# Practices for Lesson 13: Using Other Techniques to Create PDBs

Practices for Lesson 13: Overview

Practices Overview

In these practices, you will create PDBs by using various methods.

Cloning a regular PDB in hot mode and with automatic refreshing

Relocating a PDB

Practice 13-1: Cloning Remote PDBs in Hot Mode

Overview

In this practice, because you have been informed of performance issues on the PDB\_SOURCE\_FOR\_HOTCLONE PDB in ORCL, you will clone the PDB in hot mode as the pdb\_HOTCLONE PDB in the CDBTEST test instance for performance tests. The remote PDB\_SOURCE\_FOR\_HOTCLONE production PDB in ORCLCDB will still be up and fully functional while the actual clone operation is taking place. At the end of the practice, you will create a refreshable copy, refreshed manually or automatically, which will allow you to take your time to test the performance issue.

Assumptions:

CDB orclcdb contain at least two pdbs: orclpdb1, orclpdb2 and orclpdb3

Tasks

Execute the /home/oracle/labs/admin/cleanup\_PDBs.sh shell script to prepare your CDB and PDBs for this practice. The shell script drops all PDBs that may have been created in ORCLCDB. If the PDBs do not exist, error messages will show that they do not exist.

Execute the $HOME/labs/DBMod\_PDB/glogin\_4.sh shell script. It sets formatting for all columns selected in queries.

Execute the $HOME/labs/DBMod\_PDBs/setup\_hotclone.sh script which creates the production PDB\_SOURCE\_FOR\_HOTCLONE PDB from the PDB seed in ORCLCDB, creates CDBTEST, creates a local SOURCE\_USER user in PDB\_SOURCE\_FOR\_HOTCLONE, and creates the SOURCE\_USER.BIGTAB table with thousands of rows. This script will take some time to complete.

In your terminal window, set the environment variables to orclcdb, then grant privileges to the user who will perform the hot clone operation in this case SYSTEM.

As a sysdba privileged user, grant privileges to the system user who will create the hot clone.

In your terminal, log into sqlplus and start a transaction in the PDB\_SOURCE\_FOR\_HOTCLONE PDB. See *Course Practice Environment: Security Credentials* for passwords. Name the terminal window *Session ORCLPDB*. **Hint:** In the terminal menu, click Terminal > Set Title.

In CDBTEST, clone PDB\_HOTCLONE from PDB\_SOURCE\_FOR\_HOTCLONE in hot mode.

Start a new terminal window, and name it *Session CDBTEST.* Create the directory needed for PDB\_HOTCLONE.

In this terminal window, named *Session CDBTEST*, Set the sqlprompt to CDBTEST.

Clone the PDB\_HOTCLONE PDB from PDB\_SOURCE\_FOR\_HOTCLONE while the source PDB is still up and fully functional. See *Course Practice Environment: Security Credentials* for passwords.

CDBTEST> DROP PUBLIC DATABASE LINK link\_pdb\_source\_for\_hotclone;

DROP PUBLIC DATABASE LINK link\_pdb\_source\_for\_hotclone

\*

ERROR at line 1:

ORA-02024: database link not found

CDBTEST> **CREATE PUBLIC DATABASE LINK**

**link\_pdb\_source\_for\_hotclone**

**CONNECT TO system IDENTIFIED BY *password***

**USING 'pdb\_source\_for\_hotclone';**

Database link created.

CDBTEST> ALTER SESSION SET db\_create\_file\_dest= '/u01/app/oracle/oradata/CDBTEST/pdb\_hotclone';

Session altered.

CDBTEST> **CREATE PLUGGABLE DATABASE pdb\_hotclone**

**FROM pdb\_source\_for\_hotclone@link\_pdb\_source\_for\_hotclone FILE\_NAME\_CONVERT=**

**('/u01/app/oracle/oradata/ORCLCDB/pdb\_source\_for\_hotclone', '/u01/app/oracle/oradata/CDBTEST/pdb\_hotclone')**

**REFRESH MODE MANUAL;**

Pluggable database created.

CDBTEST>

Open PDB\_HOTCLONE in READ ONLY mode only.

Select the same data from SOURCE\_USER.BIGTAB in the cloned PDB.

In *Session ORCLPDB*, commit the updated and confirm the change.

In *Session CDBTEST*, refresh the data in the cloned PDB in CDBTEST.

**Note:** The refreshable copy PDB must be closed in order for refresh to be performed. If it is not closed when automatic refresh is attempted, the refresh will be deferred until the next scheduled refresh.

Drop the current refreshable copy PDB and re-create it to configure it as an automatic refreshable clone.

**Note:** The refreshable copy PDB must be closed in order for refresh to be completed. If it is not closed when automatic refresh is attempted, the refresh will be deferred until the next scheduled refresh.

In terminal *ORCLCDB*, update and commit the source data in ORCL.

In *Session CDBTEST*, check that the data in PDB\_HOTCLONE is refreshed.

Verify that the data is refreshed. It is not refreshed.

Close PDB\_HOTCLONE.

After the sleep 120 command has completed, open the PDB and verify that the data is refreshed.

Execute the $HOME/labs/PDB/cleanup\_hotclones.sh script to drop the

PDB\_SOURCE\_FOR\_HOTCLONE in orclcdb and PDB\_HOTCLONE in CDBTEST.

Close all terminal windows.

Practice 13-2: Relocating PDBs

In this practice, you will move PDB1 from ORCLCDB into CDBTEST in one step, using the Near-zero Downtime PDB Relocation feature.

Assumptions:

It is assumed that the database and listener are running. You can use the pgrep -lf smon command to verify that the database is started and the pgrep -lf tns command to verify that the listener is started. If you need to restart the database and listener, use the dbstart.sh script.

The CDBTEST instance exists and is started.

Tasks

Open a terminal and execute the setup\_pdb3.sh shell script to re-create ORCLPDB3. This script also adds the PDB3 service name in the tnsnames.ora file. You will be prompted for the system password (which will not be shown). See *Course Practice Environment: Security Credentials* for passwords.

In the current terminal window, set the title to *Session ORCLCDB1*. Set the SQL prompt to ORCLCDB1. Verify that the source, orclcdb, is configured to use local undo.

Verify that the test user has been created in the ORCLPDB3 PDB, and the table test.bigtab has been populated. See *Course Practice Environment: Security Credentials* for passwords.

**Note:** The net service name is PDB3.

In *Session ORCLCDB1*, prepare to relocate ORCLPDB3 from orclcdb into CDBTEST as

PDB\_RELOCATED.

In orclcdb, in *Session ORCLCDB 1*, create the database link to access CDBTEST. See *Course Practice Environment: Security Credentials* for passwords.

List the PDBs. The con\_id for the PDBs may vary from the values shown below.

If ORCLPDB1 or any of the pluggable databases are not open, issue the following command to open all the pluggable databases.

Re-display the status of all the pluggable databases

Open a new terminal, and set the title to *Session CDBTEST.*

Start SQL\*Plus connected to the CDBTEST instance, and set the SQL prompt to

CDBTEST.

In *Session CDBTEST*, create the database link to access ORCLPDB3 in orclcdb. See

*Course Practice Environment: Security Credentials* for passwords.

Relocate ORCLPDB3. Display the status of the new PDB.

In *Session ORCLCDB1*, grant the required privilege to SYSTEM.

In *Session CDBTEST*, relocate PDB\_RELOCATED from ORCLCDB into CDBTEST.

**Note:** You can relocate with the AVAILABILITY MAX clause, which ensures smooth migration of workload and persistent connection forwarding from ORCLCDB to CDBTEST.

The “maximum availability” mode reduces application impact by handling the migration of connections. The source PDB is preserved in mount state to guarantee the connection

forwarding of the listener to the remote listener where the PDB is now relocated. This forwarding persists even after the relocation operation has been completed and effectively allows for no changes to connect strings. It is expected that connect strings are updated at a time that is convenient for the application. Once this is done and all clients connect to the new host without forwarding, the source PDB can be dropped.

Open a terminal window, set the terminal title to *Session ORCLCDB2*. Prepare to start a session and a transaction in ORCLPDB3 to show that while PDB relocation is taking place the transaction will be transferred to the new relocated PDB.

**DO NOT start** the $HOME/labs/PDB/sessions.sh shell script until CREATE PLUGGABLE DATABASE pdb\_relocated has the RELOCATING status in the terminal window, named *Session CDBTEST*.

In *Session ORCLCDB2*, execute **$HOME/labs/PDB/sessions.sh**. It will last around 5000 seconds. Do not wait for the script to complete, continue to the next steps.

During **sessions.sh** execution:

In *Session ORCLCDB 1*, you can display the status of the source PDB.

In *Session CDBTEST*, you can open the relocated PDB in read-only mode.

Note: the value for NEW DATA during relocation may vary from the example.

If you consider that **sessions.sh** execution is taking too long, you can open the relocated PDB in force mode. When the newly created PDB is opened in read-write mode for the first time, the final steps of the relocation take effect.

The source PDB is closed and dropped from the source CDB.

Any session that was established while the PDB was first opened in read-only mode is preserved if the FORCE option is used to transition the PDB from read-only to read-write.

**Note:** Observe that this interrupts sessions.sh execution taking place in *Session ORCLCDB 2*.

Still in *Session CDBTEST*, verify that the application data is relocated in PDB\_RELOCATED

in CDBTEST.

Note: the value for NEW DATA during relocation may vary from the example.

In the *Session ORCLCDB 1*, verify that orclpdb3 does not exist in ORCLCDB anymore.

In *Session CDBTEST*, drop PDB\_RELOCATED in CDBTEST.

In *Session ORCLCDB 1*, revoke the SYSOPER privilege from SYSTEM.

Close all open terminals.