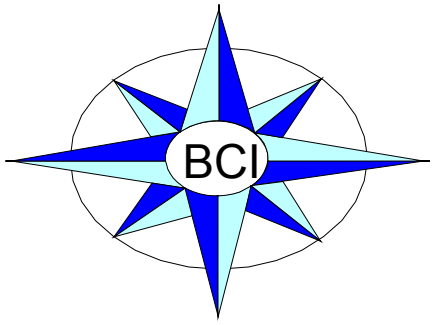


Practices for Lesson 2: Creating a Multitenant Container Database and Pluggable Databases

Chapter 2

Practices for Lesson 2

Practices Overview

In this practice you will create a new CDB named `cdb1` with DBCA with no PDB except the seed.

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 `pdb1_1` from seed in `cdb1`.
- Clone `pdb1_2` in `cdb1` from `pdb1_1`.
- Plug the non-CDB `orcl2` into the CDB `cdb1` as `pdb_orcl2`.
- Merge the two CDBs `cdb1` and `orcl12c` into `cdb1`, and optionally drop the database `cdb1` (optional practice).

Finally, you drop the `pdb1_2` using either DBCA or SQL Developer or SQL*Plus after the creation of this PDB.

Practice 2-1: Creating a New CDB

Overview

In this practice, you will create a new CDB named `cdb1` with DBCA.

Assumptions

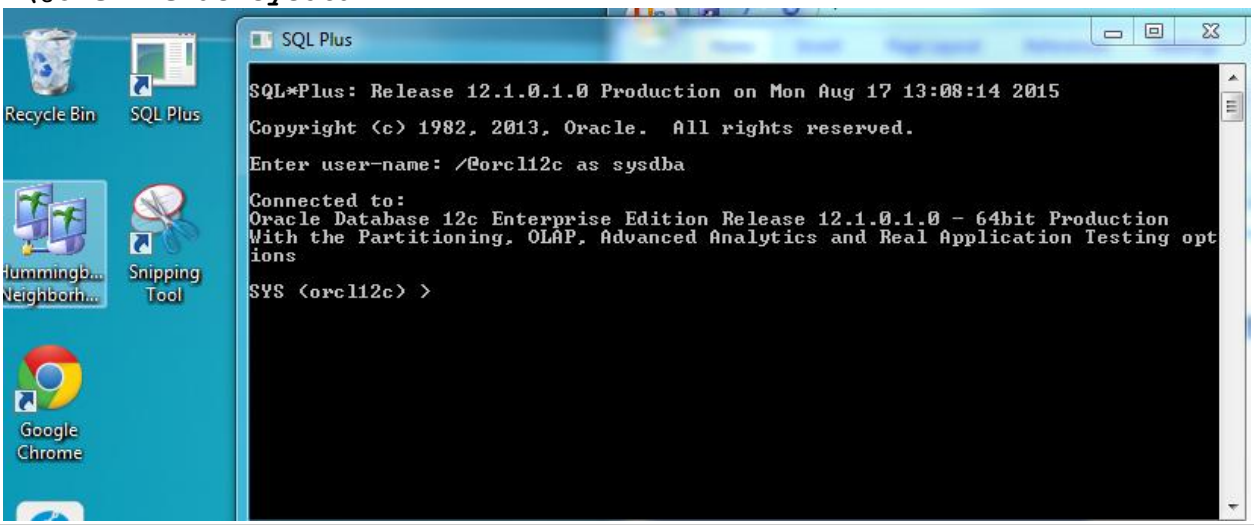
The created CDB `orcl12c` already exists.

Tasks

1. Create a CDB named `cdb1` using DBCA. First release resources held by other instances, shutting down the `orcl12c` and `pec1` instances.

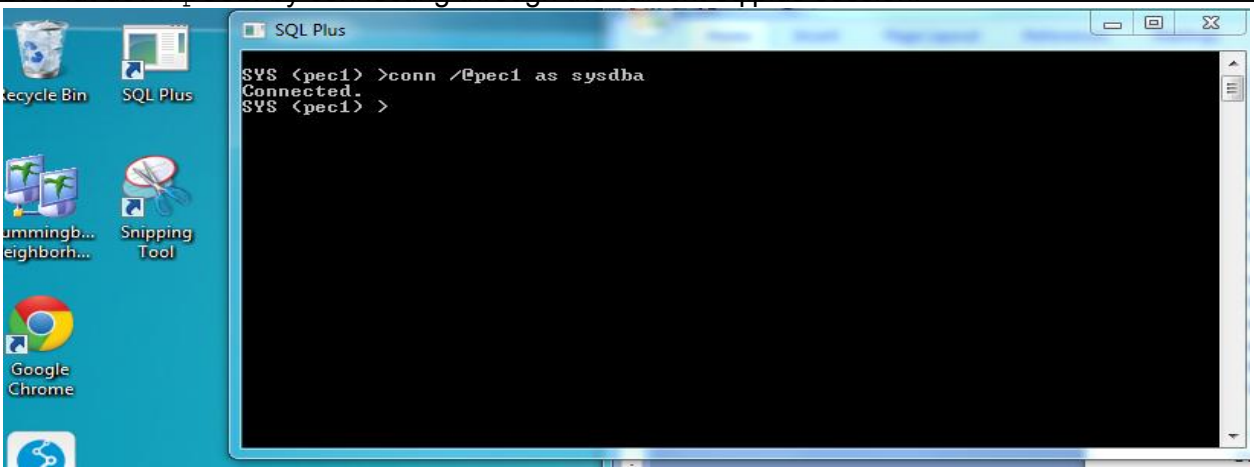
a. Shut down `orcl12c` by bringing up sqlplus via your desktop. Login as:

```
\@orcl12c as sysdba
```




```
Connected to:
Oracle Database 12c Enterprise Edition Release 12.1.0.0.2 -64bit Production
With the Partitioning, OLAP, Data Mining, Real Application
Testing
SQL> SHUTDOWN IMMEDIATE
Database closed.
Database dismounted.
ORACLE instance shut down.
SQL> EXIT
```

b. Shut down `pec1` by connecting through the current sqlplus session:



OR you can connect via command prompt as shown:

```
C:\ora12clabs sqlplus /@pecl as sysdba
Connected to:
Oracle Database 12c Enterprise Edition Release 12.1.0.0.2 -
64bit Production
With the Partitioning, OLAP, Data Mining, Real Application
Testing
SQL> SHUTDOWN IMMEDIATE
Database closed.
Database dismounted.
ORACLE instance shut down.
SQL> EXIT
c:\ora12clabs
```

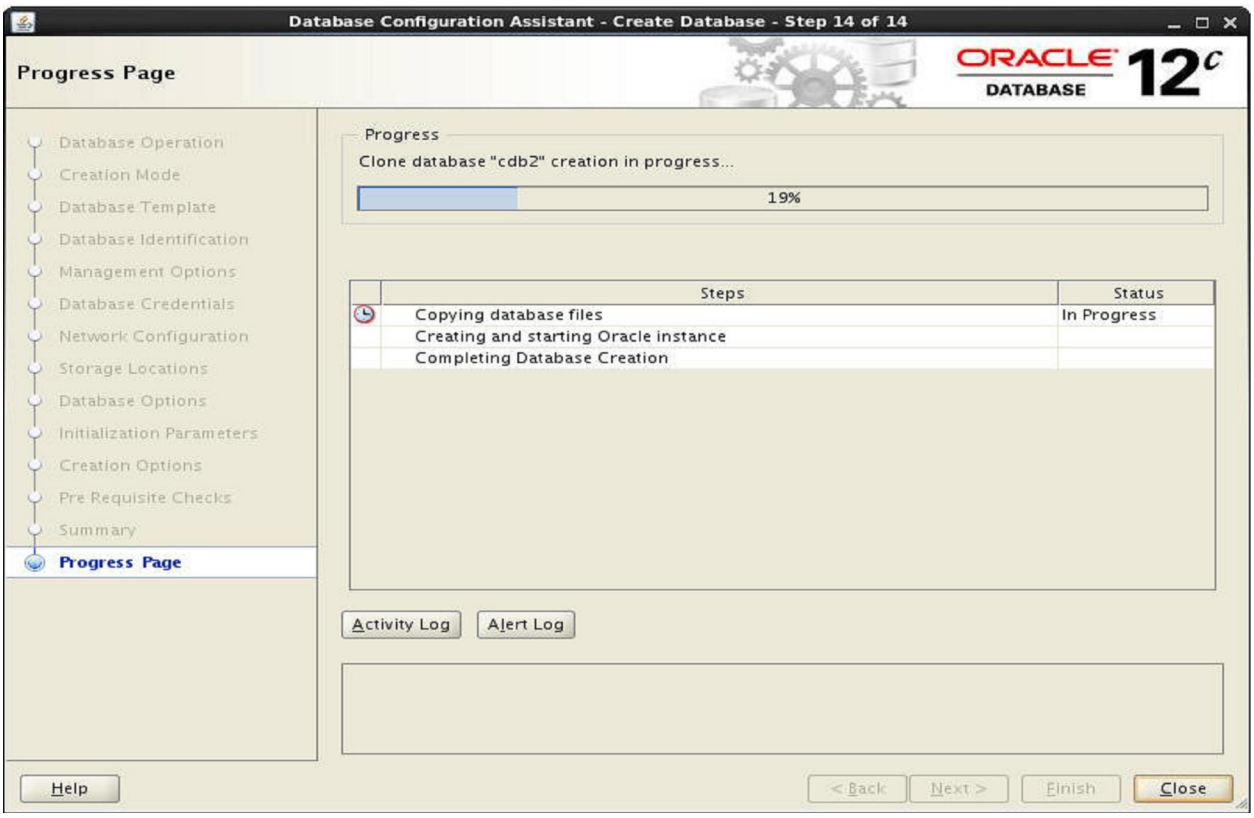
d. Start dbca and perform the following steps.

start | all programs | Oracle - OraDBHome1 | Configuration Migration Tools | Database Configuration Assistant

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: **cdb1 |** SID: cdb1 Select **“Create As Container Database”** Select **“Create An Empty Container Database”** Click Next.
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: `password` Confirm password: `password` Click Next.
g.	Step 7: Network Configuration	Listener Selection: Click Next

Step	Window\Page Description	Choices or Values
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.


Database Configuration Assistant - Create Database - Step 14 of 14
Progress Page
ORACLE DATABASE 12c
Database Operation
Creation Mode
Database Template
Database Identification
Management Options
Database Credentials
Network Configuration
Storage Locations
Database Options
Initialization Parameters
Creation Options
Pre Requisite Checks
Summary
Progress Page
Progress
Clone database "cdb2" creation in progress...
19%
Steps
Status
Copying database files
In Progress
Creating and starting Oracle instance
Completing Database Creation
Activity Log
Alert Log
Help
< Back
Next >
Finish
Close

Practice 2-2: Exploring CDB and PDB Structures

Overview

In this practice, you check the physical and logical structures of the new CDB `cdb1` and its seed PDB.

Tasks

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

From the desktop connect to: **`/@cdb1 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>
```

a. 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>
```

b. Check the instance name.

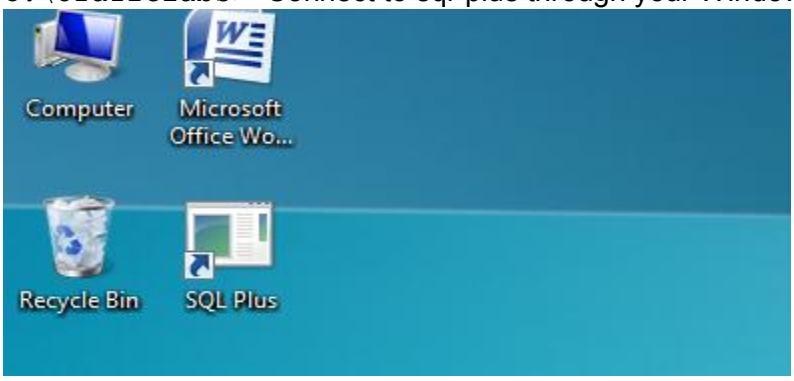
```
SQL> SELECT INSTANCE_NAME, STATUS, CON_ID from v$instance;
INSTANCE_NAME STATUS CON_ID
---------------- ------------ ----------
cdb1             OPEN             0
SQL> EXIT
```

c. To check Oracle Network services.

```
c:\ora12clabs> lsnrctl status
LSNRCTL for WIndows Version 12.1.0.1.0 - Production on 07-SEP-2012 01:10:16
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 Windows: Version 12.1.0.1.0
- Production
Start Date 14-SEP-2012 03:04:56
Uptime 16 days 21 hr. 48 min. 35 sec
Trace Level off
Security ON: Local OS Authentication
SNMP OFF
Listener Parameter File
C:c:\app\oracle\product\12.1.0\dbhome_1\network\admin\listener.ora
Listener Log File
C:c:\app\oracle\product\12.1.0\dbhome_1\log\diag\tnslsnr
\101-14\listener\alert\log.xml
```

```
Listening Endpoints Summary...
DESCRIPTION=(ADDRESS=(PROTOCOL=ipc)(PIPENAME=\\.\pipe\EXTPROC1521ipc))
  (DESCRIPTION=(ADDRESS=(PROTOCOL=tcp)(HOST=101-14)(PORT=1521)))
  (DESCRIPTION=(ADDRESS=(PROTOCOL=tcp)(HOST=127.0.0.1)(PORT=1521)))
  (DESCRIPTION=(ADDRESS=(PROTOCOL=tcps)(HOST=101-14)(PORT=5500))
(Security=(my_wllet_directory=C:c:\app\ORACLE\admin\orcl12c\xdb_wallet
))(Presentation=HTTP)(Sesson=RAW))
  (DESCRIPTION=(ADDRESS=(PROTOCOL=tcps)(HOST=101-14)(PORT=5501))
(Security=(my_wllet_directory=C:c:\app\ORACLE\admin\oms_db\xdb_wallet)
)(Presentation=HTTP)(Sessin=RAW))
  (DESCRIPTION=(ADDRESS=(PROTOCOL=tcps)(HOST=101-14)(PORT=5502))
(Security=(my_wllet_directory=C:c:\app\ORACLE\admin\pec1\xdb_wallet))
(Presentation=HTTP)(SessionRAW))
Services Summary...
Service "CLRExtProc" has 1 instance(s).
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...
  Instance "CLRExtProc", status UNKNOWN, has 1 handler(s) for this service
Service "oms_db" has 2 instance(s).
  Instance "OMS_DB", status UNKNOWN, has 1 handler(s) for this service...
  Instance "omsdb", status READY, has 1 handler(s) for this service...
Service "omsdbXDB" has 1 instance(s).
  Instance "omsdb", status READY, has 1 handler(s) for this service...
Service "orcl12c" has 2 instance(s).
  Instance "orcl12c", status UNKNOWN, has 1 handler(s) for this service...
  Instance "orcl12c", status READY, has 1 handler(s) for this service...
Service "orcl12cXDB" has 1 instance(s).
  Instance "orcl12c", status READY, has 1 handler(s) for this service...
Service "pdborcl" has 2 instance(s).
  Instance "orcl12c", status READY, has 1 handler(s) for this service...
  Instance "pdborcl", status UNKNOWN, has 1 handler(s) for this service...
Service "pec1" has 1 instance(s).
  Instance "pec1", status READY, has 1 handler(s) for this service...
Service "pec1XDB" has 1 instance(s).
  Instance "pec1", status READY, has 1 handler(s) for this service...
The command completed successfully
```

C:\ora12clabs> Connect to sql*plus through your Windows Desktop.




```
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$services;
NAME CON_ID
-------------------- ----------
cdb1XDB                       1
cdb1                          1
SYS$BACKGROUND                1
SYS$USERS                     1
SQL>
```

Notice that PDB$SEED service is not listed. No one 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> connect / as sysdba
Connected.
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      c:\app\oracle\oradata\cdb1\redo03.log       0
2      c:\app\oracle\oradata\cdb1\redo02.log       0
1      c:\app\oracle\oradata\cdb1\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
----------------------------------------------------- -------
c:\app\oracle\oradata\cdb1\control01.ctl                    0
c:\app\oracle\fast_recovery_area\cdb1\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
------------------------------                    ---------- -----
c:\app\oracle\oradata\cdb1\users01.dbf USERS              6     1
c:\app\oracle\oradata\cdb1\undotbs01.dbf UNDOTBS1         4     1
c:\app\oracle\oradata\cdb1\sysaux01.dbf SYSAUX            3     1
c:\app\oracle\oradata\cdb1\system01.dbf SYSTEM            1     1
c:\app\oracle\oradata\cdb1\pdbseed\system01.dbf SYSTEM     5     2
c:\app\oracle\oradata\cdb1\pdbseed\sysaux01.dbf SYSAUX     7     2
6 rows selected.
SQL>
```

d. Still connected to the root, now 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
------------------------------------------ ---------- -------
c:\app\oracle\oradata\cdb1\users01.dbf USERS 6
c:\app\oracle\oradata\cdb1\undotbs01.dbf UNDOTBS1 4
c:\app\oracle\oradata\cdb1\sysaux01.dbf SYSAUX 3
c:\app\oracle\oradata\cdb1\system01.dbf SYSTEM 1
SQL> EXIT
```

Notice that only root data files are listed.

e. Start the cdb1 database if it is not already started..

```
$ 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 Name page, select No, and Next.

10) On the Net Service Name Configuration Done 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 12.1.0.0.2 -
64bit Production
With the Partitioning, OLAP, Data Mining, Real Application
Testing
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.

```
sql*plus> system/password@pdb1_1
SQL*Plus: Release 12.1.0.0.2 Production on Fri Sep 7 01:28:32
2012
Copyright (c) 1982, 2012, Oracle. All rights reserved.
Last Successful login time: Wed Aug 22 2012 13:16:11 +00:00
Connected to:
Oracle Database 12c Enterprise Edition Release 12.1.0.0.2 -
64bit Production
With the Partitioning, OLAP, Advanced Analytics and Real
Application Testing options

SQL> col file_name format A65
SQL> set linesize 132
SQL> SELECT FILE_NAME, TABLESPACE_NAME, FILE_ID
2 from dba_data_files;
SYSTEM (cdb1) >l
  1* select file_name, tablespace_name TS_NAME,file_id from dba_data_files
SYSTEM (cdb1) >/  -- Note we are using the forward slash to execute

FILE_NAME                                                          TS_NAME
------------------------------------------------------------       -------
FILE_ID
C:\APP\ORACLE\ORADATA\CDB1\9E03ED8956664A0B82A5360632CAEDA7\DATAF ILE\O1_MF_SYSTEM_BX6JJX1F_.DBF  SYSTEM
7
C:\APP\ORACLE\ORADATA\CDB1\9E03ED8956664A0B82A5360632CAEDA7\DATAF ILE\O1_MF_SYSAUX_BX6JJX1G_.DBF  SYSAUX
8
C:\APP\ORACLE\ORADATA\CDB1\9E03ED8956664A0B82A5360632CAEDA7\DATAF ILE\O1_MF_USERS_BX6JL2FD_.DBF    USERS
9
3 rows selected.
SYSTEM (cdb1) >
```

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
----------       ------------    ----------  ----------
5                UNDOTBS1        2           0
7                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
-------------------------------------------------------  ---------------
C:\APP\ORACLE\ORADATA\CDB1\9E03ED8956664A0B82A5360632CAEDA7\
DATAFI LE\O1_MF_TEMP_BX6JKP7F_.DBF\pdb1_1_temp01.dbf          TEMP
SQL> EXIT
```

j. List the password file and SPFILE of orcl12c and cdb1 in Windows

```
c:\app\oracle\product\12.1.0\dbhome_1\database> dir
08\17\2015  02:59 PM    <DIR>          .
08\17\2015  02:59 PM    <DIR>          ..
07\15\2015  11:11 PM    <DIR>          archive
07\16\2015  10:41 PM             2,048 hc_cdb1.dat
07\16\2015  10:41 PM             2,048 hc_omsdb.dat
07\19\2015  10:12 PM             2,048 hc_orcl12c.dat
08\03\2015  08:33 AM             2,048 hc_pec1.dat
07\21\2015  04:17 AM             2,048 hc_pec12c.dat
12\22\2005  05:07 AM            31,744 oradba.exe
08\13\2015  03:29 AM             2,110 oradim.log
07\16\2015  10:47 PM             7,680 PWDcdb1.ora
07\16\2015  10:47 PM             7,680 PWDomsdb.ora
07\15\2015  11:21 PM             7,680 PWDorcl12c.ora
08\03\2015  08:40 AM             7,680 PWDpec1.ora
08\17\2015  08:47 AM             3,584 SPFILECDB1.ORA
08\17\2015  08:47 AM             3,584 SPFILEOMSDB.ORA
08\13\2015  04:01 AM             3,584 SPFILEORCL12C.ORA
08\13\2015  04:00 AM             3,584 SPFILEPEC1.ORA
              15 File(s)         89,150 bytes
               3 Dir(s)  495,468,797,952 bytes free
C:\app\oracle\product\12.1.0\dbhome_1\database>
```

NOTE: If we were using HP-UX for our labs, we could list the password file and SPFILE of orcl12c and cdb1 executing the following commands. The password and spfiles are in the dbs directory in HP-UX. In windows the database directory contains the spfiles and password files.

```
$ cd $ORACLE_HOME\dbs
$ dir orapw* spfile*
-rw-r----- 1 oracle oinstall 7680 Sep 5 10:43 orapwcdb1
-rw-r----- 1 oracle oinstall 7680 Sep 7 00:48 orapwcdb1
............
-rw-r----- 1 oracle oinstall 3584 Sep 7 01:23 spfilecdb1.ora
-rw-r----- 1 oracle oinstall 3584 Sep 7 01:08 spfilecdb1.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 c:\app\oracle\diag\rdbms\
 dir
cdb1 orcl12c pec1
 cd cdb1\cdb1\trace
 notepad alert_cdb1.log
Mon Feb 06 09:27:09 2012
Fri Sep 07 00:41:54 2012
create pluggable database PDB$SEED as clone using
'c:\app\oracle\product\12.1.0\dbhome_1assistants\dbca\templat
es\\pdbseed.xml' source_file_name_convert =
('\ade\b\3895122769\oracle\oradata\seeddata\pdbseed\temp01.dbf',
'c:\app\oracle\oradata\cdb1\pdbseed\pdbseed_temp01.dbf',
'\ade\b\3895122769\oracle\oradata\seeddata\pdbseed\system01.dbf'
,'c:\app\oracle\oradata\cdb1\pdbseed\system01.dbf',
'\ade\b\3895122769\oracle\oradata\seeddata\pdbseed\sysaux01.dbf'
,'c:\app\oracle\oradata\cdb1\pdbseed\sysaux01.dbf') NOCOPY
**************************************************************
Pluggable Database PDB$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
**************************************************************
…Post plug operations are now complete.
Pluggable database PDB$SEED with pdb id - 2 is now marked as NEW.
**************************************************************
Completed: create pluggable database PDB$SEED as clone using
'c:\app\oracle\product\12.1.0\dbhome_1assistants\dbca\templat
es\pdbseed.xml' source_file_name_convert =
('\ade\b\3895122769\oracle\oradata\seeddata\pdbseed\temp01.dbf',
'c:\app\oracle\oradata\cdb1\pdbseed\pdbseed_temp01.dbf',
'\ade\b\3895122769\oracle\oradata\seeddata\pdbseed\system01.dbf'
,'c:\app\oracle\oradata\cdb1\pdbseed\system01.dbf',
'\ade\b\3895122769\oracle\oradata\seeddata\pdbseed\sysaux01.dbf'
,'c:\app\oracle\oradata\cdb1\pdbseed\sysaux01.dbf') NOCOPY
alter pluggable database PDB$SEED open restricted
Pluggable database PDB$SEED dictionary check beginning
Pluggable Database PDB$SEED Dictionary check complete
```

6. List all users created in the new CDB `cdb1`.
a. Connect to `cdb1` instance.

```
$ sqlplus /@cdb1 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>
```

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' order by 1;
USERNAME
------------------------------
ANONYMOUS
APEX_040200
APEX_PUBLIC_USER
AUDSYS
CTXSYS
DBSNMP
.............
GSMUSER
ORDSYS
SYS
SYSBACKUP
SYSDG
SYSKM
SYSTEM
WMSYS
XDB
XS$NULL
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 6-3: Creating a PDB from Seed

Overview

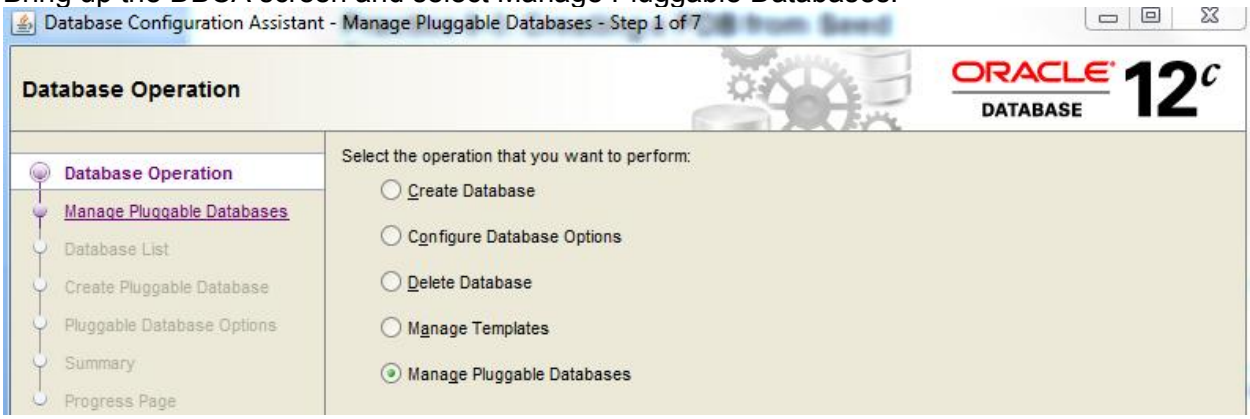
In this practice, you will create a new PDB `pdb2_1` in `cdb1` from seed.

Assumptions

The creation of the CDB `cdb1` is successful.

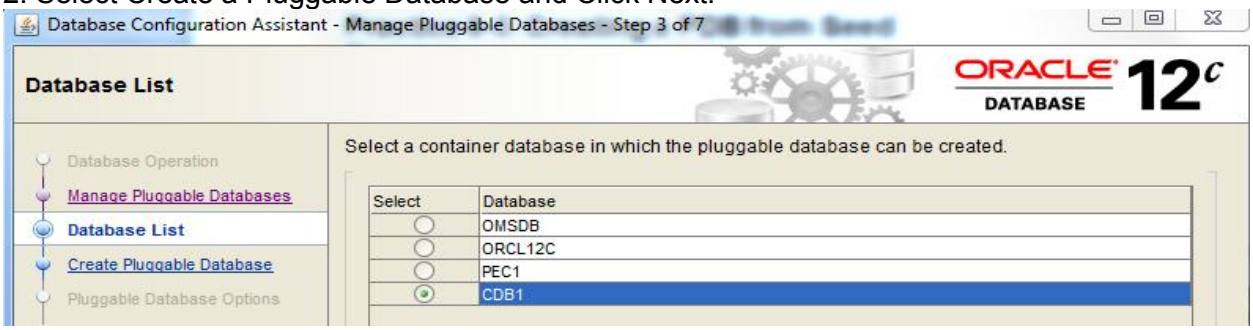
Tasks

Either use DBCA or SQL Developer or SQL commands.
The creation using DBCA is described below:
Bring up the DBCA screen and select Manage Pluggable Databases.

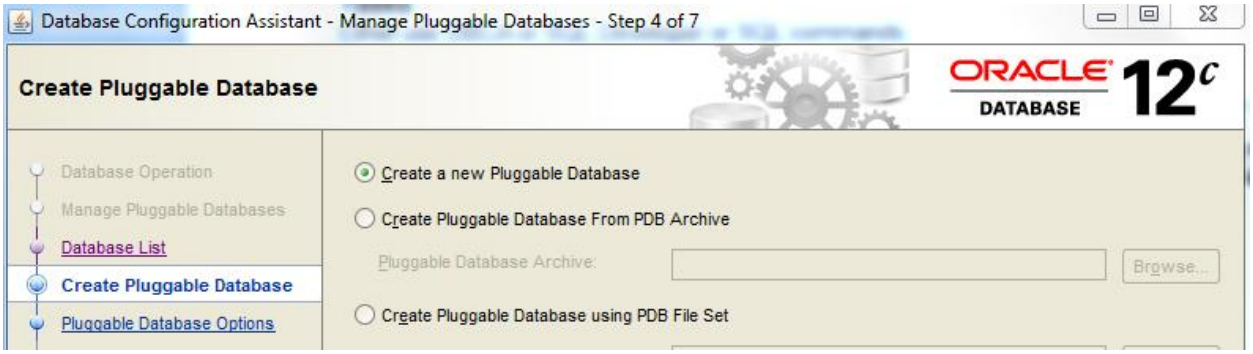


1. Click Next to Continue creation of a PDB.

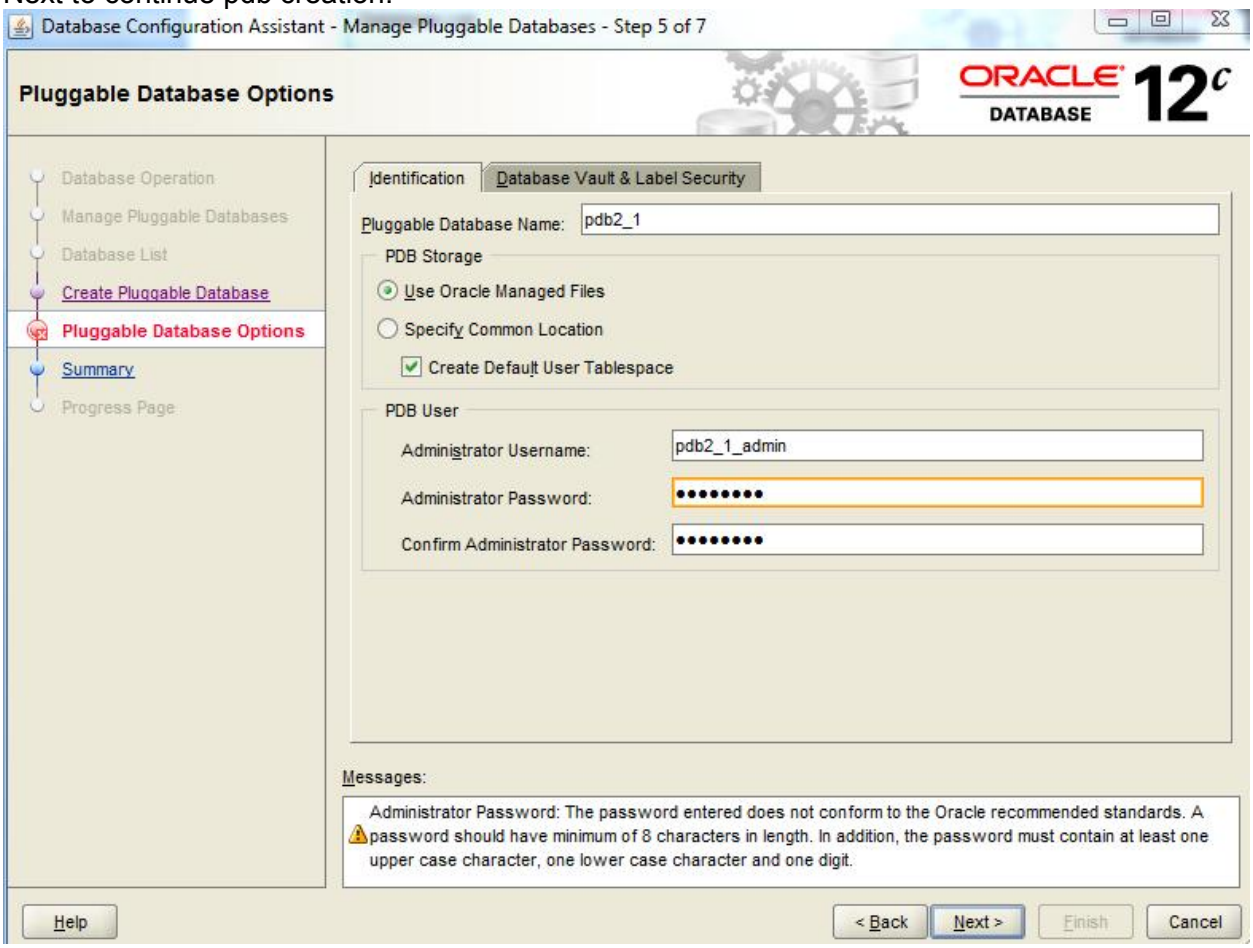
2. Select Create a Pluggable Database and Click Next.



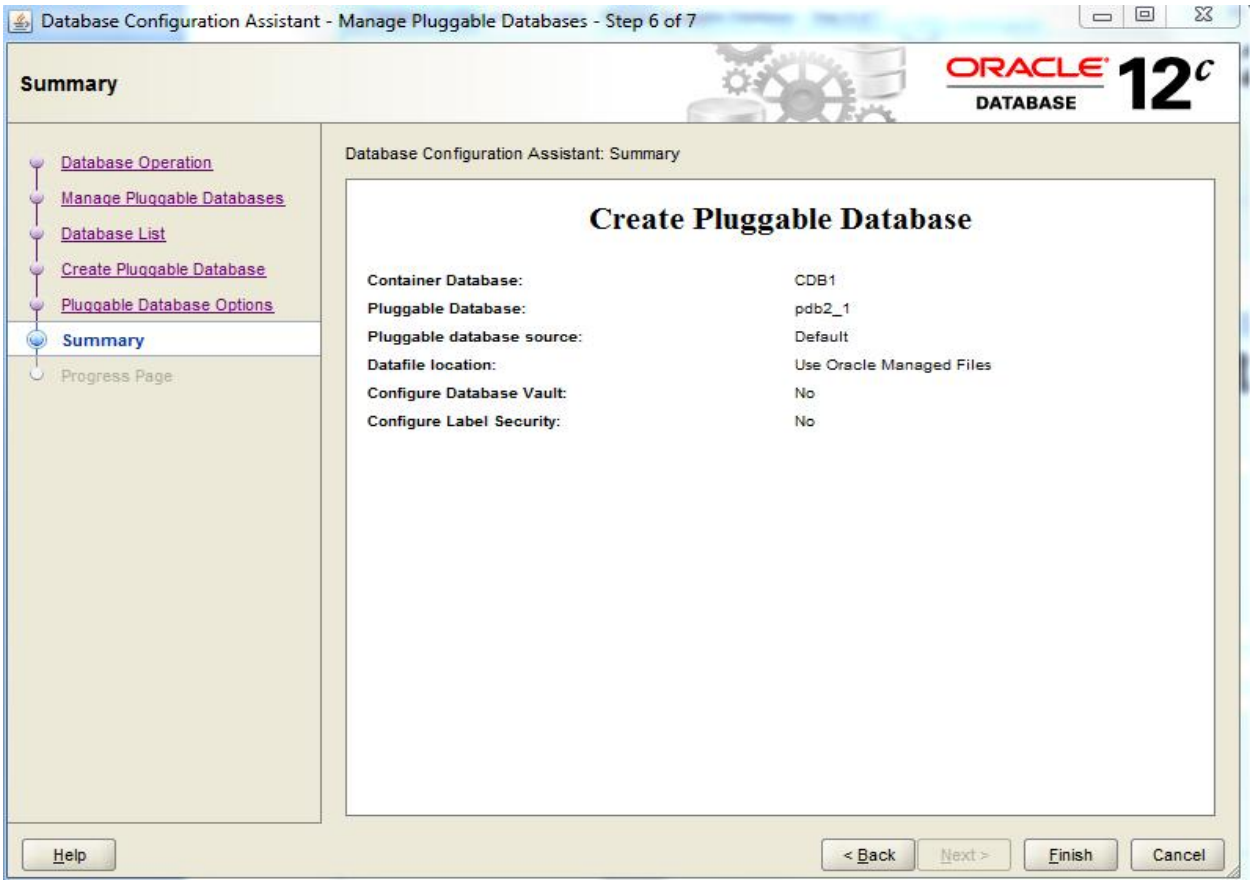
3. Select CDB1 as your container database for the new pluggable database to be created. Click Next.



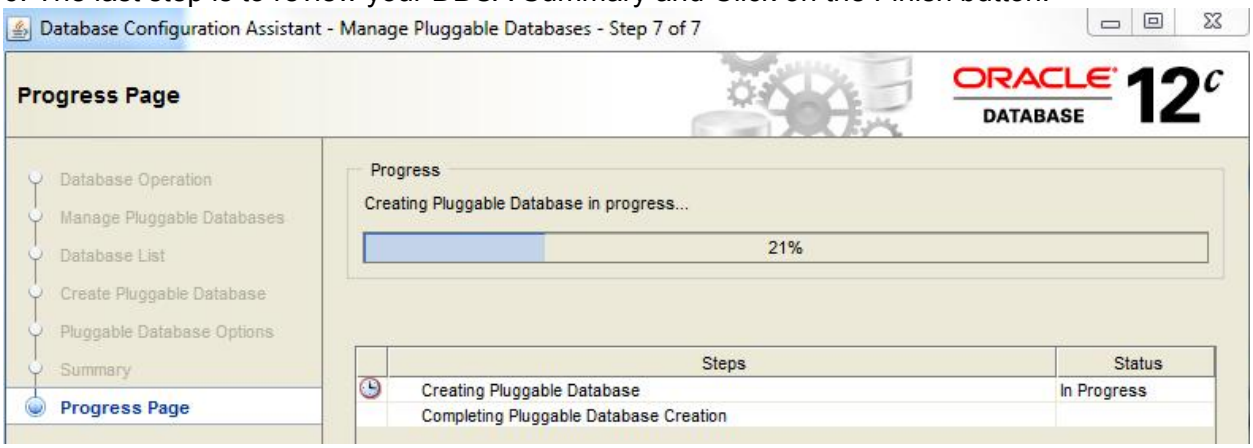
4. Select Create a New Pluggable database. You could also select to create a pluggable database from an Archive or from a PDB File Set that already exists such as PDB1_1. Click Next to continue pdb creation.



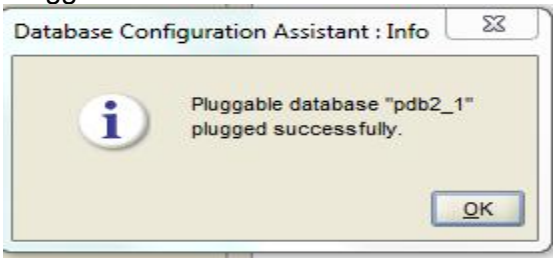
5. Select the appropriate PDB name, in our exercise PDB2_1 and identify an administrator for the new pluggable database called pdb2_1_admin with a password of password. CLick Next to continue.



6. The last step is to review your DBCA Summary and Click on the Finish button.



7. The Progress page will then be provided so the DBA can monitor the creation of the New Pluggable Database.



8. When complete a dialogue screen will appear to identify the database was successfully completed. The Pluggable database will automatically be opened with DBCA.

****

The creation using SQL is described below. This is good for all installations but because we used OMF the filenames are too long. When creating the initial container database the DBA must specify the locations without using Oracles OMF file creation.

1. Create a directory for the new data files of `pdb2_1` of `cdb1`.

```
cd c:\app\oracle\oradata\cdb1
c:\app\oracle\oradata\cdb1>  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 12.1.0.0.2 -
64bit Production
With the Partitioning, OLAP, Data Mining and Real Application
Testing options
SQL> CREATE PLUGGABLE DATABASE pdb2_1 ADMIN USER pdb2_1_admin
2 IDENTIFIED BY password ROLES=(CONNECT)
3 FILE_NAME_CONVERT=('c:\app\oracle\oradata\cdb1\pdbseed'
4 ,'c:\app\oracle\oradata\cdb1\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  4029890286 4029890286
3      PDB2_1               MOUNTED    3071827262 3071827262
SQL>
```

4. Open `pdb2_1`.

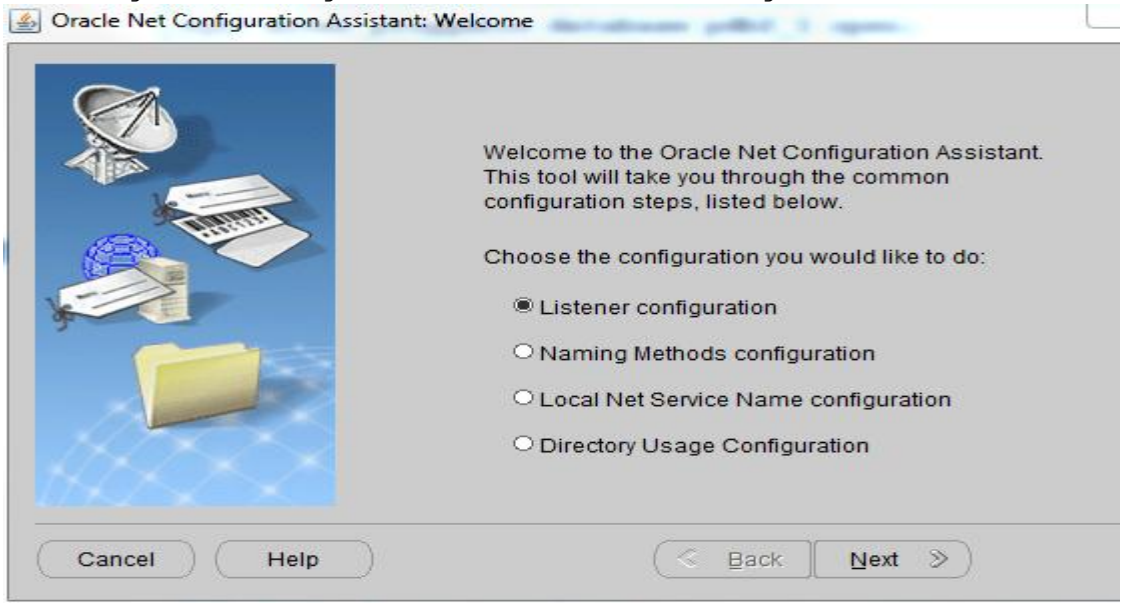
a. Open the PDB.

```
SQL> alter pluggable database pdb2_1 open;
Pluggable database altered.
SQL> EXIT
```

b. Connect to `pdb2_1 AS SYSDBA`.

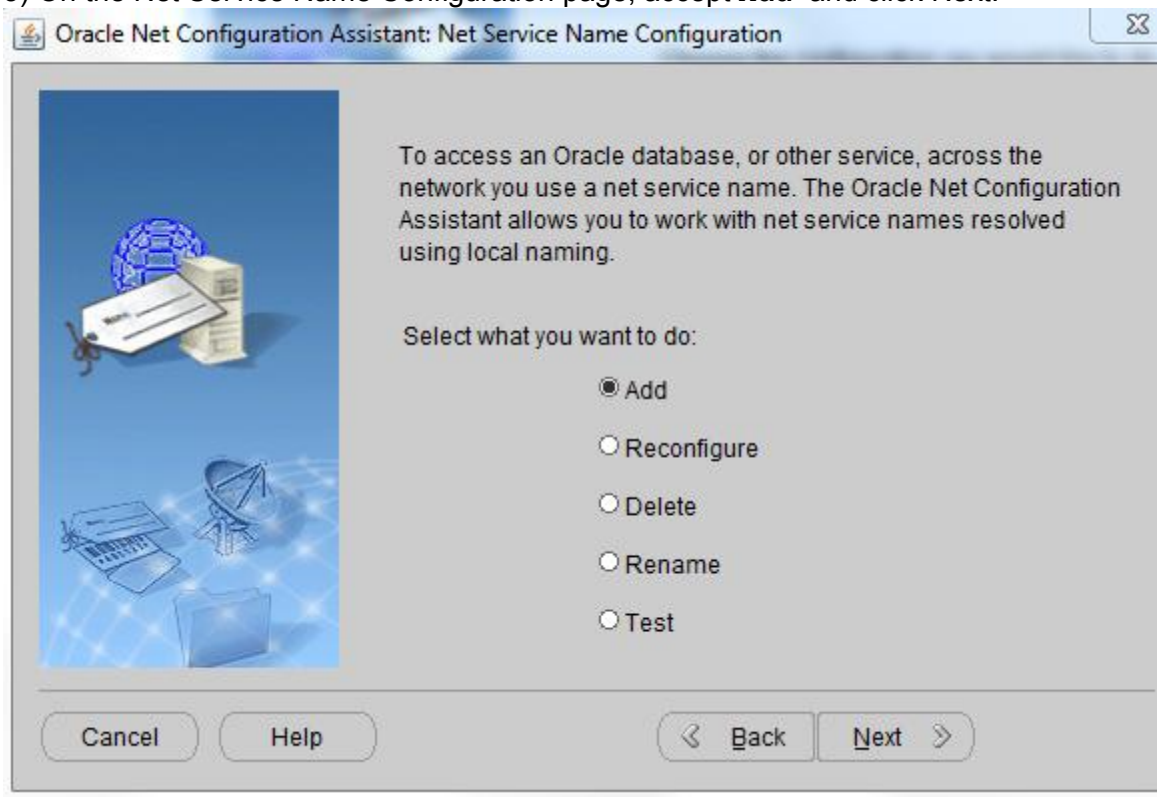
1) Use netca to add the `PDB2_1` net service name for `pdb2_1` pluggable database of `cdb1` in the `tnsnames.ora` file.

`From your desktop select START | All PROGRAMS | Oracle - OraDB12Home1 | Configuration Migration Tools | Net Configuration Assistant`

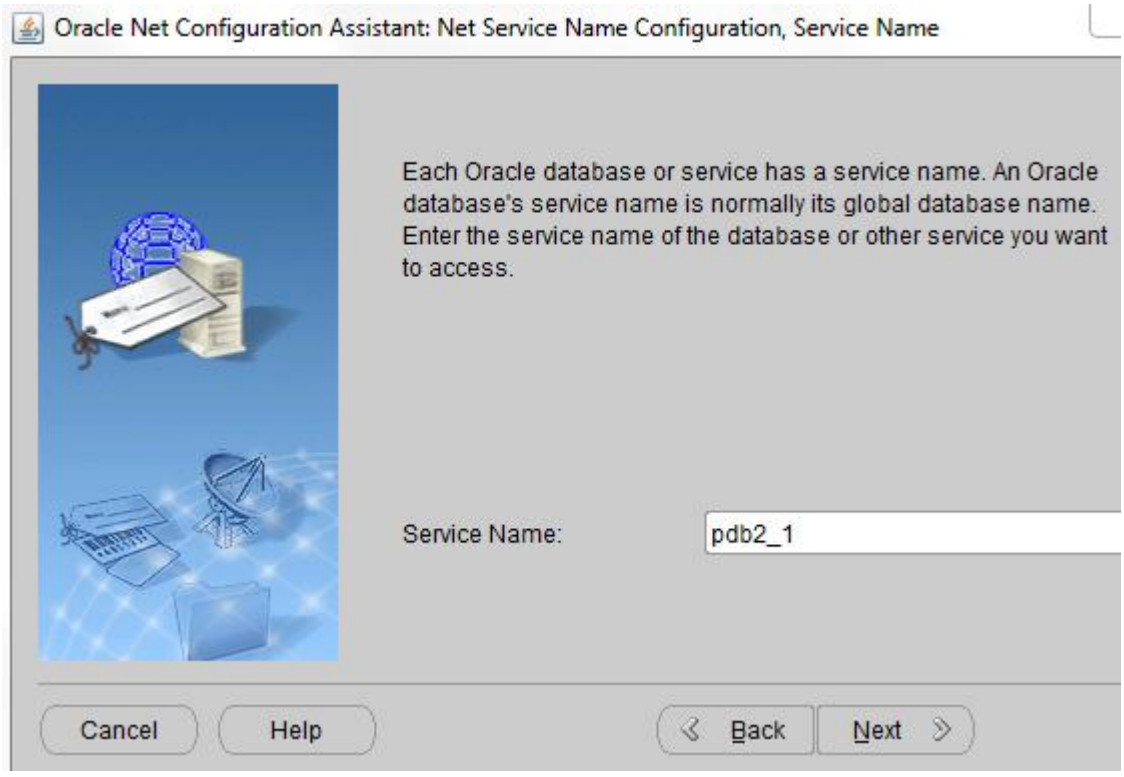


2) On the Welcome page, select the “Local Net Service Name configuration” and click Next.

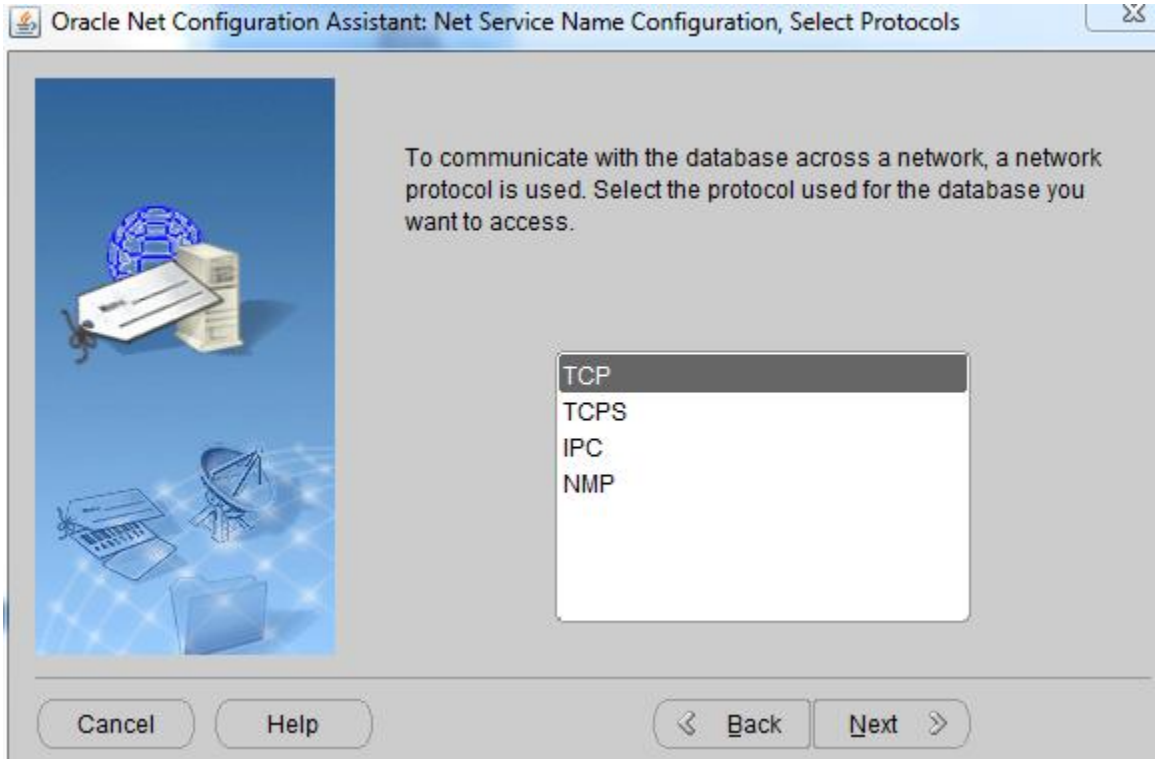
3) On the Net Service Name Configuration page, accept `Add` and click Next.



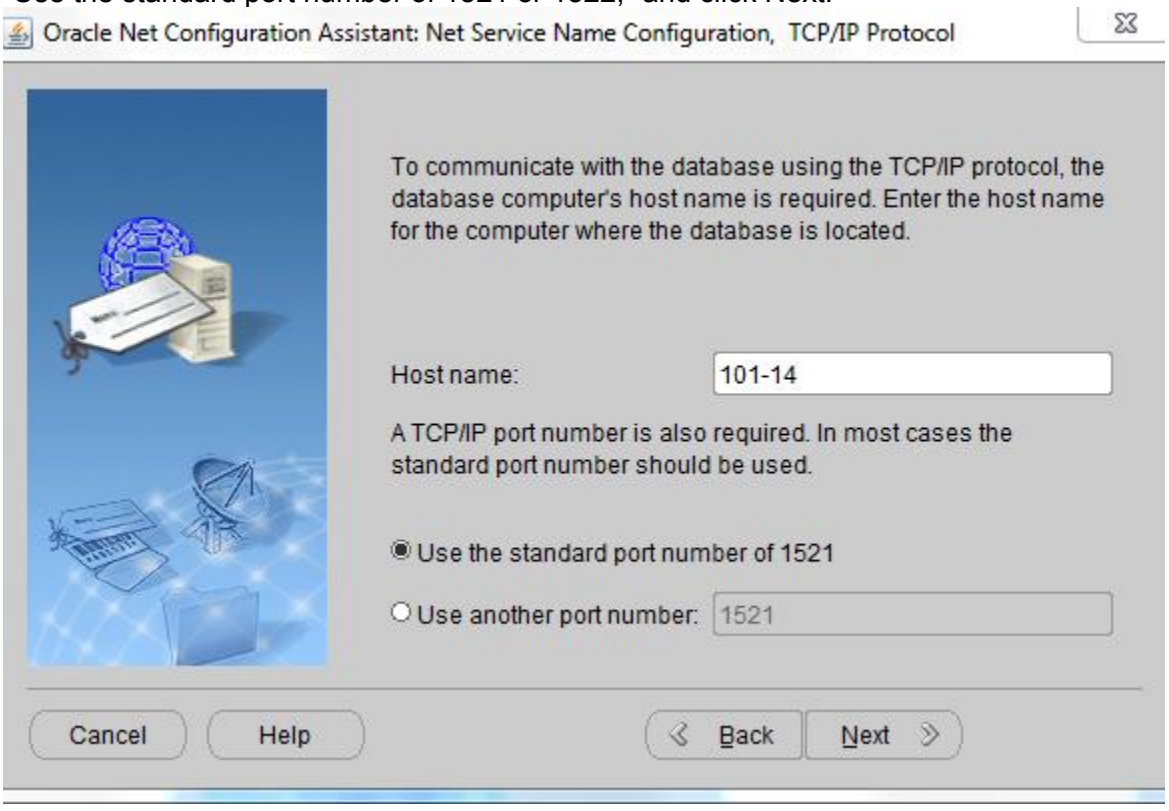
4) On the Net Service Name Configuration, Service Name page, enter `pdb2_1` as Service Name and click Next.



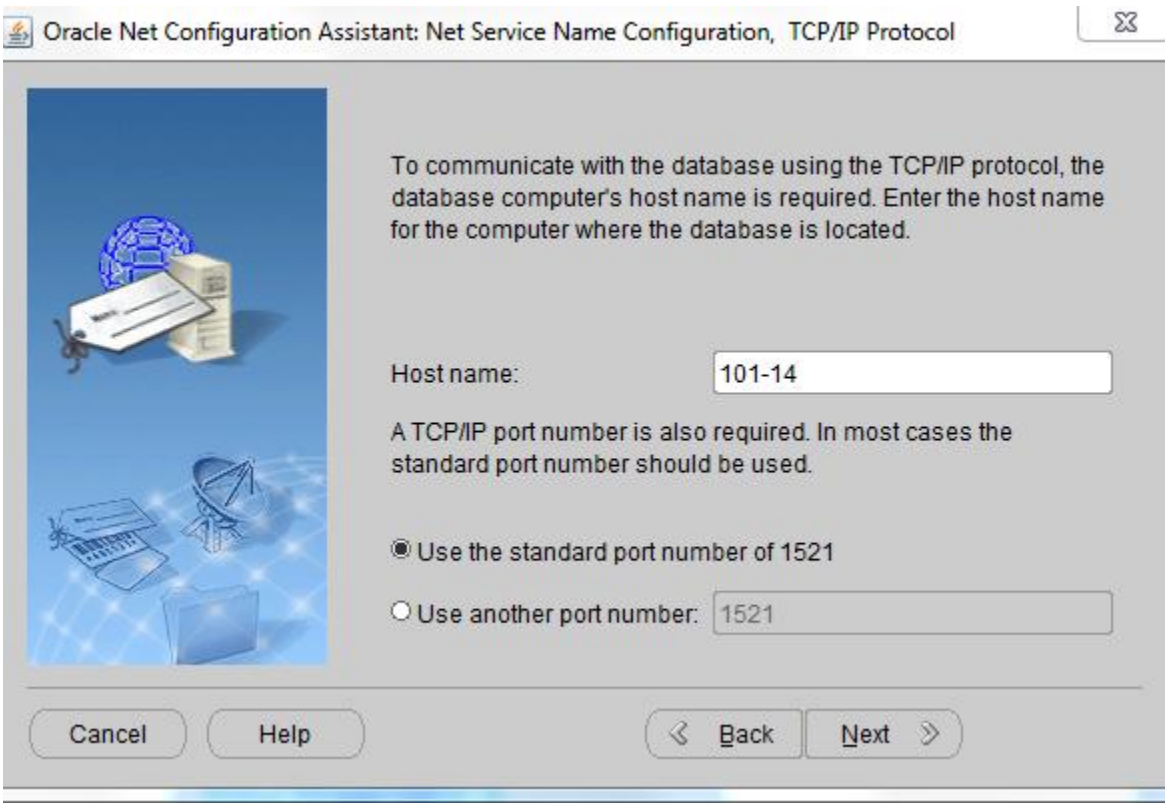
5) On the Net Service Name Configuration, Select Protocols page, select TCP and click Next.



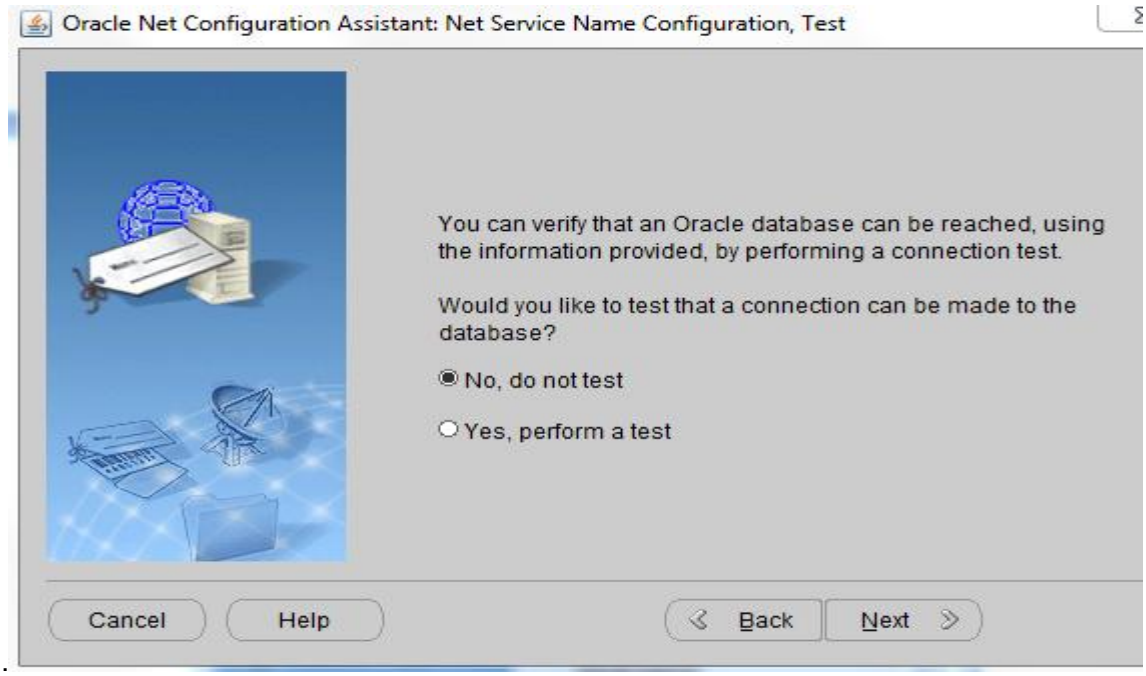
6) On the Net Service Name Configuration, TCP\IP Protocol page, enter your complete host name, for example, *<yourservername>*, or *localhost*, accept "Use the standard port number of 1521 or 1522," and click Next.



.

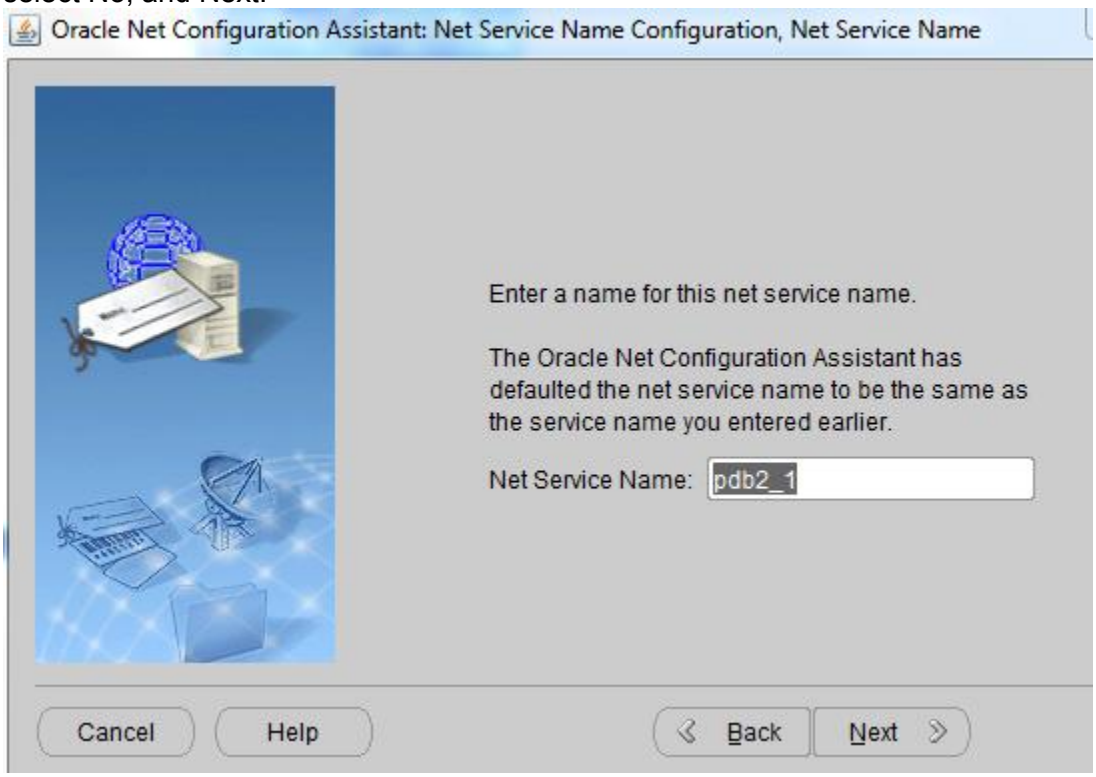


7) On the Net Service Name Configuration, Test page, select “No, do not test” (the pluggable database is not yet opened nor is the scott user unlocked) and click Next



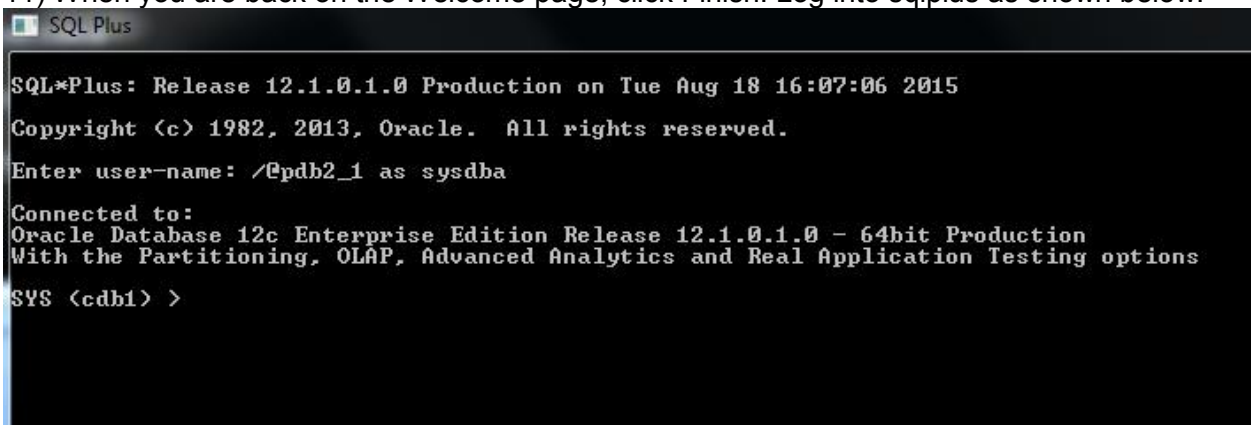
.

9) On the Net Service Name Configuration, Net Service Name page, accept `pdb2_1` as Net Service Name and click Next t Service Name Configuration, Another Net Service Name page, select No, and Next.



10) On the Net Service Name Configuration Done page, click Next.

11) When you are back on the Welcome page, click Finish. Log into sqlplus as shown below.



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

```
SQL> host
c:\ora12clabs> lsnrctl status
The command completed successfully
LSNRCTL for Linux: Version 12.1.0.0.2 - Production on 07-SEP-
2012 01:47:28
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 Windows: Version 12.1.0.0.2 - Production
Start Date 14-SEP-2012 03:04:56
Uptime 16 days 22 hr. 0 min. 0 sec
Trace Level off
Security ON: Local OS Authentication
SNMP OFF
Listener Parameter File
c:\app\oracle\product\12.1.0\dbhome_1network\admin\listener.o
ra
Listener Log File
c:\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=1522)
))
(DESCRIPTION=(ADDRESS=(PROTOCOL=tcp)(HOST=yourserver)(PORT=5501)
)(Presentation=HTTP)(Session=RAW))
(DESCRIPTION=(ADDRESS=(PROTOCOL=tcp)(HOST=yourserver)(PORT=5502)
)(Presentation=HTTP)(Session=RAW))
(DESCRIPTION=(ADDRESS=(PROTOCOL=tcp)(HOST=yourserver)(PORT=5500)
)(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 "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 "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 "cdb1", 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/password@101-14:1521\pdb2_1 AS SYSDBA
Connected.
SQL> connect pdb2_1_admin\password@PDB2_1
Connected.
SQL> show con_name
CON_NAME
------------------------------
PDB2_1
SQL>
```

7. List the data files created.

```
SQL> host

C:\app\oracle\oradata\orcl12c\pdborcl>dir
 Volume in drive C is Windows
 Volume Serial Number is 967E-5726
 Directory of C:\app\oracle\oradata\cdb1\pdb2_1

07/15/2015  11:24 PM    <DIR>          .
07/15/2015  11:24 PM    <DIR>          ..
08/13/2015  03:29 AM       375,529,472 EXAMPLE01.DBF
08/12/2015  10:00 PM       614,473,728 PDB$SEED_TEMP01.DBF
08/13/2015  03:29 AM         5,251,072 SAMPLE_SCHEMA_USERS01.DBF
08/13/2015  03:29 AM       671,096,832 SYSAUX01.DBF
08/13/2015  03:29 AM       408,952,832 SYSTEM01.DBF
               5 File(s)  2,075,303,936 bytes
               2 Dir(s)  495,203,090,432 bytes free
SQL>
```

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

```
SQL> connect system\PASSWORD@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                                          TS_NAME  FILE_ID CON_ID
-------------------------------------------------- -------- ------- ------
c:\app\oracle\oradata\cdb1\pdb2_1\sysaux01.dbf     SYSAUX     9     3
c:\app\oracle\oradata\cdb1\pdb2_1\system01.dbf     SYSTEM     8     3
SQL> select FILE_NAME, TABLESPACE_NAME, FILE_ID
```

```
2 from dba_data_files;
FILE_NAME                                          TABLESPANAME FILE_ID
-------------------------------------------------- ----------- ------
c:\app\oracle\oradata\cdb1\pdb2_1\system01.dbf      SYSTEM        8
c:\app\oracle\oradata\cdb1\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                                            TABLESPACE NAME FILE_ID
---------------------------------------------------  -------------- -------
c:\app\oracle\oradata\cdb1\pdb2_1\pdbseed_temp01.dbf  TEMP              3

SQL> select FILE_NAME, TABLESPACE_NAME, FILE_ID
2 from dba_temp_files;
FILE_NAME                                             TS_NAME    FILE_ID
----------------------------------------------------  -------    -------
c:\app\oracle\oradata\cdb1\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
cdb1XDB
cdb1
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                                          TS_NAME  FILE_ID CON_ID
-------------------------------------------------- -------- ------  ------
c:\app\oracle\oradata\cdb1\system01.dbf             SYSTEM   1          1
c:\app\oracle\oradata\cdb1\sysaux01.dbf             SYSAUX   3          1
c:\app\oracle\oradata\cdb1\undotbs01.dbf            UNDOTBS1 4          1
```

```
c:\app\oracle\oradata\cdb1\users01.dbf             USERS    6          1
c:\app\oracle\oradata\cdb1\pdbseed\system01.dbf    SYSTEM   5          2
c:\app\oracle\oradata\cdb1\pdbseed\sysaux01.dbf    SYSAUX   7          2
c:\app\oracle\oradata\cdb1\pdb2_1\system01.dbf     SYSTEM   8          3
c:\app\oracle\oradata\cdb1\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                                        TABLESPASPACE_NAME  FILE_ID
-----------------------------------------------  ------------------  -------
c:\app\oracle\oradata\cdb1\users01.dbf           USERS                 6
c:\app\oracle\oradata\cdb1\undotbs01.dbf         UNDOTBS1              4
c:\app\oracle\oradata\cdb1\sysaux01.dbf          SYSAUX                3
c:\app\oracle\oradata\cdb1\system01.dbf          SYSTEM                1
SQL> select FILE_NAME, TABLESPACE_NAME, FILE_ID
2 from cdb_temp_files;
FILE_NAME                                                TABLESPACE_NAME FILE_ID
-------------------------------------------------------  --------------  ------
c:\app\oracle\oradata\cdb1\temp01.dbf                    TEMP            1
c:\app\oracle\oradata\cdb1\pdbseed\pdbseed_temp01.dbf    TEMP            2
c:\app\oracle\oradata\cdb1\pdb2_1\pdbseed_temp01.dbf     TEMP            3
SQL> select FILE_NAME, TABLESPACE_NAME, FILE_ID
2 from dba_temp_files;
FILE_NAME                                        TABLESPACE_NAME FILE_ID
-----------------------------------------------  --------------- -------
c:\app\oracle\oradata\cdb1\temp01.dbf            TEMP             1
SQL> EXIT
```

Practice 6-4: 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 `cdb1`.

Assumptions

The `pdb2_1` creation has completed successfully in Practice 8-3.

Tasks

Either use the SQL commands OR SQL Developer. Below is the Method with SQL*Plus.

1. Create a directory for the new data files of `pdb2_2` of `cdb1`.

```
c:\ora12clabs> cd C:\app\oracle\oradata\cdb1
$ mkdir pdb2_2
```

2. Run SQL*Plus and connect as a user granted with `CREATE PLUGGABLE DATABASE` privilege. Connect as sys in the cdb1 container database.

a. Set `pdb2_1` in `READ ONLY` open mode before cloning.

```
$ sqlplus \ as sysdba
Connected to:
Oracle Database 12c Enterprise Edition Release 12.1.0.0.2 -
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 'c:\app\oracle\oradata\cdb1\pdb2_2'.

```
SQL> alter system set db_create_file_dest =
'c:\app\oracle\oradata\cdb1\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 cdb1 in the tnsnames.ora file.
Start | All Programs | Oracle - OraDB12Home1 | Configuration Migration Tools | Net Configuration Assistant
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 1522," 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 Name page, select No, and Next.
10) On the Net Service Name Configuration Done page, click Next.
11) When you are back on the Welcome page, click Finish.
From your desktop select **sqlplus with user id sys\password@pdb2_2 AS SYSDBA**

```
Connected to:
Oracle Database 12c Enterprise Edition Release 12.1.0.0.2 -
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/password@PDB2_2
Connected.
SQL> show con_name
PDB2_2
SQL> EXIT
```

6. List the data files created.

```
C:\ora11glabs>cd c:\app\oracle\oradata\cdb1
c:\app\oracle\oradata\CDB1>dir
 Volume in drive C is Windows
 Volume Serial Number is 967E-5726

 Directory of c:\app\oracle\oradata\CDB1

08/19/2015  09:18 AM   <DIR>       .
08/19/2015  09:18 AM   <DIR>       ..
08/18/2015  02:42 PM   <DIR>       9925B47F62FE4AD488E133D230DC3C0C
08/18/2015  09:33 AM   <DIR>       9E03ED8956664A0B82A5360632CAEDA7
08/18/2015  09:26 AM   <DIR>       CONTROLFILE
08/18/2015  09:27 AM   <DIR>       DATAFILE
08/18/2015  09:26 AM   <DIR>       ONLINELOG
08/18/2015  01:39 PM   <DIR>       pdb2_1
08/19/2015  09:33 AM   <DIR>       pdb2_2
08/18/2015  02:55 PM   <DIR>       pdb3_1
         0 File(s)          0 bytes
        10 Dir(s)  489,793,118,208 bytes free

c:\app\oracle\oradata\CDB1>cd pdb2_2

c:\app\oracle\oradata\CDB1\pdb2_2>dir
 Volume in drive C is Windows
 Volume Serial Number is 967E-5726

 Directory of c:\app\oracle\oradata\CDB1\pdb2_2

08/19/2015  09:33 AM    <DIR>          .
08/19/2015  09:33 AM    <DIR>          ..
08/19/2015  09:33 AM    <DIR>          CDB1
               0 File(s)              0 bytes
```

```
               3 Dir(s)  489,143,463,936 bytes free
c:\app\oracle\oradata\CDB1\pdb2_2>cd CDB1

c:\app\oracle\oradata\CDB1\pdb2_2\CDB1>dir
 Volume in drive C is Windows
 Volume Serial Number is 967E-5726

 Directory of c:\app\oracle\oradata\CDB1\pdb2_2\CDB1

08/19/2015  09:33 AM    <DIR>          .
08/19/2015  09:33 AM    <DIR>          ..
08/19/2015  09:33 AM    <DIR>
EE2D35BFF54C454599D253471A500150
               0 File(s)              0 bytes
               3 Dir(s)  489,143,373,824 bytes free


c:\app\oracle\oradata\CDB1\pdb2_2\CDB1>cd EE*.

c:\app\oracle\oradata\CDB1\pdb2_2\CDB1\EE2D35BFF54C454599D253471A50015
0>dir
 Volume in drive C is Windows
 Volume Serial Number is 967E-5726

 Directory of
c:\app\oracle\oradata\CDB1\pdb2_2\CDB1\EE2D35BFF54C454599D253471A500150

08/19/2015  09:33 AM    <DIR>          .
08/19/2015  09:33 AM    <DIR>          ..
08/19/2015  09:34 AM    <DIR>          DATAFILE
               0 File(s)              0 bytes
               3 Dir(s)  489,143,361,536 bytes free
c:\app\oracle\oradata\CDB1\pdb2_2\CDB1\EE2D35BFF54C454599D253471A50015
0>cd datafile

c:\app\oracle\oradata\CDB1\pdb2_2\CDB1\EE2D35BFF54C454599D253471A50015
0\DATAFILE>dir
 Volume in drive C is Windows
 Volume Serial Number is 967E-5726

 Directory of
c:\app\oracle\oradata\CDB1\pdb2_2\CDB1\EE2D35BFF54C454599D253471A50015
0\DATAFILE

08/19/2015  09:34 AM    <DIR>          .
08/19/2015  09:34 AM    <DIR>          ..
08/19/2015  09:53 AM       650,125,312 O1_MF_SYSAUX_BX94YM9J_.DBF
08/19/2015  09:53 AM       262,152,192 O1_MF_SYSTEM_BX94YM9H_.DBF
08/19/2015  09:34 AM        20,979,712 O1_MF_TEMP_BX94ZHH0_.DBF
08/19/2015  09:53 AM         5,251,072 O1_MF_USERS_BX94YM9G_.DBF
               4 File(s)    938,508,288 bytes
               2 Dir(s)  489,143,345,152 bytes free
```

Method with SQL Developer: (Optional)

1. If you already created pdb2_2 with SQL*Plus and would like to test the creation with SQL Developer, you first have to drop `pdb2_2` to recreate it.

a. Drop the pluggable database `pdb2_2`.

From your desktop click on **sqlplus and login as SYS**

```
SQL> / AS SYSDBA
Connected to:
Oracle Database 12c Enterprise Edition Release 12.1.0.0.2 -
64bit Production
With the Partitioning, OLAP, Advanced Analytics and Real
Application Testing options
SQL> ALTER PLUGGABLE DATABASE pdb2_2 CLOSE IMMEDIATE;
Pluggable database altered.
SQL> DROP PLUGGABLE DATABASE pdb2_2 INCLUDING DATAFILES;
Pluggable database dropped.
SQL> EXIT
```

b. Remove the directory.- In WIndows delete the files inside the pdb2_2 directory.

```
c:\ora12clabs> del c:\app\oracle\oradata\cdb1\pdb2_2
OR use your GUI tool (computer icon on desktop) to move to the
location and delete
```

2. Create a directory for the new data files of `pdb2_2` of `cdb1`.

```
c:\ora12clabs>  cd c:\app\oracle\oradata\cdb1
c:\ora12clabs> mkdir pdb2_2
```

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

From your desktop, click on your **sqlplus icon and log in as SYS**

```
SQL> \ AS SYSDBA
Connected to:
Oracle Database 12c Enterprise Edition Release 12.1.0.0.2 -
64bit Production
With the Partitioning, OLAP, Advanced Analytics and Real
Application Testing options
SQL> alter system set db_create_file_dest =
'c:\app\oracle\oradata\cdb1\pdb2_2' scope=both;
System altered.
SQL> EXIT
```

4. Launch SQL Developer.

From desktop, double click on the sqldeveloper icon

5. Create a connection as `SYS` in `cdb1`.

6. Open a connection as `SYS` in `cdb1`.

a. Choose the View option.

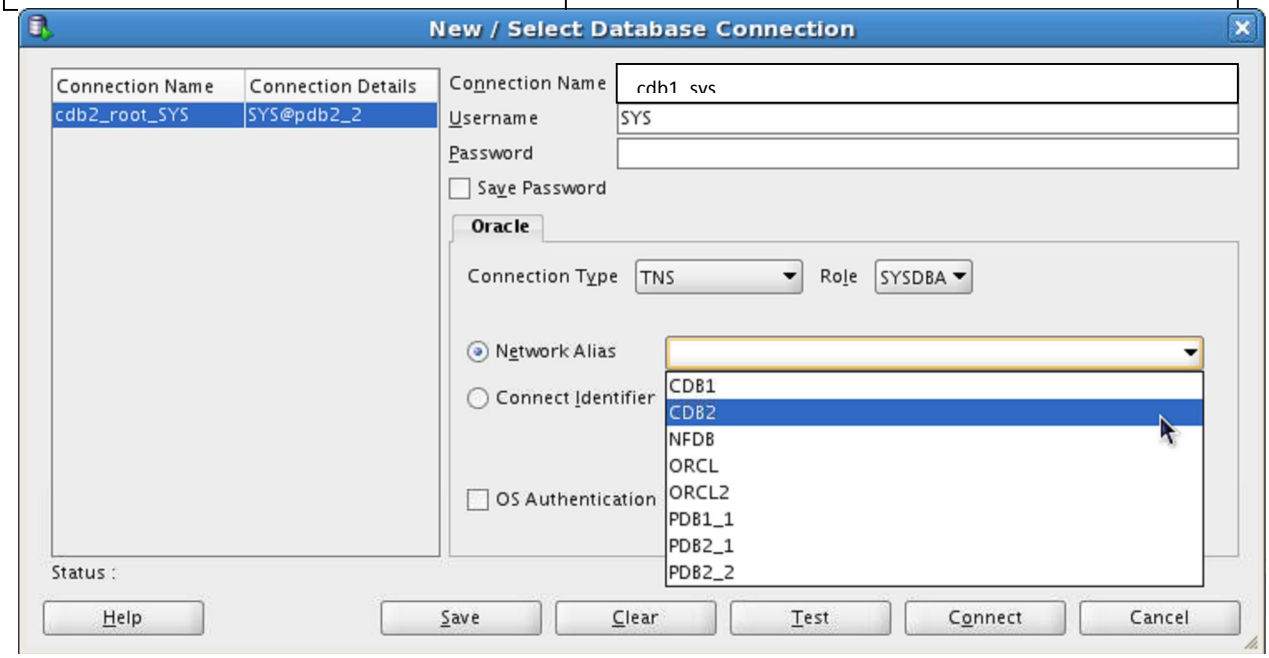
b. Click Connections.

c. Click + in the left Connections pane to add a new connection.

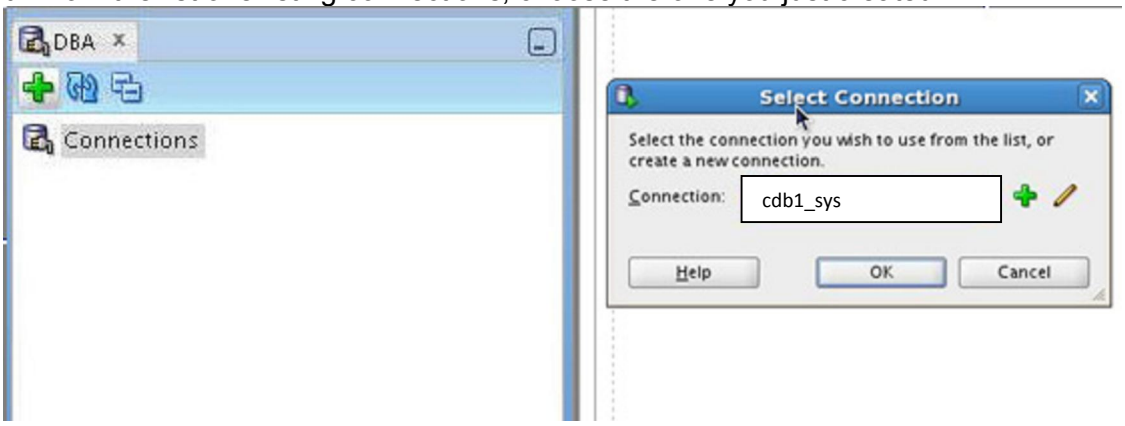
d. Fill the different fields as follows: be sure to change the host name and port number to your assigned host name and port number

.

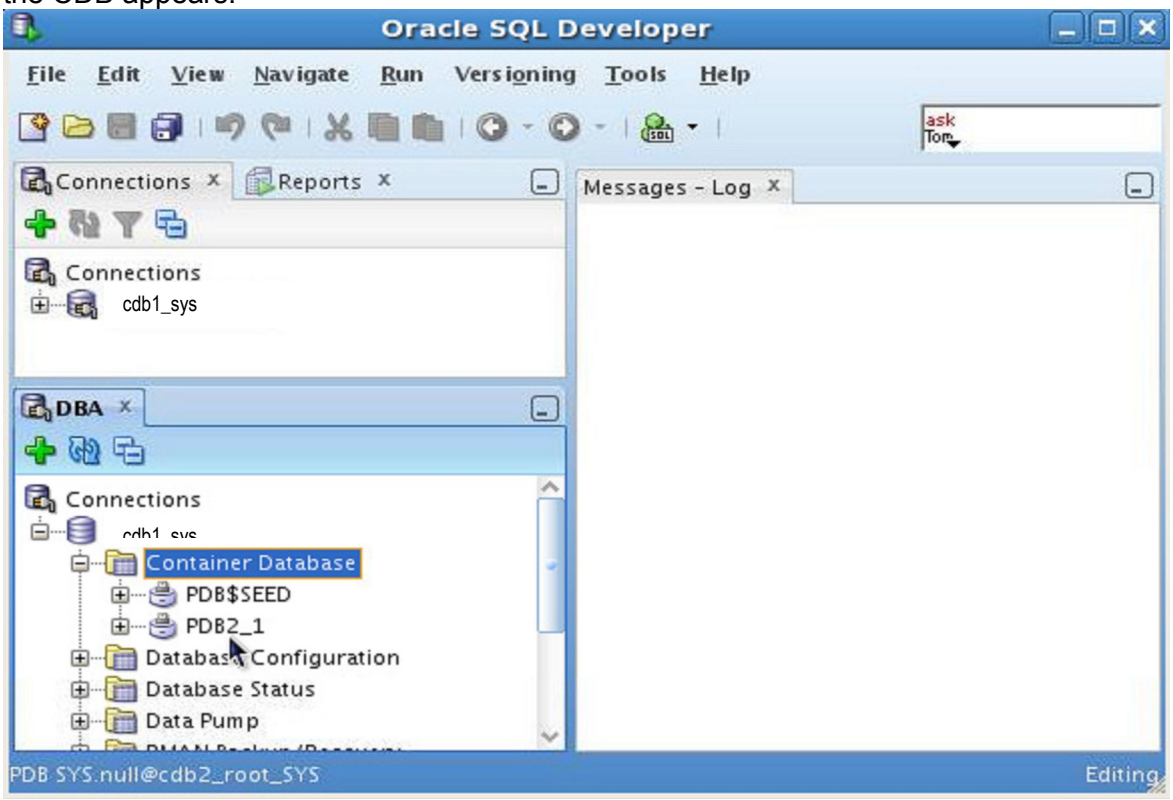
Window\Page Description	Choices or Values
Connection Name	cdb1_SYS
Username	sys
Password	password
Connection Type	TNS
Role	SYSDBA
Network Alias	cdb1



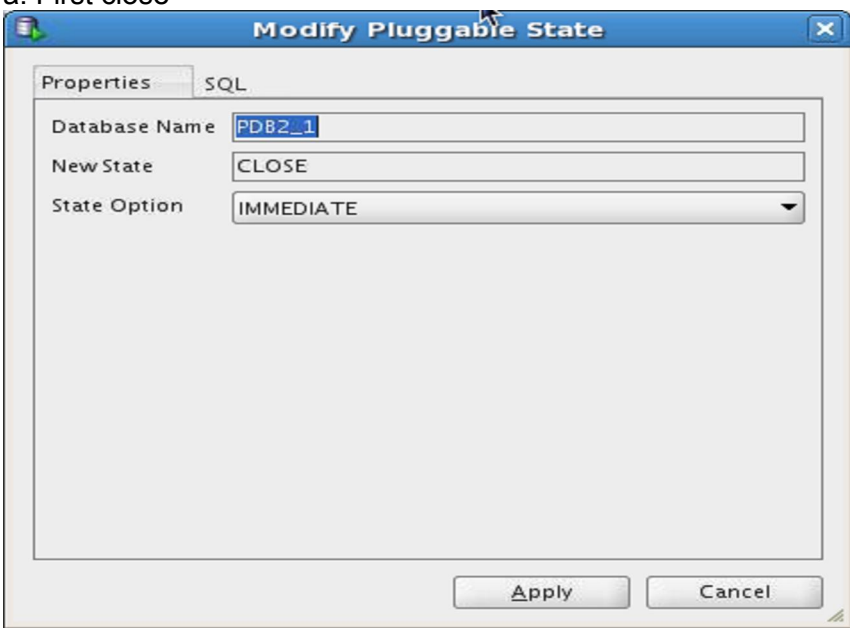
7. To manage the CDB and its PDBs:
a. Choose the View option.
b. Click DBA.
c. Click + in the left Connections pane to view an existing connection.
d. From the list of existing connections, choose the one you just created.



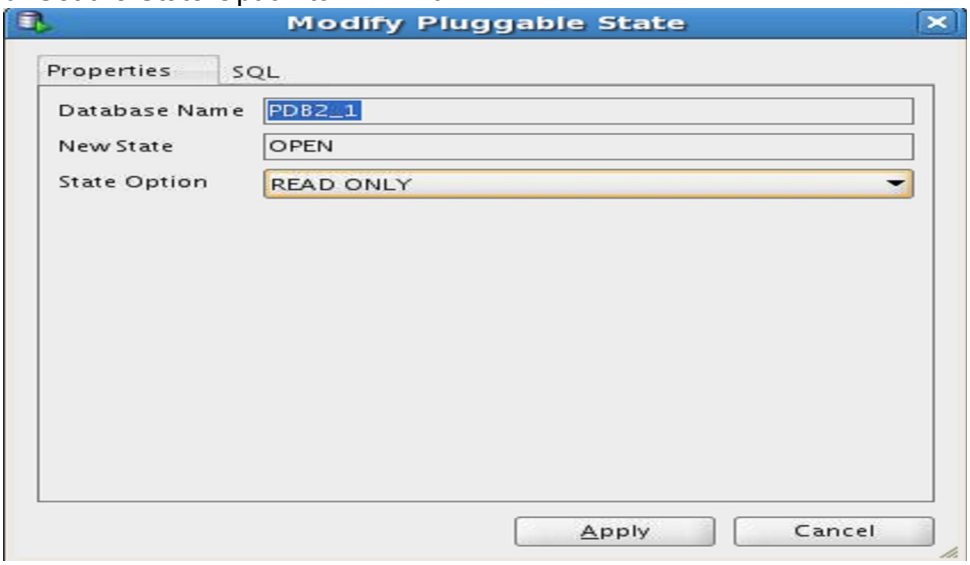
e. Click OK.
f. Click the sign + in front of the name of the `cdb1_SYS` connection to expand the folder. Then click the sign + in front of “Container Database”. The list of containers in the CDB appears.



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



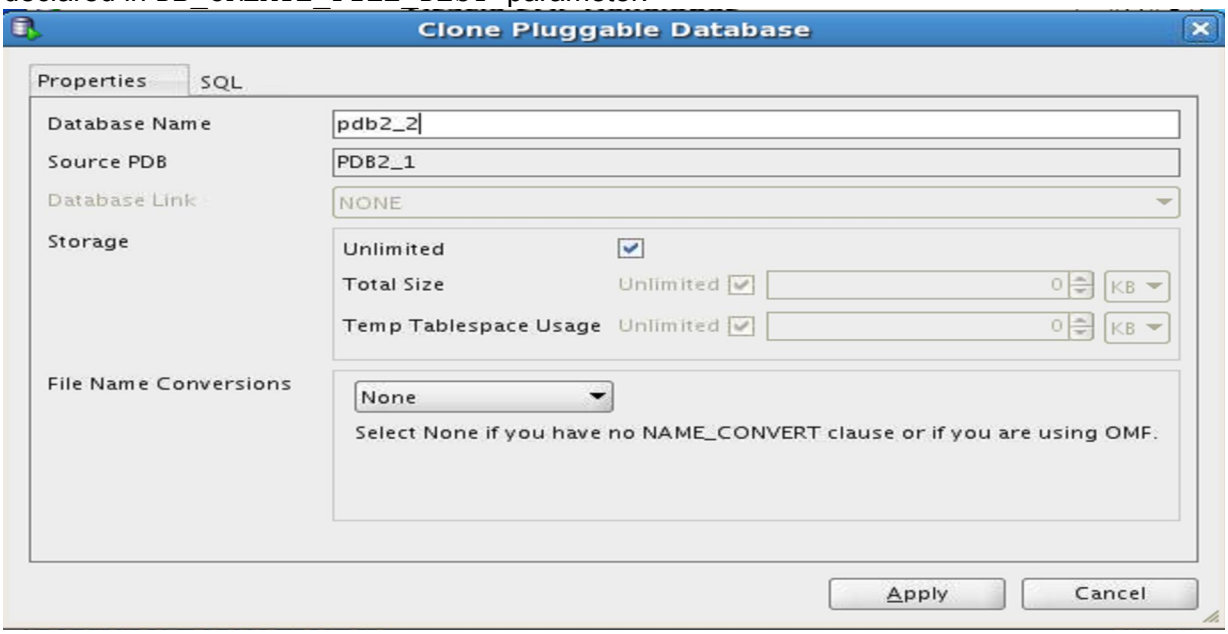
b. Click Apply.
c. Choose Modify State again.
d. Set the State Option to `READ ONLY`.



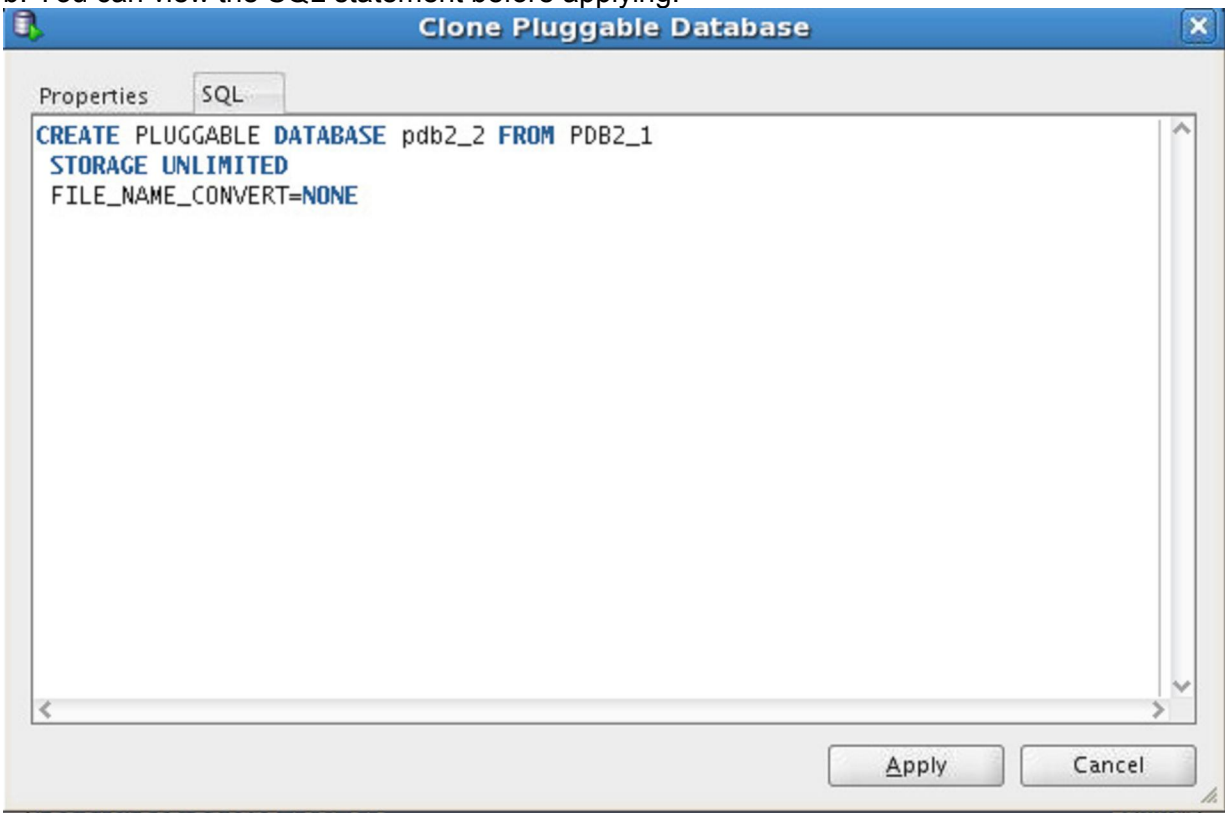
e. Click Apply then OK.
9. Right-click the pluggable database `pdb2_1` and choose Clone Pluggable Database….
a. Fill the different fields as follows.

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.

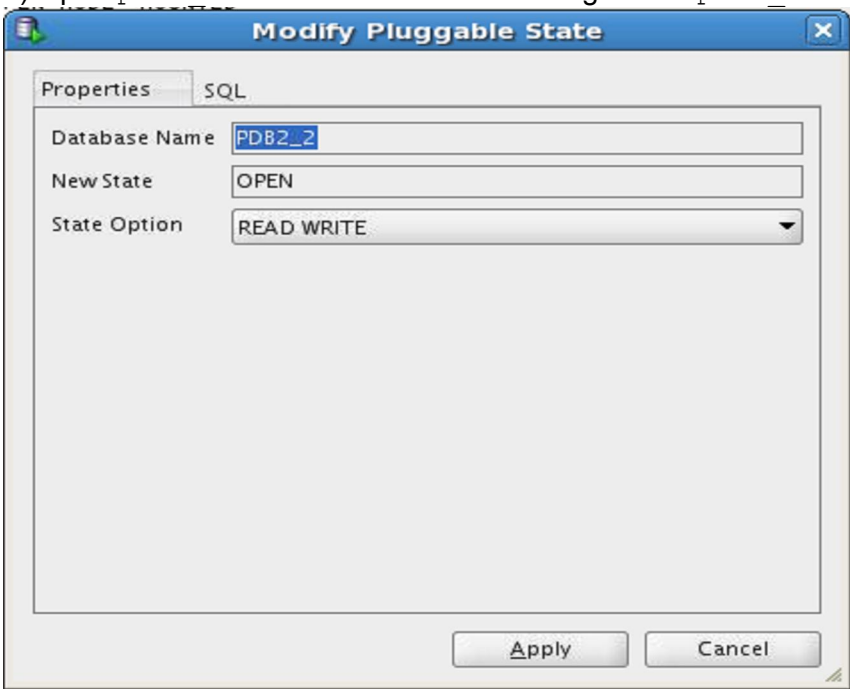


b. You can view the SQL statement before applying.



c. Click Apply then OK. The new `pdb2_2` appears in the list of PDBs in `cdb1`.
d. Open `pdb2_2`.

1) Open `pdb2_2` in `READ WRITE` mode. Right click `pdb2_2` and click Modify State.



2) Click Apply, then OK.
e. Open `pdb2_1` in `READ WRITE` mode. Right click `pdb2_1` and click Modify State. Click Apply , then OK.
10. Leave SQL Developer.
a. Click File.
b. Then click exit.

Practice 6-5: Plugging a Non-CDB into a CDB

Overview

In this practice, you will plug the non-CDB `pec1` into the CDB `cdb1`. You will not use Export\ Import DataPump, which can be a possible method, but the method with `DBMS_PDB` package. This package executed in the non-CDB `pec1` generates an XML file describing the tablespaces and data files of non-CDB `pec1`. The XML file is then used when creating `pdb_pec1` in `cdb1`.

Tasks

1. Use `DBMS_PDB.DESCRIBE` to "unplug" non-CDB `pec1`.

```
$ 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
('c:\app\oracle\oradata\pec1\xmlpec1.xml')
PL\SQL procedure successfully completed.
SQL> shutdown immediate
Database closed.
Database dismounted.
ORACLE instance shut down.
SQL> EXIT
c:\ora12clabs
```

2. Create a new PDB `pdb_pec1` to plug non-CDB `pec1` into `cdb1` 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.

```
$ 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> create pluggable database PDB_pec1 using
'c:\app\oracle\oradata\pec1\datafile\xmlpec1.xml' NOCOPY;
create pluggable database PDB_pec1 using 'xmlpec1' NOCOPY
*
ERROR at line 1:
ORA-01119: error in creating database file
'c:\app\oracle\oradata\pec1\datafile\01_MF_TEMP_BVYVRWFK_.TMP
ORA-27038: created file already exists
```

```
Additional information: 1
SQL>DEL c:\app\oracle\oradata\pec1\datafile\01_MF_TEMP_BVYVRWFK_.TMP
SQL>
SQL> create pluggable database PDB_pec1 using
'c:\app\oracle\oradata\PEC1\datafile\xmlpec1.xml' NOCOPY;
Pluggable database created.
SQL> EXIT
c:\ora12clabs>
```

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 `c:\app\oracle\product\12.1.0\dbhome_1\rdbms\admin\noncdb_to_pdb.sql` script whilst connected to the PDB.
a. Connect to `pdb2_2` as `SYSDBA`.
1) Use netca to add the `PDB_ORCL2` net service name for `pdb_orcl2` pluggable database of `cdb1` in the `tnsnames.ora` file.
`From your desktop: Start | All Programs | Oracle - OraDB12Home1 | Configuration Migration Tools | Net Configuration Assistant`
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_pec1` 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 1522," 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_pec1` as Net Service Name and click Next.
9) On the Net Service Name Configuration, Another Net Service Name page, select No, and Next.
10) On the Net Service Name Configuration Done page, click Next.
11) When you are back on the Welcome page, click Finish.
b. Now connect to `pdb_pec1` using the net service name.

```
$ sqlplus sys/password@pdb_pec1 as sysdba
Connected to:
Oracle Database 12c Enterprise Edition Release 12.1.0.0.2 -
64bit Production
With the Partitioning, OLAP, Advanced Analytics and Real
Application Testing options
SQL>
```

c. Execute the script. Expect around 35 minutes to complete.

```
SQL>
c:\app\oracle\product\12.1.0\dbhome_1\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> -- 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> alter session set container="&pdbname";
Session altered.
SQL>
SQL> WHENEVER SQLERROR CONTINUE;
SQL>
```

d. Quit the session after opening the new PDB.

```
SQL> alter pluggable database pdb_pec1 open;
Pluggable database altered.
```

```
SQL> EXIT
$
```

4. Connect to `PDB_pec1`.

```
$ sqlplus sys\password@101-xx:1521\PDB_PEC1 as SYSDBA
Connected to:
Oracle Database 12c Enterprise Edition Release 12.1.0.0.2 -
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_pec1`

```
SQL> select count(empno) from scott.emp;
COUNT(EMPNO)
------------
14
SQL> EXIT
```

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

Overview

In this practice you merge all PDBs of `orcl12c` into a single CDB, `cdb1`.

1. Merge all PDBs of `orcl12c` into cdb1.
2. Drop `orcl12c`.

Assumptions

The CDB `cdb1 and orcl12c` exists. The `cdb1` creation has completed successfully in Practice 8-1.

Tasks

1. Connect to the multitenant container database `orcl12c` to unplug all PDBs.

a. Connect to `orcl12c` root as a common user with `ALTER PLUGGABLE DATABASE` privilege to unplug `pdborcl`. If the `pdborcl` is still in `READ WRITE` mode, close the PDB.

```
$ sqlplus /@orcl12c 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 name, open_mode from v$pdbs;
NAME OPEN_MODE
------------------------------ ----------
PDB$SEED READ ONLY
PDBorcl READ WRITE
SQL> alter pluggable database PDBorcl unplug into
'xmlfilePDBorcl.xml';
alter pluggable database PDBorcl unplug into 'xmlfilePDBorcl'
*
ERROR at line 1:
ORA-65025: Pluggable database PDBorcl is not closed on all
instances.
SQL> alter pluggable database PDBorcl close immediate;
Pluggable database altered.
SQL> alter pluggable database PDBorcl unplug into
'xmlfilePDBorcl.xml';
Pluggable database altered.
SQL> col PDB_NAME format A20
SQL> select PDB_NAME, STATUS from CDB_PDBS
where PDB_NAME='PDBorcl';
PDB_NAME STATUS
-------------------- -------------
PDB1_1 UNPLUGGED
SQL> drop pluggable database PDBorcl KEEP DATAFILES;
Pluggable database dropped.
SQL> EXIT
```

b. Before plugging `pdborcl` into `cdb1`, you can optionally check whether the unplugged `pdborcl` is compatible with `cdb1` with `DBMS_PDB.CHECK_PLUG_COMPATIBILITY`

function. Connect to `cdb1` root as a common user with `CREATE PLUGGABLE DATABASE` privilege to plug `pdborcl`.
Use the following PL\SQL code:

```
DECLARE
compat BOOLEAN := FALSE;
BEGIN
compat := DBMS_PDB.CHECK_PLUG_COMPATIBILITY(
pdb_descr_file =>
'c:\app\oracle\product\12.1.0\dbhome_1dbs\xmlfilePDBorcl.xml',
pdb_name => 'pdborcl');
if compat then
DBMS_OUTPUT.PUT_LINE('Is pluggable compatible? YES');
else DBMS_OUTPUT.PUT_LINE('Is pluggable compatible? NO');
end if;
end;
/
$ sqlplus / as sysdba
Connected.
SQL> SET SERVEROUTPUT ON
SQL> DECLARE
2 compat BOOLEAN := FALSE;
3 BEGIN
4 compat := DBMS_PDB.CHECK_PLUG_COMPATIBILITY(
5 pdb_descr_file =>
'c:\app\oracle\product\12.1.0\dbhome_1dbs\xmlfilePDBorcl.xml',
pdb_name => 'pdborcl');
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 with 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
where name='PDBorcl';
MESSAGE
--------------------------------------------------------------------
-
ACTION
--------------------------------------------------------------------
-
Parameter sga_target mismatch: Previous value 503316480. CDB
value 5033164800
```

```
Change the parameter in PDB or the CDB
Parameter pga_aggregate_target mismatch: Previous value
167772160. CDB value 1677721600
Change the parameter in PDB or the 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 `pdborcl` into `cdb1`.

```
SQL> create pluggable database pdb1_1 using 'xmlfilePDBorcl.xml'
NOCOPY;
Pluggable database created.
SQL>
```

Notice that you use the clause `NOCOPY` because the `cdb1 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 `pdborcl`.

```
SQL> alter pluggable database pdborcl open;
Pluggable database altered.
SQL>
```

f. Check that the `pdborcl` is in the PDBs list in `cdb1`.

```
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 are unplugged from `orcl12c` (in case you had created other PDBs) and plugged into `cdb1`, you can drop the multitenant container database `orcl12c` with DBCA or SQL commands.

```
$ sqlplus /@orcl12c as sysdba
Connected.
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
```