

## Practice 22

# Using RMAN in Oracle Multitenant Databases

### Practice Target

In this practice, you will use RMAN to perform backup and recovery operations on a CDB database (multitenant environment).

### Practice Overview

In this practice, you will perform the following tasks:

- Create a CDB database named ORACDB in `srv1`
- Using RMAN to create backup files of CDB Database
- Using RMAN to create backups of PDBs
- Enabling a local PDB administrator to backup and restore its PDB
- Implement recovery scenarios in CDB and PDBs
- Recovering the root container
- Recovering the seed container
- Recovery from losing a non-system datafile in a PDB
- Point-in-time recovery (PITR) in a PDB

### Assumptions

This practice assumes that the `srv1` appliance is up and running and its database ORADB is running in OPEN state.

## A. Practice Preparation Steps

Perform the following steps to prepare the environment for this practice. You will basically drop ORADB and create a new CDB database.

1. Take snapshot of `srv1`. Give it the name **"Practice 22 Start"**.
2. In the VirtualBox window of `srv1`, connect as `oracle` and open a terminal window.
3. Invoke `dbca` utility and drop ORADB database.

```
dbca
```

4. Invoke `dbca` utility again and perform the following steps to create a CDB database.

Window	Response
Database Operation	Create Database
Creation Mode	Advanced Configuration
Deployment Type	General Purpose or Transaction processing
Database Identification	<p>Global Database Name: ORACDB.localdomain</p> <p>SID: ORACDB</p> <p><b>Mark the checkbox "Create as Database Container"</b></p> <p>Make sure the checkbox "Use Local Undo tablespace for PDBs" is selected</p> <p>Make sure the radio button "Create a Container database with one or more PDBs" is selected</p> <p>Number of PDBs: 1</p> <p>PDB name: <b>PDB1</b> (upper case)</p>
Storage Option	<p>Select "Use following for the storage attributes"</p> <p><b>Database files storage type:</b> File System</p> <p><b>Database files location:</b> {ORACLE_BASE}/oradata/{DB_UNIQUE_NAME}</p> <p><b>Mark "Use Oracle-Managed Files (OMF)"</b></p>
Fast Recovery Option	<p><b>Mark "Specify the Fast Recovery Area"</b></p> <p>Set the "Fast Recovery Area" to: {ORACLE_BASE}/<b>fra</b>/{DB_UNIQUE_NAME}</p> <p>Fast Recovery Area size (approx): <b>40960</b> MB</p> <p>Mark the checkbox <b>"Enable Archiving"</b>.</p>
Network Configuration	Make sure the <b>LISTENER</b> is selected
Data Vault Option	Make sure the check boxes are unmarked

<b>Configuration Options</b>	<p><b>Memory tab:</b></p> <p>Select "Use Automatic Shared Memory Management"</p> <p>SGA size: 1744</p> <p>PGA size: 652</p> <p><b>Sizing tab:</b></p> <p>Processes: 500</p> <p><b>Character Sets tab:</b></p> <p>select "Use Unicode AL32UTF8"</p> <p><b>Connection mode tab:</b></p> <p>Make sure the "Dedicated server mode" is selected</p> <p><b>Sample Schemas</b></p> <p>Keep the option "Add sample schemas to the database" unmarked</p>
<b>Management Options</b>	Unmark "Configure Enterprise Manager (EM) database express".
<b>Use Credentials</b>	<p>Select "User the same administrative password for all accounts"</p> <p>Set the password (it has been set to "oracle" in my demonstrations)</p>
<b>Creation Option</b>	Make sure "Create database" is selected.
<b>Summary</b>	click on <b>Finish</b>
<b>Finish</b>	click on <b>Close</b>

**Note:** Creating one pluggable database (PDB) in a CDB container is included in the normal database licenses. If you want to create more than one PDB within the CDB, you have to purchase a separate license named "Oracle Multitenant" licenses.

5. Test the created database by connecting to it using SQL\*Plus:

```
export ORACLE_SID=ORACDB
sqlplus system/oracle
```

6. Edit the oratab file

```
vi /etc/oratab
```

7. Change the last field for the database line from **N** to **Y**

```
ORACDB:/u01/app/oracle/product/12.2.0/db_1:Y
```

8. Open Putty, and connect to `srv1` as `oracle`.
9. Verify that the ORACDB and PDB services are registered in the listener.

```
lsnrctl services
```

10. If the services are not registered, perform the following steps:

- a. Set the variable `ORACLE_SID` to the created database.

```
export ORACLE_SID=ORACDB
```

- b. Invoke SQL\*Plus and connect as `sysdba` to the target

```
sqlplus / as sysdba
```

- c. Set the `LOCAL_LISTENER` parameter to null.

When this parameter is set to null, the instance communicates with the default listener.

```
SHOW PARAMETER LOCAL_LISTENER
ALTER SYSTEM SET LOCAL_LISTENER='' SCOPE=BOTH;
```

- d. Register the database.

```
ALTER SYSTEM REGISTER;
```

- e. Verify that the services have been registered.

```
lsnrctl services
```

11. Edit the `tnsnames.ora` file and add the following settings to it.

Do not copy paste from the PDF file. Either copy from the downloadable `tnsnames.ora` file, or from the Putty session window itself.

```
vi $TNS_ADMIN/tnsnames.ora
```

```
ORACDB =
  (DESCRIPTION =
    (ADDRESS = (PROTOCOL = TCP)(HOST = srv1)(PORT = 1521))
    (CONNECT_DATA =
      (SERVER = DEDICATED)
      (SERVICE_NAME = ORACDB.localdomain)
    )
  )
)

PDB1 =
  (DESCRIPTION =
    (ADDRESS = (PROTOCOL = TCP)(HOST = srv1)(PORT = 1521))
    (CONNECT_DATA =
      (SERVER = DEDICATED)
      (SERVICE_NAME = PDB1.localdomain)
    )
  )
)
```

- 12.** Test the connection configurations.

```
sqlplus system/oracle@oracdb  
conn system/oracle@pdb1
```

- 13.** Fix the ORACLE\_SID value in the oracle user profile.

```
vi ~/.bash_profile
```

```
ORACLE_SID=ORACDB; export ORACLE_SID
```

## B. Using RMAN to Create Backup files of CDB Database

In this section of the practice, you will use RMAN to produce and manage backup files on the CDB container.

14. Invoke RMAN and connect as target to ORACDB and make the configurations as shown below.

```
rman target ''/ AS SYSBACKUP''  
CONFIGURE DEFAULT DEVICE TYPE TO disk;  
CONFIGURE CONTROLFILE AUTOBACKUP ON;
```

15. Take backup of the CDB.

The result of the command below is to take backup of all the database datafiles (the root and all its PDBs), control files, SPFILE file, and the archived redo log files.

```
BACKUP DATABASE TAG 'CDB_FULL' PLUS ARCHIVELOG TAG 'ARC_FULL';
```

16. List the backupsets.

You should see backupsets for the following:

- o Control file and SPFILE as part of the AUTOBACKUP file
- o Archived redo log files
- o Datafiles that belong to the root container (CDB\$ROOT)
- o Datafiles that belong to PDB1
- o Datafile that belong to the seed container (PDB\$SEED)

```
LIST BACKUPSET;
```

17. Issue the following commands and observe the difference between them.

The first command lists all the backupsets. The second command lists the backupset of the root container only.

```
LIST BACKUPSET OF DATABASE;  
LIST BACKUPSET OF DATABASE ROOT;
```

- Can the taken backup be used to restore a PDB?

18. Issue the following command to know the answer.

```
LIST BACKUPSET OF PLUGGABLE DATABASE pdb1;
```

## C. Using RMAN to Create PDB Backups

In this section of the practice, you will use RMAN to produce and manage backup files on the PDBs.

19. Take a whole PDB backup of PDB1.

```
BACKUP PLUGGABLE DATABASE pdb1 TAG 'PDB1_FULL';
```

20. Retrieve the backupset of the pluggable database PDB1.

Observe that backupsets tagged as "CDB\_FULL" and "PDB1\_FULL" can both be used to restore the PDB.

```
LIST BACKUPSET OF PLUGGABLE DATABASE pdb1 SUMMARY;
```

21. Take a backup of users tablespace in PDB1

```
BACKUP TABLESPACE pdb1:users TAG 'PDB1_USERS';
```

22. List the backupsets that can be used to restore the users tablespace in pdb1.

All the backupsets taken so far can be used to restore users tablespace in pdb1.

```
LIST BACKUP OF TABLESPACE pdb1:users SUMMARY;
```

## D. Enabling a Local PDB Administrator to Backup and Restore its PDB

In this section of the practice, you will perform a procedure to enable a local admin user of PDB1 to take backup and restore its PDB.

23. In SQL\*Plus, login to the root as sysdba

```
sqlplus / as sysdba
```

24. Create a local user named as pdb1admin for PDB1.

```
ALTER SESSION SET CONTAINER=PDB1;  
CREATE USER pdb1admin IDENTIFIED BY oracle CONTAINER=current;
```

25. Grant SYSBACKUP to the local admin user of PDB1.

This grant will be applicable only in the current container (PDB1), not the CDB.

```
GRANT SYSBACKUP TO pdb1admin CONTAINER=CURRENT;
```

26. Run RMAN and login as target to PDB1 using the local admin user.

```
rman target "'pdb1admin/oracle@PDB1 AS SYSBACKUP'"
```

27. Take a whole PDB backup of PDB1.

```
# issue either of the commands below. To the user pdb1admin point of view, they both have  
# the same effect:  
BACKUP DATABASE TAG 'PDB1_FULL_LOCAL';  
BACKUP PLUGGABLE DATABASE pdb1 TAG 'PDB1_FULL';
```

28. List the backupset files.

Notice that this command does not only display the backupset taken by pdb1admin user, it also displays the backupset of PDB1 taken by SYS user.

```
LIST BACKUPSET;
```

29. Issue the following command. Type NO when it prompts for deletion.

Notice that among the files that will be deleted by the command the backups that were taken by SYS.

```
DELETE BACKUPSET;
```

30. Display backupsets of archive log files and try deleting them.

The user cannot delete backupsets of archive log files because the logged on user has SYSBACKUP privilege on PDB1 only, not the CDB. PDB administrators cannot delete or backup the archived redo log files.

```
LIST BACKUP OF ARCHIVELOG ALL SUMMARY;  
DELETE BACKUPSET OF ARCHIVELOG ALL;
```



## E. Implementing Recovery Scenarios in CDB and PDBs

In this section of the practice, you will perform various recovery scenarios in a CDB database and its PDBs. The proposed solution for each scenario is not necessarily the best solution. Different solutions are proposed to gain experience on using them.

### Scenario 1: Recovering the root container

In this scenario, the CDB system data file is lost. The proposed recovery solution will restore only the lost datafile.

#### Scenario Simulation

In the following steps you will obtain the full name of the CDB system datafile and delete it. CDB system datafile belongs to the root container.

31. Invoke RMAN with connecting to the local database as target

```
rman target /
```

32. Obtain the full name from the system datafile and take a note of it.

```
SELECT NAME FROM V$DATAFILE WHERE FILE# =1;
```

33. Delete the datafile.

```
host "rm -f <datafile name>";
```

34. Validate the database so that the lost datafile is discovered.

```
VALIDATE DATABASE;
```

#### Recovering from the loss

In the following steps, you will take actions to recover from losing the CDB system data file.

35. Issue the following commands:

You could use the "RESTORE DATABASE ROOT" command, but this command restores all the root container datafiles. Restoring the lost datafile alone is quicker.

Observe that the PDBs need to be started, by default, after starting up the CDB.

```
STARTUP FORCE MOUNT  
RESTORE DATAFILE 1;  
RECOVER DATAFILE 1;  
ALTER DATABASE OPEN;  
ALTER PLUGGABLE DATABASE ALL OPEN;
```

## Scenario 2: Recovering the seed container

In this scenario, a data file that belongs to the seed container is lost. The proposed recovery solution will restore the entire PDB.

### Scenario Simulation

In the following steps you will obtain the full name of a data file in the seed container and delete it.

36. Obtain the full names of the seed datafiles and take a note of one of them.

Observe that the seed container `CON_ID` always equals to 2.

```
SELECT NAME FROM V$DATAFILE WHERE CON_ID=2;
```

37. Delete the noted datafile.

```
host "rm -f <datafile name>";
```

38. Invoke rman with connecting to the local database as target then validate the database.

The lost datafile number is reported.

```
VALIDATE DATABASE;
```

### Recovering from the loss

In the following steps, you will take actions to recover from losing a seed datafile.

39. Issue the following shaded commands:

Observe the solution does not need to shut down the CDB. All application sessions remain in their normal operations.

It is more efficient to restore the lost datafile only, as we did in the previous section. We are using a different method just to gain experience on using it.

```
ALTER PLUGGABLE DATABASE "PDB$SEED" CLOSE;  
RESTORE PLUGGABLE DATABASE "PDB$SEED";  
RECOVER PLUGGABLE DATABASE "PDB$SEED";  
ALTER PLUGGABLE DATABASE "PDB$SEED" OPEN READ ONLY;
```

### Scenario 3: Recovery from losing a non-system datafile in a PDB

In this scenario, a data file that belongs to a user tablespace in a PDB is lost. The proposed recovery solution will restore the lost datafile tablespace.

#### Scenario Simulation

In the following steps you will obtain the full name of a data file in PDB1 and delete it.

40. Obtain the full names of the PDB1 datafiles. Take a note of a non-system datafile (it will be the users tablespace datafile).

```
SELECT NAME FROM V$DATAFILE WHERE CON_ID=(SELECT CON_ID FROM V$PDBS WHERE NAME='PDB1');
```

41. Delete the noted datafile.

```
host "rm -f <datafile name>";
```

42. Validate PDB1 datafiles and observe the reported lost datafile number.

```
VALIDATE PLUGGABLE DATABASE pdb1;
```

#### Recovery Procedure

In the following steps, you will recover from losing a non-system PDB datafile by restoring its tablespace.

43. Issue the following queries to retrieve the tablespace name of the lost datafile.

Replace <n> with the lost data file number obtained from the previous step.

In a CDB database, tablespaces are not uniquely identified by their names only. To identify a tablespace, you need to specify its CON\_ID as well.

```
SELECT CON_ID, NAME FROM V$TABLESPACE WHERE TS# = (SELECT TS# FROM V$DATAFILE WHERE FILE#=<n>);
```

# from the CON\_ID obtained above, you can obtain the PDB name:

# In our case, as PDB1 is the first PDB created in the database, its CON\_ID must be 3

```
SELECT NAME FROM V$PDBS WHERE CON_ID=<con id>;
```

44. Issue the following recovery commands.

We need to connect to the PDB in RMAN as target because we cannot pass the PDB name with the tablespace name to the commands ALTER TABLESPACE, RESTORE TABLESPACE, and RECOVER TABLESPACE.

Observe the solution does not need to shut down the CDB. Only the tablespace becomes not available.

```
rman target sys/oracle@pdb1
ALTER TABLESPACE users OFFLINE IMMEDIATE;
RESTORE TABLESPACE users;
RECOVER TABLESPACE users;
ALTER TABLESPACE users ONLINE;
```

45. Validate PDB1 datafiles. It should not return any error.

```
VALIDATE PLUGGABLE DATABASE pdb1;
```

## Scenario 4: Performing Point-in-time recovery (PITR) in a PDB

In this scenario, you will rewind PDB1 to specific point-in-time in the past to recover from losing data.

### Scenario Simulation

In the following steps you will create a table in PDB1 then drop it.

46. Invoke SQL\*Plus and login as SYSDBA to the local instance.

```
sqlplus / as sysdba
```

47. Switch the redo logfile

```
ALTER SYSTEM SWITCH LOGFILE;
```

48. Create a testing table owned by SYSTEM in users tablespace in PDB1.

```
ALTER SESSION SET CONTAINER = PDB1;  
CREATE TABLE SYSTEM.MYDATA TABLESPACE users AS SELECT TABLE_NAME FROM DBA_TABLES;
```

49. Take a note of the current date and time

```
SELECT TO_CHAR(SYSDATE, 'YYYY-MM-DD:HH24:MI:SS') FROM DUAL;
```

50. Switch the redo logfile

```
ALTER SESSION SET CONTAINER = "CDB$ROOT";  
ALTER SYSTEM SWITCH LOGFILE;
```

51. Drop the testing table in PDB1.

```
ALTER SESSION SET CONTAINER = PDB1;  
DROP TABLE SYSTEM.MYDATA;
```

### Recovery Procedure

In the following steps, you will perform PITR on PDB1 to recover the dropped table.

52. Invoke RMAN and login to the CDB as target.

```
rman target /
```

53. Issue the following commands to rewind PDB1 to the noted time. Replace the <rttime> with the noted recovery time.

```
ALTER PLUGGABLE DATABASE pdb1 CLOSE;  
RESTORE PLUGGABLE DATABASE pdb1 UNTIL TIME "TO_DATE('<rttime>', 'yyyy:mm:dd:hh24:mi:ss')";  
RECOVER PLUGGABLE DATABASE pdb1 UNTIL TIME "TO_DATE('<rttime>', 'yyyy:mm:dd:hh24:mi:ss')"  
AUXILIARY DESTINATION '/media/sf_extdisk/backup';  
ALTER PLUGGABLE DATABASE pdb1 OPEN RESETLOGS;
```

54. Verify the data has been successfully recovered.

```
sqlplus system/oracle@pdb1  
SELECT COUNT(*) FROM SYSTEM.MYDATA;
```

## Clean Up

55. (optional) You can keep the CDB database for future use or you can restore `srv1` to the snapshot **"Practice 22 Start"**.

Be aware that by restoring `srv1` to that snapshot, you are reverting the non-CDB database `ORADB` and losing the CDB database `ORACDB`.

## Summary

In this practice, you performed the following tasks:

- Create a CDB database named as ORACDB in `srv1`
- Using RMAN to create backup files of CDB Database
- Using RMAN to create backups of PDBs
- Enabling a local PDB administrator to backup and restore its PDB
- Implement recovery scenarios in CDB and PDBs
- Recovering the root container
- Recovering the seed container
- Recovery from losing a non-system datafile in a PDB
- Point-in-time recovery (PITR) in a PDB