Practice 4: Performing Basic CDB Administration Tasks

Practice Overview

In this practice, you will implement the regular tasks that you may need to do when managing the PDBs. The target is to explore and get familiar with the CDB architecture.

In this practice you will perform the following:

- Various ways of changing the states of the CDB and the PDBs.
- Modify the PDB restart states.
- Modify a parameter in a PDB
- Change the Global Database Name of a PDB
- Explore the CDB tablespaces and datafiles.

Practice Assumptions

- You have the `srv1` and its CDB database up and running.

Changing the States of the CDB and the PDBs

In the following steps, you will shut down and start up PDBs once after logging on to the root and then after switching current container to the PDB. You will also shut down and start up the CDB.

A. Changing the state of a specific PDB when the current container is the root

1. Create a Putty session to `srv1` and login as `oracle` user.

2. Login to the CDB as sysdba in SQL*Plus.

```
sqlplus / as sysdba
```

3. Check if the database is a CDB or non-CDB:

```
SELECT NAME, CDB FROM V$DATABASE;
```

4. Verify that you are connected to the root.

   The root container always has its `CON_ID` equals to 1 and its name is always `CDB$ROOT`

```
SHOW CON_ID CON_NAME
```

5. Check the state of `PDB1`.

```
SELECT OPEN_MODE FROM V$PDBS WHERE NAME='PDB1';
```

6. Shutdown `PDB1`.

```
ALTER PLUGGABLE DATABASE pdb1 CLOSE IMMEDIATE;
```

7. Check the current state of `PDB1`.

   The view displays its state as `MOUNTED`. In the context of PDB, `MOUNTED` state means it is closed.

```
SELECT OPEN_MODE FROM V$PDBS WHERE NAME='PDB1';
```

8. Startup `PDB1`. Any of the following commands gives the same result.

```
-- execute one of the following commands:
ALTER PLUGGABLE DATABASE pdb1 OPEN;
STARTUP PLUGGABLE DATABASE pdb1
```

B. Changing the state of all the PDBs

In the following steps you will learn how to close all the PDBs in one single statement. You will then start them all up.

9. Close all the PDBs

```
ALTER PLUGGABLE DATABASE ALL CLOSE;
```

10. Check the state of all the PDBs.

Observe that the seed PDB has not been affected by the close command. Only user-created PDB got closed.

```
col name format a10
SELECT NAME, OPEN_MODE FROM V$PDBS ORDER BY 1;
```

11. Start up all the PDBs.

```
ALTER PLUGGABLE DATABASE ALL OPEN;
```

C. Changing the State of the CDB

In the following steps you will shutdown and then start up the entire CDB.

12. Shutdown the entire CDB.

```
SHUTDOWN IMMEDIATE
```

13. Start up the CDB.

```
STARTUP OPEN
```

14. Check the states of the PDBs.

Observe that all the user-created PDBs are closed.

```
SELECT NAME, OPEN_MODE FROM V$PDBS ORDER BY 1;
```

D. Change the PDB default restart states

In this section you will change the PDB default restart states.

15. Startup all the PDBs

```
ALTER PLUGGABLE DATABASE ALL OPEN;
```

16. Save the PDB states.

```
ALTER PLUGGABLE DATABASE ALL SAVE STATE;
```

17. Verify the PDB saved states

```
col con_name format a10
SELECT CON_NAME, STATE FROM CDB_PDB_SAVED_STATES;
```

18. Test the changes by restarting the CDB and checking the state of the PDBs.

```
SHUTDOWN IMMEDIATE
STARTUP OPEN
SELECT NAME, OPEN_MODE FROM V$PDBS ORDER BY 1;
```

E. Changing the state of a specific PDB when the current container is that PDB

19. Switch the current container to `PDB1`

```
ALTER SESSION SET CONTAINER=PDB1;
```

20. Verify that the current container is `PDB1`

```
SHOW CON_ID CON_NAME
```

21. Shutdown the container then start it up.

I would rather use the `ALTER PLUGGABLE DATABASE` command to startup and shutdown a PDB because if the current container is the root and I am not aware about that, the `SHUTDOWN` below will shut down the entire CDB. Using the `ALTER PLUGGABLE DATABASE` command to shut down or startup the root is not possible.

```
SHUTDOWN IMMEDIATE
STARTUP OPEN
```

22. Set the current container back to the root.

```
ALTER SESSION SET CONTAINER=CDB$ROOT;
```

Modifying a Parameter in a PDB

In the following section, you will examine change the parameters at the CDB level and at the PDB level.

F. Modifying a non-PDB-modifiable Parameter

23. Verify that the parameter `DB_RECOVERY_FILE_DEST_SIZE` cannot be changed in the PDB level.

```
col value format a15
SELECT VALUE, ISPDB_MODIFIABLE
FROM   V$SYSTEM_PARAMETER
WHERE  NAME='db_recovery_file_dest_size';
```

24. Change the current container to `PDB1`, query the value of this parameter, and try to change its value.

```
ALTER SESSION SET CONTAINER=PDB1;
SELECT VALUE FROM V$SYSTEM_PARAMETER WHERE NAME='db_recovery_file_dest_size';
ALTER SYSTEM SET DB_RECOVERY_FILE_DEST_SIZE = 10903094248;
```

You should receive the following error:

```
ORA-65040: operation not allowed from within a pluggable database
```

25. Change the current container to the root, and try to change the parameter value.

```
ALTER SESSION SET CONTAINER=CDB$ROOT;
ALTER SYSTEM SET DB_RECOVERY_FILE_DEST_SIZE = 10903094248 SCOPE=BOTH;
```

26. Change the current container to `PDB1` and query the value of the parameter.

```
ALTER SESSION SET CONTAINER=PDB1;
SELECT VALUE FROM V$SYSTEM_PARAMETER WHERE NAME='db_recovery_file_dest_size';
```

27. Set the current container back to the root.

```
ALTER SESSION SET CONTAINER=CDB$ROOT;
```

G. Modifying a PDB-modifiable Parameter

28. Verify that the parameter `DDL_LOCK_TIMEOUT` can be changed in the PDB level.

```
COL VALUE FORMAT A15
SELECT VALUE, ISPDB_MODIFIABLE
FROM V$SYSTEM_PARAMETER
WHERE NAME='ddl_lock_timeout';
```

29. Change the current container to `PDB1` and try to change the parameter value.

```
ALTER SESSION SET CONTAINER=PDB1;
ALTER SYSTEM SET DDL_LOCK_TIMEOUT = 12;
```

30. Set the current container back to the root and query information about the parameter:

```
ALTER SESSION SET CONTAINER=CDB$ROOT;
col name format a20
SELECT CON_ID, NAME, VALUE
  FROM V$SYSTEM_PARAMETER
 WHERE NAME='ddl_lock_timeout';
```

Changing the Global Database Name of a PDB

In the following section, you will examine change the global database name of a PDB. You will create a new PDB, rename it, and finally drop it from the CDB.

H. Creating a new PDB from the seed

31. Create a new pluggable database from the seed.

```
CREATE PLUGGABLE DATABASE pdb_test
 ADMIN USER pdbtestadmin IDENTIFIED BY oracle;
ALTER PLUGGABLE DATABASE pdb_test OPEN;
```

I. Changing the Global Database Name of a PDB

32. Close the PDB and open it in the restricted mode.

```
ALTER PLUGGABLE DATABASE pdb_test CLOSE IMMEDIATE;
ALTER PLUGGABLE DATABASE pdb_test OPEN RESTRICTED;
SELECT CON_ID, OPEN_MODE, RESTRICTED FROM V$PDBS WHERE NAME='PDB_TEST';
```

33. Change the global database name for `pdb_test` to `pdb3`

```
ALTER SESSION SET CONTAINER=PDB_TEST;
ALTER PLUGGABLE DATABASE pdb_test RENAME GLOBAL_NAME to pdb3;
SELECT CON_ID, OPEN_MODE, RESTRICTED FROM V$PDBS WHERE NAME='PDB3';
ALTER PLUGGABLE DATABASE pdb3 CLOSE IMMEDIATE;
ALTER PLUGGABLE DATABASE pdb3 OPEN;
SELECT CON_ID, OPEN_MODE, RESTRICTED FROM V$PDBS WHERE NAME='PDB3';
```

34. Test connecting to the renamed PDB

```
conn system/oracle@//srv1:1521/pdb3.localdomain
```

J. Clean up

35. Drop the new PDB, including its datafiles.

```
conn / as sysdba
ALTER PLUGGABLE DATABASE pdb3 CLOSE IMMEDIATE;
DROP PLUGGABLE DATABASE pdb3 INCLUDING DATAFILES;

col name format a10
SELECT NAME, CON_ID, OPEN_MODE, RESTRICTED FROM V$PDBS ORDER BY 1;
```

Exploring CDB Tablespaces and Datafiles

In the following section, you will explore the data files in the CDB. This is also a demonstration of the new dictionary views family CDB_*.

36. Make sure you are connected to the CDB root.

37. View the default permanent and temporary tablespaces in `CDB1`

```
col property_name format a30
col property_value format a25

SELECT PROPERTY_NAME, PROPERTY_VALUE
FROM   DATABASE_PROPERTIES
WHERE  PROPERTY_NAME LIKE 'DEFAULT_%TABLE%';
```

38. View the permanent and temporary tablespaces in `CDB1`

- Notice that the tablespaces of the SEED PDB is are not retrieved by the dictionary view.

```
col pdb_name format a10

SELECT T.TABLESPACE_NAME, T.CON_ID, P. PDB_NAME
FROM   CDB_TABLESPACES T, CDB_PDBS P
WHERE  T.CON_ID = P.CON_ID (+)
ORDER BY 2,1;
```

39. View all the data files of the CDB.

- Observe that the datafiles of the seed (`CON_ID` = 2) were not retrieved by the query. Oracle 12.1 introduced the parameter `EXCLUDE_SEED_CDB_VIEW` to make this view retrieving the datafiles of the root. In Oracle 12.2, this parameter has been made obsolete.
- Observe that the PDB name is not included in the view queried below. To get it, you need to link the view with the `CDB_PDBS`, as demonstrated in the previous step code.

```
col file_name format a50
col tablespace_name format a8
col file_id format 9999
col con_id format 999
```

```
SELECT FILE_NAME, TABLESPACE_NAME, FILE_ID, CON_ID
FROM CDB_DATA_FILES
ORDER BY CON_ID;
```

40. Retrieve the data files from the `DBA_DATA_FILES`.

Notice only the data files of the root have been retrieved.

```
SELECT FILE_NAME, TABLESPACE_NAME, FILE_ID FROM DBA_DATA_FILES;
```

41. Retrieve the data files using the views `V$DATAFILE` and `V$TABLESPACE`.

Notice that all the data files have been retrieved, including the seed data files.

```
col name format a12
SELECT FILE#, T.NAME, T.TS#, T.CON_ID
FROM   V$DATAFILE D, V$TABLESPACE T
WHERE  D.TS#=T.TS# AND D.CON_ID=T.CON_ID
ORDER BY 4,3;
```

42. Change the current container to the seed and retrieve the data file names from `DBA_DATA_FILES`.

SEED data files are not saved in their own subdirectory, as is the case with the user-created PDBs.

```
ALTER SESSION SET CONTAINER=PDB$SEED;

col file_name format a80
SELECT FILE_NAME FROM DBA_DATA_FILES;
```

Further Exercise

This is further exercise to be done by the student.

- Can a PDB have its own alertlog file?
  Hint: Check out if the parameter `DIAGNOSTIC_DEST`, is it `ISPDB_MODIFIABLE`.

Summary

- When you connect to a root, you can change the state of any PDB. When the current container is a PDB, you can change the state of only the current PDB.
- PDB state can be saved. The next time you start the CDB, the PDB will be in the saved state.
- Some parameters can be modified in the PDB level as well as the CDB level and some parameter can be modified only in the CDB level.
- To change the Global Database Name of a PDB, the PDB must be opened in restricted mode.
- CDB_* views family retrieve information about the CDB and all the PDBs that are associated with the root. DBA_* views family retrieve information about only the current PDB.
- To retrieve all the datafiles in a CDB (including the seed datafiles), use the V$ views.