Practice 17: Using Data Pump Utility with PDBs

Practice Overview

In this practice, you will perform the following:

- Export the data from a non-CDB database and import it into a PDB.
- Export the data from a PDB and import it into another PDB within the same CDB.

Practice Assumptions

- `CDB1` database (in `srv1`) is up and running.
- `ORADB` database (in `srv2`) is up and running.

A. Export from non-CDB and Import into PDB

In this section of the practice, you will use the data pump utility to export a tablespace (USERS) from the non-CDB (ORADB) using the FULL TRANSPORTABLE mode and import it into a PDB (PDB2 in CDB1).

1. Login in sqlplus to ORADB as SYSTEM and list the tablespaces and the number of rows in HR.EMPLOYEES table.

List the user-created tablespaces (not SYSTEM, SYSAUX, UNDO, and TEMP). In our case, we have only the USERS tablespace.

```
sqlplus system/oracle@oradb

SELECT TABLESPACE_NAME FROM DBA_TABLESPACES;

SELECT COUNT(*) FROM HR.EMPLOYEES;
```

2. (optional) Check on which tablespace the HR tables are located. If they are located in a system tablespace, relocate them to a user-created tablespace (USERS tablespace in our case).

In our case, they are in SYSAUX tablespace and thus must be relocated to USERS tablespace.

```
SELECT DISTINCT TABLESPACE_NAME FROM DBA_TABLES WHERE OWNER='HR';

-- to relocate the tables:
SET SERVEROUTPUT ON
BEGIN
FOR R IN (SELECT TABLE_NAME FROM DBA_TABLES WHERE OWNER='HR') LOOP
 EXECUTE IMMEDIATE 'ALTER TABLE HR.' || R.TABLE_NAME || ' MOVE TABLESPACE USERS';
 DBMS_OUTPUT.PUT_LINE(R.TABLE_NAME);
END LOOP;
END;
/

-- relocating the tables makes their indexes unusable.
-- rebuild the indexes
BEGIN
FOR R IN (SELECT INDEX_NAME FROM DBA_INDEXES WHERE OWNER='HR' AND INDEX_NAME
<>'COUNTRY_C_ID_PK') LOOP
 DBMS_OUTPUT.PUT_LINE(R.INDEX_NAME);
 EXECUTE IMMEDIATE 'ALTER INDEX HR.' || R.INDEX_NAME || ' REBUILD TABLESPACE
USERS';
END LOOP;
END;
/

-- verify the relocation was successful:
SELECT DISTINCT TABLESPACE_NAME FROM DBA_TABLES WHERE OWNER='HR';
SELECT DISTINCT TABLESPACE_NAME, STATUS FROM DBA_INDEXES WHERE OWNER='HR';
```

3. Set the listed tablespaces (in our case it is only `USERS` tablespace) in READ-ONLY mode.

```
ALTER TABLESPACE users READ ONLY;
```

4. Issue the following data pump export command.

```
export ORACLE_SID=ORADB

expdp system/oracle DUMPFILE=expfull.dmp FULL=Y TRANSPORTABLE=ALWAYS
LOGFILE=exp.log
```

5. At the end of the utility output, you will see the filename of the generated dump file and the data files that you need to copy. Take note of each full filename.

```
******************************************************************************
Dump file set for SYSTEM.SYS_EXPORT_FULL_01 is:
/u01/app/oracle/admin/ORADB/dpdump/expfull.dmp
******************************************************************************
Datafiles required for transportable tablespace USERS:
/u01/app/oracle/oradata/ORADB/datafile/o1_mf_users_****_.dbf
```

6. Login to `PDB2` (in `CDB1`) as `SYSDBA` and check in which directory the datafiles of `PDB2` are stored. Take a note of it.

In our case, because OMF is configured, it should be in the following directory format:

`/u01/app/oracle/oradata/pdb4/CDB1/<GUID>/datafile/***.dbf`

```
sqlplus sys/oracle@pdb2 as sysdba
SELECT NAME FROM V$DATAFILE;
```

7. In `srv2`, copy the data file(s) reported by the export data pump execution to the target locations

```
scp /u01/app/oracle/oradata/ORADB/datafile/o1_mf_users_***_.dbf
oracle@srv1:/u01/app/oracle/oradata/CDB1/***/datafile/o1_mf_users_***_.dbf
```

8. After copying the file(s), you can set the `USERS` tablespaces in `ORADB` back to read/write mode.

```
sqlplus system/oracle@oradb
ALTER TABLESPACE users READ WRITE;
```

9. In `PDB2` (in `srv1`), obtain the Data Pump directory for the dump files and take a note of it.

This is the directory where you need to copy the dump file to.

```
SELECT DIRECTORY_NAME, DIRECTORY_PATH
FROM DBA_DIRECTORIES
WHERE DIRECTORY_NAME='DATA_PUMP_DIR';
```

10. In `srv2`, copy the dump file to the target data pump directory.

```
scp /u01/app/oracle/admin/ORADB/dpdump/expfull.dmp
oracle@srv1:/u01/app/oracle/admin/CDB1/dpdump/***/expfull.dmp
```

11. In PDB2, make sure that there are not any tablespaces having the same names as the tablespaces in the source ORADB database.

PDB2 does have a tablespace named USERS. Therefore, you have to use the REMAP TABLESPACE parameter when you import the data.

```
SELECT TABLESPACE_NAME FROM DBA_TABLESPACES;
```

12. Import the database into PDB2 in FULL TRANSPORTABLE mode.

```
impdp system/oracle@pdb2 FULL=Y dumpfile=expfull.dmp
TRANSPORT_DATAFILES='/u01/app/oracle/oradata/CDB1/***/datafile/o1_mf_users_***_.d
bf' logfile=import.log REMAP_TABLESPACE=users:users2
```

13. Study the errors reported by the utility (if any).

You will receive errors complaining that undo and temp tablespaces are already there and also an error about insufficient privileges to change the password of the SYS user. Those errors can be safely ignored.

```
vi /u01/app/oracle/admin/CDB1/dpdump/<GUID>/import.log
```

14. Check that the USERS tablespace is in place, and that the HR.EMPLOYEES table is created.

```
sqlplus sys/oracle@pdb2 as sysdba

SELECT TABLESPACE_NAME FROM DBA_TABLESPACES ORDER BY TABLESPACE_NAME;

SELECT FILE_NAME, TABLESPACE_NAME FROM DBA_DATA_FILES WHERE
TABLESPACE_NAME='USERS2';

SELECT count(*) FROM HR.EMPLOYEES;
```

15. Cleanup

```
-- delete the dump file and the log file.
-- in srv2:
rm /u01/app/oracle/admin/ORADB/dpdump/*

-- in srv1: delete the files generated by the data pump
```

B. Export and Import Between PDBs

In this practice section, you will export a schema (HR) from one PDB (PDB2) to another PDB (PDB1) within the same CDB (CDB1).

16. Make sure both PDB1 and PDB2 are up. If any one of them is down, start it up.

```
sqlplus / as sysdba

SELECT NAME, OPEN_MODE FROM V$PDBS;
ALTER PLUGGABLE DATABASE PDB1 OPEN;
```

17. Connect to the target PDB (PDB1) as SYSDBA and create a directory object and grant read and write privileges on the directory object to the SYSTEM user. Do the same again in the source PDB, PDB2.

```
mkdir /home/oracle/staging

sqlplus sys/oracle@pdb1 as sysdba
CREATE DIRECTORY dp_dir AS '/home/oracle/staging';
GRANT READ, WRITE ON DIRECTORY dp_dir TO SYSTEM;


conn sys/oracle@pdb2 as sysdba
CREATE DIRECTORY dp_dir AS '/home/oracle/staging';
GRANT READ, WRITE ON DIRECTORY dp_dir TO SYSTEM;
```

18. Use the data pump utility to export the HR schema from PDB2

```
expdp system/oracle@pdb2 DUMPFILE=exppdb2.dmp DIRECTORY=dp_dir SCHEMAS=hr
```

19. Verify that there is no schema in PDB1 that has the same name as the source schema name.

```
sqlplus system/oracle@pdb1
SELECT USERNAME FROM DBA_USERS WHERE USERNAME='HR';
```

20. Verify that the USERS tablespace exists in PDB1 for the HR schema. Create the tablespace, if it does not exist.

```
SELECT COUNT(*) FROM DBA_TABLESPACES WHERE TABLESPACE_NAME='USERS';

CREATE TABLESPACE users;
```

21. Use the data pump utility to import the HR schema into PDB1.

```
impdp system/oracle@pdb1 DUMPFILE=exppdb2.dmp DIRECTORY=dp_dir SCHEMAS=hr
REMAP_TABLESPACE=users2:users
```

22. Verify that there are two distinct HR local users in CDB1, one in PDB1 and another one in PDB2.

```
sqlplus sys/oracle@cdb1 as sysdba

col username format A20
SELECT USERNAME, CON_ID, COMMON FROM CDB_USERS WHERE USERNAME= 'HR';
```

23. Clean up

```
-- delete the files in /home/oracle/staging
rm /home/oracle/staging/*
```

Summary

In Oracle 12c, Data pump can be used to:

- Export the data from a non-CDB database and import it into a PDB.
- Export the data from a PDB and import into another PDB.