADMIN                 YES      3
DV_GOLDENGATE_REDO_ACCESS        YES      3
DV_AUDIT_CLEANUP                 YES      3
DV_DATAPUMP_NETWORK_LINK         YES      3
DV_REALM_RESOURCE                YES      3
DV_REALM_OWNER                   YES      3

252 rows selected.

SQL>
```

Notice that there is no local role in the root container because it is impossible to create any local role in the root.

b. Ensure that the privileges are neither common nor local by nature.

```
SQL> desc sys.system_privilege_map
Name                              Null?    Type
 ------------------------------ -------- ----------------------
 PRIVILEGE                        NOT NULL NUMBER
 NAME                             NOT NULL VARCHAR2(40)
 PROPERTY                         NOT NULL NUMBER

SQL> desc sys.table_privilege_map
Name                              Null?    Type
 ------------------------------ -------- ----------------------
 PRIVILEGE                        NOT NULL NUMBER
 NAME                             NOT NULL VARCHAR2(40)

SQL>
```

Notice that there is no COMMON column.

c. Verify that the privilege when granted becomes a common or local privilege.

```
SQL> desc CDB_SYS_PRIVS
Name                               Null?    Type
 ------------------------------- -------- --------------------
 GRANTEE                                   VARCHAR2(128)
 PRIVILEGE                                 VARCHAR2(40)
 ADMIN_OPTION                              VARCHAR2(3)
 COMMON                                    VARCHAR2(3)
 CON_ID                                    NUMBER

SQL> desc CDB_TAB_PRIVS
Name                               Null?    Type
```

```
------------------------------ -------- --------------------
 GRANTEE                                 VARCHAR2(128)
 OWNER                                   VARCHAR2(128)
 TABLE_NAME                              VARCHAR2(128)
 GRANTOR                                 VARCHAR2(128)
 PRIVILEGE                               VARCHAR2(40)
 GRANTABLE                               VARCHAR2(3)
 HIERARCHY                               VARCHAR2(3)
 COMMON                                  VARCHAR2(3)
 TYPE                                    VARCHAR2(24)
 CON_ID                                  NUMBER

SQL>
```

There is a **COMMON** column.

d. Notice that the role, though common or local depending on how the role was created, is also, like privileges, granted either commonly or locally.

```
SQL> col grantee format A10
SQL> col granted_role format A28
SQL> select grantee, granted_role, common, con_id
  2  from cdb_role_privs
  3  where grantee='SYSTEM';

GRANTEE    GRANTED_ROLE                 COM CON_ID
---------- ---------------------------- --- ------
SYSTEM     DBA                          YES      1
SYSTEM     AQ_ADMINISTRATOR_ROLE        YES      1
SYSTEM     DBA                          YES      2
SYSTEM     AQ_ADMINISTRATOR_ROLE        YES      2
SYSTEM     DBA                          YES      3
SYSTEM     AQ_ADMINISTRATOR_ROLE        YES      3

6 rows selected.

SQL> EXIT
$
```

Practices for Lesson 3: Creating a Multitenant Container Database and Pluggable Databases

Chapter 3

Practices for Lesson 3: Overview

Overview

In this practice, you will create a new CDB named `cdb2` with DBCA with no PDB except the mandatory seed PDB.

After the CDB creation is completed, check the physical and logical structures of the new CDB.

Then, you will create several PDBs using different methods:

- Create `pdb2_1` from seed in `cdb2` (using SQL*Plus first and then SQL Developer)
- Clone `pdb2_2` in `cdb2` from `pdb2_1` (using SQL*Plus first or SQL Developer)
- Plug the non-CDB `orcl2` into the CDB `cdb2` as `pdb_orcl2` (using SQL*Plus)
- Merge the two CDBs `cdb1` and `cdb2` into `cdb2`, and optionally drop the database `cdb1` (optional practice)

During these practices, you will exercise yourself to drop PDBs using DBCA, SQL Developer, or SQL*Plus.

Practice 3-1: Creating a New CDB

Overview

In this practice, you will create a new CDB named `cdb2` with DBCA.

Assumptions

The created CDB `cdb1` already exists.

Tasks

1. Create a CDB named `cdb2` using DBCA. First release resources held by other instances, shutting down the `orcl`, `orcl2` and `cdb1` instances.
   a. Shut down `orcl`.

```
$ . oraenv
ORACLE_SID = [cdb1] ? orcl
The Oracle base remains unchanged with value /u01/app/oracle
$ sqlplus / as sysdba

Connected to:
Oracle Database 12c Enterprise Edition Release 12.1.0.1.0 -
64bit Production
With the Partitioning, OLAP, Advanced Analytics and Real
Application Testing options

SQL> SHUTDOWN IMMEDIATE
Database closed.
Database dismounted.
ORACLE instance shut down.
SQL> EXIT
$
```

   b. Shut down `orcl2`.

```
$ . oraenv
ORACLE_SID = [orcl] ? orcl2
The Oracle base remains unchanged with value /u01/app/oracle

$ sqlplus / as sysdba

Connected to:
Oracle Database 12c Enterprise Edition Release 12.1.0.1.0 -
64bit Production
With the Partitioning, OLAP, Advanced Analytics and Real
Application Testing options

SQL> SHUTDOWN IMMEDIATE
```

```
Database closed.
Database dismounted.
ORACLE instance shut down.
SQL> EXIT
$
```

c. Shut down `cdb1`.

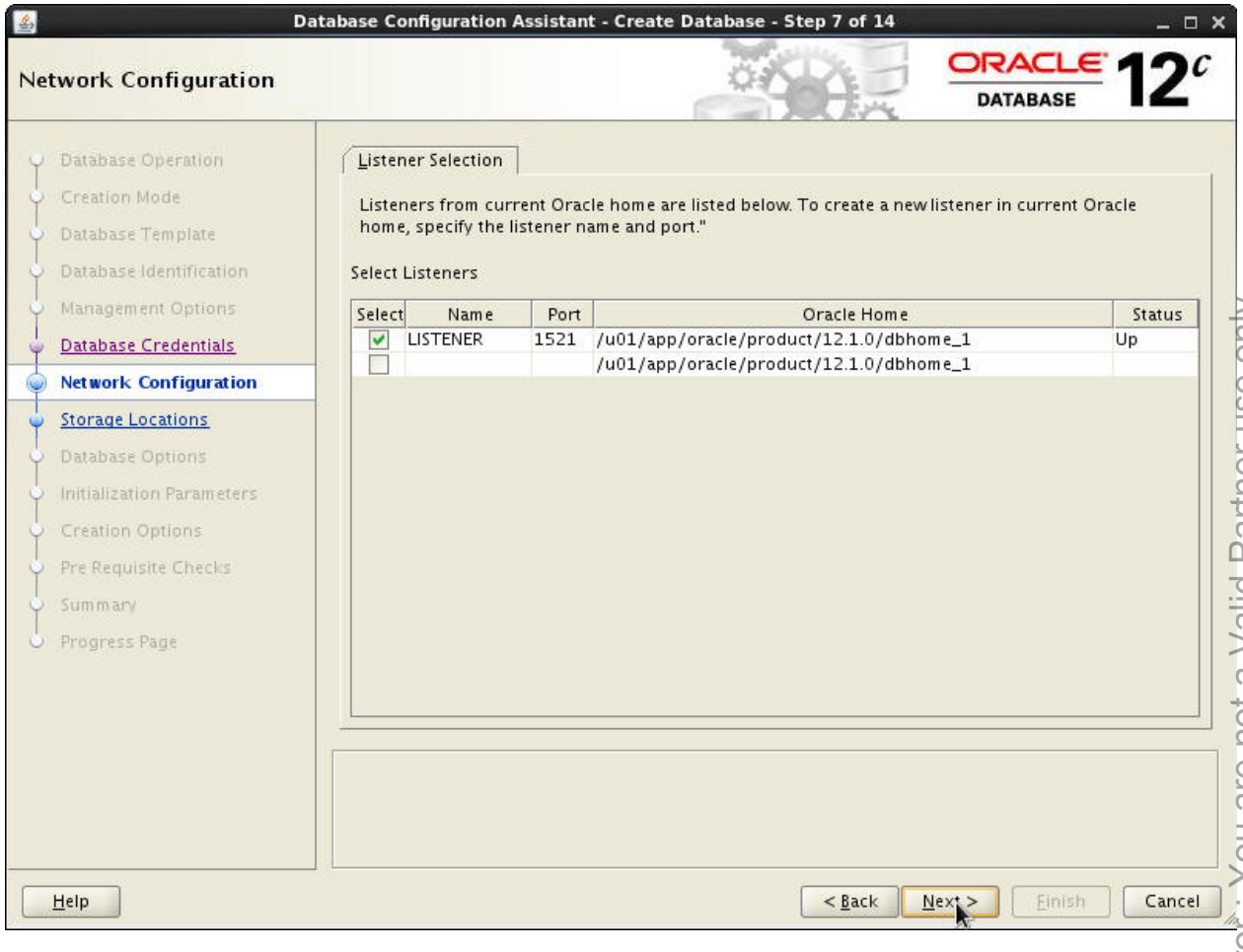
```
$ . oraenv
ORACLE_SID = [orcl2] ? cdb1
The Oracle base remains unchanged with value /u01/app/oracle

$ sqlplus / as sysdba

Connected to:
Oracle Database 12c Enterprise Edition Release 12.1.0.1.0 -
64bit Production
With the Partitioning, OLAP, Advanced Analytics and Real
Application Testing options

SQL> SHUTDOWN IMMEDIATE
Database closed.
Database dismounted.
ORACLE instance shut down.
SQL> EXIT
$
```

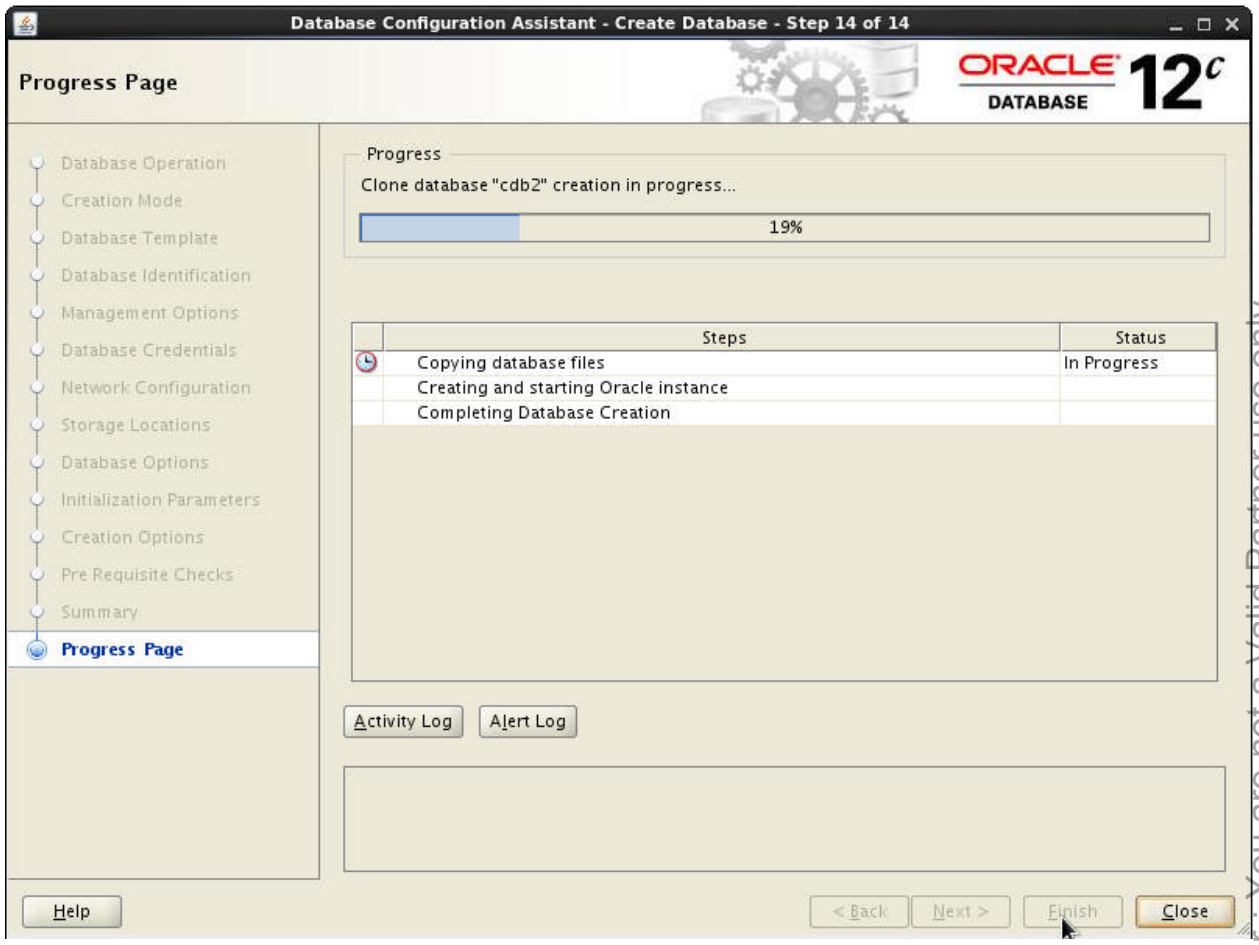
2. Start dbca and perform the following steps.

```
$ dbca
```

Step	Window/Page Description	Choices or Values
a.	Step 1: Database Operation	Select “Create Database.” Click Next.
b.	Step 2: Creation Mode	Select “Advanced Mode.” Click Next.
c.	Step 3: Database Template	Select “General Purpose or Transaction Processing.” Click Next.
d.	Step 4: Database Identification	Enter Global Database Name: `cdb2` SID: `cdb2` Select **“Create As Container Database.”** Select **“Create An Empty Container Database.”** Click Next.

Step	Window/Page Description	Choices or Values
e.	Step 5: Management Options	Deselect “Configure Enterprise Manager (EM) Database Express.” Click Next.
f.	Step 6: Database Credentials	Select “Use same Administrative password…” Enter: Password: `oracle_4U` Confirm password: `oracle_4U` Click Next.
g.	Step 7: Network Configuration	Click Next.
h.	Step 8: Storage Locations	Confirm Storage type is “File System.” Select “Use Common Location for All Database Files.” Click Next.
i.	Step 9: Database Options	Click Next.
j.	Step 10: Initialization Parameters	Select “Character Sets.” Select “Use Unicode (**AL32UTF8**).” Click Next.
k.	Step 11: Creation Option	Select “Create Database.” Click Next.
l.	Step 12: Pre Requisite Checks	Click Next.
m.	Step 13: Summary	Click Finish.
n.	Step 14: Progress Page	On the Database Configuration Assistant page (for password management), click Exit. Click Close.





Practice 3-2: Exploring CDB and PDB Structures

Overview

In this practice, you will check the physical and logical structures of the new CDB `cdb2` and its seed PDB.

Tasks

1. Connect to the multitenant container database `cdb2`.

```
$ . oraenv
ORACLE_SID = [cdb1] ? cdb2
The Oracle base remains unchanged with value /u01/app/oracle

$ sqlplus / as sysdba

Connected to:
Oracle Database 12c Enterprise Edition Release 12.1.0.1.0 -
64bit Production
With the Partitioning, OLAP, Advanced Analytics and Real
Application Testing options

SQL>
```

   a. Check if the database is a multitenant container database.

```
SQL> SELECT name, cdb, con_id from v$database;

NAME      CDB     CON_ID
--------- --- ----------
CDB2      YES          0

SQL>
```

   b. Check the instance name.

```
SQL> SELECT INSTANCE_NAME, STATUS, CON_ID from v$instance;

INSTANCE_NAME    STATUS           CON_ID
---------------- ------------ ----------
cdb2             OPEN                  0

SQL> EXIT
$
```

2. Explore the services.

   a. Check services.

```
$ lsnrctl status
```

```
LSNRCTL for Linux: Version 12.1.0.1.0 - Production on 26-OCT-
2012 10:35:55

Copyright (c) 1991, 2012, Oracle.  All rights reserved.

Connecting to
(DESCRIPTION=(ADDRESS=(PROTOCOL=IPC)(KEY=EXTPROC1521)))
STATUS of the LISTENER
------------------------
Alias                     LISTENER
Version                   TNSLSNR for Linux: Version 12.1.0.1.0
- Production
Start Date                25-OCT-2012 09:38:19
Uptime                    16 days 21 hr. 48 min. 35 sec
Trace Level               off
Security                  ON: Local OS Authentication
SNMP                      OFF
Listener Parameter File
/u01/app/oracle/product/12.1.0/dbhome_1/network/admin/listener.o
ra
Listener Log File
/u01/app/oracle/diag/tnslsnr/yourserver/listener/alert/log.xml
Listening Endpoints Summary...
  (DESCRIPTION=(ADDRESS=(PROTOCOL=ipc)(KEY=EXTPROC1521)))
(DESCRIPTION=(ADDRESS=(PROTOCOL=tcp)(HOST=yourserver)(PORT=1521)
))
Services Summary...
Service "cdb2" has 1 instance(s).
  Instance "cdb2", status READY, has 1 handler(s) for this
service...
Service "cdb2XDB" has 1 instance(s).
  Instance "cdb2", status READY, has 1 handler(s) for this
service...
Service "em12rep" has 1 instance(s).
  Instance "em12rep", status READY, has 1 handler(s) for this
service...
Service "em12repXDB" has 1 instance(s).
  Instance "em12rep", status READY, has 1 handler(s) for this
service...
The command completed successfully
$
```

b. List the services automatically created for each container.

```
$ sqlplus / as sysdba
```

```
Connected to:
Oracle Database 12c Enterprise Edition Release 12.1.0.1.0 -
64bit Production
With the Partitioning, OLAP, Advanced Analytics and Real
Application Testing options

SQL> col name format A20
SQL> SELECT name, con_id from v$services;

NAME                    CON_ID
-------------------- ----------
cdb2XDB                      1
cdb2                         1
SYS$BACKGROUND               1
SYS$USERS                    1

SQL>
```

Notice that `PDB$SEED` service is not listed. No user should connect to this service because there should be no operation performed on this container. It is reserved as a template to create other PDBs.

3. Display the pluggable databases. Use a new view `V$PDBS`.

```
SQL> SELECT CON_ID, NAME, OPEN_MODE from v$pdbs;

    CON_ID NAME                           OPEN_MODE
---------- ------------------------------ ----------
         2 PDB$SEED                       READ ONLY

SQL>
```

Notice that the seed PDB is in `READ ONLY` open mode.

4. View new family of views CDB_*xxx*:

```
SQL> col PDB_NAME format a8
SQL> col CON_ID format 999999
SQL> SELECT PDB_ID, PDB_NAME, DBID, GUID, CON_ID
  2  from cdb_pdbs order by 1;

PDB_ID PDB_NAME       DBID GUID                             CON_ID
------ -------- ---------- -------------------------------- ------
     2 PDB$SEED 4012275228 203F5F3EDB7F0000000000000000      1

SQL>
```

5. Check all files of the CDB.
   a. View the redo log files of the CDB.

```
SQL> col MEMBER format A42
SQL> SELECT GROUP#, MEMBER, CON_ID from v$logfile;

   GROUP# MEMBER                                      CON_ID
---------- ------------------------------------------ -------
         3 /u01/app/oracle/oradata/cdb2/redo03.log          0
         2 /u01/app/oracle/oradata/cdb2/redo02.log          0
         1 /u01/app/oracle/oradata/cdb2/redo01.log          0

SQL>
```

b. View the control files of the CDB.

```
SQL> col name format A55
SQL> SELECT name, con_id from v$controlfile;

NAME                                                   CON_ID
------------------------------------------------------ -------
/u01/app/oracle/oradata/cdb2/control01.ctl                   0
/u01/app/oracle/fast_recovery_area/cdb2/control02.ctl        0

SQL>
```

c. View all data files of the CDB, including those of the root and all PDBs, with `CDB_DATA_FILES` view.

```
SQL> col file_name format A65
SQL> SELECT FILE_NAME, TABLESPACE_NAME, FILE_ID, con_id
  2  from   cdb_data_files
  3  order  by con_id ;

FILE_NAME
----------------------------------------------------------------
-
TABLESPACE_NAME                   FILE_ID  CON_ID
------------------------------ ---------- -------
/u01/app/oracle/oradata/cdb2/users01.dbf
USERS                                   6       1

/u01/app/oracle/oradata/cdb2/undotbs01.dbf
UNDOTBS1                                4       1

/u01/app/oracle/oradata/cdb2/sysaux01.dbf
SYSAUX                                  3       1

/u01/app/oracle/oradata/cdb2/system01.dbf
```

```
SYSTEM                                      1       1

/u01/app/oracle/oradata/cdb2/pdbseed/system01.dbf
SYSTEM                                      5       2

/u01/app/oracle/oradata/cdb2/pdbseed/sysaux01.dbf
SYSAUX                                      7       2

6 rows selected.

SQL>
```

d. Ensure that you are still connected to the root; then use `DBA_DATA_FILES` view.

```
SQL> col file_name format A42
SQL> col tablespace_name format A10
SQL> SELECT FILE_NAME, TABLESPACE_NAME, FILE_ID
  2  from   dba_data_files;

FILE_NAME                                  TABLESPACE FILE_ID
------------------------------------------ ---------- -------
/u01/app/oracle/oradata/cdb2/users01.dbf   USERS             6
/u01/app/oracle/oradata/cdb2/undotbs01.dbf UNDOTBS1          4
/u01/app/oracle/oradata/cdb2/sysaux01.dbf  SYSAUX            3
/u01/app/oracle/oradata/cdb2/system01.dbf  SYSTEM            1

SQL> EXIT
$
```

Notice that only root data files are listed.

e. Start the `cdb1` database.

```
$ . oraenv
ORACLE_SID = [cdb2] ? cdb1
The Oracle base remains unchanged with value /u01/app/oracle

$ sqlplus / as sysdba

Connected to an idle instance.

SQL> STARTUP
ORACLE instance started.

Total System Global Area  400846848 bytes
Fixed Size                  2271568 bytes
Variable Size             243271344 bytes
```

```
Database Buffers       146800640 bytes
Redo Buffers             8503296 bytes
Database mounted.
Database opened.
SQL> EXIT
$
```

1) Use netca to add the `PDB1_1` net service name for `pdb1_1` pluggable database of `cdb1` in the `tnsnames.ora` file.

```
$ netca
```

2) On the Welcome page, select the "Local Net Service Name configuration" and click Next.
3) On the Net Service Name Configuration page, accept `Add` and click Next.
4) On the Net Service Name Configuration, Service Name page, enter `pdb1_1` as Service Name and click Next.
5) On the Net Service Name Configuration, Select Protocols page, select TCP and click Next.
6) On the Net Service Name Configuration, TCP/IP Protocol page, enter your complete host name, for example, <*yourservername*>, or *localhost,* accept "Use the standard port number of 1521," and click Next.
7) On the Net Service Name Configuration, Test page, select "No, do not test" (the pluggable database is not yet opened) and click Next.
8) On the Net Service Name Configuration, Net Service Name page, accept `pdb1_1` as Net Service Name and click Next.
9) On the Net Service Name Configuration, Another Net Service page, select No, and Next.
10) On the Net Service Name Configuration Complete page, click Next.
11) When you are back on the Welcome page, click Finish.

f. Open the `pdb1_1` pluggable database in `cdb1`.

```
$ sqlplus / as sysdba

Connected to:
Oracle Database 12c Enterprise Edition Release 12.1.0.1.0 -
64bit Production
With the Partitioning, OLAP, Advanced Analytics and Real
Application Testing options

SQL> ALTER PLUGGABLE DATABASE pdb1_1 OPEN;

Pluggable database altered.

SQL> EXIT
$
```

g. Connect to the `pdb1_1` of `cdb1`, and use `DBA_DATA_FILES` view.

```
$ sqlplus system/oracle_4U@pdb1_1

Connected to:
Oracle Database 12c Enterprise Edition Release 12.1.0.1.0 -
64bit Production
With the Partitioning, OLAP, Advanced Analytics and Real
Application Testing options

SQL> col file_name format A65
SQL> SELECT FILE_NAME, TABLESPACE_NAME, FILE_ID
  2  from   dba_data_files;

FILE_NAME
-----------------------------------------------------------------
TABLESPACE_NAME                   FILE_ID
------------------------------ ----------
/u01/app/oracle/oradata/pdb1_1/system01.dbf
SYSTEM                                  8
/u01/app/oracle/oradata/pdb1_1/sysaux01.dbf
SYSAUX                                  9
/u01/app/oracle/oradata/pdb1_1/SAMPLE_SCHEMA_users01.dbf
USERS                                  10
/u01/app/oracle/oradata/pdb1_1/example01.dbf
EXAMPLE                                11

SQL>
```

Notice that only `pdb1_1` data files are listed.

h. Now use `V$TABLESPACE` and `V$DATAFILE` view.

```
SQL> col NAME format A12
SQL> SELECT FILE#, ts.name, ts.ts#, ts.con_id
  2  from v$datafile d, v$tablespace  ts
  3  where d.ts#=ts.ts#
  4  and   d.con_id=ts.con_id
  5  order by 4;

     FILE# NAME                 TS#     CON_ID
---------- ------------ ---------- ----------
         4 UNDOTBS1              2          0
        11 EXAMPLE               4          3
        10 USERS                 3          3
         8 SYSTEM                0          3
         9 SYSAUX                1          3
```

```
SQL>
```

i. List the temp files of the PDB.

```
SQL> SELECT FILE_NAME, TABLESPACE_NAME from dba_temp_files;

FILE_NAME
------------------------------------------------------------
TABLESPACE_NAME
------------------------------
/u01/app/oracle/oradata/pdb1_1/pdb1_1_temp01.dbf
TEMP

SQL> EXIT
$
```

j. List the password file and SPFILE of both `cdb1` and `cdb2`.

```
$ cd $ORACLE_HOME/dbs
$ ls -l orapw* spfile*
-rw-r----- 1 oracle oinstall 7680 Sep  5 10:43 orapwcdb1
-rw-r----- 1 oracle oinstall 7680 Sep  7 00:48 orapwcdb2
-rw-r----- 1 oracle oinstall 7680 Sep  5 07:23 orapwem12rep
-rw-r----- 1 oracle oinstall 7680 Sep  5 10:02 orapworcl
-rw-r----- 1 oracle oinstall 7680 Sep  5 10:19 orapworcl2
-rw-r----- 1 oracle oinstall 3584 Sep  7 01:23 spfilecdb1.ora
-rw-r----- 1 oracle oinstall 3584 Sep  7 01:08 spfilecdb2.ora
-rw-r----- 1 oracle oinstall 3584 Sep  6 18:13 spfileem12rep.ora
-rw-r----- 1 oracle oinstall 3584 Sep  6 10:00 spfileorcl2.ora
-rw-r----- 1 oracle oinstall 3584 Sep  7 00:35 spfileorcl.ora
$
```

k. Check ADR files, directories, new DDL statement in alert.log.

```
$ cd $ORACLE_BASE/diag/rdbms/
$ ls
cdb1  cdb2  em12rep  orcl  orcl2
$ cd cdb2/cdb2/trace
$ vi alert_cdb2.log
…
Mon Feb 06 09:27:09 2012
Fri Oct 26 11:14:34 2012
create pluggable database PDB$SEED using
'/u01/app/oracle/product/12.1.0/dbhome_1/assistants/dbca/templat
es//pdbseed.xml'  source_file_name_convert =
('/ade/b/1201587492/oracle/oradata/seeddata/pdbseed/temp01.dbf',
'/u01/app/oracle/oradata/cdb2/pdbseed/pdbseed_temp01.dbf','/ade/
b/1201587492/oracle/oradata/seeddata/pdbseed/system01.dbf','/u01
/app/oracle/oradata/cdb2/pdbseed/system01.dbf','/ade/b/120158749
```

```
2/oracle/oradata/seeddata/pdbseed/sysaux01.dbf','/u01/app/oracle
/oradata/cdb2/pdbseed/sysaux01.dbf') NOCOPY
**************************************************************
Pluggable Database PDB$SEED with pdb id - 2 is created as
UNUSABLE.
If any errors are encountered before the pdb is marked as NEW,
then the pdb must be dropped
**************************************************************
…
alter pluggable database PDB$SEED open
…
:q
$
```

6. List all users created in the new CDB `cdb2`.
   a. Connect to `cdb2` instance.

```
$ . oraenv
ORACLE_SID = [cdb1] ? cdb2
The Oracle base for
ORACLE_HOME=/u01/app/oracle/product/12.1.0/dbhome_1 is
/u01/app/oracle
$ sqlplus / as sysdba


Connected to:
Oracle Database 12c Enterprise Edition Release 12.1.0.1.0 -
64bit Production
With the Partitioning, OLAP, Advanced Analytics and Real
Application Testing options


SQL>
```

   b. Verify that the `SYSTEM` user is created.

```
SQL> col username format A30
SQL> select username, common, con_id from cdb_users
  2  where username ='SYSTEM';


USERNAME                       COM  CON_ID
------------------------------ --- -------
SYSTEM                         YES       1
SYSTEM                         YES       2


SQL>
```

   Notice that the user `SYSTEM` exists in all containers as a common user.

   c. List all common users in the CDB.

```
SQL> select distinct username from cdb_users
  2  where common ='YES';

USERNAME
------------------------------
APEX_040200
XS$NULL
DIP
SYSBACKUP
SPATIAL_CSW_ADMIN_USR
FLOWS_FILES
CTXSYS
OUTLN
DVSYS
SYSTEM
OJVMSYS
MDDATA
ORDSYS
GSMADMIN_INTERNAL
SYSDG
LBACSYS
DVF
SYSKM
MDSYS
SPATIAL_WFS_ADMIN_USR
OLAPSYS
GSMUSER
AUDSYS
ORACLE_OCM
ORDPLUGINS
APEX_PUBLIC_USER
DBSNMP
XDB
ORDDATA
GSMCATUSER
APPQOSSYS
SYS
SI_INFORMTN_SCHEMA
ANONYMOUS
WMSYS

35 rows selected.
```

```
SQL>
```

d. List all local users in the CDB.

```
SQL> select distinct username, CON_ID from cdb_users
  2  where common ='NO';

no rows selected

SQL>
```

e. List local users in root.

```
SQL> select distinct username from dba_users
  2  where common ='NO';

no rows selected

SQL>
```

Notice that there is no local user in the root container because it is impossible to create any local user in the root.

7. View distinct accesses by different containers to the single SGA.

```
SQL> select distinct status, con_id from v_$bh order by 2 ;
STATUS         CON_ID
---------- ----------
cr                  1
free                1
xcur                1
cr                  2
xcur                2

SQL> EXIT
$
```

Practice 3-3: Creating a PDB from Seed

Overview

In this practice, you will create a new PDB `pdb2_1` in `cdb2` from seed using SQL*Plus first and then SQL Developer. This means that between the two operations, you will have to drop the PDB `pdb2_1`.

Assumptions

The creation of the CDB `cdb2` is successful.

Tasks

Method using SQL*Plus:

Use SQL*Plus first to create the PDB `pdb2_1` in `cdb2`.

1. Create a directory for the new data files of `pdb2_1` of `cdb2`.

```
$ . oraenv
ORACLE_SID = [cdb2] ? cdb2
The Oracle base remains unchanged with value /u01/app/oracle

$ cd $ORACLE_BASE/oradata/cdb2
$ mkdir pdb2_1
$
```

2. Run SQL*Plus and connect to the root with a user with `CREATE PLUGGABLE DATABASE` privilege.

```
$ sqlplus / as sysdba

Connected to:
Oracle Database 12c Enterprise Edition Release 12.1.0.1.0 -
64bit Production
With the Partitioning, OLAP, Advanced Analytics and Real
Application Testing options

SQL> CREATE PLUGGABLE DATABASE pdb2_1 ADMIN USER pdb2_1_admin
  2  IDENTIFIED BY oracle_4U ROLES=(CONNECT)
  3  FILE_NAME_CONVERT=('/u01/app/oracle/oradata/cdb2/pdbseed'
  4                  ,'/u01/app/oracle/oradata/cdb2/pdb2_1');

Pluggable database created.

SQL>
```

3. Check the open mode of `pdb2_1`.

```
SQL> col con_id format 999
SQL> col name format A10
```

```
SQL> select con_id, NAME, OPEN_MODE,DBID, CON_UID from V$PDBS;

CON_ID NAME        OPEN_MODE        DBID    CON_UID
------ ---------- ---------- ---------- ----------
     2 PDB$SEED   READ ONLY  4040266138 4040266138
     3 PDB2_1     MOUNTED    3082499546 3082499546

SQL>
```

4. Open `pdb2_1`.
   a. Open the PDB.

```
SQL> alter pluggable database pdb2_1 open;

Pluggable database altered.
SQL> EXIT
$
```

   b. Connect to the PDB.
      1) Use netca to add the `PDB2_1` net service name for `pdb2_1` pluggable database of `cdb2` in the `tnsnames.ora` file.

```
$ netca
```

      2) On the Welcome page, select “Local Net Service Name configuration” and click Next.
      3) On the Net Service Name Configuration page, accept `Add` and click Next.
      4) On the Net Service Name Configuration, Service Name page, enter `pdb2_1` as Service Name and click Next.
      5) On the Net Service Name Configuration, Select Protocols page, select TCP and click Next.
      6) On the Net Service Name Configuration, TCP/IP Protocol page, enter your complete host name, for example, <*yourservername*>, or *localhost,* accept “Use the standard port number of 1521,” and click Next.
      7) On the Net Service Name Configuration, Test page, select “No, do not test” (the pluggable database is not yet opened) and click Next.
      8) On the Net Service Name Configuration, Net Service Name page, accept `pdb2_1` as Net Service Name and click Next.
      9) On the Net Service Name Configuration, Another Net Service page, select No, and Next.
      10) On the Net Service Name Configuration Complete page, click Next.
      11) When you are back on the Welcome page, click Finish.
      12) Connect to `pdb2_1 AS SYSDBA`.

```
$ sqlplus sys/oracle_4U@pdb2_1 AS SYSDBA

Connected to:
Oracle Database 12c Enterprise Edition Release 12.1.0.1.0 -
64bit Production
```

```
With the Partitioning, OLAP, Advanced Analytics and Real
Application Testing options


SQL>
```

5. The service is now available and registered with the listener.

```
SQL> !lsnrctl status
…
Service "cdb1" has 1 instance(s).
  Instance "cdb1", status READY, has 1 handler(s) for this
service...
Service "cdb1XDB" has 1 instance(s).
  Instance "cdb1", status READY, has 1 handler(s) for this
service...
Service "cdb2" has 1 instance(s).
  Instance "cdb2", status READY, has 1 handler(s) for this
service...
Service "cdb2XDB" has 1 instance(s).
  Instance "cdb2", status READY, has 1 handler(s) for this
service...
Service "em12rep" has 1 instance(s).
  Instance "em12rep", status READY, has 1 handler(s) for this
service...
Service "em12repXDB" has 1 instance(s).
  Instance "em12rep", status READY, has 1 handler(s) for this
service...
Service "pdb1_1" has 1 instance(s).
  Instance "cdb1", status READY, has 1 handler(s) for this
service.
Service "pdb2_1" has 1 instance(s).
  Instance "cdb2", status READY, has 1 handler(s) for this
service.
The command completed successfully

SQL>
```

6. Connect to `pdb2_1` as `sys` user by using EasyConnect and then as `pdb2_1_admin` user.

```
SQL> CONNECT sys/oracle_4U@localhost:1521/pdb2_1 AS SYSDBA
Connected.
SQL> connect pdb2_1_admin/oracle_4U@PDB2_1
Connected.

SQL> show con_name
CON_NAME
------------------------------
PDB2_1
```

```
SQL>
```

7. List the data files created.

```
SQL> !ls -l $ORACLE_BASE/oradata/cdb2/pdb2_1/*
-rw-r----- 1 oracle oinstall  20979712 Oct 29 04:32
/u01/app/oracle/oradata/cdb2/pdb2_1/pdbseed_temp01.dbf
-rw-r----- 1 oracle oinstall 713039872 Oct 29 04:52
/u01/app/oracle/oradata/cdb2/pdb2_1/sysaux01.dbf
-rw-r----- 1 oracle oinstall 272637952 Oct 29 04:52
/u01/app/oracle/oradata/cdb2/pdb2_1/system01.dbf

SQL>
```

8. Check the services, data files, and tablespaces using views.

```
SQL> connect system/oracle_4U@pdb2_1
Connected.
SQL> col name format A30
SQL> select name from v$services;

NAME
------------------------------
pdb2_1

SQL> col file_name format A50
SQL> col tablespace_name format A8
SQL> col file_id format 99
SQL> col con_id format 9
SQL> select FILE_NAME, TABLESPACE_NAME, FILE_ID, con_id
  2  from   cdb_data_files
  3  order  by con_id ;

FILE_NAME                                          TABLESPA
FILE_ID CON_ID
-------------------------------------------------- -------- ----
--- ------
/u01/app/oracle/oradata/cdb2/pdb2_1/sysaux01.dbf   SYSAUX
9      3
/u01/app/oracle/oradata/cdb2/pdb2_1/system01.dbf   SYSTEM
8      3

SQL> select FILE_NAME, TABLESPACE_NAME, FILE_ID
  2  from   dba_data_files;

FILE_NAME                                          TABLESPA
FILE_ID
```

```
---------------------------------------------- -------- ----
---
/u01/app/oracle/oradata/cdb2/pdb2_1/system01.dbf   SYSTEM
8
/u01/app/oracle/oradata/cdb2/pdb2_1/sysaux01.dbf   SYSAUX
9

SQL> col file_name format A60
SQL> select FILE_NAME, TABLESPACE_NAME, FILE_ID
  2  from cdb_temp_files;

FILE_NAME                                          TABLESPA
FILE_ID
-------- ---------------------------------------------- -------
------
/u01/app/oracle/oradata/cdb2/pdb2_1/pdbseed_temp01.dbf TEMP
3

SQL> select FILE_NAME, TABLESPACE_NAME, FILE_ID
  2  from dba_temp_files;

FILE_NAME                                           TABLESPA
FILE_ID
---------------------------------------------------------- ------ -
------
/u01/app/oracle/oradata/cdb2/pdb2_1/pdbseed_temp01.dbf TEMP
3

SQL>
```

9. To be able to view all objects of all containers in the CDB, connect to the root and use CDB_*xxx* views.

```
SQL> connect / as sysdba
Connected.
SQL> show con_id

CON_ID
------------------------------
1

SQL> show con_name

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

```
SQL> select name from v$services;

NAME
------------------------------
pdb2_1
cdb2XDB
cdb2
SYS$BACKGROUND
SYS$USERS

SQL> select FILE_NAME, TABLESPACE_NAME, FILE_ID, con_id
  2  from   cdb_data_files
  3  order  by con_id, file_id ;

FILE_NAME                                          TABLESPA
FILE_ID CON_ID
-------------------------------------------------- -------- ----
--- ------
FILE_NAME                                          TABLESPA
FILE_ID CON_ID
-------------------------------------------------- -------- ----
--- ------
/u01/app/oracle/oradata/cdb2/system01.dbf          SYSTEM
1      1
/u01/app/oracle/oradata/cdb2/sysaux01.dbf          SYSAUX
3      1
/u01/app/oracle/oradata/cdb2/undotbs01.dbf         UNDOTBS1
4      1
/u01/app/oracle/oradata/cdb2/users01.dbf           USERS
6      1
/u01/app/oracle/oradata/cdb2/pdbseed/system01.dbf  SYSTEM
5      2
/u01/app/oracle/oradata/cdb2/pdbseed/sysaux01.dbf  SYSAUX
7      2
/u01/app/oracle/oradata/cdb2/pdb2_1/system01.dbf   SYSTEM
8      3
/u01/app/oracle/oradata/cdb2/pdb2_1/sysaux01.dbf   SYSAUX
9      3

8 rows selected.

SQL> select FILE_NAME, TABLESPACE_NAME, FILE_ID
  2  from   dba_data_files;
```

```
FILE_NAME                                                TABLESPA
FILE_ID
-------------------------------------------------- -------- ----
---
/u01/app/oracle/oradata/cdb2/users01.dbf                 USERS
6
/u01/app/oracle/oradata/cdb2/undotbs01.dbf               UNDOTBS1
4
/u01/app/oracle/oradata/cdb2/sysaux01.dbf                SYSAUX
3
/u01/app/oracle/oradata/cdb2/system01.dbf                SYSTEM
1

SQL> select FILE_NAME, TABLESPACE_NAME, FILE_ID
  2  from cdb_temp_files;

FILE_NAME                                                TABLESPA
FILE_ID
-------------------------------------------------------
------ -------
/u01/app/oracle/oradata/cdb2/temp01.dbf
TEMP     1
/u01/app/oracle/oradata/cdb2/pdb2_1/pdbseed_temp01.dbf
TEMP     3
/u01/app/oracle/oradata/cdb2/pdbseed/pdbseed_temp01.dbf
TEMP     2

SQL> select FILE_NAME, TABLESPACE_NAME, FILE_ID
  2  from dba_temp_files;

FILE_NAME                                                TABLESPA
FILE_ID
-------------------------------------------------------- --------
-------
/u01/app/oracle/oradata/cdb2/temp01.dbf                  TEMP
1
SQL> EXIT
$
```

Method using SQL Developer:

1. Because you already created `pdb2_1` using SQL*Plus commands and would like to test the creation by using SQL Developer, you first have to drop `pdb2_1` to recreate it.
   a. Drop the pluggable database `pdb2_1`.

```
$ sqlplus / AS SYSDBA
```

```
Connected to:
Oracle Database 12c Enterprise Edition Release 12.1.0.1.0 -
64bit Production
With the Partitioning, OLAP, Advanced Analytics and Real
Application Testing options

SQL> ALTER PLUGGABLE DATABASE pdb2_1 CLOSE IMMEDIATE;

Pluggable database altered.

SQL> DROP PLUGGABLE DATABASE pdb2_1 INCLUDING DATAFILES;

Pluggable database dropped.

SQL> EXIT
$
```

b. Remove the directory.

```
$ rm -r $ORACLE_BASE/oradata/cdb2/pdb2_1
$
```

2. Create a directory for the new data files of `pdb2_1` of `cdb2`.

```
$ cd $ORACLE_BASE/oradata/cdb2
$ mkdir pdb2_1
$
```

3. Run SQL*Plus and connect to the root to set the OMF directory to the `pdb2_1` directory.

```
$ sqlplus / AS SYSDBA

Connected to:
Oracle Database 12c Enterprise Edition Release 12.1.0.1.0 -
64bit Production
With the Partitioning, OLAP, Advanced Analytics and Real
Application Testing options

SQL> alter system set db_create_file_dest =
'/u01/app/oracle/oradata/cdb2/pdb2_1' scope=both;
System altered.

SQL> EXIT
$
```

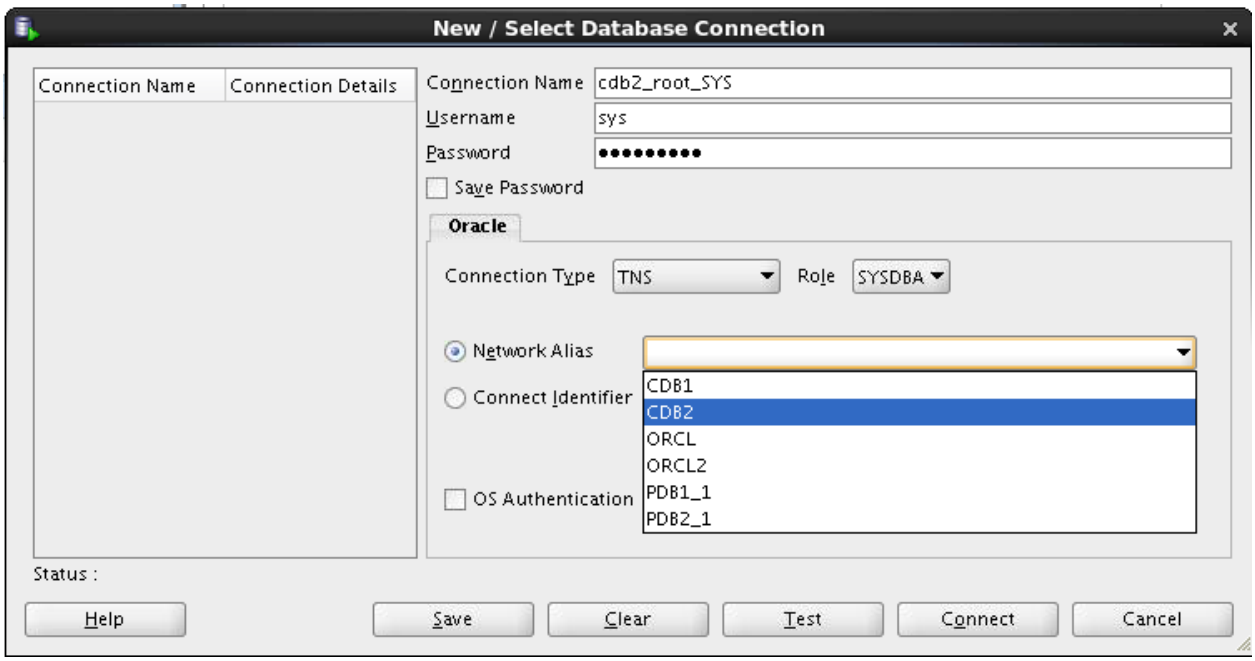
4. Launch SQL Developer.

```
$ cd $ORACLE_HOME/sqldeveloper
$ ./sqldeveloper.sh
```

```
Oracle SQL Developer
 Copyright (c) 1997, 2011, Oracle and/or its affiliates. All
rights reserved.
```

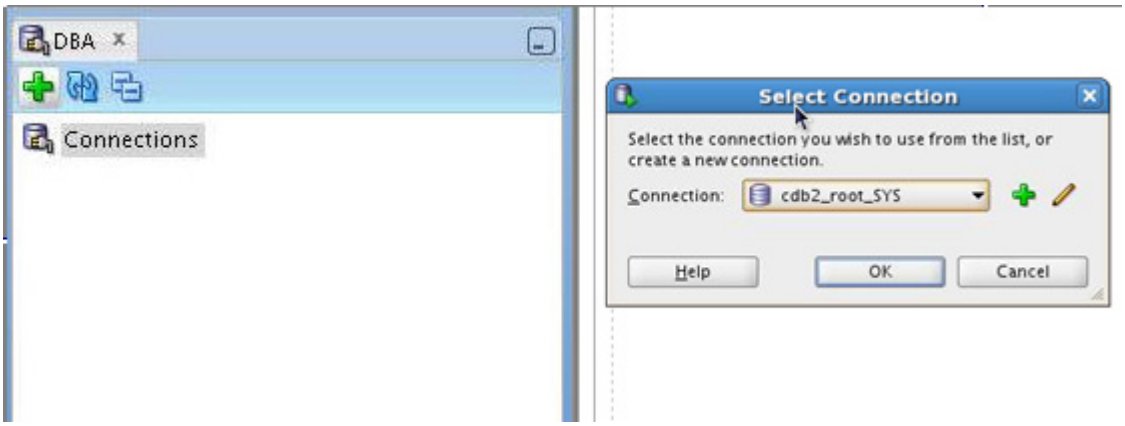
5. In SQL Developer, create a connection as `SYS` in root `cdb2`.
6. Open a connection as `SYS` in `cdb2`.
   a. Choose the View option.
   b. Click Connections.
   c. Click the sign + in the left Connections pane to add a new connection.
   d. Fill the different fields as follows: be sure to change the host name and port number to your assigned host name and port number.

Window/Page Description	Choices or Values
Connection Name	`cdb2_root_SYS`
Username	`sys`
Password	`oracle_4U`
Connection Type	`TNS`
Role	`SYSDBA`
Network Alias	`CDB2`

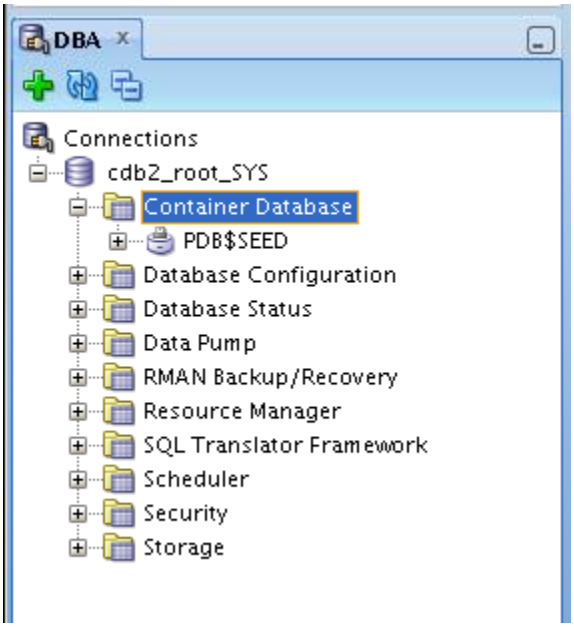


   e. Click Test.
   f. If the status is Success, click Save.
   g. Click Connect.
7. To manage the CDB and its PDBs:
   a. Choose the View option.

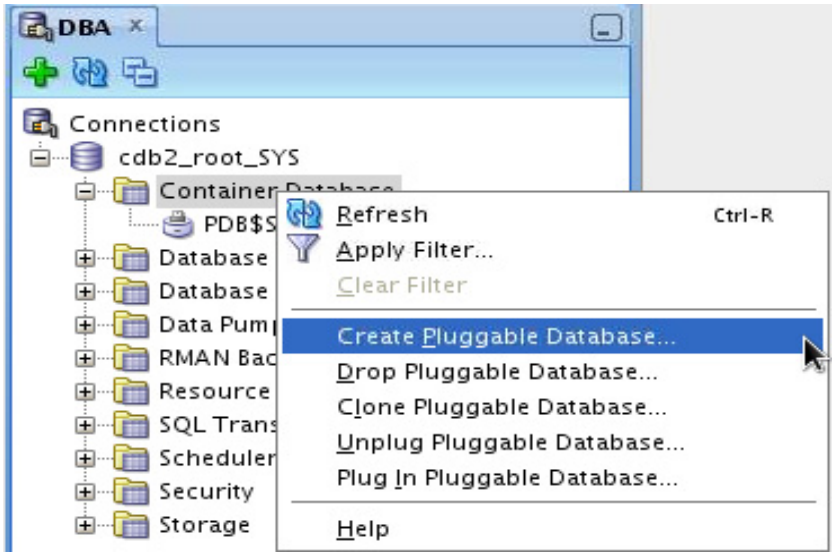
b. Click DBA.
c. Click + in the left Connections pane to view an existing connection.
d. From the list of existing connections, choose the one you just created.



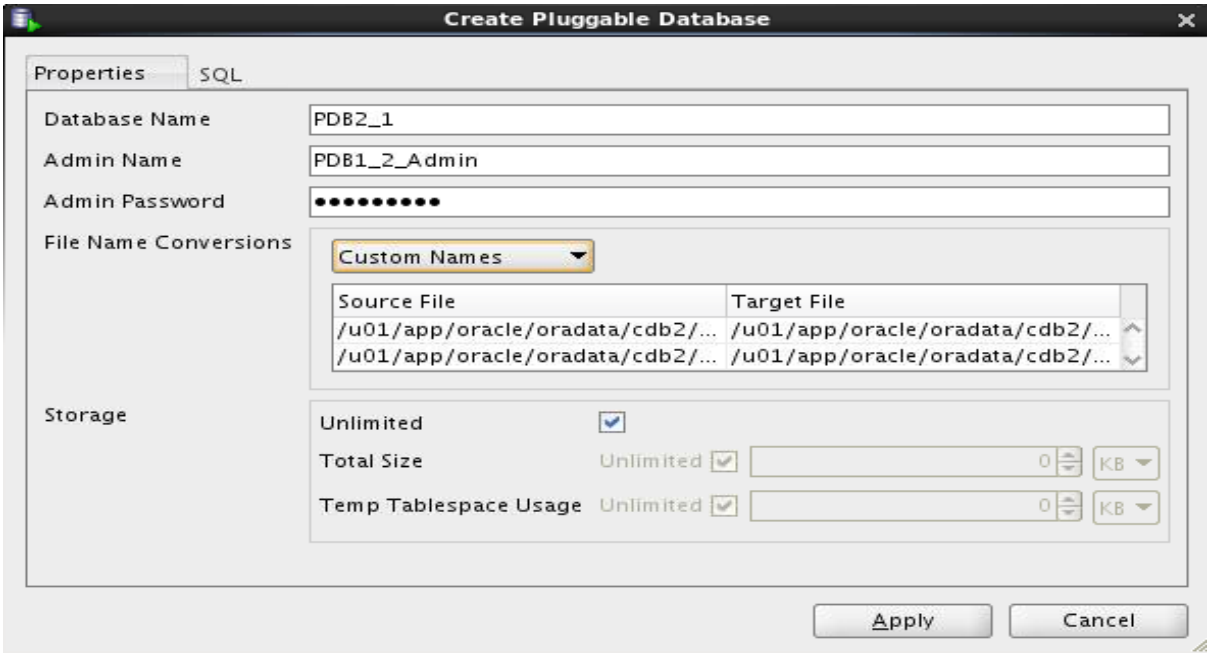
e. Click OK.
f. Click + in front of the name of the `cdb2_root_SYS` connection to expand the folder. Then click the sign + in front of "Container Database". The list of containers in the CDB appears.



8. Right-click the `Container Database` to show possible actions.
   a. Choose Create Pluggable Database ….

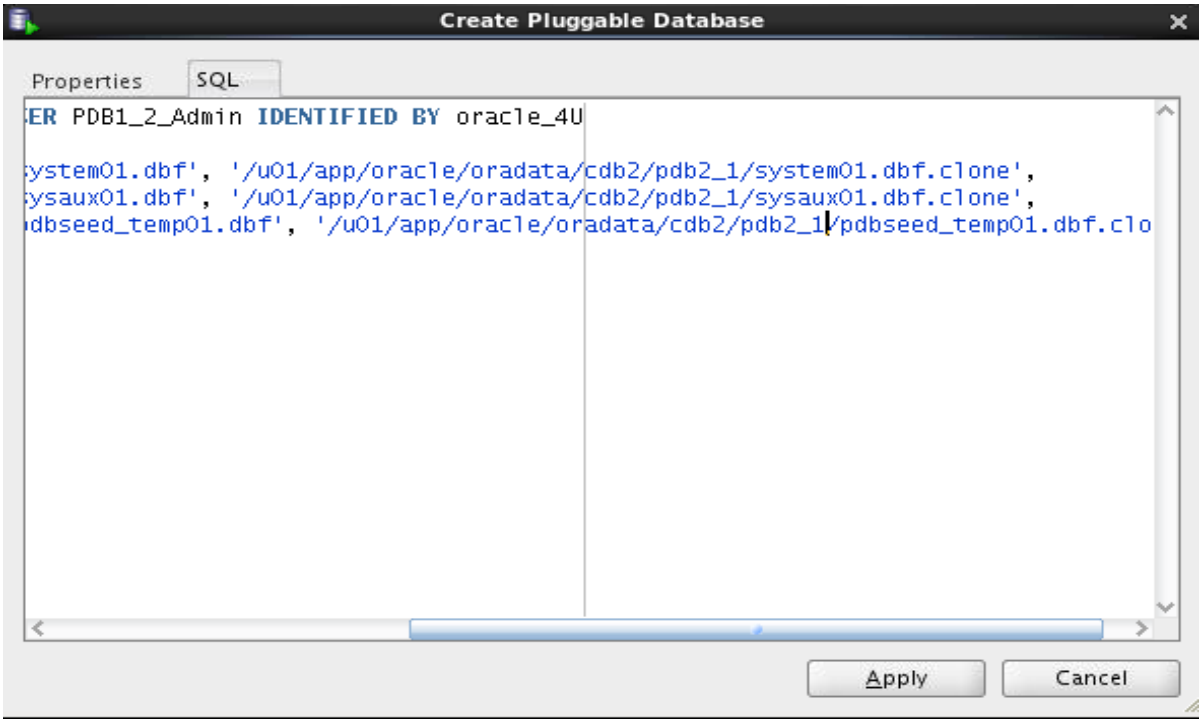


b. Fill the different fields as follows:



Window/Page Description	Choices or Values
Database Name	PDB2_1
Admin Name	PDB2_1_Admin
Admin Password	oracle_4U
File Name Conversions	Custom Names

c. View the SQL statement before applying to change the target file names. Change the `pdbseed` directory to `pdb2_1` in the three data file names and remove the `.clone` from the end of the file names.

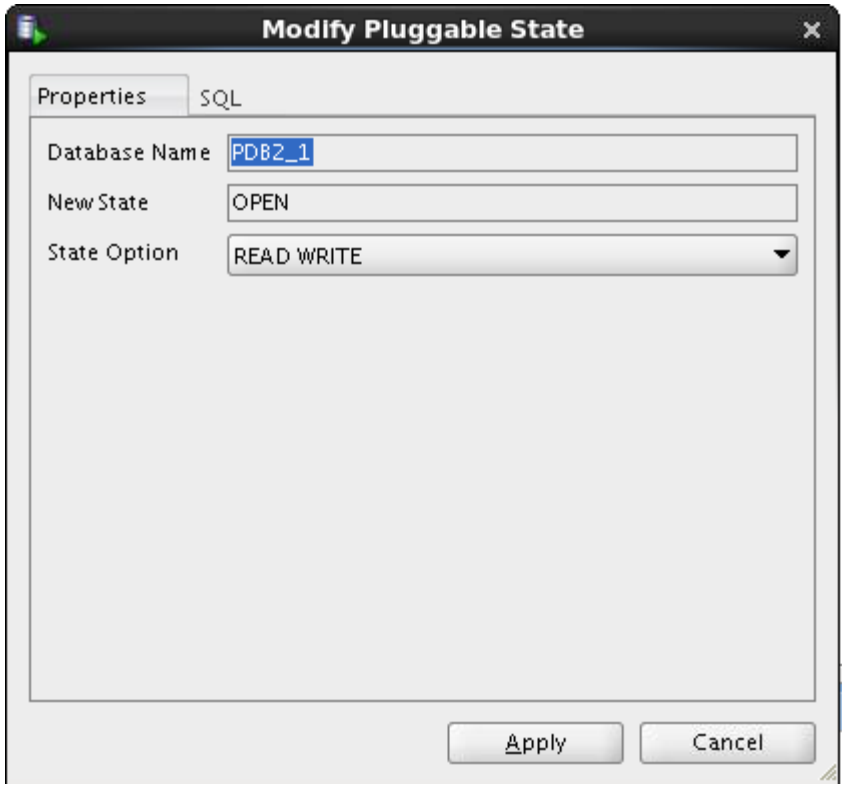


d. Click Apply then OK. The new `pdb2_1` appears in the list of PDBs in `cdb2`.

e. Open `pdb2_1` in `READ WRITE` mode.

1) Right-click the pluggable database `pdb2_1` to show possible actions.

2) Choose `Modify State` to set the State Option to `READ WRITE` open mode.



3) Click Apply, then click OK.

4) Quit SQL Developer. Click **File** and then click **Exit**.

Practice 3-4: Cloning PDB Within the Same CDB

Overview

In this practice, you will create a new PDB with the cloning method, cloning `pdb2_2` from `pdb2_1` within the same CDB `cdb2`.

Use the easiest tool for you, either SQL*Plus **OR** SQL Developer tool.

Assumptions

The `pdb2_1` creation has completed successfully in Practice 3-3.

Tasks

Use either the SQL commands OR the SQL Developer.

Method using SQL*Plus.

1. Create a directory for the new data files of `pdb2_2` of `cdb2`.

```
$ cd $ORACLE_BASE/oradata/cdb2
$ mkdir pdb2_2
```

2. Run SQL*Plus and connect to the root as a user granted with `CREATE PLUGGABLE DATABASE` privilege.
   a. Set `pdb2_1` in `READ ONLY` open mode before cloning.

```
$ sqlplus / as sysdba


Connected to:
Oracle Database 12c Enterprise Edition Release 12.1.0.1.0 -
64bit Production
With the Partitioning, OLAP, Advanced Analytics and Real
Application Testing options

SQL> alter pluggable database pdb2_1 close;
Pluggable database altered.

SQL> alter pluggable database pdb2_1 open read only;
Pluggable database altered.


SQL>
```

   b. Change OMF `DB_CREATE_FILE_DEST` parameter value to `'/u01/app/oracle/oradata/cdb2/pdb2_2'`.

```
SQL> alter system set db_create_file_dest =
'/u01/app/oracle/oradata/cdb2/pdb2_2';
System altered.

SQL>
```

   c. Clone `pdb2_2` from `pdb2_1`.

```
SQL> CREATE PLUGGABLE DATABASE pdb2_2 FROM pdb2_1;
Pluggable database created.

SQL>
```

3. Check the open mode of pdb2_2.

```
SQL> select name, open_mode from v$pdbs;

NAME                           OPEN_MODE
------------------------------ ----------
PDB$SEED                       READ ONLY
PDB2_1                         READ ONLY
PDB2_2                         MOUNTED

SQL>
```

4. Set PDB2_1 in READ WRITE open mode and open PDB2_2.
   a. Open PDB2_1 in READ WRITE mode.

```
SQL> alter pluggable database PDB2_1 close;

Pluggable database altered.

SQL> alter pluggable database PDB2_1 open;

Pluggable database altered.

SQL>
```

   b. Open PDB2_2 in READ WRITE mode.

```
SQL> alter pluggable database PDB2_2 open;

Pluggable database altered.

SQL> EXIT
$
```

   c. Connect to pdb2_2 AS SYSDBA.
      1) Use netca to add the PDB2_2 net service name for pdb2_2 pluggable database of cdb2 in the tnsnames.ora file.

```
$ netca
```

      2) On the Welcome page, select the “Local Net Service Name configuration” and click Next.
      3) On the Net Service Name Configuration page, accept Add and click Next.
      4) On the Net Service Name Configuration, Service Name page, enter pdb2_2 as Service Name and click Next.

5) On the Net Service Name Configuration, Select Protocols page, select TCP and click Next.
6) On the Net Service Name Configuration, TCP/IP Protocol page, enter your complete host name, for example, <*yourservername*>, or *localhost,* accept "Use the standard port number of 1521," and click Next.
7) On the Net Service Name Configuration, Test page, select "No, do not test" (the pluggable database is not yet opened) and click Next.
8) On the Net Service Name Configuration, Net Service Name page, accept `pdb2_2` as Net Service Name, and click Next.
9) On the Net Service Name Configuration, Another Net Service page, select No, and Next.
10) On the Net Service Name Configuration Complete page, click Next.
11) When you are back on the Welcome page, click Finish.

```
$ sqlplus sys/oracle_4U@pdb2_2 AS SYSDBA

Connected to:
Oracle Database 12c Enterprise Edition Release 12.1.0.1.0 -
64bit Production
With the Partitioning, OLAP, Advanced Analytics and Real
Application Testing options

SQL>
```

d. Check the open mode of the PDBs.

```
SQL> CONNECT / AS SYSDBA
Connected.

SQL> select name, open_mode from v$pdbs;

NAME                           OPEN_MODE
------------------------------ ----------
PDB$SEED                       READ ONLY
PDB2_1                         READ WRITE
PDB2_2                         READ WRITE

SQL>
```

5. Connect to `PDB2_2` as the `SYSTEM` user.

```
SQL> connect system/oracle_4U@PDB2_2
Connected.

SQL> show con_name
PDB2_2

SQL> EXIT
```

```
$
```

6. List the data files created.

```
$ cd $ORACLE_BASE/oradata/cdb2/pdb2_2
$ ls -l
total 4
drwxr-x--- 3 oracle oinstall 4096 Oct 29 06:34 CDB2
$ cd CDB2
$ ls -l
total 4
drwxr-x--- 3 oracle oinstall 4096 Oct 29 06:34
CD2DDD0A5BF67AB8E0436B23B98B987D
$ cd CD2DDD0A5BF67AB8E0436B23B98B987D
$ ls -l
total 4
drwxr-x--- 2 oracle oinstall 4096 Oct 29 06:35 datafile
$ cd datafile
$ ls -l
total 809836
-rw-r----- 1 oracle oinstall 713039872 Oct 29 06:42
o1_mf_sysaux_88w8vyg8_.dbf
-rw-r----- 1 oracle oinstall 272637952 Oct 29 06:42
o1_mf_system_88w8vycm_.dbf
-rw-r----- 1 oracle oinstall  20979712 Oct 29 06:44
o1_mf_temp_88w8x9kk_.dbf
$
```

OR

Method with SQL Developer:

7. Create a directory for the new data files of `pdb2_2` of `cdb2`.

```
$ cd $ORACLE_BASE/oradata/cdb2
$ mkdir pdb2_2
$
```

8. Run SQL*Plus and connect to the root to set OMF directory to the `pdb2_2` directory.

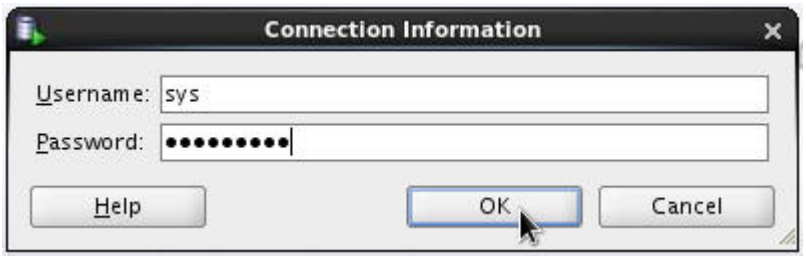
```
$ sqlplus / AS SYSDBA


Connected to:
Oracle Database 12c Enterprise Edition Release 12.1.0.1.0 -
64bit Production
With the Partitioning, OLAP, Advanced Analytics and Real
Application Testing options


SQL> alter system set db_create_file_dest =
'/u01/app/oracle/oradata/cdb2/pdb2_2' scope=both;
```

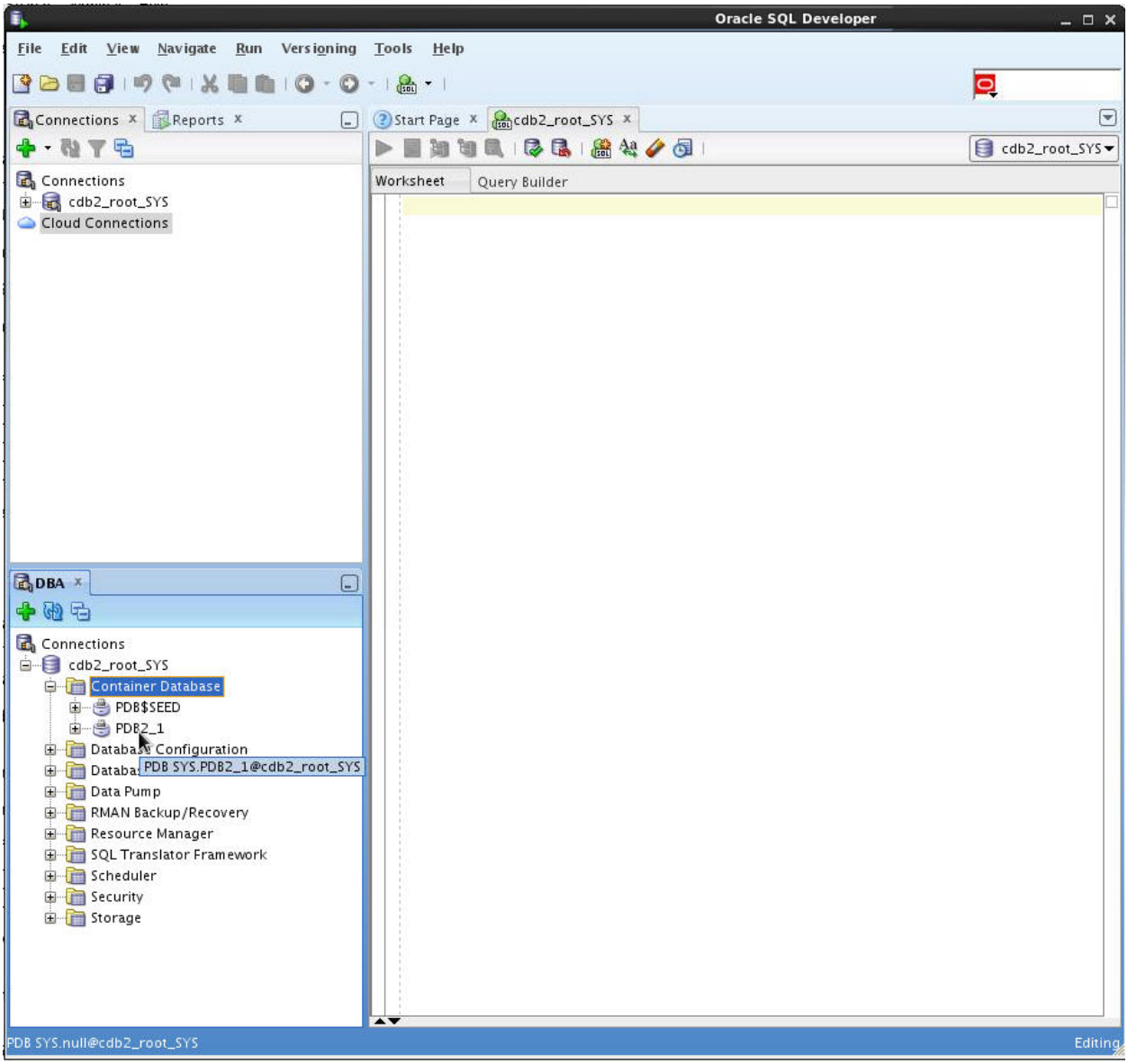
```
System altered.


SQL> EXIT
Disconnected from Oracle Database 12c Enterprise Edition Release
12.1.0.1.0 - 64bit Production
With the Partitioning, OLAP, Advanced Analytics and Real
Application Testing options
$
```

9. Launch SQL Developer.

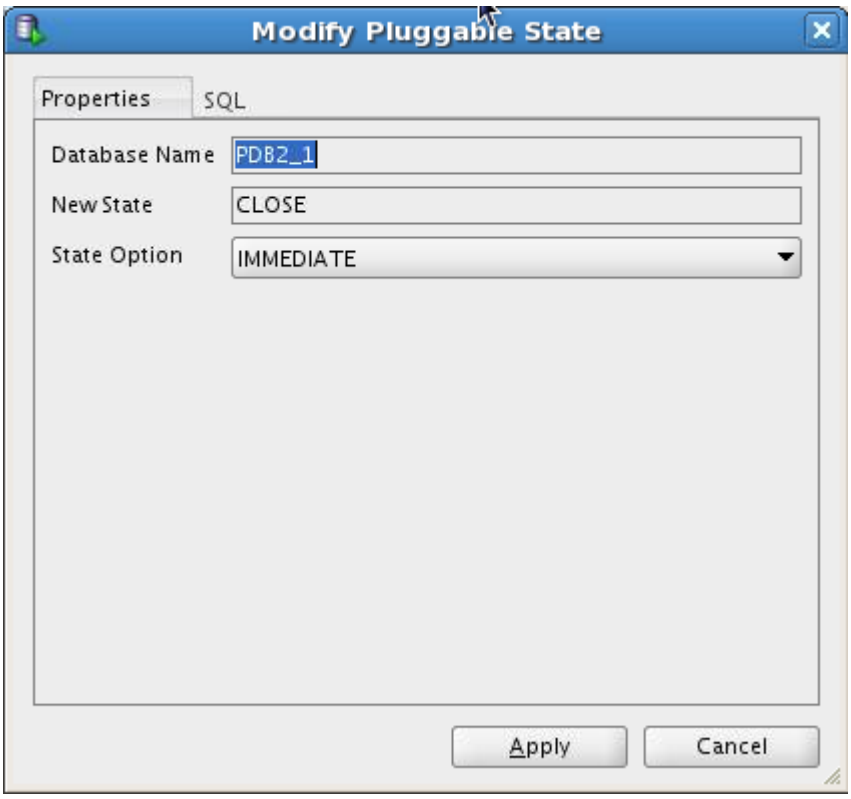
```
$ cd $ORACLE_HOME/sqldeveloper
$ ./sqldeveloper.sh
```

10. Click the sign + in front of the name of the `cdb2_root_SYS` connection to expand the folder. The Connection Information is requested. Enter `oracle_4U` for the `SYS` password. Then click the sign + in front of "Container Database". The list of containers in the CDB appears.





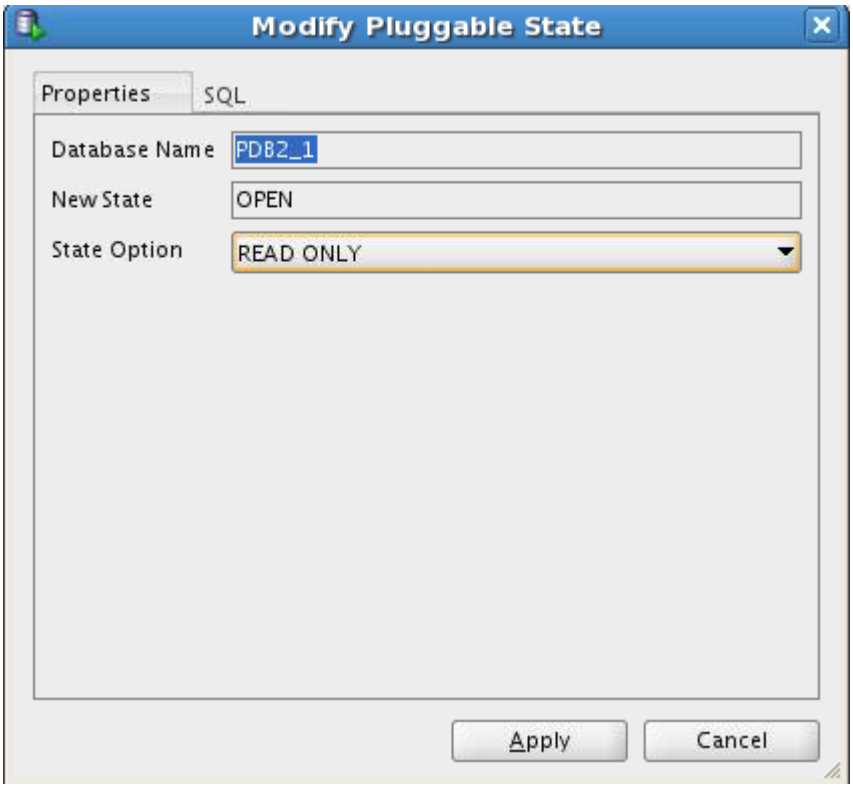
11. Right-click the pluggable database `pdb2_1` to show possible actions.
    Choose `Modify State` to set it in `READ ONLY` open mode before cloning.
    a. First close the PDB.



b. Click Apply. Then click OK.

c. Choose Modify State again.

d. Set the State Option to `READ ONLY`.

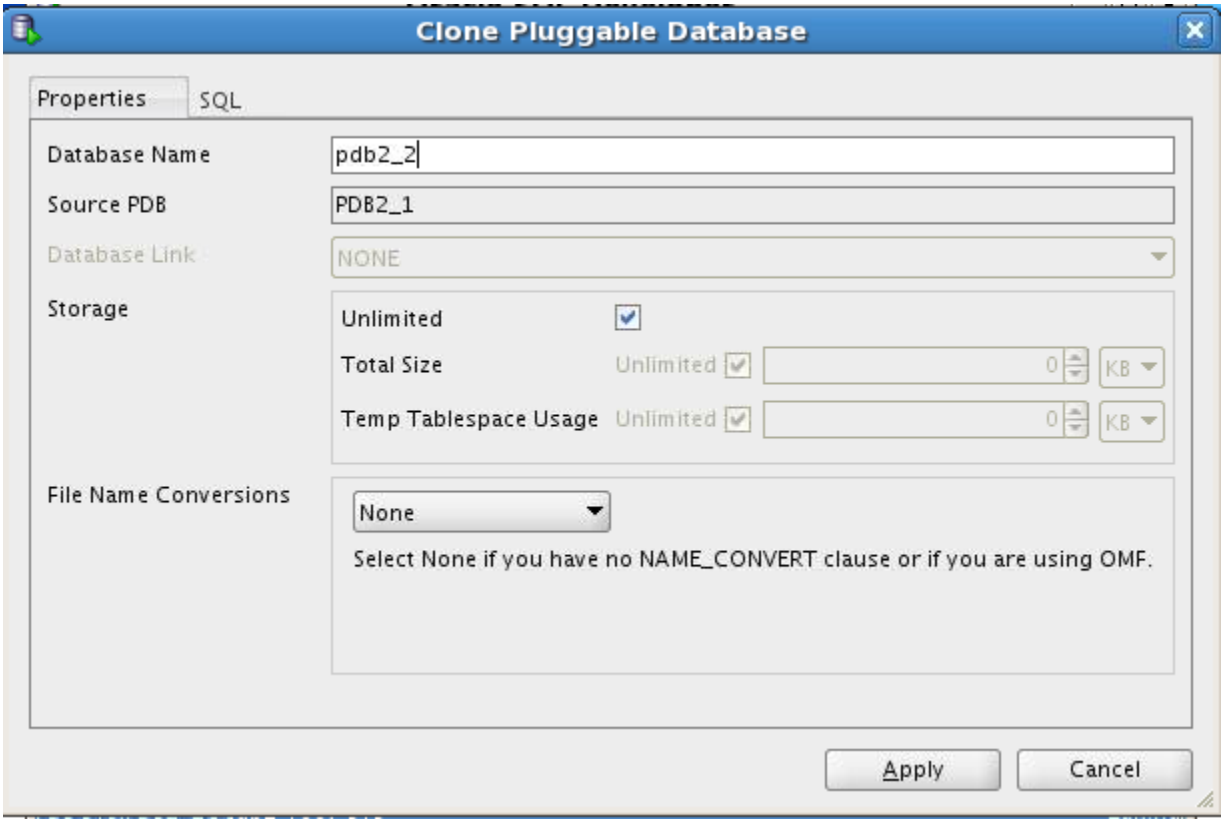


e. Click Apply and then click OK.

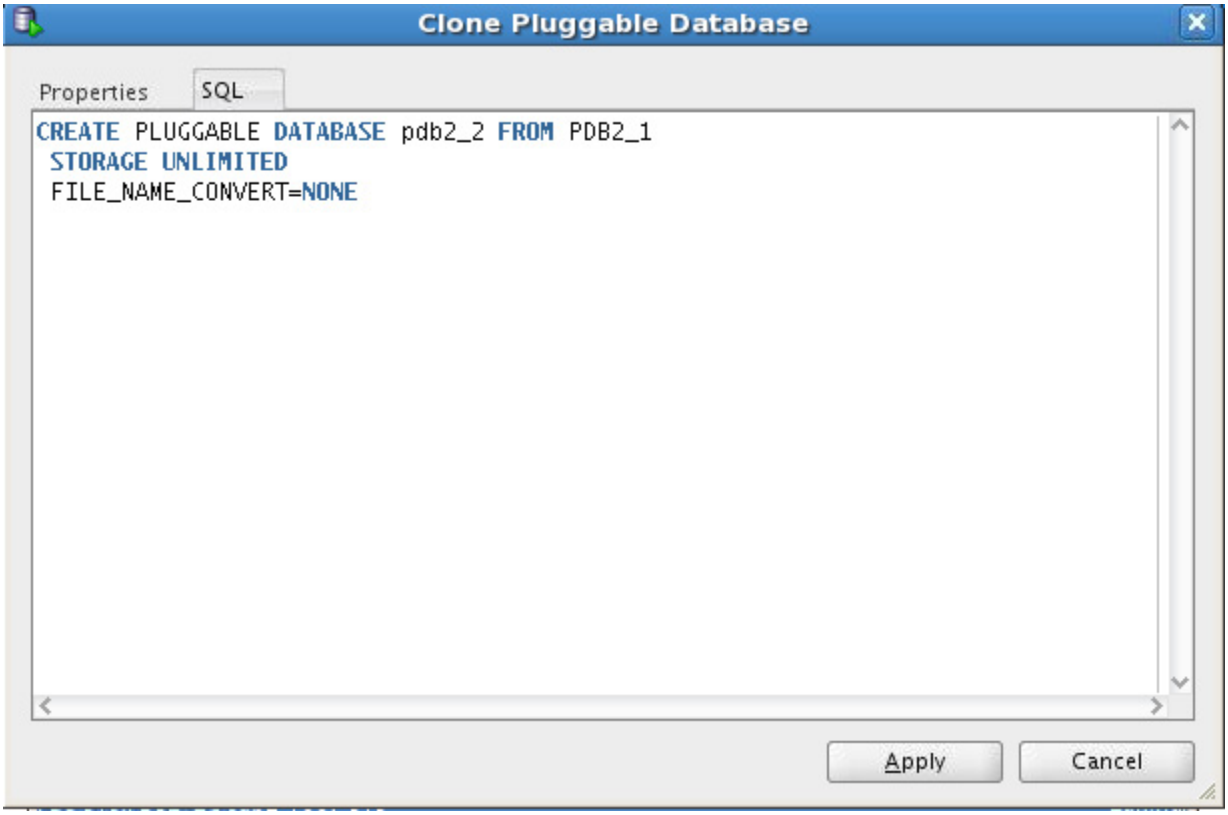
12. Right-click the pluggable database `pdb2_1` and choose Clone Pluggable Database….
    a. Fill the different fields as follows.

Window/Page Description	Choices or Values
Database Name	`pdb2_2`
Source PDB	`pdb2_1`
File Name Conversions	`None`

File Name Conversions kept to `None` means that it uses the OMF target destination declared in `DB_CREATE_FILE_DEST` parameter.



    b. You can view the SQL statement before applying.

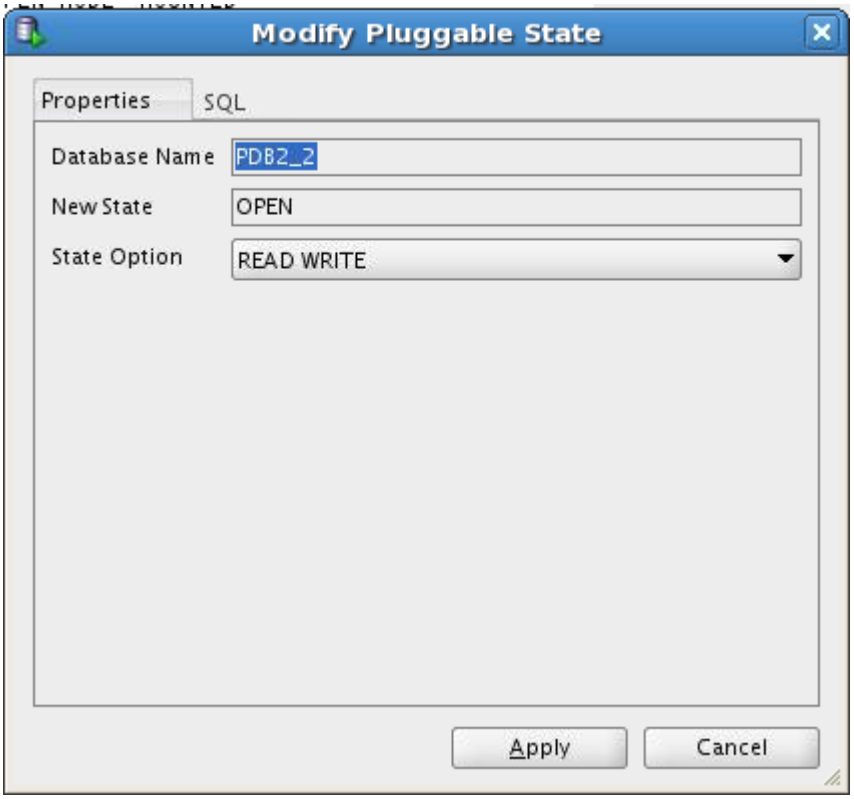


c. Click Apply and then click OK. The new `pdb2_2` appears in the list of PDBs in `cdb2`.

d. Open `pdb2_2` in `READ WRITE` mode.

   1) Right-click the pluggable database `pdb2_2` to show possible actions.

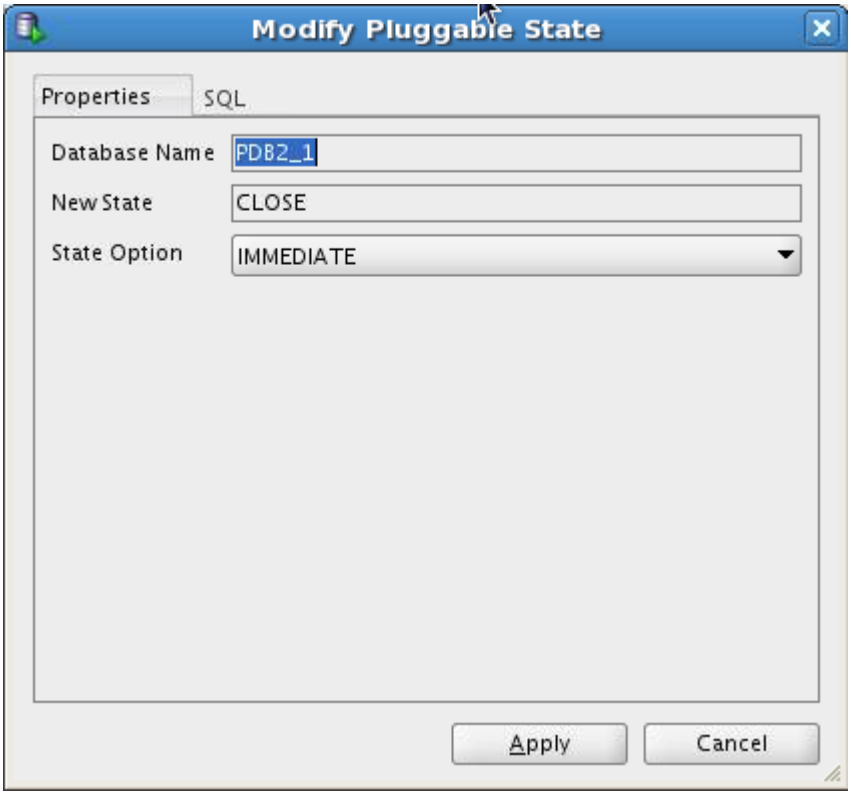
   2) Choose `Modify State` to set the State Option to `READ WRITE` open mode.



3) Click Apply, then click OK.

e. Right-click the pluggable database `pdb2_1` to set it back in `READ WRITE` mode . Choose `Modify State` to set it in `READ WRITE` open mode.

f. First close the PDB.



g. Click Apply. Then click OK.
h. Choose Modify State again.
i. Set the State Option to `READ WRITE`.
j. Click Apply then click OK.

Practice 3-5: Plugging a Non-CDB into a CDB

Overview

In this practice, you will plug the non-CDB `orcl2` into the CDB `cdb2`. You will not use Export/Import DataPump, which can be a possible method and covered in another practice for the lesson "Miscellaneous", but the method with `DBMS_PDB` package. This package executed in the non-CDB `orcl2` generates an XML file describing the tablespaces and data files of non-CDB `orcl2`. The XML file is then used when creating `pdb_orcl2` in `cdb2`.

Tasks

1. Use `DBMS_PDB.DESCRIBE` to "unplug" non-CDB `orcl2`.

```
$ . oraenv
ORACLE_SID = [cdb2] ? orcl2
The Oracle base remains unchanged with value /u01/app/oracle
$ sqlplus / as sysdba

Connected to an idle instance.

SQL> startup mount
ORACLE instance started.

Total System Global Area 1670221824 bytes
Fixed Size                  2274000 bytes
Variable Size             973081904 bytes
Database Buffers          687865856 bytes
Redo Buffers                7000064 bytes
Database mounted.
SQL>
SQL> alter database open read only;

Database altered.
SQL> exec dbms_pdb.describe
('/u01/app/oracle/oradata/orcl2/xmlorcl2.xml')

PL/SQL procedure successfully completed.

SQL> shutdown immediate
Database closed.
Database dismounted.
ORACLE instance shut down.
SQL> EXIT
$
```

2. Create a new PDB `pdb_orcl2` to plug non-CDB `orcl2` into `cdb2` using the XML file generated.
   You will have to remove the temp file because the creation cannot complete until it is removed to create it.

```
$ . oraenv
ORACLE_SID = [orcl2] ? cdb2
The Oracle base for
ORACLE_HOME=/u01/app/oracle/product/12.1.0/dbhome_1 is
/u01/app/oracle
$ sqlplus / as sysdba


Connected to:
Oracle Database 12c Enterprise Edition Release 12.1.0.1.0 -
64bit Production
With the Partitioning, OLAP, Advanced Analytics and Real
Application Testing options


SQL> create pluggable database PDB_ORCL2  using
'/u01/app/oracle/oradata/orcl2/xmlorcl2.xml' NOCOPY;
create pluggable database PDB_ORCL2  using
'/u01/app/oracle/oradata/orcl2/xmlorcl2.xml' NOCOPY
*
ERROR at line 1:
ORA-27038: created file already exists
ORA-01119: error in creating database file
'/u01/app/oracle/oradata/orcl2/temp01.dbf'


SQL> !rm /u01/app/oracle/oradata/orcl2/temp01.dbf


SQL>
SQL> create pluggable database PDB_ORCL2  using
'/u01/app/oracle/oradata/orcl2/xmlorcl2.xml' NOCOPY;


Pluggable database created.


SQL> EXIT
$
```

3. To complete the operation, you have to convert the plugged non-CDB to a proper PDB by deleting unnecessary metadata from PDB `SYSTEM` tablespace.
   For this purpose, you execute the `$ORACLE_HOME/rdbms/admin/noncdb_to_pdb.sql` script whilst connected to the PDB. The execution may last for more than 30 minutes.
   a. Connect to `pdb_orcl2` as `SYSDBA`.
      1) Use netca to add the `PDB_ORCL2` net service name for `pdb_orcl2` pluggable database of `cdb2` in the `tnsnames.ora` file.

```
$ netca
```

2) On the Welcome page, select the "Local Net Service Name configuration" and click Next.
3) On the Net Service Name Configuration page, accept `Add` and click Next.
4) On the Net Service Name Configuration, Service Name page, enter `pdb_orcl2` as Service Name and click Next.
5) On the Net Service Name Configuration, Select Protocols page, select TCP and click Next.
6) On the Net Service Name Configuration, TCP/IP Protocol page, enter your complete host name, for example, *<yourservername>*, or *localhost,* accept "Use the standard port number of 1521," and click Next.
7) On the Net Service Name Configuration, Test page, select "No, do not test" (the pluggable database is not yet opened) and click Next.
8) On the Net Service Name Configuration, Net Service Name page, accept `pdb_orcl2` as Net Service Name and click Next.
9) On the Net Service Name Configuration, Another Net Service page, select No, and Next.
10) On the Net Service Name Configuration Complete page, click Next.
11) When you are back on the Welcome page, click Finish.

b. Now connect to `pdb_orcl2` using the net service name.

```
$ sqlplus sys/oracle_4U@pdb_orcl2 as sysdba

Connected to:
Oracle Database 12c Enterprise Edition Release 12.1.0.1.0 -
64bit Production
With the Partitioning, OLAP, Advanced Analytics and Real
Application Testing options

SQL>
```

c. Execute the `$ORACLE_HOME/rdbms/admin/noncdb_to_pdb.sql` script. Expect more than 30 minutes to complete. While the execution will proceed, you can view the `NEW` status change to `CONVERTING` status of the new plugged PDB in `CDB_PDBs` view. In a parallel session connected as `SYS` to root, be ready to run the following statement:

`SELECT pdb_name, status FROM cdb_pdbs;`

```
SQL> @$ORACLE_HOME/rdbms/admin/noncdb_to_pdb.sql
SQL> SET SERVEROUTPUT ON
SQL> SET FEEDBACK 1
SQL> SET NUMWIDTH 10
SQL> SET LINESIZE 80
SQL> SET TRIMSPOOL ON
SQL> SET TAB OFF
SQL> SET PAGESIZE 100
```

```
SQL>
SQL> WHENEVER SQLERROR EXIT;
SQL>
SQL> DOC
DOC>#####################################################################
###########
DOC>#####################################################################
###########
DOC>   The following statement will cause an "ORA-01722: invalid
number"
DOC>   error if we're not in a PDB.
DOC>#####################################################################
###########
DOC>#####################################################################
###########
DOC>#
SQL>
SQL> VARIABLE pdbname VARCHAR2(128)
SQL> BEGIN
  2    SELECT sys_context('USERENV', 'CON_NAME')
  3      INTO :pdbname
  4      FROM dual
  5      WHERE sys_context('USERENV', 'CON_NAME') <> 'CDB$ROOT';
  6  END;
  7  /

PL/SQL procedure successfully completed.

…

SQL>
SQL> Rem
=====================================================================
=
SQL> Rem Run component validation procedure
SQL> Rem
=====================================================================
=
SQL>
SQL> EXECUTE dbms_registry_sys.validate_components;
...Database user "SYS", database schema "APEX_040200", user#
"98" 10:21:02
...Compiled 0 out of 2998 objects considered, 0 failed
compilation 10:21:03
```

```
...263 packages
...255 package bodies
...453 tables
...11 functions
...16 procedures
...3 sequences
...458 triggers
...1322 indexes
...207 views
...0 libraries
...6 types
...0 type bodies
...0 operators
...0 index types
...Begin key object existence check 10:21:03
...Completed key object existence check 10:21:03
...Setting DBMS Registry 10:21:03
...Setting DBMS Registry Complete 10:21:03
...Exiting validate 10:21:03

PL/SQL procedure successfully completed.

SQL> SET serveroutput off
SQL>
SQL> Rem
================================================================
=
SQL> Rem END utlrp.sql
SQL> Rem
================================================================
=
SQL>
…

SQL> alter pluggable database "&pdbname" close;

Pluggable database altered.

SQL>
SQL> alter session set container="&pdbname";

Session altered.
```

```
SQL> -- leave the PDB in the same state it was when we started
SQL> BEGIN
  2    execute immediate '&open_sql &restricted_state';
  3  EXCEPTION
  4    WHEN OTHERS THEN
  5    BEGIN
  6      IF (sqlcode <> -900) THEN
  7        RAISE;
  8      END IF;
  9    END;
 10  END;
 11  /

PL/SQL procedure successfully completed.
SQL>
SQL> WHENEVER SQLERROR CONTINUE;
SQL>
```

If you ran the statement while the PDB was converting, you would have seen the following result:

```
SQL> SELECT  pdb_name, status  FROM cdb_pdbs;

PDB_NAME                                 STATUS
---------------------------------------- -------------
PDB2_2                                   NORMAL
PDB$SEED                                 NORMAL
PDB2_1                                   NORMAL
PDB_ORCL2                                CONVERTING

SQL>
```

d. When the conversion is completed, open the PDB and quit the session.

```
SQL>  alter pluggable database pdb_orcl2 open;

Pluggable database altered.

SQL> EXIT
$
```

4. Connect to `PDB_ORCL2`.

```
$ sqlplus sys/oracle_4U@localhost:1521/PDB_ORCL2 as SYSDBA

Connected to:
Oracle Database 12c Enterprise Edition Release 12.1.0.1.0 -
64bit Production
```

```
With the Partitioning, OLAP, Advanced Analytics and Real
Application Testing options


SQL>
```

5. Verify that the application data is in the PDB `pdb_orcl2`:

```
SQL> select count(empno) from scott.emp;


COUNT(EMPNO)
------------
          14
SQL> EXIT
$
```

Practice 3-6: Merging All PDBs of CDBs into a Single CDB

Overview

In this practice, you merge all PDBs of `cdb1` into a single CDB, `cdb2`.

1. Merge all PDBs of `cdb1` into `cdb2`.
2. Drop `cdb1`.

Assumptions

The CDB `cdb2` exists. The `cdb2` creation has completed successfully in Practice 3-1.

Tasks

1. Connect to the multitenant container database `cdb1` to unplug all PDBs.
   a. Connect to `cdb1` root as a common user with `ALTER PLUGGABLE DATABASE` privilege to unplug `pdb1_1`. If the `pdb1_1` is still in `READ WRITE` mode, close the PDB. Verify the `STATUS` of the PDB when it has been unplugged in `CDB_PDBS` view.

```
$ . oraenv
ORACLE_SID = [cdb2] ? cdb1
The Oracle base remains unchanged with value /u01/app/oracle

$ sqlplus / as sysdba

Connected to:
Oracle Database 12c Enterprise Edition Release 12.1.0.1.0 -
64bit Production
With the Partitioning, OLAP, Advanced Analytics and Real
Application Testing options

SQL> select name, open_mode from v$pdbs;

NAME                           OPEN_MODE
------------------------------ ----------
PDB$SEED                       READ ONLY
PDB1_1                         READ WRITE

SQL> alter pluggable database PDB1_1 unplug
  2  into 'xmlfilePDB1_1.xml';
alter pluggable database PDB1_1 unplug
*
ERROR at line 1:
ORA-65025: Pluggable database PDB1_1 is not closed on all
instances.

SQL> alter pluggable database PDB1_1 close immediate;
```

```
Pluggable database altered.

SQL>  alter pluggable database PDB1_1 unplug into
'xmlfilePDB1_1.xml';

Pluggable database altered.

SQL> col PDB_NAME format A20
SQL> select PDB_NAME, STATUS from CDB_PDBS
     where PDB_NAME='PDB1_1';

PDB_NAME             STATUS
-------------------- -------------
PDB1_1               UNPLUGGED

SQL> drop pluggable database PDB1_1 KEEP DATAFILES;

Pluggable database dropped.

SQL> EXIT
$
```

b. Before plugging `pdb1_1` into `cdb2`, you can optionally check whether the unplugged `pdb1_1` is compatible with `cdb2` with `DBMS_PDB.CHECK_PLUG_COMPATIBILITY` function. Connect to `cdb2` root as a common user with `CREATE PLUGGABLE DATABASE` privilege to plug `pdb1_1`.

Use the following PL/SQL code:

```
DECLARE
  compat BOOLEAN := FALSE;
  BEGIN
  compat := DBMS_PDB.CHECK_PLUG_COMPATIBILITY(
  pdb_descr_file =>
'/u01/app/oracle/product/12.1.0/dbhome_1/dbs/xmlfilePDB1_1.xml',
pdb_name => 'pdb1_1');
  if compat then
  DBMS_OUTPUT.PUT_LINE('Is pluggable compatible? YES');
  else DBMS_OUTPUT.PUT_LINE('Is pluggable compatible? NO');
  end if;
end;
/
```

```
$ . oraenv
ORACLE_SID = [cdb1] ? cdb2
The Oracle base remains unchanged with value /u01/app/oracle
```

```
$ sqlplus / as sysdba

Connected to:
Oracle Database 12c Enterprise Edition Release 12.1.0.1.0 -
64bit Production
With the Partitioning, OLAP, Advanced Analytics and Real
Application Testing options

SQL> SET SERVEROUTPUT ON
SQL> DECLARE
  2  compat BOOLEAN := FALSE;
  3  BEGIN
  4   compat := DBMS_PDB.CHECK_PLUG_COMPATIBILITY(
  5  pdb_descr_file =>
'/u01/app/oracle/product/12.1.0/dbhome_1/dbs/xmlfilePDB1_1.xml',
pdb_name => 'pdb1_1');
  6  if compat then
  7       DBMS_OUTPUT.PUT_LINE('Is pluggable compatible? YES');
  8  else DBMS_OUTPUT.PUT_LINE('Is pluggable compatible? NO');
  9  end if;
 10  end;
 11  /
Is pluggable compatible? NO

PL/SQL procedure successfully completed.

SQL>
```

c. If the value returned is YES, you can immediately proceed to step d.
If the value returned is NO, examine the `PDB_PLUG_IN_VIOLATIONS` view to see why it is not compatible.

```
SQL> select message, action from pdb_plug_in_violations
  2  where name='PDB1_1';

MESSAGE
--------------------------------------------------------------------
-
ACTION
--------------------------------------------------------------------
-
CDB parameter sga_target mismatch: Previous 503316480 Current
2466250752
Please check the parameter in the current CDB
```

```
CDB parameter pga_aggregate_target mismatch: Previous 167772160
Current 81788928
Please check the parameter in the current CDB


SQL>
```

The message refers to a parameter related to PGA. The parameter will not have any impact if you create the PDB. You can proceed with the creation of the PDB.

d. Plug `pdb1_1` into `cdb2`.

```
SQL> create pluggable database pdb1_1 using 'xmlfilePDB1_1.xml'
NOCOPY;


Pluggable database created.

SQL>
```

Notice that you use the clause `NOCOPY` because the `cdb2 pdb1_1` files are located in the right place. Otherwise, you should have described the target destination to move the files from the source to the new destination.

e. Open `pdb1_1` in `cdb2`.

```
SQL> alter pluggable database pdb1_1 open;


Pluggable database altered.

SQL>
```

f. Check that the `pdb1_1` is in the PDBs list in `cdb2`.

```
SQL> select name, open_mode from v$pdbs;


NAME                           OPEN_MODE
------------------------------ ----------
PDB$SEED                       READ ONLY
PDB2_1                         READ WRITE
PDB2_2                         READ WRITE
PDB_ORCL2                      READ WRITE
PDB1_1                         READ WRITE

SQL> EXIT
$
```

2. After all PDBs except `pdb1_2` are unplugged from `cdb1` (in case you had created other PDBs) and plugged into `cdb2`, you can drop the multitenant container database `cdb1` with DBCA or SQL commands.

```
$ . oraenv
ORACLE_SID = [cdb2] ? cdb1
The Oracle base remains unchanged with value /u01/app/oracle
```

```
$ sqlplus / as sysdba

Connected to:
Oracle Database 12c Enterprise Edition Release 12.1.0.1.0 -
64bit Production
With the Partitioning, OLAP, Advanced Analytics and Real
Application Testing options

SQL> shutdown immediate
Database closed.
Database dismounted.
ORACLE instance shut down.

SQL> startup mount restrict
ORACLE instance started.

Total System Global Area  722366464 bytes
Fixed Size                  2276928 bytes
Variable Size             213909952 bytes
Database Buffers          503316480 bytes
Redo Buffers                2863104 bytes
Database mounted.

SQL> DROP DATABASE;

Database dropped.

SQL> EXIT
$
```

Remove archived logs and backups if necessary.

Practices for Lesson 4: Managing a Multitenant Container Database and Pluggable Database

Chapter 4

Practices for Lesson 4: Overview

Overview

In this practice, you will perform startup and shutdown operations on CDBs, open and close operations on PDBs, and connections to PDBs to display current context.

Assumptions

`cdb2` is successfully created after Practice 3-1.

`pdb2_1` is successfully created in `cdb2` after completion of Practice 3-3.

`pdb2_2` is successfully created in `cdb2` after completion of Practice 3-4.

It is not necessary at this step to have successfully created `pdb1_1` and `pdb_orcl2`.

Practice 4-1: Shutdown and Startup of the CDB

Overview

In this practice, you will shut down `cdb2` and start up `cdb2`.

Tasks

1. Connect to the container database `cdb2` to shut it down.
   a. Connect to the CDB as a user with `SYSDBA` privilege.

```
$ . oraenv
ORACLE_SID = [cdb1] ? cdb2
The Oracle base remains unchanged with value /u01/app/oracle

$ sqlplus / as sysdba

Connected to:
Oracle Database 12c Enterprise Edition Release 12.1.0.1.0 -
64bit Production
With the Partitioning, OLAP, Advanced Analytics and Real
Application Testing options

SQL> select name, cdb, con_id from v$database;
NAME      CDB     CON_ID
--------- --- ----------
CDB2      YES          0

SQL>
```

   b. Shut down the CDB.

```
SQL> shutdown immediate
Database closed.
Database dismounted.
ORACLE instance shut down.

SQL> EXIT
$
```

   c. Explore the background processes.

```
$ ps -ef|grep cdb2
oracle   17296 16741  0 03:41 pts/13   00:00:00 grep cdb2
$
```

2. Connect to the container database `cdb2` and start it up.

```
$ sqlplus / as sysdba
```

```
SQL*Plus: Release 12.1.0.1.0 Production on Wed Oct 31 03:41:30
2012

Copyright (c) 1982, 2012, Oracle.  All rights reserved.

Connected to an idle instance.

SQL> startup
ORACLE instance started.

Total System Global Area 1068937216 bytes
Fixed Size                  2248280 bytes
Variable Size             343933352 bytes
Database Buffers          717225984 bytes
Redo Buffers                5529600 bytes
Database mounted.
Database opened.

SQL> select name, cdb, con_id from v$database;
NAME      CDB     CON_ID
--------- --- ----------
CDB2      YES          0

SQL> EXIT
$
```

3. Explore the background processes.

```
$ ps -ef|grep cdb2
oracle    9935     1  0 13:54 ?        00:00:00 ora_pmon_cdb2
oracle    9937     1  0 13:54 ?        00:00:00 ora_psp0_cdb2
oracle    9939     1  0 13:54 ?        00:00:00 ora_vktm_cdb2
oracle    9943     1  0 13:54 ?        00:00:00 ora_gen0_cdb2
oracle    9945     1  0 13:54 ?        00:00:00 ora_mman_cdb2
oracle    9949     1  0 13:54 ?        00:00:00 ora_diag_cdb2
oracle    9951     1  0 13:54 ?        00:00:00 ora_ofsd_cdb2
oracle    9953     1  0 13:54 ?        00:00:00 ora_dbrm_cdb2
oracle    9955     1  0 13:54 ?        00:00:00 ora_dia0_cdb2
oracle    9957     1  0 13:54 ?        00:00:00 ora_dbw0_cdb2
oracle    9959     1  0 13:54 ?        00:00:00 ora_lgwr_cdb2
oracle    9961     1  0 13:54 ?        00:00:00 ora_ckpt_cdb2
oracle    9963     1  0 13:54 ?        00:00:00 ora_lg00_cdb2
oracle    9965     1  0 13:54 ?        00:00:00 ora_lg01_cdb2
oracle    9967     1  0 13:54 ?        00:00:00 ora_smon_cdb2
```

```
oracle    9969     1  0 13:54 ?        00:00:00 ora_reco_cdb2
oracle    9971     1  0 13:54 ?        00:00:00 ora_lreg_cdb2
oracle    9973     1  0 13:54 ?        00:00:00 ora_mmon_cdb2
oracle    9977     1  0 13:54 ?        00:00:00 ora_mmnl_cdb2
oracle    9979     1  0 13:54 ?        00:00:00 ora_d000_cdb2
oracle    9990     1  0 13:54 ?        00:00:00 ora_s000_cdb2
oracle   10027     1  0 13:55 ?        00:00:00 ora_tmon_cdb2
oracle   10029     1  0 13:55 ?        00:00:00 ora_tt00_cdb2
oracle   10031     1  0 13:55 ?        00:00:00 ora_smco_cdb2
oracle   10059     1  0 13:55 ?        00:00:00 ora_w000_cdb2
oracle   10061     1  0 13:55 ?        00:00:00 ora_aqpc_cdb2
oracle   10077     1  0 13:55 ?        00:00:00 ora_p000_cdb2
oracle   10079     1  0 13:55 ?        00:00:00 ora_p001_cdb2
oracle   10081     1  0 13:55 ?        00:00:00 ora_p002_cdb2
oracle   10083     1  0 13:55 ?        00:00:00 ora_p003_cdb2
oracle   10087     1  0 13:55 ?        00:00:00 ora_p004_cdb2
oracle   10090     1  0 13:55 ?        00:00:00 ora_p005_cdb2
oracle   10092     1  0 13:55 ?        00:00:00 ora_p006_cdb2
oracle   10095     1  0 13:55 ?        00:00:00 ora_p007_cdb2
oracle   10102     1  0 13:55 ?        00:00:00 oraclecdb2
(LOCAL=NO)
oracle   10107     1  0 13:55 ?        00:00:00 ora_qm02_cdb2
oracle   10112     1  6 13:55 ?        00:00:00 ora_qm00_cdb2
oracle   10115     1  0 13:55 ?        00:00:00 ora_q002_cdb2
oracle   10117     1  0 13:55 ?        00:00:00 ora_qm03_cdb2
oracle   10119     1  0 13:55 ?        00:00:00 ora_q004_cdb2
oracle   10131     1  0 13:55 ?        00:00:00 ora_cjq0_cdb2
oracle   10135 23215  0 13:55 pts/4    00:00:00 grep cdb2
$
```

4. Explore the PDBs.

```
$ sqlplus / as sysdba


Connected to:
Oracle Database 12c Enterprise Edition Release 12.1.0.1.0 -
64bit Production
With the Partitioning, OLAP, Advanced Analytics and Real
Application Testing options


SQL> select CON_ID, NAME, OPEN_MODE from v$pdbs;


    CON_ID NAME                           OPEN_MODE
---------- ------------------------------ ----------
         2 PDB$SEED                       READ ONLY
```

```
      3 PDB2_1                                  MOUNTED
      4 PDB2_2                                  MOUNTED
      5 PDB_ORCL2                               MOUNTED
      6 PDB1_1                                  MOUNTED

SQL>
```

5. Open all PDBs.

```
SQL> alter pluggable database all open;

Pluggable database altered.

SQL> select CON_ID, NAME, OPEN_MODE from v$pdbs;

    CON_ID NAME                           OPEN_MODE
---------- ------------------------------ ----------
      2 PDB$SEED                           READ ONLY
      3 PDB2_1                             READ WRITE
      4 PDB2_2                             READ WRITE
      5 PDB_ORCL2                          READ WRITE
      6 PDB1_1                             READ WRITE

SQL>
```

6. Connect to any of the PDBs in your `cdb2`, except `PDB$SEED`.

```
SQL> connect sys/oracle_4U@PDB2_1 AS SYSDBA
Connected.
SQL> select CON_ID, NAME, OPEN_MODE from v$pdbs;

    CON_ID NAME                           OPEN_MODE
---------- ------------------------------ ----------
         3 PDB2_1                         READ WRITE

SQL>
```

7. Display the context of the PDB you are connected to.

```
SQL> show con_name

CON_NAME
------------------------------
PDB2_1
SQL>
```

8. Connect to another PDB left in your `cdb2`, except `PDB$SEED`.

```
SQL> connect sys/oracle_4U@PDB2_2 AS SYSDBA
```

```
Connected.
SQL> select CON_ID, NAME, OPEN_MODE from v$pdbs;

    CON_ID NAME                           OPEN_MODE
---------- ------------------------------ ----------
         4 PDB2_2                         READ WRITE

SQL>
```

9. Display the context of the PDB you are connected to.

```
SQL> show con_name

CON_NAME
------------------------------
PDB2_2

SQL> EXIT
$
```

Practice 4-2: Closing and Opening a PDB

Overview

In this practice, you will close PDBs and open PDB.

Tasks

1. Connect to the multitenant container database `cdb2` to shut it down.
   a. Connect to `cdb2` as a user with `SYSDBA` privilege.

```
$ sqlplus / as sysdba

Connected to:
Oracle Database 12c Enterprise Edition Release 12.1.0.1.0 -
64bit Production
With the Partitioning, OLAP, Advanced Analytics and Real
Application Testing options

SQL> select name, cdb, con_id from v$database;

NAME      CDB     CON_ID
--------- --- ----------
CDB2      YES          0

SQL>
```

   b. Shut down `cdb2`.

```
SQL> shutdown immediate
Database closed.
Database dismounted.
ORACLE instance shut down.
SQL>
```

   c. Start up `cdb2`.

```
SQL> startup
ORACLE instance started.

Total System Global Area 1068937216 bytes
Fixed Size                  2248280 bytes
Variable Size             343933352 bytes
Database Buffers          717225984 bytes
Redo Buffers                5529600 bytes
Database mounted.
Database opened.
SQL>
```

   d. Notice that the PDBs are all in `MOUNTED` open mode.

```
SQL> select CON_ID, NAME, OPEN_MODE from v$pdbs;
```

```
    CON_ID NAME                           OPEN_MODE
---------- ------------------------------ ----------
         2 PDB$SEED                       READ ONLY
         3 PDB2_1                         MOUNTED
         4 PDB2_2                         MOUNTED
         5 PDB_ORCL2                      MOUNTED
         6 PDB1_1                         MOUNTED

SQL>
```

2. Open all PDBs manually.

```
SQL> alter pluggable database all open;

Pluggable database altered.

SQL>
```

3. Close PDB2_1.
   a. Start a DML transaction in another session.

```
$ . oraenv
ORACLE_SID = [oracle] ? cdb2
The Oracle base remains unchanged with value /u01/app/oracle
$ sqlplus sys/oracle_4U@pdb2_1 as sysdba

Connected to:
Oracle Database 12c Enterprise Edition Release 12.1.0.1.0 -
64bit Production
With the Partitioning, OLAP, Advanced Analytics and Real
Application Testing options
Connected.

SQL> create table system.mytab (c number);

Table created.

SQL> insert into system.mytab values (1);

1 row created.

SQL> COMMIT;

Commit complete.
```

```
SQL> exit
$
```

b. In the first session, close PDB2_1 in IMMEDIATE mode.

```
SQL> alter pluggable database pdb2_1 close immediate;

Pluggable database altered.

SQL> select CON_ID, NAME, OPEN_MODE from v$pdbs;

    CON_ID NAME                           OPEN_MODE
---------- ------------------------------ ----------
         2 PDB$SEED                       READ ONLY
         3 PDB2_1                         MOUNTED
         4 PDB2_2                         READ WRITE
         5 PDB_ORCL2                      READ WRITE
         6 PDB1_1                         READ WRITE

SQL>
```

c. Try to connect as a user of PDB2_1.

```
SQL> connect system/oracle_4U@pdb2_1
ERROR:
ORA-01033: ORACLE initialization or shutdown in progress
Process ID: 0
Session ID: 0 Serial number: 0


Warning: You are no longer connected to ORACLE.
SQL>
```

4. Open pdb2_1.

```
SQL> connect / as sysdba
Connected.
SQL> alter pluggable database PDB2_1 open;

Pluggable database altered.

SQL>
```

Reconnect to pdb2_1 and select data from SYSTEM.MYTAB table.

```
SQL> connect system/oracle_4U@PDB2_1
Connected.
SQL> select * from system.mytab;
```

```
         C
----------
         1

SQL>
```

5. Shut down the multitenant container database `cdb2` to open and close PDBs with different clauses.

```
SQL> CONNECT / AS SYSDBA
Connected.

SQL> select name, cdb, con_id from v$database;

NAME      CDB     CON_ID
--------- --- ----------
CDB2      YES          0

SQL>
```

a. Shut down `CDB2`.

```
SQL> shutdown immediate
Database closed.
Database dismounted.
ORACLE instance shut down.
SQL>
```

b. Start up `cdb2` in `NOMOUNT` mode.

```
SQL> startup NOMOUNT
ORACLE instance started.

Total System Global Area 1068937216 bytes
Fixed Size                  2248280 bytes
Variable Size             343933352 bytes
Database Buffers          717225984 bytes
Redo Buffers                5529600 bytes
SQL> select CON_ID, NAME, OPEN_MODE from v$pdbs;

No rows selected.

SQL>
```

c. Mount `cdb2`.

```
SQL> alter database mount;
```

```
Database altered.

SQL>
```

```
SQL> select CON_ID, NAME, OPEN_MODE from v$pdbs;

    CON_ID NAME                           OPEN_MODE
---------- ------------------------------ ----------
         2 PDB$SEED                       MOUNTED
         3 PDB2_1                         MOUNTED
         4 PDB2_2                         MOUNTED
         5 PDB_ORCL2                      MOUNTED
         6 PDB1_1                         MOUNTED

SQL>
```

d. Open `cdb2`.

```
SQL> alter database open;

Database altered.

SQL>
```

```
SQL> select CON_ID, NAME, OPEN_MODE from v$pdbs;

    CON_ID NAME                           OPEN_MODE
---------- ------------------------------ ----------
         2 PDB$SEED                       READ ONLY
         3 PDB2_1                         MOUNTED
         4 PDB2_2                         MOUNTED
         5 PDB_ORCL2                      MOUNTED
         6 PDB1_1                         MOUNTED
SQL>
```

e. Open all PDBs except `PDB2_2`.

```
SQL> alter pluggable database all except pdb2_2 open;

Pluggable database altered.

SQL>
```

```
SQL> select CON_ID, NAME, OPEN_MODE from v$pdbs;
```

```
    CON_ID NAME                                   OPEN_MODE
---------- ------------------------------ ----------
         2 PDB$SEED                                READ ONLY
         3 PDB2_1                                  READ WRITE
         4 PDB2_2                                  MOUNTED
         5 PDB_ORCL2                               READ WRITE
         6 PDB1_1                                  READ WRITE

SQL>
```

6. Close all pluggable databases except `pdb2_1` and `pdb1_1`.

```
SQL> alter pluggable database all except pdb2_1, pdb1_1 close;

Pluggable database altered.

SQL>
```

```
SQL> select CON_ID, NAME, OPEN_MODE from v$pdbs;

    CON_ID NAME                                   OPEN_MODE
---------- ------------------------------ ----------
     2 PDB$SEED                                READ ONLY
     3 PDB2_1                                  READ WRITE
     4 PDB2_2                                  MOUNTED
     5 PDB_ORCL2                               MOUNTED
     6 PDB1_1                                  READ WRITE

SQL>
```

Practice 4-3: Creating After Startup Trigger to Open All PDBs

Overview

In this practice, you will create `AFTER STARTUP` trigger to open all PDBs of a CDB.

Tasks

1. Create a trigger in `cdb2` to open all PDBs automatically after starting up `cdb2`.
   a. Create the trigger.

```
CREATE TRIGGER open_all_PDBs
  AFTER STARTUP ON DATABASE
begin
    execute immediate 'alter pluggable database all open';
 end open_all_PDBs;
/
```

```
SQL> CREATE TRIGGER open_all_PDBs
  2  AFTER STARTUP ON DATABASE
  3  begin
  4      execute immediate 'alter pluggable database all open';
  5  end open_all_PDBs;
  6  /

Trigger created.

SQL>
```

   b. Shut down `cdb2`.

```
SQL> shutdown immediate
Database closed.
Database dismounted.
ORACLE instance shut down.
SQL>
```

   c. Start up `cdb2`.

```
SQL> startup
ORACLE instance started.

Total System Global Area 1068937216 bytes
Fixed Size                  2248280 bytes
Variable Size             343933352 bytes
Database Buffers          717225984 bytes
Redo Buffers                5529600 bytes
Database mounted.
Database opened.
SQL>
```

d. Notice that the PDBs are all in `READ WRITE` open mode.

```
SQL> select CON_ID, NAME, OPEN_MODE from v$pdbs;

    CON_ID NAME                           OPEN_MODE
---------- ------------------------------ ----------
         2 PDB$SEED                       READ ONLY
         3 PDB2_1                         READ WRITE
         4 PDB2_2                         READ WRITE
         5 PDB_ORCL2                      READ WRITE
         6 PDB1_1                         READ WRITE

SQL>
```

Practice 4-4: Changing PDBs' Open Mode

Overview

In this practice, you will change the open mode of PDBs for specific operations.

Assumptions

If the trigger could not be created successfully, execute the following catchup script:

```
$ cd /home/oracle/solutions/catchup_04_03
$ ./cr_trig.sh
$
```

Tasks

Rename the global database `pdb2_1 as pdb2` in `cdb2`. For this purpose, you must open the PDB in `RESTRICTED` mode.

1. Connect to `pdb2_1`.

```
SQL> CONNECT sys/oracle_4U@pdb2_1 as sysdba
Connected.
SQL>
```

2. Change the global database name for `pdb2_1` to `pdb2`.

```
SQL> alter pluggable database RENAME GLOBAL_NAME TO pdb2;
alter pluggable database RENAME global_name to pdb2
                                               *
ERROR at line 1:
ORA-65045: pluggable database not in a restricted mode

SQL>
```

3. Close `pdb2_1`.

```
SQL> alter pluggable database close immediate;

Pluggable database altered.

SQL>
```

4. Open `pdb2_1` in restricted mode.

```
SQL> alter pluggable database open restricted;

Pluggable database altered.

SQL>
```

```
SQL> select CON_ID, NAME, OPEN_MODE, RESTRICTED from v$pdbs;

    CON_ID NAME                           OPEN_MODE  RES
```

```
---------- ------------------------------ ---------- ---
         3 PDB2_1                         READ WRITE YES

SQL>
```

5. Change the global database name for `pdb2_1` to `pdb2`.

```
SQL> alter pluggable database RENAME GLOBAL_NAME TO pdb2;

Pluggable database altered.

SQL>
```

```
SQL> select CON_ID, NAME, OPEN_MODE, RESTRICTED from v$pdbs;

    CON_ID NAME                           OPEN_MODE  RES
---------- ------------------------------ ---------- ---
         3 PDB2                           READ WRITE YES

SQL>
```

6. Open `PDB2`.

```
SQL> alter pluggable database close immediate;

Pluggable database altered.

SQL> alter pluggable database open;

Pluggable database altered.

SQL>
```

7. Check `PDB2` is in `READ WRITE` mode.

```
SQL> select CON_ID, NAME, OPEN_MODE, RESTRICTED from v$pdbs;

    CON_ID NAME                           OPEN_MODE  RES
---------- ------------------------------ ---------- ---
         3 PDB2                           READ WRITE NO

SQL>
```

Practice 4-5: Changing Instance Parameter

Overview

In this practice, you will discover the impact of instance parameter changes on PDBs.

Tasks

1. In this example, you will use in `cdb2` the instance parameter `OPTIMIZER_USE_SQL_PLAN_BASELINES` because it is `ISPDB_MODIFIABLE` in `V$PARAMETER`.

```
SQL> CONNECT / AS SYSDBA
Connected.
SQL> select ISPDB_MODIFIABLE from v$parameter
  2  where name='optimizer_use_sql_plan_baselines';

ISPDB
-----
TRUE

SQL>
```

2. Check the current value of instance parameter `OPTIMIZER_USE_SQL_PLAN_BASELINES`.

```
SQL> show parameter optimizer_use_sql_plan_baselines
NAME                                 TYPE        VALUE
------------------------------------ ----------- ---------
optimizer_use_sql_plan_baselines     boolean      TRUE

SQL> EXIT
$
```

3. Connect to `pdb2` in `cdb2` and check the current value of the same instance parameter `OPTIMIZER_USE_SQL_PLAN_BASELINES`.
    1) Use netca to add the `PDB2` net service name for `pdb2` pluggable database of `cdb2` in the `tnsnames.ora` file.

```
$ netca
```

    2) On the Welcome page, select the “Local Net Service Name configuration” and click Next.
    3) On the Net Service Name Configuration page, accept `Add` and click Next.
    4) On the Net Service Name Configuration, Service Name page, enter `pdb2` as Service Name and click Next.
    5) On the Net Service Name Configuration, Select Protocols page, select TCP and click Next.
    6) On the Net Service Name Configuration, TCP/IP Protocol page, enter your complete host name, for example, *<yourservername>*`.us.oracle.com`, or *localhost,* accept “Use the standard port number of 1521,” and click Next.

7) On the Net Service Name Configuration, Test page, select “No, do not test” (the pluggable database is not yet opened) and click Next.
8) On the Net Service Name Configuration, Net Service Name page, accept `pdb2` as Net Service Name and click Next.
9) On the Net Service Name Configuration, Another Net Service page, select No, and Next.
10) On the Net Service Name Configuration Complete page, click Next.
11) When you are back on the Welcome page, click Finish.
12) Reload the listener with the new configuration:

```
$ lsnrctl reload
```

If this is not sufficient, then restart the instance.

a. Connect to `pdb2` in `cdb2`.

```
$ sqlplus sys/oracle_4U@pdb2 AS SYSDBA

Connected to:
Oracle Database 12c Enterprise Edition Release 12.1.0.1.0 -
64bit Production
With the Partitioning, OLAP, Advanced Analytics and Real
Application Testing options

SQL> show parameter optimizer_use_sql_plan_baselines
NAME                                 TYPE        VALUE
------------------------------------ ----------- ---------
optimizer_use_sql_plan_baselines     boolean      TRUE

SQL>
```

4. Change the instance parameter value to `FALSE` in `pdb2`.

```
SQL> ALTER SYSTEM SET optimizer_use_sql_plan_baselines= FALSE
SCOPE=BOTH;

System altered.

SQL> show parameter optimizer_use_sql_plan_baselines
NAME                                 TYPE        VALUE
------------------------------------ ----------- ---------
optimizer_use_sql_plan_baselines     boolean     FALSE

SQL>
```

5. Check the instance parameter value in other PDBs of the same CDB.

```
SQL> CONNECT sys/oracle_4U@pdb2_2 AS SYSDBA
Connected.
SQL> show parameter optimizer_use_sql_plan_baselines
```

```
NAME                                 TYPE        VALUE
------------------------------------ ----------- ---------
optimizer_use_sql_plan_baselines     boolean     TRUE

SQL>
```

6. Close and open `pdb2`.

```
SQL> CONNECT sys/oracle_4U@pdb2 AS SYSDBA
Connected.
SQL> ALTER PLUGGABLE DATABASE CLOSE IMMEDIATE;

Pluggable database altered.

SQL> ALTER PLUGGABLE DATABASE OPEN;

Pluggable database altered.

SQL> show parameter optimizer_use_sql_plan_baselines

NAME                                 TYPE        VALUE
------------------------------------ ----------- ---------
optimizer_use_sql_plan_baselines     boolean     FALSE

SQL>
```

7. Check the instance parameter value after CDB shutdown/startup both in root and PDBs.

```
SQL> connect / as sysdba
Connected.
SQL> shutdown immediate
Database closed.
Database dismounted.
ORACLE instance shut down.
SQL>
SQL> startup
ORACLE instance started.

Total System Global Area 1068937216 bytes
Fixed Size                  2248280 bytes
Variable Size             343933352 bytes
Database Buffers          717225984 bytes
Redo Buffers                5529600 bytes
Database mounted.
Database opened.
SQL> col VALUE format a20
```

```
SQL> select CON_ID, VALUE from   V$SYSTEM_PARAMETER
  2  where name ='optimizer_use_sql_plan_baselines';

    CON_ID VALUE
---------- --------------------
         0 TRUE
         3 FALSE

SQL>
```

Practice 4-6: Changing Operations Behavior in a PDB

Overview

In this practice, you will restrict sessions in a particular PDB.

Tasks

1. Restrict sessions in `pdb2_2` only and not in other PDBs.
   a. Enable restricted session in `pdb2_2`.

```
SQL> CONNECT sys/oracle_4U@pdb2_2 AS SYSDBA
Connected.
SQL> ALTER SYSTEM ENABLE RESTRICTED SESSION;

System altered.

SQL>
```

   b. Create a user in `pdb2_2`.

```
SQL> CREATE USER u1_pdb2_2 IDENTIFIED BY oracle_4U;

User created.

SQL> GRANT create session TO u1_pdb2_2;

Grant succeeded.

SQL>
```

   c. Attempt to connect as user `u1_pdb2_2` in `pdb2_2`.

```
SQL> CONNECT u1_pdb2_2/oracle_4U@pdb2_2
ERROR:
ORA-01035: ORACLE only available to users with RESTRICTED
SESSION privilege


Warning: You are no longer connected to ORACLE.
SQL>
```

2. Create a user in another PDB, `pdb2`.

```
SQL> CONNECT sys/oracle_4U@pdb2 AS SYSDBA
Connected.
SQL>
SQL> CREATE USER u1_pdb2 IDENTIFIED BY oracle_4U;

User created.
```

```
SQL> GRANT create session TO u1_pdb2;

Grant succeeded.

SQL>
```

a. Connect as user `u1_pdb2` in `pdb2`.

```
SQL> CONNECT u1_pdb2/oracle_4U@pdb2
Connected.
SQL>
```

The restriction impacts only `pdb2_2` as expected.

3. Remove the restriction in `pdb2_2`.

```
SQL> CONNECT sys/oracle_4U@pdb2_2 AS SYSDBA
Connected.
SQL> alter pluggable database close;

Pluggable database altered.

SQL> alter pluggable database open;
```

```
Pluggable database altered.

SQL> CONNECT u1_pdb2_2/oracle_4U@pdb2_2
Connected.
SQL>
```

4. Drop the users created.
   a. Drop the `u1_pdb2_2` user in `pdb2_2`.

```
SQL> CONNECT sys/oracle_4U@pdb2_2 AS SYSDBA
Connected.
SQL> DROP USER u1_pdb2_2;

User dropped.


SQL>
```

   b. Drop the `u1_pdb2` user in `pdb2`.

```
SQL> CONNECT sys/oracle_4U@pdb2 AS SYSDBA
Connected.
SQL> DROP USER u1_pdb2;

User dropped.

SQL> EXIT
$
```

Practices for Lesson 5: Managing Tablespaces in CDB and PDBs

Chapter 5

Practices for Lesson 5: Overview

Practices Overview

In this practice, you will manage the tablespaces in the root and in the PDBs.

Assumptions

Practice 3-1 successfully created `cdb2`.

Practice 3-3 successfully created `pdb2_1`.

Practice 4-4 successfully renamed `pdb2_1` to `pdb2`.

If the trigger could not be created successfully, execute the following catchup script:

```
$ cd /home/oracle/solutions/catchup_04_03
$ ./cr_trig.sh
```

Practice 5-1: Managing Permanent and Temporary Tablespaces

Overview

In this practice, you will manage the permanent and temporary tablespaces in the root and in the PDBs.

Tasks

1. View permanent and temporary tablespaces properties in `cdb2`.

```
$ . oraenv
ORACLE_SID = [cdb2] ? cdb2
The Oracle base remains unchanged with value /u01/app/oracle

$ sqlplus / as sysdba

Connected to:
Oracle Database 12c Enterprise Edition Release 12.1.0.1.0 -
64bit Production
With the Partitioning, OLAP, Advanced Analytics and Real
Application Testing options

SQL> col PROPERTY_NAME format a30
SQL> col PROPERTY_VALUE  format a25
SQL> SELECT property_name, property_value
  2  FROM database_properties
  3  WHERE property_name LIKE 'DEFAULT_%TABLE%';

PROPERTY_NAME                  PROPERTY_VALUE
------------------------------ ---------------------------------
DEFAULT_TEMP_TABLESPACE        TEMP
DEFAULT_PERMANENT_TABLESPACE   USERS

SQL> SELECT tablespace_name, CON_ID from CDB_TABLESPACES;

TABLESPACE_NAME                     CON_ID
------------------------------ ----------
SYSTEM                                  1
SYSAUX                                  1
UNDOTBS1                                1
TEMP                                    1
USERS                                   1
SYSTEM                                  2
SYSAUX                                  2
TEMP                                    2
```

```
SYSTEM                                      3
SYSAUX                                      3
TEMP                                        3
SYSTEM                                      4
SYSAUX                                      4
TEMP                                        4
SYSTEM                                      5
SYSAUX                                      5
TEMP                                        5
USERS                                      5
EXAMPLE                                    5
SYSTEM                                      6
SYSAUX                                      6
TEMP                                        6
USERS                                      6
EXAMPLE                                    6

24 rows selected.

SQL> SELECT tablespace_name, CON_ID from CDB_TABLESPACES
  2  WHERE TABLESPACE_NAME LIKE 'TEMP%';

TABLESPACE_NAME                     CON_ID
------------------------------ ----------
TEMP                                    1
TEMP                                    2
TEMP                                    3
TEMP                                    4
TEMP                                    5
TEMP                                    6

6 rows selected.

SQL>
```

2. Create a permanent tablespace CDATA in the root container.

```
SQL> CREATE TABLESPACE CDATA
  2  DATAFILE '/u01/app/oracle/oradata/cdb2/cdata_01.dbf'
  3  SIZE 10M ;

Tablespace created.

SQL> SELECT tablespace_name, CON_ID from CDB_TABLESPACES
```

```
  2  WHERE TABLESPACE_NAME = 'CDATA';

TABLESPACE_NAME                    CON_ID
------------------------------ ----------
CDATA                                   1

SQL>
```

3. Make the CDATA tablespace the default tablespace in the root container.

```
SQL> ALTER DATABASE DEFAULT TABLESPACE CDATA ;

Database altered.

SQL> SELECT property_name, property_value
  2  FROM   database_properties
  3  WHERE property_name LIKE 'DEFAULT_%TABLE%';

PROPERTY_NAME                  PROPERTY_VALUE
------------------------------ --------------------------------
DEFAULT_TEMP_TABLESPACE        TEMP
DEFAULT_PERMANENT_TABLESPACE   CDATA

SQL>
```

4. Create a permanent tablespace, LDATA, in PDB2.

```
SQL> connect system/oracle_4U@PDB2
Connected.

SQL> CREATE TABLESPACE ldata DATAFILE
  2  '/u01/app/oracle/oradata/cdb2/pdb2_1/ldata_01.dbf'
  3  SIZE 10M ;

Tablespace created.

SQL>
```

5. Make the LDATA tablespace the default tablespace in the PDB2 container.

```
SQL> ALTER PLUGGABLE DATABASE DEFAULT TABLESPACE LDATA ;
Pluggable database altered.

SQL> SELECT property_name, property_value
  2  FROM database_properties
  3  WHERE property_name LIKE 'DEFAULT_%TABLE%';
```

```
PROPERTY_NAME                  PROPERTY_VALUE
------------------------------ ------------------------------
DEFAULT_TEMP_TABLESPACE        TEMP
DEFAULT_PERMANENT_TABLESPACE   LDATA

SQL>
```

6. Create a temporary tablespace in the root container.

```
SQL> connect system/oracle_4U
Connected.
SQL> CREATE TEMPORARY TABLESPACE TEMP_ROOT
  2  TEMPFILE '/u01/app/oracle/oradata/cdb2/temproot_01.dbf'
  3  SIZE 500M ;

Tablespace created.

SQL>
```

7. Make TEMP_ROOT the default temporary tablespace in the root container.

```
SQL> ALTER DATABASE DEFAULT TEMPORARY TABLESPACE TEMP_ROOT ;

Database altered.

SQL> SELECT property_name, property_value
  2  FROM database_properties
  3  WHERE property_name LIKE 'DEFAULT_%TABLE%';

PROPERTY_NAME                  PROPERTY_VALUE
------------------------------ ------------------------------
DEFAULT_TEMP_TABLESPACE        TEMP_ROOT
DEFAULT_PERMANENT_TABLESPACE   CDATA

SQL>
```

8. Create a temporary tablespace TEMP_PDB2 in PDB2.

```
SQL> connect system/oracle_4U@PDB2
Connected.

SQL> CREATE TEMPORARY TABLESPACE TEMP_PDB2 TEMPFILE
  2      '/u01/app/oracle/oradata/cdb2/pdb2_1/temppdb2_01.dbf'
  3      SIZE 100M ;

Tablespace created.
```

```
SQL>
```

9. Make TEMP_PDB2 the default temporary tablespace in PDB2.

```
SQL> ALTER DATABASE DEFAULT TEMPORARY TABLESPACE TEMP_PDB2 ;

Database altered.

SQL> SELECT property_name, property_value
  2  FROM database_properties
  3  WHERE property_name LIKE 'DEFAULT_%TABLE%';

PROPERTY_NAME                  PROPERTY_VALUE
------------------------------ --------------------------------
DEFAULT_TEMP_TABLESPACE        TEMP_PDB2
DEFAULT_PERMANENT_TABLESPACE   LDATA

SQL>
```

Note that you could also use ALTER PLUGGABLE DATABASE command.

10. Create a temporary tablespace MY_TEMP in PDB2.

```
SQL> CREATE TEMPORARY TABLESPACE MY_TEMP TEMPFILE
  2  '/u01/app/oracle/oradata/cdb2/pdb2_1/my_temp_pdb2_01.dbf'
  3  SIZE 10M;

Tablespace created.

SQL>
```

11. Display default tablespaces of another PDB in cdb2.

```
SQL> connect system/oracle_4U@PDB_ORCL2
Connected.

SQL> SELECT property_name, property_value
  2  FROM database_properties
  3  WHERE property_name LIKE 'DEFAULT_%TABLE%';

PROPERTY_NAME                  PROPERTY_VALUE
------------------------------ --------------------------------
DEFAULT_TEMP_TABLESPACE        TEMP
DEFAULT_PERMANENT_TABLESPACE   USERS

SQL>
```

12. Manage default permanent and temporary tablespaces of users.
    a. Create a common user C##U.

```
SQL> connect system/oracle_4U
Connected.

SQL> CREATE USER c##u IDENTIFIED BY x;

User created.

SQL>
```

b. View the default tablespace and temporary tablespace assignment for user CU in all containers.

```
SQL> COLUMN username format A12
SQL> COLUMN default_tablespace format A18
SQL> COLUMN temporary_tablespace format A20
SQL> COLUMN con_id format 999
SQL> SELECT username, default_tablespace,
  2  temporary_tablespace, con_id
  3  FROM CDB_USERS
  4  WHERE username = 'C##U';

USERNAME     DEFAULT_TABLESPACE TEMPORARY_TABLESPACE   CON_ID
------------ ------------------ ---------------------- ------
C##U         CDATA              TEMP_ROOT                   1
C##U         LDATA              TEMP_PDB2                   3
C##U         SYSTEM             TEMP                        4
C##U         USERS              TEMP                        5
C##U         USERS              TEMP                        6

SQL>
```

c. Create a local user LU in PDB2.

```
SQL> connect system/oracle_4U@PDB2
Connected.

SQL> CREATE USER lu IDENTIFIED BY x;

User created.

SQL>
```

d. View the default tablespace and temporary tablespace assignment for user LU.

```
SQL> SELECT username, default_tablespace, temporary_tablespace
  2  FROM DBA_USERS
  3  WHERE username = 'LU';
```

```
USERNAME     DEFAULT_TABLESPACE TEMPORARY_TABLESPACE
------------ ------------------ --------------------
LU           LDATA              TEMP_PDB2

SQL>
```

e. Change the temporary tablespace assignment for user `LU` to `MY_TEMP` in `PDB2`.

```
SQL> ALTER USER lu TEMPORARY TABLESPACE MY_TEMP;

User altered.

SQL>
```

f. View the default temporary tablespace assignment for user LU.

```
SQL> SELECT username, default_tablespace, temporary_tablespace
  2  FROM DBA_USERS
  3  WHERE username = 'LU';

USERNAME     DEFAULT_TABLESPACE TEMPORARY_TABLESPACE
------------ ------------------ --------------------
LU           LDATA              MY_TEMP

SQL>
```

Practice 5-2: Managing UNDO Tablespaces

Overview

In this practice, you manage UNDO tablespaces.

1. Display the UNDO tablespace used in the CDB.

```
SQL> connect system/oracle_4U
Connected.
SQL> col NAME format A12
SQL> select FILE#, ts.name, ts.ts#, ts.con_id
  2  from v$datafile d, v$tablespace  ts
  3  where d.ts#=ts.ts#
  4  and   d.con_id=ts.con_id
  5  and   ts.name like 'UNDO%';

     FILE# NAME                TS#     CON_ID
---------- ------------ ---------- ----------
         4 UNDOTBS1              2          1

SQL>
```

2. Create an UNDO tablespace in a PDB and set it as the UNDO_TABLESPACE of the CDB.

```
SQL> connect system/oracle_4U@PDB2
Connected.
SQL> CREATE UNDO TABLESPACE UNDO_PDB2 DATAFILE
  2  '/u01/app/oracle/oradata/cdb2/pdb2/undo_pdb2_01.dbf'
  3  SIZE 10M;
Tablespace created.

SQL> alter system set undo_tablespace='UNDO_PDB2' scope=both;
alter system set undo_tablespace='UNDO_PDB2' scope=both
*
ERROR at line 1:
ORA-65040: operation not allowed from within a pluggable
database

SQL> EXIT
$
```

Notice that the statement fails because the UNDO tablespace can be set only at CDB level. No tablespace was created.

Practices for Lesson 6: Managing Security in CDB and PDBs

Chapter 6

Practices for Lesson 6: Overview

Overview

In this practice, you will manage the users, privileges, and roles.

Assumption

Practice 3-1 successfully created `cdb2`.

Practice 3-3 successfully created `pdb2_1`.

Practice 4-4 successfully renamed `pdb2_1` to `pdb2`.

Practice 3-4 successfully created `pdb2_2`.

If the trigger could not be created successfully, execute the following catchup script:

```
$ cd /home/oracle/solutions/catchup_04_03
$ ./cr_trig.sh
```

If permanent and temporary tablespaces could not be created successfully, execute the following catchup script:

```
$ cd /home/oracle/solutions/catchup_05_02
$ ./cr_TABLESPACES.sh
```

Practice 6-1: Managing Common and Local Users

Overview

In this practice, you will manage the common and local users in CDB and PDBs.

Tasks

1. View all common and local users in `cdb2`.

```
$ . oraenv

ORACLE_SID = [cdb2] ? cdb2
The Oracle base remains unchanged with value /u01/app/oracle
$ sqlplus / as sysdba

Connected to:
Oracle Database 12c Enterprise Edition Release 12.1.0.1.0 -
64bit Production
With the Partitioning, OLAP, Advanced Analytics and Real
Application Testing options
SQL>
SQL> col username format a20
SQL> select USERNAME, COMMON, CON_ID  from cdb_users;

USERNAME             COM CON_ID
-------------------- --- ------
SYS                  YES      1
SYSTEM               YES      1
C##U                 YES      1
…
C##U                 YES      3
SCOTT                NO       3
…
C##U                 YES      4
…
DVSYS                YES      4
AUDSYS               YES      4
SI_INFORMTN_SCHEMA   YES      4
OLAPSYS              YES      4
…
C##U                 YES      5
…
233 rows selected.

SQL> select USERNAME, COMMON,CON_ID from cdb_users
```

```
  2  where username = 'SYSTEM';

USERNAME             COM CON_ID
-------------------- --- ------
SYSTEM               YES      1
SYSTEM               YES      2
SYSTEM               YES      3
SYSTEM               YES      4
SYSTEM               YES      5
SYSTEM               YES      6

6 rows selected.

SQL> select distinct  username from cdb_users
  2  where common='YES';

USERNAME
--------------------
DVF
SYSKM
DIP
XS$NULL
OUTLN
SYSBACKUP
SYSTEM
ORACLE_OCM
DVSYS
AUDSYS
DBSNMP
…
C##U
XDB
APPQOSSYS
SYSDG
ANONYMOUS
SYS
SI_INFORMTN_SCHEMA
ANONYMOUS
LBACSYS
WMSYS

36 rows selected.
```

```
SQL> select username,con_id  from cdb_users
  2  where common='NO';

USERNAME             CON_ID
-------------------- ------
PDB2_1_ADMIN              3
LU                        3
PDB2_1_ADMIN              4
SCOTT                     5
BI                        5
PM                        5
IX                        5
SH                        5
OE                        5
HR                        5
PDBADMIN                  6
HR                        6
OE                        6
SH                        6
IX                        6
PM                        6
BI                        6
SCOTT                     6

18 rows selected.

SQL>
```

2. Create a common user C##_USER.

```
SQL> create user C##_USER identified by x CONTAINER=ALL ;

User created.

SQL>
```

3. View the new common user C##_USER.

```
SQL> select distinct  username from cdb_users
  2  where username='C##_USER';

USERNAME
--------------------
C##_USER
```

```
SQL>
```

Notice that the common user exists in each container.

4. Grant `CREATE SESSION` as a common privilege.

```
SQL> GRANT CREATE SESSION TO c##_user CONTAINER=ALL;

Grant succeeded.

SQL>
```

5. Connect to root, `PDB2`, and `PDB2_2` as `c##_user` user.

```
SQL> connect  c##_user/x@pdb2
Connected.
SQL> connect  c##_user/x@pdb2_2
Connected.
SQL> connect  c##_user/x@cdb2
Connected.
SQL>
```

6. Create a local user `LOCAL_USER` in the root container.

```
SQL> connect / as sysdba
Connected.
SQL> create user local_user identified by x
  2  CONTAINER=CURRENT;
create user local_user identified by x
                                     *
ERROR at line 1:
ORA-65049: creation of local user or role is not allowed in
CDB$ROOT

SQL>
```

Notice that no local user is authorized in the root.

7. Create a common user and grant `CREATE SESSION` as a local privilege.
   a. Create the user and grant the privilege.

```
SQL> create user C##_USER2 identified by x CONTAINER=ALL;

User created.

SQL> GRANT CREATE SESSION TO c##_user2;

Grant succeeded.
```

```
SQL> CONNECT c##_user2/x@pdb2
ERROR:
```

```
ORA-01045: user C##_USER2 lacks CREATE SESSION privilege; logon
denied

SQL> CONNECT c##_user2/x@pdb2_2
ERROR:
ORA-01045: user C##_USER2 lacks CREATE SESSION privilege; logon
denied

SQL> CONNECT c##_user2/x@cdb2
Connected.
SQL>
```

Note that even though the user is a common user, the privilege is granted locally in root. That is why the common user can connect to the root container only where he is granted the `CREATE SESSION` privilege.

b. Drop the common user.

```
SQL> CONNECT / as sysdba
Connected.
SQL> DROP USER c##_user2;

User dropped.

SQL>
```

8. Create a local user, `LOCAL_USER_PDB2`, in `PDB2`.

a. View all users of PDB2.

```
SQL> connect  sys/oracle_4U@PDB2 as sysdba
Connected.

SQL> col username format a30
SQL> select USERNAME, COMMON, CON_ID  from cdb_users order by 1;

USERNAME                       COM     CON_ID
------------------------------ --- ----------
ANONYMOUS                      YES          3
APEX_040200                    YES          3
APEX_PUBLIC_USER               YES          3
APPQOSSYS                      YES          3
AUDSYS                         YES          3
C##U                           YES          3
C##_USER                       YES          3
CTXSYS                         YES          3
DBSNMP                         YES          3
DIP                            YES          3
```

```
DVF                            YES           3
DVSYS                          YES           3
FLOWS_FILES                    YES           3
GSMADMIN_INTERNAL              YES           3
GSMCATUSER                     YES           3
GSMUSER                        YES           3
LBACSYS                        YES           3
LU                             NO            3
MDDATA                         YES           3
MDSYS                          YES           3
OJVMSYS                        YES           3
OLAPSYS                        YES           3
ORACLE_OCM                     YES           3
ORDDATA                        YES           3
ORDPLUGINS                     YES           3
ORDSYS                         YES           3
OUTLN                          YES           3
PDB1_2_ADMIN                   NO            3
SI_INFORMTN_SCHEMA             YES           3
SPATIAL_CSW_ADMIN_USR                YES           3
SPATIAL_WFS_ADMIN_USR                YES           3
SYS                            YES           3
SYSBACKUP                      YES           3
SYSDG                          YES           3
SYSKM                          YES           3
SYSTEM                         YES           3
WMSYS                          YES           3
XDB                            YES           3
XS$NULL                        YES           3

39 rows selected.

SQL>
```

Notice that you view all common and local users of the current PDB.

```
SQL> select USERNAME, COMMON from dba_users;
```

Notice that you view the same list.

b. Attempt to create a common user, `C##_USER_PDB2`, in `PDB2`.

```
SQL> create user c##_user_pdb2 identified by x CONTAINER=ALL;
create user c##_user_pdb2 identified by x CONTAINER=ALL
*
ERROR at line 1:
ORA-65050: Common DDLs only allowed in CDB$ROOT
```

```
SQL>
```

Notice that no common user can be created except from the root.

c. Create the local user LOCAL_USER_PDB2 in PDB2.

```
SQL> create user local_user_pdb2 identified by x
  2  CONTAINER=CURRENT;

User created.

SQL> select USERNAME,COMMON,CON_ID  from cdb_users order by 1;

USERNAME                       COM     CON_ID
------------------------------ --- ----------
ANONYMOUS                      YES          3
APEX_040200                    YES          3
APEX_PUBLIC_USER               YES          3
APPQOSSYS                      YES          3
AUDSYS                         YES          3
C##U                           YES          3
C##_USER                       YES          3
CTXSYS                         YES          3
DBSNMP                         YES          3
DIP                            YES          3
DVF                            YES          3
DVSYS                          YES          3
FLOWS_FILES                    YES          3
GSMADMIN_INTERNAL              YES          3
GSMCATUSER                     YES          3
GSMUSER                        YES          3
LBACSYS                        YES          3
LOCAL_USER_PDB2                NO           3
LU                             NO           3
MDDATA                         YES          3
MDSYS                          YES          3
OJVMSYS                        YES          3
OLAPSYS                        YES          3
ORACLE_OCM                     YES          3
ORDDATA                        YES          3
ORDPLUGINS                     YES          3
ORDSYS                         YES          3
OUTLN                          YES          3
PDB1_2_ADMIN                   NO           3
```

```
SI_INFORMTN_SCHEMA              YES          3
SPATIAL_CSW_ADMIN_USR                YES          3
SPATIAL_WFS_ADMIN_USR                YES          3
SYS                             YES          3
SYSBACKUP                       YES          3
SYSDG                           YES          3
SYSKM                           YES          3
SYSTEM                          YES          3
WMSYS                           YES          3
XDB                             YES          3
XS$NULL                         YES          3

40 rows selected.

SQL>
SQL> grant create session to local_user_pdb2;

Grant succeeded.

SQL>
```

d. Connect to PDB2 as LOCAL_USER_PDB2.

```
SQL> connect local_user_pdb2/x@PDB2
Connected.
SQL>
```

e. Connect to PDB2_2 as LOCAL_USER_PDB2.

```
SQL> connect local_user_pdb2/x@PDB2_2
ERROR:
ORA-01017: invalid username/password; logon denied

Warning: You are no longer connected to ORACLE.
SQL>
```

Notice that it fails because LOCAL_USER_PDB2 does not exist in PDB2_2.

```
SQL> connect  local_user_pdb2/x@cdb2
ERROR:
ORA-01017: invalid username/password; logon denied

SQL>
```

Notice that it fails because LOCAL_USER_PDB2 does not exist in root.

f. Overview of common and local users from a PDB:

```
SQL> connect sys/oracle_4U@PDB2_2 as sysdba
Connected.
```

```
SQL> col username format a30
SQL> select USERNAME, COMMON, CON_ID from cdb_users order by 1;

USERNAME                       COM CON_ID
------------------------------ --- ------
ANONYMOUS                      YES      4
APPQOSSYS                      YES      4
…
C##U                           YES      4
C##_USER                       YES      4
…
PDB2_1_ADMIN                   NO       4
…
SYSTEM                         YES      4
…
XS$NULL                        YES      4

38 rows selected.

SQL>
```

Notice that you view all common and local users of the current PDB.

```
SQL> select USERNAME, COMMON from dba_users order by username;

USERNAME                       COM
------------------------------ ---
ANONYMOUS                      YES
APPQOSSYS                      YES
…
C##_USER                       YES
…
SYSTEM                         YES
…
XS$NULL                        YES

38 rows selected.

SQL>
```

Notice that you view the same list.

Practice 6-2: Managing Local and Common Roles

Overview

In this practice, you will manage roles created as common or local, and granted as common and or local in CDB and PDBs.

Assumptions

`C##_USER` and `LOCAL_USER_PDB2` are successfully created from the previous practice 6-1 in `cdb2` and `PDB2` respectively.

Tasks

1. Manage creation of roles in CDB and PDBs.
   a. List all predefined roles in CDB.

```
SQL> connect / as sysdba
Connected.
SQL> col role format a30
SQL> select ROLE, COMMON, CON_ID from cdb_roles order by role;

ROLE                           COM CON_ID
------------------------------ --- ------
ADM_PARALLEL_EXECUTE_TASK      YES      2
ADM_PARALLEL_EXECUTE_TASK      YES      1
ADM_PARALLEL_EXECUTE_TASK      YES      3
ADM_PARALLEL_EXECUTE_TASK      YES      4
ADM_PARALLEL_EXECUTE_TASK      YES      5
ADM_PARALLEL_EXECUTE_TASK      YES      6

…
DBA                            YES      3
DBA                            YES      1
DBA                            YES      4
DBA                            YES      2
DBA                            YES      5
DBA                            YES      6

…
PDB_DBA                        YES      3
PDB_DBA                        YES      4
PDB_DBA                        YES      6

…

504 rows selected.
```

```
SQL>
```

You can view all common and local roles of the root and PDBs.

b. List all predefined roles in root.

```
SQL> select ROLE, COMMON from dba_roles order by role;

ROLE                           COM
------------------------------ ---
ADM_PARALLEL_EXECUTE_TASK      YES
APEX_ADMINISTRATOR_ROLE        YES
AQ_ADMINISTRATOR_ROLE          YES
AQ_USER_ROLE                   YES
…
XS_CACHE_ADMIN                 YES
XS_NSATTR_ADMIN                YES
XS_RESOURCE                    YES
XS_SESSION_ADMIN               YES

84 rows selected.

SQL>
```

Notice that all roles of the root are common; there cannot be any local roles in the root.

c. Create a common C##_ROLE in root.

```
SQL> create role c##_role container=ALL;

Role created.

SQL>
```

d. Create a local LOCAL_ROLE in root.

```
SQL> create role local_role container=CURRENT;
create role local_role container=CURRENT
*
ERROR at line 1:
ORA-65049: creation of local user or role is not allowed in
CDB$ROOT

SQL>
```

You get an error message because no local role is authorized in the root.

e. List all predefined roles in PDB PDB2.

```
SQL> connect  system/oracle_4U@PDB2
Connected.
```

```
SQL> col role format a30
SQL> select ROLE, COMMON, CON_ID from cdb_roles;

ROLE                           COM CON_ID
------------------------------ --- ------
CONNECT                        YES      3
RESOURCE                       YES      3
DBA                            YES      3
AUDIT_ADMIN                    YES      3
…
C##_ROLE                      YES      3
DV_REALM_RESOURCE              YES      3
DV_REALM_OWNER                 YES      3
PDB_DBA                         YES     3
…
85 rows selected.

SQL>
```

You can view all common and local roles of the PDB only.

```
SQL> select ROLE,COMMON from dba_roles order by role;

ROLE                           COM
------------------------------ ---
CONNECT                        YES
RESOURCE                       YES
DBA                            YES
AUDIT_ADMIN                    YES
…
PDB_DBA                       YES
…
XDB_WEBSERVICES                YES
XDB_WEBSERVICES_OVER_HTTP       YES
XDB_WEBSERVICES_WITH_PUBLIC    YES
XS_CACHE_ADMIN                 YES
XS_NSATTR_ADMIN                YES
XS_RESOURCE                    YES
XS_SESSION_ADMIN               YES

85 rows selected.

SQL>
```

You view the same list.

f. Create a common role in PDB2.

```
SQL> create role c##_role_PDB2 container=ALL;
create role c##_role_PDB2 container=ALL
*
ERROR at line 1:
ORA-65050: Common DDLs only allowed in CDB$ROOT

SQL>
```

You get an error message because no common role can be created from a PDB.

g. Create a local role in PDB2.

```
SQL> create role local_role_PDB2 container=CURRENT;
Role created.

SQL> select ROLE, COMMON from dba_roles order by role;

ROLE                           COM
------------------------------ ---
ADM_PARALLEL_EXECUTE_TASK      YES
APEX_ADMINISTRATOR_ROLE        YES
AQ_ADMINISTRATOR_ROLE          YES
…
LOCAL_ROLE_PDB2                NO
…
XS_NSATTR_ADMIN                YES
XS_RESOURCE                    YES
XS_SESSION_ADMIN               YES

86 rows selected.

SQL>
```

2. Grant common or local roles as common or local.

a. Grant a common role to a common user from the root.

```
SQL> connect / as sysdba
Connected.

SQL> grant c##_role to c##_user;

Grant succeeded.

SQL> col grantee format A16
SQL> col GRANTED_ROLE format A16
SQL> select GRANTEE, GRANTED_ROLE, COMMON, CON_ID
```

```
  2  from cdb_role_privs where grantee='C##_USER';

GRANTEE          GRANTED_ROLE     COM CON_ID
---------------- ---------------- --- ------
C##_USER         C##_ROLE         NO       1

SQL>
```

Note that the common role is granted locally to the common user. The granted role is applicable only in the root.

```
SQL> connect c##_user/x
Connected.

SQL> select * from session_roles;

ROLE
------------------------------
C##_ROLE

SQL>
```

```
SQL> connect c##_user/x@PDB2
Connected.

SQL> select * from session_roles;

no rows selected

SQL>
```

b. Now grant the common role to a common user from the root as common, to be applicable in all containers.

```
SQL> connect / as sysdba
Connected.
SQL> grant c##_role to c##_user container=all;

Grant succeeded.

SQL> select GRANTEE, GRANTED_ROLE, COMMON, CON_ID
  2  from cdb_role_privs where grantee='C##_USER';

GRANTEE          GRANTED_ROLE     COM CON_ID
---------------- ---------------- --- ------
C##_USER         C##_ROLE         NO       1
```

```
C##_USER          C##_ROLE          YES      1
C##_USER          C##_ROLE          YES      3
C##_USER          C##_ROLE          YES      4
C##_USER          C##_ROLE          YES      5
C##_USER          C##_ROLE          YES      6

SQL>
```

```
SQL> connect c##_user/x
Connected.

SQL> select * from session_roles;

ROLE
------------------------------
C##_ROLE

SQL>
```

```
SQL> connect c##_user/x@PDB2
Connected.
SQL> select * from session_roles;

ROLE
------------------------------
C##_ROLE

SQL>
```

c. Revoke the common role from the common user so that the role cannot be used in any container.

```
SQL> connect / as sysdba
Connected.

SQL> revoke c##_role from c##_user container=all;
Revoke succeeded.

SQL> connect c##_user/x
Connected.

SQL> select * from session_roles;
ROLE
------------------------------
```

```
C##_ROLE

SQL>
```

```
SQL> connect c##_user/x@PDB2
Connected.
SQL> select * from session_roles;

no rows selected

SQL>
```

d. Grant a common role to a local user from the root.

```
SQL> connect / as sysdba
Connected.

SQL> grant c##_role to local_user_pdb2;
grant c##_role to local_user_pdb2
                  *
ERROR at line 1:
ORA-01917: user or role 'LOCAL_USER_PDB2' does not exist

SQL>
```

Note that the user is unknown in root. It is a local user in `PDB2`.

e. Grant a common role to a local user from `PDB2`.

```
SQL> connect system/oracle_4U@PDB2
Connected.
SQL> grant c##_role to local_user_PDB2;

Grant succeeded.

SQL> select GRANTEE, GRANTED_ROLE, COMMON, CON_ID
  2  from cdb_role_privs where grantee='LOCAL_USER_PDB2';

GRANTEE          GRANTED_ROLE     COM CON_ID
---------------- ---------------- --- ------
LOCAL_USER_PDB2  C##_ROLE         NO       3

SQL>
```

Note that the user is granted a common role locally (common column = NO) applicable only in the PDB `PDB2`.

f. Test the connection as the local user.

```
SQL> connect  local_user_pdb2/x@PDB2
Connected.

SQL> select * from session_roles;
ROLE
------------------------------
C##_ROLE

SQL>
```

g. Grant a common role to a local user from PDB2 applicable in all containers.

```
SQL> connect system/oracle_4U@PDB2
Connected.
SQL> grant c##_role to local_user_pdb2 container=all;
grant c##_role to local_user_pdb2 container=all
*
ERROR at line 1:
ORA-65030: one may not grant a Common Privilege to a Local User
or Role

SQL>
```

Notice that a common role cannot be granted globally from a PDB.

h. Grant a local role to a local user from PDB2.

```
SQL> grant local_role_pdb2 to local_user_pdb2;

Grant succeeded.

SQL> select GRANTEE, GRANTED_ROLE, COMMON, CON_ID
  2  from cdb_role_privs where grantee='LOCAL_USER_PDB2';

GRANTEE          GRANTED_ROLE     COM CON_ID
---------------- ---------------- --- ------
LOCAL_USER_PDB2  C##_ROLE         NO       3
LOCAL_USER_PDB2  LOCAL_ROLE_PDB2  NO       3

SQL>
```

i. Test the connection as the local user.

```
SQL> connect  local_user_pdb2/x@PDB2
Connected.
SQL> select * from session_roles;

ROLE
------------------------------
```

```
C##_ROLE
LOCAL_ROLE_PDB2

SQL>
```

Practice 6-3: Managing Local and Common Privileges

Overview

In this practice, you will manage privileges granted as common or local in CDB and PDBs.

Assumptions

C##_USER and LOCAL_USER_PDB2 are successfully created from previous practice 5-2 in PDB2 of cdb2.

Tasks

1. Check whether privileges are created as common or local.

```
SQL> connect / as sysdba
Connected.
SQL> desc sys.system_privilege_map
Name                                      Null?    Type
 ----------------------------------------- -------- ------------
 PRIVILEGE                                 NOT NULL NUMBER
 NAME                                      NOT NULL VARCHAR2(40)
 PROPERTY                                  NOT NULL NUMBER

SQL> desc sys.table_privilege_map
 Name                                      Null?    Type
 ----------------------------------------- -------- ------------
 PRIVILEGE                                 NOT NULL NUMBER
 NAME                                      NOT NULL VARCHAR2(40)

SQL>
```

   Notice that there is no COMMON column. Privileges are created neither as common nor as local, but they can be granted as common or local.

2. Check how the CREATE SESSION system privilege was granted to C##_USER and LOCAL_USER_PDB2 users.

```
SQL> connect system/oracle_4U
Connected.
SQL> col grantee format a18
SQL> col privilege format a14
SQL> select GRANTEE, PRIVILEGE, COMMON, CON_ID
  2  from cdb_sys_privs
  3  where grantee in ('C##_USER', 'LOCAL_USER_PDB2');

GRANTEE            PRIVILEGE      COM CON_ID
------------------ -------------- --- ------
C##_USER           CREATE SESSION YES      1
C##_USER           CREATE SESSION YES      3
```

```
LOCAL_USER_PDB2    CREATE SESSION NO       3
C##_USER           CREATE SESSION YES      4
C##_USER           CREATE SESSION YES      5
C##_USER           CREATE SESSION YES      6

6 rows selected.

SQL>
```

```
SQL> connect system/oracle_4U@PDB2
Connected.
SQL> select GRANTEE, PRIVILEGE, COMMON
  2  from dba_sys_privs
  3  where grantee in ('C##_USER', 'LOCAL_USER_PDB2');

GRANTEE            PRIVILEGE      COM
------------------ -------------- ---
LOCAL_USER_PDB2    CREATE SESSION NO
C##_USER           CREATE SESSION YES

SQL>
```

3. Grant the system privileges `CREATE TABLE` and `UNLIMITED TABLESPACE` to common user `C##_USER` to be applicable in any container. This will be a common privilege.

```
SQL> connect system/oracle_4U
Connected.
SQL> grant CREATE TABLE, UNLIMITED TABLESPACE to C##_USER
  2  CONTAINER=ALL;

Grant succeeded.

SQL> col grantee format a12
SQL> col privilege format a30

SQL> select GRANTEE, PRIVILEGE, COMMON, CON_ID
  2  from cdb_sys_privs
  3  where grantee = 'C##_USER';

GRANTEE      PRIVILEGE                      COM CON_ID
------------ ------------------------------ --- ------
C##_USER  CREATE TABLE                   YES      1
C##_USER  CREATE SESSION                 YES      1
C##_USER  UNLIMITED TABLESPACE           YES      1
```

```
C##_USER  CREATE TABLE                    YES       3
C##_USER  CREATE SESSION                  YES       3
C##_USER  UNLIMITED TABLESPACE            YES       3
C##_USER  CREATE TABLE                    YES       4
C##_USER  CREATE SESSION                  YES       4
C##_USER  UNLIMITED TABLESPACE            YES       4
C##_USER  CREATE TABLE                    YES       5
C##_USER  CREATE SESSION                  YES       5
C##_USER  UNLIMITED TABLESPACE            YES       5
C##_USER  CREATE TABLE                    YES       6
C##_USER  CREATE SESSION                  YES       6
C##_USER  UNLIMITED TABLESPACE            YES       6

15 rows selected.

SQL>
```

4. Grant the system privilege `CREATE SEQUENCE` to common user `C##_USER` to be applicable in root only. This will be a local privilege.

```
SQL> col grantee format a12
SQL> grant CREATE SEQUENCE to C##_USER CONTAINER=CURRENT;
Grant succeeded.

SQL> select GRANTEE, PRIVILEGE, COMMON, CON_ID
  2  from cdb_sys_privs
  3  where grantee = 'C##_USER';

GRANTEE      PRIVILEGE                      COM CON_ID
------------ ------------------------------ --- ------
C##_USER  CREATE SEQUENCE                 NO        1
C##_USER  CREATE TABLE                    YES       1
C##_USER  CREATE SESSION                  YES       1
C##_USER  UNLIMITED TABLESPACE            YES       1
C##_USER  CREATE TABLE                    YES       3
C##_USER  CREATE SESSION                  YES       3
C##_USER  UNLIMITED TABLESPACE            YES       3
C##_USER  CREATE TABLE                    YES       4
C##_USER  CREATE SESSION                  YES       4
C##_USER  UNLIMITED TABLESPACE            YES       4
C##_USER  CREATE TABLE                    YES       5
C##_USER  CREATE SESSION                  YES       5
C##_USER  UNLIMITED TABLESPACE            YES       5
C##_USER  CREATE TABLE                    YES       6
```

```
C##_USER  CREATE SESSION                  YES      6
C##_USER  UNLIMITED TABLESPACE            YES      6

16 rows selected.

SQL>
```

5. Grant the system privilege `CREATE SYNONYM` to common user `C##_USER` to be applicable in `PDB2` only. This will be a local privilege.

```
SQL> connect system/oracle_4U@PDB2
Connected.

SQL> col grantee format a18
SQL> grant CREATE SYNONYM to C##_USER CONTAINER=CURRENT;
Grant succeeded.

SQL> select GRANTEE, PRIVILEGE, COMMON, CON_ID
  2   from cdb_sys_privs
  3   where grantee = 'C##_USER';

GRANTEE            PRIVILEGE                      COM CON_ID
------------------ ------------------------------ --- ------
C##_USER        CREATE SYNONYM                  NO       3
C##_USER        CREATE TABLE                    YES      3
C##_USER        CREATE SESSION                  YES      3
C##_USER        UNLIMITED TABLESPACE            YES      3

SQL>
```

6. Grant the system privilege `CREATE VIEW` to common user `C##_USER` to be applicable in root only but connected in `PDB2`.

```
SQL> col grantee format a18
SQL> grant CREATE VIEW to C##_USER CONTAINER=ALL;
grant CREATE VIEW to C##_USER CONTAINER=ALL
*
ERROR at line 1:
ORA-65050: Common DDLs only allowed in CDB$ROOT

SQL>
```

Note that you cannot grant a common privilege from a PDB .

7. Grant the system privilege `CREATE ANY TABLE` to local user `LOCAL_USER_PDB2` to be applicable in any container.

```
SQL> connect system/oracle_4U
Connected.
```

```
SQL> col grantee format a18
SQL> grant CREATE ANY TABLE to LOCAL_USER_PDB2 CONTAINER=ALL;
grant CREATE ANY TABLE to LOCAL_USER_PDB2 CONTAINER=ALL
                          *
ERROR at line 1:
ORA-01917: user or role 'LOCAL_USER_PDB2' does not exist

SQL>
```

Notice that the user is unknown in root. It is a local user in PDB2.

8. Grant the system privilege CREATE ANY SEQUENCE to local user LOCAL_USER_PDB2 to be applicable in root only.

```
SQL> grant CREATE ANY SEQUENCE to LOCAL_USER_PDB2
  2  CONTAINER=CURRENT;
grant CREATE ANY SEQUENCE to LOCAL_USER_PDB2
                             *
ERROR at line 1:
ORA-01917: user or role 'LOCAL_USER_PDB2' does not exist

SQL>
```

Notice that the user is unknown in root. It is a local user in PDB2.

9. Grant the system privilege UNLIMITED TABLESPACE to local user LOCAL_USER_PDB2 to be applicable in PDB2 only. This will be a local privilege.

```
SQL> connect system/oracle_4U@PDB2
Connected.

SQL> col grantee format a18
SQL> grant UNLIMITED TABLESPACE to LOCAL_USER_PDB2;
Grant succeeded.

SQL> select GRANTEE, PRIVILEGE, COMMON, CON_ID
  2  from cdb_sys_privs
  3  where grantee = 'LOCAL_USER_PDB2';

GRANTEE            PRIVILEGE                      COM CON_ID
------------------ ------------------------------ --- ------
LOCAL_USER_PDB2    CREATE SESSION                 NO       3
LOCAL_USER_PDB2    UNLIMITED TABLESPACE           NO       3

SQL>
```

10. Grant the system privilege DROP ANY VIEW to local user LOCAL_USER_PDB2 to be applicable in root only but connected in PDB2.

```
SQL> grant DROP ANY VIEW to LOCAL_USER_PDB2 CONTAINER=ALL;

grant DROP ANY VIEW to LOCAL_USER_PDB2 CONTAINER=ALL
*
ERROR at line 1:
ORA-65030: one may not grant a Common Privilege to a Local User
or Role

SQL>
```

Notice that you cannot grant a local privilege that will be applicable in another container.

Practice 6-4: Enabling Common Users to View Information About PDB Objects

Overview

In this practice, you will manage the CONTAINER_DATA attributes of common users to enable common users to view information about PDB objects in specific PDBs.

Tasks

1. Find information about the default (user-level) and object-specific CONTAINER_DATA attributes that are explicitly set to a value other than DEFAULT in the DBA_CONTAINER_DATA data dictionary view.

```
SQL> CONNECT / as sysdba

Connected to:
Oracle Database 12c Enterprise Edition Release 12.1.0.1.0 -
64bit Production
With the Partitioning, OLAP, Advanced Analytics and Real
Application Testing options

SQL> COLUMN USERNAME FORMAT A10
SQL> COLUMN DEFAULT_ATTR FORMAT A7
SQL> COLUMN OWNER FORMAT A8
SQL> COLUMN OBJECT_NAME FORMAT A10
SQL> COLUMN ALL_CONTAINERS FORMAT A3
SQL> COLUMN CONTAINER_NAME FORMAT A10
SQL> COLUMN CON_ID FORMAT 999
SQL> set pages 100
SQL> set line 200
SQL> SELECT USERNAME, DEFAULT_ATTR, OWNER, OBJECT_NAME,
            ALL_CONTAINERS, CONTAINER_NAME, CON_ID
     FROM CDB_CONTAINER_DATA
     WHERE username NOT IN
           ('GSMADMIN_INTERNAL', 'APPQOSSYS', 'DBSNMP')
     ORDER BY OBJECT_NAME;

USERNAME   DEFAULT OWNER    OBJECT_NAM ALL CONTAINER_ CON_ID
---------- ------- -------- ---------- --- ---------- ------
SYSTEM     Y                          Y                   1
SYSBACKUP  Y                          Y                   1
SYS        Y                          Y                   1

SQL>
```

2. Create the common user c##jfv and grant c##jfv the system privileges CREATE SESSION and SET CONTAINER.

```
SQL> CREATE USER c##jfv IDENTIFIED BY oracle_4U CONTAINER=ALL;

User created.

SQL> GRANT CREATE SESSION, SET CONTAINER TO c##jfv
     CONTAINER=ALL;

Grant succeeded.

SQL>
```

3. Then grant c##jfv the object privileges SELECT on V_$SESSION view.

```
SQL> GRANT SELECT ON sys.v_$session TO c##jfv CONTAINER=ALL;

Grant succeeded.

SQL>
```

4. Create a second session connected to pdb1_1 as user SYS, and stay connected.

```
$ . oraenv
ORACLE_SID = [oracle] ? cdb2
The Oracle base for
ORACLE_HOME=/u01/app/oracle/product/12.1.0/dbhome_1 is
/home/oracle
$ sqlplus sys/oracle_4U@pdb1_1 as sysdba

Connected to:
Oracle Database 12c Enterprise Edition Release 12.1.0.1.0 -
64bit Production
With the Partitioning, OLAP, Advanced Analytics and Real
Application Testing options

SQL> SHOW CON_NAME

CON_NAME
------------------------------
PDB1_1
SQL> ALTER SESSION SET CONTAINER=pdb1_1;

Session altered.

SQL> SHOW CON_NAME
```

```
CON_NAME
------------------------------
PDB1_1
SQL>
```

5. In the first session, you should see one row for `pdb1_1`.

```
SQL> SELECT username, con_id FROM v_$session
     WHERE username IS NOT NULL AND username <> 'DBSNMP';

USERNAME   CON_ID
---------- ------
SYS             1
SYS             6

SQL>
```

6. Still in the first session, you connect as the common user `c##jfv`. The common user does not see any information in `V_$SESSION` related to `pdb1_1`.

```
SQL> CONNECT c##jfv/oracle_4U
Connected.

SQL> SELECT username, con_id FROM sys.v_$session
     WHERE username IS NOT NULL AND username <> 'DBSNMP';

USERNAME                  CON_ID
------------------------- ------
C##JFV                         1

SQL>
```

7. Enable the common user `c##jfv` to see information in `V_$SESSION` related to `pdb1_1`.

```
SQL> CONNECT / AS SYSDBA
Connected.
SQL> ALTER USER c##jfv
         SET CONTAINER_DATA = (CDB$ROOT, PDB1_1, PDB2_2)
         FOR V_$SESSION
         CONTAINER=CURRENT;

User altered.

SQL>
```

8. Connect as the common user `c##jfv` to view information in `V$SESSION` related to `pdb1_1`.

```
SQL> CONNECT c##jfv/oracle_4U
Connected.

SQL> SELECT username, con_id FROM sys.v_$session
     WHERE username IS NOT NULL AND username <> 'DBSNMP';

USERNAME             CON_ID
-------------------- ------
C##JFV                    1
SYS                       6

SQL>
```

9. View the CONTAINER_DATA attribute set for the common user C##JFV on object V$SESSION in pdb1_1, pdb2_2.

```
SQL> CONNECT / AS SYSDBA
Connected.
SQL> COLUMN USERNAME FORMAT A25
SQL> COLUMN DEFAULT_ATTR FORMAT A7
SQL> COLUMN OWNER FORMAT A15
SQL> COLUMN OBJECT_NAME FORMAT A15
SQL> COLUMN ALL_CONTAINERS FORMAT A3
SQL> COLUMN CONTAINER_NAME FORMAT A10
SQL> COLUMN CON_ID FORMAT 999
SQL> set pages 100
SQL> set line 200
SQL> SELECT USERNAME, DEFAULT_ATTR, OWNER, OBJECT_NAME,
            ALL_CONTAINERS, CONTAINER_NAME, CON_ID
     FROM CDB_CONTAINER_DATA
     WHERE username NOT IN
            ('GSMADMIN_INTERNAL', 'APPQOSSYS', 'DBSNMP')
     ORDER BY OBJECT_NAME;
  2    3    4    5    6
USERNAME   DEFAULT OWNER    OBJECT_NAM ALL CONTAINER_ CON_ID
---------- ------- -------- ---------- --- ---------- ------
C##JFV     N       SYS      V_$SESSION N   CDB$ROOT        1
C##JFV     N       SYS      V_$SESSION N   PDB2_2                1
C##JFV     N       SYS      V_$SESSION N   PDB1_1                1
SYSTEM     Y                           Y                  1
SYSBACKUP  Y                           Y                  1
SYS        Y                           Y                  1

6 rows selected.
```

```
SQL> EXIT
$
```

Exit from all SQL*Plus sessions.

Practices for Lesson 7: Backup, Recovery, Flashback CDB and PDBs

Chapter 7

Practices for Lesson 7: Overview

Practices Overview

In the followingpractices, you will perform backup and recovery operations on the CDB and PDBs.

- RMAN `cdb2` backup
- RMAN whole and partial `pdb2` backup
- Recovery from SYSTEM `pdb2` data file loss
- Recovery from non-essential `pdb2` data file loss
- SQL PDB hot backup
- SQL control file backup
- Recovery from all control files loss
- Recovery from redo log member loss
- Recovery from SYSTEM root data file loss
- Recovery from a non-essential root data file loss
- PDB point-in-time recovery
- PDB tablespaces point-in-time recovery
- CDB flashback from DROP common user
- Plugging of an unplugged PDB by using RMAN backup

Assumptions

`cdb2` is successfully created from Practice 3-1.

`pdb2_1` is successfully created from Practice 3-3.

`pdb2_1` is successfully renamed to `pdb2` from Practice 4-4.

If the trigger could not be created successfully, execute the following catchup script:

```
$ cd /home/oracle/solutions/catchup_04_03
$ ./cr_trig.sh
```

If permanent and temporary tablespaces could not be created successfully, execute the following catchup script:

```
$ cd /home/oracle/solutions/catchup_05_02
$ ./cr_TABLESPACES.sh
```

Practice 7-1: Cold CDB Backup

Overview

In this practice, you will perform a CDB cold backup that you can use in case you lose all further backups or you cannot recover from a difficult situation.

But before performing this task, make sure your database is in `ARCHIVELOG` mode.

Tasks

1. To reduce the number of pluggable databases to backup for the whole CDB, you will drop `pdb1_1` and `pdb_orcl2`.

```
$ . oraenv
ORACLE_SID = [cdb2] ? cdb2
The Oracle base remains unchanged with value /u01/app/oracle
$ sqlplus / AS SYSDBA

Connected to:
Oracle Database 12c Enterprise Edition Release 12.1.0.1.0 -
64bit Production
With the Partitioning, OLAP, Advanced Analytics and Real
Application Testing options

SQL> ALTER PLUGGABLE DATABASE pdb1_1 CLOSE IMMEDIATE;

Pluggable database altered.

SQL> DROP PLUGGABLE DATABASE pdb1_1 INCLUDING DATAFILES;

Pluggable database dropped.

SQL> ALTER PLUGGABLE DATABASE pdb_orcl2 CLOSE IMMEDIATE;

Pluggable database altered.

SQL> DROP PLUGGABLE DATABASE pdb_orcl2 INCLUDING DATAFILES;

Pluggable database dropped.

SQL> EXIT
$
```

2. Create the backup directory.

```
$ rm -Rf /home/oracle/Safe_Database_Files/cdb2
$ mkdir  /home/oracle/Safe_Database_Files
$ mkdir  /home/oracle/Safe_Database_Files/cdb2
```

```
$
```

3. Shut down the `cdb2` database before backing up all the files.

```
$ sqlplus / AS SYSDBA

Connected to:
Oracle Database 12c Enterprise Edition Release 12.1.0.1.0 -
64bit Production
With the Partitioning, OLAP, Advanced Analytics and Real
Application Testing options

SQL> select log_mode from v$database;

LOG_MODE
------------
NOARCHIVELOG

SQL> SHUTDOWN IMMEDIATE
Database closed.
Database dismounted.
ORACLE instance shut down.

SQL> STARTUP MOUNT
ORACLE instance started.

Total System Global Area 1068937216 bytes
Fixed Size                  2248280 bytes
Variable Size             343933352 bytes
Database Buffers          717225984 bytes
Redo Buffers                5529600 bytes
Database mounted.
SQL> ALTER DATABASE ARCHIVELOG;
Database altered.

SQL> ALTER DATABASE OPEN;
Database altered.

SQL> SELECT name FROM v$datafile;

NAME
--------------------------------------------------------------------
/u01/app/oracle/oradata/cdb2/system01.dbf
/u01/app/oracle/oradata/cdb2/sysaux01.dbf
```

```
/u01/app/oracle/oradata/cdb2/undotbs01.dbf
/u01/app/oracle/oradata/cdb2/pdbseed/system01.dbf
/u01/app/oracle/oradata/cdb2/users01.dbf
/u01/app/oracle/oradata/cdb2/pdbseed/sysaux01.dbf
/u01/app/oracle/oradata/cdb2/pdb2_1/system01.dbf
/u01/app/oracle/oradata/cdb2/pdb2_1/sysaux01.dbf
/u01/app/oracle/oradata/cdb2/pdb2_2/CDB2/CE88299AD6087639E0436B2
3B98B8558/datafile/o1_mf_system_8b9ocl9w_.dbf
/u01/app/oracle/oradata/cdb2/pdb2_2/CDB2/CE88299AD6087639E0436B2
3B98B8558/datafile/o1_mf_sysaux_8b9ocl7f_.dbf
/u01/app/oracle/oradata/cdb2/cdata_01.dbf
/u01/app/oracle/oradata/cdb2/pdb2_1/ldata_01.dbf

12 rows selected.

SQL> SHUTDOWN IMMEDIATE
Database closed.
Database dismounted.
ORACLE instance shut down.

SQL> EXIT

Disconnected from Oracle Database 12c Enterprise Edition Release
12.1.0.1.0 - 64bit Production
With the Partitioning, OLAP, Advanced Analytics and Real
Application Testing options
$
```

4. Copy the files to the backup directory. The message is an informative message only.

```
$ tar -czf /home/oracle/Safe_Database_Files/cdb2/db.tar.gz
/u01/app/oracle/oradata/cdb2
tar: Removing leading `/' from member names
$
```

5. Start up the `cdb2` database before performing backups with RMAN.

```
$ sqlplus / AS SYSDBA

Connected to an idle instance.

SQL> startup
ORACLE instance started.

Total System Global Area 1068937216 bytes
Fixed Size                  2248280 bytes
Variable Size             343933352 bytes
```

```
Database Buffers          717225984 bytes
Redo Buffers                5529600 bytes
Database mounted.
Database opened.
SQL> EXIT
$
```

Practice 7-2: RMAN Whole CDB Backup

Overview

In this practice, you will perform a whole CDB backup of `cdb2`.

Assumptions

The `PDB2` has been successfully created in `cdb2` after Practices 3-3 and 4-4.

Tasks

1. Run RMAN to connect to `cdb2` with a user with `SYSDBA` or `SYSBACKUP` privilege.

```
$ export NLS_DATE_FORMAT='DD-MM-YYYY HH:MI:SS'
$ rman target /

connected to target database: CDB2 (DBID=544732113)
RMAN>
```

2. As usual, back up all data files of the database (root and all PDBs), control files and SPFILE, and archive log files, after setting the **db_recovery_file_dest_size** to 18 GB.

```
RMAN> CONFIGURE DEFAULT DEVICE TYPE TO disk;
using target database control file instead of recovery catalog
new RMAN configuration parameters:
CONFIGURE DEFAULT DEVICE TYPE TO DISK;
new RMAN configuration parameters are successfully stored

RMAN> CONFIGURE CONTROLFILE AUTOBACKUP ON;
new RMAN configuration parameters:
CONFIGURE CONTROLFILE AUTOBACKUP ON;
new RMAN configuration parameters are successfully stored

RMAN> ALTER SYSTEM SET db_recovery_file_dest_size=18G
SCOPE=both;

Statement processed

RMAN> BACKUP DATABASE PLUS ARCHIVELOG;

Starting backup at 16-11-2012 12:02:55
current log archived
allocated channel: ORA_DISK_1
channel ORA_DISK_1: SID=45 device type=DISK
channel ORA_DISK_1: starting archived log backup set
channel ORA_DISK_1: specifying archived log(s) in backup set
input archived log thread=1 sequence=19 RECID=1 STAMP=799459389
```

```
channel ORA_DISK_1: starting piece 1 at 16-11-2012 12:03:13
channel ORA_DISK_1: finished piece 1 at 16-11-2012 12:03:21
piece
handle=/u01/app/oracle/fast_recovery_area/CDB2/backupset/2012_11
_16/o1_mf_annnn_TAG20121116T000312_8bc0p236_.bkp
tag=TAG20121116T000312 comment=NONE
channel ORA_DISK_1: backup set complete, elapsed time: 00:00:08
Finished backup at 16-11-2012 12:03:21

Starting backup at 16-11-2012 12:03:22
using channel ORA_DISK_1
channel ORA_DISK_1: starting full datafile backup set
channel ORA_DISK_1: specifying datafile(s) in backup set
input datafile file number=00003
name=/u01/app/oracle/oradata/cdb2/sysaux01.dbf
input datafile file number=00001
name=/u01/app/oracle/oradata/cdb2/system01.dbf
input datafile file number=00004
name=/u01/app/oracle/oradata/cdb2/undotbs01.dbf
input datafile file number=00022
name=/u01/app/oracle/oradata/cdb2/cdata_01.dbf
input datafile file number=00006
name=/u01/app/oracle/oradata/cdb2/users01.dbf
channel ORA_DISK_1: starting piece 1 at 16-11-2012 12:03:24
channel ORA_DISK_1: finished piece 1 at 16-11-2012 12:06:11
piece
handle=/u01/app/oracle/fast_recovery_area/CDB2/backupset/2012_11
_16/o1_mf_nnndf_TAG20121116T000323_8bc0pn36_.bkp
tag=TAG20121116T000323 comment=NONE
channel ORA_DISK_1: backup set complete, elapsed time: 00:02:47
channel ORA_DISK_1: starting full datafile backup set
channel ORA_DISK_1: specifying datafile(s) in backup set
input datafile file number=00009
name=/u01/app/oracle/oradata/cdb2/pdb2_1/sysaux01.dbf
input datafile file number=00008
name=/u01/app/oracle/oradata/cdb2/pdb2_1/system01.dbf
input datafile file number=00023
name=/u01/app/oracle/oradata/cdb2/pdb2_1/ldata_01.dbf
channel ORA_DISK_1: starting piece 1 at 16-11-2012 12:06:13
channel ORA_DISK_1: finished piece 1 at 16-11-2012 12:08:20
piece
handle=/u01/app/oracle/fast_recovery_area/CDB2/backupset/2012_11
_16/o1_mf_nnndf_TAG20121116T000323_8bc0vq2q_.bkp
tag=TAG20121116T000323 comment=NONE
channel ORA_DISK_1: backup set complete, elapsed time: 00:02:07
channel ORA_DISK_1: starting full datafile backup set
```

```
channel ORA_DISK_1: specifying datafile(s) in backup set
input datafile file number=00007
name=/u01/app/oracle/oradata/cdb2/pdbseed/sysaux01.dbf
input datafile file number=00005
name=/u01/app/oracle/oradata/cdb2/pdbseed/system01.dbf
channel ORA_DISK_1: starting piece 1 at 16-11-2012 12:08:22
channel ORA_DISK_1: finished piece 1 at 16-11-2012 12:11:20
piece
handle=/u01/app/oracle/fast_recovery_area/CDB2/backupset/2012_11
_16/o1_mf_nnndf_TAG20121116T000323_8bc0zq6c_.bkp
tag=TAG20121116T000323 comment=NONE
channel ORA_DISK_1: backup set complete, elapsed time: 00:02:58
channel ORA_DISK_1: starting full datafile backup set
channel ORA_DISK_1: specifying datafile(s) in backup set
input datafile file number=00011
name=/u01/app/oracle/oradata/cdb2/pdb2_2/CDB2/CE88299AD6087639E0
436B23B98B8558/datafile/o1_mf_sysaux_8b9ocl7f_.dbf
input datafile file number=00010
name=/u01/app/oracle/oradata/cdb2/pdb2_2/CDB2/CE88299AD6087639E0
436B23B98B8558/datafile/o1_mf_system_8b9ocl9w_.dbf
channel ORA_DISK_1: starting piece 1 at 16-11-2012 12:11:22
channel ORA_DISK_1: finished piece 1 at 16-11-2012 12:12:58
piece
handle=/u01/app/oracle/fast_recovery_area/CDB2/backupset/2012_11
_16/o1_mf_nnndf_TAG20121116T000323_8bc15dol_.bkp
tag=TAG20121116T000323 comment=NONE
channel ORA_DISK_1: backup set complete, elapsed time: 00:01:36
Finished backup at 16-11-2012 12:12:59

Starting backup at 16-11-2012 12:12:59
current log archived
using channel ORA_DISK_1
channel ORA_DISK_1: starting archived log backup set
channel ORA_DISK_1: specifying archived log(s) in backup set
input archived log thread=1 sequence=20 RECID=2 STAMP=799459982
channel ORA_DISK_1: starting piece 1 at 16-11-2012 12:13:05
channel ORA_DISK_1: finished piece 1 at 16-11-2012 12:13:07
piece
handle=/u01/app/oracle/fast_recovery_area/CDB2/backupset/2012_11
_16/o1_mf_annnn_TAG20121116T001303_8bc18l0m_.bkp
tag=TAG20121116T001303 comment=NONE
channel ORA_DISK_1: backup set complete, elapsed time: 00:00:02
Finished backup at 16-11-2012 12:13:07
```

```
Starting Control File and SPFILE Autobackup at 16-11-2012
12:13:07
piece
handle=/u01/app/oracle/fast_recovery_area/CDB2/autobackup/2012_1
1_16/o1_mf_s_799459987_8bc18v2d_.bkp comment=NONE
Finished Control File and SPFILE Autobackup at 16-11-2012
12:13:22


RMAN>
```

Practice 7-3: RMAN CDB / PDB Backup

Overview

In this practice, you will perform a whole and a partial PDB backup of `PDB2`.

Assumptions

The `PDB2` has been successfully created in `cdb2` after Practices 3-1, 3-3, and 4-4.

Tasks

1. Perform a whole PDB backup.
   a. A new RMAN command allows you to back up all data files of the pluggable database.

```
RMAN> BACKUP PLUGGABLE DATABASE pdb2;

Starting backup at 16-11-2012 12:14:09
using channel ORA_DISK_1
channel ORA_DISK_1: starting full datafile backup set
channel ORA_DISK_1: specifying datafile(s) in backup set
input datafile file number=00009
name=/u01/app/oracle/oradata/cdb2/pdb2_1/sysaux01.dbf
input datafile file number=00008
name=/u01/app/oracle/oradata/cdb2/pdb2_1/system01.dbf
input datafile file number=00023
name=/u01/app/oracle/oradata/cdb2/pdb2_1/ldata_01.dbf
channel ORA_DISK_1: starting piece 1 at 16-11-2012 12:14:10
channel ORA_DISK_1: finished piece 1 at 16-11-2012 12:16:16
piece
handle=/u01/app/oracle/fast_recovery_area/CDB2/backupset/2012_11
_16/o1_mf_nnndf_TAG20121116T001410_8bc1bmvw_.bkp
tag=TAG20121116T001410 comment=NONE
channel ORA_DISK_1: backup set complete, elapsed time: 00:02:06
Finished backup at 16-11-2012 12:16:16

Starting Control File and SPFILE Autobackup at 16-11-2012
12:16:16
piece
handle=/u01/app/oracle/fast_recovery_area/CDB2/autobackup/2012_1
1_16/o1_mf_s_799460176_8bc1gnds_.bkp comment=NONE
Finished Control File and SPFILE Autobackup at 16-11-2012
12:16:23

RMAN>
```

2. Perform a partial PDB backup of the tablespace **`ldata`**. A new RMAN command allows you to back up some data files of the pluggable database.

```
RMAN> BACKUP TABLESPACE pdb2:ldata;
```

```
Starting backup at 16-11-2012 12:16:49
using channel ORA_DISK_1
channel ORA_DISK_1: starting full datafile backup set
channel ORA_DISK_1: specifying datafile(s) in backup set
input datafile file number=00023
name=/u01/app/oracle/oradata/cdb2/pdb2_1/ldata_01.dbf
channel ORA_DISK_1: starting piece 1 at 16-11-2012 12:16:49
channel ORA_DISK_1: finished piece 1 at 16-11-2012 12:16:57
piece
handle=/u01/app/oracle/fast_recovery_area/CDB2/backupset/2012_11
_16/o1_mf_nnndf_TAG20121116T001649_8bc1hlo6_.bkp
tag=TAG20121116T001649 comment=NONE
channel ORA_DISK_1: backup set complete, elapsed time: 00:00:08
Finished backup at 16-11-2012 12:16:57


Starting Control File and SPFILE Autobackup at 16-11-2012
12:16:57
piece
handle=/u01/app/oracle/fast_recovery_area/CDB2/autobackup/2012_1
1_16/o1_mf_s_799460218_8bc1j07h_.bkp comment=NONE
Finished Control File and SPFILE Autobackup at 16-11-2012
12:17:12


RMAN> EXIT
$
```

Practice 7-4: RMAN Recovery from SYSTEM PDB Data File Loss

Overview

In this practice, you will recover the PDB from an essential data file loss. If the PDB is already closed before you encounter the issue, the CDB does not require to be closed. If the PDB is opened when the issue raises, then the PDB needs to be closed and as it is not possible, you have to shut down the CDB instance and mount it.

Tasks

1. Remove the `SYSTEM` data file of `PDB2`.

```
$ sqlplus system/oracle_4U@PDB2

Connected to:
Oracle Database 12c Enterprise Edition Release 12.1.0.1.0 -
64bit Production
With the Partitioning, OLAP, Advanced Analytics and Real
Application Testing options

SQL> select file_name from DBA_DATA_FILES
    WHERE TABLESPACE_NAME='SYSTEM';

FILE_NAME
-----------------------------------------------------------
/u01/app/oracle/oradata/cdb2/pdb2_1/system01.dbf

SQL> EXIT
$
```

```
$ rm /u01/app/oracle/oradata/cdb2/pdb2_1/system01.dbf
$
```

2. Run RMAN to connect to `cdb2` with a user with `SYSDBA` or `SYSBACKUP` privilege.

```
$ rman target /

connected to target database: CDB2 (DBID=544732113)
RMAN>
```

3. Proceed with the traditional procedure to restore the missing data file and recover the CDB as it were a non-CDB.

```
RMAN> SHUTDOWN ABORT;

using target database control file instead of recovery catalog
Oracle instance shut down

RMAN> STARTUP MOUNT;
```

```
connected to target database (not started)
Oracle instance started
database mounted

Total System Global Area 1068937216 bytes

Fixed Size                  2248280 bytes
Variable Size             343933352 bytes
Database Buffers          717225984 bytes
Redo Buffers                5529600 bytes

RMAN> RESTORE TABLESPACE pdb2:SYSTEM;

Starting restore at 16-11-2012 12:26:59
allocated channel: ORA_DISK_1
channel ORA_DISK_1: SID=237 device type=DISK

channel ORA_DISK_1: starting datafile backup set restore
channel ORA_DISK_1: specifying datafile(s) to restore from
backup set
…
channel ORA_DISK_1: restored backup piece 1
channel ORA_DISK_1: restore complete, elapsed time: 00:01:05
Finished restore at 16-11-2012 12:28:07

RMAN> RECOVER TABLESPACE pdb2:SYSTEM;

Starting recover at 16-11-2012 12:28:12
using channel ORA_DISK_1

starting media recovery
media recovery complete, elapsed time: 00:00:04

Finished recover at 16-11-2012 12:28:23

RMAN> ALTER DATABASE OPEN;

Statement processed

RMAN> EXIT
$
```

Or you can use the new syntax to restore and recover a whole PDB, as follows:

```
$ rm /u01/app/oracle/oradata/cdb2/pdb2_1/system01.dbf
$
```

```
$ rman target /

connected to target database: CDB2 (DBID=544732113)

RMAN> SHUTDOWN ABORT;

using target database control file instead of recovery catalog
Oracle instance shut down

RMAN> STARTUP MOUNT;
Oracle instance started
database mounted

Total System Global Area 1068937216 bytes

Fixed Size                  2248280 bytes
Variable Size             343933352 bytes
Database Buffers          717225984 bytes
Redo Buffers                5529600 bytes

RMAN> RESTORE pluggable database pdb2;

Starting restore at 16-11-2012 12:32:30
allocated channel: ORA_DISK_1
channel ORA_DISK_1: SID=237 device type=DISK

channel ORA_DISK_1: starting datafile backup set restore
channel ORA_DISK_1: specifying datafile(s) to restore from
backup set
…
channel ORA_DISK_1: restore complete, elapsed time: 00:00:01
Finished restore at 16-11-2012 12:33:48

RMAN> RECOVER pluggable database pdb2;

Starting recover at 16-11-2012 12:34:21
using channel ORA_DISK_1
```

```
starting media recovery
media recovery complete, elapsed time: 00:00:07

Finished recover at 16-11-2012 12:34:36

RMAN> ALTER DATABASE OPEN;

Statement processed

RMAN> select name, open_mode from v$pdbs;

NAME                           OPEN_MODE
------------------------------ ----------
PDB$SEED                       READ ONLY
PDB2                           READ WRITE
PDB2_2                         READ WRITE

RMAN> EXIT
$
```

Practice 7-5: RMAN Recovery from Non-Essential PDB Data File Loss

Overview

In this practice, you will recover from a non-essential PDB data file.

Assumptions

The LDATA tablespace has been successfully created in Practice 5-1.

Tasks

1. Remove a data file of the LDATA tablespace of PDB2.

```
$ sqlplus system/oracle_4U@PDB2

Connected to:
Oracle Database 12c Enterprise Edition Release 12.1.0.1.0 -
64bit Production
With the Partitioning, OLAP, Advanced Analytics and Real
Application Testing options

SQL> select file_name from dba_data_files
     where tablespace_name='LDATA';

FILE_NAME
-----------------------------------------------------------
/u01/app/oracle/oradata/cdb2/pdb2_1/ldata_01.dbf

SQL> exit
$
```

```
$ rm /u01/app/oracle/oradata/cdb2/pdb2_1/ldata_01.dbf
$
```

2. Proceed with the traditional procedure to restore the missing datafile and recover the tablespace as it were a non-CDB.
   a. Put the tablespace in OFFLINE mode.

```
$ sqlplus system/oracle_4U@PDB2

Connected to:
Oracle Database 12c Enterprise Edition Release 12.1.0.1.0 -
64bit Production
With the Partitioning, OLAP, Advanced Analytics and Real
Application Testing options

SQL> ALTER TABLESPACE ldata OFFLINE IMMEDIATE;
```

```
Tablespace altered.

SQL> exit
$
```

b. Run RMAN to connect to `cdb2` with a user with `SYSDBA` or `SYSBACKUP` privilege.

```
$ rman target /

connected to target database: CDB2 (DBID=544732113)

RMAN>
```

c. Restore and recover the tablespace.

```
RMAN> RESTORE TABLESPACE pdb2:LDATA;

Starting restore at 16-11-2012 05:22:50
using target database control file instead of recovery catalog
allocated channel: ORA_DISK_1
channel ORA_DISK_1: SID=261 device type=DISK

channel ORA_DISK_1: starting datafile backup set restore
…
channel ORA_DISK_1: restore complete, elapsed time: 00:00:02
Finished restore at 16-11-2012 05:22:57

RMAN> RECOVER TABLESPACE pdb2:LDATA;

Starting recover at 16-11-2012 05:23:04
using channel ORA_DISK_1

starting media recovery
…
media recovery complete, elapsed time: 00:00:06
Finished recover at 16-11-2012 05:23:16

RMAN> exit
$
```

3. Put the tablespace back ONLINE.

```
$ sqlplus system/oracle_4U@PDB2

Connected to:
Oracle Database 12c Enterprise Edition Release 12.1.0.1.0 -
64bit Production
```

```
With the Partitioning, OLAP, Advanced Analytics and Real
Application Testing options

SQL> ALTER TABLESPACE ldata ONLINE;

Tablespace altered.

SQL>
```

Practice 7-6: SQL PDB Hot Backup

Overview

In this practice, you will perform a hot backup of `PDB2` in `cdb2`.

Assumptions

The `PDB2` has been successfully created in `cdb2` after Practices 3-1, 3-3, and 4-4.

Tasks

1. List all data files belonging to `PDB2` to be backed up.

```
SQL> connect system/oracle_4U@PDB2
Connected.
SQL> select file_name from dba_data_files;

FILE_NAME
------------------------------------------------------------
/u01/app/oracle/oradata/cdb2/pdb2_1/system01.dbf
/u01/app/oracle/oradata/cdb2/pdb2_1/sysaux01.dbf
/u01/app/oracle/oradata/cdb2/pdb2_1/ldata_01.dbf

SQL>
```

2. Set the PDB in hot backup.

```
SQL> ALTER PLUGGABLE DATABASE pdb2 BEGIN BACKUP;

Pluggable database altered.

SQL> exit
$
```

3. Copy the data files of the pluggable database to a backup directory.

```
$ mkdir /home/oracle/backup
$ cp /u01/app/oracle/oradata/cdb2/pdb2_1/* /home/oracle/backup
$
```

4. Deactivate the backup mode.

```
$ sqlplus system/oracle_4U@PDB2

Connected to:
Oracle Database 12c Enterprise Edition Release 12.1.0.1.0 -
64bit Production
With the Partitioning, OLAP, Advanced Analytics and Real
Application Testing options

SQL> ALTER PLUGGABLE DATABASE pdb2 END BACKUP;
```

```
Pluggable database altered.

SQL>
```

Practice 7-7: SQL Control File Backup

Overview

In this practice, you will use the traditional SQL command to back up the `cdb2` control file.

Tasks

1. Connect to the `cdb2` root.

```
SQL> CONNECT / as sysdba
Connected.
SQL>
```

2. Run the `ALTER DATABASE` command to back up the control file to a script.

```
SQL> alter database backup controlfile to trace;

Database altered.

SQL> exit
$
```

3. Read the trace file generated.

```
$ cd /u01/app/oracle/diag/rdbms/cdb2/cdb2/trace
$ ls -ltr|tail -10
-rw-r----- 1 oracle oinstall   9672 Nov 16 05:23
cdb2_m000_20291.trc
-rw-r----- 1 oracle oinstall    216 Nov 16 05:23
cdb2_ora_20159.trm
-rw-r----- 1 oracle oinstall   4927 Nov 16 05:23
cdb2_ora_20159.trc
-rw-r----- 1 oracle oinstall     78 Nov 16 05:23
cdb2_ora_20327.trm
-rw-r----- 1 oracle oinstall    937 Nov 16 05:23
cdb2_ora_20327.trc
-rw-r----- 1 oracle oinstall    932 Nov 16 05:27
cdb2_mmon_29167.trm
-rw-r----- 1 oracle oinstall   8013 Nov 16 05:27
cdb2_mmon_29167.trc
-rw-r----- 1 oracle oinstall    178 Nov 16 05:27
cdb2_ora_20680.trm
-rw-r----- 1 oracle oinstall  10786 Nov 16 05:27
cdb2_ora_20680.trc
-rw-r----- 1 oracle oinstall 181604 Nov 16 05:27 alert_cdb2.log
$ cat cdb2_ora_20680.trc
…
STARTUP NOMOUNT
CREATE CONTROLFILE REUSE DATABASE "CDB2" RESETLOGS  ARCHIVELOG
    MAXLOGFILES 16
```

```
    MAXLOGMEMBERS 3
    MAXDATAFILES 1024
    MAXINSTANCES 8
    MAXLOGHISTORY 292
LOGFILE
  GROUP 1 '/u01/app/oracle/oradata/cdb2/redo01.log'  SIZE 50M
BLOCKSIZE 512,
  GROUP 2 '/u01/app/oracle/oradata/cdb2/redo02.log'  SIZE 50M
BLOCKSIZE 512,
  GROUP 3 '/u01/app/oracle/oradata/cdb2/redo03.log'  SIZE 50M
BLOCKSIZE 512
-- STANDBY LOGFILE
DATAFILE
  '/u01/app/oracle/oradata/cdb2/system01.dbf',
  '/u01/app/oracle/oradata/cdb2/sysaux01.dbf',
  '/u01/app/oracle/oradata/cdb2/undotbs01.dbf',
  '/u01/app/oracle/oradata/cdb2/pdbseed/system01.dbf',
  '/u01/app/oracle/oradata/cdb2/users01.dbf',
  '/u01/app/oracle/oradata/cdb2/pdbseed/sysaux01.dbf',
  '/u01/app/oracle/oradata/cdb2/pdb2_1/system01.dbf',
  '/u01/app/oracle/oradata/cdb2/pdb2_1/sysaux01.dbf',
'/u01/app/oracle/oradata/cdb2/pdb2_2/CDB2/CE88299AD6087639E0436B
23B98B8558/datafile/o1_mf_system_8b9ocl9w_.dbf',
'/u01/app/oracle/oradata/cdb2/pdb2_2/CDB2/CE88299AD6087639E0436B
23B98B8558/datafile/o1_mf_sysaux_8b9ocl7f_.dbf',
  '/u01/app/oracle/oradata/cdb2/cdata_01.dbf',
  '/u01/app/oracle/oradata/cdb2/pdb2_1/ldata_01.dbf'
CHARACTER SET AL32UTF8
;
-- Configure RMAN configuration record 1
VARIABLE RECNO NUMBER;
EXECUTE :RECNO := SYS.DBMS_BACKUP_RESTORE.SETCONFIG('DEFAULT
DEVICE TYPE TO','DISK');
-- Configure RMAN configuration record 2
VARIABLE RECNO NUMBER;
EXECUTE :RECNO := SYS.DBMS_BACKUP_RESTORE.SETCONFIG('CONTROLFILE
AUTOBACKUP','ON');
-- Commands to re-create incarnation table
-- Below log names MUST be changed to existing filenames on
-- disk. Any one log file from each branch can be used to
-- re-create incarnation records.
-- ALTER DATABASE REGISTER LOGFILE
'/u01/app/oracle/fast_recovery_area/CDB2/archivelog/2012_11_16/o
1_mf_1_1_%u_.arc';
```

```
-- ALTER DATABASE REGISTER LOGFILE
'/u01/app/oracle/fast_recovery_area/CDB2/archivelog/2012_11_16/o
1_mf_1_1_%u_.arc';
-- Recovery is required if any of the datafiles are restored
backups,
-- or if the last shutdown was not normal or immediate.
RECOVER DATABASE USING BACKUP CONTROLFILE
-- Database can now be opened zeroing the online logs.
ALTER DATABASE OPEN RESETLOGS;
-- Commands to add tempfiles to temporary tablespaces.
-- Online tempfiles have complete space information.
-- Other tempfiles may require adjustment.
ALTER SESSION SET CONTAINER = CDB$ROOT;
ALTER TABLESPACE TEMP ADD TEMPFILE
'/u01/app/oracle/oradata/cdb2/temp01.dbf'
     SIZE 92274688  REUSE AUTOEXTEND ON NEXT 655360  MAXSIZE
32767M;
ALTER SESSION SET CONTAINER = PDB$SEED;
ALTER TABLESPACE TEMP ADD TEMPFILE
'/u01/app/oracle/oradata/cdb2/pdbseed/pdbseed_temp01.dbf'
     SIZE 91226112  REUSE AUTOEXTEND ON NEXT 655360  MAXSIZE
32767M;
ALTER SESSION SET CONTAINER = PDB2;
ALTER TABLESPACE TEMP ADD TEMPFILE
'/u01/app/oracle/oradata/cdb2/pdb2_1/pdbseed_temp01.dbf'
     SIZE 20971520  REUSE AUTOEXTEND ON NEXT 655360  MAXSIZE
32767M;
ALTER SESSION SET CONTAINER = PDB2_2;
ALTER TABLESPACE TEMP ADD TEMPFILE
'/u01/app/oracle/oradata/cdb2/pdb2_2/CDB2/CE88299AD6087639E0436B
23B98B8558/datafile/o1_mf_temp_8b9ogz7v_.dbf'
     SIZE 20971520  REUSE AUTOEXTEND ON NEXT 655360  MAXSIZE
32767M;
ALTER SESSION SET CONTAINER = CDB$ROOT;
ALTER TABLESPACE TEMP_ROOT ADD TEMPFILE
'/u01/app/oracle/oradata/cdb2/temproot_01.dbf'
     SIZE 104857600  REUSE AUTOEXTEND OFF;
ALTER SESSION SET CONTAINER = PDB2;
ALTER TABLESPACE TEMP_PDB2 ADD TEMPFILE
'/u01/app/oracle/oradata/cdb2/pdb2_1/temppdb2_01.dbf'
     SIZE 104857600  REUSE AUTOEXTEND OFF;
ALTER TABLESPACE MY_TEMP ADD TEMPFILE
'/u01/app/oracle/oradata/cdb2/pdb2_1/my_temp_pdb2_01.dbf'
     SIZE 104857600  REUSE AUTOEXTEND OFF;
-- End of tempfile additions.
```

```
--
$
```

You find in the control file creation all data files of the root and pluggable databases, and redo log files.

Practice 7-8: RMAN Recovery from Control File Loss

Overview

In this practice, you will recover the CDB from the control file loss.

Assumptions

Practice 7-2 successfully completed the whole CDB backup of `cdb2`.

Tasks

1. Remove the control files of the CDB.

```
$ sqlplus / AS SYSDBA

Connected to:
Oracle Database 12c Enterprise Edition Release 12.1.0.1.0 -
64bit Production
With the Partitioning, OLAP, Advanced Analytics and Real
Application Testing options

SQL> select name from v$controlfile;

NAME
------------------------------------------------------------
/u01/app/oracle/oradata/cdb2/control01.ctl
/u01/app/oracle/fast_recovery_area/cdb2/control02.ctl

SQL> !rm /u01/app/oracle/oradata/cdb2/control01.ctl
SQL>
```

2. Shut down or abort the instance `cdb2`.

```
SQL> shutdown abort
ORACLE instance shut down.
SQL> exit
$
```

3. Proceed with the traditional procedure to restore the control files and recover the CDB as if it were a non-CDB database.

```
$ rman target /

connected to target database (not started)

RMAN> startup nomount;

Oracle instance started

Total System Global Area 1068937216 bytes
```

```
Fixed Size                  2248280 bytes
Variable Size             343933352 bytes
Database Buffers          717225984 bytes
Redo Buffers                5529600 bytes

RMAN> RESTORE CONTROLFILE FROM AUTOBACKUP;

Starting restore at 16-11-2012 05:32:38
using target database control file instead of recovery catalog
allocated channel: ORA_DISK_1
channel ORA_DISK_1: SID=237 device type=DISK

recovery area destination: /u01/app/oracle/fast_recovery_area
database name (or database unique name) used for search: CDB2
channel ORA_DISK_1: AUTOBACKUP
 …
Finished restore at 16-11-2012 05:33:06

RMAN> ALTER DATABASE MOUNT;

Statement processed
released channel: ORA_DISK_1

RMAN> RECOVER DATABASE;

Starting recover at 16-11-2012 05:33:25
Starting implicit crosscheck backup at 16-11-2012 05:33:25
allocated channel: ORA_DISK_1
channel ORA_DISK_1: SID=237 device type=DISK
Crosschecked 10 objects
Finished implicit crosscheck backup at 16-11-2012 05:33:34

Starting implicit crosscheck copy at 16-11-2012 05:33:34
using channel ORA_DISK_1
Finished implicit crosscheck copy at 16-11-2012 05:33:34

searching for all files in the recovery area
cataloging files...
cataloging done

List of Cataloged Files
```

```
======================
File Name:
/u01/app/oracle/fast_recovery_area/CDB2/autobackup/2012_11_16/o1
_mf_s_799460218_8bc1j07h_.bkp

using channel ORA_DISK_1

starting media recovery
…
media recovery complete, elapsed time: 00:00:43
Finished recover at 16-11-2012 05:34:30

RMAN> ALTER DATABASE OPEN RESETLOGS;

Statement processed

RMAN> select name, open_mode from v$pdbs;

NAME                           OPEN_MODE
------------------------------ ----------
PDB$SEED                       READ ONLY
PDB2                           READ WRITE
PDB2_2                         READ WRITE

RMAN>
```

4. Back up the whole `cdb2`.
   a. Use the `BACKUP` command.

```
RMAN> BACKUP DATABASE PLUS ARCHIVELOG DELETE ALL INPUT;
…
RMAN> exit
$
```

   b. If you encounter some space issues, like the following, reclaim some space and increase the fast recovery area destination size:

```
RMAN-00571:
===========================================================
RMAN-00569: =============== ERROR MESSAGE STACK FOLLOWS
===============
RMAN-00571:
===========================================================
RMAN-03002: failure of backup plus archivelog command at
06/11/2012 06:46:44
ORA-19809: limit exceeded for recovery files
ORA-19804: cannot reclaim 67108864 bytes disk space from
10737418240 limit
```

1) Reclaim some space deleting obsolete backups:

```
RMAN> delete obsolete;

RMAN retention policy will be applied to the command
RMAN retention policy is set to redundancy 1
using channel ORA_DISK_1
Deleting the following obsolete backups and copies:
…
Do you really want to delete the above objects (enter YES or
NO)?
YES
…
Deleted 25 objects

RMAN>
```

2) Increase the fast recovery area destination size to 20G.

```
RMAN> ALTER SYSTEM SET db_recovery_file_dest_size=20G
SCOPE=both;

using target database control file instead of recovery catalog
Statement processed

RMAN> EXIT
$
```

Practice 7-9: RMAN Recovery from Redo Log File Member Loss

Overview

In this practice, you will recover the `cdb2` from a redo log file member loss.

Tasks

1. Multiplex the redo log files if not already done.

```
$ sqlplus system/oracle_4U

Connected to:
Oracle Database 12c Enterprise Edition Release 12.1.0.1.0 -
64bit Production
With the Partitioning, OLAP, Advanced Analytics and Real
Application Testing options

SQL> select member from v$logfile;

MEMBER
----------------------------------------------------------------
/u01/app/oracle/oradata/cdb2/redo03.log
/u01/app/oracle/oradata/cdb2/redo02.log
/u01/app/oracle/oradata/cdb2/redo01.log

SQL>
```

```
SQL> ALTER DATABASE ADD LOGFILE MEMBER
'/u01/app/oracle/oradata/cdb2/redo01_2.log'
TO GROUP 1;

Database altered.

SQL> ALTER DATABASE ADD LOGFILE MEMBER
'/u01/app/oracle/oradata/cdb2/redo02_2.log'
TO GROUP 2;

Database altered.

SQL> ALTER DATABASE ADD LOGFILE MEMBER
'/u01/app/oracle/oradata/cdb2/redo03_2.log'
TO GROUP 3;

Database altered.
```

```
SQL>
```

```
SQL> alter system switch logfile;

System altered.

SQL> alter system switch logfile;

System altered.

SQL> alter system switch logfile;

System altered.

SQL> alter system switch logfile;

System altered.

SQL> exit
$
```

2. Remove a redo log file member of the cdb2.

```
$ rm /u01/app/oracle/oradata/cdb2/redo01.log
```

3. Proceed with the traditional procedure to regenerate the redo log file member.

```
$ sqlplus system/oracle_4U

Connected to:
Oracle Database 12c Enterprise Edition Release 12.1.0.1.0 -
64bit Production
With the Partitioning, OLAP, Advanced Analytics and Real
Application Testing options

SQL> ALTER DATABASE CLEAR LOGFILE GROUP 1;

Database altered.

SQL> SELECT member FROM v$logfile;

MEMBER
------------------------------------------------------------
/u01/app/oracle/oradata/cdb2/redo03.log
/u01/app/oracle/oradata/cdb2/redo02.log
/u01/app/oracle/oradata/cdb2/redo01.log
```

```
/u01/app/oracle/oradata/cdb2/redo01_2.log
/u01/app/oracle/oradata/cdb2/redo02_2.log
/u01/app/oracle/oradata/cdb2/redo03_2.log

6 rows selected.

SQL> ! ls /u01/app/oracle/oradata/cdb2/redo*
/u01/app/oracle/oradata/cdb2/redo01_2.log
/u01/app/oracle/oradata/cdb2/redo01.log
/u01/app/oracle/oradata/cdb2/redo02_2.log
/u01/app/oracle/oradata/cdb2/redo02.log
/u01/app/oracle/oradata/cdb2/redo03_2.log
/u01/app/oracle/oradata/cdb2/redo03.log

SQL>
```

In case you cannot complete the operation successfully and get the following message because the redo log file belongs to the current active group, switch the redo log group. And reattempt the failed statement.

```
SQL> ALTER DATABASE CLEAR LOGFILE GROUP 1;
ALTER DATABASE CLEAR LOGFILE GROUP 1
*
ERROR at line 1:
ORA-01624: log 1 needed for crash recovery of instance cdb2
(thread 1)
ORA-00312: online log 1 thread 1:
'/u01/app/oracle/oradata/cdb2/redo01.log'
ORA-00312: online log 1 thread 1:
'/u01/app/oracle/oradata/cdb2/redo01_2.log'

SQL> alter system switch logfile;

System altered.

SQL> ALTER DATABASE CLEAR UNARCHIVED LOGFILE GROUP 1;

Database altered.

SQL> EXIT
$
```

In this latter case, you have to perform a database backup because there is a missing archive log file.

```
$ rman target /
```

```
RMAN> BACKUP DATABASE PLUS ARCHIVELOG DELETE ALL INPUT;
…
RMAN> exit
$
```

Practice 7-10: RMAN Recovery from SYSTEM Root Data File Loss

Overview

In this practice, you will recover from a root data file loss, particularly the SYSTEM data file.

Tasks

1. Remove the SYSTEM data file from the root SYSTEM tablespace.

```
$ sqlplus system/oracle_4U

Connected to:
Oracle Database 12c Enterprise Edition Release 12.1.0.1.0 -
64bit Production
With the Partitioning, OLAP, Advanced Analytics and Real
Application Testing options

SQL> SELECT file_name FROM dba_data_files
     WHERE  TABLESPACE_NAME='SYSTEM';

FILE_NAME
-----------------------------------------------------------
/u01/app/oracle/oradata/cdb2/system01.dbf

SQL> exit
$
```

```
$ rm /u01/app/oracle/oradata/cdb2/system01.dbf
$
```

2. Run RMAN to connect to cdb2 with a user with SYSDBA or SYSBACKUP privilege.

```
$ rman target /
connected to target database: CDB2 (DBID=545704923)

RMAN>
```

3. Proceed with the traditional procedure to restore the missing data file and recover the CDB as it were a monolithic database.

```
RMAN> SHUTDOWN ABORT;

using target database control file instead of recovery catalog
Oracle instance shut down

RMAN> STARTUP MOUNT;

connected to target database (not started)
```

```
Oracle instance started
database mounted

Total System Global Area 1068937216 bytes

Fixed Size                  2248280 bytes
Variable Size             343933352 bytes
Database Buffers          717225984 bytes
Redo Buffers                5529600 bytes

RMAN> RESTORE TABLESPACE SYSTEM;

Starting restore at 16-11-2012 05:58:59
allocated channel: ORA_DISK_1
channel ORA_DISK_1: SID=237 device type=DISK
…
channel ORA_DISK_1: restored backup piece 1
channel ORA_DISK_1: restore complete, elapsed time: 00:03:36
Finished restore at 16-11-2012 06:02:36

RMAN> RECOVER TABLESPACE SYSTEM;

Starting recover at 16-11-2012 06:20:03
using channel ORA_DISK_1

starting media recovery

…
media recovery complete, elapsed time: 00:00:09
Finished recover at 16-11-2012 06:20:18

RMAN> ALTER DATABASE OPEN;

Statement processed
RMAN>
```

4. Back up the CDB.

```
RMAN> BACKUP DATABASE PLUS ARCHIVELOG DELETE ALL INPUT;
…
RMAN> exit
$
```

Practice 7-11: RMAN Recovery from Non-Essential Root Data File Loss

Overview

In this practice, you will recover from a non-essential root data file loss using Data Recovery Advisor RMAN command-line interface.

Tasks

1. Remove a data file of the SYSAUX tablespace of the root of cdb2.

```
$ sqlplus / as sysdba

Connected to:
Oracle Database 12c Enterprise Edition Release 12.1.0.0.2 -
64bit Production
With the Partitioning, OLAP, Data Mining and Real Application
Testing options

SQL> select file_name from dba_data_files
     where tablespace_name='SYSAUX';

FILE_NAME
-----------------------------------------------------------
/u01/app/oracle/oradata/cdb2/sysaux01.dbf

SQL> !rm /u01/app/oracle/oradata/cdb2/sysaux01.dbf
SQL> EXIT
$
```

2. Run RMAN to connect to cdb2 with a user with SYSDBA or SYSBACKUP privilege.

```
$ rman target /

connected to target database: CDB2 (DBID=546459337)
RMAN>
```

3. Proceed with Data Recovery Advisor commands to discover, restore and recover the failure. If the failure does not appear immediately when executing the LIST FAILURE statement, reiterate the statement a few seconds later.
   a. Discover the failure.

```
RMAN> LIST FAILURE;

using target database control file instead of recovery catalog
Database Role: PRIMARY

List of Database Failures
=========================
```

```
Failure ID Priority Status    Time Detected       Summary
---------- -------- --------- ------------------- -------
905        HIGH     OPEN      16-11-2012 06:40:05 One or more
non-system datafiles are missing

RMAN>
```

If you want more details about the failure, use the `DETAIL` clause in the same command.

```
RMAN> LIST FAILURE DETAIL;

Database Role: PRIMARY

List of Database Failures
=========================

Failure ID Priority Status    Time Detected       Summary
---------- -------- --------- ------------------- -------
8          HIGH     OPEN      16-11-2012 06:40:05 One or more
non-system datafiles are missing
  Impact: See impact for individual child failures
  List of child failures for parent failure ID 8
  Failure ID Priority Status    Time Detected       Summary
  ---------- -------- --------- ------------------- -------
  905        HIGH     OPEN      16-11-2012 06:40:05 Datafile 3:
'/u01/app/oracle/oradata/cdb2/sysaux01.dbf' is missing
    Impact: Some objects in tablespace SYSAUX might be
unavailable

RMAN>
```

b. Get an advice from RMAN Data Recovery Advisor.

```
RMAN> ADVISE FAILURE;

Database Role: PRIMARY

List of Database Failures
=========================

Failure ID Priority Status    Time Detected       Summary
---------- -------- --------- ------------------- -------
8          HIGH     OPEN      16-11-2012 06:40:05 One or more
non-system datafiles are missing
  Impact: See impact for individual child failures
  List of child failures for parent failure ID 8
```

```
  Failure ID Priority Status    Time Detected       Summary
  ---------- -------- --------- ------------------- -------
  905        HIGH     OPEN      16-11-2012 06:40:05 Datafile 3:
'/u01/app/oracle/oradata/cdb2/sysaux01.dbf' is missing
    Impact: Some objects in tablespace SYSAUX might be
unavailable

analyzing automatic repair options; this may take some time
allocated channel: ORA_DISK_1
channel ORA_DISK_1: SID=279 device type=DISK
analyzing automatic repair options complete

Mandatory Manual Actions
========================
no manual actions available

Optional Manual Actions
=======================
1. If file /u01/app/oracle/oradata/cdb2/sysaux01.dbf was
unintentionally renamed or moved, restore it

Automated Repair Options
========================
Option Repair Description
------ ------------------
1      Restore and recover datafile 3
  Strategy: The repair includes complete media recovery with no
data loss
  Repair script:
/u01/app/oracle/diag/rdbms/cdb2/cdb2/hm/reco_2985968883.hm

RMAN>
```

c. Preview the provided script to repair the failure.

```
RMAN> REPAIR FAILURE PREVIEW;

Strategy: The repair includes complete media recovery with no
data loss
Repair script:
/u01/app/oracle/diag/rdbms/cdb2/cdb2/hm/reco_2985968883.hm

contents of repair script:
   # restore and recover datafile
   sql 'alter database datafile 3 offline';
```

```
   restore ( datafile 3 );
   recover datafile 3;
   sql 'alter database datafile 3 online';

RMAN>
```

d. If the provided script satisfies you, repair the failure. This will execute the script.

```
RMAN> REPAIR FAILURE;

Strategy: The repair includes complete media recovery with no
data loss
Repair script:
/u01/app/oracle/diag/rdbms/cdb2/cdb2/hm/reco_2985968883.hm

contents of repair script:
   # restore and recover datafile
   sql 'alter database datafile 3 offline';
   restore ( datafile 3 );
   recover datafile 3;
   sql 'alter database datafile 3 online';

Do you really want to execute the above repair (enter YES or
NO)? YES
executing repair script

sql statement: alter database datafile 3 offline

Starting restore at 16-11-2012 06:43:00
using channel ORA_DISK_1

channel ORA_DISK_1: starting datafile backup set restore
channel ORA_DISK_1: specifying datafile(s) to restore from
backup set
channel ORA_DISK_1: restoring datafile 00003 to
/u01/app/oracle/oradata/cdb2/sysaux01.dbf
…
channel ORA_DISK_1: restored backup piece 1
channel ORA_DISK_1: restore complete, elapsed time: 00:01:35
Finished restore at 16-11-2012 06:44:36

Starting recover at 16-11-2012 06:44:36
using channel ORA_DISK_1

starting media recovery
```

```
media recovery complete, elapsed time: 00:00:40

Finished recover at 16-11-2012 06:45:51

sql statement: alter database datafile 3 online
repair failure complete

RMAN>
```

4. Back up the CDB.

```
RMAN> BACKUP DATABASE PLUS ARCHIVELOG DELETE ALL INPUT;
…
RMAN> exit;
$
```

Practice 7-12: PDB PITR

Overview

In this practice, you will perform a pluggable database point-in-time recovery. A table `DJ.T1` is created and rows loaded into table `DJ.T1` in `PDB2` pluggable database and a similar operation is executed in `PDB2_2` pluggable database. Later, rows are loaded into the wrong table in the wrong PDB. You have to restore the situation to the time before the rows were inserted inappropriately.

Tasks

1. Connect to `PDB2` and create a tablespace to store `DJ`'s table.

```
$ sqlplus system/oracle_4U@pdb2

Connected to:
Oracle Database 12c Enterprise Edition Release 12.1.0.1.0 -
64bit Production
With the Partitioning, OLAP, Advanced Analytics and Real
Application Testing options

SQL>  CREATE TABLESPACE dj_pdb2
      DATAFILE '/u01/app/oracle/oradata/cdb2/pdb2_1/dj_pdb2.f'
      SIZE 10m;
  2    3
Tablespace created.

SQL> CREATE USER dj identified by oracle_4U
     default tablespace dj_pdb2;
  2
User created.

SQL> GRANT create session, create table, unlimited tablespace
     TO dj;
  2
Grant succeeded.

SQL>
```

2. Create a `DJ.T1` table.

```
SQL> CREATE TABLE dj.t1(c varchar2(100)) TABLESPACE dj_pdb2;

Table created.

SQL>
```

3. Make sure you note the SCN value at the table creation.

```
SQL> SELECT timestamp_to_scn(sysdate) FROM v$database;

TIMESTAMP_TO_SCN(SYSDATE)
-------------------------
                  2668939

SQL>
```

4. Insert rows into DJ.T1 table.

```
BEGIN
 FOR i in 1.. 10000 LOOP
     insert into dj.t1 values ('aaaaaaaaaaaaaaaaaaaaaaaaaaa');
 END LOOP;
 COMMIT;
END;
/
```

```
SQL> BEGIN
 FOR i in 1.. 10000 LOOP
     insert into dj.t1 values ('aaaaaaaaaaaaaaaaaaaaaaaaaaa');
 END LOOP;
 COMMIT;
END;
/
  2    3    4    5    6    7
PL/SQL procedure successfully completed.

SQL>
```

```
SQL> select timestamp_to_scn(sysdate) from v$database;

TIMESTAMP_TO_SCN(SYSDATE)
-------------------------
                  2668974

SQL>
```

5. You realize that you loaded the wrong table in the wrong PDB. You create a tablespace in the right PDB PDB2_2 to store table DJ.T1 before recovering PDB2 to the time when the table was still empty.

```
SQL> CONNECT system/oracle_4U@pdb2_2
Connected.
SQL>
```

```
SQL> CREATE TABLESPACE dj_pdb2_2 DATAFILE
'/u01/app/oracle/oradata/cdb2/pdb2_2/dj_pdb2_2.f' SIZE 10m;

Tablespace created.

SQL> CREATE USER dj IDENTIFIED BY oracle_4U DEFAULT TABLESPACE
dj_pdb2_2;

User created.

SQL> GRANT create session, create table, unlimited tablespace TO
dj;

Grant succeeded.

SQL> CREATE TABLE dj.t1(c varchar2(100)) tablespace dj_pdb2_2;

Table created.

SQL>
```

6. Load rows into `DJ.T1` table.

```
BEGIN
 FOR i in 1.. 10000 LOOP
     insert into dj.t1 values ('aaaaaaaaaaaaaaaaaaaaaaaaaaaa');
 END LOOP;
 COMMIT;
END;
/
```

```
SQL> BEGIN
 FOR i in 1.. 10000 LOOP
     insert into dj.t1 values ('aaaaaaaaaaaaaaaaaaaaaaaaaaaa');
 END LOOP;
 COMMIT;
END;
/
  2    3    4    5    6    7
PL/SQL procedure successfully completed.

SQL> EXIT
$
```

7. Proceed to the PITR (point-in-time recovery) of the PDB `PDB2` to the time when the table `DJ.T1` was still empty.
   a. Connect to `cdb2` and close `PDB2`.

```
$ rman target /

connected to target database: CDB2 (DBID=546459337)

RMAN> ALTER PLUGGABLE DATABASE pdb2 CLOSE;

using target database control file instead of recovery catalog
Statement processed

RMAN>
```

b. Perform the PDB PITR of PDB2.

```
RMAN> RUN {
SET UNTIL SCN = 2668939;
RESTORE PLUGGABLE DATABASE pdb2;
RECOVER PLUGGABLE DATABASE pdb2 AUXILIARY
        DESTINATION='/u01/app/oracle/oradata';
ALTER PLUGGABLE DATABASE pdb2 OPEN RESETLOGS;
}
2> 3> 4> 5> 6>
executing command: SET until clause

Starting restore at 26-11-2012 08:54:15
allocated channel: ORA_DISK_1
channel ORA_DISK_1: SID=271 device type=DISK

creating datafile file number=35
name=/u01/app/oracle/oradata/cdb2/pdb2_1/dj_pdb2.f
channel ORA_DISK_1: starting datafile backup set restore
channel ORA_DISK_1: specifying datafile(s) to restore from
backup set
channel ORA_DISK_1: restoring datafile 00008 to
/u01/app/oracle/oradata/cdb2/pdb2_1/system01.dbf
channel ORA_DISK_1: restoring datafile 00009 to
/u01/app/oracle/oradata/cdb2/pdb2_1/sysaux01.dbf
channel ORA_DISK_1: restoring datafile 00023 to
/u01/app/oracle/oradata/cdb2/pdb2_1/ldata_01.dbf
channel ORA_DISK_1: reading from backup piece
/u01/app/oracle/fast_recovery_area/CDB2/backupset/2012_11_26/o1_
mf_nnndf_TAG20121126T073042_8c66r7tb_.bkp
channel ORA_DISK_1: piece
handle=/u01/app/oracle/fast_recovery_area/CDB2/backupset/2012_11
_26/o1_mf_nnndf_TAG20121126T073042_8c66r7tb_.bkp
tag=TAG20121126T073042
channel ORA_DISK_1: restored backup piece 1
```

```
channel ORA_DISK_1: restore complete, elapsed time: 00:00:35
Finished restore at 26-11-2012 08:54:52

Starting recover at 26-11-2012 08:54:52
current log archived
using channel ORA_DISK_1
RMAN-05026: WARNING: presuming following set of tablespaces
applies to specified Point-in-Time

List of tablespaces expected to have UNDO segments
Tablespace SYSTEM
Tablespace UNDOTBS1

Creating automatic instance, with SID='ihAB'

initialization parameters used for automatic instance:
db_name=CDB2
db_unique_name=ihAB_pitr_pdb2_CDB2
compatible=12.1.0.0.0
db_block_size=8192
db_files=200
sga_target=1G
processes=80
diagnostic_dest=/u01/app/oracle
db_create_file_dest=/u01/app/oracle/oradata
log_archive_dest_1='location=/u01/app/oracle/oradata'
enable_pluggable_database=true
_clone_one_pdb_recovery=true
#No auxiliary parameter file used


starting up automatic instance CDB2

Oracle instance started

Total System Global Area    1068937216 bytes

Fixed Size                     2287336 bytes
Variable Size                281020696 bytes
Database Buffers             780140544 bytes
Redo Buffers                   5488640 bytes
Automatic instance created
```

```
contents of Memory Script:
{
# set requested point in time
set until  scn 2668939;
# restore the controlfile
restore clone controlfile;
# mount the controlfile
sql clone 'alter database mount clone database';
}
executing Memory Script

executing command: SET until clause

Starting restore at 26-11-2012 08:56:03
allocated channel: ORA_AUX_DISK_1
channel ORA_AUX_DISK_1: SID=75 device type=DISK

channel ORA_AUX_DISK_1: starting datafile backup set restore
channel ORA_AUX_DISK_1: restoring control file
channel ORA_AUX_DISK_1: reading from backup piece
/u01/app/oracle/fast_recovery_area/CDB2/autobackup/2012_11_26/o1
_mf_s_800353504_8c69v2h6_.bkp
channel ORA_AUX_DISK_1: piece
handle=/u01/app/oracle/fast_recovery_area/CDB2/autobackup/2012_1
1_26/o1_mf_s_800353504_8c69v2h6_.bkp tag=TAG20121126T082504
channel ORA_AUX_DISK_1: restored backup piece 1
channel ORA_AUX_DISK_1: restore complete, elapsed time: 00:00:01
output file
name=/u01/app/oracle/oradata/CDB2/controlfile/o1_mf_8c6co49j_.ct
l
Finished restore at 26-11-2012 08:56:05

sql statement: alter database mount clone database

contents of Memory Script:
{
# set requested point in time
set until  scn 2668939;
# switch to valid datafilecopies
switch clone datafile  8 to datafilecopy
 "/u01/app/oracle/oradata/cdb2/pdb2_1/system01.dbf";
switch clone datafile  9 to datafilecopy
 "/u01/app/oracle/oradata/cdb2/pdb2_1/sysaux01.dbf";
```

```
switch clone datafile  23 to datafilecopy
 "/u01/app/oracle/oradata/cdb2/pdb2_1/ldata_01.dbf";
switch clone datafile  35 to datafilecopy
 "/u01/app/oracle/oradata/cdb2/pdb2_1/dj_pdb2.f";
# set destinations for recovery set and auxiliary set datafiles
set newname for clone datafile  1 to new;
set newname for clone datafile  4 to new;
set newname for clone datafile  3 to new;
set newname for clone datafile  22 to new;
# restore the tablespaces in the recovery set and the auxiliary
set
restore clone datafile  1, 4, 3, 22;
switch clone datafile all;
}
executing Memory Script

executing command: SET until clause

datafile 8 switched to datafile copy
input datafile copy RECID=3 STAMP=800355377 file
name=/u01/app/oracle/oradata/cdb2/pdb2_1/system01.dbf

datafile 9 switched to datafile copy
input datafile copy RECID=4 STAMP=800355377 file
name=/u01/app/oracle/oradata/cdb2/pdb2_1/sysaux01.dbf

datafile 23 switched to datafile copy
input datafile copy RECID=5 STAMP=800355377 file
name=/u01/app/oracle/oradata/cdb2/pdb2_1/ldata_01.dbf

datafile 35 switched to datafile copy
input datafile copy RECID=6 STAMP=800355377 file
name=/u01/app/oracle/oradata/cdb2/pdb2_1/dj_pdb2.f

executing command: SET NEWNAME

executing command: SET NEWNAME

executing command: SET NEWNAME

executing command: SET NEWNAME

Starting restore at 26-11-2012 08:56:12
```

```
using channel ORA_AUX_DISK_1

channel ORA_AUX_DISK_1: starting datafile backup set restore
channel ORA_AUX_DISK_1: specifying datafile(s) to restore from
backup set
channel ORA_AUX_DISK_1: restoring datafile 00001 to
/u01/app/oracle/oradata/CDB2/datafile/o1_mf_system_%u_.dbf
channel ORA_AUX_DISK_1: restoring datafile 00004 to
/u01/app/oracle/oradata/CDB2/datafile/o1_mf_undotbs1_%u_.dbf
channel ORA_AUX_DISK_1: restoring datafile 00003 to
/u01/app/oracle/oradata/CDB2/datafile/o1_mf_sysaux_%u_.dbf
channel ORA_AUX_DISK_1: restoring datafile 00022 to
/u01/app/oracle/oradata/CDB2/datafile/o1_mf_cdata_%u_.dbf
channel ORA_AUX_DISK_1: reading from backup piece
/u01/app/oracle/fast_recovery_area/CDB2/backupset/2012_11_26/o1_
mf_nnndf_TAG20121126T073042_8c66o326_.bkp
channel ORA_AUX_DISK_1: piece
handle=/u01/app/oracle/fast_recovery_area/CDB2/backupset/2012_11
_26/o1_mf_nnndf_TAG20121126T073042_8c66o326_.bkp
tag=TAG20121126T073042
channel ORA_AUX_DISK_1: restored backup piece 1
channel ORA_AUX_DISK_1: restore complete, elapsed time: 00:01:05
Finished restore at 26-11-2012 08:57:17

datafile 1 switched to datafile copy
input datafile copy RECID=11 STAMP=800355438 file
name=/u01/app/oracle/oradata/CDB2/datafile/o1_mf_system_8c6codro
_.dbf
datafile 4 switched to datafile copy
input datafile copy RECID=12 STAMP=800355438 file
name=/u01/app/oracle/oradata/CDB2/datafile/o1_mf_undotbs1_8c6cod
ty_.dbf
datafile 3 switched to datafile copy
input datafile copy RECID=13 STAMP=800355438 file
name=/u01/app/oracle/oradata/CDB2/datafile/o1_mf_sysaux_8c6codoz
_.dbf
datafile 22 switched to datafile copy
input datafile copy RECID=14 STAMP=800355438 file
name=/u01/app/oracle/oradata/CDB2/datafile/o1_mf_cdata_8c6codx0_
.dbf

contents of Memory Script:
{
# set requested point in time
set until  scn 2668939;
# online the datafiles restored or switched
```

```
sql clone "alter database datafile  1 online";
sql clone "alter database datafile  4 online";
sql clone "alter database datafile  3 online";
sql clone 'PDB2' "alter database datafile
 8 online";
sql clone 'PDB2' "alter database datafile
 9 online";
sql clone 'PDB2' "alter database datafile
 23 online";
sql clone 'PDB2' "alter database datafile
 35 online";
sql clone "alter database datafile  22 online";
# recover pdb
recover clone database tablespace  "SYSTEM", "UNDOTBS1",
"SYSAUX", "CDATA" pluggable database
 'PDB2'   delete archivelog;
sql clone 'alter database open read only';
plsql <<<begin
   add_dropped_ts;
end; >>>;
plsql <<<begin
   save_pdb_clean_scn;
end; >>>;
# shutdown clone before import
shutdown clone abort
plsql <<<begin
   pdbpitr_inspect(pdbname =>  'PDB2');
end; >>>;
}
executing Memory Script

executing command: SET until clause

sql statement: alter database datafile  1 online

sql statement: alter database datafile  4 online

sql statement: alter database datafile  3 online

sql statement: alter database datafile  8 online

sql statement: alter database datafile  9 online
```

```
sql statement: alter database datafile  23 online

sql statement: alter database datafile  35 online

sql statement: alter database datafile  22 online

Starting recover at 26-11-2012 08:57:19
using channel ORA_AUX_DISK_1

starting media recovery

archived log for thread 1 with sequence 35 is already on disk as
file
/u01/app/oracle/fast_recovery_area/CDB2/archivelog/2012_11_26/o1
_mf_1_35_8c66v870_.arc
archived log for thread 1 with sequence 36 is already on disk as
file
/u01/app/oracle/fast_recovery_area/CDB2/archivelog/2012_11_26/o1
_mf_1_36_8c6clxdr_.arc
archived log file
name=/u01/app/oracle/fast_recovery_area/CDB2/archivelog/2012_11_
26/o1_mf_1_35_8c66v870_.arc thread=1 sequence=35
archived log file
name=/u01/app/oracle/fast_recovery_area/CDB2/archivelog/2012_11_
26/o1_mf_1_36_8c6clxdr_.arc thread=1 sequence=36
media recovery complete, elapsed time: 00:00:10
Finished recover at 26-11-2012 08:57:30

sql statement: alter database open read only




Oracle instance shut down



Removing automatic instance
Automatic instance removed
auxiliary instance file
/u01/app/oracle/oradata/CDB2/datafile/o1_mf_sysaux_8c6codoz_.dbf
deleted
auxiliary instance file
/u01/app/oracle/oradata/CDB2/controlfile/o1_mf_8c6co49j_.ctl
deleted
Finished recover at 26-11-2012 08:57:48
```

```
Statement processed

RMAN> EXIT
$
```

8. Check that only PDB2 was restored to SCN 2668939 and not PDB2_2.

```
$ sqlplus system/oracle_4U@pdb2

Connected to:
Oracle Database 12c Enterprise Edition Release 12.1.0.1.0 -
64bit Production
With the Partitioning, OLAP, Advanced Analytics and Real
Application Testing options

SQL> SELECT * FROM DJ.T1;

no rows selected

SQL> CONNECT system/oracle_4U@pdb2_2
Connected.
SQL> SELECT COUNT(*) FROM DJ.T1;

  COUNT(*)
----------
     10000

SQL> EXIT
$
```

You undid the load into DJ.T1 in PDB2.

9. Back up the CDB.

```
$ rman target /

RMAN> BACKUP DATABASE PLUS ARCHIVELOG DELETE ALL INPUT;
…
RMAN> EXIT
$
```

Practice 7-13: PITR on PDB Tablespaces

Overview

In this practice, you will perform a PITR on a non-essential PDB data file. Rows in a table DJ.T2 in the PDB2_2 pluggable database have been incorrectly deleted. You have to restore the situation to the time before the rows were deleted and committed.

Assumptions

The PDB pdb2_2 has been successfully created after completion of Practice 3-4.

The PDB pdb2_2 has been successfully backed up in practice 7-12.

Drop the OPEN_ALL_PDBS trigger. You have to DROP it. Disabling it will not be sufficient and you will get errors preventing the operation from successfully completing.

Tasks

1. Drop the OPEN_ALL_PDBS trigger.

```
$ sqlplus / as sysdba

Connected to:
Oracle Database 12c Enterprise Edition Release 12.1.0.1.0 -
64bit Production
With the Partitioning, OLAP, Advanced Analytics and Real
Application Testing options

SQL> DROP TRIGGER open_all_pdbs;

Trigger dropped.

SQL>
```

2. Create a DJ.T2 table PDB2_2 with 10000 rows, check the SCN before the DELETE statement. Use the following PL/SQL block to load rows.

```
BEGIN
 FOR i in 1.. 10000 LOOP
     insert into dj.t2 values (100);
 END LOOP;
 COMMIT;
END;
/
```

```
SQL> connect sys/oracle_4U@PDB2_2 as sysdba

Connected.
SQL> create table DJ.T2 (c NUMBER);

Table created.
```

```
SQL>
SQL> BEGIN
 FOR i in 1.. 10000 LOOP
     insert into dj.t2 values (100);
 END LOOP;
 COMMIT;
END;
/
  2    3    4    5    6    7
PL/SQL procedure successfully completed.

SQL> select timestamp_to_scn(sysdate) from v$database;

TIMESTAMP_TO_SCN(SYSDATE)
-------------------------
                  2674656

SQL> delete from DJ.T2 where rownum < 3;

2 rows deleted.

SQL> commit;

Commit complete.

SQL> select timestamp_to_scn(sysdate) from v$database;

TIMESTAMP_TO_SCN(SYSDATE)
-------------------------
                  2674686

SQL> select count(*) from DJ.T2;

  COUNT(*)
----------
      9998

SQL>
```

3. Before restoring and recovering the tablespace, set it offline.

```
SQL> ALTER TABLESPACE dj_pdb2_2 OFFLINE IMMEDIATE;
```

```
Tablespace altered.

SQL> EXIT
$
```

4. Set the situation back when rows were all present in the table.
   a. Perform a tablespace Point-In-Time Recovery in `PDB2_2`.

```
$ rman target /

connected to target database: CDB2 (DBID=546459337)

RMAN>
RMAN> RECOVER TABLESPACE pdb2_2:dj_pdb2_2
            UNTIL SCN 2674656
            AUXILIARY DESTINATION '/u01/app/oracle/oradata';

Starting recover at 26-11-2012 09:48:56
using target database control file instead of recovery catalog
current log archived
allocated channel: ORA_DISK_1
channel ORA_DISK_1: SID=51 device type=DISK
RMAN-05026: WARNING: presuming following set of tablespaces
applies to specified Point-in-Time

List of tablespaces expected to have UNDO segments
Tablespace SYSTEM
Tablespace UNDOTBS1

Creating automatic instance, with SID='pdvF'

initialization parameters used for automatic instance:
db_name=CDB2
db_unique_name=pdvF_pitr_pdb2_2_CDB2
compatible=12.1.0.0.0
db_block_size=8192
db_files=200
sga_target=1G
processes=80
diagnostic_dest=/u01/app/oracle
db_create_file_dest=/u01/app/oracle/oradata
log_archive_dest_1='location=/u01/app/oracle/oradata'
enable_pluggable_database=true
_clone_one_pdb_recovery=true
```

```
#No auxiliary parameter file used


starting up automatic instance CDB2

Oracle instance started

Total System Global Area    1068937216 bytes

Fixed Size                     2287336 bytes
Variable Size                281020696 bytes
Database Buffers             780140544 bytes
Redo Buffers                   5488640 bytes
Automatic instance created
Running TRANSPORT_SET_CHECK on recovery set tablespaces
TRANSPORT_SET_CHECK completed successfully

contents of Memory Script:
{
# set requested point in time
set until  scn 2674656;
# restore the controlfile
restore clone controlfile;
# mount the controlfile
sql clone 'alter database mount clone database';
# archive current online log
sql 'alter system archive log current';
# avoid unnecessary autobackups for structural changes during
TSPITR
sql 'begin dbms_backup_restore.AutoBackupFlag(FALSE); end;';
}
executing Memory Script

executing command: SET until clause

Starting restore at 26-11-2012 09:50:44
allocated channel: ORA_AUX_DISK_1
channel ORA_AUX_DISK_1: SID=75 device type=DISK

channel ORA_AUX_DISK_1: starting datafile backup set restore
channel ORA_AUX_DISK_1: restoring control file
…
```

```
output file
name=/u01/app/oracle/oradata/CDB2/controlfile/o1_mf_8c6gvo26_.ct
l
Finished restore at 26-11-2012 09:50:48

sql statement: alter database mount clone database

sql statement: alter system archive log current

sql statement: begin dbms_backup_restore.AutoBackupFlag(FALSE);
end;

contents of Memory Script:
{
# set requested point in time
set until  scn 2674656;
# set destinations for recovery set and auxiliary set datafiles
set newname for clone datafile  1 to new;
set newname for clone datafile  4 to new;
set newname for clone datafile  3 to new;
set newname for clone datafile  10 to new;
set newname for clone datafile  11 to new;
set newname for clone tempfile  1 to new;
set newname for clone tempfile  4 to new;
set newname for clone tempfile  7 to new;
set newname for datafile  37 to
 "/u01/app/oracle/oradata/cdb2/pdb2_2/dj_pdb2_2.f";
# switch all tempfiles
switch clone tempfile all;
# restore the tablespaces in the recovery set and the auxiliary
set
restore clone datafile  1, 4, 3, 10, 11, 37;
switch clone datafile all;
}
executing Memory Script

executing command: SET until clause

executing command: SET NEWNAME

executing command: SET NEWNAME

executing command: SET NEWNAME
```

```
executing command: SET NEWNAME

executing command: SET NEWNAME

executing command: SET NEWNAME

executing command: SET NEWNAME

executing command: SET NEWNAME

executing command: SET NEWNAME

renamed tempfile 1 to
/u01/app/oracle/oradata/CDB2/datafile/o1_mf_temp_%u_.tmp in
control file
renamed tempfile 4 to
/u01/app/oracle/oradata/CDB2/datafile/o1_mf_temp_%u_.tmp in
control file
renamed tempfile 7 to
/u01/app/oracle/oradata/CDB2/datafile/o1_mf_temp_roo_%u_.tmp in
control file

Starting restore at 26-11-2012 09:50:55
using channel ORA_AUX_DISK_1

channel ORA_AUX_DISK_1: starting datafile backup set restore
channel ORA_AUX_DISK_1: specifying datafile(s) to restore from
backup set
…
channel ORA_AUX_DISK_1: restored backup piece 1
channel ORA_AUX_DISK_1: restore complete, elapsed time: 00:00:35
Finished restore at 26-11-2012 09:52:45

datafile 1 switched to datafile copy
input datafile copy RECID=8 STAMP=800358766 file
name=/u01/app/oracle/oradata/CDB2/datafile/o1_mf_system_8c6gvzx5
_.dbf
datafile 4 switched to datafile copy
input datafile copy RECID=9 STAMP=800358766 file
name=/u01/app/oracle/oradata/CDB2/datafile/o1_mf_undotbs1_8c6gvz
yg_.dbf
datafile 3 switched to datafile copy
```

```
input datafile copy RECID=10 STAMP=800358766 file
name=/u01/app/oracle/oradata/CDB2/datafile/o1_mf_sysaux_8c6gvzty
_.dbf
datafile 10 switched to datafile copy
input datafile copy RECID=11 STAMP=800358766 file
name=/u01/app/oracle/oradata/CDB2/datafile/o1_mf_system_8c6gyc52
_.dbf
datafile 11 switched to datafile copy
input datafile copy RECID=12 STAMP=800358766 file
name=/u01/app/oracle/oradata/CDB2/datafile/o1_mf_sysaux_8c6gyc0x
_.dbf

contents of Memory Script:
{
# set requested point in time
set until  scn 2674656;
# online the datafiles restored or switched
sql clone "alter database datafile  1 online";
sql clone "alter database datafile  4 online";
sql clone "alter database datafile  3 online";
sql clone 'PDB2_2' "alter database datafile
 10 online";
sql clone 'PDB2_2' "alter database datafile
 11 online";
sql clone 'PDB2_2' "alter database datafile
 37 online";
# recover and open resetlogs
recover clone database tablespace  "PDB2_2":"DJ_PDB2_2",
"SYSTEM", "UNDOTBS1", "SYSAUX", "PDB2_2":"SYSTEM",
"PDB2_2":"SYSAUX" delete archivelog;
alter clone database open resetlogs;
}
executing Memory Script

executing command: SET until clause

sql statement: alter database datafile  1 online

sql statement: alter database datafile  4 online

sql statement: alter database datafile  3 online

sql statement: alter database datafile  10 online
```

```
sql statement: alter database datafile  11 online

sql statement: alter database datafile  37 online

Starting recover at 26-11-2012 09:52:47
using channel ORA_AUX_DISK_1

starting media recovery

archived log for thread 1 with sequence 39 is already on disk as
file
/u01/app/oracle/fast_recovery_area/CDB2/archivelog/2012_11_26/o1
_mf_1_39_8c6gr95c_.arc
channel ORA_AUX_DISK_1: starting archived log restore to default
destination
channel ORA_AUX_DISK_1: restoring archived log
…
channel ORA_AUX_DISK_1: restored backup piece 1
channel ORA_AUX_DISK_1: restore complete, elapsed time: 00:00:01
archived log file
name=/u01/app/oracle/oradata/1_38_799412297.dbf thread=1
sequence=38
channel clone_default: deleting archived log(s)
archived log file
name=/u01/app/oracle/oradata/1_38_799412297.dbf RECID=21
STAMP=800358770
archived log file
name=/u01/app/oracle/fast_recovery_area/CDB2/archivelog/2012_11_
26/o1_mf_1_39_8c6gr95c_.arc thread=1 sequence=39
media recovery complete, elapsed time: 00:00:05
Finished recover at 26-11-2012 09:52:56

database opened

contents of Memory Script:
{
sql clone 'alter pluggable database  PDB2_2 open';
}
executing Memory Script

sql statement: alter pluggable database  PDB2_2 open

contents of Memory Script:
{
```

```
# make read only the tablespace that will be exported
sql clone 'PDB2_2' 'alter tablespace DJ_PDB2_2 read only';
# create directory for datapump import
sql 'PDB2_2' "create or replace directory TSPITR_DIROBJ_DPDIR as
''
/u01/app/oracle/oradata''";
# create directory for datapump export
sql clone 'PDB2_2' "create or replace directory
TSPITR_DIROBJ_DPDIR as ''
/u01/app/oracle/oradata''";
}
executing Memory Script

sql statement: alter tablespace  DJ_PDB2_2 read only

sql statement: create or replace directory TSPITR_DIROBJ_DPDIR
as ''/u01/app/oracle/oradata''

sql statement: create or replace directory TSPITR_DIROBJ_DPDIR
as ''/u01/app/oracle/oradata''

Performing export of metadata...
   EXPDP> Starting "SYS"."TSPITR_EXP_pdvF_dqqz":
   EXPDP> Processing object type TRANSPORTABLE_EXPORT/PLUGTS_BLK
   EXPDP> Processing object type TRANSPORTABLE_EXPORT/TABLE
   EXPDP> Processing object type
TRANSPORTABLE_EXPORT/TABLE_STATISTICS
   EXPDP> Processing object type
TRANSPORTABLE_EXPORT/STATISTICS/MARKER
   EXPDP> Processing object type
TRANSPORTABLE_EXPORT/POST_INSTANCE/PLUGTS_BLK
   EXPDP> Master table "SYS"."TSPITR_EXP_pdvF_dqqz" successfully
loaded/unloaded
   EXPDP>
******************************************************************
**************
   EXPDP> Dump file set for SYS.TSPITR_EXP_pdvF_dqqz is:
   EXPDP>   /u01/app/oracle/oradata/tspitr_pdvF_35856.dmp
   EXPDP>
******************************************************************
**************
   EXPDP> Datafiles required for transportable tablespace
DJ_PDB2_2:
   EXPDP>   /u01/app/oracle/oradata/cdb2/pdb2_2/dj_pdb2_2.f
```

```
   EXPDP> Job "SYS"."TSPITR_EXP_pdvF_dqqz" successfully
completed at Mon Nov 26 09:54:56 2012 elapsed 0 00:00:42
Export completed


contents of Memory Script:
{
# shutdown clone before import
shutdown clone abort
# drop target tablespaces before importing them back
sql 'PDB2_2' 'drop tablespace DJ_PDB2_2 including contents keep
datafiles cascade constraints';
}
executing Memory Script

Oracle instance shut down

sql statement: drop tablespace  DJ_PDB2_2 including contents
keep datafiles cascade constraints

Performing import of metadata...
   IMPDP> Master table "SYS"."TSPITR_IMP_pdvF_sdFC" successfully
loaded/unloaded
   IMPDP> Starting "SYS"."TSPITR_IMP_pdvF_sdFC":
   IMPDP> Processing object type TRANSPORTABLE_EXPORT/PLUGTS_BLK
   IMPDP> Processing object type TRANSPORTABLE_EXPORT/TABLE
   IMPDP> Processing object type
TRANSPORTABLE_EXPORT/TABLE_STATISTICS
   IMPDP> Processing object type
TRANSPORTABLE_EXPORT/STATISTICS/MARKER
   IMPDP> Processing object type
TRANSPORTABLE_EXPORT/POST_INSTANCE/PLUGTS_BLK
   IMPDP> Job "SYS"."TSPITR_IMP_pdvF_sdFC" successfully
completed at Mon Nov 26 09:55:31 2012 elapsed 0 00:00:12
Import completed


contents of Memory Script:
{
# make read write and offline the imported tablespaces
sql 'PDB2_2' 'alter tablespace  DJ_PDB2_2 read write';
sql 'PDB2_2' 'alter tablespace  DJ_PDB2_2 offline';
# enable autobackups after TSPITR is finished
sql 'begin dbms_backup_restore.AutoBackupFlag(TRUE); end;';
```

```
}
executing Memory Script

sql statement: alter tablespace  DJ_PDB2_2 read write

sql statement: alter tablespace  DJ_PDB2_2 offline

sql statement: begin dbms_backup_restore.AutoBackupFlag(TRUE);
end;

Removing automatic instance
Automatic instance removed
auxiliary instance file
/u01/app/oracle/oradata/CDB2/datafile/o1_mf_temp_roo_8c6h03x8_.t
mp deleted
auxiliary instance file
/u01/app/oracle/oradata/CDB2/datafile/o1_mf_temp_8c6h0q18_.tmp
deleted
auxiliary instance file
/u01/app/oracle/oradata/CDB2/datafile/o1_mf_temp_8c6h03nm_.tmp
deleted
auxiliary instance file
/u01/app/oracle/oradata/CDB2/onlinelog/o1_mf_3_8c6gzwsx_.log
deleted
auxiliary instance file
/u01/app/oracle/oradata/CDB2/onlinelog/o1_mf_2_8c6gzvh3_.log
deleted
auxiliary instance file
/u01/app/oracle/oradata/CDB2/onlinelog/o1_mf_1_8c6gzsds_.log
deleted
auxiliary instance file
/u01/app/oracle/oradata/CDB2/datafile/o1_mf_sysaux_8c6gyc0x_.dbf
deleted
auxiliary instance file
/u01/app/oracle/oradata/CDB2/datafile/o1_mf_system_8c6gyc52_.dbf
deleted
auxiliary instance file
/u01/app/oracle/oradata/CDB2/datafile/o1_mf_sysaux_8c6gvzty_.dbf
deleted
auxiliary instance file
/u01/app/oracle/oradata/CDB2/datafile/o1_mf_undotbs1_8c6gvzyg_.d
bf deleted
auxiliary instance file
/u01/app/oracle/oradata/CDB2/datafile/o1_mf_system_8c6gvzx5_.dbf
deleted
```

```
auxiliary instance file
/u01/app/oracle/oradata/CDB2/controlfile/o1_mf_8c6gvo26_.ctl
deleted
auxiliary instance file tspitr_pdvF_35856.dmp deleted
Finished recover at 26-11-2012 09:55:38

RMAN> EXIT
$
```

b. Online the tablespace.

```
$ sqlplus sys/oracle_4U@PDB2_2 as sysdba

Connected to:
Oracle Database 12c Enterprise Edition Release 12.1.0.1.0 -
64bit Production
With the Partitioning, OLAP, Advanced Analytics and Real
Application Testing options

SQL> ALTER TABLESPACE dj_pdb2_2 ONLINE;

Tablespace altered.

SQL>
```

c. Check the content of the `DJ.T2` table in `PDB2_2`.

```
SQL> select count(*) from DJ.T2;

  COUNT(*)
----------
     10000



SQL> EXIT
$
```

d. Back up the CDB.

```
$ rman target /

connected to target database: CDB2 (DBID=545704923)

RMAN> DELETE OBSOLETE;
…
RMAN> BACKUP DATABASE PLUS ARCHIVELOG delete all input;
…
RMAN> EXIT
```

```
$
```

e. Recreate the trigger if you want to get all PDBs opened automatically after database startup.

```
$ sqlplus / as sysdba

Connected to:
Oracle Database 12c Enterprise Edition Release 12.1.0.1.0 -
64bit Production
With the Partitioning, OLAP, Advanced Analytics and Real
Application Testing options

SQL> CREATE TRIGGER open_all_PDBs
  AFTER STARTUP ON DATABASE
begin
    execute immediate 'alter pluggable database all open';
 end open_all_PDBs;
/
2    3    4    5    6

Trigger created.

SQL> exit
$
```

Practice 7-14: Flashback from Common User Drop

Overview

In this practice, you will flash back the CDB after a common user has been dropped.

Assumptions

The `C##_USER` common user exists in `cdb2`. This has been completed in practice 6-2.

Tasks

1. Set the CDB `cdb2` in FLASHBACK mode.

```
$ export NLS_DATE_FORMAT='DD-MM-YYYY HH:MI:SS'
$ sqlplus / as sysdba

Connected to:
Oracle Database 12c Enterprise Edition Release 12.1.0.1.0 -
64bit Production
With the Partitioning, OLAP, Advanced Analytics and Real
Application Testing options

SQL> SELECT flashback_on from V$DATABASE;

FLASHBACK_ON
------------------
NO

SQL> SHUTDOWN IMMEDIATE
Database closed.
Database dismounted.
ORACLE instance shut down.

SQL> STARTUP MOUNT
ORACLE instance started.

Total System Global Area 1068937216 bytes
Fixed Size                  2248280 bytes
Variable Size             343933352 bytes
Database Buffers          717225984 bytes
Redo Buffers                5529600 bytes
Database mounted.
SQL> ALTER SYSTEM SET
           DB_FLASHBACK_RETENTION_TARGET=2880 SCOPE=BOTH;

System altered.
```

```
SQL> ALTER DATABASE FLASHBACK ON;

Database altered.

SQL> ALTER DATABASE OPEN;

Database altered.

SQL>
```

2. Drop the common user C##_USER.
   a. Verify that C##_USER exists as a common user.

```
SQL> col username format A20
SQL> select USERNAME, COMMON, CON_ID from cdb_users
     where username='C##_USER';

USERNAME             COM     CON_ID
-------------------- --- ----------
C##_USER             YES          1
C##_USER             YES          3
C##_USER             YES          4

SQL> select timestamp_to_scn(current_timestamp)
     from v$database;

TIMESTAMP_TO_SCN(CURRENT_TIMESTAMP)
-----------------------------------
                            2321219

SQL>
```

   b. Drop the user.

```
SQL> DROP USER C##_USER CASCADE;

User dropped.

SQL> alter system switch logfile;

System altered.

SQL> alter system switch logfile;

System altered.
```

```
SQL> alter system switch logfile;

System altered.

SQL> alter system switch logfile;

System altered.

SQL>
```

3. Proceed with the flashback database operation.

```
SQL> SHUTDOWN IMMEDIATE
Database closed.
Database dismounted.
ORACLE instance shut down.

SQL> STARTUP MOUNT
ORACLE instance started.

Total System Global Area 1068937216 bytes
Fixed Size                  2248280 bytes
Variable Size             343933352 bytes
Database Buffers          717225984 bytes
Redo Buffers                5529600 bytes
Database mounted.

SQL> FLASHBACK DATABASE TO SCN 2321219;

Flashback complete.

SQL>
```

4. Open the database in `READ ONLY` mode to review changes before opening CDB with `RESETLOGS`.

```
SQL> ALTER DATABASE OPEN READ ONLY;

Database altered.

SQL> select USERNAME, COMMON, CON_ID from cdb_users
  2  where  username='C##_USER';

USERNAME             COM     CON_ID
-------------------- --- ----------
```

```
C##_USER             YES          1

SQL>
```

5. Open PDBs in READ ONLY to review all changes.

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

Pluggable database altered.

SQL> select USERNAME, COMMON, CON_ID from cdb_users
  2  where  username='C##_USER';

USERNAME             COM     CON_ID
-------------------- --- ----------
C##_USER             YES          1
C##_USER             YES          3
C##_USER             YES          4

SQL>
```

6. Open the CDB with RESETLOGS.

```
SQL> SHUTDOWN IMMEDIATE
Database closed.
Database dismounted.
ORACLE instance shut down.

SQL> STARTUP MOUNT
ORACLE instance started.

Total System Global Area 1068937216 bytes
Fixed Size                  2248280 bytes
Variable Size             343933352 bytes
Database Buffers          717225984 bytes
Redo Buffers                5529600 bytes
Database mounted.

SQL> FLASHBACK DATABASE TO SCN 2321219;

Flashback complete.

SQL> ALTER DATABASE OPEN RESETLOGS;

Database altered.
```

```
SQL>
```

7. Check that the C##_USER can connect in each container.

```
SQL> connect C##_USER/x
Connected.
SQL> connect C##_USER/x@PDB2
Connected.
SQL> connect C##_USER/x@PDB2_2
Connected.
SQL> exit;
$
```

8. Back up the CDB.

```
$ rman target /

connected to target database: CDB2 (DBID=545704923)

RMAN> BACKUP DATABASE PLUS ARCHIVELOG delete all input;
…
RMAN> exit
$
```

Practice 7-15: Using RMAN Backup Set to Plug a PDB

Overview

In this practice, you will use an RMAN backup set and an XMLfile of an unplugged PDB to plug the PDB into a CDB. You decide to unplug `pdb2_2` and replug it into the same cdb2 using a backup set and an XML file. It would be the same operation if you had unplugged a PDB to plug it into another CDB.

To perform these operations, you will use Enterprise Manager Cloud Control.

Tasks

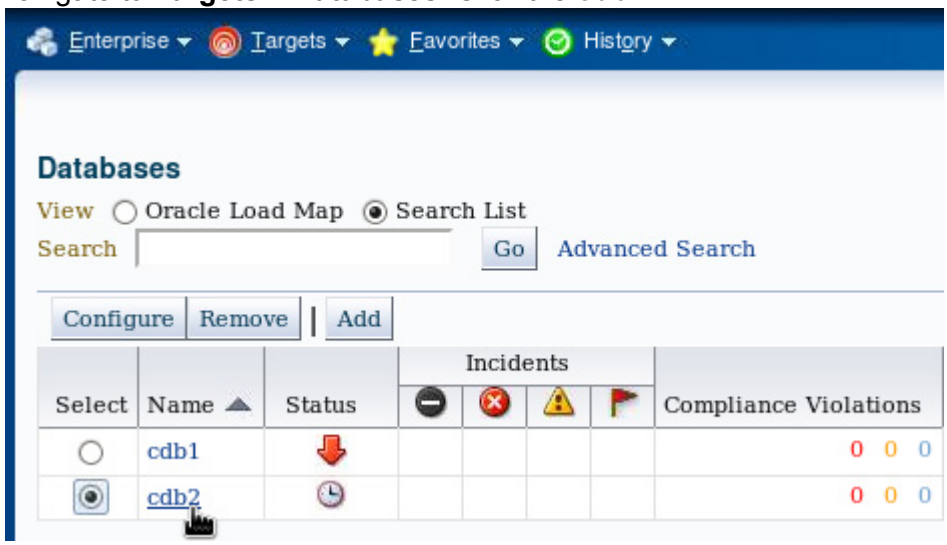
1. In Enterprise Manager Cloud Control, use the same procedure as in Practice 1-2 to add `cdb2` as a new target, managed by Enterprise Manager.
2. In Enterprise Manager Cloud Control, use the same procedure as in Practice 1-3 to create the Credential Name `CREDCDB2`.
3. In Enterprise Manager Cloud Control, create the Credential Name `MYHOST` as follows:
   a. Navigate to Setup > Security > Named Credentials.
   b. Click on **Create**.

   Enter the following values, then complete the **Access Control** section:

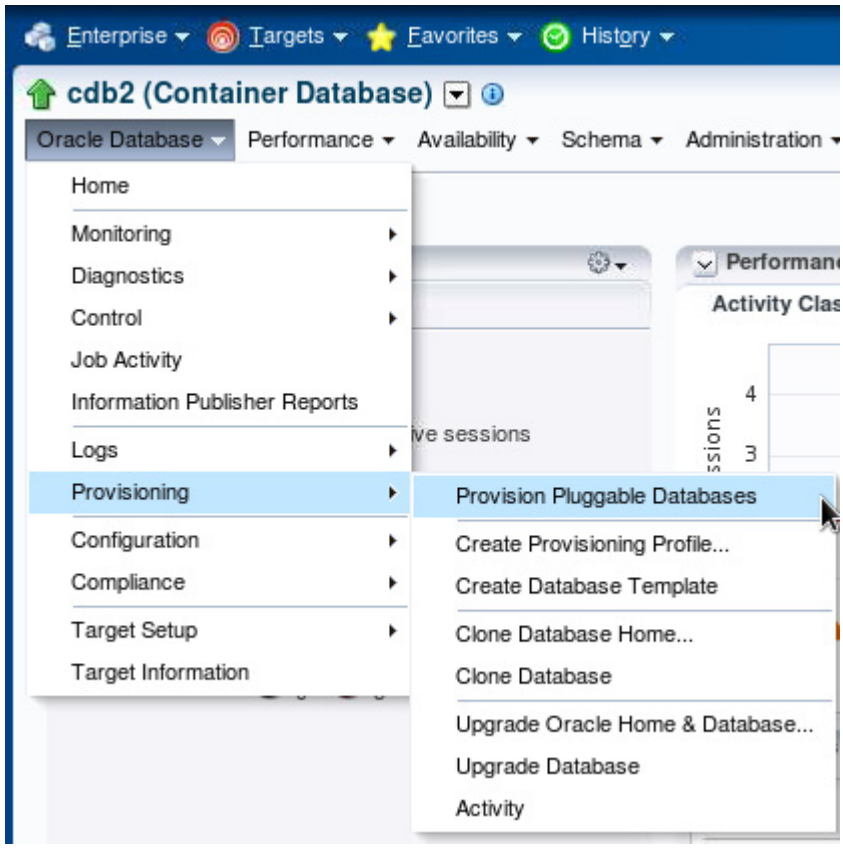
Field	Choice or Value
General Properties	
Credential Name	**myhost**
Credential description	**Credentials for Database**
Authenticating Target Type	**Host**
Credential type	**Host Credentials**
Scope	**Target**
Target type	**Host**
Target Name	***Your_server*** **(Click the magnifying glass to find *Your_server* and select)**
Credential Properties	
Username	**oracle**
Password	**oracle**
Confirm Password	**oracle**

   c. Click "**Test and Save**" until you get the following message: "**Confirmation Credential Operation Successful**." This means that the credential was successful and saved.
4. Unplug `pdb2_2`.

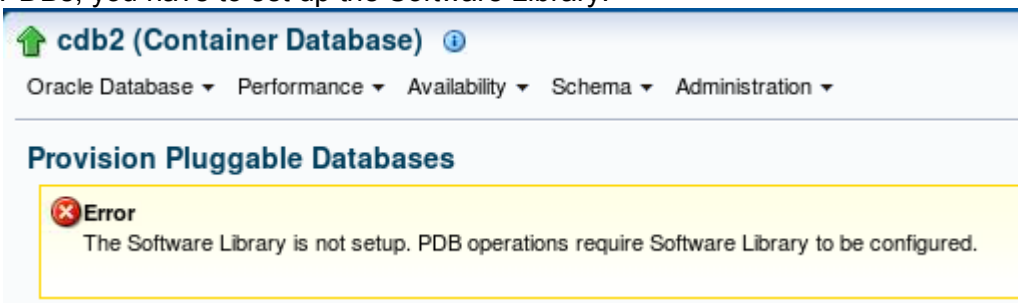
a. Relog in Enterprise Manager Cloud Control if you were automatically logged out. Then navigate to **Targets** > **Databases**. Click the `cdb2` link.



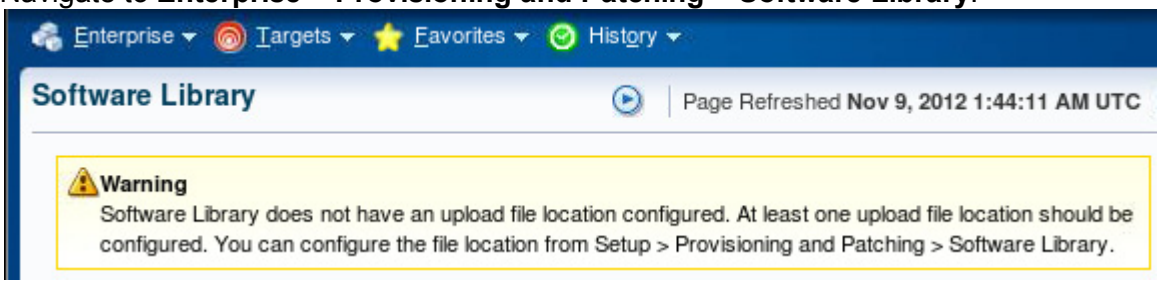
b. Then navigate to **Oracle Databases** > **Provisioning** > **Provision Pluggable Databases**.



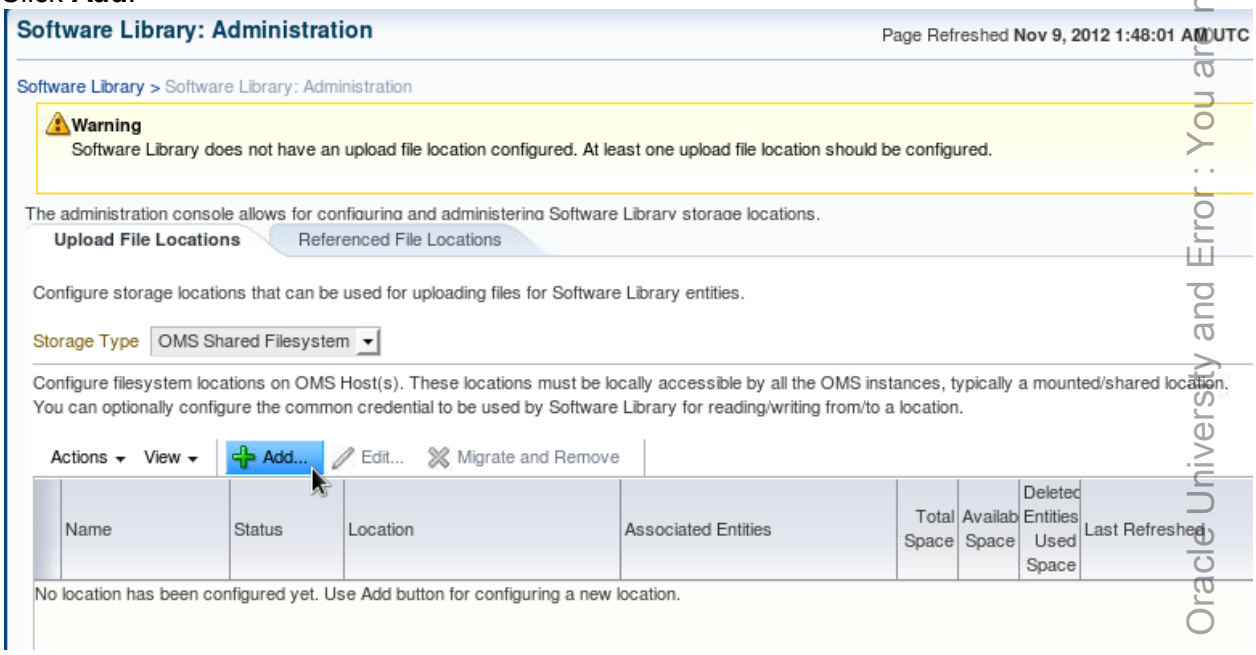
c. As it is the first time you are using Enterprise Manager Cloud Control to provision PDBs, you have to set up the Software Library.



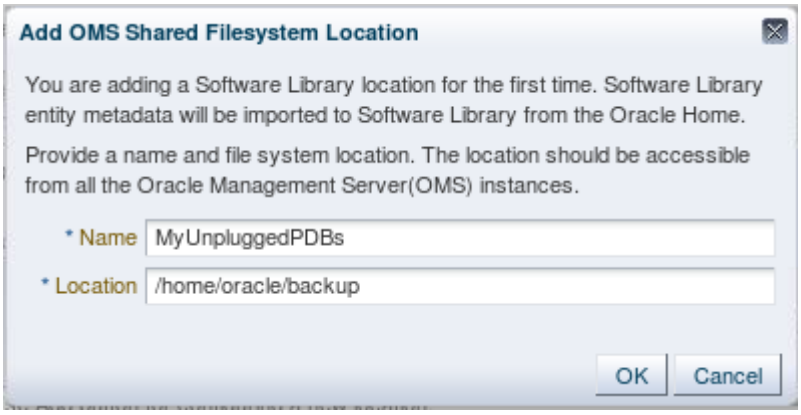
1) Navigate to **Enterprise** > **Provisioning and Patching** > **Software Library**.



2) Navigate to **Setup** > **Provisioning and Patching** > **Software Library**. Click **Add**.

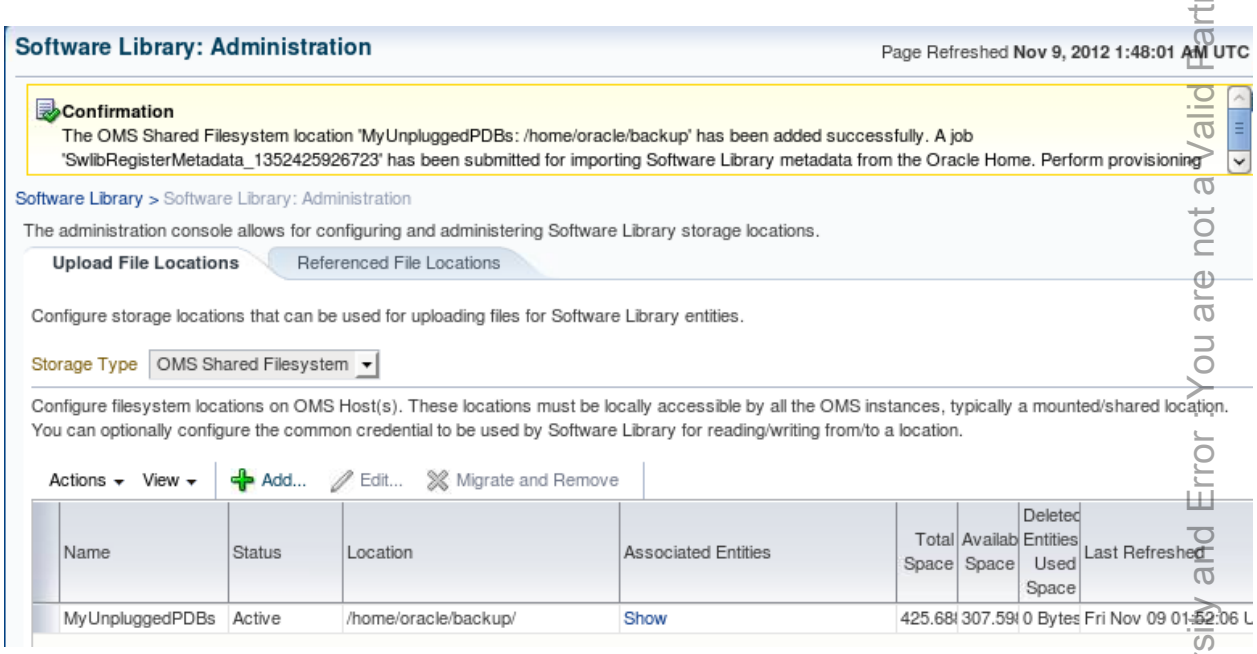


3) Enter the following values to complete the **Add OMS Shared Filesystem Location**.

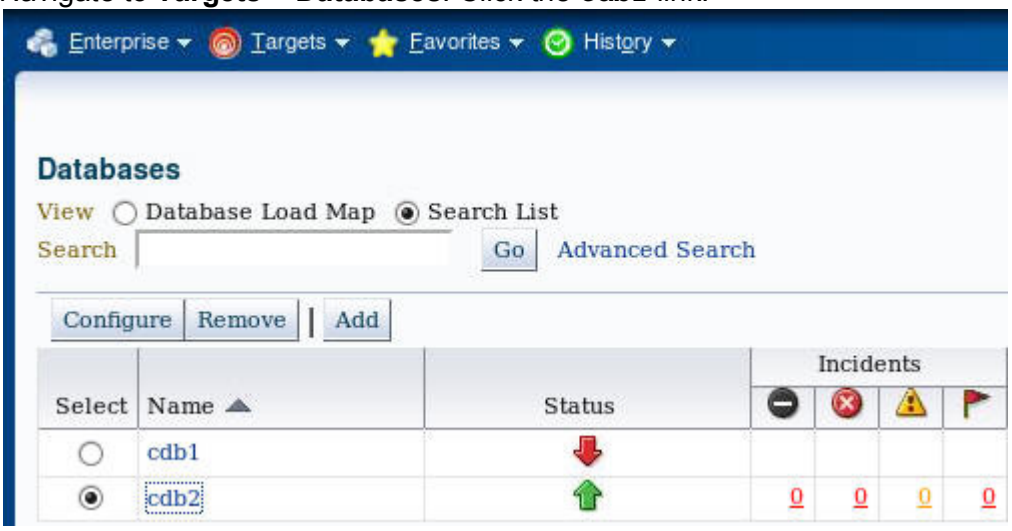


Click **OK**.

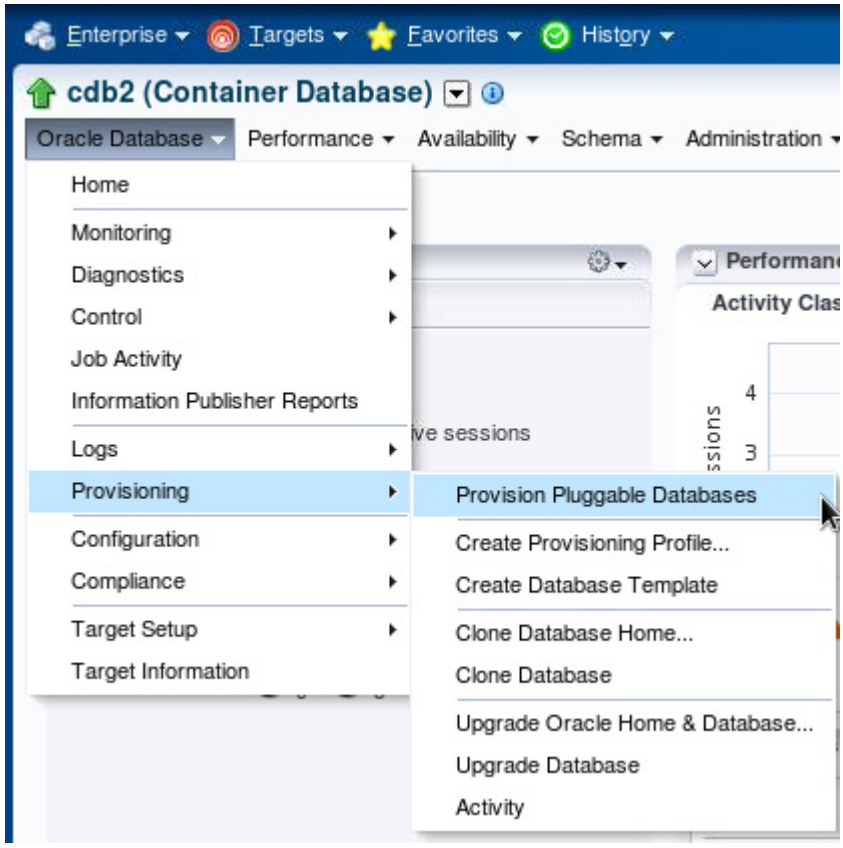
4) You get the **Confirmation** message:



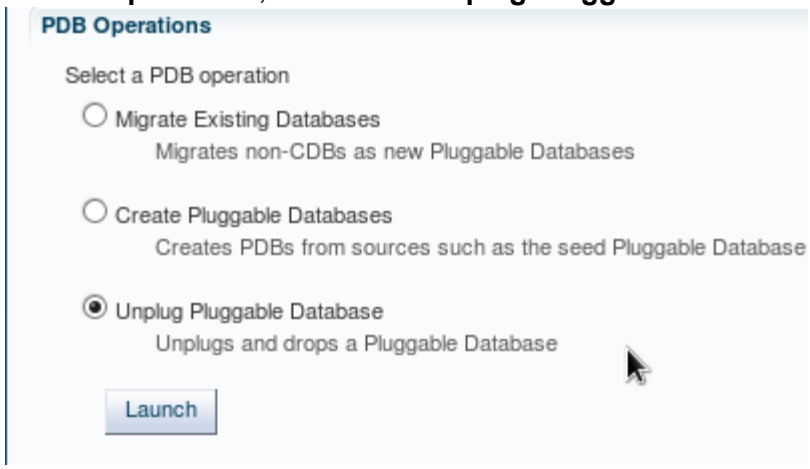
d. Navigate to **Targets** > **Databases**. Click the `cdb2` link.



e. Then navigate to **Oracle Databases** > **Provisioning** > **Provision Pluggable Databases**.



f. In **PDB Operations**, choose the **Unplug Pluggable Database** option. Click **Launch**.

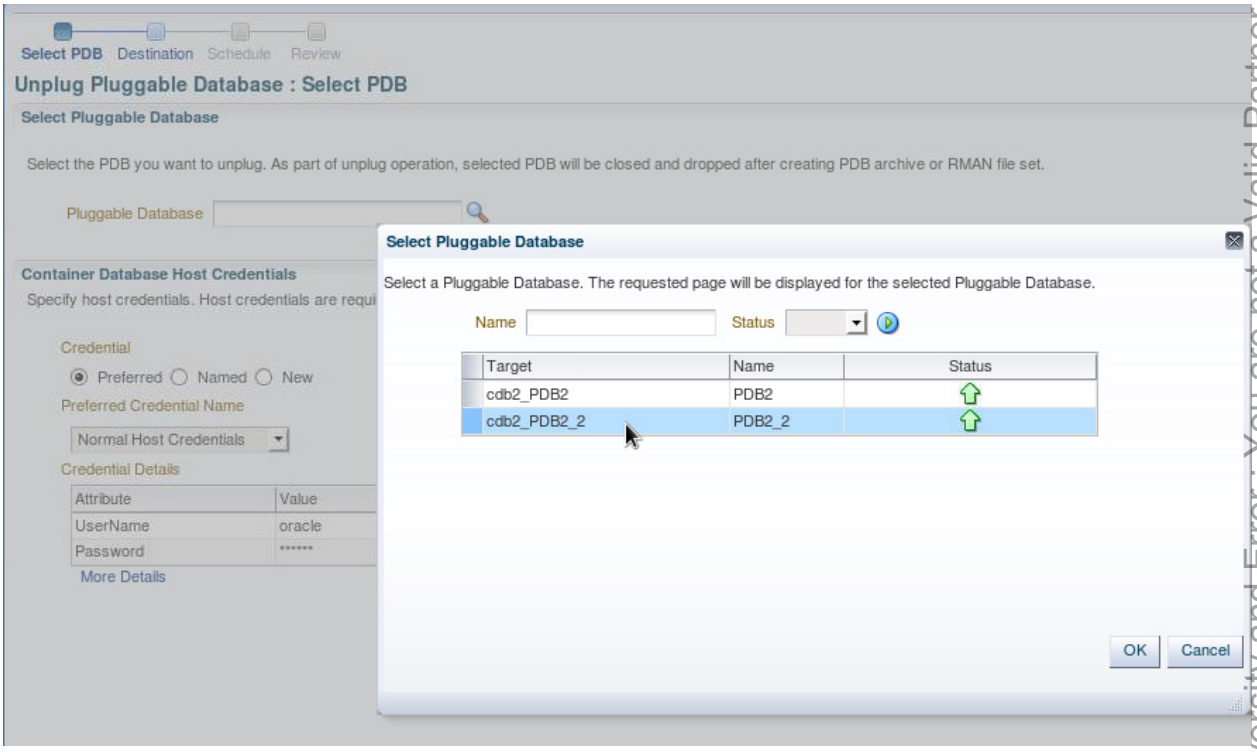


g. The **Database Login** is displayed with the named credential name `CREDCDB2`. Click **Login**.

h. Enter the following values to complete the **Unplug Pluggable Database : Select PDB** section:

Field	Choice or Value
Select Pluggable Database	

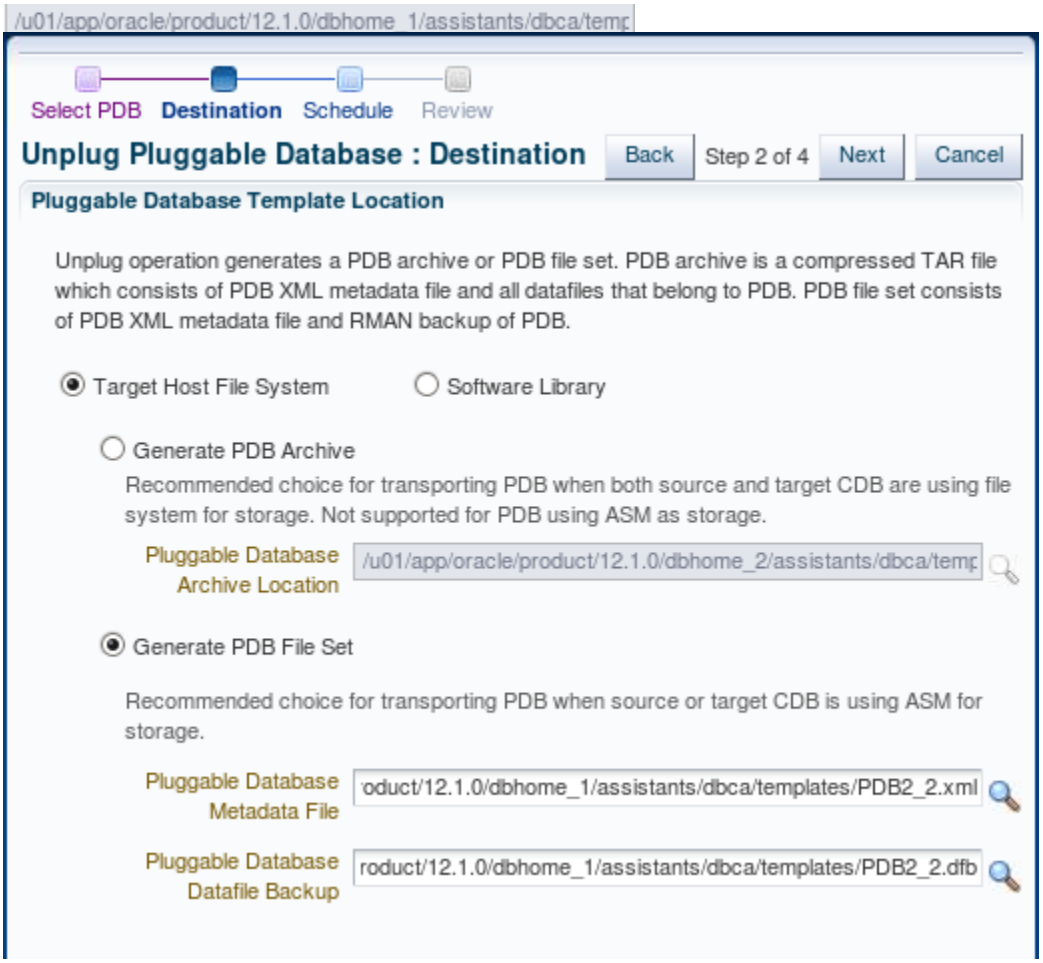
Field	Choice or Value
Pluggable Database	`PDB2_2`
Container Database Host Credentials	
Credential	`Named`
Credential Name	`MYHOST`
UserName	`oracle (pre-filled)`
Password	`****** (pre-filled)`



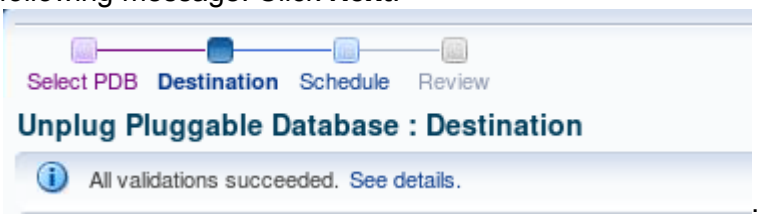
Click **Next**.

i. Enter the following values to complete the **Unplug Pluggable Database : Destination** section:

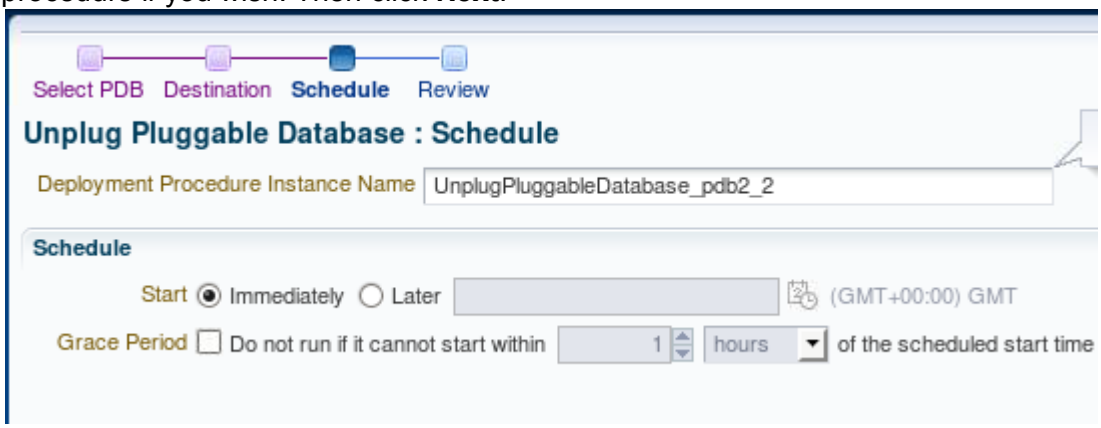
Field	Choice or Value
PDB Template Location	
Target Host File System	`Use this radio button`
Generate PDB File Set	`Use this radio button`
Pluggable Database Metadata File	`/u01/app/oracle/product/12.1.0/dbhome_1/assistants/dbca/templates/PDB2_2.xml`
Pluggable Database Datafile Backup	`/u01/app/oracle/product/12.1.0/dbhome_1/assistants/dbca/templates/PDB2_2.dfb`



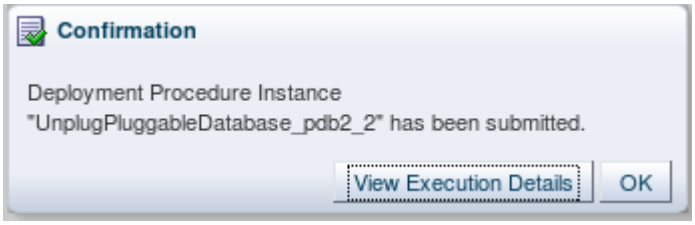
Click **Next**. Wait until the Validation in Progress page completes and sends back the following message. Click **Next**.



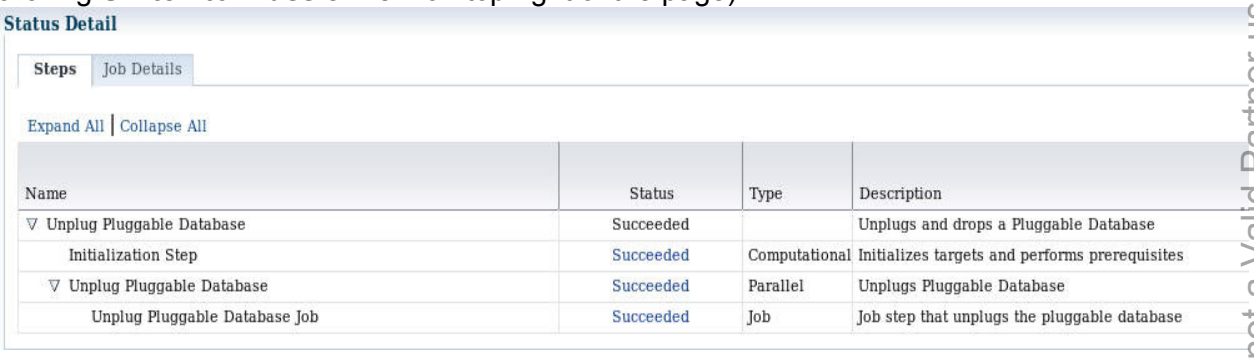
j. Schedule the unplug operation. You can change the name of the deployment procedure if you wish. Then click **Next**.



k. The Review page is displayed. Then click **Submit**. You get the **Confirmation** Message.



If you click the **View Execution Details**, you will see that the operation unplugs and drops the PDB. Refresh until you get **Succeeded** Status. (The format below is got after clicking **Switch to Classic View** on top right of the page)

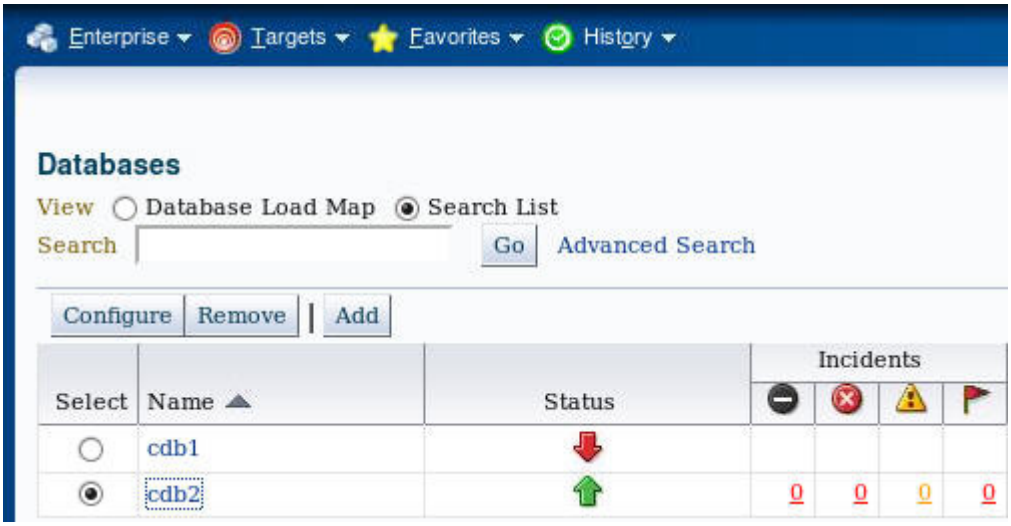


5. Now replug the `pdb2_2` into the same CDB using the backup set and an XML file created in task 1.

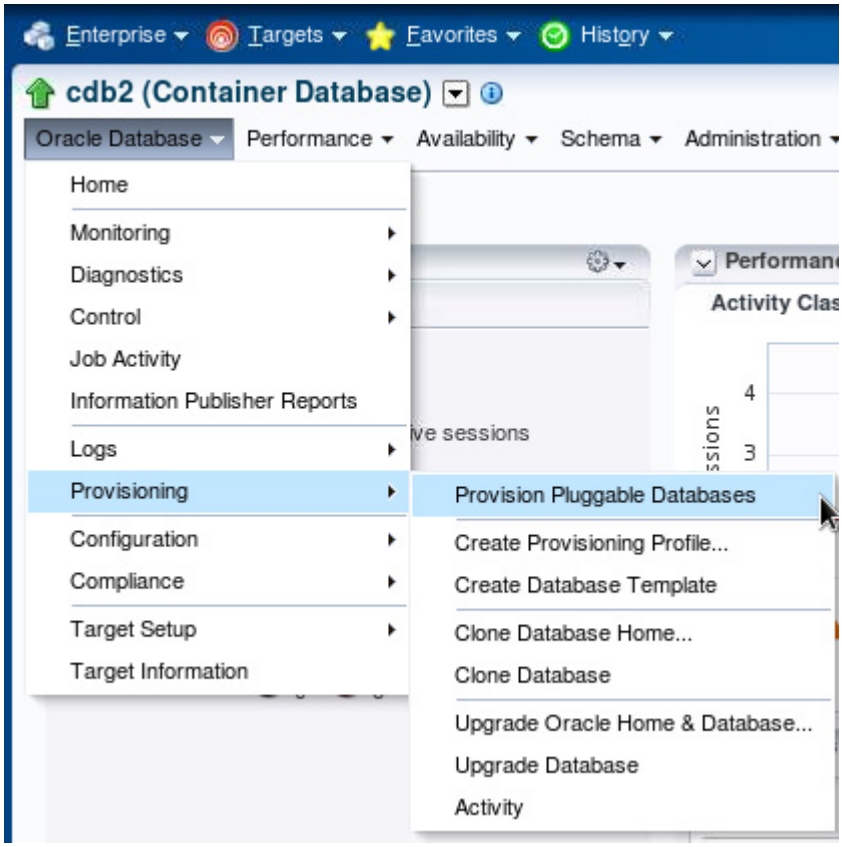
a. In the next step, you use the OMF to locate the data files of the plugged PDB. You have to set the value for the OMF destination. If the destination directory does not exist yet, create it.

```
$ . oraenv
ORACLE_SID = [cdb2] ? cdb2
The Oracle base for
ORACLE_HOME=/u01/app/oracle/product/12.1.0/dbhome_1 is
/u01/app/oracle
$ mkdir /u01/app/oracle/oradata/cdb2/pdb2_2
$ sqlplus / as sysdba
Connected to:
Oracle Database 12c Enterprise Edition Release 12.1.0.1.0 -
64bit Production
With the Partitioning, OLAP, Advanced Analytics and Real
Application Testing options


SQL> ALTER SYSTEM SET
db_create_file_dest='/u01/app/oracle/oradata/cdb2/pdb2_2';


System altered.


SQL>
```

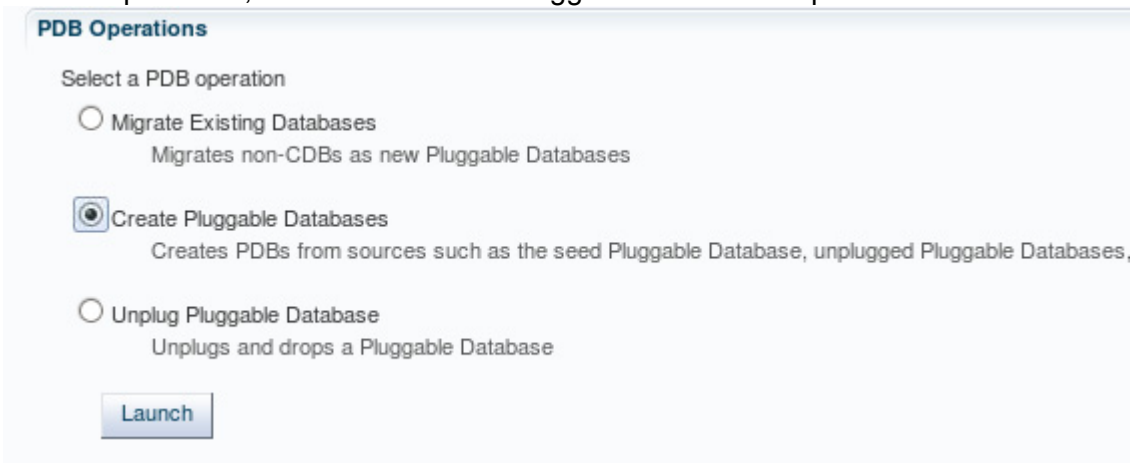
b. Navigate to **Targets** > **Databases**. Click the `cdb2` link.



c. Navigate to **Oracle Databases** > **Provisioning** > **Provision Pluggable Databases**.

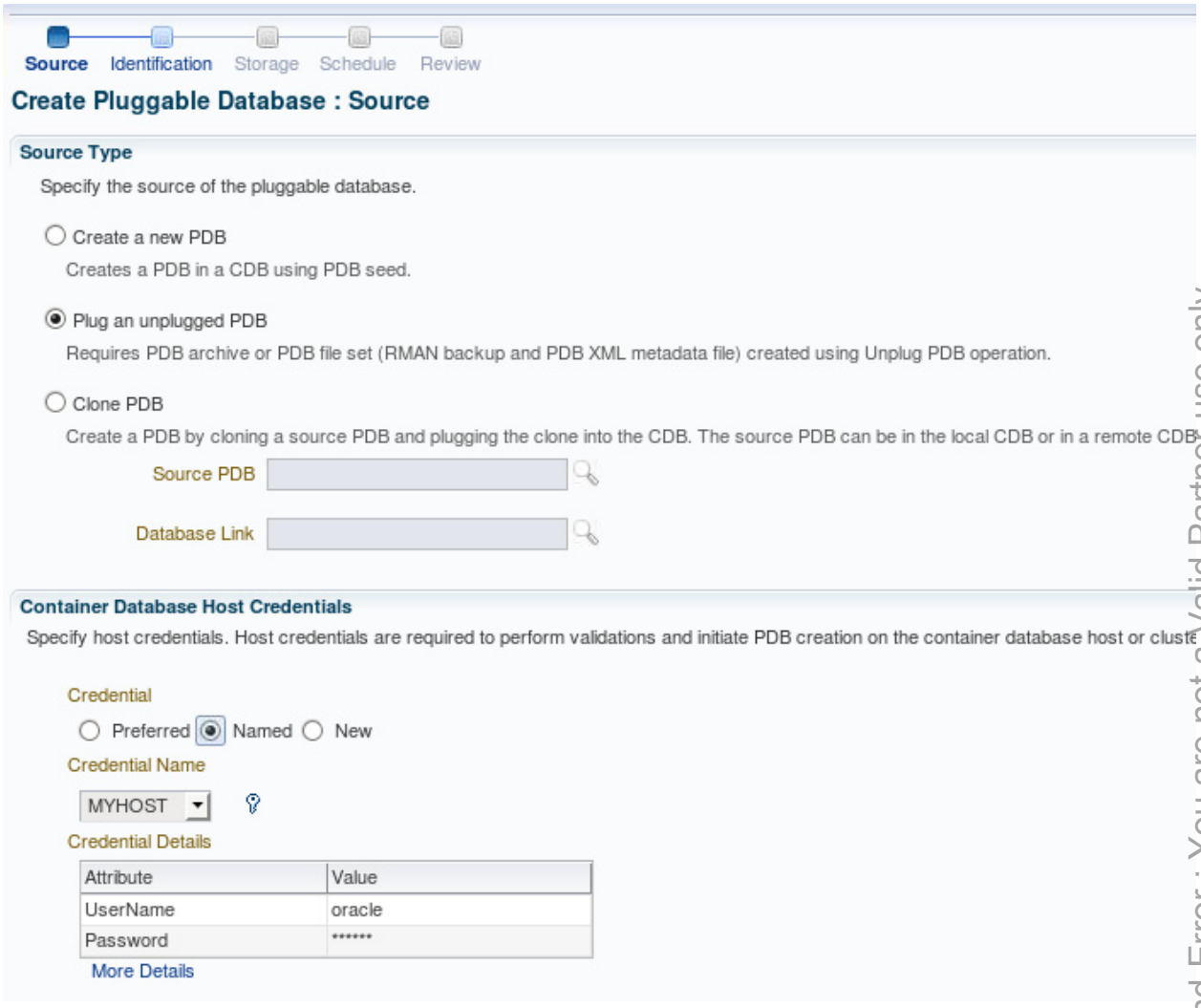


d. In PDB Operations, choose the Create Pluggable Databases option. Click Launch.



e. Enter the following values to complete the **Create Pluggable Database : Source** section:

Field	Choice or Value
Source Type	
Plug an unplugged PDB	**`Use this radio button`**
Container Database Host Credentials	
Credential	**`Named`**
Credential Name	**`MYHOST`** `(automatically appears)`
UserName	**`oracle`** `(pre-filled)`
Password	**`******`** `(pre-filled)`

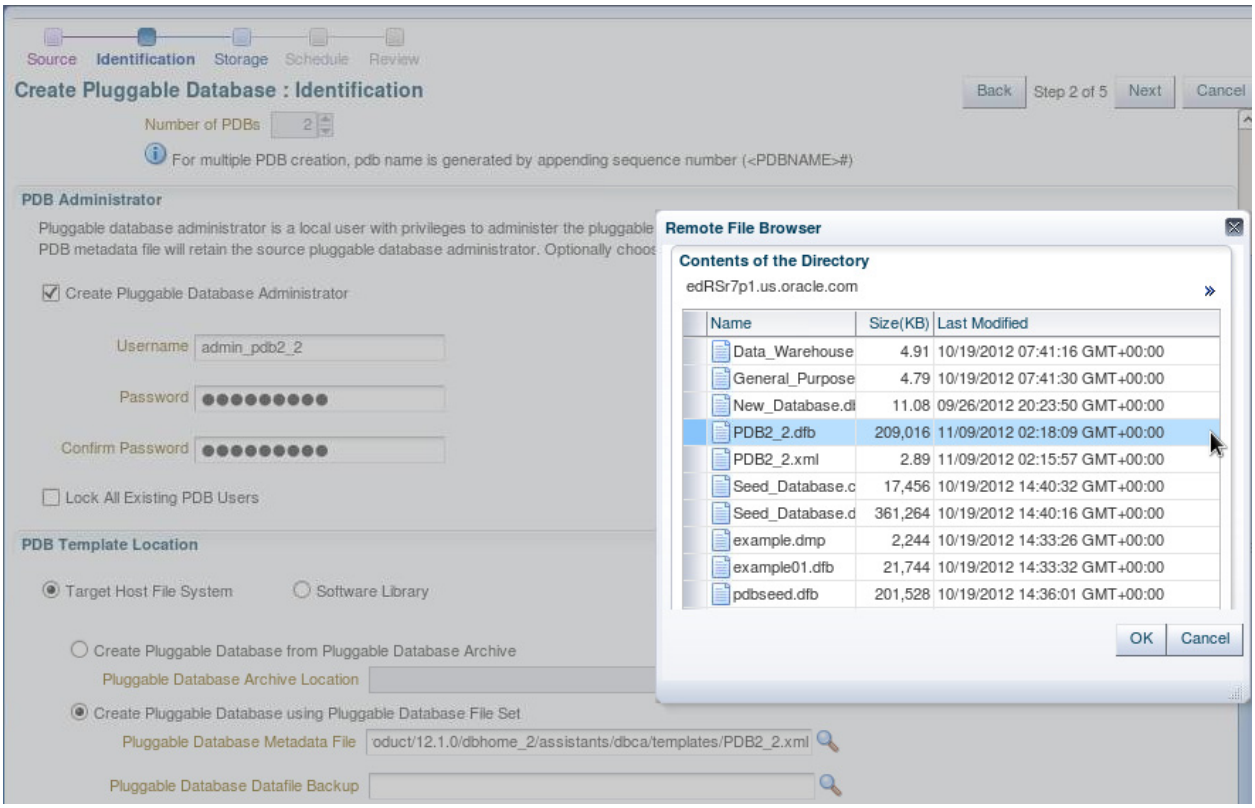


Click **Next**.

f. Enter the following values to complete the **Create Pluggable Database : Identification** section:

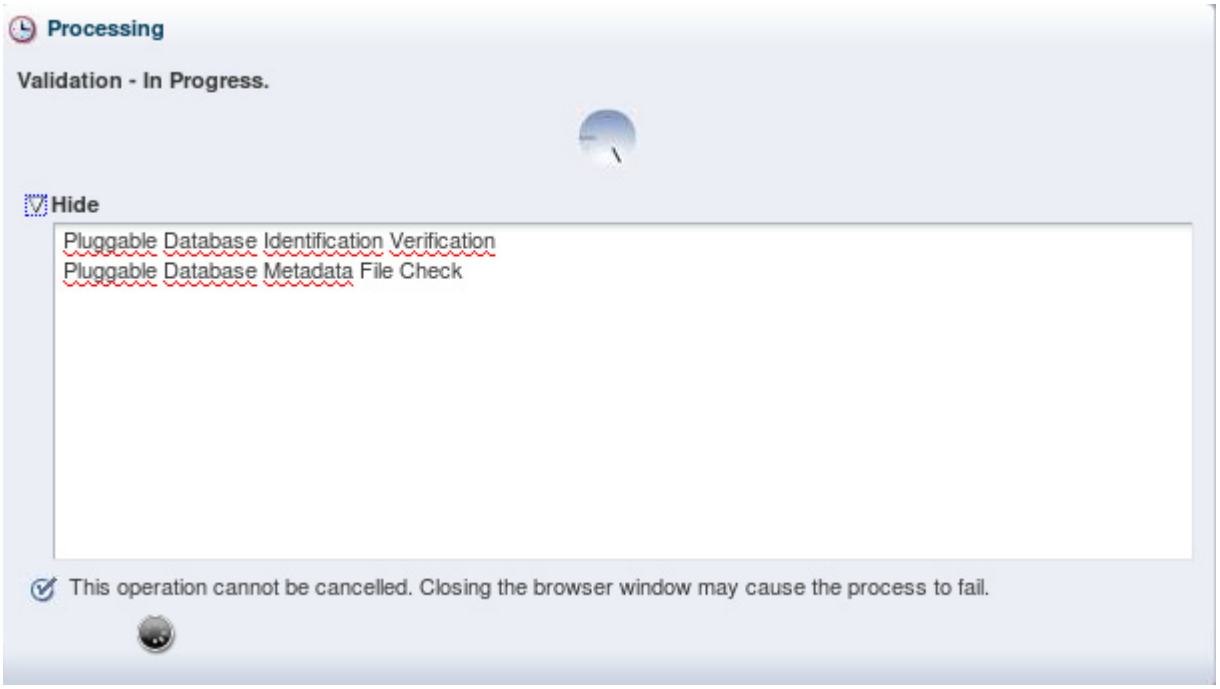
Field	Choice or Value
PDB Name	`PDB2_2`
PDB Administrator	
Create Pluggable Database Administrator	`Check this button`
Username	`admin_pdb2_2`
Password	`oracle_4U`
Confirm Password	`oracle_4U`
PDB Template Location	
Target Host File System	`Select this radio button`
Create Pluggable Database using Pluggable Database File	`Select this radio button`

Field	Choice or Value
Set	
Pluggable Database Metadata File	`/u01/app/oracle/product/12.1.0/dbhome_1/assistants/dbca/templates/PDB2_2.xml` and click **OK**
Pluggable Database Datafile Backup	`/u01/app/oracle/product/12.1.0/dbhome_1/assistants/dbca/templates/PDB2_2.dfb` and click **OK**



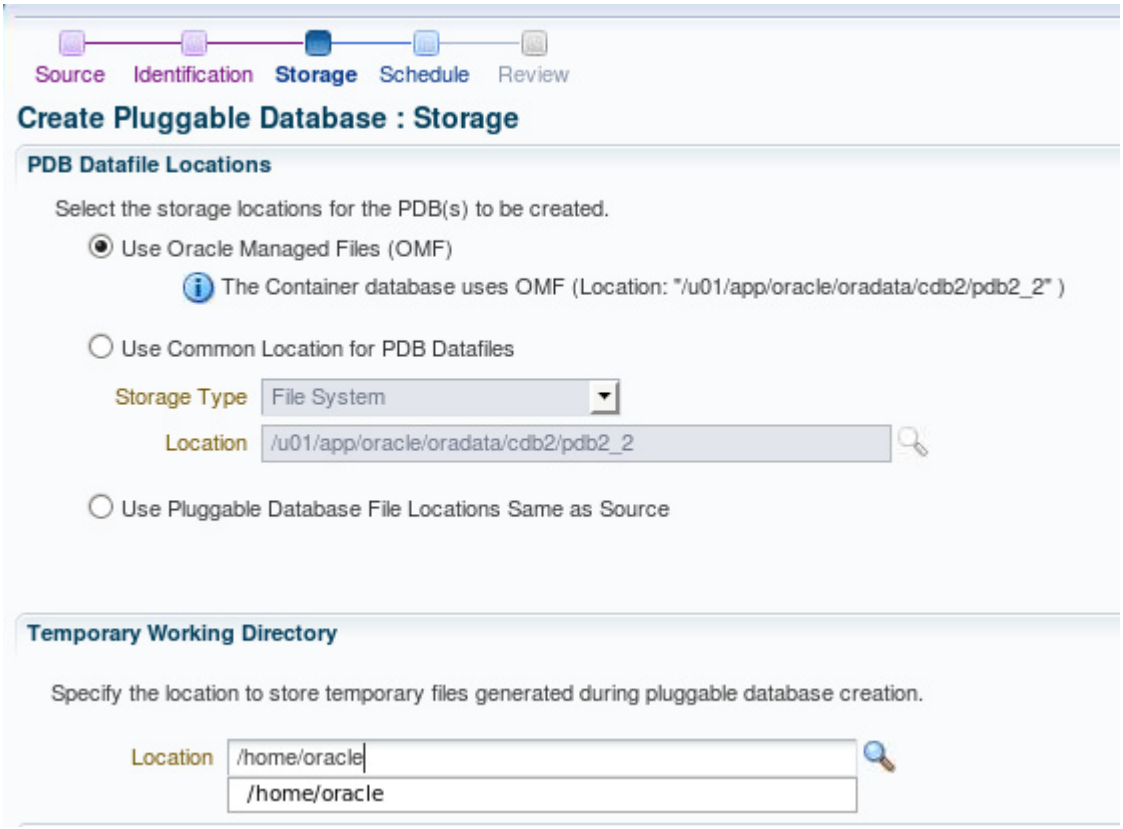
Click **Next**.

g. Before you can proceed with the next step, you get the following **Processing** page. Wait until the verification of the existence of both files selected.



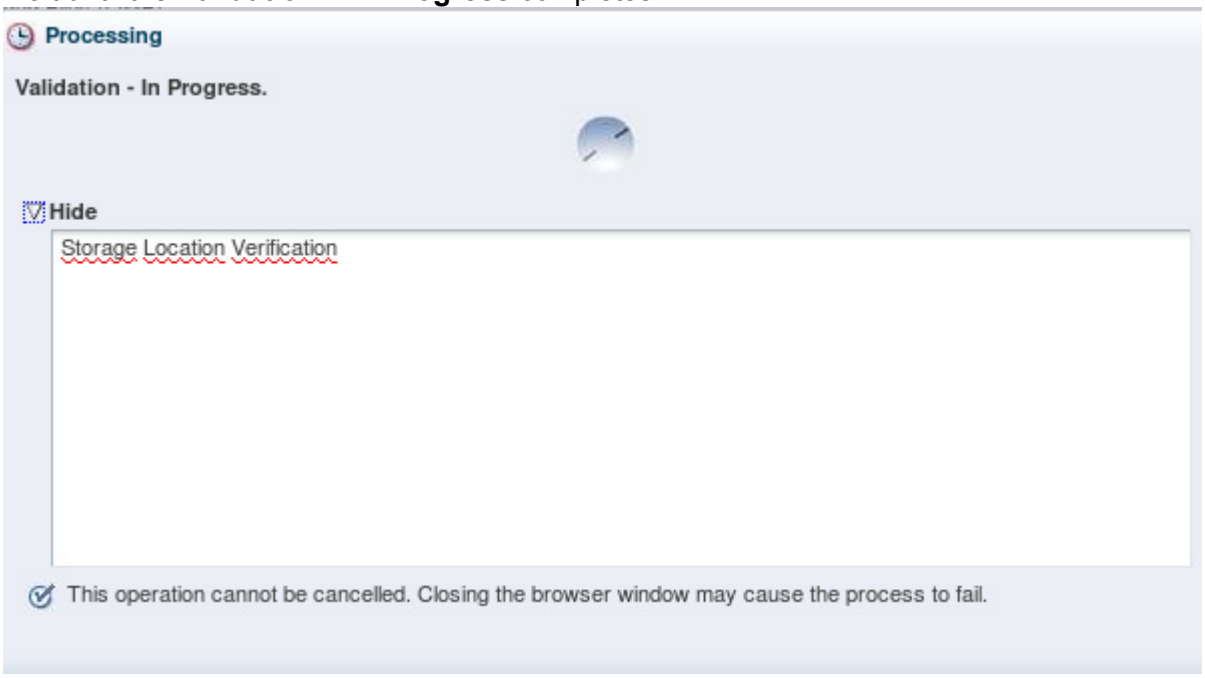
h. Enter the following values to complete the **Create Pluggable Database : Storage** section:

Field	Choice or Value
PDB Datafile Locations	
Use Oracle Managed files (OMF)	`Check this button`
Temporary Working Directory	
Location	`/home/oracle`



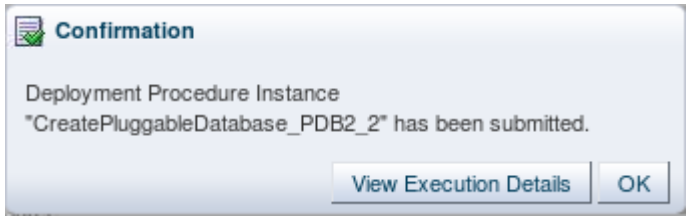
Click **Next**.

Wait until the **Validation – In Progress** completes.



i. Schedule the plug operation. You can change the name of the deployment procedure if you wish. Then click **Next**.

j. The Review page is displayed. Then click **Submit**. You get the following **Confirmation** message:



6. Verify that the `pdb2_2` is back into the CDB.

```
SQL> select name, open_mode  from v$pdbs;

NAME                           OPEN_MODE
------------------------------ ----------
PDB$SEED                        READ ONLY
PDB2                            READ WRITE
PDB2_2                          READ WRITE

SQL> EXIT
$
```

Practices for Lesson 8: Performance

Chapter 8

Practices for Lesson 8: Overview

Overview

In this practice, you create two CDB Resource Manager plans and associated directives to limit CPU resources used by two PDBs. And while checking the results on procedures execution, you will run an ADDM task to get recommendations.

Practice 8-1: Using CDB Resource Manager Plans and Directives

Overview

In this practice, you will create two CDB Resource Manager plans and associated directives to limit CPU resources used by two PDBs.

Assumptions

If `cdb2` could not be recovered during Practices 7 and is not available anymore, perform the following script, provided you successfully performed the tar backup at the beginning of practice 7.

```
$ cd /home/oracle/solutions/catchup_07_01
$ ./restore.sh
```

Tasks

1. Connect to the root of `cdb2` as `SYSDBA` and clean up your environment by executing the `rsrc_cleanup.sql` script. The script will close all PDBs except `PDB2` and `PDB2_2`.
   a. Make sure you are in the `~/labs/RM` directory and your environment points to the `cdb2` instance.

```
$ cd ~/labs/RM
$ . oraenv
ORACLE_SID = [cdb1] ? cdb2
The Oracle base remains unchanged with value /u01/app/oracle
$
```

   b. Execute the `rsrc_cleanup.sql` script.
      1) Start up the database instance if not already done.

```
$ sqlplus / as sysdba

Connected to:
Oracle Database 12c Enterprise Edition Release 12.1.0.1.0 -
64bit Production
With the Partitioning, OLAP, Advanced Analytics and Real
Application Testing options

SQL> @rsrc_cleanup.sql



Pluggable database altered.



Pluggable database altered.



NAME                                     CON_ID OPEN_MODE
```

```
------------------------------ ---------- ----------
PDB$SEED                                2 READ ONLY
PDB2                                    3 READ WRITE
PDB2_2                                  4 READ WRITE


System altered.


NAME
--------------------------------
ORA$INTERNAL_CDB_PLAN


System altered.


PL/SQL procedure successfully completed.


PL/SQL procedure successfully completed.

BEGIN
DBMS_Resource_Manager.Delete_CDB_Plan_Directive('fairplan',
'pdb2'); END;

*
ERROR at line 1:
ORA-29358: resource plan FAIRPLAN does not exist
ORA-06512: at "SYS.DBMS_RMIN_SYS", line 3176
ORA-06512: at "SYS.DBMS_RESOURCE_MANAGER", line 1605
ORA-06512: at line 1


BEGIN
DBMS_Resource_Manager.Delete_CDB_Plan_Directive('fairplan',
'pdb2_2'); END;

*
ERROR at line 1:
ORA-29358: resource plan FAIRPLAN does not exist
ORA-06512: at "SYS.DBMS_RMIN_SYS", line 3176
ORA-06512: at "SYS.DBMS_RESOURCE_MANAGER", line 1605
```

```
ORA-06512: at line 1


BEGIN DBMS_Resource_Manager.Delete_CDB_Plan('fairplan'); END;

*
ERROR at line 1:
ORA-29358: resource plan FAIRPLAN does not exist
ORA-06512: at "SYS.DBMS_RMIN_SYS", line 2871
ORA-06512: at "SYS.DBMS_RESOURCE_MANAGER", line 1451
ORA-06512: at line 1


BEGIN
DBMS_Resource_Manager.Delete_CDB_Plan_Directive('unfairplan',
'pdb2'); END;

*
ERROR at line 1:
ORA-29358: resource plan UNFAIRPLAN does not exist
ORA-06512: at "SYS.DBMS_RMIN_SYS", line 3176
ORA-06512: at "SYS.DBMS_RESOURCE_MANAGER", line 1605
ORA-06512: at line 1


BEGIN
DBMS_Resource_Manager.Delete_CDB_Plan_Directive('unfairplan',
'pdb2_2'); END;

*
ERROR at line 1:
ORA-29358: resource plan UNFAIRPLAN does not exist
ORA-06512: at "SYS.DBMS_RMIN_SYS", line 3176
ORA-06512: at "SYS.DBMS_RESOURCE_MANAGER", line 1605
ORA-06512: at line 1


BEGIN DBMS_Resource_Manager.Delete_CDB_Plan('unfairplan'); END;

*
ERROR at line 1:
ORA-29358: resource plan UNFAIRPLAN does not exist
ORA-06512: at "SYS.DBMS_RMIN_SYS", line 2871
```

```
ORA-06512: at "SYS.DBMS_RESOURCE_MANAGER", line 1451
ORA-06512: at line 1



PL/SQL procedure successfully completed.


PL/SQL procedure successfully completed.

SQL> EXIT
$
```

2. Open a terminal window (it will be referred to as window1) to connect to `pdb2` in `cdb2` and create a PL/SQL procedure that burns CPU in `PDB2` as the `SYSTEM` user. You can use the `create_burn_cpu.sql` script to create the procedure after connecting to `PDB2`.

```
$ cd ~/labs/RM
$ . oraenv
ORACLE_SID = [oracle] ? cdb2
The Oracle base remains unchanged with value /u01/app/oracle
$
$ sqlplus system/oracle_4U@pdb2

Connected to:
Oracle Database 12c Enterprise Edition Release 12.1.0.1.0 -
64bit Production
With the Partitioning, OLAP, Advanced Analytics and Real
Application Testing options

SQL> @create_burn_cpu.sql

Procedure created.

SQL>
```

3. Open a second terminal window (it will be referred to as window2) to connect to `pdb2_2` in `cdb2` and create a PL/SQL procedure that burns CPU in `PDB2_2` as the `SYSTEM` user. You can use the `create_burn_cpu.sql` script to create the procedure after connecting to `PDB2_2`.

```
$ cd ~/labs/RM
$ . oraenv
ORACLE_SID = [oracle] ? cdb2
The Oracle base remains unchanged with value /u01/app/oracle
$
$ sqlplus system/oracle_4U@pdb2_2
```

```
Connected to:
Oracle Database 12c Enterprise Edition Release 12.1.0.1.0 -
64bit Production
With the Partitioning, OLAP, Advanced Analytics and Real
Application Testing options

SQL> @create_burn_cpu.sql

Procedure created.

SQL>
```

4. From window1, create two new CDB plans called FAIRPLAN and UNFAIRPLAN. FAIRPLAN should give one share to both PDB2 and PDB2_2, and UNFAIRPLAN should give one share to PDB2 and five shares to PDB2_2.

```
SQL> alter session set container = CDB$Root;

Session altered.

SQL> EXEC DBMS_Resource_Manager.Clear_Pending_Area();

PL/SQL procedure successfully completed.

SQL> EXEC DBMS_Resource_Manager.Create_Pending_Area();

PL/SQL procedure successfully completed.

SQL> EXEC DBMS_Resource_Manager.Create_CDB_Plan('fairplan', 'One
share each');

PL/SQL procedure successfully completed.

SQL> EXEC
DBMS_Resource_Manager.Create_CDB_Plan_Directive('fairplan',
'pdb2', shares => 1);

PL/SQL procedure successfully completed.

SQL> EXEC
DBMS_Resource_Manager.Create_CDB_Plan_Directive('fairplan',
'pdb2_2', shares => 1);

PL/SQL procedure successfully completed.
```

```
SQL> EXEC DBMS_Resource_Manager.Create_CDB_Plan('unfairplan',
'one share to pdb2 and five to pdb2_2');

PL/SQL procedure successfully completed.

SQL> EXEC
DBMS_Resource_Manager.Create_CDB_Plan_Directive('unfairplan',
'pdb2', shares => 1);

PL/SQL procedure successfully completed.

SQL> EXEC
DBMS_Resource_Manager.Create_CDB_Plan_Directive('unfairplan',
'pdb2_2', shares => 5);

PL/SQL procedure successfully completed.

SQL> EXEC DBMS_Resource_Manager.Validate_Pending_Area();

PL/SQL procedure successfully completed.

SQL> EXEC DBMS_Resource_Manager.Submit_Pending_Area();

PL/SQL procedure successfully completed.

SQL>
```

5. Still from window1, make sure both plans and associated directives were created correctly.

```
SQL> SELECT Plan from CDB_CDB_Rsrc_Plans
  2  WHERE Con_ID = 1 AND Plan IN ('FAIRPLAN', 'UNFAIRPLAN')
  3  ORDER BY 1;

PLAN
--------------------------------------------------------------------
FAIRPLAN
UNFAIRPLAN

SQL> select Plan, Pluggable_Database, Shares
from CDB_CDB_Rsrc_Plan_Directives
where Con_ID = 1
and Plan in ('FAIRPLAN', 'UNFAIRPLAN')
and Pluggable_Database in ('PDB2', 'PDB2_2')
order by 1, 2;
  2    3    4    5    6
```

```
PLAN
------------------------------------------------------------------
----------------
PLUGGABLE_DATABASE
------------------------------------------------------------------
----------------
    SHARES
----------
FAIRPLAN
PDB2
         1

FAIRPLAN
PDB2_2
         1



UNFAIRPLAN
PDB2
         1

UNFAIRPLAN
PDB2_2

         5

SQL>
```

6. From window1, activate the CDB plan `FAIRPLAN`.

```
SQL> CONNECT / AS SYSDBA
Connected.
SQL> alter system set resource_manager_plan = fairplan;

System altered.

SQL> select Name from v$Rsrc_Plan where Con_ID = 1;

NAME
--------------------------------
FAIRPLAN

SQL>
```

7. From window1, connect as the SYSTEM user in PDB2 and set SERVEROUTPUT variable to ON.

```
SQL> CONNECT system/oracle_4U@pdb2
Connected.
SQL> set serveroutput on
SQL>
```

8. From window2, connect as the SYSTEM user in PDB2_2 and set SERVEROUTPUT variable to ON.

```
SQL> CONNECT system/oracle_4U@pdb2_2
Connected.
SQL> set serveroutput on
SQL>
```

9. **DO NOT WAIT AND GO TO STEP 10 RIGHT AFTER:** From window1, execute the CPU burner procedure you created at step 2.

```
SQL> EXEC Burn_CPU_For_RM_Demo();
CPU: 94.0 Wall:  218.2 k: 2000000000

PL/SQL procedure successfully completed.

SQL>
```

10. From window2, execute the CPU burner procedure you created at step 3.

```
SQL> EXEC Burn_CPU_For_RM_Demo();
CPU: 98.4 Wall:  217.1 k: 2000000000

PL/SQL procedure successfully completed.

SQL>
```

11. What do you observe?
    Both procedures finish their execution almost at the same time, and have both consumed almost the same CPU and wall-clock time during their execution.
    This is expected because each PDB is receiving one share of CPU.
12. From window1, connect as user SYS in the root, and change the Resource Manager plan to UNFAIRPLAN.

```
SQL> CONNECT / AS SYSDBA
Connected.
SQL>
SQL> alter system set resource_manager_plan = unfairplan;

System altered.

SQL> select Name from v$Rsrc_Plan where Con_ID = 1;
```

```
NAME
--------------------------------
UNFAIRPLAN

SQL>
```

13. **Go to Step 14 right after starting the execution of the procedure in Step 13:** From window1, connect as user SYSTEM in PDB2 and execute the CPU burner procedure you created at step 2.

```
SQL> CONNECT system/oracle_4U@pdb2
Connected.
SQL>
SQL> set serveroutput on
SQL>
SQL> execute Burn_CPU_For_RM_Demo();
CPU:   101.1 Wall:  204.3 k: 2000000000

PL/SQL procedure successfully completed.

SQL>
```

14. From window2, execute the CPU burner procedure you created at step 3.

```
SQL> execute Burn_CPU_For_RM_Demo();
CPU:   94.9 Wall:  116.4 k: 2000000000

PL/SQL procedure successfully completed.

SQL> EXIT
Disconnected from Oracle Database 12c Enterprise Edition Release 12.1.0.1.0 - 64bit
Production With the Partitioning, OLAP, Advanced Analytics and Real Application
Testing options
$
```

15. What do you observe?
    Now, execution of the CPU burner procedure takes much longer to execute in PDB2 than in PDB2_2.
    This is expected because PDB2_2 is assigned five shares while PDB2 only one.
    However, the difference is not five times slower simply because once the procedure is executed in PDB2_2, all CPU cycles go to PDB2.
16. Make sure you set the CDB plan back to its default and open PDB2_2.

```
$ sqlplus / as sysdba

Connected to:
Oracle Database 12c Enterprise Edition Release 12.1.0.1.0 -
64bit Production
```

```
With the Partitioning, OLAP, Advanced Analytics and Real
Application Testing options

SQL> ALTER SYSTEM SET resource_manager_plan = '';

System altered.

SQL> SELECT name FROM v$rsrc_plan WHERE con_id = 1;

NAME
--------------------------------
DEFAULT_CDB_PLAN

SQL> ALTER PLUGGABLE DATABASE all open;

Pluggable database altered.

SQL> EXIT
$
```

Practice 8-2: Run ADDM and Get Recommendations

Overview

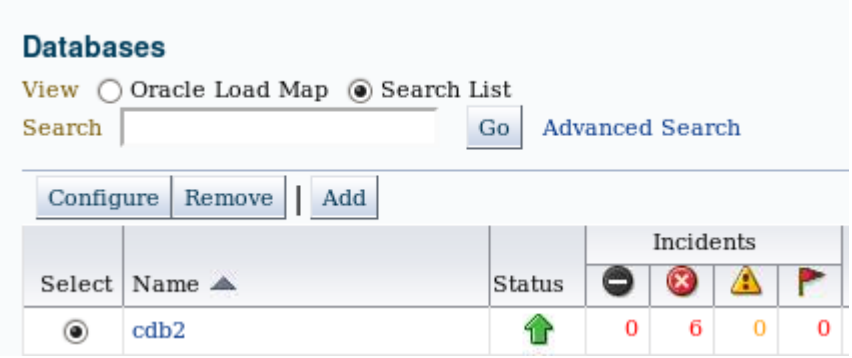
In this practice, you will run an ADDM task to get recommendations on the SQL statements previously executed in both `pdb2` and `pdb2_2`.

Assumptions

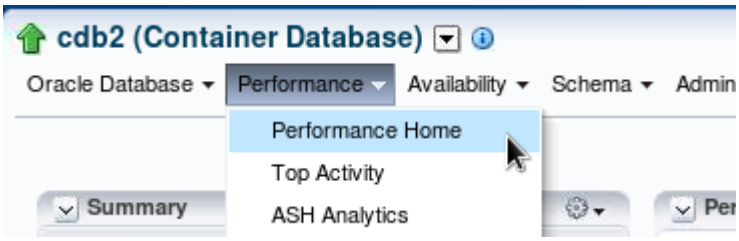
You completed Practice 8-1.

Tasks

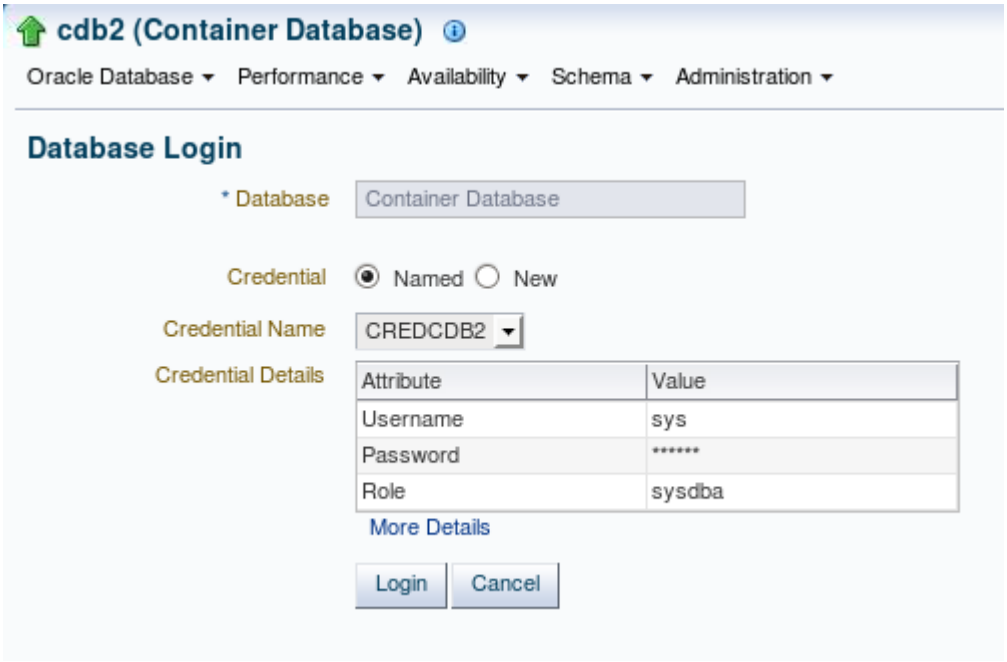
1. If you are no longer connected to Enterprise Manager Cloud Control, first log in. Enter **`sysman`** in the User Name field and **`Oracle123`** in the Password field. Then click **Login**.
2. Then navigate to **Targets**, then **Databases**, and click the `cdb2` link.



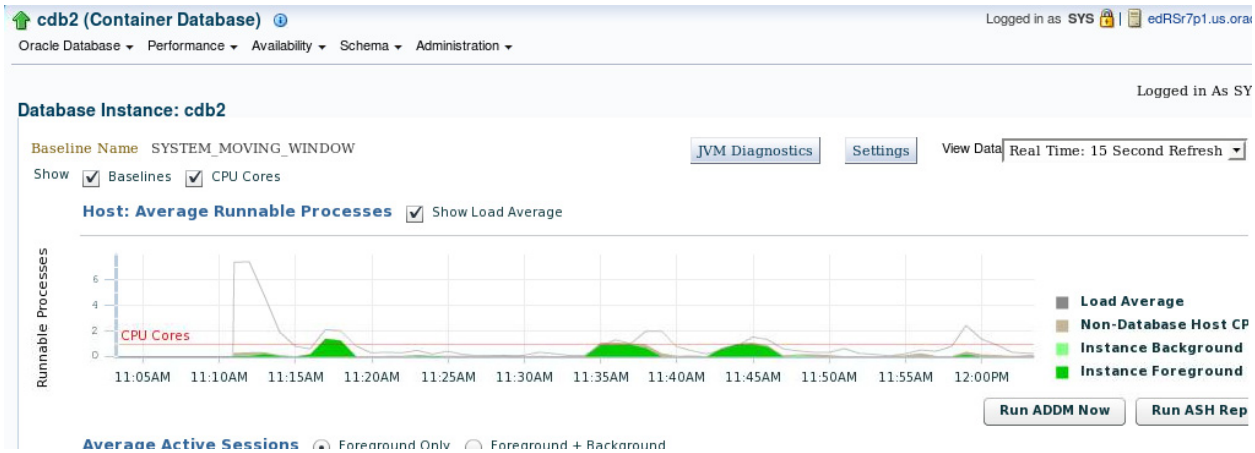
3. Then navigate to **Performance**, and click **Performance Home**.



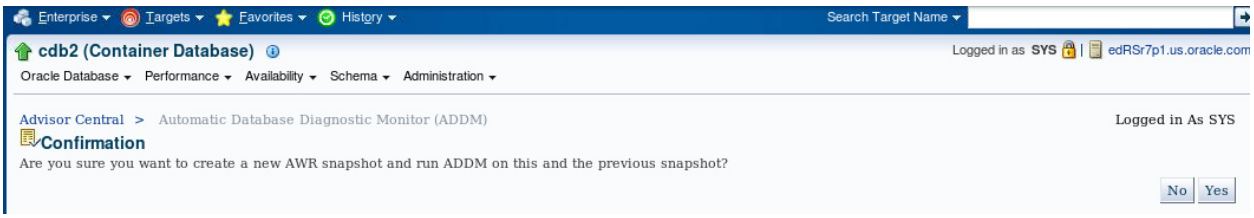
4. If you had to relog in to Enterprise Manager, you will have to click **Login** to accept the suggested credentials `CREDCDB2` to log in to `cdb2`.



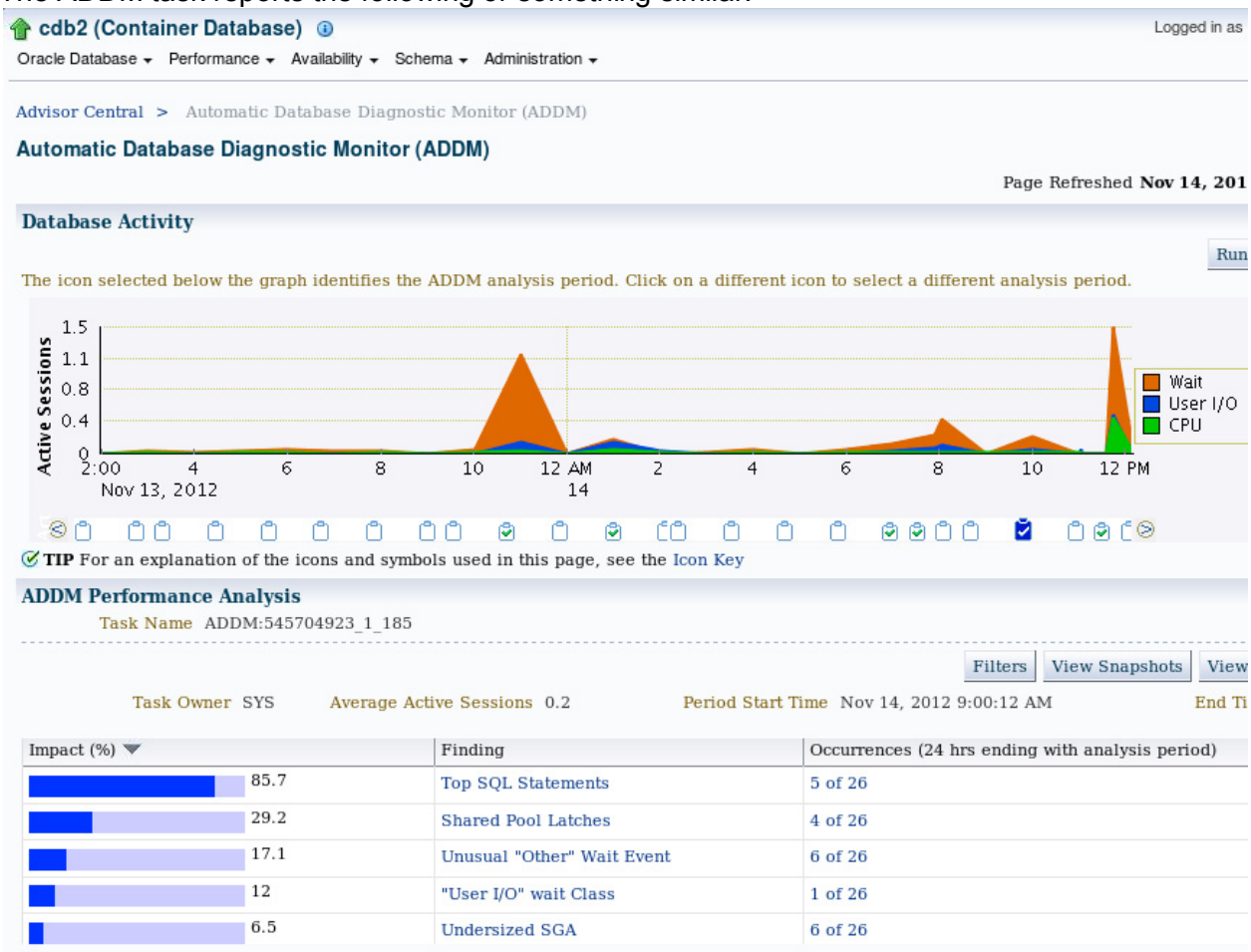
5. The **Database Instance: cdb2** page appears showing the **Runnable Processes**. Click the **Run ADDM Now** button to launch an ADDM task.



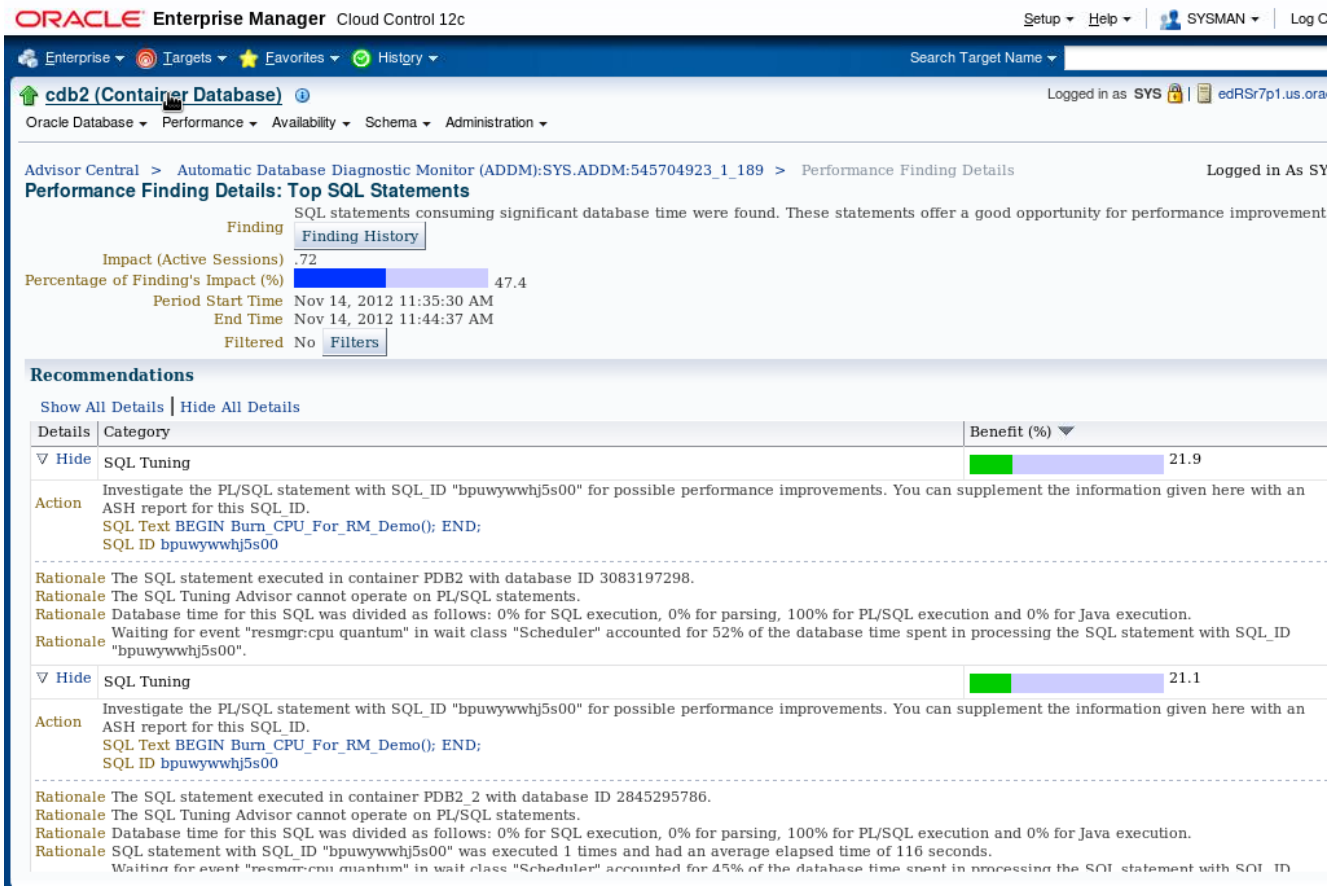
6. A Confirmation page appears. Click **Yes** to create a new AWR snapshot and run ADDM on this and the previous snapshots.



7. The ADDM task reports the following or something similar.



8. Click the **Top SQL Statements** to get detailed recommendations. Then in the **Details** column, click **Show** to view the recommended actions for each SQL statement that may require some action to perform better.



What do you observe? ADDM examined the statement that was executed in `pdb2`, `pdb2_2`, and in the `root`.

Practices for Lesson 9: Miscellaneous

Chapter 9

Practices for Lesson 9: Overview

Overview

In this practice, you will audit operations performed in PDBs, such as user creation and user drop operations performed in one PDB and create tablespace in another PDB using Unified Auditing.

Then you will perform Oracle Data Pump export and import operations between a non-CDB and a PDB, and between PDBs.

Practice 9-1: Auditing With Unified Audit

Overview

In this practice, you will configure an audit policy in the new `pdb_orcl` of `cdb2` auditing any future `CREATE TABLESPACE` statement.

You also create an audit policy in another PDB of `cdb2` auditing any `CREATE USER` or `DROP USER` operation executed by the local user `LU_PDB2` in `pdb2`.

Assumptions

cdb2 is successfully created from Practice 3-1.

pdb2_1 is successfully created from Practice 3-3.

pdb2_1 is successfully renamed to pdb2 from Practice 4-4.

If the trigger could not be created successfully, execute the following catchup script:

```
$ cd /home/oracle/solutions/catchup_04_03
$ ./cr_trig.sh
```

Tasks

1. Create a new `pdb_orcl` in `cdb2` that will be the container or recipient for the data exported from non-CDB `orcl`.
   a. Create a directory for the new data files of `pdb_orcl` of `cdb2`.

```
$ . oraenv
ORACLE_SID = [cdb2] ? cdb2
The Oracle base remains unchanged with value /u01/app/oracle

$ cd $ORACLE_BASE/oradata/cdb2
$ mkdir pdb_orcl
$
```

   b. Connect to the root with a user with `CREATE PLUGGABLE DATABASE` privilege.

```
$ sqlplus / as sysdba

Connected to:
Oracle Database 12c Enterprise Edition Release 12.1.0.1.0 -
64bit Production
With the Partitioning, OLAP, Advanced Analytics and Real
Application Testing options

SQL> CREATE PLUGGABLE DATABASE pdb_orcl ADMIN USER orcl_admin
  2  IDENTIFIED BY oracle_4U ROLES=(CONNECT)
  3  FILE_NAME_CONVERT=('/u01/app/oracle/oradata/cdb2/pdbseed'
  4                  ,'/u01/app/oracle/oradata/cdb2/pdb_orcl');

Pluggable database created.
```

```
SQL>
```

c. Check the open mode of pdb_orcl.

```
SQL> col con_id format 999
SQL> col name format A10
SQL> select con_id, NAME, OPEN_MODE, DBID, CON_UID from V$PDBS;

CON_ID NAME       OPEN_MODE         DBID    CON_UID
------ ---------- ---------- ---------- ----------
     2 PDB$SEED   READ ONLY  4041282578 4041282578
     3 PDB2       READ WRITE 3083197298 3083197298
     4 PDB2_2     READ WRITE 2845295786 2845295786
     5 PDB_ORCL    MOUNTED    4229481674 4229481674

SQL>
```

d. Open pdb_orcl.

1) Use netca to add the PDB_ORCL net service name for pdb_orcl pluggable database of cdb2 in the tnsnames.ora file.

```
$ netca
```

2) On the Welcome page, select the "Local Net Service Name configuration" and click Next.
3) On the Net Service Name Configuration page, accept Add and click Next.
4) On the Net Service Name Configuration, Service Name page, enter pdb_orcl as Service Name and click Next.
5) On the Net Service Name Configuration, Select Protocols page, select TCP and click Next.
6) On the Net Service Name Configuration, TCP/IP Protocol page, enter your complete host name, for example, <*yourservername*>, or *localhost,* accept "Use the standard port number of 1521," and click Next.
7) On the Net Service Name Configuration, Test page, select "No, do not test" (the pluggable database is not yet opened) and click Next.
8) On the Net Service Name Configuration, Net Service Name page, accept pdb_orcl as Net Service Name and click Next.
9) On the Net Service Name Configuration, Another Net Service page, select No, and Next.
10) On the Net Service Name Configuration Complete page, click Next.
11) When you are back on the Welcome page, click Finish.
12) Connect to pdb_orcl AS SYSDBA.

```
$ sqlplus sys/oracle_4U@pdb_orcl AS SYSDBA

Connected to:
Oracle Database 12c Enterprise Edition Release 12.1.0.1.0 -
64bit Production
```

```
With the Partitioning, OLAP, Advanced Analytics and Real
Application Testing options


SQL>
```

13) Open the PDB.

```
SQL> alter pluggable database pdb_orcl open;


Pluggable database altered.


SQL> EXIT
$
```

2. Enable Unified Auditing.
   a. Shut down all Oracle processes of all instances. Shut down the listener.

```
$ lsnrctl stop


LSNRCTL for Linux: Version 12.1.0.1.0 - Production on 05-JUL-
2012 09:13:24


Copyright (c) 1991, 2012, Oracle.  All rights reserved.


Connecting to
(DESCRIPTION=(ADDRESS=(PROTOCOL=IPC)(KEY=EXTPROC1521)))
The command completed successfully
$
```

   b. Shut down all instances.

```
$ ps -ef | grep pmon
oracle   18211     1  0 Sep05 ?        00:00:53 ora_pmon_em12rep
oracle   25014     1  0 Sep07 ?        00:00:28 ora_pmon_cdb2
oracle   30114 29015  0 23:38 pts/3    00:00:00 grep pmon
$
```

1) Shut down the `cdb2` instance.

```
$ sqlplus / as sysdba


Connected to:
Oracle Database 12c Enterprise Edition Release 12.1.0.1.0 -
64bit Production
With the Partitioning, OLAP, Data Mining and Real Application
Testing options


SQL> shutdown immediate
Database closed.
Database dismounted.
```

```
ORACLE instance shut down.
SQL> EXIT
$
```

2) Shut down the `em12rep` instance.

a) Stop the OMS.

```
$ cd /u01/app/oracle/product/middleware/oms
$ export OMS_HOME=/u01/app/oracle/product/middleware/oms

$ $OMS_HOME/bin/emctl stop oms
Oracle Enterprise Manager Cloud Control 12c Release 2
Copyright (c) 1996, 2012 Oracle Corporation.  All rights
reserved.
Stopping WebTier...
WebTier Successfully Stopped
Stopping Oracle Management Server...
Oracle Management Server Successfully Stopped
Oracle Management Server is Down
$
```

b) Shut down the repository database instance `em12rep`.

```
$ . oraenv
[ORACLE_SID = [cdb2] ? em12rep
The Oracle base remains unchanged with value /u01/app/oracle
$
```

```
$ sqlplus / as sysdba

Connected to:
Oracle Database 12c Enterprise Edition Release 12.1.0.1.0 -
64bit Production
With the Partitioning, OLAP, Data Mining and Real Application
Testing options

SQL> shutdown immediate
Database closed.
Database dismounted.
ORACLE instance shut down.
SQL> EXIT
$
```

3) Verify that all instances are down.

```
$ ps -ef | grep pmon
oracle    5165 13370  0 09:14 pts/0    00:00:00 grep pmon
$
```

c. Enable the Unified Audit option.

```
$ cd $ORACLE_HOME/rdbms/lib
$ make -f ins_rdbms.mk uniaud_on ioracle
ORACLE_HOME=$ORACLE_HOME
/usr/bin/ar d
/u01/app/oracle/product/12.1.0/dbhome_1/rdbms/lib/libknlopt.a
kzanang.o
/usr/bin/ar cr
/u01/app/oracle/product/12.1.0/dbhome_1/rdbms/lib/libknlopt.a
/u01/app/oracle/product/12.1.0/dbhome_1/rdbms/lib/kzaiang.o
chmod 755 /u01/app/oracle/product/12.1.0/dbhome_1/bin


 - Linking Oracle
rm -f /u01/app/oracle/product/12.1.0/dbhome_1/rdbms/lib/oracle
/u01/app/oracle/product/12.1.0/dbhome_1/bin/orald  -o
/u01/app/oracle/product/12.1.0/dbhome_1/rdbms/lib/oracle -m64 -z
noexecstack -L/u01/app/oracle/product/12.1.0/dbhome_1/rdbms/lib/
-L/u01/app/oracle/product/12.1.0/dbhome_1/lib/ -
L/u01/app/oracle/product/12.1.0/dbhome_1/lib/stubs/   -Wl,-E
/u01/app/oracle/product/12.1.0/dbhome_1/rdbms/lib/opimai.o
/u01/app/oracle/product/12.1.0/dbhome_1/rdbms/lib/ssoraed.o
/u01/app/oracle/product/12.1.0/dbhome_1/rdbms/lib/ttcsoi.o -Wl,-
-whole-archive -lperfsrv12 -Wl,--no-whole-archive
/u01/app/oracle/product/12.1.0/dbhome_1/lib/nautab.o
/u01/app/oracle/product/12.1.0/dbhome_1/lib/naeet.o
/u01/app/oracle/product/12.1.0/dbhome_1/lib/naect.o
/u01/app/oracle/product/12.1.0/dbhome_1/lib/naedhs.o
/u01/app/oracle/product/12.1.0/dbhome_1/rdbms/lib/config.o  -
lserver12 -lodm12 -lcell12 -lnnet12 -lskgxp12 -lsnls12 -lnls12
-lcore12 -lsnls12 -lnls12 -lcore12 -lsnls12 -lnls12 -lxml12 -
lcore12 -lunls12 -lsnls12 -lnls12 -lcore12 -lnls12 -lclient12  -
lvsn12 -lcommon12 -lgeneric12 -lknlopt `if /usr/bin/ar tv
/u01/app/oracle/product/12.1.0/dbhome_1/rdbms/lib/libknlopt.a |
grep xsyeolap.o > /dev/null 2>&1 ; then echo "-loraolap12" ; fi`
-lskjcx12 -lslax12 -lpls12  -lrt -lplp12 -lserver12 -lclient12
-lvsn12 -lcommon12 -lgeneric12 `if [ -f
/u01/app/oracle/product/12.1.0/dbhome_1/lib/libavserver12.a ] ;
then echo "-lavserver12" ; else echo "-lavstub12"; fi` `if [ -f
/u01/app/oracle/product/12.1.0/dbhome_1/lib/libavclient12.a ] ;
then echo "-lavclient12" ; fi` -lknlopt -lslax12 -lpls12  -lrt -
lplp12 -ljavavm12 -lserver12  -lwwg  `cat
/u01/app/oracle/product/12.1.0/dbhome_1/lib/ldflags`    -
lncrypt12 -lnsgr12 -lnzjs12 -ln12 -lnl12 -lnro12 `cat
/u01/app/oracle/product/12.1.0/dbhome_1/lib/ldflags`    -
lncrypt12 -lnsgr12 -lnzjs12 -ln12 -lnl12 -lnnz12 -lzt12 -lztkg12
-lmm -lsnls12 -lnls12  -lcore12 -lsnls12 -lnls12 -lcore12 -
lsnls12 -lnls12 -lxml12 -lcore12 -lunls12 -lsnls12 -lnls12 -
lcore12 -lnls12 -lztkg12 `cat
/u01/app/oracle/product/12.1.0/dbhome_1/lib/ldflags`    -
lncrypt12 -lnsgr12 -lnzjs12 -ln12 -lnl12 -lnro12 `cat
```

```
/u01/app/oracle/product/12.1.0/dbhome_1/lib/ldflags`    -
lncrypt12 -lnsgr12 -lnzjs12 -ln12 -lnl12 -lnnz12 -lzt12 -lztkg12
-lsnls12 -lnls12  -lcore12 -lsnls12 -lnls12 -lcore12 -lsnls12 -
lnls12 -lxml12 -lcore12 -lunls12 -lsnls12 -lnls12 -lcore12 -
lnls12 `if /usr/bin/ar tv
/u01/app/oracle/product/12.1.0/dbhome_1/rdbms/lib/libknlopt.a |
grep "kxmnsd.o" > /dev/null 2>&1 ; then echo " " ; else echo "-
lordsdo12"; fi` -
L/u01/app/oracle/product/12.1.0/dbhome_1/ctx/lib/ -lctxc12 -
lctx12 -lzx12 -lgx12 -lctx12 -lzx12 -lgx12 -lordimt12 -lclsra12
-ldbcfg12 -lhasgen12 -lskgxn2 -lnnz12 -lzt12 -lxml12 -locr12 -
locrb12 -locrutl12 -lhasgen12 -lskgxn2 -lnnz12 -lzt12 -lxml12  -
lgeneric12 -loraz -llzopro -lorabz2 -lipp_z -lipp_bz2 -
lippdcemerged -lippsemerged -lippdcmerged  -lippsmerged -
lippcore  -lippcpemerged -lippcpmerged  -lsnls12 -lnls12  -
lcore12 -lsnls12 -lnls12 -lcore12 -lsnls12 -lnls12 -lxml12 -
lcore12 -lunls12 -lsnls12 -lnls12 -lcore12 -lnls12 -lsnls12 -
lunls12  -lsnls12 -lnls12  -lcore12 -lsnls12 -lnls12 -lcore12 -
lsnls12 -lnls12 -lxml12 -lcore12 -lunls12 -lsnls12 -lnls12 -
lcore12 -lnls12 -lasmclnt12 -lcommon12 -lcore12  -laio -lons
`cat /u01/app/oracle/product/12.1.0/dbhome_1/lib/sysliblist` -
Wl,-rpath,/u01/app/oracle/product/12.1.0/dbhome_1/lib -lm
`cat /u01/app/oracle/product/12.1.0/dbhome_1/lib/sysliblist` -
ldl -lm   -L/u01/app/oracle/product/12.1.0/dbhome_1/lib
test ! -f /u01/app/oracle/product/12.1.0/dbhome_1/bin/oracle ||\
           mv -f
/u01/app/oracle/product/12.1.0/dbhome_1/bin/oracle
/u01/app/oracle/product/12.1.0/dbhome_1/bin/oracleO
mv /u01/app/oracle/product/12.1.0/dbhome_1/rdbms/lib/oracle
/u01/app/oracle/product/12.1.0/dbhome_1/bin/oracle
chmod 6751 /u01/app/oracle/product/12.1.0/dbhome_1/bin/oracle
$
```

d. Restart the processes. Restart the `cdb2` database only.

```
$ . oraenv
[ORACLE_SID = [em12rep] ? cdb2
The Oracle base remains unchanged with value /u01/app/oracle
$
```

```
$ sqlplus / as sysdba

Connected to an idle instance.

SQL> startup mount
ORACLE instance started.

Total System Global Area 1068937216 bytes
Fixed Size                  2248280 bytes
```

```
Variable Size              343933352 bytes
Database Buffers           717225984 bytes
Redo Buffers                 5529600 bytes
Database mounted.

SQL> ALTER DATABASE ARCHIVELOG;

Database altered.

SQL> ALTER DATABASE OPEN;

Database altered.

SQL>
```

You can see that the Unified Auditing option is enabled in the SQL*Plus banner if you disconnect and reconnect.

e. Restart the listener.

```
$ lsnrctl start
…
Connecting to
(DESCRIPTION=(ADDRESS=(PROTOCOL=IPC)(KEY=EXTPROC1521)))
STATUS of the LISTENER
------------------------
Alias                     LISTENER
Version                   TNSLSNR for Linux: Version 12.1.0.1.0
- Production
Start Date                05-JUL-2012 09:37:38
Uptime                    0 days 0 hr. 0 min. 0 sec
Trace Level               off
Security                  ON: Local OS Authentication
SNMP                      OFF
Listener Parameter File
/u01/app/oracle/product/12.1.0/dbhome_1/network/admin/listener.o
ra
Listener Log File
/u01/app/oracle/diag/tnslsnr/host01/listener/alert/log.xml
Listening Endpoints Summary...
  (DESCRIPTION=(ADDRESS=(PROTOCOL=ipc)(KEY=EXTPROC1521)))

(DESCRIPTION=(ADDRESS=(PROTOCOL=tcp)(HOST=Your_server)(PORT=1521
)))
The listener supports no services
The command completed successfully
```

```
$
```

3. Create an audit policy AUDIT_TABLESPACE for any CREATE TABLESPACE operation in pdb_orcl.

```
SQL> CREATE AUDIT POLICY audit_tablespace
  2        ACTIONS create tablespace;

Audit policy created.

SQL>
```

4. Enable the audit policy.

```
SQL> audit policy AUDIT_TABLESPACE;

Audit succeeded.

SQL>
```

5. Check the audit policy existence.

```
SQL> col user_name format A10
SQL> col policy_name format A20
SQL> SELECT * FROM   AUDIT_UNIFIED_ENABLED_POLICIES
  2  where POLICY_NAME like '%TABLESPACE%';

USER_NAME  POLICY_NAME          ENABLED_ SUC FAI
---------- -------------------- --------- --- ---
ALL USERS  AUDIT_TABLESPACE     BY        YES YES

SQL>
```

6. Create a new tablespace in pdb_orcl and verify that the operation has been audited.
   a. Create a new tablespace TBS_ORCL.

```
SQL> CREATE TABLESPACE tbs_orcl DATAFILE
  2  '/u01/app/oracle/oradata/cdb2/pdb_orcl/tbs_orcl01.dbf'
  3  SIZE 100M;

Tablespace created.

SQL>
```

   b. Display the audit record.

```
SQL> COL dbusername FORMAT a12
SQL> COL action_name FORMAT a20
SQL> COL object_name FORMAT a20
SQL> SELECT dbusername, action_name, object_name
  2  FROM unified_audit_trail
```

```
  3  WHERE action_name like '%TABLESPACE%';

DBUSERNAME   ACTION_NAME         OBJECT_NAME
------------ -------------------- --------------------
SYSTEM       CREATE TABLESPACE   TBS_ORCL

SQL>
```

c. What happens if you create a tablespace from root? Is it audited?

```
SQL> CONNECT / as sysdba
Connected.
SQL> CREATE TABLESPACE tbs_root DATAFILE
  2  '/u01/app/oracle/oradata/cdb2/ tbs_root01.dbf' SIZE 10M;

Tablespace created.

SQL>
```

d. Display the audit records. You can see that the audit policy was applied only on the PDB where it was created and enabled, and not in the root.

```
SQL> COL dbusername FORMAT a12
SQL> COL action_name FORMAT a20
SQL> COL object_name FORMAT a18
SQL> SELECT con_id, dbusername, action_name, object_name
  2  FROM   cdb_unified_audit_trail
  3  WHERE  action_name like '%TABLESPACE%';

CON_ID     DBUSERNAME   ACTION_NAME         OBJECT_NAME
---------- ------------ -------------------- ------------------
         5 SYSTEM      CREATE TABLESPACE    TBS_ORCL

SQL>
```

e. Drop the tablespaces `tbs_root` and `tbs_orcl`.

```
SQL> DROP TABLESPACE tbs_root INCLUDING CONTENTS AND DATAFILES;

Tablespace dropped.

SQL> CONNECT sys/oracle_4U@pdb_orcl AS SYSDBA
Connected.
SQL> DROP TABLESPACE tbs_orcl INCLUDING CONTENTS AND DATAFILES;

Tablespace dropped.

SQL>
```

7. Create an audit policy AUDIT_USER for any CREATE USER or DROP USER operation in pdb2.
   a. Before creating the policy, check if these actions are not audited by default.

```
SQL> CONNECT system/oracle_4U@pdb2
Connected.
SQL> SELECT policy_name, user_name
  2  FROM audit_unified_enabled_policies;

POLICY_NAME          USER_NAME
-------------------- ----------
ORA_SECURECONFIG     ALL USERS

SQL>
```

   b. Check which actions are audited by ORA_SECURECONFIG audit policy.

```
SQL> set pages 100
SQL> COL audit_option FORMAT A40
SQL> SELECT  audit_option FROM audit_unified_policies
  2  WHERE  policy_name ='ORA_SECURECONFIG' ORDER BY 1;


AUDIT_OPTION
----------------------------------------
ADMINISTER KEY MANAGEMENT
ALTER ANY PROCEDURE
ALTER ANY SQL TRANSLATION PROFILE
ALTER ANY TABLE
ALTER DATABASE
ALTER DATABASE LINK
ALTER PLUGGABLE DATABASE
ALTER PROFILE
ALTER ROLE
ALTER SYSTEM
ALTER USER
AUDIT SYSTEM
CREATE ANY JOB
CREATE ANY LIBRARY
CREATE ANY PROCEDURE
CREATE ANY SQL TRANSLATION PROFILE
CREATE ANY TABLE
CREATE DATABASE LINK
CREATE DIRECTORY
CREATE EXTERNAL JOB
```

```
CREATE PLUGGABLE DATABASE
CREATE PROFILE
CREATE PUBLIC SYNONYM
CREATE ROLE
CREATE SQL TRANSLATION PROFILE
CREATE USER
DROP ANY PROCEDURE
DROP ANY SQL TRANSLATION PROFILE
DROP ANY TABLE
DROP DATABASE LINK
DROP DIRECTORY
DROP PLUGGABLE DATABASE
DROP PROFILE
DROP PUBLIC SYNONYM
DROP ROLE
DROP USER
EXEMPT ACCESS POLICY
EXEMPT REDACTION POLICY
GRANT ANY OBJECT PRIVILEGE
GRANT ANY PRIVILEGE
GRANT ANY ROLE
LOGMINING
LOGOFF
LOGON
PURGE DBA_RECYCLEBIN
SET ROLE
TRANSLATE ANY SQL

47 rows selected.

SQL>
```

c. It is useless to create an audit policy for any `CREATE USER` or `DROP USER` operation because the `ORA_SECURECONFIG` audit policy is enabled by default for all users and audits any `CREATE USER` and `DROP USER`.

8. Connect as `lu_pdb2` in `pdb2` and create a new user and drop it.

```
SQL> CREATE USER lu_pdb2 IDENTIFIED BY oracle_4U;

User created.

SQL> GRANT dba TO lu_pdb2;

Granted succeeded.
```

```
SQL> CONNECT lu_pdb2/oracle_4U@pdb2
Connected.
SQL> CREATE USER test IDENTIFIED BY test;

User created.

SQL> DROP USER test;

User dropped.

SQL>
```

9. Verify that the audit policy audited the two operations. Use the UNIFIED_AUDIT_TRAIL view. In case the in-memory audit records audit information has not been flushed to tables, execute the DBMS_AUDIT_MGMT.FLUSH_UNIFIED_AUDIT_TRAIL procedure.

```
SQL> CONNECT system/oracle_4U@pdb2
Connected.
SQL> COL dbusername FORMAT a12
SQL> COL action_name FORMAT a20
SQL> COL object_name FORMAT a20
SQL> SELECT dbusername, action_name, object_name
  2  FROM unified_audit_trail
  3  WHERE dbusername='LU_PDB2';

DBUSERNAME   ACTION_NAME          OBJECT_NAME
------------ -------------------- --------------------
LU_PDB2      CREATE USER          TEST
LU_PDB2      DROP USER            TEST
LU_PDB2      LOGOFF

SQL>
```

10. Note that if you connect to root and attempt to read the audited records collected for pdb2, you do not find any information. The UNIFIED_AUDIT_TRAIL view inside root displays only the root's audit records, if any. Read from CDB_UNIFIED_AUDIT_TRAIL view, the consolidated view of all PDBs.

```
SQL> CONNECT / AS SYSDBA
Connected.
SQL> COL action_name FORMAT a20
SQL> COL object_name FORMAT a20
SQL> SELECT dbusername, action_name, object_name
  2  FROM unified_audit_trail
  3  WHERE dbusername='LU_PDB2';
```

```
no rows selected

SQL>
```

```
SQL> SELECT dbusername, action_name, object_name
  2  FROM cdb_unified_audit_trail
  3  WHERE dbusername='LU_PDB2';

DBUSERNAME   ACTION_NAME          OBJECT_NAME
------------ -------------------- --------------------
LU_PDB2      CREATE USER          TEST
LU_PDB2      DROP USER            TEST
LU_PDB2      LOGOFF

SQL> EXIT
$
```

Practice 9-2: Export From non-CDB and Import Into PDB

Overview

In this practice, you will use the FULL TRANSPORTABLE mode to export from the non-CDB `orcl` and import into a new PDB `pdb_orcl`.

Assumption

`pdb_orcl` is successfully created after practice 9-1.

Tasks

1. Export the non-CDB `orcl` using FULL TRANSPORTABLE mode.
   a. Start up the instance first.

```
$ . oraenv
ORACLE_SID = [cdb2] ? orcl
The Oracle base for
ORACLE_HOME=/u01/app/oracle/product/12.1.0/dbhome_2 is
/u01/app/oracle
$ sqlplus / as sysdba

Connected to an idle instance.

SQL> STARTUP
ORACLE instance started.

Total System Global Area  501059584 bytes
Fixed Size                  2290024 bytes
Variable Size             264244888 bytes
Database Buffers          226492416 bytes
Redo Buffers                8032256 bytes
Database mounted.
Database opened.
SQL>
```

   b. List the tablespaces and the number of rows in `HR.EMPLOYEES` table.

```
SQL> SELECT tablespace_name FROM dba_tablespaces;

TABLESPACE_NAME
------------------------------
SYSTEM
SYSAUX
UNDOTBS1
TEMP
USERS
```

```
EXAMPLE

6 rows selected.

SQL> SELECT count(*)FROM hr.employees;

  COUNT(*)
----------
       107

SQL>
```

c. Set the tablespaces in read-only mode.

```
SQL> ALTER TABLESPACE example READ ONLY;

Tablespace altered.

SQL> ALTER TABLESPACE users READ ONLY;

Tablespace altered.

SQL> EXIT
$
```

d. Proceed with the export operation.

```
$ rm /u01/app/oracle/admin/orcl/dpdump/expfull.dmp
rm: cannot remove
`/u01/app/oracle/admin/orcl/dpdump/expfull.dmp': No such file or
directory
$ expdp system/oracle_4U DUMPFILE=expfull.dmp FULL=Y
TRANSPORTABLE=ALWAYS LOGFILE=exp.log

Starting "SYSTEM"."SYS_EXPORT_FULL_01":  system/********
DUMPFILE=expfull.dmp FULL=Y TRANSPORTABLE=ALWAYS LOGFILE=exp.log
Estimate in progress using BLOCKS method...
Processing object type
DATABASE_EXPORT/PLUGTS_FULL/FULL/PLUGTS_TABLESPACE
…
Master table "SYSTEM"."SYS_EXPORT_FULL_01" successfully
loaded/unloaded
******************************************************************
*
Dump file set for SYSTEM.SYS_EXPORT_FULL_01 is:
  /u01/app/oracle/admin/orcl/dpdump/expfull.dmp
```

```
******************************************************************
*
Datafiles required for transportable tablespace EXAMPLE:
  /u01/app/oracle/oradata/orcl/example01.dbf
Datafiles required for transportable tablespace USERS:
  /u01/app/oracle/oradata/orcl/users01.dbf
Job "SYSTEM"."SYS_EXPORT_FULL_01" successfully completed at Sat
Nov 17 03:07:37 2012 elapsed 0 00:23:08


$
```

3. Copy the data files to the target locations `/u01/app/oracle/oradata/cdb2/pdb_orcl` and the export dumpfile to `/u01/app/oracle/admin/cdb2/dpdump`. Before proceeding, check that there are not any tablespaces in the target `pdb_orcl` having the same names as the tablespaces in the source `orcl` database.
   a. Set your environment to the target database `pdb_orcl`.

```
$ . oraenv
ORACLE_SID = [orcl] ? cdb2
The Oracle base for
ORACLE_HOME=/u01/app/oracle/product/12.1.0/dbhome_2 is
/u01/app/oracle
$ sqlplus system/oracle_4U@pdb_orcl


Connected to:
Oracle Database 12c Enterprise Edition Release 12.1.0.1.0 -
64bit Production
With the Partitioning, OLAP, Advanced Analytics, Real
Application Testing and Unified Auditing options


SQL> SELECT tablespace_name FROM dba_tablespaces;


TABLESPACE_NAME
------------------------------
SYSTEM
SYSAUX
TEMP


SQL>
```

   b. Create a Data Pump directory for the dump files stored for any Data Pump operations in `pdb_orcl`.

```
SQL> create directory dp_orcl as
'/u01/app/oracle/admin/cdb2/dpdump';


Directory created.
```

```
SQL> EXIT
$
```

c. Now you can copy the data files to the target locations `/u01/app/oracle/oradata/cdb2/pdb_orcl` and the export dumpfile to `/u01/app/oracle/admin/cdb2/dpdump`.

```
$ cp /u01/app/oracle/oradata/orcl/example01.dbf
/u01/app/oracle/oradata/orcl/users01.dbf
/u01/app/oracle/oradata/cdb2/pdb_orcl
$
$ cp /u01/app/oracle/admin/orcl/dpdump/expfull.dmp
/u01/app/oracle/admin/cdb2/dpdump/expfull.dmp
$
```

4. Import the `orcl` database into the `pdb_orcl` in FULL TRANSPORTABLE mode. There are many errors due to the APEX option, which needs to be handled. But for the aim of this practice, these errors can be ignored. Note that the `impdp` command includes the net service_name in the userid clause.

```
$ rm /u01/app/oracle/admin/cdb2/dpdump/import.log
rm: cannot remove
`/u01/app/oracle/admin/cdb2/dpdump/import.log': No such file or
directory
$
$ impdp system/oracle_4U@pdb_orcl FULL=Y dumpfile=expfull.dmp
directory=dp_orcl
TRANSPORT_DATAFILES='/u01/app/oracle/oradata/cdb2/pdb_orcl/users
01.dbf','/u01/app/oracle/oradata/cdb2/pdb_orcl/example01.dbf'
logfile=import.log


Master table "SYSTEM"."SYS_IMPORT_FULL_01" successfully
loaded/unloaded
Source timezone version is +00:00 and target timezone version is
-08:00.
Starting "SYSTEM"."SYS_IMPORT_FULL_01":
system/********@pdb_orcl FULL=Y dumpfile=expfull.dmp
directory=dp_orcl
TRANSPORT_DATAFILES=/u01/app/oracle/oradata/cdb2/pdb_orcl/users0
1.dbf,/u01/app/oracle/oradata/cdb2/pdb_orcl/example01.dbf
logfile=import.log
Processing object type
DATABASE_EXPORT/PRE_SYSTEM_IMPCALLOUT/MARKER
Processing object type
DATABASE_EXPORT/PRE_INSTANCE_IMPCALLOUT/MARKER
Processing object type DATABASE_EXPORT/PLUGTS_FULL/PLUGTS_BLK
Processing object type DATABASE_EXPORT/TABLESPACE
…
```

```
Job "SYSTEM"."SYS_IMPORT_FULL_01" completed with 66 error(s) at
Sat Nov 17 03:48:42 2012 elapsed 0 00:29:43


$
```

5. Check that the tablespaces EXAMPLE and USERS are in place, and that the HR.EMPLOYEES table is created.

```
$ sqlplus sys/oracle_4U@pdb_orcl as sysdba

Connected to:
Oracle Database 12c Enterprise Edition Release 12.1.0.1.0 -
64bit Production
With the Partitioning, OLAP, Advanced Analytics, Real
Application Testing and Unified Auditing options

SQL> SELECT tablespace_name from DBA_TABLESPACES;

TABLESPACE_NAME
------------------------------
SYSTEM
SYSAUX
TEMP
EXAMPLE
USERS

SQL> SELECT count(*) FROM hr.employees;

  COUNT(*)
----------
       107

SQL> EXIT
$
```

6. Set the tablespaces of orcl back to read-write mode.

```
$ . oraenv
ORACLE_SID = [cdb2] ? orcl
The Oracle base for
ORACLE_HOME=/u01/app/oracle/product/12.1.0/dbhome_2 is
/u01/app/oracle
$ sqlplus / as sysdba

Connected to:
Oracle Database 12c Enterprise Edition Release 12.1.0.1.0 -
64bit Production
```

```
With the Partitioning, OLAP, Advanced Analytics, Real
Application Testing and Unified Auditing options

SQL> ALTER TABLESPACE example READ WRITE;

Tablespace altered.

SQL> ALTER TABLESPACE users READ WRITE;

Tablespace altered.

SQL> EXIT
$
```

Practice 9-3: Export and Import Between PDBs

Overview

In this practice, you will export a whole schema from one PDB `pdb_orcl` to another PDB `pdb2` within the same CDB. The schema `HR` will be exported from `pdb_orcl` and imported into `pdb2`.

Assumptions

The `pdb_orcl` has been successfully created during practice 9-1 and stores `HR` schema tables after practice 9-2.

If `pdb_orcl` could not be created successfully and imported with `ORCL HR` schema, execute the following catchup script:

```
$ cd /home/oracle/solutions/catchup_09_02
$ ./cr_imp_PDB_ORCL.sh
```

The applications from non-CDB `orcl` has been successfully exported and then imported into `pdb_orcl` during practice 9-2.

Tasks

1. If you had to use the `catchup_09_02` script, then the `DP_ORCL` directory does not exist any more. Re-create the directory as in practice 9-2 3.b).

```
$ sqlplus system/oracle_4U@pdb_orcl


Connected to:
Oracle Database 12c Enterprise Edition Release 12.1.0.1.0 -
64bit Production
With the Partitioning, OLAP, Advanced Analytics, Real
Application Testing and Unified Auditing options


SQL> create directory dp_orcl as
'/u01/app/oracle/admin/cdb2/dpdump';


Directory created.


SQL> EXIT
$
```

2. Export the schema `HR` from `pdb_orcl`.

```
$ expdp system/oracle_4U@pdb_orcl DUMPFILE=exppdb_orcl
DIRECTORY=dp_orcl SCHEMAS=hr


Starting "SYSTEM"."SYS_EXPORT_SCHEMA_01":
system/********@pdb_orcl DUMPFILE=exppdb_orcl DIRECTORY=dp_orcl
SCHEMAS=hr
…
```

```
. . exported "HR"."COUNTRIES"                                6.437
KB      25 rows
. . exported "HR"."DEPARTMENTS"                              7.101
KB      27 rows
. . exported "HR"."EMPLOYEES"                                17.06
KB     107 rows
. . exported "HR"."JOBS"                                     7.085
KB      19 rows
. . exported "HR"."JOB_HISTORY"                              7.171
KB      10 rows
. . exported "HR"."LOCATIONS"                                8.414
KB      23 rows
. . exported "HR"."REGIONS"                                  5.523
KB       4 rows
Master table "SYSTEM"."SYS_EXPORT_SCHEMA_01" successfully
loaded/unloaded
******************************************************************
*
Dump file set for SYSTEM.SYS_EXPORT_SCHEMA_01 is:
  /u01/app/oracle/admin/cdb2/dpdump/exppdb_orcl.dmp
Job "SYSTEM"."SYS_EXPORT_SCHEMA_01" successfully completed at
Sat Nov 17 03:56:00 2012 elapsed 0 00:03:12


$
```

3. Import the schema HR into pdb2.
   a. Create a Data Pump directory in pdb2.

```
$ sqlplus system/oracle_4U@pdb2


Connected to:
Oracle Database 12c Enterprise Edition Release 12.1.0.1.0 -
64bit Production
With the Partitioning, OLAP, Advanced Analytics, Real
Application Testing and Unified Auditing options


SQL> CREATE DIRECTORY dp_pdb2 AS
  2         '/u01/app/oracle/admin/cdb2/dpdump';


Directory created.


SQL>
```

   b. Create the tablespace USERS and EXAMPLE for the HR schema.

```
SQL> CREATE TABLESPACE users DATAFILE
  2  '/u01/app/oracle/oradata/cdb2/pdb2_1/users01.dbf'
  3  SIZE 100M;
```

```
Tablespace created.

SQL> CREATE TABLESPACE example DATAFILE
  2  '/u01/app/oracle/oradata/cdb2/pdb2_1/example01.dbf'
  3  SIZE 100M;

Tablespace created.

SQL> EXIT
$
```

c. Import the schema HR into pdb2.

```
$ rm /u01/app/oracle/admin/cdb2/dpdump/import.log
$ impdp system/oracle_4U@pdb2 DUMPFILE=exppdb_orcl
DIRECTORY=dp_pdb2 SCHEMAS=hr

Master table "SYSTEM"."SYS_IMPORT_SCHEMA_01" successfully
loaded/unloaded
Starting "SYSTEM"."SYS_IMPORT_SCHEMA_01":  system/********@pdb2
DUMPFILE=exppdb_orcl DIRECTORY=dp_pdb2 SCHEMAS=hr
Processing object type SCHEMA_EXPORT/USER
Processing object type SCHEMA_EXPORT/SYSTEM_GRANT
Processing object type SCHEMA_EXPORT/ROLE_GRANT
Processing object type SCHEMA_EXPORT/DEFAULT_ROLE
Processing object type SCHEMA_EXPORT/PRE_SCHEMA/PROCACT_SCHEMA
Processing object type SCHEMA_EXPORT/SEQUENCE/SEQUENCE
Processing object type SCHEMA_EXPORT/TABLE/TABLE
Processing object type SCHEMA_EXPORT/TABLE/TABLE_DATA
. . imported "HR"."COUNTRIES"                              6.437
KB      25 rows
. . imported "HR"."DEPARTMENTS"                            7.101
KB      27 rows
. . imported "HR"."EMPLOYEES"                              17.06
KB     107 rows
. . imported "HR"."JOBS"                                   7.085
KB      19 rows
. . imported "HR"."JOB_HISTORY"                            7.171
KB      10 rows
. . imported "HR"."LOCATIONS"                              8.414
KB      23 rows
. . imported "HR"."REGIONS"                                5.523
KB       4 rows
Processing object type
SCHEMA_EXPORT/TABLE/GRANT/OWNER_GRANT/OBJECT_GRANT
```

```
ORA-39083: Object type OBJECT_GRANT failed to create with error:
ORA-01917: user or role 'OE' does not exist
Failing sql is:
GRANT SELECT ON "HR"."COUNTRIES" TO "OE"
ORA-39083: Object type OBJECT_GRANT failed to create with error:
ORA-01917: user or role 'OE' does not exist
Failing sql is:
GRANT REFERENCES ON "HR"."COUNTRIES" TO "OE"
ORA-39083: Object type OBJECT_GRANT failed to create with error:
ORA-01917: user or role 'OE' does not exist
Failing sql is:
GRANT SELECT ON "HR"."JOB_HISTORY" TO "OE"
ORA-39083: Object type OBJECT_GRANT failed to create with error:
ORA-01917: user or role 'OE' does not exist
Failing sql is:
GRANT SELECT ON "HR"."EMPLOYEES" TO "OE"
ORA-39083: Object type OBJECT_GRANT failed to create with error:
ORA-01917: user or role 'OE' does not exist
Failing sql is:
GRANT REFERENCES ON "HR"."EMPLOYEES" TO "OE"
ORA-39083: Object type OBJECT_GRANT failed to create with error:
ORA-01917: user or role 'OE' does not exist
Failing sql is:
GRANT SELECT ON "HR"."JOBS" TO "OE"
ORA-39083: Object type OBJECT_GRANT failed to create with error:
ORA-01917: user or role 'OE' does not exist
Failing sql is:
GRANT REFERENCES ON "HR"."LOCATIONS" TO "OE"
ORA-39083: Object type OBJECT_GRANT failed to create with error:
ORA-01917: user or role 'OE' does not exist
Failing sql is:
GRANT SELECT ON "HR"."LOCATIONS" TO "OE"
ORA-39083: Object type OBJECT_GRANT failed to create with error:
ORA-01917: user or role 'OE' does not exist
Failing sql is:
GRANT SELECT ON "HR"."DEPARTMENTS" TO "OE"
Processing object type SCHEMA_EXPORT/TABLE/COMMENT
Processing object type SCHEMA_EXPORT/PROCEDURE/PROCEDURE
Processing object type SCHEMA_EXPORT/PROCEDURE/ALTER_PROCEDURE
Processing object type SCHEMA_EXPORT/TABLE/INDEX/INDEX
Processing object type SCHEMA_EXPORT/TABLE/CONSTRAINT/CONSTRAINT
Processing object type
SCHEMA_EXPORT/TABLE/INDEX/STATISTICS/INDEX_STATISTICS
```

```
Processing object type SCHEMA_EXPORT/VIEW/VIEW
Processing object type
SCHEMA_EXPORT/TABLE/CONSTRAINT/REF_CONSTRAINT
Processing object type SCHEMA_EXPORT/TABLE/TRIGGER
Processing object type
SCHEMA_EXPORT/TABLE/STATISTICS/TABLE_STATISTICS
Processing object type SCHEMA_EXPORT/STATISTICS/MARKER
Job "SYSTEM"."SYS_IMPORT_SCHEMA_01" completed with 9 error(s) at
Sat Nov 17 04:04:54 2012 elapsed 0 00:05:11


$
```

Note that the HR objects are imported. The errors related to GRANT operations can be ignored. They are due to the OE user absence. What we want to show here is that we can export and import between PDBs. You may not observe errors if you ran the catchup_09_02 script.

d. Check that there are two distinct HR local users in cdb2, one in pdb_orcl and another one in pdb2.

```
$ sqlplus sys/oracle_4U@cdb2 as sysdba


Connected to:
Oracle Database 12c Enterprise Edition Release 12.1.0.1.0 -
64bit Production
With the Partitioning, OLAP, Advanced Analytics, Real
Application Testing and Unified Auditing options

SQL> COL username FORMAT A20
SQL> SELECT username, con_id, common
  2  FROM cdb_users WHERE username= 'HR';

USERNAME              CON_ID    COM
-------------------- ---------- ---
HR                            3 NO
HR                            5 NO

SQL> EXIT
$
```