Practice 7: Creating a PDB from non-CDB

Practice Overview

In this practice you will perform the following:

- Create a new PDB by cloning from non-CDB
- Create a new PDB using the `DBMS_PDB` package on a non-CDB database.

Practice Assumptions

- You have the `srv1` and its CDB database up and running.
- You have the `srv2` and its non-CDB database up and running.

A. Create a new PDB by cloning from non-CDB

In this section you will create a new PDB, named PDB7, by cloning the non-CDB database ORADB. Cloning a non-CDB can only be done online.

1. In **srv1**, connect to the root in a SQL*Plus session:

```
sqlplus / as sysdba
```

2. In **srv1**, create a database link in CDB1 that points to the non-CDB database oradb.

   You have configured the tnsnames.ora file to connect to oradb in an earlier practice.

```
CREATE DATABASE LINK oradb_lnk
CONNECT TO system IDENTIFIED BY oracle USING 'ORADB';

SELECT SYSDATE FROM DUAL@ORADB_LNK;
```

3. Retrieve the datafiles in oradb to know their locations.

   Actually, in our case, you do not need to know the location of the source non-CDB data files; because OMF is enabled in the target CDB database. This step is mentioned here to document that you need to know the location of the source datafiles, if OMF is not being used in the target CDB database.

```
SELECT FILE_NAME FROM DBA_DATA_FILES@ORADB_LNK ORDER BY 1;
```

4. Create the new PDB by cloning from oradb.

   Notice that you do not use the FILE_NAME_CONVERT in our case because OMF is being enabled in the target system. Otherwise, you have to use it.

```
CREATE PLUGGABLE DATABASE pdb7 FROM oradb@oradb_lnk;
```

5. Run the script in the PDB7. It takes a few minutes to finish (about 20 minutes in my case).

```
ALTER SESSION SET CONTAINER=pdb7;
@$ORACLE_HOME/rdbms/admin/noncdb_to_pdb.sql
```

6. Open the PDB in read/write mode

```
ALTER PLUGGABLE DATABASE pdb7 OPEN;

SELECT PDB_NAME, STATUS FROM DBA_PDBS;
```

7. Verify that the application data that was in the non-CDB is there in PDB7.

```
SELECT USERNAME FROM DBA_USERS WHERE ORACLE_MAINTAINED='N' ORDER BY 1;
SELECT COUNT(*) FROM HR.EMPLOYEES;
```

8. To save disk space, let's drop the `PDB7`

```
ALTER SESSION SET CONTAINER=CDB$ROOT;
ALTER PLUGGABLE DATABASE pdb7 CLOSE;
DROP PLUGGABLE DATABASE pdb7 INCLUDING DATAFILES;
```

B. Create a new PDB Using the DBMS_PDB Package on a Non-CDB

In this section you will create a new PDB, named PDB8, by creating an unplugged PDB from a non-CDB and then plug the unplugged PDB into the CDB. DBMS_PDB package is mainly your tool to implement this procedure.

9. In srv2, restart the database in read only mode.

```
export ORACLE_SID=ORADB
sqlplus / as sysdba
SHUTDOWN IMMEDIATE
STARTUP MOUNT
ALTER DATABASE OPEN READ ONLY;
```

10. Execute the following procedure:

```
exec DBMS_PDB.DESCRIBE( PDB_DESCR_FILE => '/home/oracle/oradb.xml');
```

11. Copy the generated xml file to srv1. You can do so by executing the following command in srv2.

```
scp -p /home/oracle/oradb.xml root@srv1:/home/oracle/oradb.xml
rm -f /home/oracle/oradb.xml
```

12. In srv1, create the directory where you will copy the datafiles in it.

```
mkdir -p /u01/app/oracle/oradata/ORADB/datafile
```

13. In srv2, generate the code to copy the datafiles of the non-CDB to srv1.

```
SET PAGESIZE 20
SET LINESIZE 200
SELECT 'scp -p ' || NAME || ' root@srv1:/u01/app/oracle/oradata/ORADB/datafile'
as command FROM V$DATAFILE ORDER BY 1;
```

14. Copy the generated code into a text file and make sure it doesn't have any formatting issue.

15. In srv2, shutdown the database.

```
export ORACLE_SID=ORADB
sqlplus / as sysdba
SHUTDOWN IMMEDIATE
```

16. In srv2, executed the generated code line by line.

17. In `srv1`, check if the unplugged PDB is compatible with your CDB:

```
sqlplus / as sysdba
SET SERVEROUTPUT ON
begin
 IF DBMS_PDB.CHECK_PLUG_COMPATIBILITY( PDB_DESCR_FILE =>
'/home/oracle/oradb.xml', PDB_NAME => 'PDB8') THEN
   DBMS_OUTPUT.PUT_LINE('COMPATIBLE');
 ELSE
   DBMS_OUTPUT.PUT_LINE('NOT COMPATIBLE');
 END IF;
end;
/
```

18. In `srv1`, change the permissions of the files copied from srv2:

```
sudo chown oracle:oinstall /home/oracle/oradb.xml
sudo chown -R oracle:oinstall /u01/app/oracle/oradata/ORADB/datafile
```

19. Create `PDB8`.

```
sqlplus / as sysdba
CREATE PLUGGABLE DATABASE pdb8 USING '/home/oracle/oradb.xml';
```

20. Run the script in the `PDB8`

```
ALTER SESSION SET CONTAINER=pdb8;
@$ORACLE_HOME/rdbms/admin/noncdb_to_pdb.sql
```

21. Open the PDB in read/write mode

```
ALTER PLUGGABLE DATABASE pdb8 OPEN;

SELECT PDB_NAME, STATUS FROM DBA_PDBS;
```

C. Cleanup

22. To save disk space, let's drop the `PDB7` and delete the staging datafiles.

```
ALTER SESSION SET CONTAINER=CDB$ROOT;
ALTER PLUGGABLE DATABASE pdb8 CLOSE;
DROP PLUGGABLE DATABASE pdb8 INCLUDING DATAFILES;
host rm -r /u01/app/oracle/oradata/ORADB/datafile/
```

Summary

In this practice, you implemented the following two methods of creating a PDB from a non-CDB database:

- by cloning the non-CDB online through a database link.
- by using the `DBMS_PDB` to plug copies of the non-CDB database datafiles into the CDB database.

In both methods:

- The non-CDB must be opened for read only operation during implementing the method.
- You must run the Oracle-supplied script `noncdb_to_pdb.sql` to convert the non-CDB into a PDB.