Oracle Database 12*c*: New Features for Administrators

Activity Guide

D77758GC10
Edition 1.0
June 2013
D80606

ORACLE

Disclaimer

This document contains proprietary information and is protected by copyright and other intellectual property laws. You may copy and print this document solely for your own use in an Oracle training course. The document may not be modified or altered in any way. Except where your use constitutes "fair use" under copyright law, you may not use, share, download, upload, copy, print, display, perform, reproduce, publish, license, post, transmit, or distribute this document in whole or in part without the express authorization of Oracle.

The information contained in this document is subject to change without notice. If you find any problems in the document, please report them in writing to: Oracle University, 500 Oracle Parkway, Redwood Shores, California 94065 USA. This document is not warranted to be error-free.

Restricted Rights Notice

If this documentation is delivered to the United States Government or anyone using the documentation on behalf of the United States Government, the following notice is applicable:

U.S. GOVERNMENT RIGHTS
The U.S. Government's rights to use, modify, reproduce, release, perform, display, or disclose these training materials are restricted by the terms of the applicable Oracle license agreement and/or the applicable U.S. Government contract.

Trademark Notice

Oracle and Java are registered trademarks of Oracle and/or its affiliates. Other names may be trademarks of their respective owners.

Authors

Dominique Jeunot, Jean-François Verrier

Technical Contributors and Reviewers

James Spiller, Donna Keesling, Maria Billings, Lachlan Williams, Peter Fusek, Dimpi Sarmah, Branislav Valny, Christina Nayagam, Frank Fu, Joel Goodman, Gerlinde Frenzen, Harald Van Breederode, Herbert Bradbury, Hermann Baer, Jim Stenoish, Malareddy Goutam, Patricia Mcelroy, Paul Needham, Puneet Sangar, Robert Mcguirk, Sailaja Pasupuleti, Sean Kim, Sharath Bhujani, Steven Wertheimer, Uwe Hesse, Vimala Jacob

This book was published using: Oracle Tutor

Table of Contents

Practices for Lesson 1: Enterprise Manager Cloud Control and Other Tools

Chapter 1

Practices for Lesson 1: Overview

Practices Overview

Your system currently has Oracle Database 12*c* software installed, as well as three pre-created database called `orcl`, `orcl2` and `cdb1`.

You act as an Enterprise Manager administrator. You access Oracle Enterprise Manager Cloud Control 12*c* as the `sysman` user with the `Oracle123` password and select **Summary** as your home page. You start exploring some of the Oracle Enterprise Manager Cloud Control 12*c* functionalities through the different menus and options. And lastly, you will add the `orcl` database as a monitored target.

Practice 1-1: Accessing Enterprise Manager

Overview

In this practice, you access Oracle Enterprise Manager Cloud Control 12*c* as the `sysman` user with the `Oracle123` password and select **Summary** as your home page.

Assumptions

You reviewed the Oracle Enterprise Manager 12*c*: Console Overview and Customization demonstration or have the equivalent navigation knowledge.

Tasks

1. Click the Firefox icon on the top panel (toolbar region) above the desktop to open a browser to access the Enterprise Manager Cloud Control console.
2. Enter the URL for Cloud Control: `https://<em_server_hostname>.<domain>:7802/em`**.** In the current setup, use https://localhost:7802/em. If an error appears, you must first start the OMS, else proceed directly with step 3.
   a. Start the Enterprise Manager Repository Database `em12rep` if not started already.

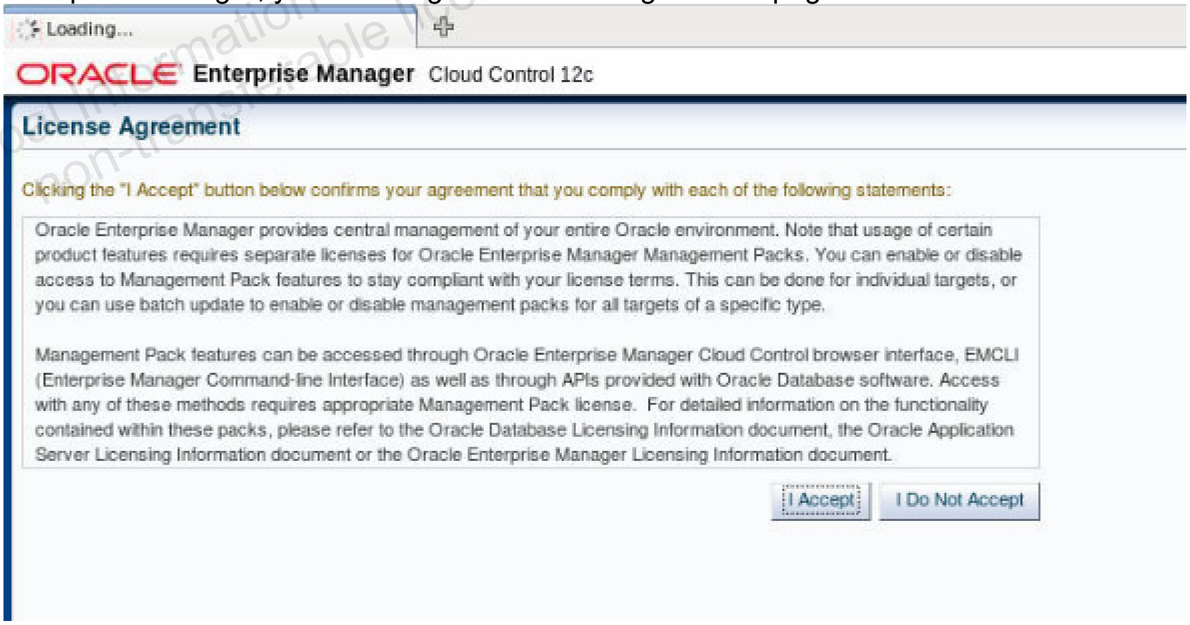
```
$ . oraenv
ORACLE_SID = [orcl] ? em12rep
The Oracle base for
ORACLE_HOME=/u01/app/oracle/product/12.1.0/dbhome_1 is
/u01/app/oracle
$ sqlplus / as sysdba

Connected to an idle instance.
SQL> startup
ORACLE instance started.

Total System Global Area  400846848 bytes
Fixed Size               2271568 bytes
Variable Size          339740336 bytes
Database Buffers        50331648 bytes
Redo Buffers             8503296 bytes
Database mounted.
Database opened.
SQL> EXIT
$
```

   b. Restart the OMS.

```
$ export OMS_HOME=/u01/app/oracle/product/middleware/oms
$ $OMS_HOME/bin/emctl start oms
Oracle Enterprise Manager Cloud Control 12c Release 2
```

```
Copyright (c) 1996, 2012 Oracle Corporation.  All rights
reserved.
Starting Oracle Management Server...
Starting WebTier...
WebTier Successfully Started
Oracle Management Server Successfully Started
Oracle Management Server is Up
WARNING: Limit of open file descriptors is found to be 1024.
The OMS has been started but it may run out of descriptors under
heavy usage.
For proper functioning of OMS, please set "ulimit -n" to be at
least 4096.
$
```

3. Most likely, you receive a Secure Connection Failed message and you need to add a security exception. Click **Or you can add an exception**.
   a. At the end of the alert box, click **I Understand the Risks**.
   b. At the bottom of the page, click **Add Exception**.
   c. In the Add Security Exception pop-up window, click **Get Certificate**.
   d. Confirm that “Permanently store this exception” is selected in your training environment and click **Confirm Security Exception**.
4. The Enterprise Manager Cloud Control console appears.
5. Enter **`sysman`** in the User Name field and **`Oracle123`** in the Password field. Then click **Login**.
6. The first time a new user logs in to Enterprise Manager, a page asks you to accept the license agreement. You have to accept only once. Then each time you will log in to Enterprise Manager, you will not get the license agreement page.



License Agreement

Clicking the "I Accept" button below confirms your agreement that you comply with each of the following statements:

Oracle Enterprise Manager provides central management of your entire Oracle environment. Note that usage of certain product features requires separate licenses for Oracle Enterprise Manager Management Packs. You can enable or disable access to Management Pack features to stay compliant with your license terms. This can be done for individual targets, or you can use batch update to enable or disable management packs for all targets of a specific type.

Management Pack features can be accessed through Oracle Enterprise Manager Cloud Control browser interface, EMCLI (Enterprise Manager Command-line Interface) as well as through APIs provided with Oracle Database software. Access with any of these methods requires appropriate Management Pack license. For detailed information on the functionality contained within these packs, please refer to the Oracle Database Licensing Information document, the Oracle Application Server Licensing Information document or the Oracle Enterprise Manager Licensing Information document.

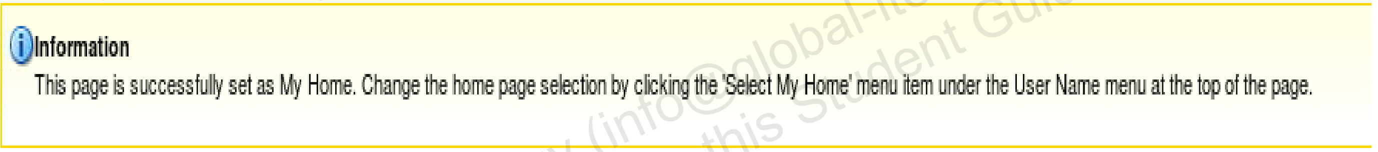
7. The “Select Enterprise Manager Home Page” page appears with choices, such as:
   - Summary
   - Databases
   - Incidents
   - SOA
   - Middleware
   - Composite Application
   - Service Request
   - Services
   - Business Applications
   - Compliance Dashboard

Each choice has a Preview and a Select As My Home button.

The page also has global menus with the following choices: Enterprise, Targets, Favorites, History, and Search Target Name (next to the search entry field). Each of the menu items has drop-down menus with further choices.

Preview any images that interest you.

8. Click the “Select As My Home” next to the **Summary** choice. After being successfully set, it informs you how to change it.

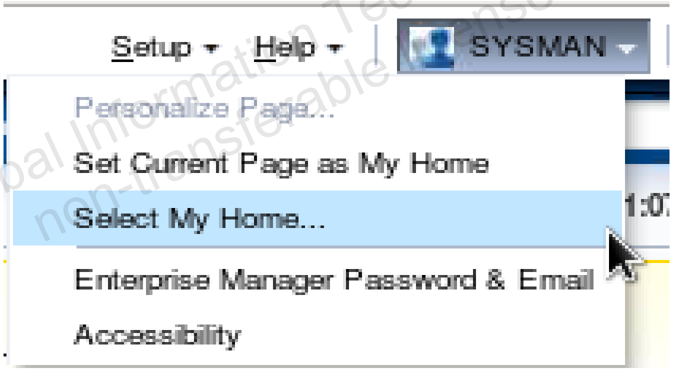


Information

This page is successfully set as My Home. Change the home page selection by clicking the 'Select My Home' menu item under the User Name menu at the top of the page.

9. Question: How can you change your home selection after the initial setup?

   Answer: SYSMAN > Select My Home… menu.



Practice 1-2: Adding a Database Instance as a New Target Monitored by EM Cloud Control

Overview

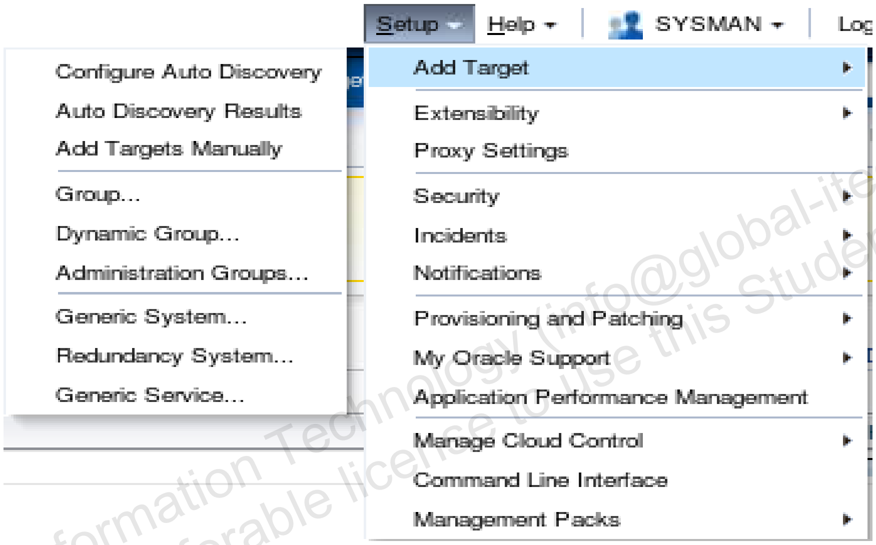
Assumptions

As a prerequisite task, you must log in to Enterprise Manager Cloud Control as the `SYSMAN` user with `Oracle123` as the password.

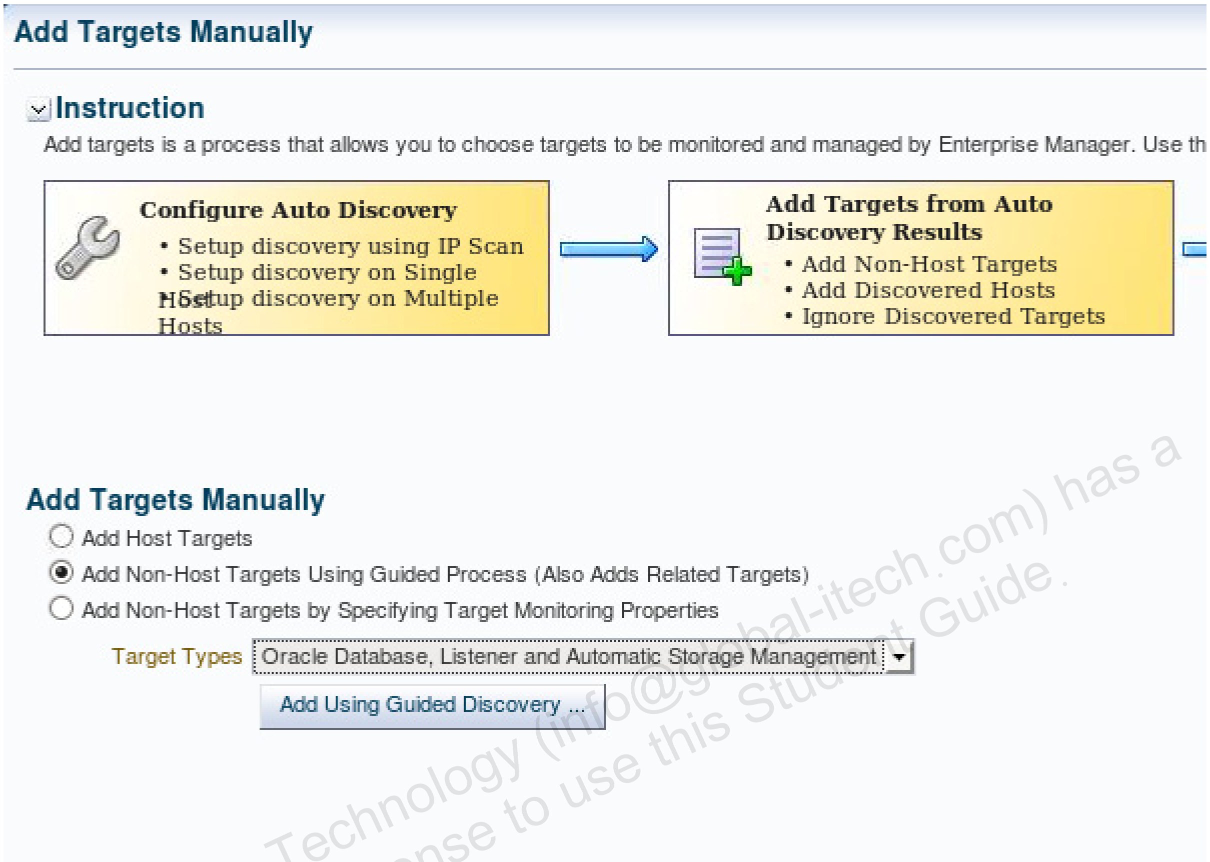
Tasks

First add the database instance `orcl` as a new target monitored by Oracle Enterprise Manager Cloud Control.

1. Add the `orcl` Database Instance as a new target in Enterprise Manager Cloud Control.
   a. In the **Summary** section, click the “Setup” > “Add Target” > “Add Targets Manually”.



b. In “Add Targets Manually”, choose “Add Non-Host Targets Using Guided Process (Also Adds Related Targets)”. Then in “Target Types”, choose “Oracle Database, Listener and Automatic Storage Management” for “Target Type”. Click “Add Using Guided Discovery …” button.



Add Targets Manually

Instruction

Add targets is a process that allows you to choose targets to be monitored and managed by Enterprise Manager. Use th

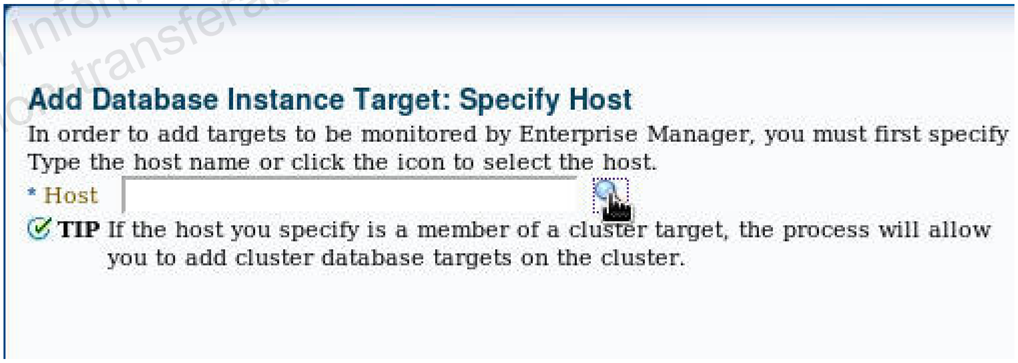
Configure Auto Discovery
• Setup discovery using IP Scan
• Setup discovery on Single Host
• Setup discovery on Multiple Hosts
Add Targets from Auto Discovery Results
• Add Non-Host Targets
• Add Discovered Hosts
• Ignore Discovered Targets

Add Targets Manually

○ Add Host Targets
● Add Non-Host Targets Using Guided Process (Also Adds Related Targets)
○ Add Non-Host Targets by Specifying Target Monitoring Properties

Target Types: Oracle Database, Listener and Automatic Storage Management

c. In “Add Database Instance target: Specify Host”, click the magnifying glass to find your host. Select your host, then click “Continue”.



Add Database Instance Target: Specify Host

In order to add targets to be monitored by Enterprise Manager, you must first specify

Type the host name or click the icon to select the host.

* Host

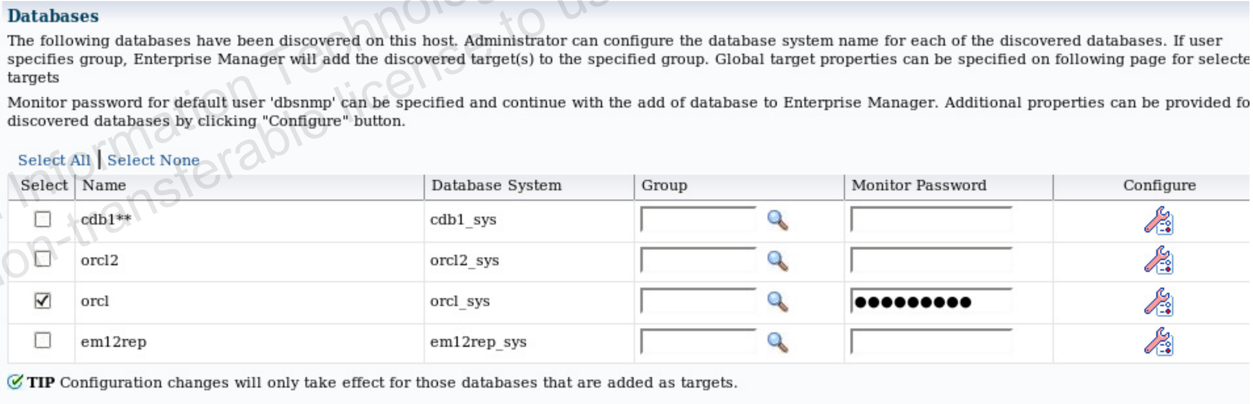
✓ TIP If the host you specify is a member of a cluster target, the process will allow you to add cluster database targets on the cluster.

d. In the “Databases” list, deselect all databases except `orcl`. Deselect the listener.

1) Unlock the `DBSNMP` user. This user is the monitoring user used to test the connection once the target is being added. Open a terminal window.

```
$ . oraenv
ORACLE_SID = [oracle] ? orcl
The Oracle base for
ORACLE_HOME=/u01/app/oracle/product/12.1.0/dbhome_1 is
/u01/app/oracle
$ sqlplus / as sysdba

Connected to:
Oracle Database 12c Enterprise Edition Release 12.1.0.0.2 -
64bit Production
With the Partitioning, OLAP, Advanced Analytics and Real
Application Testing options

SQL> alter user dbsnmp identified by oracle_4U account unlock;

User altered.

SQL> EXIT
$
```

2) Enter `oracle_4U` for the “Monitor Password”.



Databases

The following databases have been discovered on this host. Administrator can configure the database system name for each of the discovered databases. If user specifies group, Enterprise Manager will add the discovered target(s) to the specified group. Global target properties can be specified on following page for selecte targets

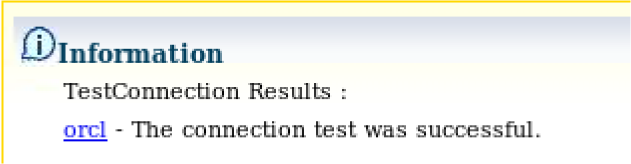
Monitor password for default user 'dbsnmp' can be specified and continue with the add of database to Enterprise Manager. Additional properties can be provided fo discovered databases by clicking "Configure" button.

Select All | Select None

Select	Name	Database System	Group	Monitor Password	Configure
□	cdb1**	cdb1_sys			
□	orcl2	orcl2_sys			
☑	orcl	orcl_sys		●●●●●●●●●	
□	em12rep	em12rep_sys			

TIP Configuration changes will only take effect for those databases that are added as targets.

e. Click the “Test Connection” button. You should receive the following message:



Information

TestConnection Results :

orcl - The connection test was successful.

f. Click the “Finish” then “Save” buttons to complete the operation, and finally “OK”.

Practice 1-3: Creating New Named Credentials

Overview

In this practice, you create the `credorcl` credential used for any connection as `SYS` user sharable in the database instance `orcl`.

Assumptions

You completed the practice 1-2 to add the `orcl` database instance as a new target monitored by Enterprise Manager Cloud Control.

Tasks

1. Navigate to Setup > Security > Named Credentials.
2. Click **Create**.
   a. Enter the following values, then complete the **Access Control** section:

Field	Choice or Value
General Properties	
Credential Name	**credorcl**
Credential description	**Credentials for Database**
Authenticating Target Type	**Database Instance**
Credential type	**Database Credentials**
Scope	**Target**
Target type	**Database Instance**
Target Name	**orcl (Click the magnifying glass to find orcl and select)**
Credential Properties	
Username	**SYS**
Password	**oracle_4U**
Confirm Password	**oracle_4U**
Role	**SYSDBA**

   b. Specify who can share, edit or even delete this shared credential using one of the three privileges (Full, Edit, View).

- SYS user with Full privilege will be able to use, edit and delete the credential.
- SYSTEM user with Edit privilege will be able to use and edit the credential.
  1) Click “Add Grant” then select the user SYS to be added in the Access Control list.
  2) Repeat this operation to add the user SYSTEM.
     By default, the selected users are granted the View privilege only.
  3) To grant Full privilege to SYS, select the SYS user and click “Change Privilege”. Choose Full and click OK.

4) To grant Edit privilege to SYSTEM, select the SYSTEM user and click “Change Privilege”. Choose Edit and click OK.

3. Test against the `orcl` database instance, click **Test and Save** until you get the following message: **Confirmation Credential Operation Successful**. This means that the credential was successful and saved.

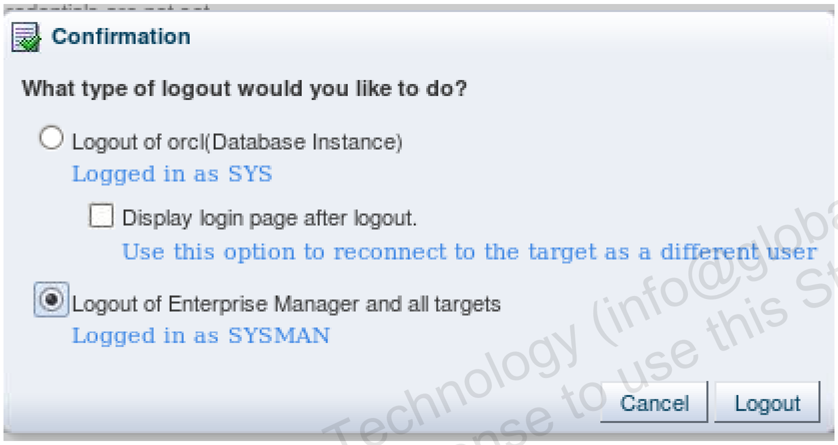
Practice 1-4: Testing the Named Credential

Overview

In this practice, you test the `credorcl` named credential to connect to `orcl` database.

Tasks

1. Test if the named credential works when you connect to the `orcl` target.
   Click **Targets** and then select **Databases**.
2. Choose `orcl`.
3. Click **Administration**, then **Storage** and then **Tablespaces.** The named credential `credorcl` is displayed.
4. Click **Login** if you accept this named credential to log in the `orcl` database else choose **New** to define new login username and password.
5. As soon as you click the Logout button, the following screenshot is displayed.



Confirmation

What type of logout would you like to do?

○ Logout of orcl(Database Instance)
Logged in as SYS

□ Display login page after logout.
Use this option to reconnect to the target as a different user

● Logout of Enterprise Manager and all targets
Logged in as SYSMAN

Choose “Logout of Enterprise Manager and all targets” and click the Logout button.

Practices for Lesson 2: Basics of Multitenant Container Database and Pluggable Databases

Chapter 2

Practices for Lesson 2: Overview

Practices Overview

In previous Oracle Database versions, you used to create, configure, and manage non-CDBs.

In Oracle Database 12*c*, you need to know how to create, configure, and manage multitenant container databases (CDBs) and pluggable databases (PDBs).

In this practice, you will explore new types of databases and get familiar with the architecture and structures of multitenant container databases (CDBs) and pluggable databases (PDBs).

Practice 2-1: Exploring CDB Architecture and Structures

Overview

In this practice, you will explore the architecture and structures of `cdb1` and its pluggable databases.

Tasks

1. Explore the `cdb1` instance, the background processes and the multitenant container database.
   a. Use the **`ps -ef|grep`** Unix command.

```
$ ps -ef|grep cdb1
oracle     378   375  0 18:05 ?        00:00:00 oraclecdb1
(DESCRIPTION=(LOCAL=YES)(ADDRESS=(PROTOCOL=beq)))
oracle     390     1  0 18:05 ?        00:00:00 ora_w001_cdb1
oracle    2711  2686  0 18:32 pts/2    00:00:00 grep cdb1
oracle   27530     1  0 13:24 ?        00:00:02 ora_pmon_cdb1
oracle   27534     1  0 13:24 ?        00:00:04 ora_psp0_cdb1
oracle   27538     1  1 13:24 ?        00:05:01 ora_vktm_cdb1
oracle   27544     1  0 13:24 ?        00:00:00 ora_gen0_cdb1
oracle   27548     1  0 13:24 ?        00:00:00 ora_mman_cdb1
oracle   27556     1  0 13:24 ?        00:00:00 ora_diag_cdb1
oracle   27560     1  0 13:24 ?        00:00:00 ora_ofsd_cdb1
oracle   27564     1  0 13:24 ?        00:00:00 ora_dbrm_cdb1
oracle   27568     1  0 13:24 ?        00:00:11 ora_dia0_cdb1
oracle   27572     1  0 13:24 ?        00:00:02 ora_dbw0_cdb1
oracle   27576     1  0 13:24 ?        00:00:01 ora_lgwr_cdb1
oracle   27580     1  0 13:24 ?        00:00:03 ora_ckpt_cdb1
oracle   27584     1  0 13:24 ?        00:00:00 ora_smon_cdb1
oracle   27588     1  0 13:24 ?        00:00:00 ora_reco_cdb1
oracle   27592     1  0 13:24 ?        00:00:00 ora_lreg_cdb1
oracle   27596     1  0 13:24 ?        00:00:15 ora_mmon_cdb1
oracle   27600     1  0 13:24 ?        00:00:06 ora_mmnl_cdb1
oracle   27604     1  0 13:24 ?        00:00:00 ora_d000_cdb1
oracle   27608     1  0 13:24 ?        00:00:00 ora_s000_cdb1
oracle   27630     1  0 13:25 ?        00:00:00 ora_tmon_cdb1
oracle   27634     1  0 13:25 ?        00:00:00 ora_tt00_cdb1
oracle   27638     1  0 13:25 ?        00:00:00 ora_fbda_cdb1
oracle   27642     1  0 13:25 ?        00:00:00 ora_aqpc_cdb1
oracle   27651     1  0 13:25 ?        00:00:00 ora_p000_cdb1
oracle   27659     1  0 13:25 ?        00:00:00 ora_p001_cdb1
oracle   27666     1  0 13:25 ?        00:00:00 ora_p002_cdb1
oracle   27670     1  0 13:25 ?        00:00:00 ora_p003_cdb1
oracle   27682     1  0 13:25 ?        00:00:09 ora_cjq0_cdb1
oracle   27734      1  0 13:25 ?        00:00:00 ora_qm01_cdb1
oracle   27738      1  0 13:25 ?        00:00:00 ora_q001_cdb1
oracle   27742      1  0 13:25 ?        00:00:00 ora_q002_cdb1
oracle   27750      1  0 13:25 ?        00:00:00 ora_smco_cdb1
oracle   31695      1  0 17:05 ?        00:00:00 ora_w002_cdb1
$
```

b. Connect to the multitenant container database `cdb1`.

```
$ . oraenv
ORACLE_SID = [orcl] ? cdb1
The Oracle base remains unchanged with value /u01/app/oracle

$ sqlplus / as sysdba

Connected to:
Oracle Database 12c Enterprise Edition Release 12.1.0.0.2 -
64bit Production
With the Partitioning, OLAP, Data Mining, Real Application
Testing

SQL>
```

c. Check if the database is a multitenant container database.

```
SQL> select name, cdb, con_id from v$database;

NAME      CDB    CON_ID
--------- --- ----------
CDB1      YES         0

SQL>
```

d. Check the instance name.

```
SQL> select INSTANCE_NAME, STATUS, CON_ID from v$instance;

INSTANCE_NAME    STATUS           CON_ID
---------------- ------------ ----------
cdb1             OPEN                  0

SQL> EXIT
$
```

2. Explore the services.

a. Start the listener if not yet started.

```
$ lsnrctl status
```

```
LSNRCTL for Linux: Version 12.1.0.0.2 - Production on 09-JUL-
2012 02:57:38


Copyright (c) 1991, 2012, Oracle.  All rights reserved.


Connecting to
(DESCRIPTION=(ADDRESS=(PROTOCOL=IPC)(KEY=EXTPROC1521)))
STATUS of the LISTENER
------------------------
Alias                     LISTENER
Version                   TNSLSNR for Linux: Version 12.1.0.0.2
- Production
Start Date                10-JUL-2012 00:15:19
Uptime                    0 days 2 hr. 42 min. 19 sec
Trace Level               off
Security                  ON: Local OS Authentication
SNMP                      OFF
Listener Parameter File
/u01/app/oracle/product/12.1.0/dbhome_1/network/admin/listener.o
ra
Listener Log File
/u01/app/oracle/diag/tnslsnr/yourserver/listener/alert/log.xml
Listening Endpoints Summary...
  (DESCRIPTION=(ADDRESS=(PROTOCOL=ipc)(KEY=EXTPROC1521)))
(DESCRIPTION=(ADDRESS=(PROTOCOL=tcp)(HOST=yourserver)(PORT=1521)
))
(DESCRIPTION=(ADDRESS=(PROTOCOL=tcps)(HOST=yourserver)(PORT=5500
))(Security=(my_wallet_directory=/u01/app/oracle/admin/orcl/xdb_
wallet)) (Presentation=HTTP)(Session=RAW))
(DESCRIPTION=(ADDRESS=(PROTOCOL=tcps)(HOST=yourserver)(PORT=5502
))(Security=(my_wallet_directory=/u01/app/oracle/admin/cdb1/xdb_
wallet)) (Presentation=HTTP)(Session=RAW))
Services Summary...
Service "cdb1" has 1 instance(s).
  Instance "cdb1", status READY, has 1 handler(s) for this
service...
Service "cdb1XDB" has 1 instance(s).
  Instance "cdb1", status READY, has 1 handler(s) for this
service...
Service "em12rep" has 1 instance(s).
  Instance "em12rep", status READY, has 1 handler(s) for this
service...
Service "em12repXDB" has 1 instance(s).
  Instance "em12rep", status READY, has 1 handler(s) for this
service...
Service "orcl" has 1 instance(s).
```

```
  Instance "orcl", status READY, has 1 handler(s) for this
service...
Service "orcl2" has 1 instance(s).
  Instance "orcl2", status READY, has 1 handler(s) for this
service...
Service "orcl2XDB" has 1 instance(s).
  Instance "orcl2", status READY, has 1 handler(s) for this
service...
Service "orclXDB" has 1 instance(s).
  Instance "orcl", status READY, has 1 handler(s) for this
service...
Service "pdb1_1" has 1 instance(s).
  Instance "cdb1", status READY, has 1 handler(s) for this
service...
The command completed successfully
$
```

The listener is already started. If it were not started, you would use the following command to start the listener:

```
$ lsnrctl start


LSNRCTL for Linux: Version 12.1.0.0.2 - Production on 09-JUL-
2012 03:08:50


Copyright (c) 1991, 2012, Oracle.  All rights reserved.


Starting /u01/app/oracle/product/12.1.0/dbhome_1/bin/tnslsnr:
please wait...


TNSLSNR for Linux: Version 12.1.0.0.2 - Production
System parameter file is
/u01/app/oracle/product/12.1.0/dbhome_1/network/admin/listener.o
ra
Log messages written to
/u01/app/oracle/diag/tnslsnr/yourserver/listener/alert/log.xml
Listening on:
(DESCRIPTION=(ADDRESS=(PROTOCOL=ipc)(KEY=EXTPROC1521)))
Listening on:
(DESCRIPTION=(ADDRESS=(PROTOCOL=tcp)(HOST=yourserver)(PORT=1521)
))


Connecting to
(DESCRIPTION=(ADDRESS=(PROTOCOL=IPC)(KEY=EXTPROC1521)))
STATUS of the LISTENER
------------------------
Alias                     LISTENER
```

```
Version                   TNSLSNR for Linux: Version 12.1.0.0.2
- Production
Start Date                09-OCT-2012 03:08:50
Uptime                    0 days 0 hr. 0 min. 0 sec
Trace Level               off
Security                  ON: Local OS Authentication
SNMP                      OFF
Listener Parameter File
/u01/app/oracle/product/12.1.0/dbhome_1/network/admin/listener.o
ra
Listener Log File
/u01/app/oracle/diag/tnslsnr/yourserver/listener/alert/log.xml
Listening Endpoints Summary...
  (DESCRIPTION=(ADDRESS=(PROTOCOL=ipc)(KEY=EXTPROC1521)))

(DESCRIPTION=(ADDRESS=(PROTOCOL=tcp)(HOST=yourserver)(PORT=1521)
))
The listener supports no services
The command completed successfully
$
```

b. Check services.

```
$ lsnrctl services


LSNRCTL for Linux: Version 12.1.0.0.2 - Production on 06-SEP-
2012 23:29:20


Copyright (c) 1991, 2012, Oracle.  All rights reserved.


Connecting to
(DESCRIPTION=(ADDRESS=(PROTOCOL=IPC)(KEY=EXTPROC1521)))
Services Summary...
Service "cdb1" has 1 instance(s).
  Instance "cdb1", status READY, has 1 handler(s) for this
service...
    Handler(s):
      "DEDICATED" established:2 refused:0 state:ready
         LOCAL SERVER
Service "cdb1XDB" has 1 instance(s).
  Instance "cdb1", status READY, has 1 handler(s) for this
service...
    Handler(s):
      "D000" established:0 refused:0 current:0 max:1022
state:ready
         DISPATCHER <machine: yourserver, pid: 27840>
         (ADDRESS=(PROTOCOL=tcp)(HOST=yourserver)(PORT=29863))
Service "em12rep" has 1 instance(s).
  Instance "em12rep", status READY, has 1 handler(s) for this
service...
    Handler(s):
      "DEDICATED" established:2748 refused:0 state:ready
         LOCAL SERVER
Service "em12repXDB" has 1 instance(s).
  Instance "em12rep", status READY, has 1 handler(s) for this
service...
    Handler(s):
      "D000" established:0 refused:0 current:0 max:1022
state:ready
         DISPATCHER <machine: yourserver, pid: 18255>
         (ADDRESS=(PROTOCOL=tcp)(HOST=yourserver)(PORT=59402))
Service "orcl" has 1 instance(s).
  Instance "orcl", status READY, has 1 handler(s) for this
service...
    Handler(s):
      "DEDICATED" established:251 refused:0 state:ready
         LOCAL SERVER
Service "orcl2" has 1 instance(s).
  Instance "orcl2", status READY, has 1 handler(s) for this
service...
    Handler(s):
      "DEDICATED" established:4 refused:0 state:ready
         LOCAL SERVER
Service "orcl2XDB" has 1 instance(s).
  Instance "orcl2", status READY, has 1 handler(s) for this
service...
    Handler(s):
      "D000" established:6 refused:0 current:0 max:1022
state:ready
         DISPATCHER <machine: yourserver, pid: 23615>
         (ADDRESS=(PROTOCOL=tcp)(HOST=yourserver)(PORT=50200))
Service "orclXDB" has 1 instance(s).
  Instance "orcl", status READY, has 1 handler(s) for this
service...
    Handler(s):
      "D000" established:0 refused:0 current:0 max:1022
state:ready
         DISPATCHER <machine: yourserver, pid: 30821>
         (ADDRESS=(PROTOCOL=tcp)(HOST=yourserver)(PORT=27384))
Service "pdb1_1" has 1 instance(s).
```

```
  Instance "cdb1", status READY, has 1 handler(s) for this
service...
    Handler(s):
      "DEDICATED" established:2 refused:0 state:ready
         LOCAL SERVER
The command completed successfully
$
```

c. List the services automatically created for each container.

```
$ sqlplus / as sysdba

Connected to:
Oracle Database 12c Enterprise Edition Release 12.1.0.0.2 -
64bit Production
With the Partitioning, OLAP, Advanced Analytics and Real
Application Testing options

SQL> col name format A20
SQL> select name, con_id from v$services;

NAME                     CON_ID
-------------------- ----------
pdb1_1                        3
cdb1XDB                       1
cdb1                          1
SYS$BACKGROUND                1
SYS$USERS                     1

SQL>
```

Notice that the `PDB$SEED` service is not listed. No one should connect to this service because there should be no operations performed on this container. It is reserved as a template to create other PDBs.

3. Display the pluggable databases.

   a. Use the new view `V$PDBS`.

```
SQL> select CON_ID, NAME, OPEN_MODE from v$pdbs;

    CON_ID NAME                           OPEN_MODE
---------- ------------------------------ ----------
         2 PDB$SEED                       READ ONLY
         3 PDB1_1                         READ WRITE


SQL>
```

Notice that the seed PDB is in `READ ONLY` open mode.

b. Use the new command `SHOW CON_NAME` and `CON_ID` to know which container you are connected to.

```
SQL> show con_name

CON_NAME
------------------------------
CDB$ROOT

SQL> show con_id

CON_ID
------------------------------
1

SQL>
```

You can also use `SYS_CONTEXT` function to view the `CON_NAME` and `CON_ID` attributes of your session context.

```
SELECT sys_context('userenv','CON_NAME') from dual;
SELECT sys_context('userenv','CON_ID') from dual;
```

4. View some of the new family of views CDB_*xxx*:

```
SQL> col PDB_NAME format a8
SQL> col CON_ID format 99
SQL> select PDB_ID, PDB_NAME, DBID, GUID, CON_ID
  2  from cdb_pdbs;

PDB_ID PDB_NAME       DBID GUID
CON_ID
---------- -------- ---------- --------------------------------     -
-----
     3 PDB1_1   3624951709 C39AE2177B2E530EE043160200C043C3
     1
     2 PDB$SEED 4029862422 C2CBF2921BD1161FE04388AAE80AB141
     1


SQL>
```

The `PDB_ID` number 2 is always assigned to the seed PDB because it is the second container to be created after the root container (`CON_ID` 1).

5. Check all files of the CDB.

a. View the redo log files of the CDB.

```
SQL> col MEMBER format A40
SQL> select GROUP#, CON_ID, MEMBER from v$logfile;
```

```
    GROUP# CON_ID MEMBER
---------- ------ ----------------------------------------
         3      0 /u01/app/oracle/oradata/cdb1/redo03.log
         2      0 /u01/app/oracle/oradata/cdb1/redo02.log
         1      0 /u01/app/oracle/oradata/cdb1/redo01.log

SQL>
```

b. View the control files of the CDB.

```
SQL> col NAME format A60
SQL> select NAME , CON_ID from v$controlfile;

NAME                                                         CON_ID
------------------------------------------------------------ ------
/u01/app/oracle/oradata/cdb1/control01.ctl                        0
/u01/app/oracle/fast_recovery_area/cdb1/control02.ctl             0

SQL>
```

c. View all data files of the CDB, including those of the root and all PDBs.

1) With CDB_DATA_FILES view:

```
SQL> col file_name format A50
SQL> col tablespace_name format A8
SQL> col file_id format 9999
SQL> col con_id  format 999
SQL> select FILE_NAME, TABLESPACE_NAME, FILE_ID, con_id
  2  from   cdb_data_files  order  by con_id ;

FILE_NAME                                          TABLESPA
FILE_ID CON_ID
-------------------------------------------------- -------- ----
--- ------
/u01/app/oracle/oradata/cdb1/users01.dbf           USERS
6      1
/u01/app/oracle/oradata/cdb1/undotbs01.dbf         UNDOTBS1
4      1
/u01/app/oracle/oradata/cdb1/sysaux01.dbf          SYSAUX
3      1
/u01/app/oracle/oradata/cdb1/system01.dbf          SYSTEM
1      1
/u01/app/oracle/oradata/cdb1/pdbseed/system01.dbf  SYSTEM
5      2
/u01/app/oracle/oradata/cdb1/pdbseed/sysaux01.dbf  SYSAUX
7      2
/u01/app/oracle/oradata/cdb1/pdb1_1/system01.dbf   SYSTEM
8      3
```

```
/u01/app/oracle/oradata/cdb1/pdb1_1/sysaux01.dbf   SYSAUX
9      3
/u01/app/oracle/oradata/cdb1/pdb1_1/SAMPLE_SCHEMA_ USERS
10     3
users01.dbf
/u01/app/oracle/oradata/cdb1/pdb1_1/example01.dbf  EXAMPLE
11     3


10 rows selected.


SQL>
```

2) With `ls` Unix command:

```
SQL> !ls -l $ORACLE_BASE/oradata/cdb1
total 2575988
-rw-r----- 1 oracle oinstall  17874944 Sep  6 23:38
control01.ctl
drwxr-xr-x 2 oracle oinstall      4096 Sep  5 10:54 pdb1_1
drwxr-x--- 2 oracle oinstall      4096 Sep  5 10:37 pdbseed
-rw-r----- 1 oracle oinstall  52429312 Sep  6 23:38 redo01.log
-rw-r----- 1 oracle oinstall  52429312 Sep  6 20:01 redo02.log
-rw-r----- 1 oracle oinstall  52429312 Sep  6 22:23 redo03.log
-rw-r----- 1 oracle oinstall 849354752 Sep  6 23:35 sysaux01.dbf
-rw-r----- 1 oracle oinstall 828383232 Sep  6 23:35 system01.dbf
-rw-r----- 1 oracle oinstall 571482112 Sep  6 23:18 temp01.dbf
-rw-r----- 1 oracle oinstall 246423552 Sep  6 23:36
undotbs01.dbf
-rw-r----- 1 oracle oinstall   5251072 Sep  6 22:29 users01.dbf
```

```
SQL> !ls -l $ORACLE_BASE/oradata/cdb1/pdbseed
total 985064
-rw-r----- 1 oracle oinstall  88088576 Sep  5 10:50
pdbseed_temp01.dbf
-rw-r----- 1 oracle oinstall 671096832 Sep  5 10:50 sysaux01.dbf
-rw-r----- 1 oracle oinstall 262152192 Sep  5 10:50 system01.dbf
SQL>
```

There are only the `SYSTEM` and `SYSAUX` data files and a temp file for the seed PDB.

d. Still connected to the root, now use `DBA_DATA_FILES` view.

```
SQL> col file_name format A42
SQL> select FILE_NAME, TABLESPACE_NAME, FILE_ID
  2  from   dba_data_files;

FILE_NAME                                  TABLESPA FILE_ID
------------------------------------------ -------- ---------
```

```
/u01/app/oracle/oradata/cdb1/users01.dbf   USERS
    6
/u01/app/oracle/oradata/cdb1/undotbs01.dbf UNDOTBS1         4
/u01/app/oracle/oradata/cdb1/sysaux01.dbf  SYSAUX           3
/u01/app/oracle/oradata/cdb1/system01.dbf  SYSTEM           1

SQL>
```

Notice that only the root data files are listed.

e. Now use `V$TABLESPACE` and `V$DATAFILE` view.

```
SQL> col NAME format A12
SQL> select FILE#, ts.name, ts.ts#, ts.con_id
  2  from v$datafile d, v$tablespace  ts
  3  where d.ts#=ts.ts#
  4  and   d.con_id=ts.con_id
  5  order by 4,3;

     FILE# NAME                TS# CON_ID
---------- ------------ ---------- ------
         1 SYSTEM               0    1
         3 SYSAUX               1    1
         4 UNDOTBS1             2    1
         6 USERS                4    1
         5 SYSTEM               0    2
         7 SYSAUX               1    2
         8 SYSTEM               0    3
         9 SYSAUX               1    3
        10 USERS                3    3
        11 EXAMPLE              4    3

10 rows selected.

SQL>
```

f. List the temp files of the CDB.

```
SQL> col file_name format A47
SQL> select FILE_NAME, TABLESPACE_NAME, FILE_ID
  2  from cdb_temp_files;

FILE_NAME                                                TABLESPA
FILE_ID
-------------------------------------------------------- --------
-------
```

```
/u01/app/oracle/oradata/cdb1/temp01.dbf                  TEMP
1
/u01/app/oracle/oradata/cdb1/pdbseed/pdbseed_temp01.dbf TEMP
2
/u01/app/oracle/oradata/cdb1/pdb1_1/pdb1_1_temp01.dbf   TEMP
3


SQL>
```

6. List all users created.

   a. Verify that the SYSTEM user is created.

```
SQL> col username format A22
SQL> select username, common, con_id from cdb_users
  2  where username ='SYSTEM';


USERNAME               COM CON_ID
---------------------- --- ------
SYSTEM                 YES      1
SYSTEM                 YES      2
SYSTEM                 YES      3


SQL>
```

Notice that the user SYSTEM exists in all containers as a common user.

   b. List all common users of the CDB.

```
SQL> select distinct username from cdb_users
  2  where common ='YES';


USERNAME
--------------------
DVF
MDSYS
XS$NULL
SYSKM
APEX_040100
DIP
SPATIAL_WFS_ADMIN_USR
FLOWS_FILES
SYSBACKUP
CTXSYS
OUTLN
SPATIAL_CSW_ADMIN_USR
GSMUSER
OLAPSYS
SYSTEM
```

```
ORACLE_OCM
DVSYS
AUDSYS
ORDSYS
DBSNMP
OJVMSYS
GSMADMIN_INTERNAL
MDDATA
APEX_PUBLIC_USER
ORDPLUGINS
APPQOSSYS
GSMCATUSER
ORDDATA
SYSDG
XDB
SYS
WMSYS
LBACSYS
ANONYMOUS
SI_INFORMTN_SCHEMA

35 rows selected.

SQL>
```

c. List all local users of the CDB.

```
SQL> select distinct username, con_id from cdb_users
  2  where common ='NO';

USERNAME                 CON_ID
-------------------- ------
SCOTT                         3
BI                            3
PM                            3
IX                            3
SH                            3
OE                            3
HR                            3
PDBADMIN                      3

8 rows selected.

SQL>
```

d. List local users in the root.

```
SQL> select username, con_id from cdb_users
  2  where common ='NO';

USERNAME                CON_ID
-------------------- ------
SCOTT                        3
BI                           3
PM                           3
IX                           3
SH                           3
OE                           3
HR                           3
PDBADMIN                     3

8 rows selected.

SQL>
```

Notice that there is no local user in the root container because it is impossible to create any local user in the root.

7. List all roles and privileges of the CDB.

   a. List all roles of the CDB.

```
SQL> col role format A30
SQL> select role, common, con_id from cdb_roles;

ROLE                           COM CON_ID
------------------------------ --- ------
CONNECT                        YES      1
RESOURCE                       YES      1
DBA                            YES      1
AUDIT_ADMIN                    YES      1
AUDIT_VIEWER                   YES      1
SELECT_CATALOG_ROLE            YES      1
EXECUTE_CATALOG_ROLE           YES      1
DELETE_CATALOG_ROLE            YES      1
PROF_ADMIN                     YES      1
EXP_FULL_DATABASE              YES      1
IMP_FULL_DATABASE              YES      1
CDB_DBA                        YES      1
…
DV_XSTREAM_ADMIN               YES      2
DV_GOLDENGATE_REDO_ACCESS      YES      2
DV_AUDIT_CLEANUP               YES      2
DV_REALM_OWNER                 YES      2
PDB_DBA                        YES      2
…
DV_AUDIT_CLEANUP               YES      3
DV_REALM_RESOURCE              YES      3
DV_REALM_OWNER                 YES      3
PDB_DBA                        YES      3

252 rows selected.

SQL>
```

Notice that there is no local role in the root container because it is impossible to create any local role in the root.

b. Check that the privileges are neither common nor local by nature.

```
SQL> desc sys.system_privilege_map
Name                              Null?    Type
 ----------------------------- -------- ----------------------
 PRIVILEGE                        NOT NULL NUMBER
 NAME                             NOT NULL VARCHAR2(40)
 PROPERTY                         NOT NULL NUMBER

SQL> desc sys.table_privilege_map
Name                              Null?    Type
 ----------------------------- -------- ----------------------
 PRIVILEGE                        NOT NULL NUMBER
 NAME                             NOT NULL VARCHAR2(40)

SQL>
```

Notice that there is no COMMON column.

c. Verify that the privilege, when granted, becomes a common or local privilege.

```
SQL> desc CDB_SYS_PRIVS
Name                                Null?    Type
 ------------------------------- -------- --------------------
 GRANTEE                                    VARCHAR2(128)
 PRIVILEGE                                  VARCHAR2(40)
 ADMIN_OPTION                               VARCHAR2(3)
 COMMON                                     VARCHAR2(3)
 CON_ID                                     NUMBER

SQL> desc CDB_TAB_PRIVS
Name                               Null?    Type
------------------------------- -------- --------------------
 GRANTEE                                   VARCHAR2(128)
 OWNER                                     VARCHAR2(128)
 TABLE_NAME                                VARCHAR2(128)
 GRANTOR                                   VARCHAR2(128)
 PRIVILEGE                                 VARCHAR2(40)
 GRANTABLE                                 VARCHAR2(3)
 HIERARCHY                                 VARCHAR2(3)
 COMMON                                    VARCHAR2(3)
 TYPE                                      VARCHAR2(24)
 CON_ID                                    NUMBER

SQL>
```

There is a **COMMON** column.

d. Notice that the role, though common or local depending on how the role was created is also, like privileges, either granted commonly or locally.

```
SQL> col grantee format A10
SQL> col granted_role format A28
SQL> select grantee, granted_role, common, con_id
  2  from cdb_role_privs
  3  where grantee='SYSTEM';

GRANTEE    GRANTED_ROLE                 COM CON_ID
---------- ---------------------------- --- ------
SYSTEM     DBA                          YES      1
SYSTEM     AQ_ADMINISTRATOR_ROLE        YES      1
SYSTEM     DBA                          YES      2
SYSTEM     AQ_ADMINISTRATOR_ROLE        YES      2
SYSTEM     DBA                          YES      3
SYSTEM     AQ_ADMINISTRATOR_ROLE        YES      3

6 rows selected.
SQL> EXIT
$
```

Practices for Lesson 3: Creating a Multitenant Container Database and Pluggable Databases

Chapter 3

Practices for Lesson 3

Practices Overview

In this practice you will create a new CDB named cdb2 with DBCA with no PDB except the seed.

After the CDB creation is completed, check the physical and logical structures of the new CDB.

Then, you will create several PDBs using different methods.

- Create pdb2_1 from seed in cdb2.
- Clone pdb2_2 in cdb2 from pdb2_1.
- Plug the non-CDB orcl2 into the CDB cdb2 as pdb_orcl2.
- Merge the two CDBs cdb1 and cdb2 into cdb2, and optionally drop the database cdb1 (optional practice).

Finally, you drop the pdb2_3 using either DBCA or SQL Developer or SQL*Plus after the creation of this PDB.

Practice 3-1: Creating a New CDB

Overview

In this practice, you will create a new CDB named `cdb2` with DBCA.

Assumptions

The created CDB `cdb1` already exists.

Tasks

1. Create a CDB named `cdb2` using DBCA. First release resources held by other instances, shutting down the `orcl`, `orcl2` and `cdb1` instances.
   a. Shut down `orcl`.

```
$ . oraenv
ORACLE_SID = [cdb1] ? orcl
The Oracle base remains unchanged with value /u01/app/oracle

$ sqlplus / as sysdba

Connected to:
Oracle Database 12c Enterprise Edition Release 12.1.0.0.2 -
64bit Production
With the Partitioning, OLAP, Data Mining, Real Application
Testing

SQL> SHUTDOWN IMMEDIATE
Database closed.
Database dismounted.
ORACLE instance shut down.
SQL> EXIT
$
```

   b. Shut down `orcl2`.

```
$ . oraenv
ORACLE_SID = [orcl] ? orcl2
The Oracle base remains unchanged with value /u01/app/oracle

$ sqlplus / as sysdba

Connected to:
Oracle Database 12c Enterprise Edition Release 12.1.0.0.2 -
64bit Production
With the Partitioning, OLAP, Data Mining, Real Application
Testing
```

```
SQL> SHUTDOWN IMMEDIATE
Database closed.
Database dismounted.
ORACLE instance shut down.
SQL> EXIT
$
```

c. Shut down `cdb1`.

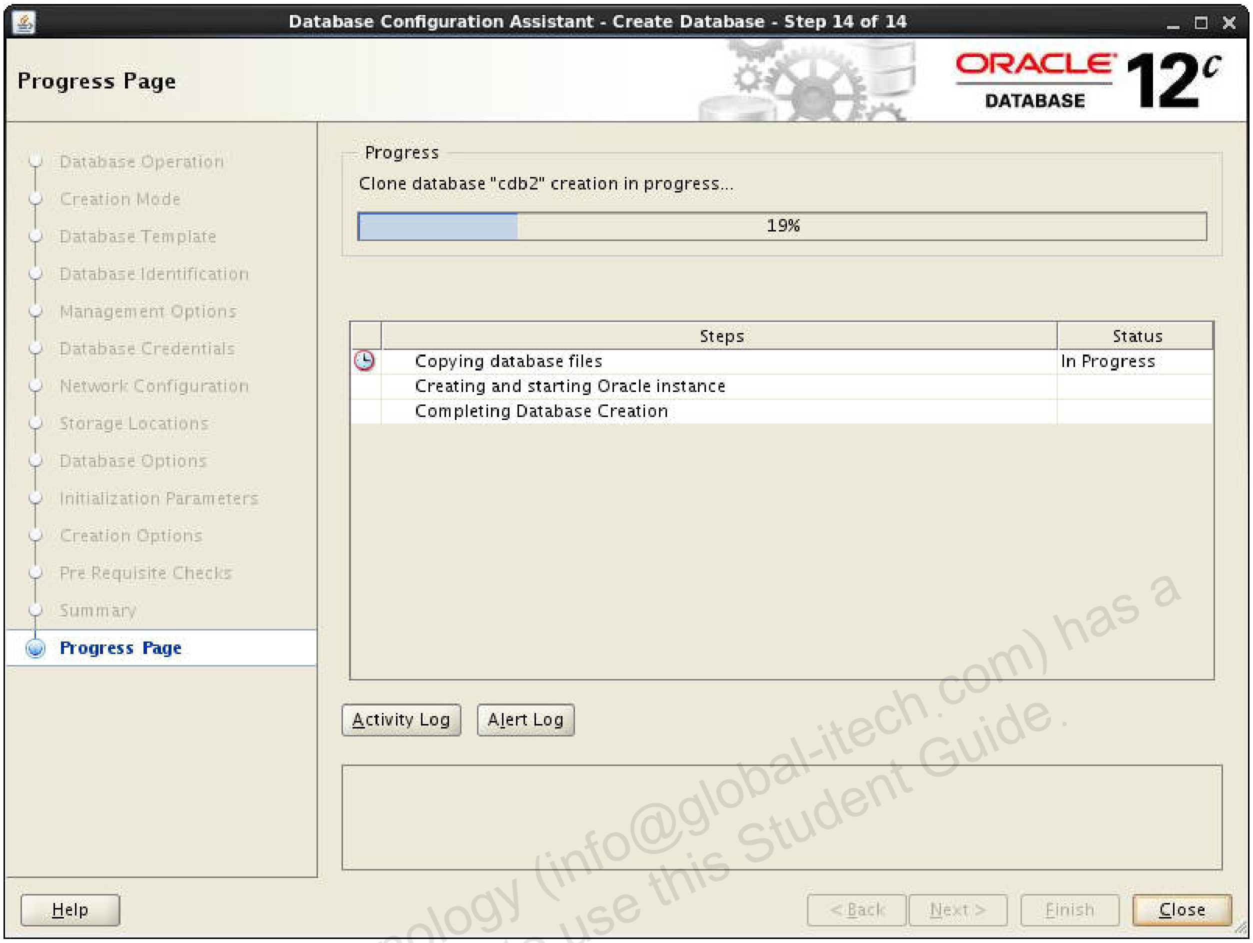
```
$ . oraenv
ORACLE_SID = [orcl2] ? cdb1
The Oracle base remains unchanged with value /u01/app/oracle

$ sqlplus / as sysdba

Connected to:
Oracle Database 12c Enterprise Edition Release 12.1.0.0.2 -
64bit Production
With the Partitioning, OLAP, Data Mining, Real Application
Testing

SQL> SHUTDOWN IMMEDIATE
Database closed.
Database dismounted.
ORACLE instance shut down.
SQL> EXIT
$
```

d. Start dbca and perform the following steps.

```
$ dbca
```

Step	Window/Page Description	Choices or Values
a.	Step 1: Database Operation	Select “Create Database”. Click Next.
b.	Step 2: Creation Mode	Select “Advanced Mode”. Click Next.
c.	Step 3: Database Template	Select “General Purpose or Transaction Processing”. Click Next.
d.	Step 4: Database Identification	Enter Global Database Name: **cdb2** SID: cdb2 Select **“Create As Container Database”** Select **“Create An Empty Container Database”** Click Next.

Step	Window/Page Description	Choices or Values
e.	Step 5: Management Options	Deselect “Configure Enterprise Manager (EM) Database Express”. Click Next.
f.	Step 6: Database Credentials	Select “Use same Administrative password…” Enter: Password: `oracle_4U` Confirm password: `oracle_4U` Click Next.
g.	Step 7: Network Configuration	Listener Selection: Click Next
h.	Step 8: Storage Locations	Confirm Storage type is “File System”. Select “Use Common Location for All Database Files”. Click Next.
i.	Step 9: Database Options	Click Next.
j.	Step 10: Initialization Parameters	Select “Character Sets”. Select “Use Unicode (**AL32UTF8**)”. Click Next.
k.	Step 11: Creation Option	Select “Create Database”. Click Next.
l.	Step 12: Pre Requisite Checks	Click Next.
m.	Step 13: Summary	Click Finish.
n.	Step 14: Progress Page	On the Database Configuration Assistant page (for password management) click Exit. Click Close.



Database Configuration Assistant - Create Database - Step 14 of 14

Progress Page

Progress

Clone database "cdb2" creation in progress...

19%

	Steps	Status
◷	Copying database files	In Progress
	Creating and starting Oracle instance	
	Completing Database Creation	

Practice 3-2: Exploring CDB and PDB Structures

Overview

In this practice, you check the physical and logical structures of the new CDB `cdb2` and its seed PDB.

Tasks

1. Connect to the multitenant container database `cdb2`.

```
$ . oraenv
ORACLE_SID = [cdb1] ? cdb2
The Oracle base remains unchanged with value /u01/app/oracle

$ sqlplus / as sysdba

Connected to:
Oracle Database 12c Enterprise Edition Release 12.1.0.0.2 -
64bit Production
With the Partitioning, OLAP, Data Mining and Real Application
Testing options

SQL>
```

a. Check if the database is a multitenant container database.

```
SQL> SELECT name, cdb, con_id from v$database;

NAME      CDB     CON_ID
--------- --- ----------
CDB2      YES          0

SQL>
```

b. Check the instance name.

```
SQL> SELECT INSTANCE_NAME, STATUS, CON_ID from v$instance;

INSTANCE_NAME    STATUS           CON_ID
---------------- ------------ ----------
cdb2             OPEN                  0

SQL> EXIT
$
```

2. Explore the services.

a. Check services.

```
$ lsnrctl status
```

```
LSNRCTL for Linux: Version 12.1.0.0.2 - Production on 07-SEP-
2012 01:10:16


Copyright (c) 1991, 2012, Oracle.  All rights reserved.


Connecting to
(DESCRIPTION=(ADDRESS=(PROTOCOL=IPC)(KEY=EXTPROC1521)))
STATUS of the LISTENER
------------------------
Alias                     LISTENER
Version                   TNSLSNR for Linux: Version 12.1.0.0.2
- Production
Start Date                14-SEP-2012 03:04:56
Uptime                    16 days 21 hr. 48 min. 35 sec
Trace Level               off
Security                  ON: Local OS Authentication
SNMP                      OFF
Listener Parameter File
/u01/app/oracle/product/12.1.0/dbhome_1/network/admin/listener.o
ra
Listener Log File
/u01/app/oracle/diag/tnslsnr/yourserver/listener/alert/log.xml
Listening Endpoints Summary...
  (DESCRIPTION=(ADDRESS=(PROTOCOL=ipc)(KEY=EXTPROC1521)))
(DESCRIPTION=(ADDRESS=(PROTOCOL=tcp)(HOST=yourserver)(PORT=1521)
))
(DESCRIPTION=(ADDRESS=(PROTOCOL=tcp)(HOST=yourserver)(PORT=5501)
)(Presentation=HTTP)(Session=RAW))
(DESCRIPTION=(ADDRESS=(PROTOCOL=tcp)(HOST=yourserver)(PORT=5502)
)(Presentation=HTTP)(Session=RAW))
(DESCRIPTION=(ADDRESS=(PROTOCOL=tcp)(HOST=yourserver)(PORT=5500)
)(Presentation=HTTP)(Session=RAW))
Services Summary...
Service "cdb2" has 1 instance(s).
  Instance "cdb2", status READY, has 1 handler(s) for this
service...
Service "cdb2XDB" has 1 instance(s).
  Instance "cdb2", status READY, has 1 handler(s) for this
service...
Service "em12rep" has 1 instance(s).
  Instance "em12rep", status READY, has 1 handler(s) for this
service...
Service "em12repXDB" has 1 instance(s).
  Instance "em12rep", status READY, has 1 handler(s) for this
service...
The command completed successfully
```

```
$
```

b. List the services automatically created for each container.

```
$ sqlplus / as sysdba


Connected to:
Oracle Database 12c Enterprise Edition Release 12.1.0.1.0 -
64bit Production
With the Partitioning, OLAP, Advanced Analytics and Real
Application Testing options
SQL> col name format A20
SQL> SELECT name, con_id from v$services;

NAME                     CON_ID
-------------------- ----------
cdb2XDB                       1
cdb2                          1
SYS$BACKGROUND                1
SYS$USERS                     1

SQL>
```

Notice that PDB$SEED service is not listed. No one should connect to this service because there should be no operation performed on this container. It is reserved as a template to create other PDBs.

3. Display the pluggable databases. Use a new view V$PDBS.

```
SQL> SELECT CON_ID, NAME, OPEN_MODE from v$pdbs;

    CON_ID NAME                           OPEN_MODE
---------- ------------------------------ ----------
         2 PDB$SEED                       READ ONLY

SQL>
```

Notice that the seed PDB is in READ ONLY open mode.

4. View new family of views CDB_*xxx*:

```
SQL> connect / as sysdba
Connected.
SQL> col PDB_NAME format a8
SQL> col CON_ID format 999999
SQL> SELECT PDB_ID, PDB_NAME, DBID, GUID, CON_ID
  2  from cdb_pdbs order by 1;

PDB_ID PDB_NAME       DBID GUID                             CON_ID
------ -------- ---------- -------------------------------- ------
```

```
     2 PDB$SEED 4012275228 203F5F3EDB7F0000000000000000      1

SQL>
```

5. Check all files of the CDB.

   a. View the redo log files of the CDB.

```
SQL> col MEMBER format A42
SQL> SELECT GROUP#, MEMBER, CON_ID from v$logfile;

   GROUP# MEMBER                                     CON_ID
---------- ------------------------------------------ -------
         3 /u01/app/oracle/oradata/cdb2/redo03.log          0
         2 /u01/app/oracle/oradata/cdb2/redo02.log          0
         1 /u01/app/oracle/oradata/cdb2/redo01.log          0

SQL>
```

   b. View the control files of the CDB.

```
SQL> col name format A55
SQL> SELECT name, con_id from v$controlfile;

NAME                                                     CON_ID
------------------------------------------------------- -------
/u01/app/oracle/oradata/cdb2/control01.ctl                    0
/u01/app/oracle/fast_recovery_area/cdb2/control02.ctl         0

SQL>
```

   c. View all data files of the CDB, including those of the root and all PDBs, with CDB_DATA_FILES view.

```
SQL> col file_name format A65
SQL> SELECT FILE_NAME, TABLESPACE_NAME, FILE_ID, con_id
  2  from   cdb_data_files
  3  order  by con_id ;

FILE_NAME
----------------------------------------------------------------
-
TABLESPACE_NAME                   FILE_ID  CON_ID
------------------------------ ---------- -------
/u01/app/oracle/oradata/cdb2/users01.dbf
USERS                                   6       1

/u01/app/oracle/oradata/cdb2/undotbs01.dbf
UNDOTBS1                                4       1
```

```
/u01/app/oracle/oradata/cdb2/sysaux01.dbf
SYSAUX                                  3       1

/u01/app/oracle/oradata/cdb2/system01.dbf
SYSTEM                                  1       1

/u01/app/oracle/oradata/cdb2/pdbseed/system01.dbf
SYSTEM                                  5       2

/u01/app/oracle/oradata/cdb2/pdbseed/sysaux01.dbf
SYSAUX                                  7       2

6 rows selected.

SQL>
```

d. Still connected to the root, now use `DBA_DATA_FILES` view.

```
SQL> col file_name format A42
SQL> col tablespace_name format A10
SQL> SELECT FILE_NAME, TABLESPACE_NAME, FILE_ID
  2 from   dba_data_files;

FILE_NAME                                  TABLESPACE FILE_ID
------------------------------------------ ---------- -------
/u01/app/oracle/oradata/cdb2/users01.dbf   USERS            6
/u01/app/oracle/oradata/cdb2/undotbs01.dbf UNDOTBS1         4
/u01/app/oracle/oradata/cdb2/sysaux01.dbf  SYSAUX           3
/u01/app/oracle/oradata/cdb2/system01.dbf  SYSTEM           1

SQL> EXIT
$
```

Notice that only root data files are listed.

e. Start the `cdb1` database.

```
$ . oraenv
ORACLE_SID = [cdb2] ? cdb1
The Oracle base remains unchanged with value /u01/app/oracle

$ sqlplus / as sysdba

Connected to an idle instance.

SQL> STARTUP
```

```
ORACLE instance started.

Total System Global Area  400846848 bytes
Fixed Size                  2271568 bytes
Variable Size             243271344 bytes
Database Buffers          146800640 bytes
Redo Buffers                8503296 bytes
Database mounted.
Database opened.
SQL> EXIT
$
```

1) Use netca to add the `PDB1_1` net service name for `pdb1_1` pluggable database of `cdb1` in the `tnsnames.ora` file.

```
$ netca
```

2) On the Welcome page, select the "Local Net Service Name configuration" and click Next.
3) On the Net Service Name Configuration page, accept Add and click Next.
4) On the Net Service Name Configuration, Service Name page, enter `pdb1_1` as Service Name and click Next.
5) On the Net Service Name Configuration, Select Protocols page, select TCP and click Next.
6) On the Net Service Name Configuration, TCP/IP Protocol page, enter your complete host name, for example, *`<yourservername>`*, or *`localhost`*, accept "Use the standard port number of 1521," and click Next.
7) On the Net Service Name Configuration, Test page, select "No, do not test" (the pluggable database is not yet opened) and click Next.
8) On the Net Service Name Configuration, Net Service Name page, accept `pdb1_1` as Net Service Name and click Next.
9) On the Net Service Name Configuration, Another Net Service Name page, select No, and Next.
10) On the Net Service Name Configuration Done page, click Next.
11) When you are back on the Welcome page, click Finish.

f. Open the `pdb1_1` pluggable database in `cdb1`.

```
$ sqlplus / as sysdba


Connected to:
Oracle Database 12c Enterprise Edition Release 12.1.0.0.2 -
64bit Production
With the Partitioning, OLAP, Data Mining, Real Application
Testing


SQL> ALTER PLUGGABLE DATABASE pdb1_1 OPEN;
```

```
Pluggable database altered.

SQL> EXIT
$
```

g. Connect to the `pdb1_1` of `cdb1`, and use `DBA_DATA_FILES` view.

```
$ sqlplus system/oracle_4U@pdb1_1
SQL*Plus: Release 12.1.0.0.2 Production on Fri Sep 7 01:28:32
2012


Copyright (c) 1982, 2012, Oracle.  All rights reserved.


Last Successful login time: Wed Aug 22 2012 13:16:11 +00:00


Connected to:
Oracle Database 12c Enterprise Edition Release 12.1.0.0.2 -
64bit Production
With the Partitioning, OLAP, Advanced Analytics and Real
Application Testing options


SQL> col file_name format A65
SQL> SELECT FILE_NAME, TABLESPACE_NAME, FILE_ID
  2  from   dba_data_files;


FILE_NAME
----------------------------------------------------------------
-
TABLESPACE_NAME                   FILE_ID
------------------------------ ----------
/u01/app/oracle/oradata/cdb1/pdb1_1/system01.dbf
SYSTEM                                  8

/u01/app/oracle/oradata/cdb1/pdb1_1/sysaux01.dbf
SYSAUX                                  9

/u01/app/oracle/oradata/cdb1/pdb1_1/SAMPLE_SCHEMA_users01.dbf
USERS                                  10

/u01/app/oracle/oradata/cdb1/pdb1_1/example01.dbf
EXAMPLE                                11


SQL>
```

Notice that only `pdb1_1` data files are listed.

h. Now use V$TABLESPACE and V$DATAFILE view.

```
SQL> col NAME format A12
SQL> SELECT FILE#, ts.name, ts.ts#, ts.con_id
  2  from v$datafile d, v$tablespace  ts
  3  where d.ts#=ts.ts#
  4  and   d.con_id=ts.con_id
  5  order by 4;
```

```
     FILE# NAME                TS#     CON_ID
---------- ------------ ---------- ----------
         4 UNDOTBS1              2          0
        11 EXAMPLE               4          3
        10 USERS                 3          3
         8 SYSTEM                0          3
         9 SYSAUX                1          3

SQL>
```

i. List the temp files of the PDB.

```
SQL> SELECT FILE_NAME, TABLESPACE_NAME from dba_temp_files;

FILE_NAME
--------------------------------------------------------------
TABLESPACE_NAME
------------------------------
/u01/app/oracle/oradata/cdb1/pdb1_1/pdb1_1_temp01.dbf
TEMP

SQL> EXIT
$
```

j. List the password file and SPFILE of both cdb1 and cdb2.

```
$ cd $ORACLE_HOME/dbs
$ ls -l orapw* spfile*
-rw-r----- 1 oracle oinstall 7680 Sep  5 10:43 orapwcdb1
-rw-r----- 1 oracle oinstall 7680 Sep  7 00:48 orapwcdb2
-rw-r----- 1 oracle oinstall 7680 Sep  5 07:23 orapwem12rep
-rw-r----- 1 oracle oinstall 7680 Sep  5 10:02 orapworcl
-rw-r----- 1 oracle oinstall 7680 Sep  5 10:19 orapworcl2
-rw-r----- 1 oracle oinstall 3584 Sep  7 01:23 spfilecdb1.ora
-rw-r----- 1 oracle oinstall 3584 Sep  7 01:08 spfilecdb2.ora
-rw-r----- 1 oracle oinstall 3584 Sep  6 18:13 spfileem12rep.ora
-rw-r----- 1 oracle oinstall 3584 Sep  6 10:00 spfileorcl2.ora
-rw-r----- 1 oracle oinstall 3584 Sep  7 00:35 spfileorcl.ora
```

```
$
```

k. Check ADR files, directories, new DDL statement in alert.log.

```
$ cd $ORACLE_BASE/diag/rdbms/
$ ls
cdb1  cdb2  em12rep  orcl  orcl2
$ cd cdb2/cdb2/trace
$ vi alert_cdb2.log
…
Mon Feb 06 09:27:09 2012
Fri Sep 07 00:41:54 2012
create pluggable database PDB$SEED as clone  using
'/u01/app/oracle/product/12.1.0/dbhome_1/assistants/dbca/templat
es//pdbseed.xml'  source_file_name_convert =
('/ade/b/3895122769/oracle/oradata/seeddata/pdbseed/temp01.dbf',
'/u01/app/oracle/oradata/cdb2/pdbseed/pdbseed_temp01.dbf',
'/ade/b/3895122769/oracle/oradata/seeddata/pdbseed/system01.dbf'
,'/u01/app/oracle/oradata/cdb2/pdbseed/system01.dbf',
'/ade/b/3895122769/oracle/oradata/seeddata/pdbseed/sysaux01.dbf'
,'/u01/app/oracle/oradata/cdb2/pdbseed/sysaux01.dbf') NOCOPY
****************************************************************
Pluggable Database PDB$SEED with pdb id - 2 is created as
UNUSABLE.
If any errors are encountered before the pdb is marked as NEW,
then the pdb must be dropped
****************************************************************
…
Post plug operations are now complete.
Pluggable database PDB$SEED with pdb id - 2 is now marked as
NEW.
****************************************************************
Completed: create pluggable database PDB$SEED as clone  using
'/u01/app/oracle/product/12.1.0/dbhome_1/assistants/dbca/templat
es//pdbseed.xml'  source_file_name_convert =
('/ade/b/3895122769/oracle/oradata/seeddata/pdbseed/temp01.dbf',
'/u01/app/oracle/oradata/cdb2/pdbseed/pdbseed_temp01.dbf',
'/ade/b/3895122769/oracle/oradata/seeddata/pdbseed/system01.dbf'
,'/u01/app/oracle/oradata/cdb2/pdbseed/system01.dbf',
'/ade/b/3895122769/oracle/oradata/seeddata/pdbseed/sysaux01.dbf'
,'/u01/app/oracle/oradata/cdb2/pdbseed/sysaux01.dbf') NOCOPY
alter pluggable database PDB$SEED open restricted
Pluggable database PDB$SEED dictionary check beginning
Pluggable Database PDB$SEED Dictionary check complete
…
$
```

6. List all users created in the new CDB `cdb2`.

   a. Connect to `cdb2` instance.

```
$ . oraenv
ORACLE_SID = [orcl] ? cdb2
The Oracle base for
ORACLE_HOME=/u01/app/oracle/product/12.1.0/dbhome_1 is
/u01/app/oracle
$ sqlplus / as sysdba

Connected to:
Oracle Database 12c Enterprise Edition Release 12.1.0.0.2 -
64bit Production
With the Partitioning, OLAP, Data Mining and Real Application
Testing options

SQL>
```

   b. Verify that the `SYSTEM` user is created.

```
SQL> col username format A30
SQL> select username, common, con_id from cdb_users
  2  where username ='SYSTEM';

USERNAME                       COM  CON_ID
------------------------------ --- ----------
SYSTEM                         YES       1
SYSTEM                         YES       2

SQL>
```

Notice that the user `SYSTEM` exists in all containers as a common user.

   c. List all common users in the CDB.

```
SQL> select distinct username from cdb_users
  2  where common ='YES' order by 1;

USERNAME
------------------------------
ANONYMOUS
APEX_040200
APEX_PUBLIC_USER
APPQOSSYS
AUDSYS
CTXSYS
DBSNMP
DIP
DVF
```

```
DVSYS
FLOWS_FILES
GSMADMIN_INTERNAL
GSMCATUSER
GSMUSER
LBACSYS
MDDATA
MDSYS
OJVMSYS
OLAPSYS
ORACLE_OCM
ORDDATA
ORDPLUGINS
ORDSYS
OUTLN
SI_INFORMTN_SCHEMA
SPATIAL_CSW_ADMIN_USR
SPATIAL_WFS_ADMIN_USR
SYS
SYSBACKUP
SYSDG
SYSKM
SYSTEM
WMSYS
XDB
XS$NULL

35 rows selected.

SQL>
```

d. List all local users in the CDB.

```
SQL> select distinct username, CON_ID from cdb_users
  2  where common ='NO';
```

```
no rows selected

SQL>
```

e. List local users in root.

```
SQL> select distinct username from dba_users
  2  where common ='NO';


no rows selected


SQL>
```

Notice that there is no local user in the root container because it is impossible to create any local user in the root.

7. View distinct accesses by different containers to the single SGA.

```
SQL> select distinct status, con_id from v_$bh order by 2 ;
STATUS         CON_ID
---------- ----------
cr                  1
free                1
xcur                1
cr                  2
xcur                2

SQL> EXIT
$
```

Practice 3-3: Creating a PDB from Seed

Overview

In this practice, you will create a new PDB `pdb2_1` in `cdb2` from seed.

Assumptions

The creation of the CDB `cdb2` is successful.

Tasks

Either use DBCA or SQL Developer or SQL commands.

The creation using SQL is described below.

1. Create a directory for the new data files of `pdb2_1` of `cdb2`.

```
$ . oraenv
ORACLE_SID = [cdb2] ? cdb2
The Oracle base remains unchanged with value /u01/app/oracle

$ cd $ORACLE_BASE/oradata/cdb2
$ mkdir pdb2_1
$
```

2. Run SQL*Plus and connect to the root with a user with `CREATE PLUGGABLE DATABASE` privilege.

```
$ sqlplus / as sysdba

Connected to:
Oracle Database 12c Enterprise Edition Release 12.1.0.0.2 -
64bit Production
With the Partitioning, OLAP, Data Mining and Real Application
Testing options

SQL> CREATE PLUGGABLE DATABASE pdb2_1 ADMIN USER pdb2_1_admin
  2  IDENTIFIED BY oracle_4U ROLES=(CONNECT)
  3  FILE_NAME_CONVERT=('/u01/app/oracle/oradata/cdb2/pdbseed'
  4                  ,'/u01/app/oracle/oradata/cdb2/pdb2_1');

Pluggable database created.

SQL>
```

3. Check the open mode of `pdb2_1`.

```
SQL> col con_id format 999
SQL> col name format A10
SQL> select con_id, NAME, OPEN_MODE,DBID, CON_UID from V$PDBS;

CON_ID NAME                        OPEN_MODE        DBID    CON_UID
```

```
------ -------------------- ---------- ---------- ----------
     2 PDB$SEED             READ ONLY  4029890286 4029890286
     3 PDB2_1               MOUNTED    3071827262 3071827262

SQL>
```

4. Open `pdb2_1`.

   a. Open the PDB.

```
SQL> alter pluggable database pdb2_1 open;

Pluggable database altered.
SQL> EXIT
$
```

   b. Connect to `pdb2_1 AS SYSDBA`.

      1) Use netca to add the `PDB2_1` net service name for `pdb2_1` pluggable database of `cdb2` in the `tnsnames.ora` file.

```
$ netca
```

      2) On the Welcome page, select the “Local Net Service Name configuration” and click Next.

      3) On the Net Service Name Configuration page, accept `Add` and click Next.

      4) On the Net Service Name Configuration, Service Name page, enter `pdb2_1` as Service Name and click Next.

      5) On the Net Service Name Configuration, Select Protocols page, select TCP and click Next.

      6) On the Net Service Name Configuration, TCP/IP Protocol page, enter your complete host name, for example, *`<yourservername>`*, or *`localhost`,* accept “Use the standard port number of 1521,” and click Next.

      7) On the Net Service Name Configuration, Test page, select “No, do not test” (the pluggable database is not yet opened) and click Next.

      8) On the Net Service Name Configuration, Net Service Name page, accept `pdb2_1` as Net Service Name and click Next.

      9) On the Net Service Name Configuration, Another Net Service Name page, select No, and Next.

      10) On the Net Service Name Configuration Done page, click Next.

      11) When you are back on the Welcome page, click Finish.

```
$ sqlplus sys/oracle_4U@pdb2_1 AS SYSDBA

Connected to:
Oracle Database 12c Enterprise Edition Release 12.1.0.0.2 -
64bit Production
With the Partitioning, OLAP, Advanced Analytics, Real
Application Testing

SQL>
```

5. The service is now available and registered with the listener.

```
SQL> !lsnrctl status

The command completed successfully
LSNRCTL for Linux: Version 12.1.0.0.2 - Production on 07-SEP-
2012 01:47:28

Copyright (c) 1991, 2012, Oracle.  All rights reserved.

Connecting to
(DESCRIPTION=(ADDRESS=(PROTOCOL=IPC)(KEY=EXTPROC1521)))
STATUS of the LISTENER
------------------------
Alias                     LISTENER
Version                   TNSLSNR for Linux: Version 12.1.0.0.2
- Production
Start Date                14-SEP-2012 03:04:56
Uptime                    16 days 22 hr. 0 min. 0 sec
Trace Level               off
Security                  ON: Local OS Authentication
SNMP                      OFF
Listener Parameter File
/u01/app/oracle/product/12.1.0/dbhome_1/network/admin/listener.o
ra
Listener Log File
/u01/app/oracle/diag/tnslsnr/yourserver/listener/alert/log.xml
Listening Endpoints Summary...
  (DESCRIPTION=(ADDRESS=(PROTOCOL=ipc)(KEY=EXTPROC1521)))

(DESCRIPTION=(ADDRESS=(PROTOCOL=tcp)(HOST=yourserver)(PORT=1521)
))
(DESCRIPTION=(ADDRESS=(PROTOCOL=tcp)(HOST=yourserver)(PORT=5501)
)(Presentation=HTTP)(Session=RAW))
(DESCRIPTION=(ADDRESS=(PROTOCOL=tcp)(HOST=yourserver)(PORT=5502)
)(Presentation=HTTP)(Session=RAW))
(DESCRIPTION=(ADDRESS=(PROTOCOL=tcp)(HOST=yourserver)(PORT=5500)
)(Presentation=HTTP)(Session=RAW))
Services Summary...
Service "cdb1" has 1 instance(s).
  Instance "cdb1", status READY, has 1 handler(s) for this
service...
Service "cdb1XDB" has 1 instance(s).
  Instance "cdb1", status READY, has 1 handler(s) for this
service...
Service "cdb2" has 1 instance(s).
```

```
  Instance "cdb2", status READY, has 1 handler(s) for this
service...
Service "cdb2XDB" has 1 instance(s).
  Instance "cdb2", status READY, has 1 handler(s) for this
service...
Service "em12rep" has 1 instance(s).
  Instance "em12rep", status READY, has 1 handler(s) for this
service...
Service "em12repXDB" has 1 instance(s).
  Instance "em12rep", status READY, has 1 handler(s) for this
service...
Service "pdb1_1" has 1 instance(s).
  Instance "cdb1", status READY, has 1 handler(s) for this
service.
Service "pdb2_1" has 1 instance(s).
  Instance "cdb2", status READY, has 1 handler(s) for this
service.
The command completed successfully

SQL>
```

6. Connect to `pdb2_1` as `sys` user by using EasyConnect and then as `pdb2_1_admin` user.

```
SQL> CONNECT sys/oracle_4U@localhost:1521/pdb2_1 AS SYSDBA
Connected.
SQL> connect pdb2_1_admin/oracle_4U@PDB2_1
Connected.

SQL> show con_name

CON_NAME
------------------------------
PDB2_1
SQL>
```

7. List the data files created.

```
SQL> !ls -l $ORACLE_BASE/oradata/cdb2/pdb2_1/*
-rw-r----- 1 oracle oinstall  20979712 Sep  7 01:47
/u01/app/oracle/oradata/cdb2/pdb2_1/pdbseed_temp01.dbf
-rw-r----- 1 oracle oinstall 671096832 Sep  7 01:47
/u01/app/oracle/oradata/cdb2/pdb2_1/sysaux01.dbf
-rw-r----- 1 oracle oinstall 262152192 Sep  7 01:47
/u01/app/oracle/oradata/cdb2/pdb2_1/system01.dbf
SQL>
```

8. Check the services, data files, and tablespaces using views.

```
SQL> connect system/oracle_4U@pdb2_1
Connected.
SQL> col name format A30
SQL> select name from v$services;

NAME
----------------------------
pdb2_1

SQL> col file_name format A50
SQL> col tablespace_name format A8
SQL> col file_id format 99
SQL> col con_id format 9
SQL> select FILE_NAME, TABLESPACE_NAME, FILE_ID, con_id
  2  from   cdb_data_files
  3  order  by con_id ;

FILE_NAME                                          TABLESPA
FILE_ID CON_ID
-------------------------------------------------- -------- ----
--- ------
/u01/app/oracle/oradata/cdb2/pdb2_1/sysaux01.dbf   SYSAUX
9      3
/u01/app/oracle/oradata/cdb2/pdb2_1/system01.dbf   SYSTEM
8      3

SQL> select FILE_NAME, TABLESPACE_NAME, FILE_ID
  2  from   dba_data_files;

FILE_NAME                                          TABLESPA
FILE_ID
-------------------------------------------------- -------- ----
---
/u01/app/oracle/oradata/cdb2/pdb2_1/system01.dbf   SYSTEM
8
/u01/app/oracle/oradata/cdb2/pdb2_1/sysaux01.dbf   SYSAUX
9

SQL> col file_name format A60
SQL> select FILE_NAME, TABLESPACE_NAME, FILE_ID
  2  from cdb_temp_files;

FILE_NAME                                                    TABLESPA
FILE_ID
```

```
-------------------------------------------------------- --------
------
/u01/app/oracle/oradata/cdb2/pdb2_1/pdbseed_temp01.dbf TEMP
3

SQL> select FILE_NAME, TABLESPACE_NAME, FILE_ID
  2  from dba_temp_files;

FILE_NAME                                                TABLESPA
FILE_ID
--------------------------------------------------------- --------
-------
/u01/app/oracle/oradata/cdb2/pdb2_1/ pdbseed_temp01.dbf TEMP
3

SQL>
```

9. To be able to view all objects of all containers in the CDB, connect to the root and use CDB_*xxx* views.

```
SQL> connect / as sysdba
Connected.
SQL> show con_id

CON_ID
------------------------------
1
SQL> show con_name

CON_NAME
------------------------------
CDB$ROOT
SQL> select name from v$services;

NAME
------------------------------
pdb2_1
cdb2XDB
cdb2
SYS$BACKGROUND
SYS$USERS

SQL> select FILE_NAME, TABLESPACE_NAME, FILE_ID, con_id
  2  from   cdb_data_files
  3  order  by con_id, file_id ;
```

```
FILE_NAME                                           TABLESPA
FILE_ID CON_ID
--------------------------------------------------- -------- ----
--- ------
FILE_NAME                                           TABLESPA
FILE_ID CON_ID
--------------------------------------------------- -------- ----
--- ------
/u01/app/oracle/oradata/cdb2/system01.dbf           SYSTEM
1      1
/u01/app/oracle/oradata/cdb2/sysaux01.dbf           SYSAUX
3      1
/u01/app/oracle/oradata/cdb2/undotbs01.dbf          UNDOTBS1
4      1
/u01/app/oracle/oradata/cdb2/users01.dbf            USERS
6      1
/u01/app/oracle/oradata/cdb2/pdbseed/system01.dbf   SYSTEM
5      2
/u01/app/oracle/oradata/cdb2/pdbseed/sysaux01.dbf   SYSAUX
7      2
/u01/app/oracle/oradata/cdb2/pdb2_1/system01.dbf    SYSTEM
8      3
/u01/app/oracle/oradata/cdb2/pdb2_1/sysaux01.dbf    SYSAUX
9      3


8 rows selected.

SQL> select FILE_NAME, TABLESPACE_NAME, FILE_ID
  2  from   dba_data_files;

FILE_NAME                                           TABLESPA
FILE_ID
--------------------------------------------------- -------- ----
-
/u01/app/oracle/oradata/cdb2/users01.dbf            USERS
6
/u01/app/oracle/oradata/cdb2/undotbs01.dbf          UNDOTBS1
4
/u01/app/oracle/oradata/cdb2/sysaux01.dbf           SYSAUX
3
/u01/app/oracle/oradata/cdb2/system01.dbf           SYSTEM
1


SQL> select FILE_NAME, TABLESPACE_NAME, FILE_ID
  2  from cdb_temp_files;
```

```

FILE_NAME                                                TABLESPA
FILE_ID
-------------------------------------------------------- --------
-------
/u01/app/oracle/oradata/cdb2/temp01.dbf                  TEMP
1
/u01/app/oracle/oradata/cdb2/pdbseed/pdbseed_temp01.dbf TEMP
2
/u01/app/oracle/oradata/cdb2/pdb2_1/pdbseed_temp01.dbf  TEMP
3

SQL> select FILE_NAME, TABLESPACE_NAME, FILE_ID
  2  from dba_temp_files;

FILE_NAME                                                TABLESPA
FILE_ID
-------------------------------------------------------- --------
-------
/u01/app/oracle/oradata/cdb2/temp01.dbf                  TEMP
1
SQL> EXIT
$
```

Practice 3-4: Cloning PDB Within the Same CDB

Overview

In this practice, you will create a new PDB with the cloning method, cloning `pdb2_2` from `pdb2_1` within the same CDB `cdb2`.

Assumptions

The `pdb2_1` creation has completed successfully in Practice 3-3.

Tasks

Either use the SQL commands OR SQL Developer.

Method with SQL*Plus.

1. Create a directory for the new data files of `pdb2_2` of `cdb2`.

```
$ . oraenv
ORACLE_SID = [cdb2] ? cdb2
The Oracle base remains unchanged with value /u01/app/oracle

$ cd $ORACLE_BASE/oradata/cdb2
$ mkdir pdb2_2
$
```

2. Run SQL*Plus and connect to the root as a user granted with `CREATE PLUGGABLE DATABASE` privilege.

   a. Set `pdb2_1` in `READ ONLY` open mode before cloning.

```
$ sqlplus / as sysdba

Connected to:
Oracle Database 12c Enterprise Edition Release 12.1.0.0.2 -
64bit Production
With the Partitioning, OLAP, Advanced Analytics and Real
Application Testing options

SQL> alter pluggable database pdb2_1 close;

Pluggable database altered.

SQL> alter pluggable database pdb2_1 open read only;

Pluggable database altered.

SQL>
```

b. Change OMF DB_CREATE_FILE_DEST parameter value to '/u01/app/oracle/oradata/cdb2/pdb2_2'.

```
SQL> alter system set db_create_file_dest =
'/u01/app/oracle/oradata/cdb2/pdb2_2';


System altered.

SQL>
```

c. Clone pdb2_2 from pdb2_1.

```
SQL> CREATE PLUGGABLE DATABASE pdb2_2 FROM pdb2_1;


Pluggable database created.


SQL>
```

3. Check the open mode of pdb2_2.

```
SQL> select name, open_mode from v$pdbs;


NAME                           OPEN_MODE
------------------------------ ----------
PDB$SEED                       READ ONLY
PDB2_1                         READ ONLY
PDB2_2                         MOUNTED

SQL>
```

4. Set PDB2_1 in READ WRITE open mode and open PDB2_2.

a. Open PDB2_1 in READ WRITE mode.

```
SQL> alter pluggable database PDB2_1 close;


Pluggable database altered.

SQL> alter pluggable database PDB2_1 open;


Pluggable database altered.

SQL>
```

b. Open PDB2_2 in READ WRITE mode.

```
SQL> alter pluggable database PDB2_2 open;


Pluggable database altered.


SQL> EXIT
$
```

c. Connect to `pdb2_2 AS SYSDBA`.

1) Use netca to add the `PDB2_2` net service name for `pdb2_2` pluggable database of `cdb2` in the `tnsnames.ora` file.

```
$ netca
```

2) On the Welcome page, select the “Local Net Service Name configuration” and click Next.

3) On the Net Service Name Configuration page, accept `Add` and click Next.

4) On the Net Service Name Configuration, Service Name page, enter `pdb2_2` as Service Name and click Next.

5) On the Net Service Name Configuration, Select Protocols page, select TCP and click Next.

6) On the Net Service Name Configuration, TCP/IP Protocol page, enter your complete host name, for example, *`<yourservername>`*, or *`localhost,`* accept “Use the standard port number of 1521,” and click Next.

7) On the Net Service Name Configuration, Test page, select “No, do not test” (the pluggable database is not yet opened) and click Next.

8) On the Net Service Name Configuration, Net Service Name page, accept `pdb2_2` as Net Service Name and click Next.

9) On the Net Service Name Configuration, Another Net Service Name page, select No, and Next.

10) On the Net Service Name Configuration Done page, click Next.

11) When you are back on the Welcome page, click Finish.

```
$ sqlplus sys/oracle_4U@pdb2_2 AS SYSDBA

Connected to:
Oracle Database 12c Enterprise Edition Release 12.1.0.0.2 -
64bit Production
With the Partitioning, OLAP, Advanced Analytics and Real
Application Testing options

SQL>
```

d. Check the open mode of the PDBs.

```
SQL> CONNECT / AS SYSDBA
Connected.

SQL> select name, open_mode from v$pdbs;

NAME                           OPEN_MODE
------------------------------ ----------
PDB$SEED                       READ ONLY
PDB2_1                         READ WRITE
PDB2_2                         READ WRITE
```

```
SQL>
```

5. Connect to `PDB2_2` as the `SYSTEM` user.

```
SQL> connect system/oracle_4U@PDB2_2
Connected.

SQL> show con_name
PDB2_2
SQL> EXIT
$
```

6. List the data files created.

```
$ cd $ORACLE_BASE/oradata/cdb2/pdb2_2
$ ls -l
total 4
drwxr-x--- 3 oracle oinstall 4096 Feb  6 13:46 CDB2
$ cd CDB2
$ ls -l
total 4
drwxr-x--- 3 oracle oinstall 4096 Jun 29 21:26
C3A419D23D3F1DE5E043160200C04142
$ cd C3A419D23D3F1DE5E043160200C04142
$ ls –l
total 4
drwxr-x--- 2 oracle oinstall 4096 Jun 29 21:26 datafile
$ cd datafile
$ ls -l
total 809836
-rw-r----- 1 oracle oinstall 566239232 Jun 29 21:27
o1_mf_sysaux_7yw7d38o_.dbf
-rw-r----- 1 oracle oinstall 262152192 Jun 29 21:27
o1_mf_system_7yw7d38h_.dbf
-rw-r----- 1 oracle oinstall  20979712 Jun 29 21:26
o1_mf_temp_7yw7dg92_.dbf
$
```

Method with SQL Developer:

1. If you already created pdb2_2 with SQL*Plus and would like to test the creation with SQL Developer, you first have to drop `pdb2_2` to recreate it.
   a. Drop the pluggable database `pdb2_2`.

```
$ sqlplus / AS SYSDBA


Connected to:
```

```
Oracle Database 12c Enterprise Edition Release 12.1.0.0.2 -
64bit Production
With the Partitioning, OLAP, Advanced Analytics and Real
Application Testing options

SQL> ALTER PLUGGABLE DATABASE pdb2_2 CLOSE IMMEDIATE;

Pluggable database altered.

SQL> DROP PLUGGABLE DATABASE pdb2_2 INCLUDING DATAFILES;

Pluggable database dropped.

SQL> EXIT
$
```

b. Remove the directory.

```
$ rm -r $ORACLE_BASE/oradata/cdb2/pdb2_2
$
```

2. Create a directory for the new data files of `pdb2_2` of `cdb2`.

```
$ . oraenv
ORACLE_SID = [cdb2] ? cdb2
The Oracle base remains unchanged with value /u01/app/oracle

$ cd $ORACLE_BASE/oradata/cdb2
$ mkdir pdb2_2
$
```

3. Run SQL*Plus and connect to the root to set OMF directory to the `pdb2_2` directory.

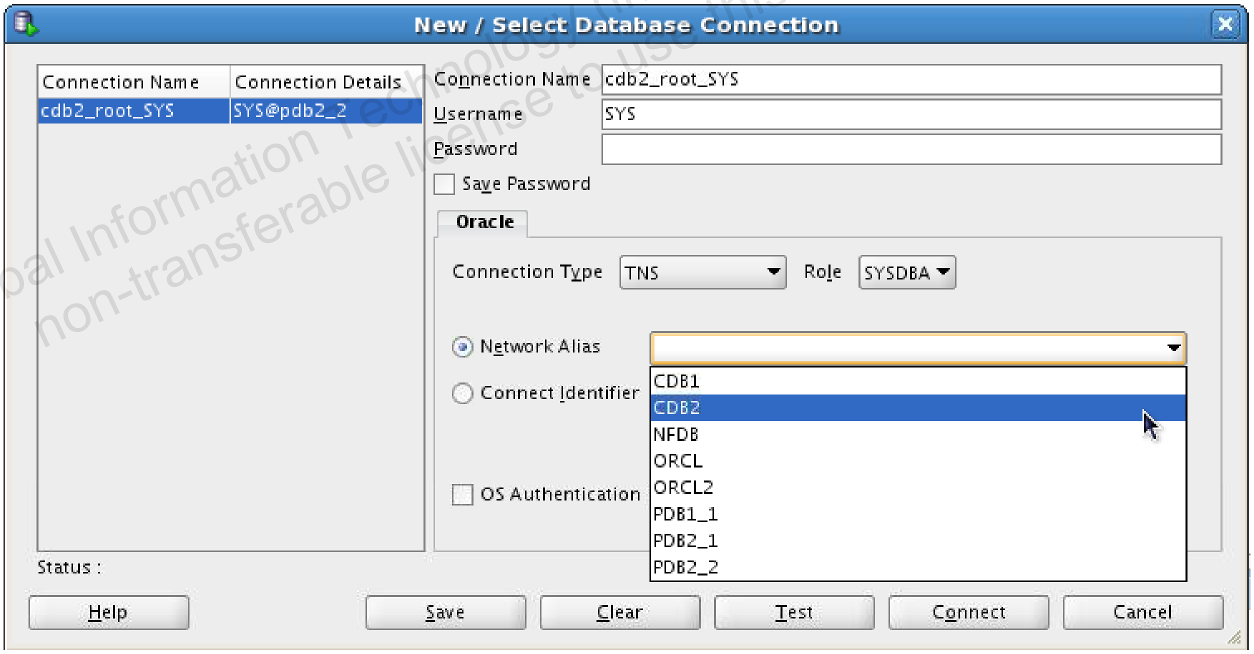
```
$ sqlplus / AS SYSDBA

Connected to:
Oracle Database 12c Enterprise Edition Release 12.1.0.0.2 -
64bit Production
With the Partitioning, OLAP, Advanced Analytics and Real
Application Testing options

SQL> alter system set db_create_file_dest =
'/u01/app/oracle/oradata/cdb2/pdb2_2' scope=both;

System altered.

SQL> EXIT
$
```

4. Launch SQL Developer.

```
$ cd $ORACLE_HOME/sqldeveloper
$ ./sqldeveloper.sh
$
```

5. Create a connection as `SYS` in root `cdb2`.
6. Open a connection as `SYS` in `cdb2`.
   a. Choose the View option.
   b. Click Connections.
   c. Click + in the left Connections pane to add a new connection.
   d. Fill the different fields as follows: be sure to change the host name and port number to your assigned host name and port number.

Window/Page Description	Choices or Values
Connection Name	`cdb2_root_SYS`
Username	`sys`
Password	`oracle_4U`
Connection Type	`TNS`
Role	`SYSDBA`
Network Alias	`cdb2`



New / Select Database Connection

Connection Name	Connection Details
cdb2_root_SYS	SYS@pdb2_2

Connection Name: cdb2_root_SYS
Username: SYS
Password:
□ Save Password

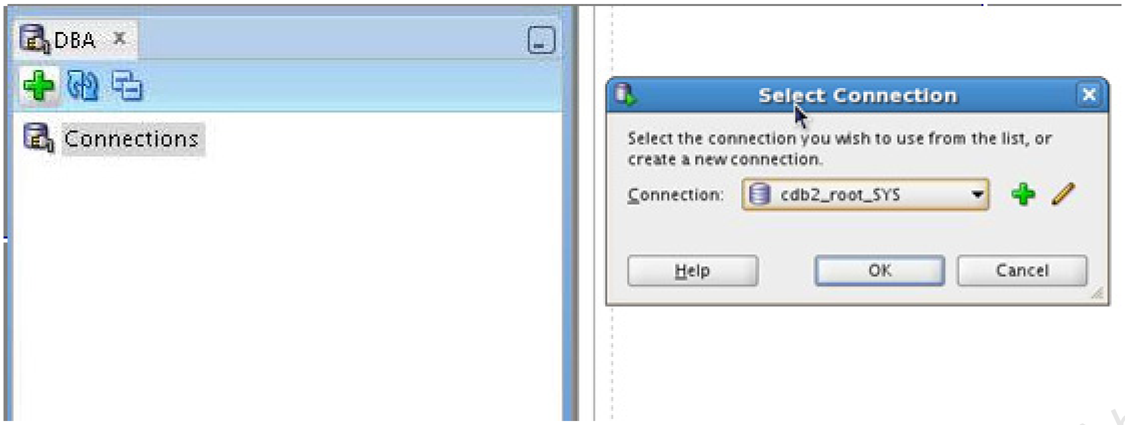
Oracle

Connection Type: TNS
Role: SYSDBA

● Network Alias
○ Connect Identifier
□ OS Authentication

   e. Click Test.
   f. If the status is Success, click Save.
   g. Click Connect.

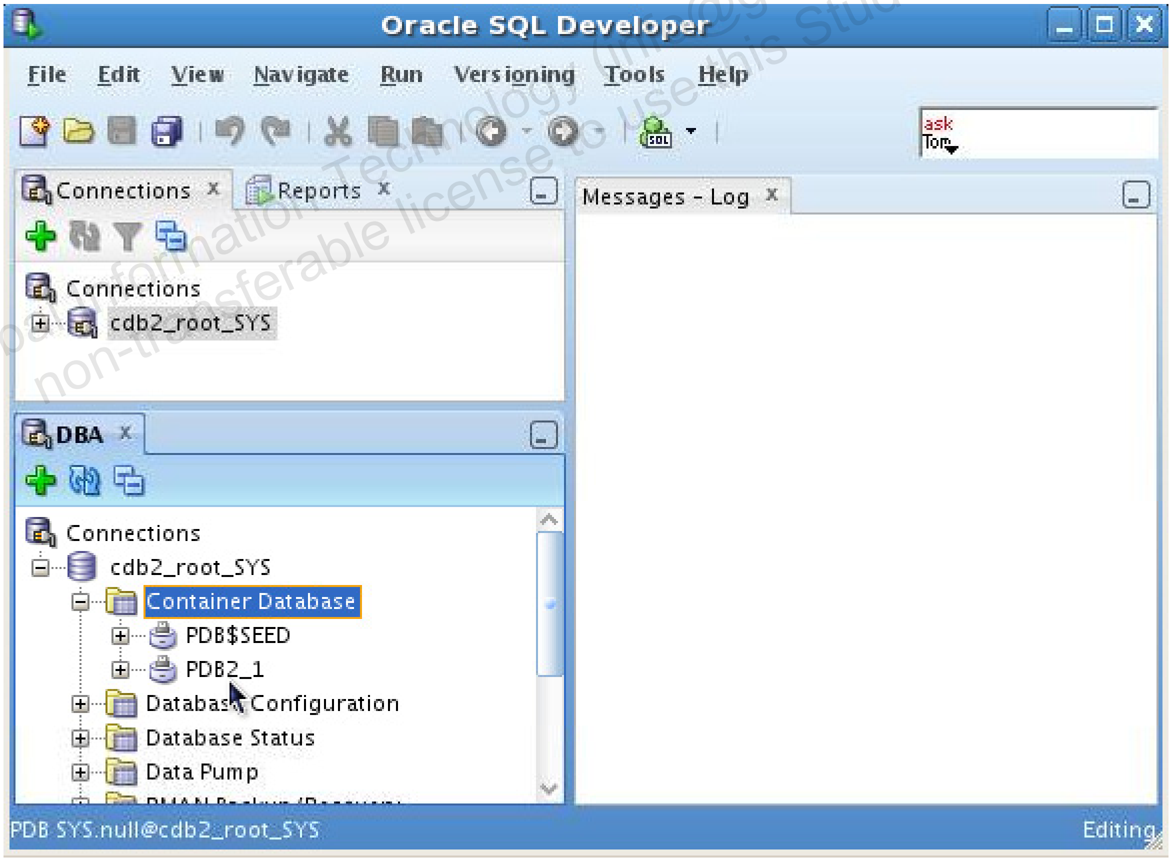
7. To manage the CDB and its PDBs:
   a. Choose the View option.
   b. Click DBA.
   c. Click + in the left Connections pane to view an existing connection.
   d. From the list of existing connections, choose the one you just created.



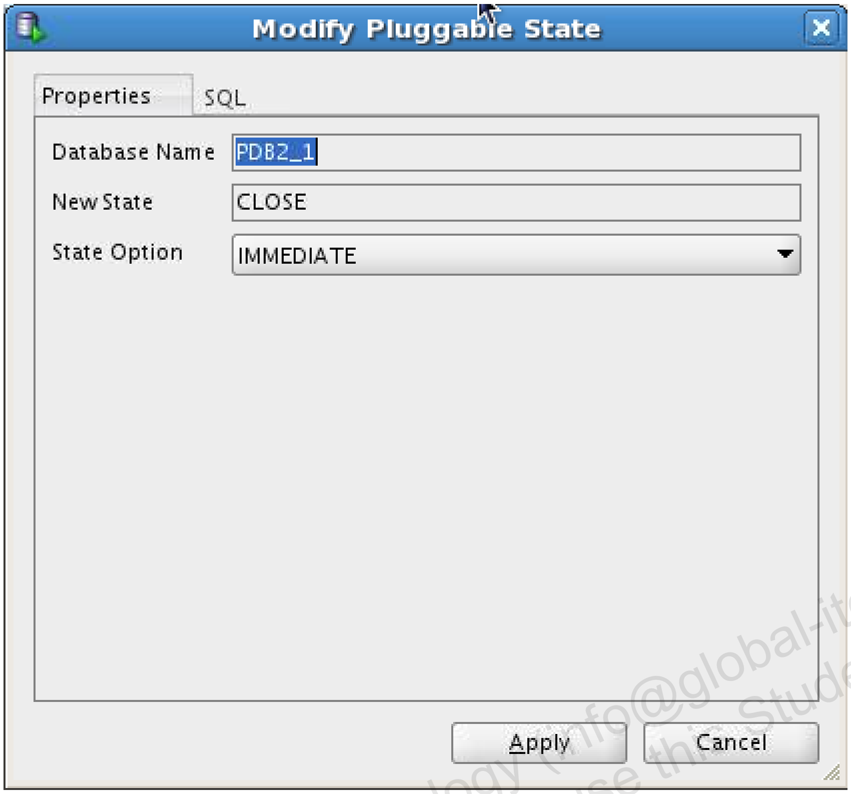
Select the connection you wish to use from the list, or create a new connection.

Connection: cdb2_root_SYS

   e. Click OK.
   f. Click the sign + in front of the name of the `cdb2_root_SYS` connection to expand the folder. Then click the sign + in front of "Container Database". The list of containers in the CDB appears.



8. Right-click the pluggable database `pdb2_1` to show possible actions. Choose `Modify State` to set it in `READ ONLY` open mode before cloning.
   a. First close.



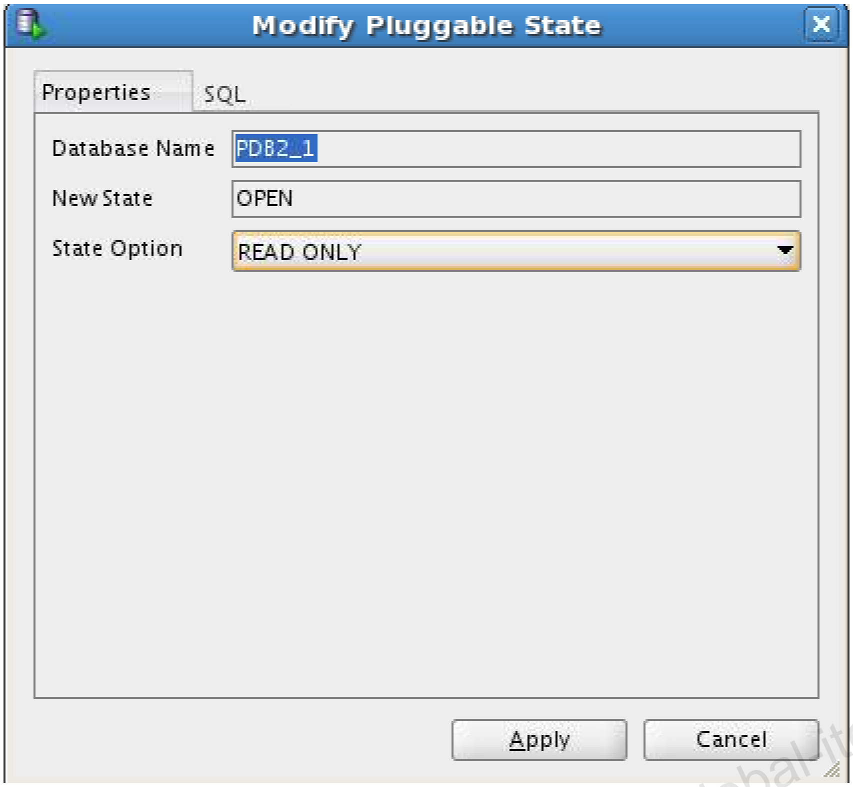
Modify Pluggable State

Database Name: PDB2_1

New State: CLOSE

State Option: IMMEDIATE

   b. Click Apply.
   c. Choose Modify State again.
   d. Set the State Option to `READ ONLY`.



Database Name: PDB2_1

New State: OPEN

State Option: READ ONLY

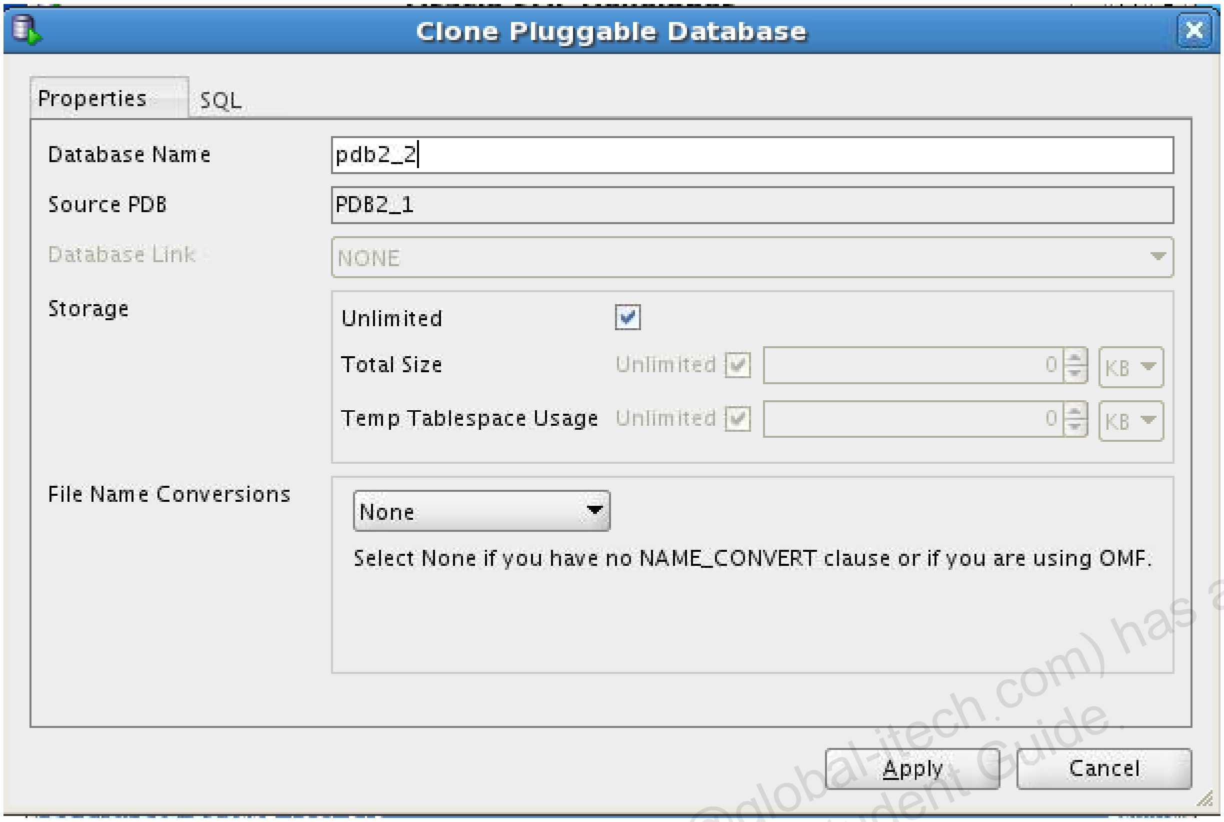
e. Click Apply then OK.

9. Right-click the pluggable database `pdb2_1` and choose Clone Pluggable Database….

a. Fill the different fields as follows.

Window/Page Description	Choices or Values
Database Name	`pdb2_2`
Source PDB	`pdb2_1`
File Name Conversions	`None`

File Name Conversions kept to `None` means that it uses the OMF target destination declared in `DB_CREATE_FILE_DEST` parameter.



Clone Pluggable Database

Properties | SQL

Database Name: pdb2_2

Source PDB: PDB2_1

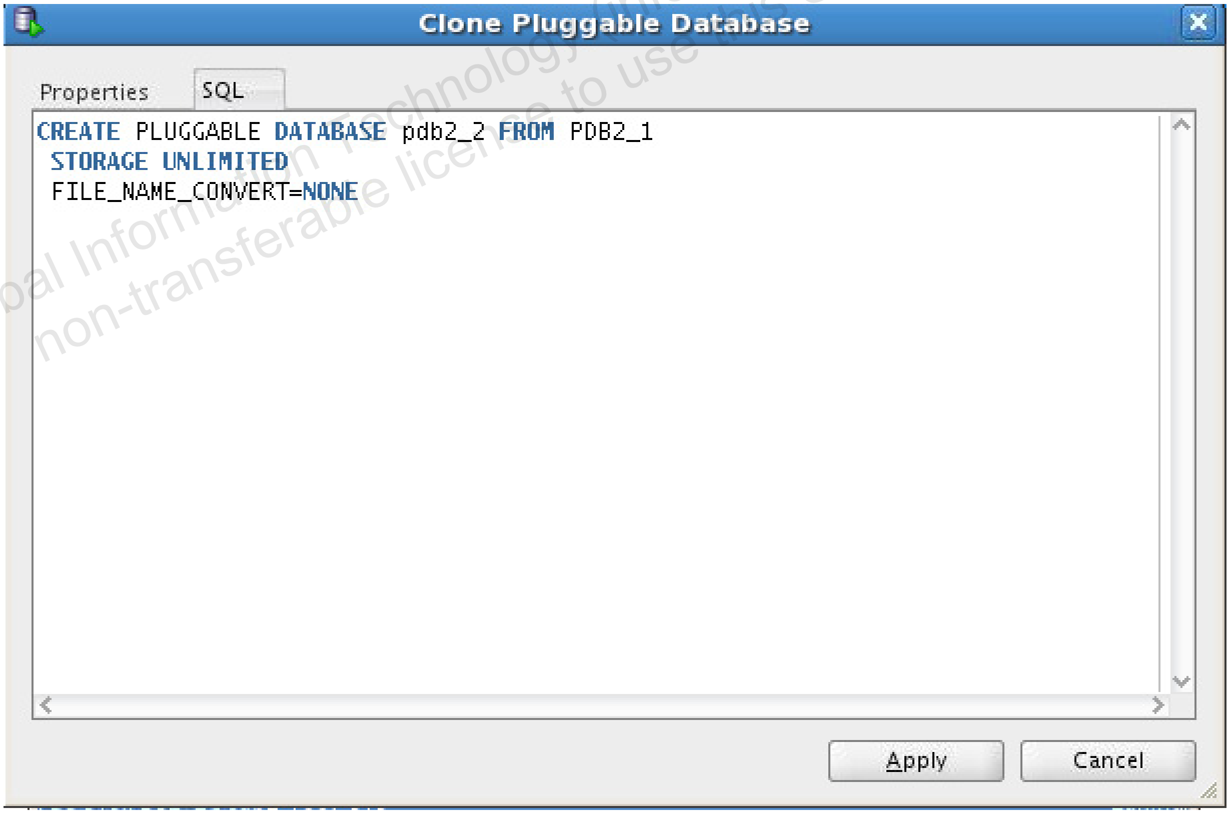
Database Link: NONE

Storage
- Unlimited ☑
- Total Size: Unlimited ☑ 0 KB
- Temp Tablespace Usage: Unlimited ☑ 0 KB

File Name Conversions: None

Select None if you have no NAME_CONVERT clause or if you are using OMF.

b. You can view the SQL statement before applying.



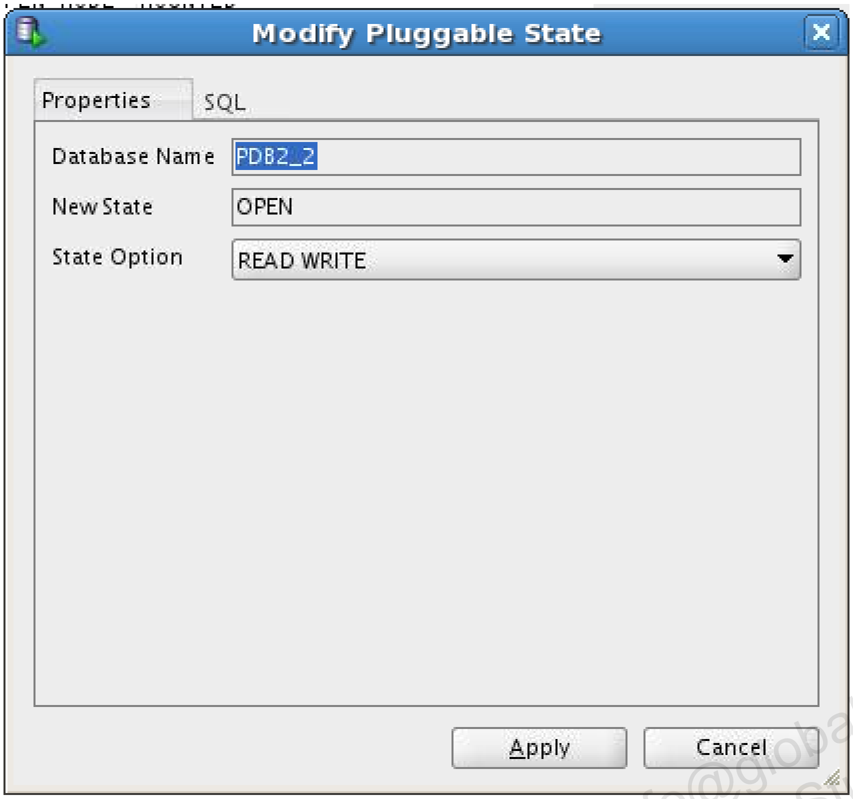
Clone Pluggable Database

```
CREATE PLUGGABLE DATABASE pdb2_2 FROM PDB2_1
 STORAGE UNLIMITED
 FILE_NAME_CONVERT=NONE
```

c. Click Apply then OK. The new `pdb2_2` appears in the list of PDBs in `cdb2`.

d. Open `pdb2_2`.

1) Open `pdb2_2` in `READ WRITE` mode. Right click `pdb2_2` and click Modify State.



Database Name: PDB2_2
New State: OPEN
State Option: READ WRITE

2) Click Apply, then OK.

e. Open `pdb2_1` in `READ WRITE` mode. Right click `pdb2_1` and click Modify State. Click Apply , then OK.

10. Leave SQL Developer.
    a. Click File.
    b. Then click exit.

Practice 3-5: Plugging a Non-CDB into a CDB

Overview

In this practice, you will plug the non-CDB `orcl2` into the CDB `cdb2`. You will not use Export/Import DataPump, which can be a possible method, but the method with `DBMS_PDB` package. This package executed in the non-CDB `orcl2` generates an XML file describing the tablespaces and data files of non-CDB `orcl2`. The XML file is then used when creating `pdb_orcl2` in `cdb2`.

Tasks

1. Use `DBMS_PDB.DESCRIBE` to “unplug” non-CDB `orcl2`.

```
$ . oraenv
ORACLE_SID = [cdb2] ? orcl2
The Oracle base remains unchanged with value /u01/app/oracle
$ sqlplus / as sysdba

Connected to an idle instance.
SQL> startup mount
ORACLE instance started.

Total System Global Area 1670221824 bytes
Fixed Size                  2274000 bytes
Variable Size             973081904 bytes
Database Buffers          687865856 bytes
Redo Buffers                7000064 bytes
Database mounted.
SQL>
SQL> alter database open read only;

Database altered.
SQL> exec dbms_pdb.describe
('/u01/app/oracle/oradata/orcl2/xmlorcl2.xml')

PL/SQL procedure successfully completed.

SQL> shutdown immediate
Database closed.
Database dismounted.
ORACLE instance shut down.
SQL> EXIT
$
```

2. Create a new PDB `pdb_orcl2` to plug non-CDB `orcl2` into `cdb2` using the XML file generated.
   You will have to remove the temp file because the creation cannot complete until it is removed to create it.

```
$ . oraenv
ORACLE_SID = [orcl2] ? cdb2
The Oracle base for
ORACLE_HOME=/u01/app/oracle/product/12.1.0/dbhome_1 is
/u01/app/oracle
$ sqlplus / as sysdba


Connected to:
Oracle Database 12c Enterprise Edition Release 12.1.0.0.2 -
64bit Production
With the Partitioning, OLAP, Data Mining and Real Application
Testing options


SQL> create pluggable database PDB_ORCL2  using
'/u01/app/oracle/oradata/orcl2/xmlorcl2.xml' NOCOPY;
create pluggable database PDB_ORCL2  using 'xmlorcl2' NOCOPY
*
ERROR at line 1:
ORA-01119: error in creating database file
'/u01/app/oracle/oradata/orcl2/temp01.dbf'
ORA-27038: created file already exists
Additional information: 1


SQL>!rm /u01/app/oracle/oradata/orcl2/temp01.dbf


SQL>
SQL> create pluggable database PDB_ORCL2 using
'/u01/app/oracle/oradata/orcl2/xmlorcl2.xml' NOCOPY;


Pluggable database created.


SQL> EXIT
$
```

3. To complete the operation, you have to convert the plugged non-CDB to a proper PDB by deleting unnecessary metadata from PDB SYSTEM tablespace.
   For this purpose, you execute the $ORACLE_HOME/rdbms/admin/noncdb_to_pdb.sql script whilst connected to the PDB.

   a. Connect to pdb2_2 as SYSDBA.

      1) Use netca to add the PDB_ORCL2 net service name for pdb_orcl2 pluggable database of cdb2 in the tnsnames.ora file.

```
$ netca
```

      2) On the Welcome page, select the “Local Net Service Name configuration” and click Next.
      3) On the Net Service Name Configuration page, accept Add and click Next.
      4) On the Net Service Name Configuration, Service Name page, enter pdb_orcl2 as Service Name and click Next.
      5) On the Net Service Name Configuration, Select Protocols page, select TCP and click Next.
      6) On the Net Service Name Configuration, TCP/IP Protocol page, enter your complete host name, for example, *<yourservername>*, or *localhost,* accept “Use the standard port number of 1521,” and click Next.
      7) On the Net Service Name Configuration, Test page, select “No, do not test” (the pluggable database is not yet opened) and click Next.
      8) On the Net Service Name Configuration, Net Service Name page, accept pdb_orcl2 as Net Service Name and click Next.
      9) On the Net Service Name Configuration, Another Net Service Name page, select No, and Next.
      10) On the Net Service Name Configuration Done page, click Next.
      11) When you are back on the Welcome page, click Finish.

   b. Now connect to pdb_orcl2 using the net service name.

```
$ sqlplus sys/oracle_4U@pdb_orcl2 as sysdba

Connected to:
Oracle Database 12c Enterprise Edition Release 12.1.0.0.2 -
64bit Production
With the Partitioning, OLAP, Advanced Analytics and Real
Application Testing options
SQL>
```

   c. Execute the script. Expect around 35 minutes to complete.

```
SQL> @$ORACLE_HOME/rdbms/admin/noncdb_to_pdb.sql
SQL> SET SERVEROUTPUT ON
SQL> SET FEEDBACK 1
SQL> SET NUMWIDTH 10
SQL> SET LINESIZE 80
SQL> SET TRIMSPOOL ON
SQL> SET TAB OFF
SQL> SET PAGESIZE 100
SQL>
SQL> WHENEVER SQLERROR EXIT;
SQL>
SQL> DOC
DOC>############################################################
###########
DOC>############################################################
###########
DOC>   The following statement will cause an "ORA-01722: invalid
number"
DOC>   error if we're not in a PDB.
DOC>############################################################
###########
DOC>############################################################
###########
DOC>#
SQL>
SQL> VARIABLE pdbname VARCHAR2(128)
SQL> BEGIN
  2    SELECT sys_context('USERENV', 'CON_NAME')
  3      INTO :pdbname
  4      FROM dual
  5      WHERE sys_context('USERENV', 'CON_NAME') <> 'CDB$ROOT';
  6  END;
  7  /

PL/SQL procedure successfully completed.

…

SQL>
SQL> Rem
================================================================
=
SQL> Rem Run component validation procedure
SQL> Rem
================================================================
=
SQL>
SQL> EXECUTE dbms_registry_sys.validate_components;
...Database user "SYS", database schema "APEX_040200", user#
"98" 10:21:02
```

```
...Compiled 0 out of 2998 objects considered, 0 failed
compilation 10:21:03
...263 packages
...255 package bodies
...453 tables
...11 functions
...16 procedures
...3 sequences
...458 triggers
...1322 indexes
...207 views
...0 libraries
...6 types
...0 type bodies
...0 operators
...0 index types
...Begin key object existence check 10:21:03
...Completed key object existence check 10:21:03
...Setting DBMS Registry 10:21:03
...Setting DBMS Registry Complete 10:21:03
...Exiting validate 10:21:03

PL/SQL procedure successfully completed.

SQL> SET serveroutput off
SQL>
SQL> Rem
================================================================
=
SQL> Rem END utlrp.sql
SQL> Rem
================================================================
=
SQL>
…

SQL> alter pluggable database "&pdbname" close;

Pluggable database altered.

SQL>
SQL> -- leave the PDB in the same state it was when we started
SQL> BEGIN
```

```
  2    execute immediate '&open_sql &restricted_state';
  3  EXCEPTION
  4    WHEN OTHERS THEN
  5    BEGIN
  6      IF (sqlcode <> -900) THEN
  7        RAISE;
  8      END IF;
  9    END;
 10  END;
 11  /

PL/SQL procedure successfully completed.

SQL>
SQL> alter session set container="&pdbname";

Session altered.

SQL>
SQL> WHENEVER SQLERROR CONTINUE;
SQL>
SQL>
```

d. Quit the session after opening the new PDB.

```
SQL> alter pluggable database pdb_orcl2 open;

Pluggable database altered.

SQL> EXIT
$
```

4. Connect to PDB_ORCL2.

```
$ sqlplus sys/oracle_4U@localhost:1521/PDB_ORCL2 as SYSDBA


Connected to:
Oracle Database 12c Enterprise Edition Release 12.1.0.0.2 -
64bit Production
With the Partitioning, OLAP, Advanced Analytics and Real
Application Testing options


SQL>
```

5. Verify that the application data is in the PDB `pdb_orcl2`:

```
SQL> select count(empno) from scott.emp;

COUNT(EMPNO)
------------
          14
SQL> EXIT
$
```

Practice 3-6: Merging All PDBs of CDBs into a Single CDB

Overview

In this practice you merge all PDBs of `cdb1` into a single CDB, `cdb2`.

1. Merge all PDBs of `cdb1` into `cdb2`.
2. Drop `cdb1`.

Assumptions

The CDB `cdb2` exists. The `cdb2` creation has completed successfully in Practice 3-1.

Tasks

1. Connect to the multitenant container database `cdb1` to unplug all PDBs.
   a. Connect to `cdb1` root as a common user with `ALTER PLUGGABLE DATABASE` privilege to unplug `pdb1_1`. If the `pdb1_1` is still in `READ WRITE` mode, close the PDB.

```
$ . oraenv
ORACLE_SID = [cdb2] ? cdb1
The Oracle base remains unchanged with value /u01/app/oracle

$ sqlplus / as sysdba

Connected to:
Oracle Database 12c Enterprise Edition Release 12.1.0.0.2 -
64bit Production
With the Partitioning, OLAP, Data Mining and Real Application
Testing options

SQL> select name, open_mode from v$pdbs;

NAME                           OPEN_MODE
------------------------------ ----------
PDB$SEED                       READ ONLY
PDB1_1                         READ WRITE

SQL> alter pluggable database PDB1_1 unplug into
'xmlfilePDB1_1.xml';
alter pluggable database PDB1_1 unplug into 'xmlfilePDB1_1'
*
ERROR at line 1:
ORA-65025: Pluggable database PDB1_1 is not closed on all
instances.

SQL> alter pluggable database PDB1_1 close immediate;
```

```
Pluggable database altered.

SQL>  alter pluggable database PDB1_1 unplug into
'xmlfilePDB1_1.xml';

Pluggable database altered.

SQL> col PDB_NAME format A20
SQL> select PDB_NAME, STATUS from CDB_PDBS
     where PDB_NAME='PDB1_1';

PDB_NAME              STATUS
-------------------- -------------
PDB1_1                UNPLUGGED

SQL> drop pluggable database PDB1_1 KEEP DATAFILES;

Pluggable database dropped.

SQL> EXIT
$
```

b. Before plugging `pdb1_1` into `cdb2`, you can optionally check whether the unplugged `pdb1_1` is compatible with `cdb2` with `DBMS_PDB.CHECK_PLUG_COMPATIBILITY` function. Connect to `cdb2` root as a common user with `CREATE PLUGGABLE DATABASE` privilege to plug `pdb1_1`.

Use the following PL/SQL code:

```
DECLARE
  compat BOOLEAN := FALSE;
  BEGIN
  compat := DBMS_PDB.CHECK_PLUG_COMPATIBILITY(
    pdb_descr_file =>
'/u01/app/oracle/product/12.1.0/dbhome_1/dbs/xmlfilePDB1_1.xml',
pdb_name => 'pdb1_1');
  if compat then
  DBMS_OUTPUT.PUT_LINE('Is pluggable compatible? YES');
  else DBMS_OUTPUT.PUT_LINE('Is pluggable compatible? NO');
  end if;
end;
/
```

```
$ . oraenv
ORACLE_SID = [cdb1] ? cdb2
The Oracle base remains unchanged with value /u01/app/oracle
```

```
$ sqlplus / as sysdba
Connected.

SQL> SET SERVEROUTPUT ON
SQL> DECLARE
  2  compat BOOLEAN := FALSE;
  3  BEGIN
  4   compat := DBMS_PDB.CHECK_PLUG_COMPATIBILITY(
  5  pdb_descr_file =>
'/u01/app/oracle/product/12.1.0/dbhome_1/dbs/xmlfilePDB1_1.xml',
pdb_name => 'pdb1_1');
  6  if compat then
  7       DBMS_OUTPUT.PUT_LINE('Is pluggable compatible? YES');
  8  else DBMS_OUTPUT.PUT_LINE('Is pluggable compatible? NO');
  9  end if;
 10  end;
 11  /
Is pluggable compatible? NO

PL/SQL procedure successfully completed.

SQL>
```

c. If the value returned is YES, you can immediately proceed with step d.
If the value returned is NO, examine the `PDB_PLUG_IN_VIOLATIONS` view to see why it is not compatible.

```
SQL> select message, action from pdb_plug_in_violations
     where name='PDB1_1';

MESSAGE
-----------------------------------------------------------------
-
ACTION
-----------------------------------------------------------------
-
Parameter sga_target mismatch: Previous value 503316480. CDB
value 5033164800
Change the parameter in PDB or the CDB


Parameter pga_aggregate_target mismatch: Previous value
167772160. CDB value 1677721600
Change the parameter in PDB or the CDB


SQL>
```

The message refers to a parameter related to PGA. The parameter will not have any impact if you create the PDB. You can proceed with the creation of the PDB.

d. Plug pdb1_1 into cdb2.

```
SQL> create pluggable database pdb1_1 using 'xmlfilePDB1_1.xml'
NOCOPY;

Pluggable database created.

SQL>
```

Notice that you use the clause NOCOPY because the cdb2 pdb1_1 files are located in the right place. Otherwise, you should have described the target destination to move the files from the source to the new destination.

e. Open pdb1_1.

```
SQL> alter pluggable database pdb1_1 open;

Pluggable database altered.

SQL>
```

f. Check that the pdb1_1 is in the PDBs list in cdb2.

```
SQL> select name, open_mode from v$pdbs;

NAME                           OPEN_MODE
------------------------------ ----------
PDB$SEED                       READ ONLY
PDB2_1                         READ WRITE
PDB2_2                         READ WRITE
PDB_ORCL2                      READ WRITE
PDB1_1                         READ WRITE

SQL> EXIT
$
```

2. After all PDBs are unplugged from cdb1 (in case you had created other PDBs) and plugged into cdb2, you can drop the multitenant container database cdb1 with DBCA or SQL commands.

```
$ . oraenv
ORACLE_SID = [cdb2] ? cdb1
The Oracle base remains unchanged with value /u01/app/oracle

$ sqlplus / as sysdba
Connected.
SQL> shutdown immediate
Database closed.
Database dismounted.
ORACLE instance shut down.

SQL> startup mount restrict
ORACLE instance started.

Total System Global Area  722366464 bytes
Fixed Size                  2276928 bytes
Variable Size             213909952 bytes
Database Buffers          503316480 bytes
Redo Buffers                2863104 bytes
Database mounted.

SQL> DROP DATABASE;

Database dropped.

SQL> EXIT
$
```

Remove archived logs and backups if necessary.

Practice 3-7: Dropping a PDB (optional)

Overview

In this practice, you drop the PDB `pdb2_3` of `cdb2` that you quickly create from the seed.

Tasks

Either use DBCA or SQL Developer or SQL commands.

The creation using SQL is described below.

1. Create a directory for the new data files of `pdb2_3` of `cdb2`.

```
$ . oraenv
ORACLE_SID = [cdb1] ? cdb2
The Oracle base remains unchanged with value /u01/app/oracle

$ cd $ORACLE_BASE/oradata/cdb2
$ mkdir pdb2_3
$
```

2. Run SQL*Plus and connect to the root with a user with `CREATE PLUGGABLE DATABASE` privilege.

```
$ sqlplus / as sysdba

Connected to:
Oracle Database 12c Enterprise Edition Release 12.1.0.0.2 -
64bit Production
With the Partitioning, OLAP, Data Mining and Real Application
Testing options

SQL> CREATE PLUGGABLE DATABASE pdb2_3 ADMIN USER pdb2_3_admin
       IDENTIFIED BY oracle_4U ROLES=(CONNECT)
       FILE_NAME_CONVERT=('/u01/app/oracle/oradata/cdb2/pdbseed'
                         ,'/u01/app/oracle/oradata/cdb2/pdb2_3');

Pluggable database created.

SQL>
```

3. Check the open mode of pdb2_3.

```
SQL> col con_id format 999
SQL> col name format A10
SQL> select NAME, OPEN_MODE from V$PDBS;

NAME                           OPEN_MODE
------------------------------ ----------
PDB$SEED                        READ ONLY
PDB2_1                          READ WRITE
PDB2_2                          READ WRITE
PDB_ORCL2                       READ WRITE
PDB1_1                          READ WRITE
PDB2_3                          MOUNTED

SQL>
```

4. Open pdb2_3.

```
SQL> alter pluggable database pdb2_3 open;

Pluggable database altered.

SQL>
```

5. Still connected to cdb2 as a common user with ALTER PLUGGABLE DATABASE privilege, you close pdb2_3 if the PDB is not already in MOUNTED mode.

```
SQL> alter pluggable database PDB2_3 close immediate;

Pluggable database altered.

SQL>
```

6. You drop the PDB pdb2_3 and the data files .

```
SQL> drop pluggable database PDB2_3 INCLUDING DATAFILES;

Pluggable database dropped.

SQL>  EXIT
$
```

Practices for Lesson 4: Managing a Multitenant Container Database and Pluggable Databases

Chapter 4

Practices for Lesson 4

Practices Overview

In this practice, you will perform startup and shutdown operations on CDBs, open and close operations on PDBs, and connections to PDBs to display current context.

Assumptions

cdb2 is successfully created after Practice 3-1.

pdb2_1 is successfully created in cdb2 after completion of Practice 3-3.

pdb2_2 is successfully created in cdb2 after completion of Practice 3-4.

It is not necessary at this step to have successfully created pdb1_1 and pdb_orcl2.

Practice 4-1: Shutdown and Startup of the CDB

Overview

In this practice you shut down `cdb2` and start up `cdb2`.

Tasks

1. Connect to the multitenant container database `cdb2` to shut it down.

   a. Connect to the CDB as a user with `SYSDBA` privilege.

```
$ . oraenv
ORACLE_SID = [cdb2] ? cdb2
The Oracle base remains unchanged with value /u01/app/oracle

$ sqlplus / as sysdba


Connected to:
Oracle Database 12c Enterprise Edition Release 12.1.0.0.2 -
64bit Production
With the Partitioning, OLAP, Data Mining and Real Application
Testing options

SQL> select name, cdb, con_id from v$database;
NAME      CDB     CON_ID
--------- --- ----------
CDB2      YES          0

SQL>
```

   b. Shut down the CDB.

```
SQL> shutdown immediate
Database closed.
Database dismounted.
ORACLE instance shut down.

SQL> EXIT
$
```

   c. Explore the background processes.

```
$ ps -ef|grep cdb2
oracle   22991  2686  0 13:56 pts/2    00:00:00 grep cdb2
$
```

2. Connect to the multitenant container database `cdb2` and start it up.

```
$ sqlplus / as sysdba

Connected to an idle instance.

SQL> startup
ORACLE instance started.

Total System Global Area 1068937216 bytes
Fixed Size                  2248280 bytes
Variable Size             343933352 bytes
Database Buffers          717225984 bytes
Redo Buffers                5529600 bytes
Database mounted.
Database opened.

SQL> select name, cdb, con_id from v$database;
NAME      CDB     CON_ID
--------- --- ----------
CDB2      YES          0

SQL> EXIT
$
```

3. Explore the background processes.

```
$ ps -ef|grep cdb2
oracle   26393     1  0 03:11 ?        00:00:00 ora_pmon_cdb2
oracle   26395     1  0 03:11 ?        00:00:00 ora_psp0_cdb2
oracle   26397     1  0 03:11 ?        00:00:00 ora_vktm_cdb2
oracle   26403     1  0 03:11 ?        00:00:00 ora_gen0_cdb2
oracle   26405     1  0 03:11 ?        00:00:00 ora_mman_cdb2
oracle   26409     1  0 03:11 ?        00:00:00 ora_diag_cdb2
oracle   26411     1  0 03:11 ?        00:00:00 ora_ofsd_cdb2
oracle   26413     1  0 03:11 ?        00:00:00 ora_dbrm_cdb2
oracle   26415     1  0 03:11 ?        00:00:00 ora_dia0_cdb2
oracle   26417     1  0 03:11 ?        00:00:00 ora_dbw0_cdb2
oracle   26419     1  0 03:11 ?        00:00:00 ora_lgwr_cdb2
oracle   26421     1  0 03:11 ?        00:00:00 ora_ckpt_cdb2
oracle   26423     1  0 03:11 ?        00:00:00 ora_lg00_cdb2
oracle   26425     1  0 03:11 ?        00:00:00 ora_lg01_cdb2
oracle   26427     1  0 03:11 ?        00:00:00 ora_smon_cdb2
oracle   26429     1  0 03:11 ?        00:00:00 ora_reco_cdb2
oracle   26431     1  0 03:11 ?        00:00:00 ora_lreg_cdb2
oracle   26433     1  1 03:11 ?        00:00:00 ora_mmon_cdb2
oracle   26435     1  0 03:11 ?        00:00:00 ora_mmnl_cdb2
oracle   26437     1  0 03:11 ?        00:00:00 ora_d000_cdb2
oracle   26439     1  0 03:11 ?        00:00:00 ora_s000_cdb2
oracle   26451     1  0 03:11 ?        00:00:00 ora_tmon_cdb2
oracle   26453     1  0 03:11 ?        00:00:00 ora_tt00_cdb2
oracle   26455     1  0 03:11 ?        00:00:00 ora_smco_cdb2
oracle   26457     1  0 03:11 ?        00:00:00 ora_fbda_cdb2
oracle   26459     1  0 03:11 ?        00:00:00 ora_aqpc_cdb2
oracle   26461     1  0 03:11 ?        00:00:00 ora_w000_cdb2
oracle   26465     1  0 03:12 ?        00:00:00 ora_p000_cdb2
oracle   26467     1  0 03:12 ?        00:00:00 ora_p001_cdb2
oracle   26469     1  0 03:12 ?        00:00:00 ora_p002_cdb2
oracle   26471     1  0 03:12 ?        00:00:00 ora_p003_cdb2
oracle   26473     1  0 03:12 ?        00:00:00 ora_p004_cdb2
oracle   26475     1  0 03:12 ?        00:00:00 ora_p005_cdb2
oracle   26477     1  0 03:12 ?        00:00:00 ora_p006_cdb2
oracle   26479     1  0 03:12 ?        00:00:00 ora_p007_cdb2
oracle   26505     1  0 03:12 ?        00:00:00 ora_qm02_cdb2
oracle   26507     1  0 03:12 ?        00:00:00 ora_q001_cdb2
oracle   26509     1  0 03:12 ?        00:00:00 ora_q002_cdb2
oracle   26513     1  1 03:12 ?        00:00:00 ora_cjq0_cdb2
oracle   26549  9398  0 03:12 pts/0    00:00:00 grep cdb2
$
```

4. Explore the PDBs.

```
$ sqlplus / as sysdba

Connected to:
Oracle Database 12c Enterprise Edition Release 12.1.0.0.2 -
64bit Production
With the Partitioning, OLAP, Data Mining and Real Application
Testing options
SQL> select CON_ID, NAME, OPEN_MODE from v$pdbs;

    CON_ID NAME                            OPEN_MODE
---------- -------------------------------- ----------
     2 PDB$SEED                           READ ONLY
     3 PDB2_1                             MOUNTED
     4 PDB2_2                             MOUNTED
     5 PDB_ORCL2                          MOUNTED
     6 PDB1_1                             MOUNTED

SQL>
```

5. Open all PDBs.

```
SQL> alter pluggable database all open;
Pluggable database altered.

SQL> select CON_ID, NAME, OPEN_MODE from v$pdbs;

    CON_ID NAME                           OPEN_MODE
---------- ------------------------------ ----------
     2 PDB$SEED                           READ ONLY
     3 PDB2_1                             READ WRITE
     4 PDB2_2                             READ WRITE
     5 PDB_ORCL2                          READ WRITE
     6 PDB1_1                             READ WRITE

SQL>
```

6. Connect to any of the PDBs in your `cdb2`, except `PDB$SEED`.

```
SQL> connect sys/oracle_4U@PDB2_1 AS SYSDBA
Connected.
SQL> select CON_ID, NAME, OPEN_MODE from v$pdbs;

    CON_ID NAME                           OPEN_MODE
---------- ------------------------------ ----------
         3 PDB2_1                         READ WRITE

SQL>
```

7. Display the context of the PDB you are connected to.

```
SQL> show con_name

CON_NAME
------------------------------
PDB2_1

SQL>
```

8. Connect to another PDB left in your `cdb2`, except `PDB$SEED`.

```
SQL> connect sys/oracle_4U@PDB2_2 AS SYSDBA
Connected.
SQL> select CON_ID, NAME, OPEN_MODE from v$pdbs;

    CON_ID NAME                           OPEN_MODE
---------- ------------------------------ ----------
         4 PDB2_2                         READ WRITE

SQL>
```

9. Display the context of the PDB you are connected to.

```
SQL> show con_name

CON_NAME
------------------------------
PDB2_2

SQL> EXIT
$
```

Practice 4-2: Closing and Opening a PDB

Overview

In this practice you close PDBs and open PDBs, and create triggers to automatically open PDBs after CDB startup.

Tasks

1. Connect to the multitenant container database `cdb2` to shut it down.
   a. Connect to `cdb2` as a user with `SYSDBA` privilege.

```
$ . oraenv
ORACLE_SID = [cdb2] ? cdb2
The Oracle base remains unchanged with value /u01/app/oracle

$ sqlplus / as sysdba

Connected to:
Oracle Database 12c Enterprise Edition Release 12.1.0.0.2 -
64bit Production
With the Partitioning, OLAP, Data Mining and Real Application
Testing options

SQL> select name, cdb, con_id from v$database;
NAME      CDB     CON_ID
--------- --- ----------
CDB2      YES          0

SQL>
```

   b. Shut down `cdb2`.

```
SQL> shutdown immediate
Database closed.
Database dismounted.
ORACLE instance shut down.
SQL>
```

   c. Start up `cdb2`.

```
SQL> startup
ORACLE instance started.

Total System Global Area 1068937216 bytes
Fixed Size                  2248280 bytes
Variable Size             343933352 bytes
Database Buffers          717225984 bytes
Redo Buffers                5529600 bytes
```

```
Database mounted.
Database opened.
SQL>
```

d. Notice that the PDBs are all in MOUNTED open mode.

```
SQL> select CON_ID, NAME, OPEN_MODE from v$pdbs;
```

```
    CON_ID NAME                           OPEN_MODE
---------- ------------------------------ ----------
         2 PDB$SEED                       READ ONLY
         3 PDB2_1                         MOUNTED
         4 PDB2_2                         MOUNTED
         5 PDB_ORCL2                      MOUNTED
         6 PDB1_1                         MOUNTED

SQL>
```

2. Open all PDBs manually.

```
SQL> alter pluggable database all open;

Pluggable database altered.

SQL>
```

3. Close PDB2_1.

a. Start a DML transaction in another session.

```
$ . oraenv
ORACLE_SID = [cdb2] ? cdb2
The Oracle base remains unchanged with value /u01/app/oracle
$ sqlplus sys/oracle_4U@pdb2_1 as sysdba

Connected to:
Oracle Database 12c Enterprise Edition Release 12.1.0.0.2 -
64bit Production
With the Partitioning, OLAP, Data Mining and Real Application
Testing options

SQL> create table system.mytab (c number);
Table created.

SQL> insert into system.mytab values (1);
1 row created.

SQL> commit;
```

```
Commit complete.

SQL> exit
$
```

b. In the first session, close `PDB2_1` in `IMMEDIATE` mode.

```
SQL> alter pluggable database pdb2_1 close immediate;
Pluggable database altered.

SQL> select CON_ID, NAME, OPEN_MODE from v$pdbs;

    CON_ID NAME                           OPEN_MODE
---------- ------------------------------ ----------
         2 PDB$SEED                       READ ONLY
         3 PDB2_1                         MOUNTED
         4 PDB2_2                         READ WRITE
         5 PDB_ORCL2                      READ WRITE
         6 PDB1_1                         READ WRITE

SQL>
```

c. Try to connect as a user of `PDB2_1`.

```
SQL> connect system/oracle_4U@pdb2_1
ERROR:
ORA-01033: ORACLE initialization or shutdown in progress
Process ID: 0
Session ID: 0 Serial number: 0


Warning: You are no longer connected to ORACLE.

SQL>
```

4. Open `pdb2_1`.

```
SQL> connect / as sysdba
Connected.
SQL> alter pluggable database PDB2_1 open;

Pluggable database altered.

SQL>
```

Reconnect to `pdb2_1` and select data from `SYSTEM.MYTAB` table.

```
SQL> connect system/oracle_4U@PDB2_1
Connected.
```

```
SQL> select * from system.mytab;

         C
----------
         1

SQL>
```

5. Shut down the multitenant container database `cdb2` to open and close PDBs with different clauses.

```
SQL> CONNECT / AS SYSDBA
Connected.

SQL> select name, cdb, con_id from v$database;

NAME      CDB     CON_ID
--------- --- ----------
CDB2      YES          0

SQL>
```

a. Shut down `CDB2`.

```
SQL> shutdown immediate
Database closed.
Database dismounted.
ORACLE instance shut down.
SQL>
```

b. Start up `cdb2` in `NOMOUNT` mode.

```
SQL> startup nomount
ORACLE instance started.

Total System Global Area 1068937216 bytes
Fixed Size                  2248280 bytes
Variable Size             343933352 bytes
Database Buffers          717225984 bytes
Redo Buffers                5529600 bytes

SQL> select CON_ID, NAME, OPEN_MODE from v$pdbs;

No rows selected.

SQL>
```

c. Mount cdb2.

```
SQL> alter database mount;

Database altered.

SQL>
SQL> select CON_ID, NAME, OPEN_MODE from v$pdbs;

    CON_ID NAME                             OPEN_MODE
---------- -------------------------------- ----------
         2 PDB$SEED                         MOUNTED
         3 PDB2_1                           MOUNTED
         4 PDB2_2                           MOUNTED
         5 PDB_ORCL2                        MOUNTED
         6 PDB1_1                           MOUNTED
SQL>
```

d. Open cdb2.

```
SQL> alter database open;
Database altered.
```

```
SQL> select CON_ID, NAME, OPEN_MODE from v$pdbs;

    CON_ID NAME                             OPEN_MODE
---------- -------------------------------- ----------
         2 PDB$SEED                         READ ONLY
         3 PDB2_1                           MOUNTED
         4 PDB2_2                           MOUNTED
         5 PDB_ORCL2                        MOUNTED
         6 PDB1_1                           MOUNTED
SQL>
```

e. Open all PDBs except PDB2_2.

```
SQL> alter pluggable database all except pdb2_2 open;
Pluggable database altered.
SQL> select CON_ID, NAME, OPEN_MODE from v$pdbs;

    CON_ID NAME                           OPEN_MODE
---------- ------------------------------ ----------
         2 PDB$SEED                        READ ONLY
         3 PDB2_1                          READ WRITE
         4 PDB2_2                          MOUNTED
         5 PDB_ORCL2                       READ WRITE
         6 PDB1_1                          READ WRITE
SQL>
```

Practice 4-3: Creating After Startup Trigger to Open All PDBs

Overview

In this practice, you create `AFTER STARTUP` trigger to open all PDBs of a CDB.

Tasks

1. Create a trigger in `cdb2` to open all PDBs automatically after starting up `cdb2`.
   a. Create the trigger.

```
CREATE TRIGGER open_all_PDBs
  AFTER STARTUP ON DATABASE
begin
    execute immediate 'alter pluggable database all open';
 end open_all_PDBs;
/
```

```
SQL> CREATE TRIGGER Open_All_PDBs
  2  after startup on database
  3  begin
  4      execute immediate 'alter pluggable database ALL open';
  5  end Open_All_PDBs;
  6  /

Trigger created.

SQL>
```

   b. Shut down `cdb2`.

```
SQL> shutdown immediate
Database closed.
Database dismounted.
ORACLE instance shut down.
SQL>
```

   c. Start up `cdb2`.

```
SQL> startup
ORACLE instance started.

Total System Global Area 1068937216 bytes
Fixed Size                  2248280 bytes
Variable Size             343933352 bytes
Database Buffers          717225984 bytes
Redo Buffers                5529600 bytes
Database mounted.
Database opened.
SQL>
```

d. Notice that the PDBs are all in `READ WRITE` open mode. Then disconnect.

```
SQL> select CON_ID, NAME, OPEN_MODE from v$pdbs;


    CON_ID NAME                             OPEN_MODE
---------- -------------------------------- ----------
         2 PDB$SEED                          READ ONLY
         3 PDB2_1                            READ WRITE
         4 PDB2_2                            READ WRITE
         5 PDB_ORCL2                         READ WRITE
         6 PDB1_1                            READ WRITE

SQL>
```

Practice 4-4: Changing PDBs' Open Mode

Overview

In this practice you will change the open mode of PDBs for specific operations.

Assumptions

If the trigger could not be created successfully, execute the following catchup script:

```
$ cd /home/oracle/recovery/catchup_04_03
$ ./cr_trig.sh
$
```

Tasks

Rename the global database name for `pdb2_1` to `pdb2` in `cdb2`. For this purpose, you must open the PDB in `RESTRICTED` mode.

1. Connect to `pdb2_1`.

```
SQL> CONNECT sys/oracle_4U@pdb2_1 as sysdba
Connected.
SQL>
```

2. Change the global database name for `pdb2_1` to `pdb2`.

```
SQL> alter pluggable database RENAME GLOBAL_NAME TO pdb2;
alter pluggable database RENAME global_name to pdb2
                                               *
ERROR at line 1:
ORA-65045: pluggable database not in a restricted mode

SQL>
```

3. Close `pdb2_1`.

```
SQL> alter pluggable database close immediate;

Pluggable database altered.

SQL>
```

4. Open `pdb2_1` in restricted mode.

```
SQL> alter pluggable database open restricted;

Pluggable database altered.

SQL>
```

```
SQL> select CON_ID, NAME, OPEN_MODE, RESTRICTED from v$pdbs;

    CON_ID NAME                                  OPEN_MODE  RES
```

```
---------- ------------------------------ ---------- ---
         3 PDB2_1                         READ WRITE YES

SQL>
```

5. Change the global database name for `pdb2_1` to `pdb2`.

```
SQL> alter pluggable database RENAME GLOBAL_NAME TO pdb2;

Pluggable database altered.

SQL>
```

```
SQL> select CON_ID, NAME, OPEN_MODE, RESTRICTED from v$pdbs;

    CON_ID NAME                           OPEN_MODE  RES
---------- ------------------------------ ---------- ---
         3 PDB2                           READ WRITE YES

SQL>
```

6. Open `PDB2`.

```
SQL> alter pluggable database close immediate;

Pluggable database altered.

SQL> alter pluggable database open;

Pluggable database altered.

SQL>
```

7. Check `PDB2` is in `READ WRITE` mode.

```
SQL> select CON_ID, NAME, OPEN_MODE, RESTRICTED from v$pdbs;

    CON_ID NAME                           OPEN_MODE  RES
---------- ------------------------------ ---------- ---
         3 PDB2                           READ WRITE NO

SQL>
```

Practice 4-5: Instance Parameter Changes: Impact on PDBs (optional)

Overview

In this practice you will discover the impact of instance parameter changes on PDBs.

Tasks

1. In this example, you will use in `cdb2` the instance parameter `OPTIMIZER_USE_SQL_PLAN_BASELINES` because it is `ISPDB_MODIFIABLE` in `V$PARAMETER.`

```
SQL> CONNECT / AS SYSDBA
Connected.
SQL> select ISPDB_MODIFIABLE from v$parameter
  2  where name='optimizer_use_sql_plan_baselines';

ISPDB
-----
TRUE

SQL>
```

2. Check the current value of instance parameter `OPTIMIZER_USE_SQL_PLAN_BASELINES`.

```
SQL> show parameter optimizer_use_sql_plan_baselines
NAME                                 TYPE        VALUE
------------------------------------ ----------- ---------
optimizer_use_sql_plan_baselines     boolean      TRUE

SQL> EXIT
$
```

3. Connect to `pdb2` in `cdb2` and check the current value of the same instance parameter `OPTIMIZER_USE_SQL_PLAN_BASELINES`.
   1) Use netca to add the `PDB2` net service name for `pdb2` pluggable database of `cdb2` in the `tnsnames.ora` file.

```
$ netca
```

   2) On the Welcome page, select the “Local Net Service Name configuration” and click Next.
   3) On the Net Service Name Configuration page, accept `Add` and click Next.
   4) On the Net Service Name Configuration, Service Name page, enter `pdb2` as Service Name and click Next.
   5) On the Net Service Name Configuration, Select Protocols page, select TCP and click Next.
   6) On the Net Service Name Configuration, TCP/IP Protocol page, enter your complete host name, for example, *`<yourservername>`*`.us.oracle.com`, or *`localhost`*, accept “Use the standard port number of 1521,” and click Next.

7) On the Net Service Name Configuration, Test page, select “No, do not test” (the pluggable database is not yet opened) and click Next.
8) On the Net Service Name Configuration, Net Service Name page, accept `pdb2` as Net Service Name and click Next.
9) On the Net Service Name Configuration, Another Net Service Name page, select No, and Next.
10) On the Net Service Name Configuration Done page, click Next.
11) When you are back on the Welcome page, click Finish.
12) Reload the listener with the new configuration:

```
$ lsnrctl reload
```

If this is not sufficient, then restart the instance.

a. Connect to `pdb2` in `cdb2`.

```
$ sqlplus sys/oracle_4U@pdb2 AS SYSDBA


Connected to:
Oracle Database 12c Enterprise Edition Release 12.1.0.0.2 -
64bit Production
With the Partitioning, OLAP, Data Mining, Real Application
Testing


SQL> show parameter optimizer_use_sql_plan_baselines
NAME                                 TYPE        VALUE
------------------------------------ ----------- ---------
optimizer_use_sql_plan_baselines     boolean      TRUE


SQL>
```

4. Change the instance parameter value to `FALSE` in `pdb2`.

```
SQL> ALTER SYSTEM SET optimizer_use_sql_plan_baselines= FALSE
SCOPE=BOTH;
System altered.
SQL>
SQL> show parameter optimizer_use_sql_plan_baselines
NAME                                 TYPE        VALUE
------------------------------------ ----------- ---------
optimizer_use_sql_plan_baselines     boolean     FALSE


SQL>
```

5. Check the instance parameter value in other PDBs of the same CDB.

```
SQL> CONNECT sys/oracle_4U@pdb2_2 AS SYSDBA
Connected.
SQL> show parameter optimizer_use_sql_plan_baselines
NAME                                 TYPE        VALUE
------------------------------------ ----------- ---------
```

```
optimizer_use_sql_plan_baselines     boolean     TRUE

SQL>
```

6. Close and open `pdb2`.

```
SQL> CONNECT sys/oracle_4U@pdb2 AS SYSDBA
Connected.
SQL> ALTER PLUGGABLE DATABASE CLOSE IMMEDIATE;
Pluggable database altered.

SQL> ALTER PLUGGABLE DATABASE OPEN;
Pluggable database altered.

SQL> show parameter optimizer_use_sql_plan_baselines
NAME                                 TYPE        VALUE
------------------------------------ ----------- ---------
optimizer_use_sql_plan_baselines     boolean     FALSE

SQL>
```

7. Check the instance parameter value after CDB shutdown/startup both in root and PDBs.

```
SQL> connect / as sysdba
Connected.
SQL> shutdown immediate
Database closed.
Database dismounted.
ORACLE instance shut down.
SQL>
SQL> startup
ORACLE instance started.

Total System Global Area 1068937216 bytes
Fixed Size                  2248280 bytes
Variable Size             377487784 bytes
Database Buffers          683671552 bytes
Redo Buffers                5529600 bytes
Database mounted.
Database opened.
SQL> col VALUE format a20
SQL> select CON_ID, VALUE from   V$SYSTEM_PARAMETER
  2  where name ='optimizer_use_sql_plan_baselines';

    CON_ID VALUE
---------- --------------------
```

```
         0 TRUE
         3 FALSE

SQL> EXIT
$
```

Practices for Lesson 5: Managing Tablespaces and Users in a CDB and PDBs

Chapter 5

Practices for Lesson 5

Practices Overview

In this practice, you will manage the tablespaces, users, privileges, and roles.

Assumptions

Practice 3-1 successfully created `cdb2`.

Practice 3-3 successfully created `pdb2_1`.

Practice 4-4 successfully renamed `pdb2_1` to `pdb2`.

If the trigger could not be created successfully, execute the following catchup script:

```
$ cd /home/oracle/recovery/catchup_04_03
$ ./cr_trig.sh
$
```

Practice 5-1: Managing Tablespaces

Overview

In this practice, you will manage the tablespaces in the CDB and PDBs.

Tasks

1. View permanent and temporary tablespace properties in `cdb2`.

```
$ . oraenv
ORACLE_SID = [cdb2] ? cdb2
The Oracle base remains unchanged with value /u01/app/oracle

$ sqlplus / as sysdba

Connected to:
Oracle Database 12c Enterprise Edition Release 12.1.0.0.2 -
64bit Production
With the Partitioning, OLAP, Data Mining and Real Application
Testing options

SQL> col PROPERTY_NAME format a30
SQL> col PROPERTY_VALUE  format a25
SQL> SELECT property_name, property_value
  2  FROM database_properties
  3  WHERE property_name LIKE 'DEFAULT_%TABLE%';

PROPERTY_NAME                  PROPERTY_VALUE
------------------------------ -------------------------------
DEFAULT_TEMP_TABLESPACE        TEMP
DEFAULT_PERMANENT_TABLESPACE   USERS

SQL> SELECT tablespace_name, CON_ID from CDB_TABLESPACES;

TABLESPACE_NAME                    CON_ID
------------------------------ ----------
SYSTEM                                  1
SYSAUX                                  1
UNDOTBS1                                1
TEMP                                    1
USERS                                   1
SYSTEM                                  2
SYSAUX                                  2
TEMP                                    2
SYSTEM                                  3
```

```
SYSAUX                                  3
TEMP                                    3
SYSTEM                                  4
SYSAUX                                  4
TEMP                                    4
SYSTEM                                  5
SYSAUX                                  5
TEMP                                    5
USERS                                  5
EXAMPLE                                5
SYSTEM                                  6
SYSAUX                                  6
TEMP                                    6
USERS                                  6
EXAMPLE                                6

24 rows selected.

SQL> SELECT tablespace_name, CON_ID from CDB_TABLESPACES
  2 WHERE TABLESPACE_NAME LIKE 'TEMP%';

TABLESPACE_NAME                    CON_ID
------------------------------ ----------
TEMP                                    1
TEMP                                    2
TEMP                                    3
TEMP                                    4
TEMP                                    5
TEMP                                    6

6 rows selected.

SQL>
```

The number of tablespaces may differ from your result and the one shown in the first statement. It depends if PDBs are created with or without the `EXAMPLE` and the `USERS` tablespaces.

Manage permanent tablespaces.

a. Create a permanent tablespace `CDATA` in the root container.

```
SQL> CREATE TABLESPACE CDATA
  2 DATAFILE '/u01/app/oracle/oradata/cdb2/cdata_01.dbf'
  3 SIZE 10M ;
```

```
Tablespace created.

SQL> SELECT tablespace_name, CON_ID from CDB_TABLESPACES
  2  WHERE TABLESPACE_NAME = 'CDATA';

TABLESPACE_NAME                    CON_ID
------------------------------ ----------
CDATA                                   1

SQL>
```

b. Make the CDATA tablespace the default tablespace in the root container.

```
SQL> ALTER DATABASE DEFAULT TABLESPACE CDATA ;
Database altered.

SQL> SELECT property_name, property_value
  2  FROM   database_properties
  3  WHERE property_name LIKE 'DEFAULT_%TABLE%';

PROPERTY_NAME                  PROPERTY_VALUE
------------------------------ ------------------------------
DEFAULT_TEMP_TABLESPACE        TEMP
DEFAULT_PERMANENT_TABLESPACE   CDATA

SQL>
```

c. Create permanent tablespace, LDATA in PDB2.

```
SQL> connect system/oracle_4U@PDB2
Connected.

SQL> CREATE TABLESPACE ldata DATAFILE
  2  '/u01/app/oracle/oradata/cdb2/pdb2_1/ldata_01.dbf'
  3  SIZE 10M ;

Tablespace created.

SQL>
```

d. Make the LDATA tablespace the default tablepace in the PDB2 container.

```
SQL> ALTER PLUGGABLE DATABASE DEFAULT TABLESPACE LDATA ;
Pluggable database altered.

SQL> SELECT property_name, property_value
  2  FROM database_properties
  3  WHERE property_name LIKE 'DEFAULT_%TABLE%';

PROPERTY_NAME                  PROPERTY_VALUE
------------------------------ ------------------------------
DEFAULT_TEMP_TABLESPACE        TEMP
DEFAULT_PERMANENT_TABLESPACE   LDATA

SQL>
```

3. Manage temporary tablespaces (**optional**).
   a. Create a temporary tablespace `TEMP_ROOT` in the root container.

```
SQL> connect system/oracle_4U
Connected.
SQL> CREATE TEMPORARY TABLESPACE TEMP_ROOT
  2  TEMPFILE '/u01/app/oracle/oradata/cdb2/temproot_01.dbf'
  3  SIZE 100M ;

Tablespace created.

SQL>
```

   b. Make `TEMP_ROOT` the default temporary tablespace in the root container.

```
SQL> ALTER DATABASE DEFAULT TEMPORARY TABLESPACE TEMP_ROOT ;

Database altered.

SQL> SELECT property_name, property_value
  2  FROM database_properties
  3  WHERE property_name LIKE 'DEFAULT_%TABLE%';

PROPERTY_NAME                  PROPERTY_VALUE
------------------------------ ------------------------------
DEFAULT_TEMP_TABLESPACE        TEMP_ROOT
DEFAULT_PERMANENT_TABLESPACE   CDATA

SQL>
```

   c. Create a temporary tablepace `TEMP_PDB2` in `PDB2`.

```
SQL> connect system/oracle_4U@PDB2
Connected.

SQL> CREATE TEMPORARY TABLESPACE TEMP_PDB2 TEMPFILE
  2      '/u01/app/oracle/oradata/cdb2/pdb2_1/temppdb2_01.dbf'
  3      SIZE 100M ;
```

```

Tablespace created.

SQL>
```

d. Make `TEMP_PDB2` the default temporary tablespace in `PDB2`.

```
SQL> ALTER DATABASE DEFAULT TEMPORARY TABLESPACE TEMP_PDB2 ;

Database altered.

SQL> SELECT property_name, property_value
  2  FROM database_properties
  3  WHERE property_name LIKE 'DEFAULT_%TABLE%';

PROPERTY_NAME                  PROPERTY_VALUE
------------------------------ ------------------------------
DEFAULT_TEMP_TABLESPACE        TEMP_PDB2
DEFAULT_PERMANENT_TABLESPACE   LDATA

SQL>
```

Note that you could also use the `ALTER PLUGGABLE DATABASE` command.

e. Create a temporary tablespace `MY_TEMP` in `PDB2`.

```
SQL> CREATE TEMPORARY TABLESPACE MY_TEMP TEMPFILE
  2  '/u01/app/oracle/oradata/cdb2/pdb2_1/my_temp_pdb2_01.dbf'
  3  SIZE 10M;

Tablespace created.

SQL>
```

f. Display default tablespaces of another PDB in `cdb2`.

```
SQL> connect system/oracle_4U@PDB_ORCL2
Connected.

SQL> SELECT property_name, property_value
  2  FROM database_properties
  3  WHERE property_name LIKE 'DEFAULT_%TABLE%';

PROPERTY_NAME                  PROPERTY_VALUE
------------------------------ ------------------------------
DEFAULT_TEMP_TABLESPACE        TEMP
DEFAULT_PERMANENT_TABLESPACE   USERS

SQL>
```

4. Manage default permanent and temporary tablespaces of users.

   a. Create a common user C##U.

```
SQL> connect system/oracle_4U
Connected.

SQL> CREATE USER c##u IDENTIFIED BY x;

User created.

SQL>
```

   b. View the default tablespace and temporary tablespace assignment for user CU in all containers.

```
SQL> COLUMN username format A12
SQL> COLUMN default_tablespace format A18
SQL> COLUMN temporary_tablespace format A20
SQL> COLUMN con_id format 999
SQL> SELECT username, default_tablespace,
  2  temporary_tablespace, con_id
  3  FROM CDB_USERS
  4  WHERE username = 'C##U';

USERNAME     DEFAULT_TABLESPACE TEMPORARY_TABLESPACE   CON_ID
------------ ------------------ -------------------- ------
C##U         CDATA              TEMP_ROOT                 1
C##U         LDATA              TEMP_PDB2                 3
C##U         SYSTEM             TEMP                      4
C##U         USERS              TEMP                      5
C##U         USERS              TEMP                      6

SQL>
```

   c. Create a local user LU in PDB2.

```
SQL> connect system/oracle_4U@PDB2
Connected.

SQL> CREATE USER lu IDENTIFIED BY x;

User created.

SQL>
```

   d. View the default tablespace and temporary tablespace assignment for user LU.

```
SQL> SELECT username, default_tablespace, temporary_tablespace
  2  FROM DBA_USERS
  3  WHERE username = 'LU';
```

```

USERNAME     DEFAULT_TABLESPACE TEMPORARY_TABLESPACE
------------ ------------------ --------------------
LU           LDATA              TEMP_PDB2

SQL>
```

e. Change the temporary tablespace assignment for user `LU` to `MY_TEMP` in `PDB2`.

```
SQL> ALTER USER LU TEMPORARY TABLESPACE MY_TEMP;

User altered.

SQL>
```

f. View the default temporary tablespace assignment for user `LU`.

```
SQL> SELECT username, default_tablespace, temporary_tablespace
  2  FROM DBA_USERS
  3  WHERE username = 'LU';

USERNAME     DEFAULT_TABLESPACE TEMPORARY_TABLESPACE
------------ ------------------ --------------------
LU           LDATA              MY_TEMP

SQL>
```

5. Manage UNDO tablespaces (**optional**).

a. Display the UNDO tablespace used in the CDB.

```
SQL> connect system/oracle_4U
Connected.
SQL> col NAME format A12
SQL> select FILE#, ts.name, ts.ts#, ts.con_id
  2  from v$datafile d, v$tablespace  ts
  3  where d.ts#=ts.ts#
  4  and   d.con_id=ts.con_id
  5  and   ts.name like 'UNDO%';

     FILE# NAME                TS#     CON_ID
---------- ------------ ---------- ----------
         4 UNDOTBS1              2          1

SQL>
```

b. Create an UNDO tablespace in a PDB and set it as the `UNDO_TABLESPACE` of the CDB.

```
SQL> connect system/oracle_4U@PDB2
Connected.
SQL> CREATE UNDO TABLESPACE UNDO_PDB2 DATAFILE
```

```
  2  '/u01/app/oracle/oradata/cdb2/pdb2/undo_pdb2_01.dbf'
  3  SIZE 10M;
Tablespace created.

SQL> alter system set undo_tablespace='UNDO_PDB2' scope=both;
alter system set undo_tablespace='UNDO_PDB2' scope=both
*
ERROR at line 1:
ORA-65040: operation not allowed from within a pluggable
database

SQL>
```

Notice that the statement fails because the UNDO tablespace can only be set at CDB level.

Practice 5-2: Managing Common and Local Users

Overview

In this practice, you will manage common and local users in the CDB and PDBs.

Tasks

1. View all common and local users in cdb2.

```
SQL> connect  / as sysdba
Connected.
SQL> col username format a20
SQL> select USERNAME,COMMON,CON_ID  from cdb_users;

USERNAME             COM CON_ID
-------------------- --- ------
SYS                  YES      1
SYSTEM               YES      1
C##U                 YES      1
…
C##U                 YES      3
JONES                NO       3
ADAMS                NO       3
SCOTT                NO       3
…
C##U                 YES      4
…
DVSYS                YES      4
AUDSYS               YES      4
SI_INFORMTN_SCHEMA   YES      4
OLAPSYS              YES      4

239 rows selected.

SQL> select USERNAME,COMMON,CON_ID  from cdb_users
  2  where username='SYSTEM';

USERNAME             COM CON_ID
-------------------- --- ------
SYSTEM               YES      1
SYSTEM               YES      2
SYSTEM               YES      3
SYSTEM               YES      4
SYSTEM               YES      5
SYSTEM               YES      6
```

```
6 rows selected.

SQL> select distinct  username from cdb_users
  2  where common='YES';

USERNAME
--------------------
DVF
SYSKM
DIP
XS$NULL
OUTLN
SYSBACKUP
SYSTEM
ORACLE_OCM
DVSYS
AUDSYS
DBSNMP
…
C##U
XDB
APPQOSSYS
SYSDG
ANONYMOUS
SYS
SI_INFORMTN_SCHEMA
ANONYMOUS
LBACSYS
WMSYS

37 rows selected.

SQL> select username,con_id  from cdb_users
  2  where common='NO';

USERNAME             CON_ID
-------------------- ------
PDB2_1_ADMIN              3
LU                       3
PDB2_1_ADMIN              4
SCOTT                     5
```

```
BI                         5
PM                         5
IX                         5
SH                         5
OE                         5
HR                         5
PDBADMIN                   6
HR                         6
OE                         6
SH                         6
IX                         6
PM                         6
BI                         6
SCOTT                       6

18 rows selected.

SQL>
```

2. Create a common user `C##_USER`.

```
SQL> create user C##_USER identified by x CONTAINER=ALL;

User created.

SQL>
```

3. View the new common user `C##_USER`.

```
SQL> select distinct  username from cdb_users
  2  where username='C##_USER';

USERNAME
--------------------
C##_USER

SQL>
```

Notice that the common user exists in each container.

4. Grant `CREATE SESSION` as a common privilege

```
SQL> GRANT CREATE SESSION TO c##_user CONTAINER=ALL;


Grant succeeded.


SQL>
```

5. Connect to root, `PDB2`, and `PDB2_2` as `c##_user` user.

```
SQL> connect  c##_user/x@pdb2
Connected.
SQL> connect  c##_user/x@pdb2_2
Connected.
SQL> connect  c##_user/x@cdb2
Connected.
SQL>
```

6. Create a local user `LOCAL_USER` in the root container.

```
SQL> connect / as sysdba
Connected.

SQL> create user local_user identified by x
  2  CONTAINER=CURRENT;
create user local_user identified by x
                                     *
ERROR at line 1:
ORA-65049: creation of local user or role is not allowed in
CDB$ROOT

SQL>
```

Notice that no local user is authorized in the root.

7. Create a local user `LOCAL_USER_PDB2` in `PDB2`.

   a. View all users of PDB2.

```
SQL> connect  sys/oracle_4U@PDB2 as sysdba
Connected.

SQL> col username format a25
SQL> select USERNAME,COMMON,CON_ID  from cdb_users;

USERNAME             COM     CON_ID
-------------------- --- ----------
SYS                  YES     3
SYSTEM               YES     3
OLAPSYS              YES     3
SI_INFORMTN_SCHEMA   YES     3
DVSYS                YES     3
AUDSYS               YES     3
GSMUSER              YES     3
ORDPLUGINS           YES     3
C##_USER             YES     3
SPATIAL_WFS_ADMIN_US YES     3
```

```
R
SPATIAL_CSW_ADMIN_US YES    3
R
XDB                  YES    3
APEX_PUBLIC_USER     YES    3
SYSDG               YES     3
DIP                  YES    3
OUTLN               YES     3
ANONYMOUS            YES    3
CTXSYS              YES     3
ORDDATA              YES    3
SYSBACKUP            YES    3
MDDATA              YES     3
GSMCATUSER          YES     3
GSMADMIN_INTERNAL    YES    3
LBACSYS              YES    3
C##U                 YES    3
SYSKM               YES     3
XS$NULL              YES    3
OJVMSYS              YES    3
APPQOSSYS            YES    3
ORACLE_OCM          YES     3
APEX_040200         YES     3
PDB2_1_ADMIN        NO      3
WMSYS               YES     3
DBSNMP              YES     3
ORDSYS              YES     3
MDSYS               YES     3
LU                   NO     3
DVF                  YES    3
FLOWS_FILES         YES     3

39 rows selected.

SQL>
```

Notice that you view all common and local users of the current PDB.

```
SQL> select USERNAME,COMMON from dba_users;
```

Notice that you view the same list.

b. Attempt to create a common user C##_USER_PDB2 in PDB2.

```
SQL> create user c##_user_pdb2 identified by x
  2  CONTAINER=ALL;
create user c##_user_pdb2 identified by x CONTAINER=ALL
```

```
*
ERROR at line 1:
ORA-65050: Common DDLs only allowed in CDB$ROOT

SQL>
```

Notice that no common user can be created except from the root.

c. Create the local user `LOCAL_USER_PDB2` in `PDB2`.

```
SQL> create user local_user_pdb2 identified by x
  2  CONTAINER=CURRENT;

User created.

SQL> select USERNAME,COMMON,CON_ID  from cdb_users
  2> order by username;

USERNAME             COM     CON_ID
-------------------- --- ----------
ANONYMOUS            YES    3
APEX_040200         YES     3
APEX_PUBLIC_USER     YES    3
APPQOSSYS            YES    3
AUDSYS              YES     3
C##U                 YES    3
C##_USER             YES    3
CTXSYS              YES     3
DBSNMP              YES     3
DIP                  YES    3
DVF                  YES    3
DVSYS               YES     3
FLOWS_FILES         YES     3
GSMADMIN_INTERNAL    YES    3
GSMCATUSER          YES     3
GSMUSER              YES    3
LBACSYS              YES    3
LOCAL_USER_PDB2      NO     3
LU                   NO     3
MDDATA              YES     3
MDSYS               YES     3
OJVMSYS              YES    3
OLAPSYS              YES    3
ORACLE_OCM          YES     3
ORDDATA              YES    3
```

```
ORDPLUGINS          YES     3
ORDSYS              YES     3
OUTLN               YES     3
PDB2_1_ADMIN        NO      3
SI_INFORMTN_SCHEMA   YES    3
SPATIAL_CSW_ADMIN_US YES    3
R
SPATIAL_WFS_ADMIN_US YES    3
R
SYS                  YES    3
SYSBACKUP            YES    3
SYSDG               YES     3
SYSKM               YES     3
SYSTEM              YES     3
WMSYS               YES     3
XDB                  YES    3
XS$NULL              YES    3

40 rows selected.

SQL> grant create session to local_user_pdb2;

Grant succeeded.

SQL>
```

d. Connect to `PDB2` as `LOCAL_USER_PDB2`.

```
SQL> connect local_user_pdb2/x@PDB2
Connected.
SQL>
```

e. Connect to `PDB2_2` as `LOCAL_USER_PDB2`.

```
SQL> connect local_user_pdb2/x@PDB2_2
ERROR:
ORA-01017: invalid username/password; logon denied

Warning: You are no longer connected to ORACLE.

SQL>
```

Notice that it fails because `LOCAL_USER_PDB2` does not exist in `PDB2_2`.

```
SQL> connect  local_user_pdb2/x@cdb2
ERROR:
ORA-01017: invalid username/password; logon denied
SQL>
```

Notice that it fails because LOCAL_USER_PDB2 does not exist in root.

f. Overview of common and local users from a PDB.

```
SQL> connect  sys/oracle_4U@PDB2_2 as sysdba
Connected.
SQL> col username format a20
SQL> select USERNAME, COMMON, CON_ID from cdb_users
  2> order by username;

USERNAME             COM CON_ID
-------------------- --- ------
ANONYMOUS            YES      4
APPQOSSYS            YES      4
APPS                 NO       4
…
C##_USER             YES      4
…
SYSTEM               YES      4
…
XS$NULL              YES      4

39 rows selected.

SQL>
```

Notice that you view all common and local users of the current PDB.

```
SQL> select USERNAME, COMMON from dba_users order by username;

USERNAME             COM
-------------------- ---
ANONYMOUS            YES
APPQOSSYS            YES
APPS                 NO
…
C##_USER             YES
…
SYSTEM               YES
…
XS$NULL              YES

39 rows selected.

SQL>
```

Notice that you view the same list.

Practice 5-3: Managing Local and Common Roles

Overview

In this practice, you will manage roles created as common or local, and granted as common and/or local in CDB and PDBs.

Assumptions

C##_USER and LOCAL_USER_PDB2 are successfully created from the previous Practice 5-2 in cdb2 PDB2.

Tasks

1. Manage creation of roles in CDB and PDBs.
   a. List all predefined roles in CDB.

```
SQL> connect / as sysdba
Connected.
SQL> col role format a30
SQL> select ROLE, COMMON, CON_ID from cdb_roles order by role;

ROLE                           COM CON_ID
------------------------------ --- ------
ADM_PARALLEL_EXECUTE_TASK      YES      2
ADM_PARALLEL_EXECUTE_TASK      YES      1
ADM_PARALLEL_EXECUTE_TASK      YES      3
ADM_PARALLEL_EXECUTE_TASK      YES      4
ADM_PARALLEL_EXECUTE_TASK      YES      5
ADM_PARALLEL_EXECUTE_TASK      YES      6

…
DBA                            YES      3
DBA                            YES      1
DBA                            YES      4
DBA                            YES      2
DBA                            YES      5
DBA                            YES      6


…
PDB_DBA                        YES      3
PDB_DBA                        YES      4
PDB_DBA                        YES      6


…


495 rows selected.
```

```

SQL>
```

You can view all common and local roles of the root and PDBs.

b. List all predefined roles in root.

```
SQL> select ROLE, COMMON from dba_roles order by role;

ROLE                           COM
------------------------------ ---
ADM_PARALLEL_EXECUTE_TASK      YES
APEX_ADMINISTRATOR_ROLE        YES
AQ_ADMINISTRATOR_ROLE          YES
AQ_USER_ROLE                   YES
…
XS_CACHE_ADMIN                 YES
XS_NSATTR_ADMIN                YES
XS_RESOURCE                    YES
XS_SESSION_ADMIN               YES

82 rows selected.

SQL>
```

Notice that all roles of the root are common: there cannot be any local roles in the root.

c. Create a common `C##_ROLE` in root.

```
SQL> create role c##_role container=ALL;

Role created.

SQL>
```

d. Create a local `LOCAL_ROLE` in root.

```
SQL> create role local_role container=CURRENT;
create role local_role container=CURRENT
*
ERROR at line 1:
ORA-65049: creation of local user or role is not allowed in
CDB$ROOT

SQL>
```

You get an error message because no local role is authorized in the root.

e. List all predefined roles in PDB `PDB2`.

```
SQL> connect  system/oracle_4U@PDB2
Connected.
```

```
SQL> col role format a30
SQL> select ROLE, COMMON, CON_ID from cdb_roles;

ROLE                           COM CON_ID
------------------------------ --- ------
CONNECT                         YES      3
RESOURCE                        YES      3
DBA                             YES      3
AUDIT_ADMIN                     YES      3
…
C##_ROLE                       YES      3
DV_REALM_RESOURCE               YES      3
DV_REALM_OWNER                  YES      3
PDB_DBA                        YES      3

84 rows selected.

SQL>
```

You can view all common and local roles of the PDB only.

```
SQL> select ROLE,COMMON from dba_roles order by role;

ROLE                           COM
------------------------------ ---
CONNECT                         YES
RESOURCE                        YES
DBA                             YES
AUDIT_ADMIN                     YES
…
PDB_DBA                        YES
…
XDB_WEBSERVICES                 YES
XDB_WEBSERVICES_OVER_HTTP       YES
XDB_WEBSERVICES_WITH_PUBLIC     YES
XS_CACHE_ADMIN                  YES
XS_NSATTR_ADMIN                 YES
XS_RESOURCE                     YES
XS_SESSION_ADMIN                YES

84 rows selected.

SQL>
```

You view the same list.

f. Create a common role in PDB2.

```
SQL> create role c##_role_PDB2 container=ALL;
create role c##_role_PDB2 container=ALL
*
ERROR at line 1:
ORA-65050: Common DDLs only allowed in CDB$ROOT

SQL>
```

You get an error message because no common role can be created from a PDB.

g. Create a local role in `PDB2`.

```
SQL> create role local_role_PDB2 container=CURRENT;
Role created.

SQL> select ROLE,COMMON from dba_roles order by role;

ROLE                           COM
------------------------------ ---
ADM_PARALLEL_EXECUTE_TASK      YES
APEX_ADMINISTRATOR_ROLE        YES
AQ_ADMINISTRATOR_ROLE          YES
…
LOCAL_ROLE_PDB2                NO
…
XS_NSATTR_ADMIN                YES
XS_RESOURCE                    YES
XS_SESSION_ADMIN               YES

85 rows selected.

SQL>
```

2. Grant common or local roles as common or local.

a. Grant a common role to a common user from the root.

```
SQL> connect / as sysdba
Connected.

SQL> grant c##_role to c##_user;
Grant succeeded.

SQL> col grantee format A16
SQL> col GRANTED_ROLE format A16
SQL> select GRANTEE, GRANTED_ROLE, COMMON, CON_ID
  2  from cdb_role_privs where grantee='C##_USER';
```

```
GRANTEE          GRANTED_ROLE     COM CON_ID
---------------- ---------------- --- ------
C##_USER         C##_ROLE         NO       1

SQL>
```

Note that the common role is granted locally to the common user. The granted role is only applicable in the root.

```
SQL> connect c##_user/x
Connected.

SQL> select * from session_roles;

ROLE
------------------------------
C##_ROLE

SQL>
```

```
SQL> connect c##_user/x@PDB2
Connected.

SQL> select * from session_roles;
no rows selected

SQL>
```

b. Now grant the common role to a common user from the root as common, to be applicable in all containers.

```
SQL> connect / as sysdba
Connected.
SQL> grant c##_role to c##_user container=all;
Grant succeeded.

SQL> select GRANTEE, GRANTED_ROLE, COMMON, CON_ID
  2  from cdb_role_privs where grantee='C##_USER';

GRANTEE          GRANTED_ROLE     COM CON_ID
---------------- ---------------- --- ------
C##_USER         C##_ROLE         NO       1
C##_USER         C##_ROLE         YES      1
C##_USER         C##_ROLE         YES      3
C##_USER         C##_ROLE         YES      4
C##_USER         C##_ROLE         YES      5
```

```
C##_USER         C##_ROLE         YES      6

SQL>
```

```
SQL> connect c##_user/x
Connected.

SQL> select * from session_roles;

ROLE
------------------------------
C##_ROLE

SQL>
```

```
SQL> connect c##_user/x@PDB2
Connected.
SQL> select * from session_roles;

ROLE
------------------------------
C##_ROLE

SQL>
```

c. Revoke the common role from the common user so that the role cannot be used in any container.

```
SQL> connect / as sysdba
Connected.

SQL> revoke c##_role from c##_user container=all;
Revoke succeeded.

SQL> connect c##_user/x
Connected.

SQL> select * from session_roles;
ROLE
------------------------------
C##_ROLE

SQL>
SQL> connect c##_user/x@PDB2
Connected.
SQL> select * from session_roles;
no rows selected

SQL>
```

d. Grant a common role to a local user from the root.

```
SQL> connect / as sysdba
Connected.

SQL> grant c##_role to local_user_pdb2;
grant c##_role to local_user_pdb2
                     *
ERROR at line 1:
ORA-01917: user or role 'LOCAL_USER_PDB2' does not exist

SQL>
```

Note that the user is unknown in root. It is a local user in PDB2.

e. Grant a common role to a local user from PDB2.

```
SQL> connect system/oracle_4U@PDB2
Connected.
SQL> grant c##_role to local_user_PDB2;
Grant succeeded.

SQL> select GRANTEE, GRANTED_ROLE, COMMON, CON_ID
  2  from cdb_role_privs where grantee='LOCAL_USER_PDB2';

GRANTEE          GRANTED_ROLE     COM CON_ID
---------------- ---------------- --- ------
LOCAL_USER_PDB2  C##_ROLE         NO       3

SQL>
```

Note that the user is granted a common role locally (common column = NO) applicable only in the PDB PDB2.

f. Test the connection as the local user.

```
SQL> connect  local_user_pdb2/x@PDB2
Connected.
SQL> select * from session_roles;
ROLE
------------------------------
C##_ROLE
```

```
SQL>
```

g. Grant a common role to a local user from PDB2 applicable in all containers.

```
SQL> connect system/oracle_4U@PDB2
Connected.
SQL> grant c##_role to local_user_pdb2 container=all;
grant c##_role to local_user_pdb2 container=all
*
ERROR at line 1:
ORA-65030: one may not grant a Common Privilege to a Local User
or Role

SQL>
```

Notice that a common role cannot be granted globally from a PDB.

h. Grant a local role to a local user from PDB2.

```
SQL> grant local_role_pdb2 to local_user_pdb2;
Grant succeeded.
SQL> select GRANTEE, GRANTED_ROLE, COMMON, CON_ID
  2  from cdb_role_privs where grantee='LOCAL_USER_PDB2';

GRANTEE          GRANTED_ROLE     COM CON_ID
---------------- ---------------- --- ------
LOCAL_USER_PDB2  C##_ROLE         NO       3
LOCAL_USER_PDB2  LOCAL_ROLE_PDB2  NO       3

SQL>
```

i. Test the connection as the local user.

```
SQL> connect  local_user_pdb2/x@PDB2
Connected.
SQL> select * from session_roles;

ROLE
------------------------------
C##_ROLE
LOCAL_ROLE_PDB2

SQL>
```

Practice 5-4: Managing Local and Common Privileges

Overview

In this practice, you will manage privileges granted as common and/or local in the CDB and PDBs.

Assumptions

C##_USER and LOCAL_USER_PDB2 are successfully created from the previous Practice 5-2 in cdb2 PDB2.

Tasks

1. Check whether privileges are created as common or local.

```
SQL> connect / as sysdba
Connected.
SQL> desc sys.system_privilege_map
Name                                      Null?    Type
 ----------------------------------------- -------- -----------
 PRIVILEGE                                  NOT NULL NUMBER
 NAME                                       NOT NULL VARCHAR2(40)
 PROPERTY                                   NOT NULL NUMBER

SQL> desc sys.table_privilege_map
 Name                                       Null?    Type
 ----------------------------------------- -------- -----------
 PRIVILEGE                                  NOT NULL NUMBER
 NAME                                       NOT NULL VARCHAR2(40)

SQL>
```

Notice that there is no COMMON column. Privileges are not created as common or local, but they can be granted as common or local.

2. Check how the CREATE SESSION system privilege was granted to C##_USER and LOCAL_USER_PDB2 users.

```
SQL> connect system/oracle_4U
Connected.
SQL> col grantee format a18
SQL> col privilege format a14
SQL> select GRANTEE, PRIVILEGE, COMMON, CON_ID
  2  from cdb_sys_privs
  3  where grantee in ('C##_USER', 'LOCAL_USER_PDB2');

GRANTEE            PRIVILEGE      COM CON_ID
------------------ -------------- --- ------
C##_USER           CREATE SESSION YES      1
```

```
C##_USER           CREATE SESSION YES      3
LOCAL_USER_PDB2    CREATE SESSION NO       3
C##_USER           CREATE SESSION YES      4
C##_USER           CREATE SESSION YES      5
C##_USER           CREATE SESSION YES      6

6 rows selected.

SQL>
```

```
SQL> connect system/oracle_4U@PDB2
Connected.
SQL> select GRANTEE, PRIVILEGE, COMMON
  2  from dba_sys_privs
  3  where grantee in ('C##_USER', 'LOCAL_USER_PDB2');

GRANTEE            PRIVILEGE      COM
------------------ -------------- ---
LOCAL_USER_PDB2    CREATE SESSION NO
C##_USER           CREATE SESSION YES

SQL>
```

3. Grant the system privileges `CREATE TABLE` and `UNLIMITED TABLESPACE` to common user `C##_USER` to be applicable in any container. This will be a common privilege.

```
SQL> connect system/oracle_4U
Connected.
SQL> grant CREATE TABLE, UNLIMITED TABLESPACE to C##_USER
  2  CONTAINER=ALL;
Grant succeeded.

SQL> col grantee format a12
SQL> col privilege format a30

SQL> select GRANTEE, PRIVILEGE, COMMON, CON_ID
  2  from cdb_sys_privs
  3  where grantee = 'C##_USER';

GRANTEE      PRIVILEGE                      COM CON_ID
------------ ------------------------------ --- ------
C##_USER  CREATE TABLE                   YES      1
C##_USER  CREATE SESSION                 YES      1
C##_USER  UNLIMITED TABLESPACE           YES      1
```

```
C##_USER  CREATE TABLE                   YES      3
C##_USER  CREATE SESSION                 YES      3
C##_USER  UNLIMITED TABLESPACE           YES      3
C##_USER  CREATE TABLE                   YES      4
C##_USER  CREATE SESSION                 YES      4
C##_USER  UNLIMITED TABLESPACE           YES      4
C##_USER  CREATE TABLE                   YES      5
C##_USER  CREATE SESSION                 YES      5
C##_USER  UNLIMITED TABLESPACE           YES      5
C##_USER  CREATE TABLE                   YES      6
C##_USER  CREATE SESSION                 YES      6
C##_USER  UNLIMITED TABLESPACE           YES      6

15 rows selected.

SQL>
```

4. Grant the system privilege `CREATE SEQUENCE` to common user `C##_USER` to be applicable in root only. This will be a local privilege.

```
SQL> connect system/oracle_4U
Connected.
SQL> col grantee format a12
SQL> grant CREATE SEQUENCE to C##_USER CONTAINER=CURRENT;
Grant succeeded.

SQL> select GRANTEE, PRIVILEGE, COMMON, CON_ID
  2  from cdb_sys_privs
  3  where grantee = 'C##_USER';

GRANTEE      PRIVILEGE                      COM CON_ID
------------ ------------------------------ --- ------
C##_USER  CREATE SEQUENCE                NO       1
C##_USER  CREATE TABLE                   YES      1
C##_USER  CREATE SESSION                 YES      1
C##_USER  UNLIMITED TABLESPACE           YES      1
C##_USER  CREATE TABLE                   YES      3
C##_USER  CREATE SESSION                 YES      3
C##_USER  UNLIMITED TABLESPACE           YES      3
C##_USER  CREATE TABLE                   YES      4
C##_USER  CREATE SESSION                 YES      4
C##_USER  UNLIMITED TABLESPACE           YES      4
C##_USER  CREATE TABLE                   YES      5
C##_USER  CREATE SESSION                 YES      5
```

```
C##_USER  UNLIMITED TABLESPACE           YES      5
C##_USER  CREATE TABLE                   YES      6
C##_USER  CREATE SESSION                 YES      6
C##_USER  UNLIMITED TABLESPACE           YES      6

16 rows selected.

SQL>
```

5. Grant the system privilege `CREATE SYNONYM` to common user `C##_USER` to be applicable in `PDB2` only. This will be a local privilege.

```
SQL> connect system/oracle_4U@PDB2
Connected.

SQL> col grantee format a18
SQL> grant CREATE SYNONYM to C##_USER CONTAINER=CURRENT;
Grant succeeded.

SQL> select GRANTEE, PRIVILEGE, COMMON, CON_ID
  2   from cdb_sys_privs
  3   where grantee = 'C##_USER';

GRANTEE            PRIVILEGE                      COM CON_ID
------------------ ------------------------------ --- ------
C##_USER        CREATE SYNONYM                 NO        3
C##_USER        CREATE TABLE                   YES       3
C##_USER        CREATE SESSION                 YES       3
C##_USER        UNLIMITED TABLESPACE           YES       3

SQL>
```

6. Grant the system privilege `CREATE VIEW` to common user `C##_USER` to be applicable in root only, but connected in `PDB2`.

```
SQL> col grantee format a18
SQL> grant CREATE VIEW to C##_USER CONTAINER=ALL;
grant CREATE VIEW to C##_USER CONTAINER=ALL
*
ERROR at line 1:
ORA-65050: Common DDLs only allowed in CDB$ROOT

SQL>
```

Note that you cannot grant a common privilege from a PDB.

7. Grant the system privilege `CREATE ANY TABLE` to local user `LOCAL_USER_PDB2` to be applicable in any container.

```
SQL> connect system/oracle_4U
Connected.

SQL> col grantee format a18
SQL> grant CREATE ANY TABLE to LOCAL_USER_PDB2 CONTAINER=ALL;
grant CREATE ANY TABLE to LOCAL_USER_PDB2 CONTAINER=ALL
                          *
ERROR at line 1:
ORA-01917: user or role 'LOCAL_USER_PDB2' does not exist

SQL>
```

Notice that the user is unknown in root. It is a local user in PDB2.

8. Grant the system privilege CREATE ANY SEQUENCE to local user LOCAL_USER_PDB2 to be applicable in root only.

```
SQL> grant CREATE ANY SEQUENCE to LOCAL_USER_PDB2
  2  CONTAINER=CURRENT;
grant CREATE ANY SEQUENCE to LOCAL_USER_PDB2
                             *
ERROR at line 1:
ORA-01917: user or role 'LOCAL_USER_PDB2' does not exist

SQL>
```

Notice that the user is unknown in root. It is a local user in PDB2.

9. Grant the system privilege UNLIMITED TABLESPACE to local user LOCAL_USER_PDB2 to be applicable in PDB2 only. This will be a local privilege.

```
SQL> connect system/oracle_4U@PDB2
Connected.

SQL> col grantee format a18
SQL> grant UNLIMITED TABLESPACE to LOCAL_USER_PDB2;
Grant succeeded.

SQL> select GRANTEE, PRIVILEGE, COMMON, CON_ID
  2  from cdb_sys_privs
  3  where grantee = 'LOCAL_USER_PDB2';

GRANTEE            PRIVILEGE                      COM CON_ID
------------------ ------------------------------ --- ------
LOCAL_USER_PDB2    CREATE SESSION                 NO       3
LOCAL_USER_PDB2    UNLIMITED TABLESPACE           NO       3

SQL>
```

10. Grant the system privilege `DROP ANY VIEW` to local user `LOCAL_USER_PDB2` to be applicable in root only but connected in `PDB2`.

```
SQL> grant DROP ANY VIEW to LOCAL_USER_PDB2 CONTAINER=ALL;


grant DROP ANY VIEW to LOCAL_USER_PDB2 CONTAINER=ALL
*
ERROR at line 1:
ORA-65030: one may not grant a Common Privilege to a Local User
or Role


SQL> EXIT
$
```

Notice that you cannot grant a local privilege that will be applicable in another container.

Practices for Lesson 6: Backup, Recovery, Flashback CDB and PDBs

Chapter 6

Practices for Lesson 6

Practices Overview

In the next practices you will perform backup and recovery operations on the CDB and PDBs.

- RMAN `cdb2` backup
- RMAN whole and partial `pdb2` backup
- Recovery from SYSTEM `pdb2` data file loss
- Recovery from non-essential `pdb2` data file loss

Other optional scenarios are proposed for the attendees who still have time within the dedicated hour and want to run some more practices.

- SQL PDB hot backup
- SQL control file backup
- Recovery from all control files loss
- Recovery from redo log member loss
- Recovery from SYSTEM root data file loss
- Recovery from a non-essential root data file loss
- Point-in-time recovery on PDB tablespaces
- CDB flashback from DROP common user

Assumptions

`cdb2` is successfully created from previous Practice 3-1.

`pdb2_1` is successfully created from previous Practice 3-3.

`pdb2_1` is successfully renamed to `pdb2` from previous Practice 4-4.

If permanent and temporary tablespaces could not be created successfully, execute the following catchup script:

```
$ cd /home/oracle/recovery/catchup_05_02
$ ./cr_TABLESPACES.sh
$
```

Practice 6-1: Cold CDB Backup

Overview

In this practice, you will perform a CDB cold backup that you can use in case you lose all further backups or you cannot recover from a difficult situation.

But before performing this task, make sure your database is in `ARCHIVELOG` mode.

Tasks

1. Create the backup directory.

```
$ rm -Rf /home/oracle/Safe_Database_Files/cdb2
$ mkdir  /home/oracle/Safe_Database_Files
$ mkdir  /home/oracle/Safe_Database_Files/cdb2
$
```

2. Shut down the `cdb2` database before backing up all files.

```
$ . oraenv
ORACLE_SID = [cdb2] ? cdb2
The Oracle base remains unchanged with value /u01/app/oracle
$ sqlplus / AS SYSDBA

Connected to:
Oracle Database 12c Enterprise Edition Release 12.1.0.0.2 -
64bit Production
With the Partitioning, OLAP, Data Mining and Real Application
Testing options

SQL> select log_mode from v$database;

LOG_MODE
------------
NOARCHIVELOG

SQL> SHUTDOWN IMMEDIATE
Database closed.
Database dismounted.
ORACLE instance shut down.

SQL> STARTUP MOUNT
ORACLE instance started.

Total System Global Area  655556608 bytes
Fixed Size                  2276288 bytes
Variable Size             188744768 bytes
Database Buffers          461373440 bytes
```

```
Redo Buffers                  3162112 bytes
Database mounted.

SQL> ALTER DATABASE ARCHIVELOG;
Database altered.

SQL> ALTER DATABASE OPEN;
Database altered.

SQL> SHUTDOWN IMMEDIATE
Database closed.
Database dismounted.
ORACLE instance shut down.

SQL> EXIT
$
```

3. Copy the files to the backup directory. The message is an informative message only.

```
$ tar -czf /home/oracle/Safe_Database_Files/cdb2/db.tar.gz
/u01/app/oracle/oradata/cdb2
tar: Removing leading `/' from member names
$
```

4. Start up the `cdb2` database before performing backups with RMAN.

```
$ sqlplus / AS SYSDBA

Connected to an idle instance.

SQL> STARTUP
ORACLE instance started.

Total System Global Area  655556608 bytes
Fixed Size                  2276288 bytes
Variable Size             188744768 bytes
Database Buffers          461373440 bytes
Redo Buffers                3162112 bytes
Database mounted.
Database opened.
SQL> EXIT
$
```

Practice 6-2: RMAN Whole CDB Backup

Overview

In this practice, you will perform a whole CDB backup of `cdb2`.

Assumptions

The `PDB2` has been successfully created in `cdb2` after Practices 3-3 and 4-4.

Tasks

1. Run RMAN to connect to `cdb2` with a user with `SYSDBA` or `SYSBACKUP` privilege.

```
$ export NLS_DATE_FORMAT='DD-MM-YYYY HH:MI:SS'
$ rman target /

connected to target database: cdb2 (DBID=534508813)
RMAN>
```

2. As usual, backup all data files of the database (root and all PDBs), control files and SPFILE, and archive log files, after setting the **`db_recovery_file_dest_size`** to 18 GB.

```
RMAN> CONFIGURE DEFAULT DEVICE TYPE TO disk;
using target database control file instead of recovery catalog
new RMAN configuration parameters:
CONFIGURE DEFAULT DEVICE TYPE TO DISK;
new RMAN configuration parameters are successfully stored

RMAN> CONFIGURE CONTROLFILE AUTOBACKUP ON;
new RMAN configuration parameters:
CONFIGURE CONTROLFILE AUTOBACKUP ON;
new RMAN configuration parameters are successfully stored

RMAN> ALTER SYSTEM SET db_recovery_file_dest_size=18G
SCOPE=both;

Statement processed

RMAN> BACKUP DATABASE PLUS ARCHIVELOG;
Starting backup at 02-07-2012 02:53:38
current log archived
allocated channel: ORA_DISK_1
channel ORA_DISK_1: SID=46 device type=DISK
channel ORA_DISK_1: starting archived log backup set
channel ORA_DISK_1: specifying archived log(s) in backup set
input archived log thread=1 sequence=10 RECID=1 STAMP=787589619
channel ORA_DISK_1: starting piece 1 at 02-07-2012 02:53:40
channel ORA_DISK_1: finished piece 1 at 02-07-2012 02:53:41
piece
handle=/u01/app/oracle/fast_recovery_area/cdb2/backupset/2012_07
_02/o1_mf_annnn_TAG20120702T145339_7z3fhn5p_.bkp
tag=TAG20120702T145339 comment=NONE
channel ORA_DISK_1: backup set complete, elapsed time: 00:00:01
Finished backup at 02-07-2012 02:53:41

Starting backup at 02-07-2012 02:53:41
using channel ORA_DISK_1
channel ORA_DISK_1: starting full datafile backup set
channel ORA_DISK_1: specifying datafile(s) in backup set
input datafile file number=00001
name=/u01/app/oracle/oradata/cdb2/system01.dbf
input datafile file number=00003
name=/u01/app/oracle/oradata/cdb2/sysaux01.dbf
input datafile file number=00007
name=/u01/app/oracle/oradata/cdb2/pdbseed/sysaux01.dbf
input datafile file number=00009
name=/u01/app/oracle/oradata/cdb2/pdb2_1/sysaux01.dbf
input datafile file number=00005
name=/u01/app/oracle/oradata/cdb2/pdbseed/system01.dbf
input datafile file number=00008
name=/u01/app/oracle/oradata/cdb2/pdb2_1/system01.dbf
input datafile file number=00004
name=/u01/app/oracle/oradata/cdb2/undotbs01.dbf
input datafile file number=00006
name=/u01/app/oracle/oradata/cdb2/users01.dbf
input datafile file number=00010
name=/u01/app/oracle/oradata/cdb2/pdb2_1/pdb2_1_users01.dbf
channel ORA_DISK_1: starting piece 1 at 02-07-2012 02:53:41

Starting backup at 02-07-2012 02:54:26
current log archived
using channel ORA_DISK_1
channel ORA_DISK_1: starting archived log backup set
channel ORA_DISK_1: specifying archived log(s) in backup set
input archived log thread=1 sequence=11 RECID=2 STAMP=787589666
channel ORA_DISK_1: starting piece 1 at 02-07-2012 02:54:26
channel ORA_DISK_1: finished piece 1 at 02-07-2012 02:54:27
piece
handle=/u01/app/oracle/fast_recovery_area/cdb2/backupset/2012_07
_02/o1_mf_annnn_TAG20120702T145426_7z3fk2qy_.bkp
tag=TAG20120702T145426 comment=NONE
channel ORA_DISK_1: backup set complete, elapsed time: 00:00:01
```

```
Finished backup at 02-07-2012 02:54:27


Starting Control File and SPFILE Autobackup at 02-07-2012
02:54:27
piece
handle=/u01/app/oracle/fast_recovery_area/cdb2/autobackup/2012_0
7_02/o1_mf_s_787589667_7z3fk46t_.bkp comment=NONE
Finished Control File and SPFILE Autobackup at 02-07-2012
02:54:28


RMAN>
```

Practice 6-3: RMAN CDB / PDB Backup

Overview

In this practice, you will perform a whole and a partial PDB backup of PDB2.

Assumptions

The PDB2 has been successfully created in cdb2 after Practices 3-1, 3-3, and 4-4.

Tasks

1. Perform a whole PDB backup.
   a. A new RMAN command allows you to back up all data files of the pluggable database.

```
RMAN> BACKUP PLUGGABLE DATABASE pdb2;

Starting backup at 13-09-2012 12:09:24
using channel ORA_DISK_1
channel ORA_DISK_1: starting full datafile backup set
channel ORA_DISK_1: specifying datafile(s) in backup set
input datafile file number=00009
name=/u01/app/oracle/oradata/cdb2/pdb2_1/sysaux01.dbf
input datafile file number=00008
name=/u01/app/oracle/oradata/cdb2/pdb2_1/system01.dbf
input datafile file number=00027
name=/u01/app/oracle/oradata/cdb2/pdb2_1/ldata_01.dbf
channel ORA_DISK_1: starting piece 1 at 13-09-2012 12:09:25
channel ORA_DISK_1: finished piece 1 at 13-09-2012 12:11:11
piece
handle=/u01/app/oracle/fast_recovery_area/CDB2/backupset/2012_09
_13/o1_mf_nnndf_TAG20120913T000925_85291pr4_.bkp
tag=TAG20120913T000925 comment=NONE
channel ORA_DISK_1: backup set complete, elapsed time: 00:01:46
Finished backup at 13-09-2012 12:11:11

Starting Control File and SPFILE Autobackup at 13-09-2012
12:11:11
piece
handle=/u01/app/oracle/fast_recovery_area/CDB2/autobackup/2012_0
9_13/o1_mf_s_793843873_852957tb_.bkp comment=NONE
Finished Control File and SPFILE Autobackup at 13-09-2012
12:11:27

RMAN>
```

2. Perform a partial PDB backup of the tablespace **ldata**.

   a. A new RMAN command allows you to back up some data files of the pluggable database.

```
RMAN> BACKUP TABLESPACE pdb2:ldata;

Starting backup at 13-09-2012 12:11:38
using channel ORA_DISK_1
channel ORA_DISK_1: starting full datafile backup set
channel ORA_DISK_1: specifying datafile(s) in backup set
input datafile file number=00027
name=/u01/app/oracle/oradata/cdb2/pdb2_1/ldata_01.dbf
channel ORA_DISK_1: starting piece 1 at 13-09-2012 12:11:39
channel ORA_DISK_1: finished piece 1 at 13-09-2012 12:11:42
piece
handle=/u01/app/oracle/fast_recovery_area/CDB2/backupset/2012_09
_13/o1_mf_nnndf_TAG20120913T001139_85295w9g_.bkp
tag=TAG20120913T001139 comment=NONE
channel ORA_DISK_1: backup set complete, elapsed time: 00:00:03
Finished backup at 13-09-2012 12:11:42

Starting Control File and SPFILE Autobackup at 13-09-2012
12:11:43
piece
handle=/u01/app/oracle/fast_recovery_area/CDB2/autobackup/2012_0
9_13/o1_mf_s_793843903_852962yc_.bkp comment=NONE
Finished Control File and SPFILE Autobackup at 13-09-2012
12:11:50

RMAN> EXIT
$
```

Practice 6-4: RMAN Recovery from SYSTEM PDB Data File Loss

Overview

In this practice, you will recover the PDB from an essential data file loss.

Tasks

1. Remove the `SYSTEM` data file of `PDB2`.

```
$ sqlplus system/oracle_4U@PDB2

Connected to:
Oracle Database 12c Enterprise Edition Release 12.1.0.0.2 -
64bit Production
With the Partitioning, OLAP, Data Mining and Real Application
Testing options

SQL> select file_name from DBA_DATA_FILES
     WHERE TABLESPACE_NAME='SYSTEM';

FILE_NAME
-------------------------------------------------------------
/u01/app/oracle/oradata/cdb2/pdb2_1/system01.dbf

SQL> exit
$
```

```
$ rm /u01/app/oracle/oradata/cdb2/pdb2_1/system01.dbf
```

2. Run RMAN to connect to `cdb2` with a user with `SYSDBA` or `SYSBACKUP` privilege.

```
$ rman target /

connected to target database: cdb2 (DBID=534631279)
RMAN>
```

3. Proceed with the traditional procedure to restore the missing data file and recover the CDB as it were a non-CDB.

```
RMAN> SHUTDOWN ABORT;

using target database control file instead of recovery catalog
Oracle instance shut down

RMAN> STARTUP MOUNT;

connected to target database (not started)
Oracle instance started
```

```
database mounted

Total System Global Area     655556608 bytes

Fixed Size                     2276288 bytes
Variable Size                260047936 bytes
Database Buffers             390070272 bytes
Redo Buffers                   3162112 bytes

RMAN> RESTORE TABLESPACE pdb2:SYSTEM;

Starting restore at 13-SEP-12
allocated channel: ORA_DISK_1
channel ORA_DISK_1: SID=7 device type=DISK

channel ORA_DISK_1: starting datafile backup set restore
channel ORA_DISK_1: specifying datafile(s) to restore from
backup set
channel ORA_DISK_1: restoring datafile 00008 to
/u01/app/oracle/oradata/cdb2/pdb2_1/system01.dbf
channel ORA_DISK_1: reading from backup piece
/u01/app/oracle/fast_recovery_area/CDB2/backupset/2012_09_13/o1_
mf_nnndf_TAG20120913T000925_85291pr4_.bkp
channel ORA_DISK_1: piece
handle=/u01/app/oracle/fast_recovery_area/CDB2/backupset/2012_09
_13/o1_mf_nnndf_TAG20120913T000925_85291pr4_.bkp
tag=TAG20120913T000925
channel ORA_DISK_1: restored backup piece 1
channel ORA_DISK_1: restore complete, elapsed time: 00:00:15
Finished restore at 13-SEP-12

RMAN> RECOVER TABLESPACE pdb2:SYSTEM;

Starting recover at 13-SEP-12
using channel ORA_DISK_1

starting media recovery
media recovery complete, elapsed time: 00:00:01

Finished recover at 13-SEP-12

RMAN> ALTER DATABASE OPEN;

Statement processed
```

```
RMAN> ALTER PLUGGABLE DATABASE ALL OPEN;
Statement processed

RMAN>
```

Or you could have used the new syntax to restore and recover a whole PDB, as follows:

```
RMAN> SHUTDOWN ABORT;

using target database control file instead of recovery catalog
Oracle instance shut down

RMAN> startup mount;

Oracle instance started
database mounted

Total System Global Area     634679296 bytes

Fixed Size                     2263616 bytes
Variable Size                213910976 bytes
Database Buffers             415236096 bytes
Redo Buffers                   3268608 bytes

RMAN> RESTORE pluggable database pdb2;

Starting restore at 13-SEP-12
allocated channel: ORA_DISK_1
channel ORA_DISK_1: SID=21 device type=DISK

channel ORA_DISK_1: starting datafile backup set restore
channel ORA_DISK_1: specifying datafile(s) to restore from
backup set
channel ORA_DISK_1: restoring datafile 00008 to
/u01/app/oracle/oradata/cdb2/pdb2_1/system01.dbf
channel ORA_DISK_1: restoring datafile 00009 to
/u01/app/oracle/oradata/cdb2/pdb2_1/sysaux01.dbf
channel ORA_DISK_1: reading from backup piece
/u01/app/oracle/fast_recovery_area/CDB2/backupset/2012_09_13/o1_
mf_nnndf_TAG20120913T000925_85291pr4_.bkp
channel ORA_DISK_1: piece
handle=/u01/app/oracle/fast_recovery_area/CDB2/backupset/2012_09
_13/o1_mf_nnndf_TAG20120913T000925_85291pr4_.bkp
tag=TAG20120913T000925
channel ORA_DISK_1: restored backup piece 1
channel ORA_DISK_1: restore complete, elapsed time: 00:00:45
channel ORA_DISK_1: starting datafile backup set restore
channel ORA_DISK_1: specifying datafile(s) to restore from
backup set
channel ORA_DISK_1: restoring datafile 00027 to
/u01/app/oracle/oradata/cdb2/pdb2_1/ldata_01.dbf
channel ORA_DISK_1: reading from backup piece
/u01/app/oracle/fast_recovery_area/CDB2/backupset/2012_09_13/o1_
mf_nnndf_TAG20120913T001139_85295w9g_.bkp
channel ORA_DISK_1: piece
handle=/u01/app/oracle/fast_recovery_area/CDB2/backupset/2012_09
_13/o1_mf_nnndf_TAG20120913T001139_85295w9g_.bkp
tag=TAG20120913T001139
channel ORA_DISK_1: restored backup piece 1
channel ORA_DISK_1: restore complete, elapsed time: 00:00:01
Finished restore at 13-SEP-12

RMAN> RECOVER pluggable database pdb2;

Starting recover at 13-SEP-12
using channel ORA_DISK_1

starting media recovery
media recovery complete, elapsed time: 00:00:02

Finished recover at 13-SEP-12

RMAN> ALTER DATABASE OPEN;

Statement processed

RMAN> select name, open_mode from v$pdbs;

NAME                           OPEN_MODE
------------------------------ ----------
PDB$SEED                       READ ONLY
PDB2                           READ WRITE
PDB2_2                         READ WRITE
PDB_ORCL2                      READ WRITE
PDB1_1                         READ WRITE

RMAN> exit
$
```

Practice 6-5: RMAN Recovery from Non-Essential PDB Data File Loss

Overview

In this practice, you will recover from a non-essential PDB data file loss.

Assumptions

The LDATA tablespace has been successfully created in Practice 5-1.

Tasks

1. Remove a data file of the LDATA tablespace of PDB2.

```
$ sqlplus system/oracle_4U@PDB2

Connected to:
Oracle Database 12c Enterprise Edition Release 12.1.0.0.2 -
64bit Production
With the Partitioning, OLAP, Data Mining and Real Application
Testing options

SQL> select file_name from dba_data_files
     where tablespace_name='LDATA';

FILE_NAME
-------------------------------------------------------------
/u01/app/oracle/oradata/cdb2/pdb2_1/ldata_01.dbf

SQL> exit
$
```

```
$ rm /u01/app/oracle/oradata/cdb2/pdb2_1/ldata_01.dbf
```

2. Proceed with the traditional procedure to restore the missing data file and recover the tablespace as if it were a non-CDB.

   a. Put the tablespace in OFFLINE mode.

```
$ sqlplus system/oracle_4U@PDB2

Connected to:
Oracle Database 12c Enterprise Edition Release 12.1.0.0.2 -
64bit Production
With the Partitioning, OLAP, Data Mining and Real Application
Testing options

SQL> ALTER TABLESPACE ldata OFFLINE IMMEDIATE;

Tablespace altered.

SQL> exit
$
```

   b. Run RMAN to connect to `cdb2` with a user with `SYSDBA` or `SYSBACKUP` privilege.

```
$ rman target /

connected to target database: CDB2 (DBID=534631279)
RMAN>
```

   c. Restore and recover the tablespace.

```
RMAN> RESTORE TABLESPACE pdb2:LDATA;

Starting restore at 13-SEP-12
using target database control file instead of recovery catalog
allocated channel: ORA_DISK_1
channel ORA_DISK_1: SID=20 device type=DISK

channel ORA_DISK_1: starting datafile backup set restore
channel ORA_DISK_1: specifying datafile(s) to restore from
backup set
channel ORA_DISK_1: restoring datafile 00027 to
/u01/app/oracle/oradata/cdb2/pdb2_1/ldata_01.dbf
channel ORA_DISK_1: reading from backup piece
/u01/app/oracle/fast_recovery_area/CDB2/backupset/2012_09_13/o1_
mf_nnndf_TAG20120913T001139_85295w9g_.bkp
channel ORA_DISK_1: piece
handle=/u01/app/oracle/fast_recovery_area/CDB2/backupset/2012_09
_13/o1_mf_nnndf_TAG20120913T001139_85295w9g_.bkp
tag=TAG20120913T001139
channel ORA_DISK_1: restored backup piece 1
channel ORA_DISK_1: restore complete, elapsed time: 00:00:01
Finished restore at 13-SEP-12

RMAN> RECOVER TABLESPACE pdb2:LDATA;

Starting recover at 13-SEP-12
using channel ORA_DISK_1

starting media recovery
media recovery complete, elapsed time: 00:00:01

Finished recover at 13-SEP-12

RMAN> exit
$
```

d. Put the tablespace back ONLINE.

```
$ sqlplus system/oracle_4U@PDB2

Connected to:
Oracle Database 12c Enterprise Edition Release 12.1.0.0.2 -
64bit Production
With the Partitioning, OLAP, Data Mining and Real Application
Testing options

SQL> ALTER TABLESPACE ldata ONLINE;
Tablespace altered.

SQL> exit
$
```

3. If you do not intend to perform any further practices on CDBs, shut down the `cdb1` and `cdb2` instances if not dropped in Practice 3-6.

```
$ . oraenv
ORACLE_SID = [orcl] ? cdb1
The Oracle base remains unchanged with value /u01/app/oracle
$ sqlplus / as sysdba

Connected to:
Oracle Database 12c Enterprise Edition Release 12.1.0.0.2 -
64bit Production
With the Partitioning, OLAP, Data Mining and Real Application
Testing options

SQL> shutdown immediate;
```

```
Database closed.
Database dismounted.
ORACLE instance shut down.
SQL> exit
Disconnected from Oracle Database 12c Enterprise Edition Release
12.1.0.0.2 - 64bit Production
With the Partitioning, OLAP, Data Mining and Real Application
Testing options
$
```

```
$ . oraenv
ORACLE_SID = [orcl] ? cdb2
The Oracle base remains unchanged with value /u01/app/oracle $
sqlplus / as sysdba


SQL*Plus: Release 12.1.0.0.2 Production on Thu Jul 5 09:15:29
2012


Copyright (c) 1982, 2012, Oracle.  All rights reserved.



Connected to:
Oracle Database 12c Enterprise Edition Release 12.1.0.0.2 -
64bit Production
With the Partitioning, OLAP, Data Mining and Real Application
Testing options

SQL> shutdown immediate;
Database closed.
Database dismounted.
ORACLE instance shut down.
SQL> exit
$
```

Practice 6-6: SQL PDB Hot Backup (optional)

Overview

In this practice, you will perform a hot backup of PDB2 in cdb2.

Assumptions

The PDB2 has been successfully created in cdb2 after Practices 3-1, 3-3, and 4-4.

Tasks

1. List all data files belonging to PDB2 to be backed up.

```
$ sqlplus system/oracle_4U@PDB2

Connected to:
Oracle Database 12c Enterprise Edition Release 12.1.0.0.2 -
64bit Production
With the Partitioning, OLAP, Data Mining and Real Application
Testing options

SQL> select file_name from dba_data_files;

FILE_NAME
--------------------------------------------------------------
/u01/app/oracle/oradata/cdb2/pdb2_1/system01.dbf
/u01/app/oracle/oradata/cdb2/pdb2_1/sysaux01.dbf
/u01/app/oracle/oradata/cdb2/pdb2_1/ldata_01.dbf

SQL>
```

2. Set the PDB in hot backup.

```
SQL> ALTER PLUGGABLE DATABASE pdb2 BEGIN BACKUP;

Pluggable database altered.

SQL> exit
$
```

3. Copy the data files of the pluggable database to a backup directory.

```
$ mkdir /home/oracle/backup
$ cp /u01/app/oracle/oradata/cdb2/pdb2_1/*.dbf
/home/oracle/backup
$
```

4. Deactivate the backup mode.

```
$ sqlplus system/oracle_4U@PDB2

Connected to:
Oracle Database 12c Enterprise Edition Release 12.1.0.0.2 -
64bit Production
With the Partitioning, OLAP, Data Mining and Real Application
Testing options

SQL> ALTER PLUGGABLE DATABASE pdb2 END BACKUP;

Pluggable database altered.

SQL>
```

Practice 6-7: SQL Control File Backup (optional)

Overview

In this practice, you will use the traditional SQL command to back up the `cdb2` control file.

Tasks

1. Connect to the `cdb2` root.

```
SQL> CONNECT / as sysdba

Connected to:
Oracle Database 12c Enterprise Edition Release 12.1.0.0.2 -
64bit Production
With the Partitioning, OLAP, Data Mining and Real Application
Testing options

SQL>
```

2. Run the `ALTER DATABASE` command to back up the control file to a script.

```
SQL> alter database backup controlfile to trace;

Database altered.

SQL>
```

Practice 6-8: RMAN Recovery from Control File Loss (optional)

Overview

In this practice, you will recover the CDB from the control file loss.

Assumptions

Practice 6-2 successfully completed the whole CDB backup of `cdb2`.

Tasks

1. Remove the control files of the CDB.

```
SQL> CONNECT / AS SYSDBA

Connected to:
Oracle Database 12c Enterprise Edition Release 12.1.0.0.2 -
64bit Production
With the Partitioning, OLAP, Data Mining and Real Application
Testing options

SQL> select name from v$controlfile;

NAME
--------------------------------------------------------------
/u01/app/oracle/oradata/cdb2/control01.ctl
/u01/app/oracle/fast_recovery_area/cdb2/control02.ctl

SQL> !rm /u01/app/oracle/oradata/cdb2/control01.ctl
SQL>
```

2. Shut down / abort the instance `cdb2`.

```
SQL> shutdown abort
ORACLE instance shut down.
SQL> exit
$
```

3. Proceed with the traditional procedure to restore the control files and recover the CDB as if it were a non-CDB database.

```
$ rman target /

connected to target database (not started)
RMAN> startup nomount;
Oracle instance started

Total System Global Area     634679296 bytes

Fixed Size                     2263616 bytes
```

```
Variable Size                226493888 bytes
Database Buffers             402653184 bytes
Redo Buffers                   3268608 bytes

RMAN> RESTORE CONTROLFILE FROM AUTOBACKUP;

Starting restore at 07-09-2012 08:40:42
using target database control file instead of recovery catalog
allocated channel: ORA_DISK_1
channel ORA_DISK_1: SID=21 device type=DISK

recovery area destination: /u01/app/oracle/fast_recovery_area
database name (or database unique name) used for search: CDB2
channel ORA_DISK_1: AUTOBACKUP
/u01/app/oracle/fast_recovery_area/CDB2/autobackup/2012_09_07/o1
_mf_s_793355103_84mcv293_.bkp found in the recovery area
AUTOBACKUP search with format "%F" not attempted because DBID
was not set
channel ORA_DISK_1: restoring control file from AUTOBACKUP
/u01/app/oracle/fast_recovery_area/CDB2/autobackup/2012_09_07/o1
_mf_s_793355103_84mcv293_.bkp
channel ORA_DISK_1: control file restore from AUTOBACKUP
complete
output file name=/u01/app/oracle/oradata/cdb2/control01.ctl
output file
name=/u01/app/oracle/fast_recovery_area/cdb2/control02.ctl
Finished restore at 07-09-2012 08:41:18

RMAN> ALTER DATABASE MOUNT;

Statement processed
released channel: ORA_DISK_1

RMAN> RECOVER DATABASE;

Starting recover at 07-09-2012 08:41:50
Starting implicit crosscheck backup at 07-09-2012 08:41:50
allocated channel: ORA_DISK_1
channel ORA_DISK_1: SID=240 device type=DISK
Crosschecked 6 objects
Finished implicit crosscheck backup at 07-09-2012 08:41:51

Starting implicit crosscheck copy at 07-09-2012 08:41:51
using channel ORA_DISK_1
```

```
Finished implicit crosscheck copy at 07-09-2012 08:41:51

searching for all files in the recovery area
cataloging files...
cataloging done

List of Cataloged Files
=======================
File Name:
/u01/app/oracle/fast_recovery_area/CDB2/autobackup/2012_09_07/o1
_mf_s_793355103_84mcv293_.bkp
File Name:
/u01/app/oracle/fast_recovery_area/CDB2/archivelog/2012_09_07/o1
_mf_1_57_84md0zx0_.arc

using channel ORA_DISK_1

starting media recovery

archived log for thread 1 with sequence 57 is already on disk as
file
/u01/app/oracle/fast_recovery_area/CDB2/archivelog/2012_09_07/o1
_mf_1_57_84md0zx0_.arc
archived log for thread 1 with sequence 58 is already on disk as
file /u01/app/oracle/oradata/cdb2/redo01.log
archived log file
name=/u01/app/oracle/fast_recovery_area/CDB2/archivelog/2012_09_
07/o1_mf_1_57_84md0zx0_.arc thread=1 sequence=57
archived log file name=/u01/app/oracle/oradata/cdb2/redo01.log
thread=1 sequence=58
media recovery complete, elapsed time: 00:00:12
Finished recover at 07-09-2012 08:42:08

RMAN> ALTER DATABASE OPEN RESETLOGS;

Statement processed

RMAN> select name, open_mode from v$pdbs;

NAME                           OPEN_MODE
------------------------------ ----------
PDB$SEED                       READ ONLY
PDB2                           READ WRITE
PDB2_2                         READ WRITE
```

```
PDB_ORCL2                      READ WRITE
PDB1_1                         READ WRITE

RMAN>
```

4. Back up the whole cdb2.

   a. Use the BACKUP command.

```
RMAN> BACKUP DATABASE PLUS ARCHIVELOG;
…
RMAN> exit
$
```

   b. If you encounter some space issues, like the following, reclaim some space and increase the fast recovery area destination size:

```
RMAN-00571:
===========================================================
RMAN-00569: =============== ERROR MESSAGE STACK FOLLOWS
===============
RMAN-00571:
===========================================================
RMAN-03002: failure of backup plus archivelog command at
10/09/2012 03:46:44
ORA-19809: limit exceeded for recovery files
ORA-19804: cannot reclaim 67108864 bytes disk space from
10737418240 limit
```

      1) Reclaim some space deleting obsolete backups:

```
RMAN> delete obsolete;
RMAN retention policy will be applied to the command
RMAN retention policy is set to redundancy 1
using channel ORA_DISK_1
Deleting the following obsolete backups and copies:
Type                 Key    Completion Time    Filename/Handle
-------------------- ------ ------------------ -----------------
-
Backup Set           44     13-SEP-12
  Backup Piece       44     13-SEP-12
/u01/app/oracle/fast_recovery_area/CDB2/backupset/2012_09_13/o1_
mf_nnndf_TAG20120913T000151_8528mvgr_.bkp
Backup Set           45     13-SEP-12
  Backup Piece       45     13-SEP-12
/u01/app/oracle/fast_recovery_area/CDB2/backupset/2012_09_13/o1_
mf_annnn_TAG20120913T000620_8528vwkw_.bkp
Backup Set           47     13-SEP-12
  Backup Piece       47     13-SEP-12
/u01/app/oracle/fast_recovery_area/CDB2/backupset/2012_09_13/o1_
mf_nnndf_TAG20120913T000925_85291pr4_.bkp
Backup Set           49     13-SEP-12
```

```
  Backup Piece       49     13-SEP-12
/u01/app/oracle/fast_recovery_area/CDB2/backupset/2012_09_13/o1_
mf_nnndf_TAG20120913T001139_85295w9g_.bkp
Backup Set           51     13-SEP-12
  Backup Piece       51     13-SEP-12
/u01/app/oracle/fast_recovery_area/CDB2/autobackup/2012_09_13/o1
_mf_s_793845475_852bq621_.bkp


Do you really want to delete the above objects (enter YES or
NO)? Do you really want to delete the above objects (enter YES
or NO)? YES
deleted backup piece
backup piece
handle=/u01/app/oracle/fast_recovery_area/CDB2/backupset/2012_09
_13/o1_mf_nnndf_TAG20120913T000151_8528mvgr_.bkp RECID=44
STAMP=793843322
deleted backup piece
backup piece
handle=/u01/app/oracle/fast_recovery_area/CDB2/backupset/2012_09
_13/o1_mf_annnn_TAG20120913T000620_8528vwkw_.bkp RECID=45
STAMP=793843580
deleted backup piece
backup piece
handle=/u01/app/oracle/fast_recovery_area/CDB2/backupset/2012_09
_13/o1_mf_nnndf_TAG20120913T000925_85291pr4_.bkp RECID=47
STAMP=793843766
deleted backup piece
backup piece
handle=/u01/app/oracle/fast_recovery_area/CDB2/backupset/2012_09
_13/o1_mf_nnndf_TAG20120913T001139_85295w9g_.bkp RECID=49
STAMP=793843899
deleted backup piece
backup piece
handle=/u01/app/oracle/fast_recovery_area/CDB2/autobackup/2012_0
9_13/o1_mf_s_793845475_852bq621_.bkp RECID=51 STAMP=793845477
Deleted 5 objects
RMAN>
```

2) Increase the fast recovery area destination size to 20G.

```
RMAN> ALTER SYSTEM SET db_recovery_file_dest_size=20G
SCOPE=both;


using target database control file instead of recovery catalog
Statement processed


RMAN> EXIT
$
```

Practice 6-9: RMAN Recovery from Redo Log File Member Loss (optional)

Overview

In this practice, you recover the `cdb2` from a redo log file member loss.

Tasks

1. Multiplex the redo log files if not already done.

```
$ sqlplus system/oracle_4U

Connected to:
Oracle Database 12c Enterprise Edition Release 12.1.0.0.2 -
64bit Production
With the Partitioning, OLAP, Data Mining and Real Application
Testing options

SQL> select member from v$logfile;

MEMBER
-------------------------------------------------------------
/u01/app/oracle/oradata/cdb2/redo03.log
/u01/app/oracle/oradata/cdb2/redo02.log
/u01/app/oracle/oradata/cdb2/redo01.log

SQL>
```

```
SQL> ALTER DATABASE ADD LOGFILE MEMBER
'/u01/app/oracle/oradata/cdb2/redo01_2.log'
TO GROUP 1;

Database altered.

SQL> ALTER DATABASE ADD LOGFILE MEMBER
'/u01/app/oracle/oradata/cdb2/redo02_2.log'
TO GROUP 2;

Database altered.

SQL> ALTER DATABASE ADD LOGFILE MEMBER
'/u01/app/oracle/oradata/cdb2/redo03_2.log'
TO GROUP 3;

Database altered.
```

```

SQL>
```

```
SQL> alter system switch logfile;

System altered.

SQL> alter system switch logfile;

System altered.

SQL> alter system switch logfile;

System altered.

SQL> alter system switch logfile;

System altered.

SQL> exit
$
```

2. Remove a redo log file member of the `cdb2`.

```
$ rm /u01/app/oracle/oradata/cdb2/redo01.log
$
```

3. Proceed with the traditional procedure to regenerate the redo log file member. If you encounter any ORA-01609 error, execute the `alter system switch logfile` command until the redo log file is in `INACTIVE` status (view the `STATUS` in `V$LOG` view).

```
$ sqlplus system/oracle_4U

Connected to:
Oracle Database 12c Enterprise Edition Release 12.1.0.0.2 -
64bit Production
With the Partitioning, OLAP, Data Mining and Real Application
Testing options

SQL> ALTER DATABASE DROP LOGFILE MEMBER
        '/u01/app/oracle/oradata/cdb2/redo01.log';

Database altered.

SQL> ALTER DATABASE ADD LOGFILE MEMBER
       '/u01/app/oracle/oradata/cdb2/redo01.log'
```

```
     TO GROUP 1;

Database altered.

SQL> exit
$
```

Practice 6-10: RMAN Recovery from SYSTEM Root Data File Loss (optional)

Overview

In this practice, you will recover from a root data file loss, particularly the SYSTEM data file.

Tasks

1. Remove the SYSTEM data file from the root SYSTEM tablespace.

```
$ sqlplus system/oracle_4U

Connected to:
Oracle Database 12c Enterprise Edition Release 12.1.0.0.2 -
64bit Production
With the Partitioning, OLAP, Data Mining and Real Application
Testing options

SQL> select file_name from DBA_DATA_FILES
     WHERE TABLESPACE_NAME='SYSTEM';

FILE_NAME
-----------------------------------------------------------
/u01/app/oracle/oradata/cdb2/system01.dbf

SQL> exit
$
```

```
$ rm /u01/app/oracle/oradata/cdb2/system01.dbf
```

2. Run RMAN to connect to `cdb2` with a user with `SYSDBA` or `SYSBACKUP` privilege.

```
$ rman target /
RMAN-00571: ===========================================================
RMAN-00569: =============== ERROR MESSAGE STACK FOLLOWS =====
RMAN-00571: ===========================================================
RMAN-00554: initialization of internal recovery manager package
failed
RMAN-04005: error from target database:
ORA-00604: error occurred at recursive SQL level 1
ORA-01116: error in opening database file 1
ORA-01110: data file 1:
'/u01/app/oracle/oradata/cdb2/system01.dbf'
ORA-27041: unable to open file
Linux-x86_64 Error: 2: No such file or directory
Additional information: 3
$
```

3. Proceed with the traditional procedure to restore the missing data file and recover the CDB as it were a non-CDB.

```
$ sqlplus / as sysdba

Connected to:
Oracle Database 12c Enterprise Edition Release 12.1.0.1.0 -
64bit Production
With the Partitioning, OLAP, Advanced Analytics and Real
Application Testing options
```

```

SQL> SHUTDOWN ABORT
ORACLE instance shut down.
SQL> STARTUP MOUNT
Oracle instance started
Total System Global Area 5010685952 bytes
Fixed Size                  2298064 bytes
Variable Size            1040191280 bytes
Database Buffers         3959422976 bytes
Redo Buffers                8773632 bytes
Database mounted.
SQL> exit
$
```

```
$ rman target /
connected to target database: CDB2 (DBID=562519177, not open)
RMAN> RESTORE TABLESPACE SYSTEM;
```

```
Starting restore at 07-09-2012 09:16:02
```

```
allocated channel: ORA_DISK_1
channel ORA_DISK_1: SID=240 device type=DISK

channel ORA_DISK_1: starting datafile backup set restore
channel ORA_DISK_1: specifying datafile(s) to restore from
backup set
channel ORA_DISK_1: restoring datafile 00001 to
/u01/app/oracle/oradata/cdb2/system01.dbf
channel ORA_DISK_1: reading from backup piece
/u01/app/oracle/fast_recovery_area/CDB2/backupset/2012_09_07/o1_
mf_nnndf_TAG20120907T085859_84mftokq_.bkp
channel ORA_DISK_1: piece
handle=/u01/app/oracle/fast_recovery_area/CDB2/backupset/2012_09
_07/o1_mf_nnndf_TAG20120907T085859_84mftokq_.bkp
tag=TAG20120907T085859
channel ORA_DISK_1: restored backup piece 1
channel ORA_DISK_1: restore complete, elapsed time: 00:02:17
Finished restore at 07-09-2012 09:18:20
Starting restore at 07-09-2012 09:16:02
allocated channel: ORA_DISK_1
channel ORA_DISK_1: SID=240 device type=DISK

channel ORA_DISK_1: starting datafile backup set restore
channel ORA_DISK_1: specifying datafile(s) to restore from
backup set
channel ORA_DISK_1: restoring datafile 00001 to
/u01/app/oracle/oradata/cdb2/system01.dbf
channel ORA_DISK_1: reading from backup piece
/u01/app/oracle/fast_recovery_area/CDB2/backupset/2012_09_07/o1_
mf_nnndf_TAG20120907T085859_84mftokq_.bkp
channel ORA_DISK_1: piece
handle=/u01/app/oracle/fast_recovery_area/CDB2/backupset/2012_09
_07/o1_mf_nnndf_TAG20120907T085859_84mftokq_.bkp
tag=TAG20120907T085859
channel ORA_DISK_1: restored backup piece 1
channel ORA_DISK_1: restore complete, elapsed time: 00:02:17
Finished restore at 07-09-2012 09:18:20

RMAN> RECOVER TABLESPACE SYSTEM;

Starting recover at 07-09-2012 09:24:59
using channel ORA_DISK_1
```

```
starting media recovery

archived log for thread 1 with sequence 4 is already on disk as
file
/u01/app/oracle/fast_recovery_area/CDB2/archivelog/2012_09_07/o1
_mf_1_4_84mg9v6y_.arc
archived log for thread 1 with sequence 5 is already on disk as
file
/u01/app/oracle/fast_recovery_area/CDB2/archivelog/2012_09_07/o1
_mf_1_5_84mgns9p_.arc
archived log for thread 1 with sequence 6 is already on disk as
file
/u01/app/oracle/fast_recovery_area/CDB2/archivelog/2012_09_07/o1
_mf_1_6_84mgntkz_.arc
archived log for thread 1 with sequence 7 is already on disk as
file
/u01/app/oracle/fast_recovery_area/CDB2/archivelog/2012_09_07/o1
_mf_1_7_84mgnyld_.arc
archived log for thread 1 with sequence 8 is already on disk as
file
/u01/app/oracle/fast_recovery_area/CDB2/archivelog/2012_09_07/o1
_mf_1_8_84mgnxtq_.arc
archived log file
name=/u01/app/oracle/fast_recovery_area/CDB2/archivelog/2012_09_
07/o1_mf_1_4_84mg9v6y_.arc thread=1 sequence=4
archived log file
name=/u01/app/oracle/fast_recovery_area/CDB2/archivelog/2012_09_
07/o1_mf_1_5_84mgns9p_.arc thread=1 sequence=5
archived log file
name=/u01/app/oracle/fast_recovery_area/CDB2/archivelog/2012_09_
07/o1_mf_1_6_84mgntkz_.arc thread=1 sequence=6
media recovery complete, elapsed time: 00:00:03
Finished recover at 07-09-2012 09:25:04

RMAN> ALTER DATABASE OPEN;

Statement processed
RMAN>
```

4. Back up the CDB.

```
RMAN> BACKUP DATABASE PLUS ARCHIVELOG DELETE INPUT;
…
RMAN> exit
$
```

Practice 6-11: RMAN Recovery from Non-Essential Root Data File Loss (optional)

Overview

In this practice, you will recover from a non-essential root data file loss.

Tasks

1. Remove a data file of the SYSAUX tablespace of the root of cdb2.

```
$ sqlplus / as sysdba

Connected to:
Oracle Database 12c Enterprise Edition Release 12.1.0.0.2 -
64bit Production
With the Partitioning, OLAP, Data Mining and Real Application
Testing options

SQL> select file_name from dba_data_files
     where tablespace_name='SYSAUX';

FILE_NAME
-------------------------------------------------------------
/u01/app/oracle/oradata/cdb2/sysaux01.dbf

SQL> !rm /u01/app/oracle/oradata/cdb2/sysaux01.dbf
SQL> EXIT
$
```

2. Run RMAN to connect to cdb2 with a user with SYSDBA or SYSBACKUP privilege.

```
$ rman target /

connected to target database: CDB2 (DBID=534631279)
RMAN>
```

3. Proceed with the traditional procedure to restore the missing data file and recover the tablespace as it were a non-CDB.

```
RMAN> ALTER TABLESPACE sysaux OFFLINE IMMEDIATE;

using target database control file instead of recovery catalog
Statement processed

RMAN> RESTORE TABLESPACE sysaux;

Starting restore at 07-09-2012 09:46:36
allocated channel: ORA_DISK_1
channel ORA_DISK_1: SID=237 device type=DISK
channel ORA_DISK_1: starting datafile backup set restore
channel ORA_DISK_1: specifying datafile(s) to restore from
backup set
channel ORA_DISK_1: restoring datafile 00003 to
/u01/app/oracle/oradata/cdb2/sysaux01.dbf
channel ORA_DISK_1: reading from backup piece
/u01/app/oracle/fast_recovery_area/CDB2/backupset/2012_09_07/o1_
mf_nnndf_TAG20120907T093436_84mhxgdw_.bkp
channel ORA_DISK_1: piece
handle=/u01/app/oracle/fast_recovery_area/CDB2/backupset/2012_09
_07/o1_mf_nnndf_TAG20120907T093436_84mhxgdw_.bkp
tag=TAG20120907T093436
channel ORA_DISK_1: restored backup piece 1
channel ORA_DISK_1: restore complete, elapsed time: 00:01:35
Finished restore at 07-09-2012 09:48:12

RMAN> RECOVER TABLESPACE sysaux;

Starting recover at 07-09-2012 09:59:33
using channel ORA_DISK_1

starting media recovery
media recovery complete, elapsed time: 00:00:02

Finished recover at 07-09-2012 09:59:36

RMAN> ALTER TABLESPACE sysaux ONLINE;

Statement processed

RMAN> exit
$
```

Practice 6-12: PITR on PDB Tablespaces (optional)

Overview

In this practice, you will perform a PITR on a non-essential PDB data file. Rows in a table in the `PDB2_2` pluggable database `TEST_TBS` tablespace have been incorrectly deleted. You have to restore the situation to the time before the rows were deleted and committed.

Assumptions

The PDB `pdb2_2` has been successfully created after completion of Practice 3-4.

Tasks

1. Set the situation where deleted rows have been committed.
   a. Create a `TEST_TBS` tablespace in `PDB2_2`, a local user `LOCAL_TEST`, and a table.

```
$ sqlplus sys/oracle_4U@PDB2_2 as sysdba
Connected.
SQL> create tablespace test_pdb datafile
     '/u01/app/oracle/oradata/cdb2/pdb2_2/test_pdb1.f' size 3m;

Tablespace created.

SQL>
```

```
SQL> create user local_test identified by p
     default tablespace test_pdb;

User created.

SQL>
```

```
SQL> grant create session, create table, unlimited tablespace
     to local_test;

Grant succeeded.

SQL> EXIT
$
```

   b. Back up the new tablespace.

```
$ rman target /

connected to target database: CDB2 (DBID=540373866)

RMAN> backup pluggable database pdb2_2;
```

```
Starting backup at 02-07-2012 05:14:16
using channel ORA_DISK_1
channel ORA_DISK_1: starting full datafile backup set
channel ORA_DISK_1: specifying datafile(s) in backup set
input datafile file number=00013
name=/u01/app/oracle/oradata/cdb2/pdb2_2/CDB2/C3DCC6FF8EFD51EEE0
43160200C0E8F2/datafile/o1_mf_sysaux_7z3o334q_.dbf
input datafile file number=00012
name=/u01/app/oracle/oradata/cdb2/pdb2_2/CDB2/C3DCC6FF8EFD51EEE0
43160200C0E8F2/datafile/o1_mf_system_7z3o333q_.dbf
input datafile file number=00015
name=/u01/app/oracle/oradata/cdb2/pdb2_2/CDB2/C3DCC6FF8EFD51EEE0
43160200C0E8F2/datafile/o1_mf_ldata_7z3o3b4z_.dbf
input datafile file number=00014
name=/u01/app/oracle/oradata/cdb2/pdb2_2/CDB2/C3DCC6FF8EFD51EEE0
43160200C0E8F2/datafile/o1_mf_users_7z3o3b2h_.dbf
input datafile file number=00016
name=/u01/app/oracle/oradata/cdb2/pdb2_2/test_pdb1.f
channel ORA_DISK_1: starting piece 1 at 02-07-2012 05:14:16
channel ORA_DISK_1: finished piece 1 at 02-07-2012 05:14:31
piece
handle=/u01/app/oracle/fast_recovery_area/CDB2/backupset/2012_07
_02/o1_mf_nnndf_TAG20120702T171416_7z3oq86v_.bkp
tag=TAG20120702T171416 comment=NONE
channel ORA_DISK_1: backup set complete, elapsed time: 00:00:15
Finished backup at 02-07-2012 05:14:31

Starting Control File and SPFILE Autobackup at 02-07-2012
05:14:31
piece
handle=/u01/app/oracle/fast_recovery_area/CDB2/autobackup/2012_0
7_02/o1_mf_s_787598071_7z3oqqbf_.bkp comment=NONE
Finished Control File and SPFILE Autobackup at 02-07-2012
05:14:32

RMAN> exit
$
```

c. Create a table with 4 rows, check the SCN, delete 2 rows and recheck the SCN.

```
$ sqlplus sys/oracle_4U@PDB2_2 as sysdba
Connected to:
Oracle Database 12c Enterprise Edition Release 12.1.0.1.0 -
64bit Production
With the Partitioning, OLAP, Advanced Analytics and Real
Application Testing options
SQL> create table local_test.tab_test (c number);
```

```
Table created.

SQL> insert into local_test.tab_test values (1);

1 row created.

SQL> insert into local_test.tab_test values (2);

1 row created.

SQL> insert into local_test.tab_test values (3);

1 row created.

SQL> insert into local_test.tab_test values (4);

1 row created.

SQL> commit;

Commit complete.

SQL> select timestamp_to_scn(sysdate) from v$database;

TIMESTAMP_TO_SCN(SYSDATE)
-------------------------
                  1861446

SQL> delete from local_test.tab_test where rownum < 3;

2 rows deleted.

SQL> commit;

Commit complete.

SQL> select * from local_test.tab_test;

         C
----------
         3
```

```
          4

SQL> exit
$
```

3. Set the situation back when rows were all present in the table.

   There are three solutions.

   - An incomplete CDB recovery. Start the CDB in mount state.
   - PDB tablespace Point-In-Time Recovery in PDB2_2: If you intend to perform a PDB tablespace Point-In-Time Recovery, you must drop the OPEN_ALL_PDBS trigger before proceeding.
   - PDB Point-In-Time Recovery.

   The following steps show how to perform a PDB Point-In-Time Recovery.

   a. Connect to cdb2 and close PDB2_2.

```
$ rman target /

connected to target database: CDB2 (DBID=546459337)

RMAN> ALTER PLUGGABLE DATABASE pdb2_2 CLOSE;

using target database control file instead of recovery catalog
Statement processed

RMAN>
```

   b. Restore and recover the pluggable database back to the SCN before the delete.

```
RMAN> RUN {
SET UNTIL SCN = 1861446;
RESTORE PLUGGABLE DATABASE pdb2_2;
RECOVER PLUGGABLE DATABASE pdb2_2 AUXILIARY
       DESTINATION='/u01/app/oracle/oradata';
ALTER PLUGGABLE DATABASE pdb2_2 OPEN RESETLOGS;
}
executing command: SET until clause

Starting restore at 02-07-2012 05:15:32
allocated channel: ORA_DISK_1
channel ORA_DISK_1: SID=20 device type=DISK

channel ORA_DISK_1: starting datafile backup set restore
channel ORA_DISK_1: specifying datafile(s) to restore from
backup set
channel ORA_DISK_1: restoring datafile 00014 to
/u01/app/oracle/oradata/cdb2/pdb2_2/CDB2/DCBF06D282BC62C8E043652
3B98BA901/datafile/o1_mf_system_8s6p6w8q_.dbf
```

```
channel ORA_DISK_1: restoring datafile 00015 to
/u01/app/oracle/oradata/cdb2/pdb2_2/CDB2/DCBF06D282BC62C8E043652
3B98BA901/datafile/o1_mf_sysaux_8s6p6w92_.dbf
channel ORA_DISK_1: restoring datafile 00030 to
/u01/app/oracle/oradata/cdb2/pdb2_2/test_pdb1.f
channel ORA_DISK_1: reading from backup piece
/u01/app/oracle/fast_recovery_area/CDB2/backupset/2012_05_16/o1_
mf_nnndf_TAG20130516T085734_8s97vz9y_.bkp
channel ORA_DISK_1: piece
handle=/u01/app/oracle/fast_recovery_area/CDB2/backupset/2012_05
_16/o1_mf_nnndf_TAG20130516T085734_8s97vz9y_.bkp
tag=TAG20130516T085734
channel ORA_DISK_1: restored backup piece 1
channel ORA_DISK_1: restore complete, elapsed time: 00:00:15
Finished restore at 02-07-2012 05:16:32

Starting recover at 02-07-2012 05:17:32
current log archived
using channel ORA_DISK_1
RMAN-05026: WARNING: presuming following set of tablespaces
applies to specified Point-in-Time

List of tablespaces expected to have UNDO segments
Tablespace SYSTEM
Tablespace UNDOTBS1

Creating automatic instance, with SID='qgkd'

initialization parameters used for automatic instance:
db_name=CDB2
db_unique_name=qgkd_pitr_pdb2_2_CDB2
compatible=12.0.0.0.0
db_block_size=8192
db_files=200
sga_target=1G
processes=80
diagnostic_dest=/u01/app/oracle
db_create_file_dest=/u01/app/oracle/oradata
log_archive_dest_1='location=/u01/app/oracle/oradata'
enable_pluggable_database=true
_clone_one_pdb_recovery=true
#No auxiliary parameter file used
```

```
starting up automatic instance CDB2
…
Finished recover at 02-07-2012 05:25:32

sql statement: alter database open read only

Oracle instance shut down

Removing automatic instance
Automatic instance removed
auxiliary instance file
/u01/app/oracle/oradata/CDB2/datafile/o1_mf_sysaux_8s98gkgd_.dbf
deleted
auxiliary instance file
/u01/app/oracle/oradata/CDB2/controlfile/o1_mf_8s98g4tr_.ctl
deleted
Finished recover at 02-07-2012 05:26:30

Statement processed

RMAN> exit
$
```

c. Check the content of the `local_test.tab_test` table.

```
$ sqlplus sys/oracle_4U@pdb2_2 as sysdba

Connected to:
Oracle Database 12c Enterprise Edition Release 12.1.0.0.2 -
64bit Production
With the Partitioning, OLAP, Data Mining and Real Application
Testing options

SQL> select * from local_test.tab_test;

         C
----------
         1
         2
         3
         4
SQL> EXIT
$
```

d. Back up the CDB.

```
$ rman target /

connected to target database: CDB2 (DBID=534631279)

RMAN> DELETE OBSOLETE;

using target database control file instead of recovery catalog
RMAN retention policy will be applied to the command
RMAN retention policy is set to redundancy 1
allocated channel: ORA_DISK_1
channel ORA_DISK_1: SID=260 device type=DISK
Deleting the following obsolete backups and copies:
…
Do you really want to delete the above objects (enter YES or
NO)? yes
…
Deleted 5 objects

RMAN> BACKUP DATABASE PLUS ARCHIVELOG delete all input;
…
RMAN> exit
$
```

Practice 6-13: Flashback from Common User Drop (optional)

Overview

In this practice, you will flash back the CDB after a common user has been dropped.

Assumptions

The `C##_USER` common user exists in `cdb2`. This has been completed in practice 5-2.

Tasks

1. Set the CDB `cdb2` in FLASHBACK mode.

```
$ export NLS_DATE_FORMAT='DD-MM-YYYY HH:MI:SS'
$ sqlplus / as sysdba

Connected to:
Oracle Database 12c Enterprise Edition Release 12.1.0.0.2 -
64bit Production
With the Partitioning, OLAP, Data Mining and Real Application
Testing options

SQL> SELECT flashback_on from V$DATABASE;

FLASHBACK_ON
------------------
NO

SQL> SHUTDOWN IMMEDIATE
Database closed.
Database dismounted.
ORACLE instance shut down.

SQL> STARTUP MOUNT
ORACLE instance started.

Total System Global Area 1068937216 bytes
Fixed Size                  2248280 bytes
Variable Size             281018792 bytes
Database Buffers          780140544 bytes
Redo Buffers                5529600 bytes
Database mounted.

SQL> ALTER SYSTEM SET
           DB_FLASHBACK_RETENTION_TARGET=2880 SCOPE=BOTH;
```

```
System altered.

SQL> ALTER DATABASE FLASHBACK ON;

Database altered.

SQL> ALTER DATABASE OPEN;

Database altered.

SQL>
```

2. Drop the common user `C##_USER`.

   a. Verify that `C##_USER` exists as a common user.

```
SQL> col username format A20
SQL> select USERNAME, COMMON, CON_ID from cdb_users
     where username='C##_USER';

USERNAME             COM     CON_ID
-------------------- --- ----------
C##_USER             YES          1
C##_USER             YES          3
C##_USER             YES          4
C##_USER             YES          5
C##_USER             YES          6

SQL> select timestamp_to_scn(current_timestamp) from v$database;

TIMESTAMP_TO_SCN(CURRENT_TIMESTAMP)
------------------------------------
                             2455483

SQL>
```

   b. Drop the user.

```
SQL> DROP USER C##_USER CASCADE;
User dropped.

SQL> alter system switch logfile;

System altered.

SQL> alter system switch logfile;
```

```
System altered.

SQL> alter system switch logfile;

System altered.

SQL> alter system switch logfile;

System altered.

SQL>
```

3. Proceed with the flashback database operation.

```
SQL> SHUTDOWN IMMEDIATE
Database closed.
Database dismounted.
ORACLE instance shut down.

SQL> STARTUP MOUNT
ORACLE instance started.

Total System Global Area 1068937216 bytes
Fixed Size                  2248280 bytes
Variable Size             281018792 bytes
Database Buffers          780140544 bytes
Redo Buffers                5529600 bytes
Database mounted.

SQL> FLASHBACK DATABASE TO SCN 2455483;

Flashback complete.

SQL>
```

4. Open the database in `READ ONLY` mode to review changes before opening CDB with `RESETLOGS`.

```
SQL> ALTER DATABASE OPEN READ ONLY;

Database altered.

SQL> select USERNAME, COMMON, CON_ID from cdb_users
     where  username='C##_USER';

USERNAME             COM     CON_ID
-------------------- --- ----------
C##_USER             YES          1

SQL>
```

5. Open PDBs in READ ONLY to review all changes.

```
SQL> ALTER PLUGGABLE DATABASE ALL OPEN READ ONLY;

Pluggable database altered.

SQL> select USERNAME, COMMON, CON_ID from cdb_users
     where  username='C##_USER';

USERNAME             COM     CON_ID
-------------------- --- ----------
C##_USER             YES          1
C##_USER             YES          3
C##_USER             YES          4
C##_USER             YES          5
C##_USER             YES          6

SQL>
```

6. Open the CDB with `RESETLOGS`.

```
SQL> SHUTDOWN IMMEDIATE
Database closed.
Database dismounted.
ORACLE instance shut down.

SQL> STARTUP MOUNT
ORACLE instance started.

Total System Global Area 1068937216 bytes
Fixed Size                  2248280 bytes
```

```
Variable Size             281018792 bytes
Database Buffers          780140544 bytes
Redo Buffers                5529600 bytes
Database mounted.

SQL> FLASHBACK DATABASE TO SCN 2455483;

Flashback complete.

SQL> ALTER DATABASE OPEN RESETLOGS;

Database altered.

SQL>
```

7. Check that the C##_USER can connect in each container.

```
SQL> connect C##_USER/x
Connected.
SQL> connect C##_USER/x@PDB2
Connected.
SQL> connect C##_USER/x@PDB2_2
Connected.
SQL> exit
$
```

8. Back up the CDB.

```
$ rman target /

connected to target database: CDB2 (DBID=534631279)

RMAN> BACKUP DATABASE PLUS ARCHIVELOG delete all input;
…

RMAN> exit
$
```

Practices for Lesson 7: Heat Map, Automatic Data Optimization and Online Datafile Move

Chapter 7

Practices for Lesson 7

Practices Overview

In these practices, you will exercise yourself on new features of ILM and Online Move operations.

In the first practices, you will exercise yourself on new features of ILM and more precisely on Heat Map and Automatic Data Optimization (ADO).

In the last practice, you will exercise yourself on moving datafiles online.

Assumptions

The environment is prepared beforehand, namely installed an Oracle database 12c non-CDB `orcl`. The ILM new features are not supported in a multitenant container database (CDB). Any attempt to enable this feature will raise user exceptions.

Practice 7-1: Enabling Heat Map

Overview

In this practice, you will enable activity tracking or heat map.

Tasks

1. Perform several operations to cleanup any existing ADO policies and tablespaces.

   a. Make sure you are in the ~/labs/ADO directory.

```
$ cd ~/labs/ADO
$
```

   b. Ensure your environment points to the `orcl` instance.

```
$ . oraenv
ORACLE_SID = [orcl] ? orcl
The Oracle base remains unchanged with value /u01/app/oracle
$
```

   c. Start the `orcl` instance up.

```
$ sqlplus / as sysdba

Connected to an idle instance.
SQL> startup
ORACLE instance started.

Total System Global Area  400846848 bytes
Fixed Size                  2271568 bytes
Variable Size             339740336 bytes
Database Buffers           50331648 bytes
Redo Buffers                8503296 bytes
Database mounted.
Database opened.
SQL> EXIT
$
```

   d. Run the `ADO_cleanup.sh` script to cleanup any existing ADO policy and tablespaces.

```
$ ./ADO_cleanup.sh
$
```

2. Set the `HEAT_MAP` instance parameter to `ON` at the instance scope.

```
$ sqlplus / as sysdba
```

```
Connected to:
Oracle Database 12c Enterprise Edition Release 12.1.0.0.2 -
64bit Production
With the Partitioning, OLAP, Advanced Analytics, Real
Application Testing and Unified Auditing options

SQL> ALTER SYSTEM SET heat_map=on SCOPE=BOTH;

System altered.

SQL> EXIT
$
```

Practice 7-2: Automatic Data Optimization – Creating a TIER Policy

Overview

ADO allows you to automate the movement of a segment to another tablespace under certain circumstances. The default implicit condition under which the tiering policy would automatically move a table to another tablespace is based on the fullness of the source tablespace where the table resides in.

In this practice, you will create and enable an ADO tiering policy on the SCOTT.EMPLOYEE table. The policy will move the table to the LOW_COST_STORAGE tablespace when the source ILMTBS tablespace where the table resides on is less than 95% free.

Tasks

1. Set up the environment before creating the tiering storage ADO policy on SCOTT.EMPLOYEE table.
   a. Run the ADO_setup.sh script to ensure that the user SCOTT has the required privileges to execute the various SELECT statements on dictionary views.

```
$ ./ADO_setup.sh
$
```

   b. Create the ADOTBS tablespace to store the SCOTT.EMPLOYEE and insert rows into the SCOTT.EMPLOYEE table.

```
$ sqlplus system/oracle_4U

Connected to:
Oracle Database 12c Enterprise Edition Release 12.1.0.0.2 -
64bit Production
With the Partitioning, OLAP, Advanced Analytics, Real
Application Testing and Unified Auditing options

SQL> CREATE TABLESPACE adotbs
  DATAFILE  '/u01/app/oracle/oradata/orcl/adotbs1.dbf'
  size 10m reuse autoextend off extent management local;
2    3

Tablespace created.

SQL>
```

   c. Create the tablespace LOW_COST_STORE where the SCOTT.EMPLOYEE table may be moved to due to space pressure.

```
SQL> CREATE TABLESPACE low_cost_store
  DATAFILE  '/u01/app/oracle/oradata/orcl/lcs.dbf'
  size 200M;
2    3

Tablespace created.
```

```
SQL>
```

2. Create and store the `SCOTT.EMPLOYEE` table on the `ILMTBS` tablespace and blow it up with about 3500 rows. The rows inserted should raise the percentage of empty space in `ILMTBS` tablespace to less than 95%. When the ILM policy will be created and evaluated, this will trigger an ADO action to move the segment to the `LOW_COST_STORE` tablespace.
   a. Create and store the `SCOTT.EMPLOYEE` table on the `ADOTBS` tablespace and blow it up with about 3500 rows.

```
SQL> CONNECT scott/oracle_4U
Connected.
SQL> drop table employee purge;
drop table employee purge
           *
ERROR at line 1:
ORA-00942: table or view does not exist

SQL> create table employee (
         EMPNO     NUMBER(4) NOT NULL,
        ENAME     VARCHAR2(10),
        JOB       VARCHAR2(9),
        MGR       NUMBER(4),
        HIREDATE  DATE,
        SAL       NUMBER(7,2),
        COMM      NUMBER(7,2),
        DEPTNO    NUMBER(2)
     )
     tablespace adotbs;
  2    3    4    5    6    7    8    9   10   11
Table created.

SQL> insert into employee (empno, ename, job, mgr, hiredate,
sal, comm, deptno)
select empno, ename, job, mgr, hiredate, sal, comm, deptno from
emp;
  2
14 rows created.

SQL> declare
       blowup PLS_INTEGER := 8;
       sql_test clob;
   begin
     for i in 1..blowup loop
      sql_test := 'insert /*+ append */ into employee
```

```
                    select * from   scott.employee';
      execute immediate sql_test;
      commit;
     end loop;
   end;
/
  2    3    4    5    6    7    8    9   10   11
PL/SQL procedure successfully completed.

SQL> SELECT count(*) FROM scott.employee;

  COUNT(*)
----------
      3584

SQL>
```

b. Verify that heat map statistics are already collected.

```
SQL> alter session set nls_date_format='dd-mon-yy hh:mi:ss';

Session altered.

SQL> COL object_name FORMAT A12
SQL> COL "Seg write" FORMAT A10
SQL> SELECT object_name, to_char(track_time,'DD-MON-YY
HH:MI:SS') "Tracking Time", segment_write "Seg write", FULL_SCAN
"Full Scan", lookup_scan "Lookup Scan"
FROM DBA_HEAT_MAP_SEG_HISTOGRAM
WHERE object_name='EMPLOYEE';
  2    3

OBJECT_NAME  Tracking Time        Seg write  Ful Loo
------------ ------------------------ ---------- --- ---
EMPLOYEE     06-MAR-13 12:29:32        YES        YES NO

SQL> SELECT object_name, segment_write_time, FULL_SCAN,
lookup_scan
FROM dba_heat_map_segment
WHERE object_name='EMPLOYEE';
  2    3

OBJECT_NAME  SEGMENT_WRITE_TIME FULL_SCAN          LOOKUP_SCAN
------------ ------------------ ------------------ -----------
EMPLOYEE     06-mar-13 12:53:07 06-mar-13 12:53:07
```

```
SQL> COL "Seg_write" FORMAT A10
SQL> COL "Seg_read" FORMAT A10
SQL> SELECT OBJECT_NAME, TRACK_TIME, SEGMENT_WRITE "Seg_write",
SEGMENT_READ "Seg_read", FULL_SCAN, LOOKUP_SCAN
     FROM v$heat_map_segment
     WHERE object_name='EMPLOYEE';

  2    3    4
OBJECT_NAME  TRACK_TIME         Seg_write  Seg_read   FUL LOO
------------ ------------------ ---------- ---------- --- ---
EMPLOYEE     06-mar-13 12:53:44 NO          NO         NO  NO

SQL>
```

c. Check the current free space in ADOTBS tablespace.

```
col tablespace format A16
SELECT /* + RULE */ df.tablespace_name "Tablespace",
df.bytes / (1024 * 1024) "Size (MB)",
SUM(fs.bytes) / (1024 * 1024) "Free (MB)",
Nvl(Round(SUM(fs.bytes) * 100 / df.bytes),1) "% Free",
Round((df.bytes - SUM(fs.bytes)) * 100 / df.bytes) "% Used"
FROM dba_free_space fs,
(SELECT tablespace_name,SUM(bytes) bytes
FROM dba_data_files
GROUP BY tablespace_name) df
WHERE fs.tablespace_name (+) = df.tablespace_name
GROUP BY df.tablespace_name,df.bytes
Order by 4;
```

```
SQL> col tablespace format A16
SQL> SELECT /* + RULE */ df.tablespace_name "Tablespace",
df.bytes / (1024 * 1024) "Size (MB)",
SUM(fs.bytes) / (1024 * 1024) "Free (MB)",
Nvl(Round(SUM(fs.bytes) * 100 / df.bytes),1) "% Free",
Round((df.bytes - SUM(fs.bytes)) * 100 / df.bytes) "% Used"
FROM dba_free_space fs,
(SELECT tablespace_name,SUM(bytes) bytes
FROM dba_data_files
GROUP BY tablespace_name) df
WHERE fs.tablespace_name (+) = df.tablespace_name
GROUP BY df.tablespace_name,df.bytes
Order by 4;
2    3    4    5    6    7    8    9   10   11   12
```

```

Tablespace         Size (MB)  Free (MB)     % Free     % Used
---------------- ---------- ---------- ---------- ----------
SYSTEM                  900          5          1         99
SYSAUX                 7590   363.9375          5         95
USERS                  8.75      .8125          9         91
EXAMPLE             358.125    33.6875          9         91
UNDOTBS1                165   139.6875         85         15
ADOTBS                   10     8.6875         87         13
LOW_COST_STORE          200        199        100          1

7 rows selected.
SQL>
```

3. Create a storage tiering policy on `SCOTT.EMPLOYEE` table.

```
SQL> ALTER TABLE scott.employee ILM ADD POLICY TIER TO
low_cost_store;

Table altered.

SQL>
```

Note: If you had not enabled the heat map, you would have received the following error message:

```
SQL> ALTER TABLE scott.employee ILM ADD POLICY TIER TO
low_cost_store;
ALTER TABLE scott.employee ILM ADD POLICY TIER TO low_cost_store
*
ERROR at line 1:
ORA-38342: heat map not enabled

SQL>
```

4. Verify that the policy is added.

```
SQL> COL policy_name format A12
SQL> COL TIER_TBS format A20
SQL> SELECT  policy_name, action_type, scope,
             tier_tablespace "TIER_TBS"
     FROM  user_ilmdatamovementpolicies
     ORDER BY policy_name;
  2    3    4
```

```

POLICY_NAME  ACTION_TYPE SCOPE   TIER_TBS
------------ ----------- ------- --------------------
P304         STORAGE     SEGMENT LOW_COST_STORE
```

```
SQL> SELECT policy_name, object_name, inherited_from, enabled
     FROM user_ilmobjects;

POLICY_NAME  OBJECT_NAME  INHERITED_FROM       ENA
------------ ------------ -------------------- ---
P304         EMPLOYEE     POLICY NOT INHERITED YES

SQL>
```

5. The ADO decision to move segments also depends on the default thresholds defined at the database level for all user-defined tablespaces. Set the `TBS_PERCENT_FREE` threshold to 95% and the `TBS_PERCENT_USED` threshold to 5%.

```
SQL> CONNECT / AS SYSDBA
Connected.
SQL> COL name format A20
SQL> COL value format 9999
SQL> SELECT * FROM dba_ilmparameters;

NAME                 VALUE
-------------------- -----
ENABLED                  1
JOB LIMIT               10
EXECUTION MODE           3
EXECUTION INTERVAL      15
TBS PERCENT USED        85
TBS PERCENT FREE        25

6 rows selected.

SQL> EXEC
dbms_ilm_admin.customize_ilm(DBMS_ILM_ADMIN.TBS_PERCENT_FREE,95)

PL/SQL procedure successfully completed.

SQL> EXEC
dbms_ilm_admin.customize_ilm(DBMS_ILM_ADMIN.TBS_PERCENT_USED,5)

PL/SQL procedure successfully completed.

SQL> SELECT * FROM dba_ilmparameters;

NAME                 VALUE
-------------------- -----
```

```
ENABLED                     1
JOB LIMIT                  10
EXECUTION MODE              3
EXECUTION INTERVAL         15
TBS PERCENT USED            5
TBS PERCENT FREE           95

6 rows selected.

SQL>
```

6. Step 2-c showed that the 5% `TBS_PERCENT_USED` threshold is already reached on the `ADOTBS` tablespace.
   For the purpose of the demo, we will not wait for the maintenance window to open that will trigger the ADO policies jobs. Instead, you are going to use the following PL/SQL block connected as the ADO policy owner.

```
SQL> CONNECT scott/oracle_4U
Connected.
SQL> declare
v_executionid number;
begin
dbms_ilm.execute_ILM (ILM_SCOPE => dbms_ilm.SCOPE_SCHEMA,
            execution_mode => dbms_ilm.ilm_execution_offline,
            task_id   => v_executionid);
end;
/
  2    3    4    5    6    7    8
PL/SQL procedure successfully completed.

SQL>
```

7. Check the current free space in `ADOTBS` tablespace. The `LOW_COST_STORE` may show a value for the column % Used, although the space used in `ADOTBS` may not have decreased. If this is the case, a few seconds later, run the same statement and you will see that the data dictionary has been updated to reflect the new situation.

```
COL tablespace format A16
SELECT /* + RULE */ df.tablespace_name "Tablespace",
df.bytes / (1024 * 1024) "Size (MB)",
SUM(fs.bytes) / (1024 * 1024) "Free (MB)",
Nvl(Round(SUM(fs.bytes) * 100 / df.bytes),1) "% Free",
Round((df.bytes - SUM(fs.bytes)) * 100 / df.bytes) "% Used"
FROM dba_free_space fs,
(SELECT tablespace_name,SUM(bytes) bytes
FROM dba_data_files
GROUP BY tablespace_name) df
```

```
WHERE fs.tablespace_name (+) = df.tablespace_name
GROUP BY df.tablespace_name,df.bytes
Order by 4;
```

```
SQL> COL tablespace format A16
SQL> SELECT /* + RULE */ df.tablespace_name "Tablespace",
df.bytes / (1024 * 1024) "Size (MB)",
SUM(fs.bytes) / (1024 * 1024) "Free (MB)",
Nvl(Round(SUM(fs.bytes) * 100 / df.bytes),1) "% Free",
Round((df.bytes - SUM(fs.bytes)) * 100 / df.bytes) "% Used"
FROM dba_free_space fs,
(SELECT tablespace_name,SUM(bytes) bytes
FROM dba_data_files
GROUP BY tablespace_name) df
WHERE fs.tablespace_name (+) = df.tablespace_name
GROUP BY df.tablespace_name,df.bytes
Order by 4;
   2    3    4    5    6    7    8    9   10   11   12

Tablespace          Size (MB)  Free (MB)     % Free     % Used
---------------- ---------- ---------- ---------- ----------
SYSTEM                  900          5          1         99
SYSAUX                 7600   362.8125          5         95
USERS                  8.75      .8125          9         91
EXAMPLE             358.125    33.6875          9         91
UNDOTBS1                165   139.6875         85         15
ADOTBS                   10          9         90         10
LOW_COST_STORE          200     198.75         99          1

7 rows selected.

SQL>
```

8. Display the task that evaluated the ADO policy and the job executed.

```
SQL> COL job_name format A20
SQL> COL object_name format A8
SQL> COL task_id format 99999
SQL> SELECT task_id, state FROM user_ilmtasks;

TASK_ID STATE
------- ---------
    1688 COMPLETED

SQL> SELECT TASK_ID, POLICY_NAME, OBJECT_NAME,
SELECTED_FOR_EXECUTION, JOB_NAME
FROM user_ilmevaluationdetails;

TASK_ID POLICY_NAME  OBJECT_N SELECTED_FOR_EXECUTION JOB_NAME
------- ------------ -------- ---------------------- ---------
   1688 P304        EMPLOYEE SELECTED FOR EXECUTION ILMJOB1204

SQL> SELECT task_id, job_name, job_state FROM user_ilmresults;

TASK_ID JOB_NAME             JOB_STATE
------- -------------------- ---------------------------------
   1688 ILMJOB1204           COMPLETED SUCCESSFULLY

SQL>
```

9. Find the segment `SCOTT.EMPLOYEE` that has been moved to the `LOW_COST_STORE` tablespace.

```
SQL> SELECT table_name, tablespace_name FROM  user_tables
 WHERE table_name = 'EMPLOYEE';
  2    3    4
TABLE_NAME                     TABLESPACE_NAME
------------------------------ ---------------------
EMPLOYEE                        LOW_COST_STORE

SQL> exit
$
```

10. Clean up your environment to get it ready for the next practice by running the following script.

```
$ ./ADO_cleanup.sh
$
```

Practice 7-3: Automatic Data Optimization – Creating a COMPRESS Policy

Overview

In this practice, you will create and enable a Row Store compression policy on the SCOTT.EMPLOYEE table so that rows get automatically compressed after 30 days of no modification on the table.

ADO for compression can only work if statistics related to data accesses at segment level and or data modifications at row and segment level are collected. Statistics are collected because Heat Map is already enabled. Starting the collection causes DML and access of all segments to be tracked in memory and then flushed to an on-disk statistics table, but only statistics post the time you enabled heat map are valid and will be considered by ADO. You will simulate a situation where statistics were collected dating one month before the heat map collection started. Therefore you have to use a procedure that sets the heat map start time earlier.

Tasks

1. Because you will introduce statistics dating a month, you have to set back the heat map start time.

```
$ sqlplus / as sysdba

Connected to:
Oracle Database 12c Enterprise Edition Release 12.1.0.0.2 -
64bit Production
With the Partitioning, OLAP, Advanced Analytics, Real
Application Testing and Unified Auditing options

SQL> exec dbms_ilm_admin.set_heat_map_start(sysdate-50)

PL/SQL procedure successfully completed.

SQL>
```

2. Create the procedure sys.print_compression_stats. This procedure uses the dbms_compression.get_compression_type predefined function which returns the compression type for a specified row of a table. You will use the procedure to verify that the rows are compressed after ADO policy task execution.

```
SQL> create or replace procedure print_compression_stats
     (owner varchar2,tabname varchar2) as
      type r_cursor is REF CURSOR;
      cmp_rec r_cursor;
      type rec_ctype is record
      (cmp_type   number(6));
      rec_cmp rec_ctype;
      stmt varchar2(200);
      qot varchar2(1) := '''';
      n_uncmp number :=0;
      n_advanced number :=0;
      n_other number :=0;
begin
  stmt := 'select dbms_compression.get_compression_type(';
  stmt := stmt||qot||owner||qot||','||qot||tabname||qot;
  stmt := stmt||',rowid)  from '||owner||'.'||tabname;
  open cmp_rec for stmt;
  loop
      fetch cmp_rec into rec_cmp;
      exit when cmp_rec%notfound;
      case rec_cmp.cmp_type
         when dbms_compression.COMP_NOCOMPRESS then
            n_uncmp    := n_uncmp + 1;
         when dbms_compression.COMP_ADVANCED then
            n_advanced := n_advanced + 1;
         when dbms_compression.COMP_BASIC then
            n_advanced := n_advanced + 1;
         else
            n_other := n_other + 1;
      end case;
  end loop;
  close cmp_rec;
  dbms_output.put_line('Compression Stats');
  dbms_output.put_line('------------------');
  dbms_output.put_line('Uncompressed          : ' || n_uncmp);
  dbms_output.put_line('Adv/basic compressed  : ' ||
n_advanced);
  dbms_output.put_line('Others                : ' || n_other);
end;
/
  2    3    4    5    6    7    8    9   10   11   12   13   14
15   16   17   18   19   20   21   22   23   24   25   26   27
28   29   30   31   32   33   34   35   36   37   38   39
Procedure created.

SQL> grant execute on print_compression_stats to public;


Grant succeeded.


SQL> create or replace public synonym print_compression_stats
            for sys.print_compression_stats;
  2
Synonym created.
```

```

SQL>
```

3. Create the `SCOTT.EMPLOYEE` table and insert rows.

```
SQL> CONNECT scott/oracle_4U
Connected.
SQL> drop table employee purge;
drop table employee purge
           *
ERROR at line 1:
ORA-00942: table or view does not exist

SQL> create table employee (
        EMPNO      NUMBER(4) NOT NULL,
        ENAME      VARCHAR2(10),
        JOB        VARCHAR2(9),
        MGR        NUMBER(4),
        HIREDATE   DATE,
        SAL        NUMBER(7,2),
        COMM       NUMBER(7,2),
        DEPTNO     NUMBER(2)
        ) tablespace example;
  2    3    4    5    6    7    8    9   10
Table created.

SQL> insert into employee (empno, ename, job, mgr, hiredate,
sal, comm, deptno)
select empno, ename, job, mgr, hiredate, sal, comm, deptno from
emp;
  2
14 rows created.

SQL> declare
       blowup PLS_INTEGER := 8;
       sql_test clob;
       begin
        for i in 1..blowup loop
        sql_test := 'insert /*+ append */ into employee
                     select * from   scott.employee';
        execute immediate sql_test;
        commit;
       end loop;
     end;
```

```
/
2   3   4   5   6   7   8   9   10   11   12
PL/SQL procedure successfully completed.

SQL> SELECT count(*) FROM scott.employee;


  COUNT(*)
----------
      3584

SQL>
```

4. Add a compression policy on `SCOTT.EMPLOYEE` table.

```
SET ECHO ON
SET NUMWIDTH 10
SET LINESIZE 300
SET TRIMSPOOL ON
SET TAB OFF
SET PAGESIZE 1000
COLUMN JOB_NAME FORMAT A15
COLUMN COMPRESSION_LEVEL FORMAT A17
COLUMN COMPLETION_TIME FORMAT A30
COLUMN COMMENTS FORMAT A10
COLUMN policy_name FORMAT A8
```

```
SQL> SET ECHO ON
SQL> SET NUMWIDTH 10
SQL> SET LINESIZE 300
SQL> SET TRIMSPOOL ON
SQL> SET TAB OFF
SQL> SET PAGESIZE 1000
SQL> COLUMN JOB_NAME FORMAT A15
SQL> COLUMN COMPRESSION_LEVEL FORMAT A17
SQL> COLUMN COMPLETION_TIME FORMAT A30
SQL> COLUMN COMMENTS FORMAT A10
SQL> COLUMN policy_name FORMAT A8
SQL> ALTER TABLE scott.employee ILM ADD POLICY ROW STORE
COMPRESS ADVANCED SEGMENT AFTER 30 DAYS OF NO MODIFICATION;


Table altered.


SQL>
```

Note: If you had not enabled heat map, you would have received the following error message:

```
SQL> ALTER TABLE scott.employee ILM ADD POLICY ROW STORE
COMPRESS ADVANCED SEGMENT AFTER 30 DAYS OF NO MODIFICATION;
ALTER TABLE scott.employee ILM ADD POLICY ROW STORE COMPRESS
ADVANCED SEGMENT AFTER 30 DAYS OF NO MODIFICATION
*
ERROR at line 1:
ORA-38342: heat map not enabled


SQL>
```

5. Verify that the policy is added.

```
SQL> SELECT  policy_name, action_type, scope, compression_level,
condition_type, condition_days
FROM     user_ilmdatamovementpolicies
ORDER BY policy_name;
  2    3    4


POLICY_N ACTION_TYPE SCOPE   COMPRESSION_LEVEL CONDITION_TYPE
CONDITION_DAYS
-------- ----------- ------- ----------------- ----------------
----- --------------
P324     COMPRESSION SEGMENT ADVANCED          LAST MODIFICATION
TIME             30


SQL>
```

6. Check if the COMPRESSION attribute of the table is disabled before ADO enables it.

```
SQL> SELECT compression, compress_for
FROM user_tables WHERE table_name = 'EMPLOYEE';


COMPRESS COMPRESS_FOR
-------- ------------------------------
DISABLED


SQL>
```

Check that no blocks are compressed.

```
SQL> set serveroutput on
SQL> exec print_compression_stats('SCOTT','EMPLOYEE')


Compression Stats
------------------
Uncompressed          : 3584
Adv/basic compressed  : 0
Others                : 0
```

```
PL/SQL procedure successfully completed.

SQL> ANALYZE TABLE scott.employee COMPUTE STATISTICS;

Table analyzed.

SQL> SELECT object_name, nrows_nc "Uncomp Rows", nrows_advanced
"Comp Rows", nrows_ehcc "HCC Comp Rows"
FROM   sys.compression_stat$ c, user_objects o
WHERE  c.obj#=o.object_id
AND    o.object_name='EMPLOYEE';
2    3    4
no rows selected

SQL>
```

7. Verify that heat map statistics are collected.

```
SQL> alter session set nls_date_format='dd-mon-yy hh:mi:ss';

Session altered.

SQL> COL object_name FORMAT A12
SQL> COL "Tracking Time" FORMAT A20
SQL> COL "Seg_write" FORMAT A9
SQL> COL "Full" FORMAT A4
SQL> COL "Seg_read" FORMAT A8
SQL> COL "Lookup Scan" FORMAT A12
SQL> SELECT object_name, to_char(track_time,'DD-MON-YY
HH:MI:SS') "Tracking Time", segment_write "Seg_write", FULL_SCAN
"Full", lookup_scan "Lookup Scan"
FROM DBA_HEAT_MAP_SEG_HISTOGRAM
WHERE object_name='EMPLOYEE';
  2    3

OBJECT_NAME  Tracking Time    Seg_write Full Lookup Scan
------------ -------------------- --------- ---- ------------
EMPLOYEE     06-MAR-13 01:19:57   YES    YES  NO

SQL> SELECT object_name, segment_write_time, FULL_SCAN,
lookup_scan
FROM dba_heat_map_segment
WHERE object_name='EMPLOYEE';
  2    3
OBJECT_NAME  SEGMENT_WRITE_TIME FULL_SCAN            LOOKUP_SCAN
```

```
------------ ------------------ ------------------ -----------
EMPLOYEE     06-mar-13 01:20:19 06-mar-13 01:20:19

SQL> SELECT OBJECT_NAME, TRACK_TIME,
            SEGMENT_WRITE "Seg_write",
            SEGMENT_READ "Seg_read",
            FULL_SCAN "Full", LOOKUP_SCAN "Look"
     FROM  v$heat_map_segment
     WHERE object_name='EMPLOYEE';

  2    3    4    5    6
OBJECT_NAME  TRACK_TIME         Seg_write Seg_read Full Loo
------------ ------------------ --------- -------- ---- ---
EMPLOYEE     06-mar-13 01:24:08 YES        NO       YES  NO

SQL>
```

8. Fake passage of time so that compression policy qualifies for ADO action. In real life situation, this would happen naturally through ADO activity tracking.
   a. Execute the procedure `dbms_ilm_admin.set_heat_map_table`. This procedure simulates the passage of time in read access only so that the compression policy qualifies for ADO action. In real life situation, this would happen naturally through a normal activity on the table and a regular ADO policy evaluation.

```
SQL> connect / as sysdba
Connected.
SQL> alter session set nls_date_format='dd-mon-yy hh:mi:ss';

Session altered.

SQL> exec
dbms_ilm_admin.set_heat_map_table('SCOTT','EMPLOYEE',null,sysdat
e-31,2)

PL/SQL procedure successfully completed.

SQL> COL object_name FORMAT A12
SQL> COL "Tracking Time" FORMAT A20
SQL> COL "Seg_write" FORMAT A9
SQL> COL "Full" FORMAT A4
SQL> COL "Seg_read" FORMAT A8
SQL> COL "Write" FORMAT A6
SQL> COL "Full Scan" FORMAT A4
SQL> COL "Lookup Scan" FORMAT A12
SQL> select object_name,
      to_char(track_time,'DD-MON-YY HH:MI:SS') "Tracking Time",
      segment_write "Write",
      FULL_SCAN "Full Scan", lookup_scan "Lookup Scan"
     from  DBA_HEAT_MAP_SEG_HISTOGRAM
     where object_name='EMPLOYEE'
     and   owner='SCOTT';

  2    3    4    5    6    7
OBJECT_NAME  Tracking Time    Write  Full Lookup Scan
------------ -------------------- ------ ---- ------------
EMPLOYEE     06-MAR-13 01:31:33   YESYES  NO
EMPLOYEE     03-FEB-13 01:27:48   NO      NO   NO

SQL>
```

There are two rows: one row relating recent write access and one row faking a previous read access.

```
SQL> select OBJECT_NAME, TRACK_TIME, SEGMENT_WRITE "Write",
            SEGMENT_READ "Read", FULL_SCAN "Full Scan",
            LOOKUP_SCAN "Lookup"
     from v$heat_map_segment
     where object_name='EMPLOYEE';
2    3    4    5
OBJECT_NAME  TRACK_TIME          Write  Rea Full Loo
------------ ------------------ ------ --- ---- ---
EMPLOYEE     06-mar-13 01:32:07 YES    NO  YES   NO

SQL>
```

There is only one row relating recent write access.

```
SQL> select OBJECT_NAME, TRACK_TIME,
            decode(SEGMENT_ACCESS,1, 'Write',2, 'Read',
                   'Other Acc') "Access"
     from  sys.heat_map_stat$ h, dba_objects o
     where o.object_id=h.obj#
     and   o.object_name='EMPLOYEE';

  2    3    4    5    6
OBJECT_NAME  TRACK_TIME          Access
------------ ------------------ ---------
EMPLOYEE     03-feb-13 01:27:48 Read

SQL>
```

There is only one row relating old read access.

b. Restart the instance to clear all statistics from memory, and mainly the statistics related to recent write access.

```
SQL> shutdown immediate
Database closed.
Database dismounted.
ORACLE instance shut down.
SQL> startup
ORACLE instance started.

Total System Global Area  663908352 bytes
Fixed Size                  2291280 bytes
Variable Size             285215152 bytes
Database Buffers          369098752 bytes
Redo Buffers                7303168 bytes
Database mounted.
Database opened.
SQL>
```

c. Check that the statistics are cleared up.

```
SQL> alter session set nls_date_format='dd-mon-yy hh:mi:ss';

Session altered.

SQL> COL object_name FORMAT A12
SQL> COL "Tracking Time" FORMAT A20
SQL> COL "Write" FORMAT A6
SQL> COL "Full Scan" FORMAT A4
SQL> COL "Lookup Scan" FORMAT A12
SQL> select object_name,
      to_char(track_time,'DD-MON-YY HH:MI:SS') "Tracking Time",
      segment_write "Seg_write",
      FULL_SCAN "Full Scan", lookup_scan "Lookup Scan"
     from  DBA_HEAT_MAP_SEG_HISTOGRAM
     where object_name='EMPLOYEE'
     and   owner='SCOTT';

  2    3    4    5    6    7
OBJECT_NAME  Tracking Time        Seg_write Full Lookup Scan
------------ -------------------- --------- ---- ------------
EMPLOYEE     03-FEB-13 01:27:48   NO          NO   NO

SQL>
```

There is now only one row: the one faking a previous read access.

```
SQL> select OBJECT_NAME, TRACK_TIME, SEGMENT_WRITE "Write",
            SEGMENT_READ "Read", FULL_SCAN "Full Scan",
            LOOKUP_SCAN "Lookup"
     from v$heat_map_segment
     where object_name='EMPLOYEE';
  2    3    4    5
no rows selected
SQL>
```

There are no more rows relating recent write access.

```
SQL> select OBJECT_NAME, TRACK_TIME,
            decode(SEGMENT_ACCESS,1, 'Write',2, 'Read',
                   'Other Acc') "Access"
     from  sys.heat_map_stat$ h, dba_objects o
     where o.object_id=h.obj#
     and   o.object_name='EMPLOYEE';
2    3    4    5    6
OBJECT_NAME  TRACK_TIME          Access
------------ ------------------ ---------
EMPLOYEE     03-feb-13 01:27:48 Read

SQL>
```

There is still one row relating old read access.

9. Now one month has passed without any modification on SCOTT.EMPLOYEE table. For the purpose of the demo, you will not wait until the maintenance window opens to execute the ADO policies job. You launch the ADO policy evaluation and ADO task execution immediately.

```
SQL> connect scott/oracle_4U
Connected.
SQL> alter session set nls_date_format='dd-mon-yy hh:mi:ss';

Session altered.

SQL> declare
      v_executionid number;
begin
    dbms_ilm.EXECUTE_ILM (
      ILM_SCOPE      => dbms_ilm.SCOPE_SCHEMA,
      execution_mode => dbms_ilm.ilm_execution_offline,
      task_id        => v_executionid);
end;
/
```

```
   2     3     4     5     6     7     8     9

PL/SQL procedure successfully completed.

SQL>
```

10. Display the result of the executed task.

```
SQL> COL task_id format 99999
SQL> COL task_owner format A8
SQL> COL policy_name format A4
SQL> COL job_name format A10
SQL> COL SELECTED_FOR_EXECUTION format A22
SQL> select task_id, task_owner, state
     from dba_ilmtasks where task_owner='SCOTT';

   TASK_ID TASK_OWNER STATE
---------- ---------- ----------------
      1710 SCOTT      COMPLETED

SQL>
```

11. If the STATE column displays ACTIVE, the task is still executing. Rerun the SELECT statement until STATE shows COMPLETED.

```
SQL> select task_id, policy_name, object_name,
selected_for_execution, job_name
FROM dba_ilmevaluationdetails where object_name='EMPLOYEE';

TASK_ID POLI OBJECT_NAME  SELECTED_FOR_EXECUTION JOB_NAME
------- ---- ------------ ---------------------- ----------
   1710 P324 EMPLOYEE       SELECTED FOR EXECUTION ILMJOB1284

SQL>
```

It is possible that several rows are displayed with a SELECTED_FOR_EXECUTION value of PRECONDITION NOT SATISFIED. The reason for the existence of these rows is that the evaluation of the condition set in the ADO policy was not yet satisfied. (there were still rows podified within the last month).

12. Verify the compression statistics for the SCOTT.EMPLOYEE segment. Use the print_compression_stats procedure created in task 1.

```
SQL> set serveroutput on
SQL> exec print_compression_stats('SCOTT','EMPLOYEE')

Compression Stats
------------------
Uncompressed         : 0
```

```
Adv/basic compressed : 3584
Others               : 0

PL/SQL procedure successfully completed.

SQL> SELECT compression, compress_for
     FROM user_tables WHERE table_name = 'EMPLOYEE';
  2
COMPRESS COMPRESS_FOR
-------- ------------------------------
ENABLED  ADVANCED

SQL> analyze table scott.employee compute statistics;

Table analyzed.

SQL> COL object_name format A8
SQL> SELECT object_name,
       nblk_nc "Uncomp Blocks",
       nrows_advanced "Comp Rows",
       nblk_advanced "Comp Blocks",
       nblk_ehcc "HCC Comp Blocks"
 FROM   sys.compression_stat$ c, user_objects o
 WHERE  c.obj#=o.object_id
 AND    o.object_name='EMPLOYEE';

2    3    4    5    6    7    8
OBJECT_N Uncomp Blocks  Comp Rows Comp Blocks HCC Comp Blocks
-------- ------------- ---------- ----------- ---------------
EMPLOYEE             0       3584           6               0

SQL>
```

Practice 7-4: Cleanup ADO Policies and Heat Map Statistics

Overview

In this practice you delete all ADO policies on SCOTT.EMPLOYEE table, stop collecting heat map statistics and clean up all heat map statistics.

Tasks

1. Delete all ADO policies on SCOTT.EMPLOYEE table.

```
SQL> ALTER TABLE scott.employee ILM DELETE_ALL;

Table altered.

SQL> connect / as sysdba
Connected.
SQL>
```

2. Stop heat map statistics collection.

```
SQL> ALTER SYSTEM SET heat_map=off SCOPE=BOTH;

System altered.

SQL>
```

3. Clean up all heat map statistics.

```
SQL> select OBJ#, TS#, TRACK_TIME from sys.heat_map_stat$;

      OBJ#        TS# TRACK_TIM
---------- ---------- ---------
    -1             -1 06-MAR-13
     30127          8 06-MAR-13
     26491          4 06-MAR-13

SQL> exec dbms_ilm_admin.clear_heat_map_all

PL/SQL procedure successfully completed.

SQL> select OBJ#, TS#, TRACK_TIME from sys.heat_map_stat$;

      OBJ#        TS# TRACK_TIM
---------- ---------- ---------
        -1         -1 06-MAR-13

SQL>
```

You notice that the procedure deletes all rows in `HEAT_MAP_STAT$` table except the dummy row.

Practice 7-5: Moving Data File Online

Overview

In this practice, you will move a data file to another location online.

Tasks

1. Create a tablespace `ONLINE_TBS` and find the list of data files in the `orcl` database.

```
SQL> COL name FORMAT A60
SQL> create tablespace ONLINE_TBS
datafile '/u01/app/oracle/oradata/orcl/online_tbs01.dbf'
size 10m;

Tablespace created.

SQL> select name from v$datafile;

NAME
--------------------------------------------------------------
/u01/app/oracle/oradata/orcl/system01.dbf
/u01/app/oracle/oradata/orcl/example01.dbf
/u01/app/oracle/oradata/orcl/sysaux01.dbf
/u01/app/oracle/oradata/orcl/undotbs01.dbf
/u01/app/oracle/oradata/orcl/users01.dbf
/u01/app/oracle/oradata/orcl/online_tbs01.dbf

SQL>
```

2. Move the datafile `/u01/app/oracle/oradata/orcl/online_tbs01.dbf` to `/u01/app/oracle/oradata/orcl/online` destination, online without taking it offline.
   a. Create the destination directory `/u01/app/oracle/oradata/orcl/online`.

```
SQL> !mkdir /u01/app/oracle/oradata/orcl/online
SQL>
```

   b. Move the data file `/u01/app/oracle/oradata/orcl/online_tbs01.dbf` to `/u01/app/oracle/oradata/orcl/online` destination, online.

```
SQL> ALTER DATABASE MOVE DATAFILE
  2  '/u01/app/oracle/oradata/orcl/online_tbs01.dbf'
  3  TO '/u01/app/oracle/oradata/orcl/online/online_tbs01.dbf';

Database altered.

SQL>
```

```
SQL> !ls -l /u01/app/oracle/oradata/orcl/online_tbs01.dbf
```

```
ls: /u01/app/oracle/oradata/orcl/online_tbs01.dbf: No such file
or directory

SQL> !ls -l /u01/app/oracle/oradata/orcl/online
-rw-r----- 1 oracle oinstall 9183232 Dec 12 11:26
online_tbs01.dbf

SQL>
```

3. Move the data file `/u01/app/oracle/oradata/orcl/online/online_tbs01.dbf` online back to `/u01/app/oracle/oradata/orcl` destination and keep the original file.

```
SQL> ALTER DATABASE MOVE DATAFILE
'/u01/app/oracle/oradata/orcl/online/online_tbs01.dbf'
  TO '/u01/app/oracle/oradata/orcl/online_tbs01.dbf' KEEP;

Database altered.

SQL>
```

```
SQL> !ls -l /u01/app/oracle/oradata/orcl/online_tbs01.dbf
-rw-r----- 1 oracle oinstall 9183232 Dec 12 11:28
/u01/app/oracle/oradata/orcl/online_tbs01.dbf

SQL> !ls -l /u01/app/oracle/oradata/orcl/online/online*
-rw-r----- 1 oracle oinstall 9183232 Dec 12 11:28
/u01/app/oracle/oradata/orcl/online/online_tbs01.dbf

SQL>
```

4. Move the data file `/u01/app/oracle/oradata/orcl/online_tbs01.dbf` online overwriting the `/u01/app/oracle/oradata/orcl/online/online_tbs01.dbf` file.

```
SQL> ALTER DATABASE MOVE DATAFILE
  '/u01/app/oracle/oradata/orcl/online_tbs01.dbf'
  TO '/u01/app/oracle/oradata/orcl/online/online_tbs01.dbf'
REUSE;

Database altered.

SQL>
```

```
SQL> !ls -l /u01/app/oracle/oradata/orcl/online_tbs01.dbf
ls: cannot access /u01/app/oracle/oradata/orcl/online_tbs01.dbf:
No such file or directory

SQL> !ls -l /u01/app/oracle/oradata/orcl/online/online*
```

```
-rw-r----- 1 oracle oinstall 9183232 Dec 12 11:30
/u01/app/oracle/oradata/orcl/online/online_tbs01.dbf


SQL>
```

5. Drop the tablespace `ONLINE_TBS` including the data files.

```
SQL> drop tablespace ONLINE_TBS including contents and
datafiles;


Tablespace dropped.


SQL> EXIT
$ ls -l /u01/app/oracle/oradata/orcl/online_tbs01.dbf
ls: cannot access /u01/app/oracle/oradata/orcl/online_tbs01.dbf:
No such file or directory
$ ls -l /u01/app/oracle/oradata/orcl/online/online_tbs01.dbf
ls: cannot access
/u01/app/oracle/oradata/orcl/online/online_tbs01.dbf: No such
file or directory
$
```

Practices for Lesson 8: In-Database Archiving and Temporal Validity

Chapter 8

Practices for Lesson 8

Practices Overview

In this practice, you will exercise yourself on new features like In-Database Archiving and Temporal Validity.

Practice 8-1: In-Database Archiving – Row-archival

Overview

In this practice, you will enable row-archival on HR.EMP_ARCH table in orcl database and sometimes display active rows only and sometimes display active and non-active rows.

1. Connected under SYSTEM, create the table HR.EMP_ARCH with ROW ARCHIVAL attribute.

```
$ . oraenv
ORACLE_SID = [orcl] ? orcl
The Oracle base remains unchanged with value /u01/app/oracle
$ sqlplus system/oracle_4U

Connected to:
Oracle Database 12c Enterprise Edition Release 12.1.0.0.2 -
64bit Production
With the Partitioning, OLAP, Data Mining and Real Application
Testing options

SQL> DROP TABLE hr.emp_arch PURGE;
DROP TABLE hr.emp_arch PURGE
              *
ERROR AT LINE 1:
ORA-00942: TABLE OR VIEW DOES NOT EXIST

SQL> CREATE TABLE hr.emp_arch
  (EMPNO  NUMBER(7), FULLNAME VARCHAR2(100),
   JOB VARCHAR2(9), MGR NUMBER(7))
   ROW ARCHIVAL;
  2    3    4
Table created.

SQL> DESC hr.emp_arch
 Name                             Null?    Type
 -------------------------------- -------- ------------------
 EMPNO                                      NUMBER(7)
 FULLNAME                                   VARCHAR2(100)
 JOB                                       VARCHAR2(9)
 MGR                                       NUMBER(7)

SQL>
```

2. Insert new rows in the table and verify the value of the archival column.

```
SQL> INSERT INTO hr.emp_arch (EMPNO , FULLNAME , JOB)
VALUES (100, 'JEAN', 'MGR');
```

```
  2
1 row created.

SQL> INSERT INTO hr.emp_arch (EMPNO , FULLNAME , JOB , MGR)
VALUES (101, 'ADAM', 'CLERK' ,100);
  2
1 row created.

SQL> INSERT INTO hr.emp_arch (EMPNO , FULLNAME , JOB , MGR)
VALUES (102, 'TOM', 'ADMIN' ,100);
  2
1 row created.

SQL> INSERT INTO hr.emp_arch (EMPNO , FULLNAME , JOB , MGR)
VALUES (103, 'JIM', 'WRITER' ,100);
  2
1 row created.

SQL> COMMIT;

Commit complete.

SQL>
```

3. Verify that the new row-archival column is displayed if explicitly required and that the default value is 0 for all active rows.

```
SQL> COL fullname FORMAT A10
SQL> COL ORA_ARCHIVE_STATE FORMAT A30
SQL> SELECT ORA_ARCHIVE_STATE, fullname FROM hr.emp_arch;

ORA_ARCHIVE_STATE            FULLNAME
--------------------------- ----------
0                            JEAN
0                            ADAM
0                            TOM
0                            JIM

SQL>
```

4. Performing a CTAS (`create table as select`) of a row-archival enabled table does not propagate the row-archival state column to the new table.

```
SQL> CREATE TABLE hr.emp AS SELECT * FROM hr.emp_arch;

Table created.

SQL> SELECT ORA_ARCHIVE_STATE, fullname FROM hr.emp;
SELECT ORA_ARCHIVE_STATE, fullname FROM hr.emp
       *
ERROR at line 1:
ORA-00904: "ORA_ARCHIVE_STATE": invalid identifier

SQL>
```

5. Update `ORA_ARCHIVE_STATE` column to reflect a non-active state for employee numbers `101` and `102`.

```
SQL> UPDATE hr.emp_arch
SET ORA_ARCHIVE_STATE=DBMS_ILM.ARCHIVESTATENAME(1)
WHERE empno IN (101,102);
2    3
2 rows updated.

SQL> COMMIT;

Commit complete.

SQL>
```

6. Verify that a `SELECT` statement displays only active rows by default.

```
SQL> SELECT ORA_ARCHIVE_STATE, fullname FROM hr.emp_arch;

ORA_ARCHIVE_STATE               FULLNAME
------------------------------ ----------
0                              JEAN
0                              JIM

SQL>
```

7. Display all rows of the tables, non-active and active rows.

```
SQL> ALTER SESSION SET ROW ARCHIVAL VISIBILITY = ALL;

Session altered.

SQL> SELECT ORA_ARCHIVE_STATE, fullname FROM hr.emp_arch;
```

```
ORA_ARCHIVE_STATE            FULLNAME
------------------------------ ----------
0                      JEAN
1                      ADAM
1                      TOM
0                      JIM

SQL>
```

8. Verify that an `INSERT AS SELECT` where the source and target tables are row-archival-enabled does not populate the target table's `ORA_ARCHIVE_STATE` column with the value of the corresponding column from the source table's. Instead, the default active row-archival state will be set.

```
SQL> ALTER TABLE hr.emp ROW ARCHIVAL;

Table altered.

SQL> INSERT INTO hr.emp
 SELECT EMPNO+100 , FULLNAME ||'_SENIOR'  , JOB , MGR
 FROM hr.emp_arch;
  2    3
4 rows created.

SQL> SELECT ORA_ARCHIVE_STATE, fullname FROM hr.emp_arch;

ORA_ARCHIVE_STATE    FULLNAME
-------------------- ----------
0                    JEAN
1                    ADAM
1                    TOM
0                    JIM

SQL> COL fullname FORMAT A30
SQL> SELECT ORA_ARCHIVE_STATE, fullname FROM hr.emp;

ORA_ARCHIVE_STATE    FULLNAME
-------------------- ------------------------------
0                    JEAN
0                    ADAM
0                    TOM
0                    JIM
0                    JEAN_SENIOR
0                    ADAM_SENIOR
0                    TOM_SENIOR
0                    JIM_SENIOR
```

```
8 rows selected.

SQL>
```

9. Disable the row-archival attribute on `HR.ARCH_EMP` table.

```
SQL> ALTER TABLE hr.emp_arch NO ROW ARCHIVAL;

Table altered.

SQL>
```

10. Verify that the row-archival column has been dropped.

```
SQL> SELECT ORA_ARCHIVE_STATE, fullname FROM hr.emp_arch;
SELECT ORA_ARCHIVE_STATE, fullname FROM hr.emp_arch
       *
ERROR at line 1:
ORA-00904: "ORA_ARCHIVE_STATE": invalid identifier



SQL> EXIT
$
```

Practice 8-2: Temporal Validity

Overview

In this practice, you set a valid time dimension on `HR.EMP` table to define a period of validity for each employee in the table.

Tasks

1. Connected under `SYSTEM`, Set the valid-time dimension on a table using existing columns.

```
$ . oraenv
ORACLE_SID = [orcl] ? orcl
The Oracle base remains unchanged with value /u01/app/oracle
$
```

```
$ sqlplus system/oracle_4U

Connected to:
Oracle Database 12c Enterprise Edition Release 12.1.0.0.2 -
64bit Production
With the Partitioning, OLAP, Data Mining and Real Application
Testing options

SQL> DROP TABLE hr.emp PURGE;

Table dropped.

SQL> CREATE TABLE HR.EMP
          (EMPNO NUMBER, SALARY NUMBER, DEPTID NUMBER,
           NAME VARCHAR2(100),
           USER_TIME_START DATE,
           USER_TIME_END   DATE,
        PERIOD FOR USER_TIME
                  (USER_TIME_START, USER_TIME_END));
    2    3    4    5    6    7
Table created.

SQL>
```

2. Check the implicit constraint created with the valid-time dimension.

```
SQL> select constraint_name from dba_constraints
     where table_name ='EMP' and OWNER='HR';
 2
CONSTRAINT_NAME
--------------------------------------------------------------
USER_TIME1ABD76

SQL>
```

3. Insert rows with start and and valid time values.

```
SQL> INSERT INTO hr.emp (empno , salary , deptid , name ,
                        USER_TIME_START, USER_TIME_END)
     VALUES (101,1900,90,'ADAM',
             to_date('01-JAN-2000', 'dd-mon-yyyy') ,
             to_date('31-DEC-2011', 'dd-mon-yyyy'));
  2    3    4    5
1 row created.

SQL> commit;

Commit complete.

SQL>
```

4. The DESCRIBE command shows the columns of the USER TIME valid-time dimension because they were explicitly defined in the table structure.

```
SQL> col name format A10
SQL> select NAME,
         to_char(USER_time_start,'dd-mon-yyyy') "Start",
         to_char(USER_time_end, 'dd-mon-yyyy') "End"
     from hr.emp;
  2    3    4
NAME       Start                End
---------- -------------------- --------------------
ADAM       01-jan-2000          31-dec-2011

SQL>
```

```
SQL> DESC hr.emp

Name                          Null?    Type
----------------------------- -------- -----------
 EMPNO                                 NUMBER
 SALARY                             NUMBER
 DEPTID                             NUMBER
 NAME                               VARCHAR2(100)
 USER_TIME_START                    DATE
 USER_TIME_END                      DATE

SQL>
```

```
SQL> select TABLE_NAME, COLUMN_NAME
     from dba_tab_cols where owner='HR' and table_name='EMP';
  2
TABLE_NAME      COLUMN_NAME
--------------- --------------------
EMP             USER_TIME
EMP             EMPNO
EMP             SALARY
EMP             DEPTID
EMP             NAME
EMP             USER_TIME_START
EMP             USER_TIME_END

7 rows selected.

SQL>
```

5. If you want to disassociate the columns of the valid-time dimension, drop the dimension, and redefine a new one without explicitly naming the two columns. The implicit columns created are disassociated. The `DESCRIBE` command does not show the `VALID_TIME_START` and `VALID_TIME_END` columns anymore as they are disassociated and were not explicitly defined at the table creation.

   Note: You still see the `USER_START_TIME` and `USER_END_TIME` columns in both describes because they are part of the user's definition of the table.

```
SQL> ALTER TABLE hr.emp DROP (PERIOD FOR user_time);

Table altered.

SQL> ALTER TABLE hr.emp ADD (PERIOD FOR VALID_time);

Table altered.

SQL> desc hr.emp

 Name                           Null?    Type
```

```
------------------------- -------- -----------
 EMPNO                             NUMBER
 SALARY                            NUMBER
 DEPTID                            NUMBER
 NAME                              VARCHAR2(100)
 USER_TIME_START                   DATE
 USER_TIME_END                     DATE

SQL>
```

```
SQL> select TABLE_NAME, COLUMN_NAME
     from dba_tab_cols where owner='HR' and table_name='EMP';
 2
TABLE_NAME          COLUMN_NAME
------------------- --------------------
EMP                  VALID_TIME_START
EMP                  EMPNO
EMP                  SALARY
EMP                  DEPTID
EMP                  NAME
EMP                  USER_TIME_START
EMP                  USER_TIME_END
EMP                  VALID_TIME_END
EMP                  VALID_TIME

9 rows selected.

SQL>
```

6. You can display them if you explicitly name them in the projection.

```
SQL> select NAME,
        to_char(valid_time_start, 'dd-mon-yyyy'),
        to_char(valid_time_end, 'dd-mon-yyyy')
     from hr.emp;
  2    3    4
NAME       TO_CHAR(VALID_TIME_S TO_CHAR(VALID_TIME_E
---------- -------------------- --------------------
ADAM

SQL>
```

7. Insert rows with different start and end dates of validity using the `/home/oracle/labs/VT/ins.sql` script.

```
SQL> DROP TABLE hr.emp PURGE;

Table dropped.

SQL> CREATE TABLE HR.EMP
           (EMPNO NUMBER, SALARY NUMBER, DEPTID NUMBER,
            NAME VARCHAR2(100),
      PERIOD FOR VALID_TIME );

Table created.

SQL>
```

```
SQL> DESC hr.emp

Name                       Null?    Type
-------------------------- -------- -----------
 EMPNO                             NUMBER
 SALARY                            NUMBER
 DEPTID                            NUMBER
 NAME                              VARCHAR2(100)

SQL>
```

```
SQL> @/home/oracle/labs/VT/ins.sql

1 row created.

1 row created.

1 row created.

1 row created.

1 row created.

1 row created.

1 row created.

1 row created.
```

```
Commit complete.

SQL>
```

8. View all rows. The disassociated columns don’t appear.

```
SQL> COL NAME FORMAT A8
SQL> COL EMPNO FORMAT 999
SQL> COL VALID_TIME_START FORMAT A35
SQL> COL VALID_TIME_END  FORMAT A35
SQL> select * from hr.emp;

     EMPNO     SALARY     DEPTID NAME
---------- ---------- ---------- ----------
       101       1900         90 ADAM
       102       1991         91 SCOTT
       103       1992         92 JIM
       104       1992         92 JEAN
       105       1993         93 MARIA
       106       1994         94 TOM
       107       1996         92 KIM
       108       1996         92 JAMES

8 rows selected.

SQL>
```

9. View all rows with explicit named valid-time columns.

```
SQL> select NAME,
         to_char(valid_time_start, 'dd-mon-yyyy') "Start",
         to_char(valid_time_end, 'dd-mon-yyyy') "End"
     from hr.emp order BY 2;

NAME      Start                  End
--------- --------------------- --------------------
ADAM      01-jan-1990           31-dec-2010
SCOTT     01-jan-1991           31-dec-2011
JIM       01-jan-1992           31-dec-2013
JEAN      01-jan-1992           31-dec-2012
MARIA     01-jan-1993           31-dec-2011
TOM       01-jan-1994
KIM       01-jan-1994           30-jun-1994
JAMES     31-dec-1992           31-dec-1994
```

```

8 rows selected.

SQL>
```

10. View rows using Valid-time temporal flashback queries.
    a. Using an `AS OF` query:

```
SQL> select NAME,
         to_char(valid_time_start, 'dd-mon-yyyy') "Start",
         to_char(valid_time_end, 'dd-mon-yyyy') "End"
     from hr.emp AS OF PERIOD FOR valid_time
                to_date('31-DEC-1992', 'dd-mon-yyyy') ;
  2    3    4    5
NAME     Start                End
-------- -------------------- --------------------
ADAM     01-jan-1990          31-dec-2010
SCOTT    01-jan-1991          31-dec-2011
JIM      01-jan-1992          31-dec-2013
JEAN     01-jan-1992          31-dec-2012
JAMES    31-dec-1992          31-dec-1994

SQL>
```

Are displayed only employees who were still valid at the date of '01-DEC-1992'.

b. Another example:

```
SQL> select NAME,
          to_char(valid_time_start, 'dd-mon-yyyy') "Start",
          to_char(valid_time_end, 'dd-mon-yyyy') "End"
     from hr.emp order BY 2;

NAME     Start                End
-------- -------------------- --------------------
ADAM     01-jan-1990          31-dec-2010
SCOTT    01-jan-1991          31-dec-2011
JIM      01-jan-1992          31-dec-2013
JEAN     01-jan-1992          31-dec-2012
MARIA    01-jan-1993          31-dec-2011
TOM      01-jan-1994
KIM      01-jan-1994          30-jun-1994
JAMES    31-dec-1992          31-dec-1994

8 rows selected.

```

```
SQL>
```

```
SQL> select NAME,
          to_char(valid_time_start, 'dd-mon-yyyy') "Start",
          to_char(valid_time_end, 'dd-mon-yyyy') "End"
     from hr.emp
          AS OF PERIOD FOR valid_time to_date('01-JAN-2013');

NAME     Start                End
-------- -------------------- --------------------
JIM      01-jan-1992          31-dec-2013
TOM      01-jan-1994

SQL>
```

Are displayed only employees who were still valid at the date of '01-JAN-2013'.

c. Using the following new clause `VERSIONS PERIOD FOR BETWEEN`:

```
SQL> select NAME,
         to_char(valid_time_start, 'dd-mon-yyyy') "Start",
         to_char(valid_time_end, 'dd-mon-yyyy') "End"
     from hr.emp order BY 2;

NAME     Start                End
-------- -------------------- --------------------
ADAM     01-jan-1990          31-dec-2010
SCOTT    01-jan-1991          31-dec-2011
JIM      01-jan-1992          31-dec-2013
JEAN     01-jan-1992          31-dec-2012
MARIA    01-jan-1993          31-dec-2011
TOM      01-jan-1994
KIM      01-jan-1994          30-jun-1994
JAMES    31-dec-1992          31-dec-1994

8 rows selected.

SQL>
```

```
SQL> select NAME,
          to_char(VALID_TIME_START, 'dd-mon-yyyy') "Start",
          to_char(VALID_TIME_END, 'dd-mon-yyyy') "End"
     from hr.emp VERSIONS PERIOD FOR valid_time
          BETWEEN to_date('31-DEC-1992', 'dd-mon-yyyy')
          AND     to_date('31-DEC-1993', 'dd-mon-yyyy');
```

```

NAME     Start                End
-------- -------------------- --------------------
ADAM     01-jan-1990          31-dec-2010
SCOTT    01-jan-1991          31-dec-2011
JIM      01-jan-1992          31-dec-2013
JEAN     01-jan-1992          31-dec-2012
MARIA    01-jan-1993          31-dec-2011
JAMES    31-dec-1992          31-dec-1994

6 rows selected.

SQL>
```

The query displays all employees whose `VALID_TIME_START` is less than or equal to '31-DEC-1992' and `VALID_TIME_END` greater than '31-DEC-1993'.

d. Another example:

```
SQL> select NAME,
       to_char(VALID_TIME_START, 'dd-mon-yyyy') "Start",
       to_char(VALID_TIME_END, 'dd-mon-yyyy') "End"
     from hr.emp
      VERSIONS PERIOD FOR valid_time
      BETWEEN to_date('31-DEC-2011', 'dd-mon-yyyy')
      AND     to_date('31-DEC-2012', 'dd-mon-yyyy');

NAME     Start                End
----------------------------- --------------------
JIM      01-jan-1992          31-dec-2013
JEAN     01-jan-1992          31-dec-2012
TOM      01-jan-1994

SQL>
```

The query displays all employees whose `VALID_TIME_START` is less than or equal to '31-DEC-2011' and `VALID_TIME_END` greater than '31-DEC-2012'.

11. Use new procedures of `DBMS_FLASHBACK_ARCHIVE` package to set the time visibility. Set the visibility of temporal data to currently valid data within the valid time period at the session level.

```
SQL> exec DBMS_FLASHBACK_ARCHIVE.ENABLE_AT_VALID_TIME('CURRENT')

PL/SQL procedure successfully completed.

SQL> select NAME,
         to_char(VALID_TIME_START, 'dd-mon-yyyy') "Start",
         to_char(VALID_TIME_END, 'dd-mon-yyyy') "End"
     from hr.emp;

NAME     Start                End
-------- -------------------- --------------------
JIM      01-jan-1992          31-dec-2013
TOM      01-jan-1994

SQL>
```

12. Set the visibility of temporal data to the full table.

```
SQL> exec DBMS_FLASHBACK_ARCHIVE.ENABLE_AT_VALID_TIME('ALL')

PL/SQL procedure successfully completed.

SQL> /

NAME     Start                End
-------- -------------------- --------------------
ADAM     01-jan-1990          31-dec-2010
SCOTT    01-jan-1991          31-dec-2011
JIM      01-jan-1992          31-dec-2013
JEAN     01-jan-1992          31-dec-2012
MARIA    01-jan-1993          31-dec-2011
TOM      01-jan-1994
KIM      01-jan-1994          30-jun-1994
JAMES    31-dec-1992          31-dec-1994

8 rows selected.

SQL> exit
$
```

Practice 8-3: Collecting User Context in FDA History Tables (optional)

Overview

In this practice, you collect user context information in the history table of an FDA enabled table.

Tasks

1. Create the tablespace for Temporal History tables and enable the HR.TEST_TABLE1 table for FDA.
   a. Make sure you are at the ~/labs/FDA directory and your environment points to the orcl instance.

```
$ cd ~/labs/FDA
$ . oraenv
ORACLE_SID = [orcl] ? orcl
The Oracle base remains unchanged with value /u01/app/oracle
$
```

   b. Create the hr.test_table1 table.

```
$ sqlplus / as sysdba

Connected to:
Oracle Database 12c Enterprise Edition Release 12.1.0.0.2 -
64bit Production
With the Partitioning, OLAP, Advanced Analytics, Real
Application Testing and Unified Auditing options

SQL> CREATE TABLE hr.test_table1
     (NUM number(8), NAME varchar2(25), NOW  date ) ;
  2
Table created.

SQL> INSERT INTO hr.test_table1 VALUES (1,'First test
row',sysdate);

1 row created.

SQL> INSERT INTO hr.test_table1 VALUES (2,'Second test
row',sysdate);

1 row created.

SQL> INSERT INTO hr.test_table1 VALUES (3,'Third test
row',NULL);

1 row created.
```

```
SQL> COMMIT;

Commit complete.

SQL>
```

c. Execute the `FDA_setup.sql` script.

```
SQL> @FDA_setup.sql

ALTER TABLE hr.test_table1  NO FLASHBACK ARCHIVE
*
ERROR at line 1:
ORA-55602: The table "HR"."TEST_TABLE1" is not enabled for
Flashback Archive



DROP FLASHBACK ARCHIVE fla1
*
ERROR at line 1:
ORA-55605: Incorrect Flashback Archive is specified



DROP TABLESPACE fda_tbs INCLUDING CONTENTS AND DATAFILES
*
ERROR at line 1:
ORA-00959: tablespace 'FDA_TBS' does not exist



Tablespace created.

Flashback archive created.

Table altered.

SQL>
```

2. Set the context-level collection to TYPICAL. You want to collect the username and module name of the user performing UPDATE executions on HR.EMPLOYEES. This level is sufficient to retrieve this information.

```
SQL> EXEC DBMS_FLASHBACK_ARCHIVE.SET_CONTEXT_LEVEL('TYPICAL')

PL/SQL procedure successfully completed.

SQL>
```

3. Perform some changes to the HR.EMPLOYEES table rows.

   a. Perform two UPDATE statements as HR user. If the user HR is locked, unlock the account first.

```
SQL> ALTER USER hr IDENTIFIED BY oracle_4U ACCOUNT UNLOCK;

User altered.

SQL> CONNECT HR/oracle_4U
Connected.
SQL> UPDATE hr.test_table1 SET NAME='Premier test row'
     WHERE num=1;

1 row updated.

SQL> COMMIT;

Commit complete.

SQL> UPDATE hr.test_table1 SET NAME='Primero test row'
     WHERE num=1;

1 row updated.

SQL>  COMMIT;

Commit complete.

SQL>
```

b. Retrieve the name of the flashback table.

```
SQL> select * from DBA_FLASHBACK_ARCHIVE_TABLES;

TABLE_NAME   OWNER_NAME FLASHBACK_ARCHIVE_NAME
------------ ---------- ----------------------
ARCHIVE_TABLE_NAME      STATUS
----------------------- ------------
TEST_TABLE1  HR         FLA1
SYS_FBA_HIST_92273      ENABLED

SQL>
```

c. Retrieve the transaction ID inserted into the flashback archive table. If the rows do not yet appear, truncate the `HR.TEST_TABLE1` table.

```
SQL> COL rid FORMAT A20
SQL> COL name FORMAT A20
SQL> truncate table hr.test_table1;

Table truncated.

SQL> select * from hr.SYS_FBA_HIST_92273;

RID                    STARTSCN     ENDSCN  XID              O
-------------------- ---------- ---------- ---------------- -
         NUM NAME                NOW
---------- -------------------- ---------
AAAXI4AAGAAAAD9AAB      3550155
           2 Second test row   10-SEP-12

AAAXI4AAGAAAAD9AAC      3550155
           3 Third test row    10-SEP-12

AAAXI4AAGAAAAD9AAA      3548223
           1 First test row     10-SEP-12

AAAXI4AAGAAAAD9AAA      3548223    3548739 08001100030D0000 U
           1 Premier test row  10-SEP-12

AAAXI4AAGAAAAD9AAA     3548739    3550155 0A000B0043090000 U
           1 Primero test row  10-SEP-12

SQL>
```

4. Collect the usernames of users who performed the `UPDATE` operations.

```
SQL> CONNECT / AS SYSDBA
Connected.
SQL> set pages 100
SQL> select DBMS_FLASHBACK_ARCHIVE.GET_SYS_CONTEXT
     ('08001100030D0000', 'USERENV','SESSION_USER') "User Name",
     VERSIONS_XID,VERSIONS_STARTTIME,VERSIONS_ENDTIME,
     num, name
     FROM hr.test_table1 VERSIONS BETWEEN scn minvalue
     AND maxvalue
    WHERE num=1;
  2   3   4   5   6   7
User Name
----------------------------------------------------------------
-
VERSIONS_XID
----------------
VERSIONS_STARTTIME
----------------------------------------------------------------
-
VERSIONS_ENDTIME
----------------------------------------------------------------
-
        NUM NAME
----------- -------------------
HR
0A000B0043090000
10-SEP-12 01.24.01.000000000 AM
10-SEP-12 01.24.14.000000000 AM
          1 Primero test row

HR

10-SEP-12 01.24.01.000000000 AM
          1 First test row

HR
08001100030D0000
10-SEP-12 01.24.01.000000000 AM
10-SEP-12 01.24.14.000000000 AM
          1 Premier test row

SQL>
```

5. Collect the module names of the users who performed the UPDATE operations.

```
SQL> select DBMS_FLASHBACK_ARCHIVE.GET_SYS_CONTEXT
     ('0A000B0043090000', 'USERENV','module') "Module Name",
     num, name
     FROM hr.test_table1 VERSIONS BETWEEN scn minvalue
     AND maxvalue
     WHERE num=1;
  2   3   4   5   6
Module Name
----------------------------------------------------------------
-
       NUM NAME
---------- -------------------------
SQL*Plus
     1 Primero test row

SQL*Plus
     1 First test row

SQL*Plus
     1 Premier test row

SQL> EXIT
$
```

Practice 8-4: Cleaning Up FDA

Overview

In this practice, you clean up the FDA tablespace.

Assumptions

You created the tablespace during Practice 8-3.

Tasks

1. Reconnect as SYSDBA to the de-active flashback archive on HR.TEST_TABLE1 table and drop the FDA_TBS tablespace.

```
$ sqlplus / as sysdba

Connected to:
Oracle Database 12c Enterprise Edition Release 12.1.0.0.2 -
64bit Production
With the Partitioning, OLAP, Advanced Analytics, Real
Application Testing and Unified Auditing options

SQL>
SQL> ALTER TABLE hr.test_table1  NO FLASHBACK ARCHIVE;

Table altered.

SQL> DROP FLASHBACK ARCHIVE fla1;

Flashback archive dropped.

SQL> DROP TABLESPACE fda_tbs INCLUDING CONTENTS AND DATAFILES;

Tablespace dropped.

SQL> EXIT
$
```

Practices for Lesson 9: Auditing

Chapter 9

Practices for Lesson 9

Practices Overview

In the practices for this lesson, you enable unified audit, configure for Data Pump export auditing, and audit export and RMAN operations. You then view the audited data in the `UNIFIED_AUDIT_TRAIL` view.

Practice 9-1: Enabling Unified Auditing

Overview

In this practice, you enable unified auditing.

Tasks

1. Shut down all Oracle processes of all instances
   a. Shut down the listener.

```
$ . oraenv
[ORACLE_SID = [orcl] ? orcl
The Oracle base remains unchanged with value /u01/app/oracle
$
```

```
$ lsnrctl stop


LSNRCTL for Linux: Version 12.1.0.0.2 - Production on 05-JUL-
2012 09:13:24


Copyright (c) 1991, 2012, Oracle.  All rights reserved.


Connecting to
(DESCRIPTION=(ADDRESS=(PROTOCOL=IPC)(KEY=EXTPROC1521)))
The command completed successfully
$
```

   b. Shut down all instances.

```
$ ps -ef | grep pmon
oracle    5111     1  0 Sep07 ?        00:00:31 ora_pmon_orcl
oracle   18211     1  0 Sep05 ?        00:00:53 ora_pmon_em12rep
oracle   25014     1  0 Sep07 ?        00:00:28 ora_pmon_cdb2
oracle   30114 29015  0 23:38 pts/3    00:00:00 grep pmon
$
```

      1) Shut down the `orcl` instance.

```
$ sqlplus / as sysdba


Connected to:
Oracle Database 12c Enterprise Edition Release 12.1.0.0.2 -
64bit Production
With the Partitioning, OLAP, Data Mining and Real Application
Testing options


SQL> shutdown immediate
Database closed.
Database dismounted.
```

```
ORACLE instance shut down.
SQL> EXIT
$
```

2) Shut down the cdb2 instance.

```
$ . oraenv
[ORACLE_SID = [orcl] ? cdb2
The Oracle base remains unchanged with value /u01/app/oracle
$
```

```
$ sqlplus / as sysdba

Connected to:
Oracle Database 12c Enterprise Edition Release 12.1.0.0.2 -
64bit Production
With the Partitioning, OLAP, Data Mining and Real Application
Testing options

SQL> shutdown immediate
Database closed.
Database dismounted.
ORACLE instance shut down.
SQL> EXIT
$
```

3) Shut down the em12rep instance.

a) Stop the OMS.

```
$ cd /u01/app/oracle/product/middleware/oms
$ export OMS_HOME=/u01/app/oracle/product/middleware/oms

$ $OMS_HOME/bin/emctl stop oms
Oracle Enterprise Manager Cloud Control 12c Release 2
Copyright (c) 1996, 2012 Oracle Corporation.  All rights
reserved.
Stopping WebTier...
WebTier Successfully Stopped
Stopping Oracle Management Server...
Oracle Management Server Successfully Stopped
Oracle Management Server is Down
$
```

b) Shut down the repository database instance em12rep.

```
$ . oraenv
[ORACLE_SID = [orcl] ? em12rep
The Oracle base remains unchanged with value /u01/app/oracle
```

```
$
```

```
$ sqlplus / as sysdba

Connected to:
Oracle Database 12c Enterprise Edition Release 12.1.0.0.2 -
64bit Production
With the Partitioning, OLAP, Data Mining and Real Application
Testing options

SQL> shutdown immediate
Database closed.
Database dismounted.
ORACLE instance shut down.
SQL> EXIT
$
```

4) Verify that all instances are down.

```
$ ps -ef | grep pmon
oracle    5165 13370  0 09:14 pts/0    00:00:00 grep pmon
$
```

2. Enable the Unified Audit option.

```
$ cd $ORACLE_HOME/rdbms/lib
$ make -f ins_rdbms.mk uniaud_on ioracle
ORACLE_HOME=$ORACLE_HOME
/usr/bin/ar d
/u01/app/oracle/product/12.1.0/dbhome_1/rdbms/lib/libknlopt.a
kzanang.o
/usr/bin/ar cr
/u01/app/oracle/product/12.1.0/dbhome_1/rdbms/lib/libknlopt.a
/u01/app/oracle/product/12.1.0/dbhome_1/rdbms/lib/kzaiang.o
chmod 755 /u01/app/oracle/product/12.1.0/dbhome_1/bin

 - Linking Oracle
rm -f /u01/app/oracle/product/12.1.0/dbhome_1/rdbms/lib/oracle
/u01/app/oracle/product/12.1.0/dbhome_1/bin/orald  -o
/u01/app/oracle/product/12.1.0/dbhome_1/rdbms/lib/oracle -m64 -z
noexecstack -L/u01/app/oracle/product/12.1.0/dbhome_1/rdbms/lib/
-L/u01/app/oracle/product/12.1.0/dbhome_1/lib/ -
L/u01/app/oracle/product/12.1.0/dbhome_1/lib/stubs/   -Wl,-E
/u01/app/oracle/product/12.1.0/dbhome_1/rdbms/lib/opimai.o
/u01/app/oracle/product/12.1.0/dbhome_1/rdbms/lib/ssoraed.o
/u01/app/oracle/product/12.1.0/dbhome_1/rdbms/lib/ttcsoi.o -Wl,-
-whole-archive -lperfsrv12 -Wl,--no-whole-archive
/u01/app/oracle/product/12.1.0/dbhome_1/lib/nautab.o
/u01/app/oracle/product/12.1.0/dbhome_1/lib/naeet.o
```

```
/u01/app/oracle/product/12.1.0/dbhome_1/lib/naect.o
/u01/app/oracle/product/12.1.0/dbhome_1/lib/naedhs.o
/u01/app/oracle/product/12.1.0/dbhome_1/rdbms/lib/config.o  -
lserver12 -lodm12 -lcell12 -lnnet12 -lskgxp12 -lsnls12 -lnls12
-lcore12 -lsnls12 -lnls12 -lcore12 -lsnls12 -lnls12 -lxml12 -
lcore12 -lunls12 -lsnls12 -lnls12 -lcore12 -lnls12 -lclient12  -
lvsn12 -lcommon12 -lgeneric12 -lknlopt `if /usr/bin/ar tv
/u01/app/oracle/product/12.1.0/dbhome_1/rdbms/lib/libknlopt.a |
grep xsyeolap.o > /dev/null 2>&1 ; then echo "-loraolap12" ; fi`
-lskjcx12 -lslax12 -lpls12  -lrt -lplp12 -lserver12 -lclient12
-lvsn12 -lcommon12 -lgeneric12 `if [ -f
/u01/app/oracle/product/12.1.0/dbhome_1/lib/libavserver12.a ] ;
then echo "-lavserver12" ; else echo "-lavstub12"; fi` `if [ -f
/u01/app/oracle/product/12.1.0/dbhome_1/lib/libavclient12.a ] ;
then echo "-lavclient12" ; fi` -lknlopt -lslax12 -lpls12  -lrt -
lplp12 -ljavavm12 -lserver12  -lwwg  `cat
/u01/app/oracle/product/12.1.0/dbhome_1/lib/ldflags`    -
lncrypt12 -lnsgr12 -lnzjs12 -ln12 -lnl12 -lnro12 `cat
/u01/app/oracle/product/12.1.0/dbhome_1/lib/ldflags`    -
lncrypt12 -lnsgr12 -lnzjs12 -ln12 -lnl12 -lnnz12 -lzt12 -lztkg12
-lmm -lsnls12 -lnls12  -lcore12 -lsnls12 -lnls12 -lcore12 -
lsnls12 -lnls12 -lxml12 -lcore12 -lunls12 -lsnls12 -lnls12 -
lcore12 -lnls12 -lztkg12 `cat
/u01/app/oracle/product/12.1.0/dbhome_1/lib/ldflags`    -
lncrypt12 -lnsgr12 -lnzjs12 -ln12 -lnl12 -lnro12 `cat
/u01/app/oracle/product/12.1.0/dbhome_1/lib/ldflags`    -
lncrypt12 -lnsgr12 -lnzjs12 -ln12 -lnl12 -lnnz12 -lzt12 -lztkg12
-lsnls12 -lnls12  -lcore12 -lsnls12 -lnls12 -lcore12 -lsnls12 -
lnls12 -lxml12 -lcore12 -lunls12 -lsnls12 -lnls12 -lcore12 -
lnls12 `if /usr/bin/ar tv
/u01/app/oracle/product/12.1.0/dbhome_1/rdbms/lib/libknlopt.a |
grep "kxmnsd.o" > /dev/null 2>&1 ; then echo " " ; else echo "-
lordsdo12"; fi` -
L/u01/app/oracle/product/12.1.0/dbhome_1/ctx/lib/ -lctxc12 -
lctx12 -lzx12 -lgx12 -lctx12 -lzx12 -lgx12 -lordimt12 -lclsra12
-ldbcfg12 -lhasgen12 -lskgxn2 -lnnz12 -lzt12 -lxml12 -locr12 -
locrb12 -locrutl12 -lhasgen12 -lskgxn2 -lnnz12 -lzt12 -lxml12  -
lgeneric12 -loraz -llzopro -lorabz2 -lipp_z -lipp_bz2 -
lippdcemerged -lippsemerged -lippdcmerged  -lippsmerged -
lippcore  -lippcpemerged -lippcpmerged  -lsnls12 -lnls12  -
lcore12 -lsnls12 -lnls12 -lcore12 -lsnls12 -lnls12 -lxml12 -
lcore12 -lunls12 -lsnls12 -lnls12 -lcore12 -lnls12 -lsnls12 -
lunls12  -lsnls12 -lnls12  -lcore12 -lsnls12 -lnls12 -lcore12 -
lsnls12 -lnls12 -lxml12 -lcore12 -lunls12 -lsnls12 -lnls12 -
lcore12 -lnls12 -lasmclnt12 -lcommon12 -lcore12  -laio -lons
`cat /u01/app/oracle/product/12.1.0/dbhome_1/lib/sysliblist` -
Wl,-rpath,/u01/app/oracle/product/12.1.0/dbhome_1/lib -lm
`cat /u01/app/oracle/product/12.1.0/dbhome_1/lib/sysliblist` -
ldl -lm   -L/u01/app/oracle/product/12.1.0/dbhome_1/lib
test ! -f /u01/app/oracle/product/12.1.0/dbhome_1/bin/oracle ||\
           mv -f
/u01/app/oracle/product/12.1.0/dbhome_1/bin/oracle
/u01/app/oracle/product/12.1.0/dbhome_1/bin/oracleO
mv /u01/app/oracle/product/12.1.0/dbhome_1/rdbms/lib/oracle
/u01/app/oracle/product/12.1.0/dbhome_1/bin/oracle
chmod 6751 /u01/app/oracle/product/12.1.0/dbhome_1/bin/oracle
$
```

3. Restart the processes.

   a. Restart the database `orcl` only.

```
$ . oraenv
[ORACLE_SID = [cdb2] ? orcl
The Oracle base remains unchanged with value /u01/app/oracle
$
```

```
$ sqlplus / as sysdba

Connected to an idle instance.

SQL> startup mount
ORACLE instance started.

Total System Global Area  847630336 bytes
Fixed Size                  2266072 bytes
Variable Size             557845544 bytes
Database Buffers          285212672 bytes
Redo Buffers                2306048 bytes
Database mounted.

SQL> ALTER DATABASE ARCHIVELOG;

Database altered.

SQL> ALTER DATABASE OPEN;

Database altered.

SQL> EXIT
$
```

You can see that the Unified Auditing option is enabled in the SQL*Plus banner.

   b. Restart the listener.

```
$ lsnrctl start
```

```
LSNRCTL for Linux: Version 12.1.0.0.2 - Production on 05-JUL-
2012 09:37:38


Copyright (c) 1991, 2012, Oracle.  All rights reserved.


Starting /u01/app/oracle/product/12.1.0/dbhome_1/bin/tnslsnr:
please wait...


TNSLSNR for Linux: Version 12.1.0.0.2 - Production
System parameter file is
/u01/app/oracle/product/12.1.0/dbhome_1/network/admin/listener.o
ra
Log messages written to
/u01/app/oracle/diag/tnslsnr/host01/listener/alert/log.xml
Listening on:
(DESCRIPTION=(ADDRESS=(PROTOCOL=ipc)(KEY=EXTPROC1521)))
Listening on:
(DESCRIPTION=(ADDRESS=(PROTOCOL=tcp)(HOST=host01.example.com)(PO
RT=1521)))


Connecting to
(DESCRIPTION=(ADDRESS=(PROTOCOL=IPC)(KEY=EXTPROC1521)))
STATUS of the LISTENER
------------------------
Alias                     LISTENER
Version                   TNSLSNR for Linux: Version 12.1.0.0.2
- Production
Start Date                05-JUL-2012 09:37:38
Uptime                    0 days 0 hr. 0 min. 0 sec
Trace Level               off
Security                  ON: Local OS Authentication
SNMP                      OFF
Listener Parameter File
/u01/app/oracle/product/12.1.0/dbhome_1/network/admin/listener.o
ra
Listener Log File
/u01/app/oracle/diag/tnslsnr/host01/listener/alert/log.xml
Listening Endpoints Summary...
  (DESCRIPTION=(ADDRESS=(PROTOCOL=ipc)(KEY=EXTPROC1521)))

(DESCRIPTION=(ADDRESS=(PROTOCOL=tcp)(HOST=host01.example.com)(PO
RT=1521)))
The listener supports no services
The command completed successfully
$
```

Practice 9-2: Auditing Data Pump Export

Overview

In this practice, you create an audit policy to audit Data Pump export operations. Then you will view the audited data after the export is completed.

Assumptions

Practice 9-1 successfully enabled unified audit.

Tasks

1. Create a DP_POLICY for the component Data Pump, and more specifically for export operations.

```
$ sqlplus system/oracle_4U

Connected to:
Oracle Database 12c Enterprise Edition Release 12.1.0.0.2 -
64bit Production
With the Partitioning, OLAP, Data Mining, Real Application
Testing
and Unified Auditing options

SQL> create audit policy DP_POL actions COMPONENT=datapump
export;

Audit policy created.

SQL>
```

2. Enable the audit policy.

```
SQL> audit policy DP_POL;

Audit succeeded.

SQL>
```

3. Verify that the policy exists.

```
SQL> col user_name format A10
SQL> col policy_name format A10
SQL> SELECT * FROM   AUDIT_UNIFIED_ENABLED_POLICIES
  2  where POLICY_NAME like '%DP%';

USER_NAME  POLICY_NAM ENABLED_ SUC FAI
---------- ---------- -------- --- ---
ALL USERS  DP_POL     BY       YES YES
```

```
SQL> EXIT
$
```

4. Perform an export operation. Before exporting, ensure that the dump file does not exist; else, the export command will fail.

```
$ rm /u01/app/oracle/admin/orcl/dpdump/HR_tables.dmp
rm: cannot remove
`/u01/app/oracle/admin/orcl/dpdump/HR_tables.dmp': No such file
or directory
$ expdp system/oracle_4U dumpfile=HR_tables tables=HR.EMPLOYEES


Connected to: Oracle Database 12c Enterprise Edition Release
12.1.0.0.2 - 64bit Production
With the Partitioning, OLAP, Data Mining, Real Application
Testing
and Unified Auditing options
Starting "SYSTEM"."SYS_EXPORT_TABLE_01":  system/********
dumpfile=HR_tables tables=HR.EMPLOYEES
Estimate in progress using BLOCKS method...
Processing object type TABLE_EXPORT/TABLE/TABLE_DATA
Total estimation using BLOCKS method: 64 KB
Processing object type TABLE_EXPORT/TABLE/TABLE
Processing object type
TABLE_EXPORT/TABLE/GRANT/OWNER_GRANT/OBJECT_GRANT
Processing object type TABLE_EXPORT/TABLE/COMMENT
Processing object type TABLE_EXPORT/TABLE/INDEX/INDEX
Processing object type TABLE_EXPORT/TABLE/CONSTRAINT/CONSTRAINT
Processing object type
TABLE_EXPORT/TABLE/INDEX/STATISTICS/INDEX_STATISTICS
Processing object type
TABLE_EXPORT/TABLE/CONSTRAINT/REF_CONSTRAINT
Processing object type TABLE_EXPORT/TABLE/TRIGGER
Processing object type
TABLE_EXPORT/TABLE/STATISTICS/TABLE_STATISTICS
Processing object type TABLE_EXPORT/TABLE/STATISTICS/MARKER
. . exported "HR"."EMPLOYEES"                           17.06
KB     107 rows
Master table "SYSTEM"."SYS_EXPORT_TABLE_01" successfully
loaded/unloaded
**************************************************************
Dump file set for SYSTEM.SYS_EXPORT_TABLE_01 is:
  /u01/app/oracle/admin/orcl/dpdump/HR_tables.dmp
Job "SYSTEM"."SYS_EXPORT_TABLE_01" successfully completed at Thu
Jul 5 10:08:37 2012 elapsed 0 00:00:23
```

```
$
```

5. View the resulting audit data. If the result shows no rows, then proceed with step 6, else you will get the result of step 7.

```
$ sqlplus system/oracle_4U

Connected to:
Oracle Database 12c Enterprise Edition Release 12.1.0.0.2 -
64bit Production
With the Partitioning, OLAP, Data Mining, Real Application
Testing and Unified Auditing options

SQL> select DBUSERNAME, DP_TEXT_PARAMETERS1,
  2         DP_BOOLEAN_PARAMETERS1
  3  from   UNIFIED_AUDIT_TRAIL
  4  where  DP_TEXT_PARAMETERS1 is not null;

no rows selected

SQL>
```

6. If the audited data is still in memory, flush the data to disk.

```
SQL> EXEC SYS.DBMS_AUDIT_MGMT.FLUSH_UNIFIED_AUDIT_TRAIL

PL/SQL procedure successfully completed.

SQL>
```

7. View the resulting audit data.

```
SQL> /

DBUSERNAME
------------------------------
DP_TEXT_PARAMETERS1
--------------------------------------------------------------
DP_BOOLEAN_PARAMETERS1
--------------------------------------------------------------
SYSTEM
MASTER TABLE:  "SYSTEM"."SYS_EXPORT_TABLE_01" , JOB_TYPE:
EXPORT, METADATA_JOB_M
ODE: TABLE_EXPORT, JOB VERSION: 12.0.0.0.0, ACCESS METHOD:
AUTOMATIC, DATA OPTIO
NS: 0, DUMPER DIRECTORY: NULL  REMOTE LINK: NULL, TABLE EXISTS:
NULL, PARTITION
OPTIONS: NONE
```

```
MASTER_ONLY: FALSE, DATA_ONLY: FALSE, METADATA_ONLY: FALSE,
DUMPFILE_PRESENT: TR
UE, JOB_RESTARTED: FALSE

SQL> exit
$
```

Practice 9-3: Auditing RMAN Backups

In this practice, you perform RMAN backups. Then you will view the audited data after RMAN backups are completed. You do not have to create any audit policy for RMAN operations. RMAN is by default audited.

Assumptions

Practice 9-1 successfully enabled unified audit.

Tasks

1. Perform a RMAN backup of the `USERS` tablespace.

```
$ rman target /

connected to target database: ORCL (DBID=1315477536)

RMAN> backup tablespace USERS;

Starting backup at 05-JUL-12
using target database control file instead of recovery catalog
allocated channel: ORA_DISK_1
channel ORA_DISK_1: SID=52 device type=DISK
channel ORA_DISK_1: starting full datafile backup set
channel ORA_DISK_1: specifying datafile(s) in backup set
input datafile file number=00006
name/u01/app/oracle/oradata/orcl/users01.dbf

channel ORA_DISK_1: starting piece 1 at 05-JUL-12
channel ORA_DISK_1: finished piece 1 at 05-JUL-12
piece
handle=/u01/app/oracle/fast_recovery_area/ORCL/backupset/2012_07
_05/o1_mf_nnndf_TAG20120705T102453_7zbtvp2x_.bkp
tag=TAG20120705T102453 comment=NONE
channel ORA_DISK_1: backup set complete, elapsed time: 00:00:01
Finished backup at 05-JUL-12

RMAN> exit;

Recovery Manager complete.
$
```

2. Perform a restore and recover after removing the `USERS` tablespace file.

   a. Find the data file name of the `USERS` tablespace and remove the file.

```
$ sqlplus / as sysdba

Connected to:
```

```
Oracle Database 12c Enterprise Edition Release 12.1.0.0.2 -
64bit Production
With the Partitioning, OLAP, Advanced Analytics, Real
Application Testing and Unified Auditing options

SQL> select name from v$datafile;

NAME
--------------------------------------------------------------
/u01/app/oracle/oradata/orcl/system01.dbf
/u01/app/oracle/oradata/orcl/example01.dbf
/u01/app/oracle/oradata/orcl/sysaux01.dbf
/u01/app/oracle/oradata/orcl/undotbs01.dbf
/u01/app/oracle/oradata/orcl/users01.dbf

SQL> !rm /u01/app/oracle/oradata/orcl/users01.dbf

SQL>
```

b. Put the tablespace OFFLINE.

```
SQL> alter tablespace users offline immediate;

Tablespace altered.

SQL> exit
$
```

c. Restore and recover the data file.

```
$ rman target /

connected to target database: ORCL (DBID=1315477536)

RMAN> restore tablespace USERS;

Starting restore at 05-JUL-12
using target database control file instead of recovery catalog
allocated channel: ORA_DISK_1
channel ORA_DISK_1: SID=59 device type=DISK

channel ORA_DISK_1: starting datafile backup set restore
channel ORA_DISK_1: specifying datafile(s) to restore from
backup set
channel ORA_DISK_1: restoring datafile 00006 to
/u01/app/oracle/oradata/orcl/users01.dbf
```

```
channel ORA_DISK_1: reading from backup piece
/u01/app/oracle/fast_recovery_area/ORCL/backupset/2012_07_05/o1_
mf_nnndf_TAG20120705T102453_7zbtvp2x_.bkp
channel ORA_DISK_1: piece
handle=/u01/app/oracle/fast_recovery_area/ORCL/backupset/2012_07
_05/o1_mf_nnndf_TAG20120705T102453_7zbtvp2x_.bkp
tag=TAG20120705T102453
channel ORA_DISK_1: restored backup piece 1
channel ORA_DISK_1: restore complete, elapsed time: 00:00:01
Finished restore at 05-JUL-12

RMAN> recover tablespace USERS;

Starting recover at 05-JUL-12
using channel ORA_DISK_1

starting media recovery
media recovery complete, elapsed time: 00:00:00

Finished recover at 05-JUL-12

RMAN> exit;
$
```

d. Put the tablespace USERS back online.

```
$ sqlplus system/oracle_4U

Connected to:
Oracle Database 12c Enterprise Edition Release 12.1.0.0.2 -
64bit Production
With the Partitioning, OLAP, Data Mining, Real Application
Testing and Unified Auditing options

SQL> alter tablespace USERS online;

Tablespace altered.

SQL>
```

3. View the resulting audit data. If the result shows no rows, then proceed with step 4, else you will get the result of step 5.

```
SQL> select DBUSERNAME, RMAN_OPERATION
  2  from   UNIFIED_AUDIT_TRAIL
  3  where  RMAN_OPERATION is not null;

no rows selected

SQL>
```

4. If the audited data is still in memory, flush the data to disk. But it is possible that the audit data is already flushed in the audit tables.

```
SQL> EXEC SYS.DBMS_AUDIT_MGMT.FLUSH_UNIFIED_AUDIT_TRAIL

PL/SQL procedure successfully completed.
```

5. View the resulting audit data.

```
SQL> /

DBUSERNAME               RMAN_OPERATION
------------------------ --------------------
SYS                      Recover
SYS                      Restore
SYS                      Backup

SQL> exit
$
```

Practices for Lesson 10: Privileges

Chapter 10

Practices for Lesson 10

Practices Overview

In the practices for this lesson, you use SYSBACKUP new administrative privilege and manage the password file, configure privilege capture for revoking privileges, and use the new INHERIT PRIVILEGES privilege.

Practice 10-1: Manage Password File with SYSBACKUP Entry

Overview

In this practice, you manage the password file with the new 12 format dedicated to new administrative privileges like SYSBACKUP.

Tasks

1. Make sure you are in the ~/labs/Security directory and your environment points to the orcl instance.

```
$ cd ~/labs/Security
$ . oraenv
ORACLE_SID = [orcl] ? orcl
The Oracle base remains unchanged with value /u01/app/oracle
$
```

2. Run the SYSBACKUP_setup.sh script to recreate the password file.

```
$ ./SYSBACKUP_setup.sh
$
```

3. Connect with OS authentication with AS SYSBACKUP and check the user connected.

```
$ sqlplus / as sysbackup

Connected to:
Oracle Database 12c Enterprise Edition Release 12.1.0.0.2 -
64bit Production
With the Partitioning, OLAP, Data Mining, Real Application
Testing and Unified Auditing options

SQL> show user
USER is "SYSBACKUP"

SQL>
```

4. List the privileges granted to SYSBACKUP user. Only a few privileges are granted to SYSBACKUP user. The SYSBACKUP privilege is granted to SYSBACKUP user.

```
SQL> select * from session_privs;

PRIVILEGE
----------------------------------------
SYSBACKUP
SELECT ANY TRANSACTION
SELECT ANY DICTIONARY
RESUMABLE
CREATE ANY DIRECTORY
ALTER DATABASE
```

```
AUDIT ANY
CREATE ANY CLUSTER
CREATE ANY TABLE
UNLIMITED TABLESPACE
DROP TABLESPACE
ALTER TABLESPACE
ALTER SESSION
ALTER SYSTEM

14 rows selected.

SQL>
```

5. Connect `AS SYSDBA` and list the privileges granted to `SYS` user. There are much more privileges granted to `SYS` user.

```
SQL> connect / as sysdba
Connected.
SQL> select * from session_privs;

PRIVILEGE
----------------------------------------
EXEMPT DDL REDACTION POLICY
EXEMPT DML REDACTION POLICY
LOGMINING
CREATE ANY CREDENTIAL
CREATE CREDENTIAL
SET CONTAINER
CLONE PLUGGABLE DATABASE
CREATE PLUGGABLE DATABASE
TRANSLATE ANY SQL
INHERIT ANY PRIVILEGES
EXEMPT REDACTION POLICY
FLASHBACK ARCHIVE ADMINISTER
PURGE DBA_RECYCLEBIN
EM EXPRESS CONNECT
KEEP SYSGUID
KEEP DATE TIME
ADMINISTER KEY MANAGEMENT
DROP ANY SQL TRANSLATION PROFILE
USE ANY SQL TRANSLATION PROFILE
ALTER ANY SQL TRANSLATION PROFILE
CREATE ANY SQL TRANSLATION PROFILE
CREATE SQL TRANSLATION PROFILE
```

```
ALTER DATABASE LINK
ALTER PUBLIC DATABASE LINK
ADMINISTER SQL MANAGEMENT OBJECT
UPDATE ANY CUBE DIMENSION
UPDATE ANY CUBE BUILD PROCESS
DROP ANY CUBE BUILD PROCESS
CREATE ANY CUBE BUILD PROCESS
CREATE CUBE BUILD PROCESS
INSERT ANY MEASURE FOLDER
DROP ANY MEASURE FOLDER
DELETE ANY MEASURE FOLDER
CREATE ANY MEASURE FOLDER
CREATE MEASURE FOLDER
UPDATE ANY CUBE
SELECT ANY CUBE
DROP ANY CUBE
CREATE ANY CUBE
ALTER ANY CUBE
CREATE CUBE
SELECT ANY CUBE DIMENSION
INSERT ANY CUBE DIMENSION
…
AUDIT SYSTEM
ALTER SYSTEM

233 rows selected.

SQL>
```

6. Display from the V$PWFILE_USERS view. SYS user is the only user defined in the password file with SYSDBA and SYSOPER privileges only. SYSBACKUP user is not registered in the password file.

```
SQL> select * from v$pwfile_users;

USERNAME SYSDB SYSOP SYSAS SYSBA SYSDG SYSKM CON_ID
-------- ----- ----- ----- ----- ----- ----- ------
SYS       TRUE  TRUE  FALSE FALSE FALSE FALSE      0

SQL>
```

7. Create a new user JOHN that will be granted the SYSBACKUP privilege in order to perform backup, restore, and recover operations, hence act as the SYSBACKUP user.

```
SQL> CREATE USER john IDENTIFIED BY oracle_4U;

User created.

SQL> GRANT create session, sysbackup TO john;
GRANT create session, sysbackup TO john
*
ERROR at line 1:
ORA-28017: The password file is in the legacy format.


SQL> exit
$
```

8. Because the password file had been created in legacy format, not compatible with the SYSBACKUP entry, it does not accept any SYSBACKUP entry.

   a. Recreate the file in 12 format, compatible with the SYSBACKUP entry.

```
$ cd $ORACLE_HOME/dbs
$ rm orapworcl
$ orapwd file=orapworcl password=oracle_4U entries=10 format=12
$
```

   b. Finally register JOHN in the password file.

```
$ sqlplus / as sysdba

Connected to:
Oracle Database 12c Enterprise Edition Release 12.1.0.0.2 -
64bit Production
With the Partitioning, OLAP, Data Mining, Real Application
Testing and Unified Auditing options
SQL> grant create session, SYSBACKUP to john;

Grant succeeded.

SQL> select * from v$pwfile_users;

USERNAME       SYSDB SYSOP SYSAS SYSBA SYSDG SYSKM     CON_ID
-------------- ----- ----- ----- ----- ----- ----- ----------
SYS             TRUE  TRUE  FALSE FALSE FALSE FALSE          0
JOHN           FALSE FALSE FALSE TRUE  FALSE FALSE          0

SQL>
```

c. Attempt a remote connection in SQL*Plus.

```
SQL> connect john/oracle_4U@orcl as SYSBACKUP
Connected.
SQL> SHOW USER
USER is "SYSBACKUP"
SQL> exit
$
```

d. Test the remote connection in RMAN.

```
$ rman target john/oracle_4U@orcl

Recovery Manager: Release 12.1.0.1.0 - Production on Mon Nov 26
06:28:43 2012

Copyright (c) 1982, 2012, Oracle and/or its affiliates.  All
rights reserved.

RMAN-00571: ===========================================================
RMAN-00569: =============== ERROR MESSAGE STACK FOLLOWS
RMAN-00571: ===========================================================
RMAN-00554: initialization of internal recovery manager package
failed
RMAN-04005: error from target database:
ORA-01031: insufficient privileges
$ rman target '"john/oracle_4U@orcl AS SYSBACKUP"'

connected to target database: ORCL (DBID=1327161403)

RMAN> select user from dual;

using target database control file instead of recovery catalog
USER
------------------------------
SYSBACKUP

RMAN> exit
$
```

Practice 10-2: Capturing Privileges

Overview

In this practice, you capture privileges used by users during a short period, generate the capture results, compare between used and unused privileges to decide which privileges might need to be revoked.

Tasks

1. Make sure you are at the `~/labs/Security` directory and your environment points to the `orcl` instance.

```
$ cd ~/labs/Security
$ . oraenv
ORACLE_SID = [orcl] ? orcl
The Oracle base remains unchanged with value /u01/app/oracle
$
```

2. Run the `priv_setup.sql` script to create `JIM` and `TOM` users, `HR_MGR` and `SALES_CLERK` roles.

```
$ sqlplus system/oracle_4U

Connected to:
Oracle Database 12c Enterprise Edition Release 12.1.0.0.2 -
64bit Production
With the Partitioning, OLAP, Data Mining, Real Application
Testing and Unified Auditing options

SQL> @priv_setup.sql
Connected.
drop user jim cascade
          *
ERROR at line 1:
ORA-01918: user 'JIM' does not exist



User created.

drop user tom  cascade
          *
ERROR at line 1:
ORA-01918: user 'TOM' does not exist
```

```
User created.


Grant succeeded.

drop  role HR_MGR
           *
ERROR at line 1:
ORA-01919: role 'HR_MGR' does not exist


drop role SALES_CLERK
          *
ERROR at line 1:
ORA-01919: role 'SALES_CLERK' does not exist


drop role HR_MGR_JUNIOR
          *
ERROR at line 1:
ORA-01919: role 'HR_MGR_JUNIOR' does not exist



Role created.


Grant succeeded.


Grant succeeded.


Role created.


Grant succeeded.


Grant succeeded.

revoke select any table from oe
```

```
*
ERROR at line 1:
ORA-01952: system privileges not granted to 'OE'


drop user u1 cascade
          *
ERROR at line 1:
ORA-01918: user 'U1' does not exist


drop user u2 cascade
          *
ERROR at line 1:
ORA-01918: user 'U2' does not exist


drop user kate
          *
ERROR at line 1:
ORA-01918: user 'KATE' does not exist



User created.


Grant succeeded.


Revoke succeeded.


User created.


Grant succeeded.


User created.
```

```
Grant succeeded.


Table created.


1 row created.


Commit complete.


Grant succeeded.
SQL>
```

3. Define a capture of privileges used by all users. Use the following procedure.

```
exec SYS.DBMS_PRIVILEGE_CAPTURE.CREATE_CAPTURE ( -
name           =>   'All_privs', -
description      =>  'All privs used', -
     type            =>   dbms_privilege_capture.g_database)
```

```
SQL> exec SYS.DBMS_PRIVILEGE_CAPTURE.CREATE_CAPTURE ( -
          name       =>   'All_privs', -
          description=>   'All privs used', -
           type       =>  dbms_privilege_capture.g_database)
> > >
PL/SQL procedure successfully completed.

SQL>
```

4. Start capturing the privileges while users are performing their daily work using privileges.
   a. Start the capture.

```
SQL> exec SYS.DBMS_PRIVILEGE_CAPTURE.ENABLE_CAPTURE (name =>
'All_privs')

PL/SQL procedure successfully completed.

SQL>
```

   b. Run the `priv_used_by_users.sql` script. The script connects as `JIM` who deletes rows from `HR.EMPLOYEES` table and `TOM` who selects rows from `SH.SALES` table.

```
SQL> @priv_used_by_users.sql
Connected.

24 rows deleted.

Commit complete.

Connected.

PROD_ID CUST_ID TIME_ID   CHANNEL_ID PROMO_ID QUANTITY_SOLD
AMOUNT_SOLD
------- ------- --------- ---------- -------- ------------- ----
------------
    120    6452 29-SEP-00          2      999             1
        6.4
    120    6452 29-SEP-00          4      999             1
        6.4

SQL>
```

5. Stop the capture.

```
SQL> connect / as sysdba
Connected.
SQL> exec SYS.DBMS_PRIVILEGE_CAPTURE.DISABLE_CAPTURE (name
=> 'All_privs')

PL/SQL procedure successfully completed.

SQL>
```

6. Generate the capture results. It may take a few minutes.

```
SQL> exec SYS.DBMS_PRIVILEGE_CAPTURE.GENERATE_RESULT (name =>
'All_privs')

PL/SQL procedure successfully completed.
SQL>
```

7. Display the object privileges used during the capture period.

```
SQL> COL username FORMAT A10
SQL> COL object_owner FORMAT A12
SQL> COL object_name FORMAT A30
SQL> COL obj_priv FORMAT A25
SQL> SELECT  username, object_owner, object_name, obj_priv
     FROM  dba_used_objprivs
     WHERE username IN ('JIM', 'TOM');
  2   3
USERNAME   OBJECT_OWNER OBJECT_NAME                    OBJ_PRIV
---------- ------------ ------------------------------ -------
JIM        SYS          DUAL                           SELECT
JIM        SYSTEM       PRODUCT_PRIVS                  SELECT
TOM        SYS          ORA$BASE                       USE
TOM        SYSTEM       PRODUCT_PRIVS                  SELECT
JIM        SYS          DBMS_APPLICATION_INFO          EXECUTE
JIM        SYS          ORA$BASE                       USE
TOM        SYS          DUAL                           SELECT
TOM        SH           SALES                          SELECT
JIM        HR           EMPLOYEES                      DELETE
TOM        SYS          DBMS_APPLICATION_INFO          EXECUTE
JIM        SYS          DUAL                           SELECT
TOM        SYS          DUAL                           SELECT

12 rows selected.

SQL>
```

8. Display the system privileges used.

```
SQL> SELECT username, sys_priv FROM dba_used_sysprivs
     WHERE username IN ('JIM', 'TOM');
   2
USERNAME   SYS_PRIV
---------- ----------------------------------------
TOM        CREATE SESSION
JIM        CREATE SESSION
```

```
SQL>
```

9. Display the path of the privileges used if the privileges were granted to roles, and roles to users.

```
SQL> COL object FORMAT A12
SQL> COL path FORMAT A32
SQL> COL obj_priv FORMAT A10
SQL> SELECT username, obj_priv, object_name, path
     FROM   dba_used_objprivs_path
     WHERE  username IN ('TOM','JIM')
     AND object_name IN ('SALES','EMPLOYEES');
  2   3   4
USERNAME   OBJ_PRIV   OBJECT       PATH
---------- ---------- ------------ ---------------------------
TOM        SELECT     SALES        GRANT_PATH('TOM',
'SALES_CLERK')

JIM        DELETE     EMPLOYEES    GRANT_PATH('JIM', 'HR_MGR')

SQL>
```

10. JIM is granted select, update, delete, insert privileges on HR.EMPLOYEES table through HR_MGR role. He used only the DELETE privilege until now.
The unused privileges are visible in DBA_UNUSED_PRIVS view.

```
SQL> SELECT username, sys_priv, obj_priv, object_name, path
     FROM   dba_unused_privs
     WHERE  username='JIM';
  2   3
USERNAME   SYS_PRIV   OBJ_PRIV   OBJECT       PATH
---------- ---------- ---------- ------------ ------------------
------------
JIM                   SELECT     EMPLOYEES    GRANT_PATH('JIM',
'HR_MGR')
JIM                   INSERT     EMPLOYEES    GRANT_PATH('JIM',
'HR_MGR')
JIM                   UPDATE     EMPLOYEES    GRANT_PATH('JIM',
'HR_MGR')

SQL>
```

11. Compare used and unused privileges. Finally you decide to revoke the INSERT privilege from JIM, but not impact other users who benefit from the HR_MGR role.

    a. You will first create a new role without the INSERT privilege and finally revoke the HR_MGR role from JIM.

```
SQL> create role HR_MGR_JUNIOR;

Role created.

SQL> GRANT select, update, delete ON hr.employees
  2  TO hr_mgr_junior;

Grant succeeded.

SQL>
```

    b. Grant the new role to JIM.

```
SQL> grant HR_MGR_JUNIOR to JIM;

Grant succeeded.

SQL>
```

    c. Finally revoke the powerful privileged role HR_MGR from JIM.

```
SQL> revoke HR_MGR from JIM;

Revoke succeeded.

SQL>
```

12. Display the definition of the capture. The ENABLED column that the All_privs capture has been stopped.

```
SQL> COL name FORMAT A12
SQL> COL type FORMAT A12
SQL> COL enabled FORMAT A2
SQL> COL roles FORMAT A26
SQL> SELECT name, type, enabled,roles, context
  2  FROM dba_priv_captures;

NAME         TYPE         EN ROLES
------------ ------------ -- --------------------------
CONTEXT
----------------------------------------------------------------------
All_privs    DATABASE     N

SQL>
```

13. Delete the capture so as to remove all previous captured information from the views.

    a. Execute the procedure.

```
SQL> exec SYS.DBMS_PRIVILEGE_CAPTURE.DROP_CAPTURE (name=>
'All_privs')

PL/SQL procedure successfully completed.

SQL>
```

    b. Verify that there is no data left of the `All_privs` capture.

```
SQL> SELECT username, sys_priv, obj_priv, object_name, path
  2  FROM   dba_unused_privs
  3  WHERE  username='JIM';

no rows selected

SQL>
```

Practice 10-3: Capture Privileges Used Through Roles *(Optional)*

Overview

In this practice, you capture the privileges used by roles during a short period, generate the capture results, compare between used and unused privileges to decide which privileges might need to be revoked.

Tasks

1. Define a capture of privileges used by roles `HR_MGR_JUNIOR` and `SALES_CLERK`. Use the following procedure.

```
exec SYS.DBMS_PRIVILEGE_CAPTURE.CREATE_CAPTURE ( -
name            => 'Role_privs', -
description     => 'Privs used by HR_MGR_JUNIOR, SALES_CLERK', -
type            => dbms_privilege_capture.g_role, -
roles      => role_name_list('HR_MGR_JUNIOR', 'SALES_CLERK'))
```

```
SQL> exec SYS.DBMS_PRIVILEGE_CAPTURE.CREATE_CAPTURE ( -
name   => 'Role_privs', -
description => 'Privs used by HR_MGR_JUNIOR, SALES_CLERK', -
type        => dbms_privilege_capture.g_role, -
roles => role_name_list('HR_MGR_JUNIOR', 'SALES_CLERK'))
> > > >
PL/SQL procedure successfully completed.

SQL>
```

2. Start capturing the privileges while users perform their daily work.
   a. Start the capture.

```
SQL> exec SYS.DBMS_PRIVILEGE_CAPTURE.ENABLE_CAPTURE (name =>
'Role_privs')

PL/SQL procedure successfully completed.

SQL>
```

   b. Run the `priv_used_by_users.sql` script. The script connects as `JIM` who deletes rows from `HR.EMPLOYEES` table and `TOM` who selects rows from `SH.SALES` table.

```
SQL> @priv_used_by_users.sql
Connected.

0 rows deleted.

Commit complete.

Connected.
```

```
PROD_ID CUST_ID TIME_ID   CHANNEL_ID PROMO_ID QUANTITY_SOLD
AMOUNT_SOLD
------- ------- --------- ---------- -------- ------------- ----
------------
    120    6452 29-SEP-00          2      999             1
        6.4
    120    6452 29-SEP-00          4      999             1
        6.4

SQL>
```

3. Stop the capture.

```
SQL> connect / as sysdba
Connected.
SQL> exec SYS.DBMS_PRIVILEGE_CAPTURE.DISABLE_CAPTURE (name
=> 'Role_privs')

PL/SQL procedure successfully completed.

SQL>
```

4. Generate the capture results.

```
SQL> exec SYS.DBMS_PRIVILEGE_CAPTURE.GENERATE_RESULT (name =>
'Role_privs')

PL/SQL procedure successfully completed.

SQL>
```

5. Display the object privileges used by the roles HR_MGR_JUNIOR and SALES_CLERK during the capture period.

```
SQL> COL username FORMAT A10
SQL> COL owner FORMAT A8
SQL> COL object FORMAT A16
SQL> COL obj_priv FORMAT A10
SQL> COL used_role FORMAT A14
SQL> SELECT  username, object_owner, object_name, obj_priv,
used_role
     FROM  dba_used_objprivs
     WHERE used_role IN ('HR_MGR_JUNIOR', 'SALES_CLERK');
  2   3
USERNAME   OWNER    OBJECT           OBJ_PRIV   USED_ROLE
---------- -------- ---------------- ---------- --------------
JIM        HR       EMPLOYEES        DELETE     HR_MGR_JUNIOR
TOM        SH       SALES            SELECT     SALES_CLERK
```

```

SQL>
```

6. Display the system privileges used by the roles HR_MGR_JUNIOR and SALES_CLERK.

```
SQL> SELECT username, sys_priv, used_role
     FROM dba_used_sysprivs
     WHERE used_role IN ('HR_MGR_JUNIOR', 'SALES_CLERK');
  2   3
no rows selected

SQL>
```

7. HR_MGR_JUNIOR is granted select, update, delete on HR.EMPLOYEES table. The role used by JIM during the capture period used the DELETE privilege until now.
The unused privileges are visible in DBA_UNUSED_PRIVS view.

```
SQL> COL username FORMAT A12
SQL> COL path FORMAT A32
SQL> COL object FORMAT A10
SQL> COL sys_priv FORMAT A10
SQL> COL obj_priv FORMAT A10
SQL> SELECT sys_priv, obj_priv, object_name, path
     FROM   dba_unused_privs
     WHERE  rolename IN ('HR_MGR_JUNIOR', 'SALES_CLERK');
  2   3
SYS_PRIV   OBJ_PRIV   OBJECT     PATH
---------- ---------- ---------- ----------------------------
           SELECT     EMPLOYEES  GRANT_PATH('HR_MGR_JUNIOR')
           UPDATE     EMPLOYEES  GRANT_PATH('HR_MGR_JUNIOR')

SQL>
```

View the list of unused privileges: this list helps you decide whether to revoke or not the UPDATE and SELECT privileges granted through the HR_MGR_JUNIOR role.

8. Display the definition of the capture. The ENABLED column shows that the Role_privs capture has been stopped.

```
SQL> COL name FORMAT A12
SQL> COL type FORMAT A12
SQL> COL enabled FORMAT A2
SQL> COL roles FORMAT A26
SQL> SELECT name, type, enabled,roles, context
     FROM dba_priv_captures;
  2
NAME         TYPE         EN ROLES
------------ ------------ -- --------------------------
CONTEXT
```

```
--------------------------------------------------------------
Role_privs   ROLE         N  ROLE_ID_LIST(119, 115)

SQL>
```

9. Delete the capture so as to remove all previous captured information from the views.

   a. Execute the procedure.

```
SQL> exec SYS.DBMS_PRIVILEGE_CAPTURE.DROP_CAPTURE (name=>
'Role_privs')

PL/SQL procedure successfully completed.

SQL>
```

   b. Verify that there is no data left of the `Role_privs` capture.

```
SQL> SELECT sys_priv, obj_priv, object_name, path
     FROM   dba_unused_privs
     WHERE  rolename IN ('HR_MGR_JUNIOR', 'SALES_CLERK');
  2   3
no rows selected

SQL>
```

Practice 10-4: Capture Privileges Used In Contexts *(Optional)*

Overview

In this practice, you capture privileges used by the user TOM or by the specific role SALES_CLERK during a short period, generate the capture results, compare between used and unused privileges to decide which privileges might need to be revoked.

Tasks

1. Define a capture of privileges used by the user TOM or by the specific role SALES_CLERK. Use the following procedure.

```
exec SYS.DBMS_PRIVILEGE_CAPTURE.CREATE_CAPTURE ( -
name      => 'Special_capt', -
description  => 'Special', -
type      =>  dbms_privilege_capture.g_role_and_context, -
roles     =>   role_name_list('SALES_CLERK'), -
condition =>
'SYS_CONTEXT(''USERENV'',''SESSION_USER'')=''TOM''')
```

```
SQL> exec SYS.DBMS_PRIVILEGE_CAPTURE.CREATE_CAPTURE ( -
name        => 'Special_capt', -
description => 'Special', -
type    =>  dbms_privilege_capture.g_role_and_context, -
roles   =>  role_name_list('SALES_CLERK'), -
condition =>
'SYS_CONTEXT(''USERENV'',''SESSION_USER'')=''TOM''')
> > > > >
PL/SQL procedure successfully completed.

SQL>
```

2. Start capturing privileges while users perform their daily work using the privileges.
   a. Start the capture.

```
SQL> exec SYS.DBMS_PRIVILEGE_CAPTURE.ENABLE_CAPTURE (name =>
'Special_capt')

PL/SQL procedure successfully completed.

SQL>
```

   b. Run the priv_used_by_users.sql script. The script connects as JIM who deletes rows from HR.EMPLOYEES table and TOM who selects rows from SH.SALES table.

```
SQL> @priv_used_by_users.sql
Connected.

0 rows deleted.
```

```
Commit complete.

Connected.

PROD_ID CUST_ID TIME_ID   CHANNEL_ID PROMO_ID QUANTITY_SOLD
AMOUNT_SOLD
------- ------- --------- ---------- -------- ------------ ----
------------
    120    6452 29-SEP-00          2      999            1
        6.4
    120    6452 29-SEP-00          4      999            1
        6.4

SQL>
```

3. Stop the capture.

```
SQL> connect / as sysdba
Connected.
SQL> exec SYS.DBMS_PRIVILEGE_CAPTURE.DISABLE_CAPTURE (name
=> 'Special_capt')

PL/SQL procedure successfully completed.

SQL>
```

4. Generate the capture results. It may take a few minutes.

```
SQL> exec SYS.DBMS_PRIVILEGE_CAPTURE.GENERATE_RESULT (name =>
'Special_capt')

PL/SQL procedure successfully completed.

SQL>
```

5. Display the object privileges used.

```
SQL> COL username FORMAT A10
SQL> COL owner FORMAT A8
SQL> COL object FORMAT A16
SQL> COL obj_priv FORMAT A10
SQL> COL used_role FORMAT A14
SQL> SELECT  username, object_owner, object_name, obj_priv,
used_role
     FROM  dba_used_objprivs
     WHERE username ='TOM' OR used_role='SALES_CLERK';
  2   3
USERNAME   OWNER    OBJECT           OBJ_PRIV   USED_ROLE
---------- -------- ---------------- ---------- --------------
```

```
TOM          SH        SALES             SELECT     SALES_CLERK


SQL>
```

6. Display the system privileges used.

```
SQL> SELECT username, sys_priv FROM dba_used_sysprivs;


no rows selected


SQL>
```

7. TOM is granted the select privilege on the SH.SALES table through SALES_CLERK role. He used the privilege.
The unused privs are visible in DBA_UNUSED_PRIVS view.
There are not any unused privileges. So there is no privilege that has been unnecessarily granted.

```
SQL> SELECT username, sys_priv, obj_priv, object_name, path
     FROM   dba_unused_privs
     WHERE  username='TOM' OR rolename='SALES_CLERK';
  2   3
no rows selected


SQL>
```

8. Delete the capture so as to remove all previous captured information from the views.

```
SQL> exec SYS.DBMS_PRIVILEGE_CAPTURE.DROP_CAPTURE (name=>
'Special_capt')


PL/SQL procedure successfully completed.


SQL>
```

Practice 10-5: Use INHERIT PRIVILEGES Privilege

Overview

In this practice you will use the new INHERIT PRIVILEGES privilege when creating invoker's rights procedures.

Assumptions

The priv_setup.sql script has been successfully executed at the beginning of Practice 10-1.

Tasks

1. The developer U1 creates an invoker's rights procedure that selects rows from U2.T1 table. The user U1 is granted the SELECT privilege on U2.T1 table

   a. Connect as user U1.

```
SQL> connect u1/u1
Connected.
SQL>
```

   b. Create the U1.PROC2 procedure.

```
CREATE OR REPLACE PROCEDURE u1.proc2 (CODE in varchar2)
AUTHID CURRENT_USER AS
v_code  number;
BEGIN
SELECT code INTO v_code FROM u2.t1;
dbms_output.put_line('Code is: '||v_code);
END PROC2;
/
```

```
SQL> CREATE OR REPLACE PROCEDURE u1.proc2 (CODE in varchar2)
AUTHID CURRENT_USER AS
v_code  number;
BEGIN
SELECT code INTO v_code FROM u2.t1;
dbms_output.put_line('Code is: '||v_code);
END PROC2;
/
  2    3    4    5    6    7    8
Procedure created.

SQL>
```

   c. Execute the procedure to test that it works successfully.

```
SQL> set serveroutput on
SQL> exec U1.PROC2('Code')
Code is: 1

PL/SQL procedure successfully completed.
```

```
SQL>
```

d. The developer U1 grants the EXECUTE privilege to the KATE user.

```
SQL> grant execute on U1.PROC2 to KATE;

Grant succeeded.

SQL>
```

2. KATE wants to test the procedure.

a. KATE has no privilege on U2.T1 table. KATE connects and executes the procedure.

```
SQL> CONNECT kate/kate
Connected.
SQL> set serveroutput on
SQL> exec U1.PROC2('Code')
SQL> BEGIN U1.PROC2('Code'); END;

*
ERROR at line 1:
ORA-06598: insufficient INHERIT PRIVILEGES privilege
ORA-06512: at "U1.PROC2", line 1
ORA-06512: at line 1

SQL>
```

b. KATE grants the INHERIT PRIVILEGES on user KATE to procedure owner U1 thus allowing U1 to inherit her privileges during the execution of the procedure

```
SQL> grant INHERIT PRIVILEGES ON USER kate TO U1;

Grant succeeded.

SQL>
```

c. KATE re-executes the procedure.

```
SQL> set serveroutput on
SQL> exec U1.PROC2('Code')
Code is: 1

PL/SQL procedure successfully completed.

SQL>
```

3. Display the users being granted the INHERIT PRIVILEGES privilege. There is a new object type 'USER' and the table name is the user name controlling who can access his privileges when he runs an invoker's rights procedure.

```
SQL> connect / as sysdba
Connected.

SQL> COL privilege FORMAT A20
SQL> COL type FORMAT A6
SQL> COL table_name FORMAT A10
SQL> COL grantee FORMAT A8
SQL> select PRIVILEGE, TYPE, TABLE_NAME, GRANTEE
  2  from DBA_TAB_PRIVS  where grantee='U1';

PRIVILEGE            TYPE   TABLE_NAME GRANTEE
-------------------- ------ ---------- --------
SELECT               TABLE  T1         U1
INHERIT PRIVILEGES   USER   KATE       U1

SQL>
```

4. Be aware that newly created users are granted the `INHERIT PRIVILEGES` privilege because the `INHERIT PRIVILEGES` privilege is granted to `PUBLIC`. The user `KATE` was revoked the `INHERIT PRIVILEGES` privilege at the beginning of the practice.
   a. Create a new user.

```
SQL> CREATE USER newuser IDENTIFIED BY newuser;

User created.

SQL>
```

   b. Check the privileges granted to NEWUSER.

```
SQL> select PRIVILEGE, TYPE, TABLE_NAME, GRANTEE
     from DBA_TAB_PRIVS
     where type='USER'
     and     table_name='NEWUSER';
  2   3   4
PRIVILEGE            TYPE   TABLE_NAME GRANTEE
-------------------- ------ ---------- --------
INHERIT PRIVILEGES   USER   NEWUSER    PUBLIC

SQL> EXIT
$
```

Practice 10-6: `INHERIT PRIVILEGES` Privilege and `BEQUEATH` Views

Overview

In this practice you understand the different types of `BEQUEATH` views: the `CURRENT_USER` and the `DEFINER` views.

Assumptions

The `bequeath_setup.sql` script is successfully completed.

Tasks

1. Make sure you are at the `~/labs/Security` directory and your environment points to the `orcl` instance. Connect under `SYSTEM` user.

```
$ cd ~/labs/Security
$ . oraenv
ORACLE_SID = [orcl] ? orcl
The Oracle base remains unchanged with value /u01/app/oracle
$
```

2. Execute the `bequeath_setup.sql` script. The script creates users and grants appropriate privileges to the developer `U1` and the end user `KATE`.

```
$ sqlplus SYSTEM/oracle_4U

Connected to:
Oracle Database 12c Enterprise Edition Release 12.1.0.0.2 -
64bit Production
With the Partitioning, OLAP, Advanced Analytics, Real
Application Testing and Unified Auditing options

SQL>
SQL> @bequeath_setup.sql
Connected.
REVOKE select any table from OE
*
ERROR at line 1:
ORA-01952: system privileges not granted to 'OE'

User dropped.


User dropped.


User dropped.


User created.
```

```
Grant succeeded.

Revoke succeeded.

User created.

Grant succeeded.

SQL>
```

3. The developer `U1` creates a `BEQUEATH CURRENT_USER` view. The view displays the current user connected.

   a. The user `U1` connects and creates the view `V_WHOAMI`.

```
SQL> CONNECT u1/u1
Connected.
SQL> CREATE OR REPLACE VIEW u1.v_whoami
     BEQUEATH CURRENT_USER
     AS SELECT ORA_INVOKING_USER "WHOAMI" FROM DUAL;
  2   3
View created.

SQL>
```

   b. The developer checks that the view `V_WHOAMI` works successfully.

```
SQL> select * from U1.V_WHOAMI;

WHOAMI
----------------------------------------------------------
U1

SQL>
```

4. The same developer `U1` creates an `BEQUEATH DEFINER` view. The view displays the current user connected.

   a. The user `U1` connects and creates the view `V_WHOAMI_DEF`.

```
SQL> CREATE OR REPLACE VIEW u1.v_whoami_def
     BEQUEATH DEFINER
     AS SELECT ORA_INVOKING_USER "WHOAMI" FROM DUAL;
  2   3
View created.

SQL>
```

b. The developer checks that the view V_WHOAMI_DEF works successfully.

```
SQL> select * from U1.V_WHOAMI_DEF;

WHOAMI
---------------------------------------------------------
U1

SQL>
```

5. The developer U1 grants the SELECT privilege to KATE on both views.

```
SQL> grant SELECT  on U1.V_WHOAMI to KATE;

Grant succeeded.

SQL> grant SELECT  on U1.V_WHOAMI_DEF to KATE;

Grant succeeded.

SQL>
```

6. KATE connects and selects data from the BEQUEATH DEFINER view.

```
SQL> CONNECT kate/kate
Connected.
SQL> select * from U1.V_WHOAMI_DEF;

WHOAMI
--------------------------------------------------------
KATE

SQL>
```

7. KATE selects data from the BEQUEATH CURRENT_USER view.

```
SQL> SELECT * FROM U1.V_WHOAMI;
select * from U1.V_WHOAMI
                 *
ERROR at line 1:
ORA-06598: insufficient INHERIT PRIVILEGES privilege

SQL>
```

8. KATE grants the INHERIT PRIVILEGES ON USER KATE to the view owner U1, allowing U1 to use her privileges during the view execution.

```
SQL> grant INHERIT PRIVILEGES ON USER kate TO U1;


Grant succeeded.

SQL>
```

9. KATE attempts the statement on the BEQUEATH CURRENT_USER view.

```
SQL> select * from U1.V_WHOAMI;


WHOAMI
----------------------------------------------------------
KATE

SQL> EXIT
$
```

Practices for Lesson 11: Oracle Data Redaction

Chapter 11

Practices for Lesson 11

Practices Overview

In the practice for this lesson, you use Oracle Data Redaction to redact values of shielded columns of the HR.EMPLOYEES table.

Practice 11-1: Redacting Protected Column Values with FULL Redaction

Overview

In this practice you use FULL data redaction to display the SALARY column values from the HR.EMPLOYEES as 0 instead of the real values.

Tasks

1. Display the current values from the HR.EMPLOYEES table before redaction.

```
$ . oraenv
ORACLE_SID = [orcl] ? orcl
The Oracle base for
ORACLE_HOME=/u01/app/oracle/product/12.1.0/dbhome_1 is
/u01/app/oracle
$ sqlplus system/oracle_4U

Connected to:
Oracle Database 12c Enterprise Edition Release 12.1.0.0.2 -
64bit Production
With the Partitioning, OLAP, Advanced Analytics, Real
Application Testing and Unified Auditing options

SQL> SELECT employee_id, last_name, salary
     FROM hr.employees
     WHERE department_id = 100;
  2   3
EMPLOYEE_ID LAST_NAME                     SALARY
----------- ------------------------- ----------
        108 Greenberg                      12008
        109 Faviet                          9000
        110 Chen                            8200
        111 Sciarra                         7700
        112 Urman                           7800
        113 Popp                            6900

6 rows selected.

SQL>
```

2. Define a masking policy for the HR.EMPLOYEES table specifying full masking for the SALARY column. SALARY is defined as NUMBER(8,2). In this example, by setting EXPRESSION to 1=1, redaction is always performed because the expression always evaluates to true.
The policy is enabled by default.

```
BEGIN
```

```
  DBMS_REDACT.ADD_POLICY
  (object_schema => 'HR',
   object_name   => 'EMPLOYEES',
   policy_name   => 'EMPSAL_POLICY',
   column_name   => 'SALARY',
   function_type => DBMS_REDACT.FULL,
   expression => '1=1');
  END;
 /
```

```
SQL> BEGIN
  DBMS_REDACT.ADD_POLICY
  (object_schema => 'HR',
   object_name   => 'EMPLOYEES',
   policy_name   => 'EMPSAL_POLICY',
   column_name   => 'SALARY',
   function_type => DBMS_REDACT.FULL,
   expression => '1=1');
  END;
  2    3    4    5    6    7    8    9   10
 11  /

PL/SQL procedure successfully completed.

SQL>
```

3. Query REDACTION_POLICIES to verify that the policy has been created and is enabled. This view also shows under what condition the redaction will be performed as shown in the EXPRESSION column.

```
SQL> COL object_owner FORMAT A12
SQL> COL object_name FORMAT A12
SQL> COL policy_name FORMAT A14
SQL> COL expression FORMAT A12
SQL> COL enable FORMAT A6
SQL> COL policy_description FORMAT A10
SQL> SELECT * FROM redaction_policies;

OBJECT_OWNER OBJECT_NAME  POLICY_NAME    EXPRESSION   ENABLE
------------ ------------ -------------- ------------ ------
POLICY_DES
----------
HR           EMPLOYEES    EMPSAL_POLICY  1=1          YES

```

```
SQL>
```

4. Display which columns will be redacted and what type of redaction will take place.

```
SQL> COL column_name FORMAT A10
SQL> COL function_typ FORMAT A14
SQL> COL function_parameters FORMAT A20
SQL> SELECT object_owner, object_name, column_name,
     function_type, function_parameters
     FROM redaction_columns;
  2   3
OBJECT_OWNER OBJECT_NAME      COLUMN_NAM FUNCTION_TYPE
------------ ---------------- ---------- --------------------
FUNCTION_PARAMETERS
--------------------
HR           EMPLOYEES        SALARY     FULL REDACTION

SQL>
```

5. Now query the HR.EMPLOYEES table again and note that the value of the SALARY column is 0 for all displayed rows.

   a. First grant the SELECT privilege to SH.

```
SQL> GRANT select ON hr.employees TO sh;

Grant succeeded.

SQL>
```

   b. Connect as SH. If SH is locked, unlock the account.

```
SQL> ALTER USER sh IDENTIFIED BY oracle_4U ACCOUNT UNLOCK;

User altered.

SQL> CONNECT sh/oracle_4U
Connected.
SQL>
```

   c. Run the same select as in task 1.

```
SQL> SELECT employee_id, last_name, salary
     FROM hr.employees
     WHERE department_id = 100;
  2   3
EMPLOYEE_ID LAST_NAME                     SALARY
----------- ------------------------- ----------
        108 Greenberg                          0
        109 Faviet                             0
        110 Chen                               0
        111 Sciarra                            0
        112 Urman                              0
        113 Popp                               0

6 rows selected.

SQL>
```

6. If you query as `SYSDBA`, the “real” value is displayed, not the redacted value as shown in this example. Any user who is granted the `EXEMPT REDACTION POLICY` privilege bypasses any redaction policy.

   a. Connect as `SYSDBA`.

```
SQL> CONNECT / AS SYSDBA
Connected.
SQL>
```

   b. Run the same select as in task 1.

```
SQL> /

EMPLOYEE_ID LAST_NAME                     SALARY
----------- ------------------------- ----------
        108 Greenberg                      12008
        109 Faviet                          9000
        110 Chen                            8200
        111 Sciarra                         7700
        112 Urman                           7800
        113 Popp                            6900

6 rows selected.

SQL>
```

Practice 11-2: Redacting Protected Column Values with PARTIAL Redaction (*optional*)

Overview

In this practice, you use PARTIAL data redaction to display the HIRE_DATE column values from the HR.EMPLOYEES as a partially redacted value instead of the real values.

Tasks

1. Query the HR.EMPLOYEES table again and display the HIRE_DATE column.

```
SQL> SELECT employee_id, last_name, hire_date
     FROM hr.employees
     WHERE department_id = 100;
  2   3
EMPLOYEE_ID LAST_NAME                 HIRE_DATE
----------- ------------------------- ---------
        108 Greenberg                 17-AUG-02
        109 Faviet                    16-AUG-02
        110 Chen                      28-SEP-05
        111 Sciarra                   30-SEP-05
        112 Urman                     07-MAR-06
        113 Popp                      07-DEC-07

6 rows selected.

SQL>
```

2. Alter the masking policy to redact the HIRE_DATE column. In this example, partial redaction is used to mask the actual year of hire.

```
BEGIN
  DBMS_REDACT.ALTER_POLICY
  (object_schema => 'HR',
  object_name   => 'EMPLOYEES',
  policy_name   => 'EMPSAL_POLICY',
  action        => DBMS_REDACT.ADD_COLUMN,
  column_name   => 'HIRE_DATE',
  function_type => DBMS_REDACT.PARTIAL,
  function_parameters=> 'MDy2012',
 expression => '1=1');
 END;
 /
```

```
SQL> BEGIN
     DBMS_REDACT.ALTER_POLICY
     (object_schema => 'HR',
      object_name   => 'EMPLOYEES',
      policy_name   => 'EMPSAL_POLICY',
      action        => DBMS_REDACT.ADD_COLUMN,
      column_name   => 'HIRE_DATE',
      function_type => DBMS_REDACT.PARTIAL,
      function_parameters=> 'MDy2012',
      expression => '1=1');
     END;
     /
  2   3   4   5   6   7   8   9  10  11  12
PL/SQL procedure successfully completed.

SQL>
```

3. Query REDACTION_POLICIES to verify that the policy has been created and is enabled. This view also shows under what condition the redaction will be performed as shown in the EXPRESSION column.

```
SQL> select * from redaction_policies;

OBJECT_OWNER OBJECT_NAME  POLICY_NAME       EXPRESSION ENABLE
------------ ------------ ----------------- ---------- ------
POLICY_DES
----------
HR           EMPLOYEES    EMPSAL_POLICY     1=1         YES

SQL>
```

4. The REDACTION_COLUMNS view shows both masking functions defined on the HR.EMPLOYEES table.

```
SQL> SELECT object_owner, object_name, column_name,
     function_type, function_parameters
     FROM redaction_columns;
  2   3
OBJECT_OWNER OBJECT_NAME  COLUMN_NAM FUNCTION_TYPE
------------ ------------ ---------- ------------------
FUNCTION_PARAMETERS
--------------------
HR           EMPLOYEES    SALARY     FULL REDACTION

HR           EMPLOYEES    HIRE_DATE  PARTIAL REDACTION
MDy2012

SQL>
```

5. Query HR.EMPLOYEES again as the SH user. ’12’ is displayed as the hire year for all the rows selected.

```
SQL> CONNECT sh/oracle_4U
Connected.

SQL> select employee_id, last_name, hire_date
     from hr.employees
     where department_id = 100;
  2   3
EMPLOYEE_ID LAST_NAME                 HIRE_DATE
----------- ------------------------- ---------
        108 Greenberg                 17-AUG-12
        109 Faviet                    16-AUG-12
        110 Chen                      28-SEP-12
        111 Sciarra                   30-SEP-12
        112 Urman                     07-MAR-12
        113 Popp                      07-DEC-12

6 rows selected.

SQL>
```

Practice 11-3: Cleaning Up Redaction Policies

Overview

In this practice you clean up the redaction policy applied on the HR.EMPLOYEES table.

1. Drop the redaction policy as SYSTEM user.

```
BEGIN
  DBMS_REDACT.DROP_POLICY
  (object_schema => 'HR',
  object_name   => 'EMPLOYEES',
  policy_name   => 'EMPSAL_POLICY');
  END;
/
```

```
SQL> CONNECT system/oracle_4U
Connected.
SQL> BEGIN
     DBMS_REDACT.DROP_POLICY
     (object_schema => 'HR',
      object_name   => 'EMPLOYEES',
      policy_name   => 'EMPSAL_POLICY');
     END;
    /
  2  3  4  5  6  7
PL/SQL procedure successfully completed.

SQL>
```

2. Check that the values for the `SALARY` and `HIRE_DATE` columns are displayed without redaction.

```
SQL> select employee_id, last_name, salary, hire_date
     from hr.employees
     where department_id = 100;
  2    3
EMPLOYEE_ID LAST_NAME                     SALARY HIRE_DATE
----------- ------------------------- ---------- ---------
        108 Greenberg                      12008 17-AUG-02
        109 Faviet                          9000 16-AUG-02
        110 Chen                            8200 28-SEP-05
        111 Sciarra                         7700 30-SEP-05
        112 Urman                           7800 07-MAR-06
        113 Popp                            6900 07-DEC-07

6 rows selected.

SQL> EXIT
$
```

Practices for Lesson 12: Recovery Manager New Features and Temporal History Enhancements

Chapter 12

Practices for Lesson 12

Practices Overview

In the practice for this lesson, you use SYSBACKUP connection to RMAN and perform RMAN commands. You use table recovery to recover the HR.EMPLOYEES table instead of using Tablespace Point-In-Time Recovery (TSPITR) of the EXAMPLE tablespace or flashing back the table.

Practice 12-1: Using SYSBACKUP in RMAN

Overview

In this practice, you connect to RMAN and execute several SQL and SQL*Plus commands. You will find out that some commands are not allowed to SYSBACKUP due to restricted privileges.

Tasks

1. Connect to RMAN.

```
$ . oraenv
[ORACLE_SID = [orcl] ? orcl
The Oracle base remains unchanged with value /u01/app/oracle
$ rman TARGET '"/ AS SYSBACKUP"'

Recovery Manager: Release 12.1.0.0.2 - Production on Mon Jul 9
14:34:13 2012

Copyright (c) 1982, 2012, Oracle and/or its affiliates.  All
rights reserved.

connected to target database: ORCL (DBID=1315953682)

RMAN> select user from dual;

using target database control file instead of recovery catalog
USER
------------------------------
SYSBACKUP

RMAN>
```

2. Execute a backup command to back up the whole database and the archive logs.

```
RMAN> CONFIGURE CONTROLFILE AUTOBACKUP ON;
RMAN> BACKUP DATABASE PLUS ARCHIVELOG;

Starting backup at 10-09-2012 01:06:37
current log archived
allocated channel: ORA_DISK_1
channel ORA_DISK_1: SID=274 device type=DISK
channel ORA_DISK_1: starting archived log backup set
channel ORA_DISK_1: specifying archived log(s) in backup set
input archived log thread=1 sequence=23 RECID=1 STAMP=793291556
input archived log thread=1 sequence=24 RECID=2 STAMP=793291556
input archived log thread=1 sequence=25 RECID=3 STAMP=793314040
input archived log thread=1 sequence=26 RECID=4 STAMP=793317891
input archived log thread=1 sequence=27 RECID=5 STAMP=793318314
input archived log thread=1 sequence=28 RECID=6 STAMP=793318704
input archived log thread=1 sequence=29 RECID=7 STAMP=793318955
input archived log thread=1 sequence=30 RECID=8 STAMP=793319345
input archived log thread=1 sequence=31 RECID=9 STAMP=793319735
input archived log thread=1 sequence=32 RECID=10 STAMP=793320125
input archived log thread=1 sequence=33 RECID=11 STAMP=793320513
input archived log thread=1 sequence=34 RECID=12 STAMP=793320900
input archived log thread=1 sequence=35 RECID=13 STAMP=793321271
input archived log thread=1 sequence=36 RECID=14 STAMP=793321648
input archived log thread=1 sequence=37 RECID=15 STAMP=793322034
input archived log thread=1 sequence=38 RECID=16 STAMP=793322420
input archived log thread=1 sequence=39 RECID=17 STAMP=793327717
input archived log thread=1 sequence=40 RECID=18 STAMP=793327977
input archived log thread=1 sequence=41 RECID=19 STAMP=793328218
input archived log thread=1 sequence=42 RECID=20 STAMP=793328337
input archived log thread=1 sequence=43 RECID=21 STAMP=793328933
input archived log thread=1 sequence=44 RECID=22 STAMP=793328940
input archived log thread=1 sequence=45 RECID=23 STAMP=793342851
input archived log thread=1 sequence=46 RECID=24 STAMP=793378856
input archived log thread=1 sequence=47 RECID=25 STAMP=793404978
input archived log thread=1 sequence=48 RECID=26 STAMP=793414892
input archived log thread=1 sequence=49 RECID=27 STAMP=793433084
input archived log thread=1 sequence=50 RECID=28 STAMP=793447359
input archived log thread=1 sequence=51 RECID=29 STAMP=793461726
input archived log thread=1 sequence=52 RECID=30 STAMP=793476131
input archived log thread=1 sequence=53 RECID=31 STAMP=793492662
input archived log thread=1 sequence=54 RECID=32 STAMP=793519428
input archived log thread=1 sequence=55 RECID=33 STAMP=793533910
input archived log thread=1 sequence=56 RECID=34 STAMP=793548323
input archived log thread=1 sequence=57 RECID=35 STAMP=793562742
input archived log thread=1 sequence=58 RECID=36 STAMP=793577207
input archived log thread=1 sequence=59 RECID=37 STAMP=793587998
channel ORA_DISK_1: starting piece 1 at 10-09-2012 01:06:40
channel ORA_DISK_1: finished piece 1 at 10-09-2012 01:07:35
piece
handle=/u01/app/oracle/fast_recovery_area/ORCL/backupset/2012_09
_10/o1_mf_annnn_TAG20120910T010639_84th903m_.bkp
tag=TAG20120910T010639 comment=NONE
channel ORA_DISK_1: backup set complete, elapsed time: 00:00:55
Finished backup at 10-09-2012 01:07:35


Starting backup at 10-09-2012 01:07:35
```

```
using channel ORA_DISK_1
channel ORA_DISK_1: starting full datafile backup set
channel ORA_DISK_1: specifying datafile(s) in backup set
input datafile file number=00003
name=/u01/app/oracle/oradata/orcl/sysaux01.dbf
input datafile file number=00001
name=/u01/app/oracle/oradata/orcl/system01.dbf
input datafile file number=00002
name=/u01/app/oracle/oradata/orcl/example01.dbf
input datafile file number=00004
name=/u01/app/oracle/oradata/orcl/undotbs01.dbf
input datafile file number=00006
name=/u01/app/oracle/oradata/orcl/users01.dbf
channel ORA_DISK_1: starting piece 1 at 10-09-2012 01:07:35
channel ORA_DISK_1: finished piece 1 at 10-09-2012 01:08:30
piece
handle=/u01/app/oracle/fast_recovery_area/ORCL/backupset/2012_09
_10/o1_mf_nnndf_TAG20120910T010735_84thbr78_.bkp
tag=TAG20120910T010735 comment=NONE
channel ORA_DISK_1: backup set complete, elapsed time: 00:00:55
Finished backup at 10-09-2012 01:08:30

Starting backup at 10-09-2012 01:08:30
current log archived
using channel ORA_DISK_1
channel ORA_DISK_1: starting archived log backup set
channel ORA_DISK_1: specifying archived log(s) in backup set
input archived log thread=1 sequence=60 RECID=38 STAMP=793588111
channel ORA_DISK_1: starting piece 1 at 10-09-2012 01:08:31
channel ORA_DISK_1: finished piece 1 at 10-09-2012 01:08:32
piece
handle=/u01/app/oracle/fast_recovery_area/ORCL/backupset/2012_09
_10/o1_mf_annnn_TAG20120910T010831_84thdhnl_.bkp
tag=TAG20120910T010831 comment=NONE
channel ORA_DISK_1: backup set complete, elapsed time: 00:00:01
Finished backup at 10-09-2012 01:08:32

Starting Control File and SPFILE Autobackup at 10-09-2012
01:08:32
piece
handle=/u01/app/oracle/fast_recovery_area/ORCL/autobackup/2012_0
9_10/o1_mf_s_793588112_84thdkwt_.bkp comment=NONE
Finished Control File and SPFILE Autobackup at 10-09-2012
01:08:35

RMAN>
```

3. Execute the REPORT SCHEMA command.

```
RMAN> REPORT SCHEMA;

Report of database schema for database with db_unique_name ORCL

List of Permanent Datafiles
===========================
File Size(MB) Tablespace           RB segs Datafile Name
---- -------- -------------------- ------- ---------------------
-
1    800      SYSTEM               ***
/u01/app/oracle/oradata/orcl/system01.dbf
2    500      EXAMPLE              ***
/u01/app/oracle/oradata/orcl/example01.dbf
3    970      SYSAUX               ***
/u01/app/oracle/oradata/orcl/sysaux01.dbf
4    385      UNDOTBS1             ***
/u01/app/oracle/oradata/orcl/undotbs01.dbf
6    5        USERS                ***
/u01/app/oracle/oradata/orcl/users01.dbf

List of Temporary Files
=======================
File Size(MB) Tablespace           Maxsize(MB) Tempfile Name
---- -------- -------------------- ----------- -----------------
-
1    198      TEMP                 32767
/u01/app/oracle/oradata/orcl/temp01.dbf

RMAN>
```

4. Execute a SELECT command on an application table. The SYSBACKUP user is not granted any object privilege on any application object, or the SELECT ANY TABLE system privilege.

```
RMAN> SELECT * FROM hr.employees;
using target database control file instead of recovery catalog
RMAN-00571: ==================================================
RMAN-00569: =============== ERROR MESSAGE STACK FOLLOWS ======
RMAN-00571: ==================================================
RMAN-03002: failure of sql statement command at 07/09/2012
14:40:31
ORA-01031: insufficient privileges

RMAN>
```

5. Execute a `SELECT` command on a dictionary table. The `SYSBACKUP` user is granted the system privilege `SELECT ANY DICTIONARY`.
   a. Select from `DBA_TABLES`.

```
RMAN> SELECT owner  FROM DBA_TABLES WHERE table_name='JOBS';

OWNER
-------------------------------------------------------------
HR

RMAN>
```

   b. Select from `V$DATABASE`.

```
RMAN> SELECT log_mode FROM v$database;


using target database control file instead of recovery catalog
LOG_MODE
------------
ARCHIVELOG

RMAN>
```

6. Execute the `DESCRIBE` SQL*Plus command.

```
RMAN> DESC v$database

 Name                                   Null?   Type
 --------------------------------------- ------- -----------
 DBID                                           NUMBER
 NAME                                           VARCHAR2(9)
 CREATED                                        DATE
 RESETLOGS_CHANGE#                              NUMBER
 RESETLOGS_TIME                                 DATE
 PRIOR_RESETLOGS_CHANGE#                        NUMBER
 PRIOR_RESETLOGS_TIME                           DATE
 LOG_MODE                                       VARCHAR2(12)
 CHECKPOINT_CHANGE#                             NUMBER
 ARCHIVE_CHANGE#                                NUMBER
 CONTROLFILE_TYPE                               VARCHAR2(7)
 CONTROLFILE_CREATED                            DATE
 CONTROLFILE_SEQUENCE#                          NUMBER
 CONTROLFILE_CHANGE#                            NUMBER
 CONTROLFILE_TIME                               DATE
 OPEN_RESETLOGS                                 VARCHAR2(11)
```

```
 VERSION_TIME                                   DATE
 OPEN_MODE                                      VARCHAR2(20)
 PROTECTION_MODE                                VARCHAR2(20)
 PROTECTION_LEVEL                                VARCHAR2(20)
 REMOTE_ARCHIVE                                  VARCHAR2(8)
 ACTIVATION#                                     NUMBER
 SWITCHOVER#                                     NUMBER
 DATABASE_ROLE                                   VARCHAR2(16)
 ARCHIVELOG_CHANGE#                              NUMBER
 ARCHIVELOG_COMPRESSION                          VARCHAR2(8)
 SWITCHOVER_STATUS                               VARCHAR2(20)
 DATAGUARD_BROKER                                VARCHAR2(8)
 GUARD_STATUS                                    VARCHAR2(7)
 SUPPLEMENTAL_LOG_DATA_MIN                       VARCHAR2(8)
 SUPPLEMENTAL_LOG_DATA_PK                        VARCHAR2(3)
 SUPPLEMENTAL_LOG_DATA_UI                        VARCHAR2(3)
 FORCE_LOGGING                                   VARCHAR2(39)
 PLATFORM_ID                                     NUMBER
 PLATFORM_NAME                                   VARCHAR2(101)
 RECOVERY_TARGET_INCARNATION#                    NUMBER
 LAST_OPEN_INCARNATION#                          NUMBER
 CURRENT_SCN                                     NUMBER
 FLASHBACK_ON                                    VARCHAR2(18)
 SUPPLEMENTAL_LOG_DATA_FK                        VARCHAR2(3)
 SUPPLEMENTAL_LOG_DATA_ALL                       VARCHAR2(3)
 DB_UNIQUE_NAME                                  VARCHAR2(30)
 STANDBY_BECAME_PRIMARY_SCN                      NUMBER
 FS_FAILOVER_STATUS                              VARCHAR2(22)
 FS_FAILOVER_CURRENT_TARGET                      VARCHAR2(30)
 FS_FAILOVER_THRESHOLD                           NUMBER
 FS_FAILOVER_OBSERVER_PRESENT                    VARCHAR2(7)
 FS_FAILOVER_OBSERVER_HOST                       VARCHAR2(512)
 CONTROLFILE_CONVERTED                           VARCHAR2(3)
 PRIMARY_DB_UNIQUE_NAME                          VARCHAR2(30)
 SUPPLEMENTAL_LOG_DATA_PL                        VARCHAR2(3)
 MIN_REQUIRED_CAPTURE_CHANGE#                    NUMBER
 CDB                                             VARCHAR2(3)
 CON_ID                                          NUMBER
 PENDING_ROLE_CHANGE_TASKS                       VARCHAR2(512)
 CON_DBID                                        NUMBER

RMAN> EXIT

Recovery Manager complete.
$
```

Practice 12-2: Recovering a Table by Using Table Recovery

In this practice, you perform a table recovery of HR.TEST_TABLE1 after having inadvertently purged the table.

Assumption

The whole database backup has been successfully completed in the previous practice in task 2.

Tasks

1. Create a new HR.TEST table and insert a few rows.

```
$ sqlplus / as sysdba

Connected to:
Oracle Database 12c Enterprise Edition Release 12.1.0.0.2 -
64bit Production
With the Partitioning, OLAP, Advanced Analytics, Real
Application Testing and Unified Auditing options

SQL> CREATE TABLE hr.test_table1
     (NUM number(8), NAME varchar2(25), NOW  date ) ;
  2
Table created.

SQL> INSERT INTO hr.test_table1 VALUES (1,'First test
row',sysdate);

1 row created.

SQL> INSERT INTO hr.test_table1 VALUES (2,'Second test
row',sysdate);

1 row created.

SQL> INSERT INTO hr.test_table1 VALUES (3,'Third test
row',NULL);

1 row created.

SQL> COMMIT;

Commit complete.

SQL> EXIT
$
```

2. Back up the tablespace where the table is stored.

```
$ rman TARGET /

connected to target database: ORCL (DBID=1319927350)

RMAN> backup tablespace users;

Starting backup at 10-09-2012 01:13:27
using target database control file instead of recovery catalog
allocated channel: ORA_DISK_1
channel ORA_DISK_1: SID=37 device type=DISK
channel ORA_DISK_1: starting full datafile backup set
channel ORA_DISK_1: specifying datafile(s) in backup set
input datafile file number=00006
name=/u01/app/oracle/oradata/orcl/users01.dbf
channel ORA_DISK_1: starting piece 1 at 10-09-2012 01:13:27
channel ORA_DISK_1: finished piece 1 at 10-09-2012 01:13:28
piece
handle=/u01/app/oracle/fast_recovery_area/ORCL/backupset/2012_09
_10/o1_mf_nnndf_TAG20120910T011327_84thoqyr_.bkp
tag=TAG20120910T011327 comment=NONE
channel ORA_DISK_1: backup set complete, elapsed time: 00:00:01
Finished backup at 10-09-2012 01:13:28

Starting Control File and SPFILE Autobackup at 10-09-2012
01:13:28
piece
handle=/u01/app/oracle/fast_recovery_area/ORCL/autobackup/2012_0
9_10/o1_mf_s_793588409_84thospn_.bkp comment=NONE
Finished Control File and SPFILE Autobackup at 10-09-2012
01:13:31

RMAN> exit
$
```

3. You inadvertently purge the table.

   a. Select the current sysdate. This date will help you recover the table back to the time when the table was purged.

```
$ sqlplus / as sysdba

Connected to:
Oracle Database 12c Enterprise Edition Release 12.1.0.0.2 -
64bit Production
With the Partitioning, OLAP, Advanced Analytics and Real
Application Testing options
```

```
SQL> select timestamp_to_scn(current_timestamp)
  2  from v$database;

TIMESTAMP_TO_SCN(CURRENT_TIMESTAMP)
-----------------------------------
                            2634118
SQL>
```

b. Purge the table.

```
SQL> DROP TABLE hr.test_table1 PURGE;

Table dropped.

SQL> exit
$
```

4. Perform the table recovery as the `SYS` user using `/tmp/backup_test` as the auxiliary destination using your own SCN number retrieved in step 3.a.

```
$ mkdir /u01/app/oracle/backup_test
$ rman TARGET /

connected to target database: ORCL (DBID=1315953682)

RMAN> RECOVER TABLE hr.test_table1 UNTIL SCN 2634118 AUXILIARY
DESTINATION '/u01/app/oracle/backup_test';



Starting recover at 10-09-2012 01:14:51
using target database control file instead of recovery catalog
current log archived
allocated channel: ORA_DISK_1
channel ORA_DISK_1: SID=37 device type=DISK
RMAN-05026: WARNING: presuming following set of tablespaces
applies to specified Point-in-Time

List of tablespaces expected to have UNDO segments
Tablespace SYSTEM
Tablespace UNDOTBS1

Creating automatic instance, with SID='zrik'

initialization parameters used for automatic instance:
db_name=ORCL
```

```
db_unique_name=zrik_pitr_ORCL
compatible=12.0.0.0.0
db_block_size=8192
db_files=200
sga_target=1G
processes=80
db_create_file_dest=/u01/app/oracle/backup_test
log_archive_dest_1='location=/u01/app/oracle/backup_test'
#No auxiliary parameter file used


starting up automatic instance ORCL

Oracle instance started

Total System Global Area    1068937216 bytes

Fixed Size                     2279272 bytes
Variable Size                281020568 bytes
Database Buffers             780140544 bytes
Redo Buffers                   5496832 bytes
Automatic instance created

contents of Memory Script:
{
# set requested point in time
set until  scn 2634118;
# restore the controlfile
restore clone controlfile;
# mount the controlfile
sql clone 'alter database mount clone database';
# archive current online log
sql 'alter system archive log current';
}
executing Memory Script

executing command: SET until clause

Starting restore at 10-09-2012 01:15:50
allocated channel: ORA_AUX_DISK_1
channel ORA_AUX_DISK_1: SID=72 device type=DISK
channel ORA_AUX_DISK_1: starting datafile backup set restore
channel ORA_AUX_DISK_1: restoring control file
channel ORA_AUX_DISK_1: reading from backup piece
/u01/app/oracle/fast_recovery_area/ORCL/autobackup/2012_09_10/o1
_mf_s_793588409_84thospn_.bkp
channel ORA_AUX_DISK_1: piece
handle=/u01/app/oracle/fast_recovery_area/ORCL/autobackup/2012_0
9_10/o1_mf_s_793588409_84thospn_.bkp tag=TAG20120910T011328
channel ORA_AUX_DISK_1: restored backup piece 1
channel ORA_AUX_DISK_1: restore complete, elapsed time: 00:00:01
output file
name=/u01/app/oracle/backup_test/ORCL/controlfile/o1_mf_84tht7co
_.ctl
Finished restore at 10-09-2012 01:15:52

sql statement: alter database mount clone database

sql statement: alter system archive log current

contents of Memory Script:
{
# set requested point in time
set until  scn 2634118;
# set destinations for recovery set and auxiliary set datafiles
set newname for clone datafile  1 to new;
set newname for clone datafile  4 to new;
set newname for clone datafile  3 to new;
set newname for clone tempfile  1 to new;
# switch all tempfiles
switch clone tempfile all;
# restore the tablespaces in the recovery set and the auxiliary
set
restore clone datafile  1, 4, 3;
switch clone datafile all;
}
executing Memory Script

executing command: SET until clause

executing command: SET NEWNAME

executing command: SET NEWNAME

executing command: SET NEWNAME
executing command: SET NEWNAME

renamed tempfile 1 to
/u01/app/oracle/backup_test/ORCL/datafile/o1_mf_temp_%u_.tmp in
control file

Starting restore at 10-09-2012 01:15:58
using channel ORA_AUX_DISK_1

channel ORA_AUX_DISK_1: starting datafile backup set restore
channel ORA_AUX_DISK_1: specifying datafile(s) to restore from
backup set
channel ORA_AUX_DISK_1: restoring datafile 00001 to
/u01/app/oracle/backup_test/ORCL/datafile/o1_mf_system_%u_.dbf
channel ORA_AUX_DISK_1: restoring datafile 00004 to
/u01/app/oracle/backup_test/ORCL/datafile/o1_mf_undotbs1_%u_.dbf
channel ORA_AUX_DISK_1: restoring datafile 00003 to
/u01/app/oracle/backup_test/ORCL/datafile/o1_mf_sysaux_%u_.dbf
channel ORA_AUX_DISK_1: reading from backup piece
/u01/app/oracle/fast_recovery_area/ORCL/backupset/2012_09_10/o1_
mf_nnndf_TAG20120910T010735_84thbr78_.bkp
channel ORA_AUX_DISK_1: piece
handle=/u01/app/oracle/fast_recovery_area/ORCL/backupset/2012_09
_10/o1_mf_nnndf_TAG20120910T010735_84thbr78_.bkp
tag=TAG20120910T010735
channel ORA_AUX_DISK_1: restored backup piece 1
channel ORA_AUX_DISK_1: restore complete, elapsed time: 00:00:45
Finished restore at 10-09-2012 01:16:44

datafile 1 switched to datafile copy
input datafile copy RECID=4 STAMP=793588605 file
name=/u01/app/oracle/backup_test/ORCL/datafile/o1_mf_system_84th
th32_.dbf
datafile 4 switched to datafile copy
input datafile copy RECID=5 STAMP=793588605 file
name=/u01/app/oracle/backup_test/ORCL/datafile/o1_mf_undotbs1_84
thth4f_.dbf
datafile 3 switched to datafile copy
input datafile copy RECID=6 STAMP=793588605 file
name=/u01/app/oracle/backup_test/ORCL/datafile/o1_mf_sysaux_84th
th08_.dbf

contents of Memory Script:
{
# set requested point in time
set until  scn 2634118;
# online the datafiles restored or switched
sql clone "alter database datafile  1 online";
sql clone "alter database datafile  4 online";
sql clone "alter database datafile  3 online";
# recover and open database read only
recover clone database tablespace  "SYSTEM", "UNDOTBS1",
"SYSAUX";
sql clone 'alter database open read only';
}
executing Memory Script

executing command: SET until clause

sql statement: alter database datafile  1 online

sql statement: alter database datafile  4 online

sql statement: alter database datafile  3 online

Starting recover at 10-09-2012 01:16:46
using channel ORA_AUX_DISK_1

starting media recovery

archived log for thread 1 with sequence 60 is already on disk as
file
/u01/app/oracle/fast_recovery_area/ORCL/archivelog/2012_09_10/o1
_mf_1_60_84thdh45_.arc
archived log for thread 1 with sequence 61 is already on disk as
file
/u01/app/oracle/fast_recovery_area/ORCL/archivelog/2012_09_10/o1
_mf_1_61_84thrcth_.arc
archived log file
name=/u01/app/oracle/fast_recovery_area/ORCL/archivelog/2012_09_
10/o1_mf_1_60_84thdh45_.arc thread=1 sequence=60
archived log file
name=/u01/app/oracle/fast_recovery_area/ORCL/archivelog/2012_09_
10/o1_mf_1_61_84thrcth_.arc thread=1 sequence=61
media recovery complete, elapsed time: 00:00:01
Finished recover at 10-09-2012 01:16:49

sql statement: alter database open read only
```

```
contents of Memory Script:
{
   sql clone "create spfile from memory";
   shutdown clone immediate;
   startup clone nomount;
   sql clone "alter system set  control_files =

''/u01/app/oracle/backup_test/ORCL/controlfile/o1_mf_84tht7co_.c
tl'' comment=
 ''RMAN set'' scope=spfile";
   shutdown clone immediate;
   startup clone nomount;
# mount database
sql clone 'alter database mount clone database';
}
executing Memory Script

sql statement: create spfile from memory

database closed
database dismounted
Oracle instance shut down

connected to auxiliary database (not started)
Oracle instance started

Total System Global Area    1068937216 bytes

Fixed Size                     2279272 bytes
Variable Size                285214872 bytes
Database Buffers             775946240 bytes
Redo Buffers                   5496832 bytes

sql statement: alter system set  control_files =
''/u01/app/oracle/backup_test/ORCL/controlfile/o1_mf_84tht7co_.c
tl'' comment= ''RMAN set'' scope=spfile

Oracle instance shut down

connected to auxiliary database (not started)
Oracle instance started

Total System Global Area    1068937216 bytes
```

```
Fixed Size                     2279272 bytes
Variable Size                285214872 bytes
Database Buffers             775946240 bytes
Redo Buffers                   5496832 bytes

sql statement: alter database mount clone database

contents of Memory Script:
{
# set requested point in time
set until  scn 2634118;
# set destinations for recovery set and auxiliary set datafiles
set newname for datafile  6 to new;
# restore the tablespaces in the recovery set and the auxiliary
set
restore clone datafile  6;
switch clone datafile all;
}
executing Memory Script

executing command: SET until clause

executing command: SET NEWNAME

Starting restore at 10-09-2012 01:17:26
allocated channel: ORA_AUX_DISK_1
channel ORA_AUX_DISK_1: SID=83 device type=DISK

channel ORA_AUX_DISK_1: starting datafile backup set restore
channel ORA_AUX_DISK_1: specifying datafile(s) to restore from
backup set
channel ORA_AUX_DISK_1: restoring datafile 00006 to
/u01/app/oracle/backup_test/ZRIK_PITR_ORCL/datafile/o1_mf_users_
%u_.dbf
channel ORA_AUX_DISK_1: reading from backup piece
/u01/app/oracle/fast_recovery_area/ORCL/backupset/2012_09_10/o1_
mf_nnndf_TAG20120910T011327_84thoqyr_.bkp
channel ORA_AUX_DISK_1: piece
handle=/u01/app/oracle/fast_recovery_area/ORCL/backupset/2012_09
_10/o1_mf_nnndf_TAG20120910T011327_84thoqyr_.bkp
tag=TAG20120910T011327
channel ORA_AUX_DISK_1: restored backup piece 1
channel ORA_AUX_DISK_1: restore complete, elapsed time: 00:00:01
```

```
Finished restore at 10-09-2012 01:17:28

datafile 6 switched to datafile copy
input datafile copy RECID=8 STAMP=793588648 file
name=/u01/app/oracle/backup_test/ZRIK_PITR_ORCL/datafile/o1_mf_u
sers_84thx7b9_.dbf

contents of Memory Script:
{
# set requested point in time
set until  scn 2634118;
# online the datafiles restored or switched
sql clone "alter database datafile  6 online";
# recover and open resetlogs
recover clone database tablespace  "USERS", "SYSTEM",
"UNDOTBS1", "SYSAUX" delete archivelog;
alter clone database open resetlogs;
}
executing Memory Script

executing command: SET until clause

sql statement: alter database datafile  6 online

Starting recover at 10-09-2012 01:17:28
using channel ORA_AUX_DISK_1

starting media recovery

archived log for thread 1 with sequence 61 is already on disk as
file
/u01/app/oracle/fast_recovery_area/ORCL/archivelog/2012_09_10/o1
_mf_1_61_84thrcth_.arc
archived log file
name=/u01/app/oracle/fast_recovery_area/ORCL/archivelog/2012_09_
10/o1_mf_1_61_84thrcth_.arc thread=1 sequence=61
media recovery complete, elapsed time: 00:00:00
Finished recover at 10-09-2012 01:17:29

database opened

contents of Memory Script:
{
# create directory for datapump import
sql "create or replace directory TSPITR_DIROBJ_DPDIR as ''
/u01/app/oracle/backup_test''";
# create directory for datapump export
sql clone "create or replace directory TSPITR_DIROBJ_DPDIR as ''
/u01/app/oracle/backup_test''";
}
executing Memory Script

sql statement: create or replace directory TSPITR_DIROBJ_DPDIR
as ''/u01/app/oracle/backup_test''

sql statement: create or replace directory TSPITR_DIROBJ_DPDIR
as ''/u01/app/oracle/backup_test''

Performing export of tables...
   EXPDP> Starting "SYS"."TSPITR_EXP_zrik_pgCo":
   EXPDP> Estimate in progress using BLOCKS method...
   EXPDP> Processing object type TABLE_EXPORT/TABLE/TABLE_DATA
   EXPDP> Total estimation using BLOCKS method: 64 KB
   EXPDP> Processing object type TABLE_EXPORT/TABLE/TABLE
   EXPDP> Processing object type
TABLE_EXPORT/TABLE/STATISTICS/TABLE_STATISTICS
   EXPDP> Processing object type
TABLE_EXPORT/TABLE/STATISTICS/MARKER
   EXPDP> ORA-39127: unexpected error from call to export_string
:=SYS.DBMS_TRANSFORM_EXIMP.INSTANCE_INFO_EXP('AQ$_ORDERS_QUEUETA
BLE_S','IX',1,1,'12.00.00.00.00',newblock)
ORA-00376: file 2 cannot be read at this time
ORA-01110: data file 2:
'/u01/app/oracle/oradata/orcl/example01.dbf'
ORA-06512: at "SYS.DBMS_TRANSFORM_EXIMP", line 197
ORA-06512: at line 1
ORA-06512: at "SYS.DBMS_METADATA", line 9796
ORA-39127: unexpected error from call to export_string
:=SYS.DBMS_TRANSFORM_EXIMP.INSTANCE_INFO_EXP('AQ$_STREAMS_QUEUE_
TABLE_S','IX',1,1,'12.00.00.00.00',newblock)
ORA-00376: file 2 cannot be read at this time
ORA-01110: data file 2:
'/u01/app/oracle/oradata/orcl/example01.dbf'
ORA-06512: at "SYS.DBMS_TRANSFORM_EXIMP", line 197
ORA-06512: at line 1
ORA-06512: at "SYS.DBMS_METADATA", line 9796
   EXPDP> . . exported "HR"."TEST_TABLE1"
5.992 KB       3 rows
```

```
   EXPDP> Master table "SYS"."TSPITR_EXP_zrik_pgCo" successfully
loaded/unloaded
   EXPDP>
*****************************************************************
*
   EXPDP> Dump file set for SYS.TSPITR_EXP_zrik_pgCo is:
   EXPDP>   /u01/app/oracle/backup_test/tspitr_zrik_42203.dmp
   EXPDP> Job "SYS"."TSPITR_EXP_zrik_pgCo" completed with 2
error(s) at Mon Sep 10 01:18:13 2012 elapsed 0 00:00:25
Export completed


contents of Memory Script:
{
# shutdown clone before import
shutdown clone abort
}
executing Memory Script

Oracle instance shut down

Performing import of tables...
   IMPDP> Master table "SYS"."TSPITR_IMP_zrik_fmvv" successfully
loaded/unloaded
   IMPDP> Starting "SYS"."TSPITR_IMP_zrik_fmvv":
   IMPDP> Processing object type TABLE_EXPORT/TABLE/TABLE
   IMPDP> Processing object type TABLE_EXPORT/TABLE/TABLE_DATA
   IMPDP> . . imported "HR"."TEST_TABLE1"
5.992 KB       3 rows
   IMPDP> Processing object type
TABLE_EXPORT/TABLE/STATISTICS/TABLE_STATISTICS
   IMPDP> Processing object type
TABLE_EXPORT/TABLE/STATISTICS/MARKER
   IMPDP> Job "SYS"."TSPITR_IMP_zrik_fmvv" successfully
completed at Mon Sep 10 01:18:26 2012 elapsed 0 00:00:04
Import completed


Removing automatic instance
Automatic instance removed
auxiliary instance file
/u01/app/oracle/backup_test/ORCL/datafile/o1_mf_temp_84thw2c6_.t
mp deleted
```

```
auxiliary instance file
/u01/app/oracle/backup_test/ZRIK_PITR_ORCL/onlinelog/o1_mf_3_84t
hxdfy_.log deleted
auxiliary instance file
/u01/app/oracle/backup_test/ZRIK_PITR_ORCL/onlinelog/o1_mf_2_84t
hxcdy_.log deleted
auxiliary instance file
/u01/app/oracle/backup_test/ZRIK_PITR_ORCL/onlinelog/o1_mf_1_84t
hxb7j_.log deleted
auxiliary instance file
/u01/app/oracle/backup_test/ZRIK_PITR_ORCL/datafile/o1_mf_users_
84thx7b9_.dbf deleted
auxiliary instance file
/u01/app/oracle/backup_test/ORCL/datafile/o1_mf_sysaux_84thth08_
.dbf deleted
auxiliary instance file
/u01/app/oracle/backup_test/ORCL/datafile/o1_mf_undotbs1_84thth4
f_.dbf deleted
auxiliary instance file
/u01/app/oracle/backup_test/ORCL/datafile/o1_mf_system_84thth32_
.dbf deleted
auxiliary instance file
/u01/app/oracle/backup_test/ORCL/controlfile/o1_mf_84tht7co_.ctl
deleted
auxiliary instance file tspitr_zrik_42203.dmp deleted
Finished recover at 10-09-2012 01:18:28

RMAN>
```

5. Check that the table is fully recovered.

```
RMAN> select * from hr.test_table1;

       NUM NAME                      NOW
---------- ------------------------- ---------
         1 First test row            10-SEP-12
         2 Second test row           10-SEP-12
         3 Third test row

RMAN> exit
$
```

Practices for Lesson 13: Real-Time Database Operation Monitoring

Chapter 13

Practices for Lesson 13: Overview

Practices Overview

In the practices for this lesson, you monitor database operations in the orcl database. You will use DBMS_SQL_MONITOR new functions to identify and start operations and Enterprise Manager Database Express to monitor database operations execution.

Practice 13-1: Starting Enterprise Manager Database Express

Overview

In this practice, you will use Enterprise Manager Database Express or Enterprise Manager Cloud Control to monitor the various database operations started in the `orcl` database.

Tasks

1. Check if Enterprise Manager Database Express is started.
   a. Verify that the value of the `DISPATCHERS` instance parameter is set to `(PROTOCOL=TCP)(SERVICE=orclXDB)`.

```
$ sqlplus / as sysdba

Connected to:
Oracle Database 12c Enterprise Edition Release 12.1.0.0.2 -
64bit Production
With the Partitioning, OLAP, Advanced Analytics, Real
Application Testing and Unified Auditing options

SQL> SHOW PARAMETER dispatchers
NAME                 TYPE       VALUE
-------------------- ---------- -------------
dispatchers          string     (PROTOCOL=TCP)(SERVICE=orclXDB)
max_dispatchers      integer

SQL>
```

   b. Select the port number used for Enterprise Manager Database Express.

```
SQL> SELECT dbms_xdb_config.gethttpport FROM DUAL;

GETHTTPPORT
-----------
       5500

SQL> EXIT
$
```

   c. Verify that the listener is running and listens to the localhost (*yourserver*) using TCP protocol, the port 5500, the http presentation with RAW session data.

```
$ lsnrctl status


Connecting to
(DESCRIPTION=(ADDRESS=(PROTOCOL=IPC)(KEY=EXTPROC1521)))
STATUS of the LISTENER
------------------------
Alias                     LISTENER
```

```
Version                    TNSLSNR for Linux: Version 12.1.0.0.2
- Production
Start Date                 14-AUG-2012 09:05:04
Uptime                     0 days 6 hr. 12 min. 20 sec
Trace Level                off
Security                   ON: Local OS Authentication
SNMP                       OFF
Listener Parameter File
/u01/app/oracle/product/12.1.0/dbhome_1/network/admin/listener.o
ra
Listener Log File
/u01/app/oracle/diag/tnslsnr/youserver/listener/alert/log.xml
Listening Endpoints Summary...
  (DESCRIPTION=(ADDRESS=(PROTOCOL=ipc)(KEY=EXTPROC1521)))
  (DESCRIPTION=(ADDRESS=(PROTOCOL=tcp)(HOST=<Your
hostname>)(PORT=1521)))
  (DESCRIPTION=(ADDRESS=(PROTOCOL=tcp)(HOST=<Your
hostname>)(PORT=5500))(Presentation=HTTP)(Session=RAW))
Services Summary...
Service "orcl" has 1 instance(s).
  Instance "orcl", status READY, has 1 handler(s) for this
service...
Service "orclXDB" has 1 instance(s).
  Instance "orcl", status READY, has 1 handler(s) for this
service...
The command completed successfully
$
```

d. Launch a browser and use the following URL http://localhost:5500/em.

e. Log in with `SYS` and `oracle_4U` password as `SYSDBA`.

2. If EM Database Express is not started, proceed with the following steps; else, go to the next practice.

   a. Set the value of the `DISPATCHERS` instance parameter to `(PROTOCOL=TCP)(SERVICE=orclXDB)`.

```
$ sqlplus / as sysdba

Connected to:
Oracle Database 12c Enterprise Edition Release 12.1.0.0.2 -
64bit Production
With the Partitioning, OLAP, Advanced Analytics, Real
Application Testing and Unified Auditing options

SQL> ALTER SYSTEM SET dispatchers =
'(PROTOCOL=TCP)(SERVICE=orclXDB)' scope=both;
```

```
System altered.

SQL>
```

b. Set the HTTP port.

```
SQL> EXEC DBMS_XDB_CONFIG.setHTTPSPort(5500)

PL/SQL procedure successfully completed.

SQL> EXIT
$
```

c. Restart or start the listener.

```
$ lsnrctl stop

Connecting to
(DESCRIPTION=(ADDRESS=(PROTOCOL=IPC)(KEY=EXTPROC1521)))
The command completed successfully
$
```

```
$ lsnrctl start

Starting /u01/app/oracle/product/12.1.0/dbhome_1/bin/tnslsnr:
please wait...

TNSLSNR for Linux: Version 12.1.0.0.2 - Production
System parameter file is
/u01/app/oracle/product/12.1.0/dbhome_1/network/admin/listener.o
ra
Log messages written to
/u01/app/oracle/diag/tnslsnr/youserver/listener/alert/log.xml
Listening on:
(DESCRIPTION=(ADDRESS=(PROTOCOL=ipc)(KEY=EXTPROC1521)))
Listening on:
(DESCRIPTION=(ADDRESS=(PROTOCOL=tcp)(HOST=youserver)(PORT=1521))
)

Connecting to
(DESCRIPTION=(ADDRESS=(PROTOCOL=IPC)(KEY=EXTPROC1521)))
STATUS of the LISTENER
------------------------
Alias                     LISTENER
Version                   TNSLSNR for Linux: Version 12.1.0.0.2
- Production
Start Date                14-AUG-2012 15:25:04
```

```
Uptime                    0 days 0 hr. 0 min. 0 sec
Trace Level               off
Security                  ON: Local OS Authentication
SNMP                      OFF
Listener Parameter File
/u01/app/oracle/product/12.1.0/dbhome_1/network/admin/listener.o
ra
Listener Log File
/u01/app/oracle/diag/tnslsnr/youserver/listener/alert/log.xml
Listening Endpoints Summary...
  (DESCRIPTION=(ADDRESS=(PROTOCOL=ipc)(KEY=EXTPROC1521)))

(DESCRIPTION=(ADDRESS=(PROTOCOL=tcp)(HOST=youserver)(PORT=1521))
)
The listener supports no services
The command completed successfully
$
```

d. Launch a browser and use the following URL https://localhost:5500/em.

Practice 13-2: Identifying and Starting Database Operations

Overview

In this practice, you use the DBMS_SQL_MONITOR.BEGIN_OPERATION function to identify and start several database operations.

Tasks

1. Make sure you are at the ~/labs/DBOps directory and your environment points to the orcl instance.

```
$ cd ~/labs/DBOps
$ . oraenv
ORACLE_SID = [orcl] ? orcl
The Oracle base remains unchanged with value /u01/app/oracle
$
```

2. Run the DBOps_setup.sh script to ensure that the users SH and HR can connect and get the SELECT ANY DICTIONARY privilege for the purpose of this practice.

```
$ ./DBOps_setup.sh

SQL*Plus: Release 12.1.0.0.2 Production on Mon Sep 10 01:38:52
2012

Copyright (c) 1982, 2012, Oracle.  All rights reserved.

Last Successful login time: Mon Sep 10 2012 01:38:09 +00:00

Connected to:
Oracle Database 12c Enterprise Edition Release 12.1.0.0.2 -
64bit Production
With the Partitioning, OLAP, Advanced Analytics, Real
Application Testing and Unified Auditing options

User altered.

User altered.

Grant succeeded.

Database altered.

Disconnected from Oracle Database 12c Enterprise Edition Release
12.1.0.0.2 - 64bit Production
With the Partitioning, OLAP, Advanced Analytics, Real
Application Testing
and Unified Auditing options
```

```
$
```

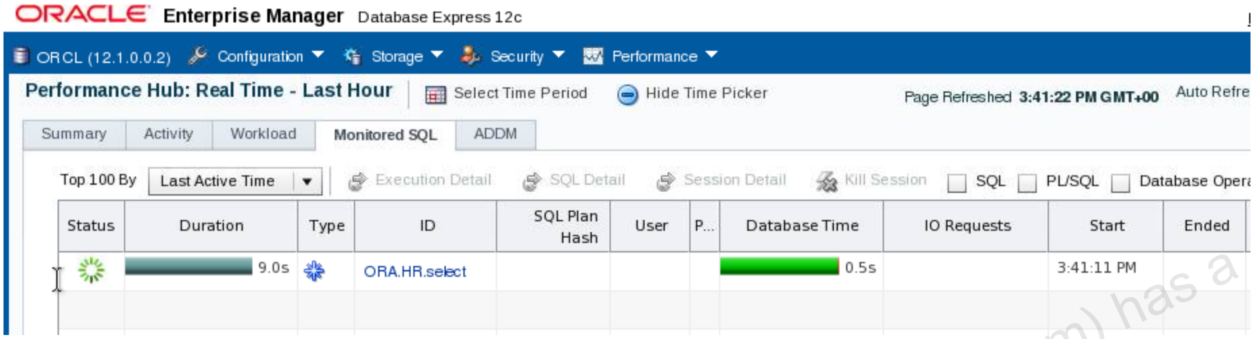
3. Start a first database operation. Name the database operation `ORA.HR.select`. The database operation performs several `SELECT` statements as `HR` user. The database operation is started and completed with the `DBMS_MONITOR.BEGIN_OPERATION` and `DBMS_MONITOR.END_OPERATION` procedures. **Move on to the next step while the statement is running, to monitor the database operation with EM Database Express**. If you want to have time to view the database operation `ORA.HR.select` in EM Database Express or EM Cloud Control, do not execute `EXEC DBMS_SQL_MONITOR.END_OPERATION` right after the third `SELECT` statement.

```
VAR dbop_eid NUMBER;
EXEC :dbop_eid := DBMS_SQL_MONITOR.BEGIN_OPERATION
('ORA.HR.select', forced_tracking => 'Y')
select a.employee_id, b.employee_id from hr.employees a,
hr.employees b;
select * from hr.departments;
select  a.table_name , b.table_name FROM dict a, dict b;
EXEC DBMS_SQL_MONITOR.END_OPERATION('ORA.HR.select', :dbop_eid)
```

```
$ sqlplus hr/oracle_4U

Connected to:
Oracle Database 12c Enterprise Edition Release 12.1.0.0.2 -
64bit Production
With the Partitioning, OLAP, Data Mining, Real Application
Testing
and Unified Auditing options

SQL> VAR dbop_eid NUMBER;
SQL> EXEC :dbop_eid := DBMS_SQL_MONITOR.BEGIN_OPERATION
('ORA.HR.select', forced_tracking => 'Y')

PL/SQL procedure successfully completed.

SQL> SELECT a.employee_id, b.employee_id FROM hr.employees a,
hr.employees b;
…
6889 rows selected.

SQL> SELECT * FROM hr.departments;
…
27 rows selected.
SQL> SELECT  a.table_name , b.table_name FROM dict a, dict b;
…
SQL> EXEC DBMS_SQL_MONITOR.END_OPERATION('ORA.HR.select',
:dbop_eid)
```

```
PL/SQL procedure successfully completed.

SQL>
```

4. View the database operation currently executing using Enterprise Manager Database Express. From Enterprise Manager Database Express, click the “Performance” menu, then the “Performance Hub” option, then the “Monitored SQL” tab. The list of database operations appears.



Performance Hub: Real Time - Last Hour

Page Refreshed 3:41:22 PM GMT+00

Top 100 By: Last Active Time

☐ SQL ☐ PL/SQL ☐ Database Opera

Status	Duration	Type	ID	SQL Plan Hash	User	P...	Database Time	IO Requests	Start	Ended
	9.0s		ORA.HR.select				0.5s		3:41:11 PM	

Note: You could also use Enterprise Manager Cloud Control.

1) Restart the Enterprise Manager Repository Database `em12rep`.

```
$ . oraenv
ORACLE_SID = [orcl] ? em12rep
The Oracle base for
ORACLE_HOME=/u01/app/oracle/product/12.1.0/dbhome_1 is
/u01/app/oracle
$ sqlplus / as sysdba

Connected to an idle instance.
SQL> startup
ORACLE instance started.

Total System Global Area  400846848 bytes
Fixed Size                  2271568 bytes
Variable Size             339740336 bytes
Database Buffers           50331648 bytes
Redo Buffers                8503296 bytes
Database mounted.
Database opened.
SQL> EXIT
$
```

2) Restart the OMS.

```
$ export OMS_HOME=/u01/app/oracle/product/middleware/oms
$ $OMS_HOME/bin/emctl start oms
Oracle Enterprise Manager Cloud Control 12c Release 2
```

```
Copyright (c) 1996, 2012 Oracle Corporation.  All rights
reserved.
Starting Oracle Management Server...
Starting WebTier...
WebTier Successfully Started
Oracle Management Server Successfully Started
Oracle Management Server is Up
WARNING: Limit of open file descriptors is found to be 1024.
The OMS has been started but it may run out of descriptors under
heavy usage.
For proper functioning of OMS, please set "ulimit -n" to be at
least 4096.
$
```

3) Use https://localhost:7802/em to get the Enterprise Manager Cloud Control console appear, enter **sysman** in the User Name field and **Oracle123** in the Password field. Then click Login.
The status of the orcl database agent might be in unreachable state because the oms was stopped in practice 9. However this has no incidence on other practices.
From the "Targets" menu, click "Databases" and select orcl. Log in as SYS with oracle_4U password as SYSDBA. From the "Performance" menu, click the "SQL "Monitoring" option. You will see the same list as the one from Enterprise Manager Database Express.

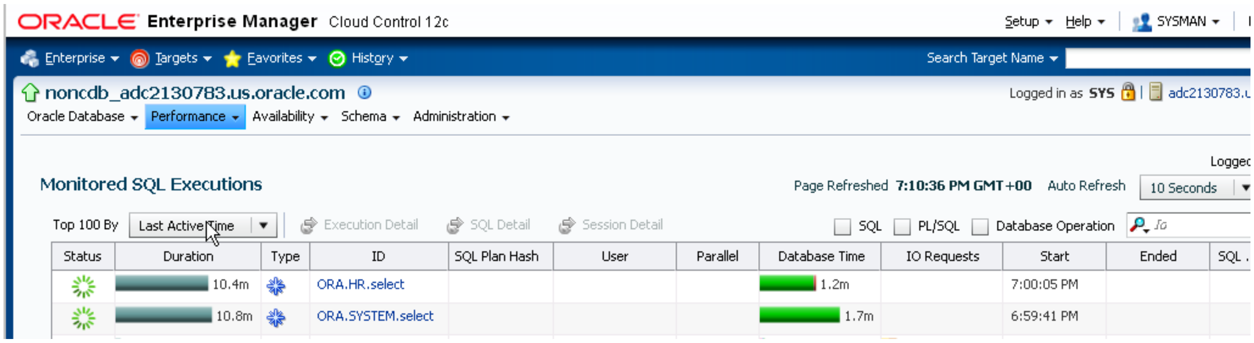
5. From another session connected to orcl, start a second database operation. Name the database operation ORA.SYSTEM.select. The database operation performs several SELECT statements as SYSTEM user using the DBMS_MONITOR.BEGIN_OPERATION procedure to monitor.

```
VAR dbop_eid NUMBER;
EXEC :dbop_eid := DBMS_SQL_MONITOR.BEGIN_OPERATION
('ORA.SYSTEM.select', forced_tracking => 'Y')
SELECT a.table_name , b.table_name FROM dict a, dict b;
SELECT c.cust_id, c.cust_last_name, c.cust_first_name,
       s.prod_id, p.prod_name, s.time_id
FROM   sh.sales s, sh.customers c, sh.products p
WHERE  s.cust_id = c.cust_id
AND    s.prod_id = p.prod_id
ORDER BY c.cust_id, s.time_id;
EXEC DBMS_SQL_MONITOR.END_OPERATION('ORA.SYSTEM.select',
:dbop_eid)
```

```
$ . oraenv
ORACLE_SID = [em12rep] ? orcl
The Oracle base for
ORACLE_HOME=/u01/app/oracle/product/12.1.0/dbhome_1 is
/u01/app/oracle
$ sqlplus system/oracle_4U
```

```
Connected to:
Oracle Database 12c Enterprise Edition Release 12.1.0.0.2 -
64bit Production
With the Partitioning, OLAP, Data Mining, Real Application
Testing and Unified Auditing options

SQL> VAR dbop_eid NUMBER;
SQL> EXEC :dbop_eid := DBMS_SQL_MONITOR.BEGIN_OPERATION
('ORA.SYSTEM.select', forced_tracking => 'Y')

PL/SQL procedure successfully completed.

SQL> SELECT a.table_name , b.table_name FROM dict a, dict b;
…

SQL> SELECT c.cust_id, c.cust_last_name, c.cust_first_name,
s.prod_id, p.prod_name, s.time_id
FROM   sh.sales s, sh.customers c, sh.products p
WHERE  s.cust_id = c.cust_id
AND    s.prod_id = p.prod_id
ORDER BY c.cust_id, s.time_id;
…

SQL> EXEC DBMS_SQL_MONITOR.END_OPERATION('ORA.SYSTEM.select',
:dbop_eid)

PL/SQL procedure successfully completed.

SQL>
```

6. View the new database operation currently executing using Enterprise Manager Database Express. From Enterprise Manager Database Express, refresh the list of database operations. From Enterprise Manager Cloud Control, you will view the same list of monitored database operations.


ORACLE Enterprise Manager Cloud Control 12c
Setup ▾ Help ▾ SYSMAN ▾
Enterprise ▾ Targets ▾ Favorites ▾ History ▾
Search Target Name ▾
noncdb_adc2130783.us.oracle.com
Logged in as SYS | adc2130783.u
Oracle Database ▾ Performance ▾ Availability ▾ Schema ▾ Administration ▾
Logged
Execution Detail
SQL Detail
Session Detail

Monitored SQL Executions

Page Refreshed **7:10:36 PM GMT+00** Auto Refresh 10 Seconds

Top 100 By Last Active Time

□ SQL □ PL/SQL □ Database Operation

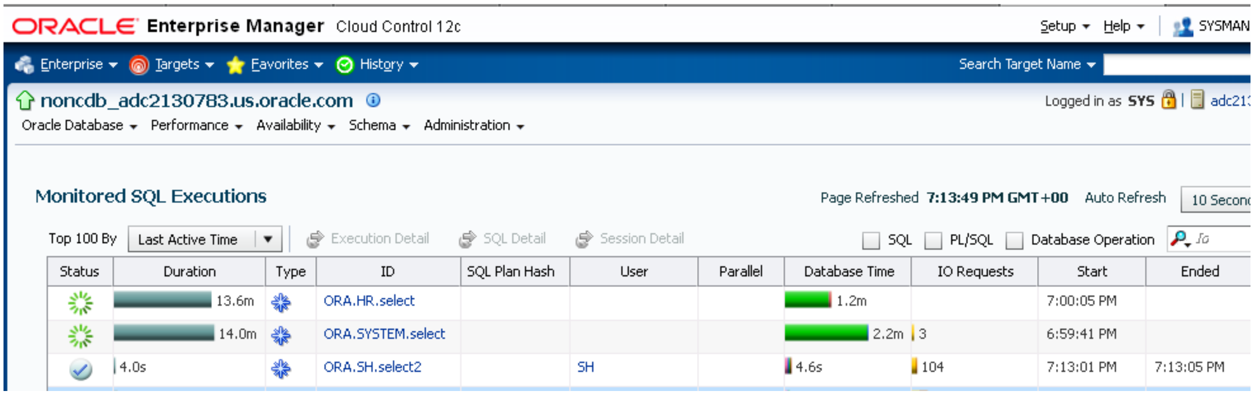
Status	Duration	Type	ID	SQL Plan Hash	User	Parallel	Database Time	IO Requests	Start	Ended	SQL .
	10.4m		ORA.HR.select				1.2m		7:00:05 PM		
	10.8m		ORA.SYSTEM.select				1.7m		6:59:41 PM		

7. Start a third database operation in another session. Name the database operation `ORA.SH.select2`. The database operation performs several `SELECT` statements as `SH` user using the `DBMS_MONITOR.BEGIN_OPERATION` procedure to monitor.

```
VAR dbop_eid NUMBER;
EXEC :dbop_eid := DBMS_SQL_MONITOR.BEGIN_OPERATION
('ORA.SH.select2', forced_tracking => 'Y')
SELECT c.cust_id, c.cust_last_name, c.cust_first_name,
       s.prod_id, p.prod_name, s.time_id
FROM   sh.sales s, sh.customers c, sh.products p
WHERE  s.cust_id = c.cust_id
AND    s.prod_id = p.prod_id
ORDER BY c.cust_id, s.time_id;
EXEC DBMS_SQL_MONITOR.END_OPERATION('ORA.SH.select2', :dbop_eid)
```

```
$ . oraenv
ORACLE_SID = [em12rep] ? orcl
The Oracle base for
ORACLE_HOME=/u01/app/oracle/product/12.1.0/dbhome_1 is
/u01/app/oracle
$ sqlplus sh/oracle_4U


Connected to:
Oracle Database 12c Enterprise Edition Release 12.1.0.0.2 -
64bit Production
With the Partitioning, OLAP, Data Mining, Real Application
Testing and Unified Auditing options


SQL> VAR dbop_eid NUMBER;
SQL> EXEC :dbop_eid := DBMS_SQL_MONITOR.BEGIN_OPERATION
('ORA.SH.select2', forced_tracking => 'Y')

PL/SQL procedure successfully completed.

SQL> SELECT c.cust_id, c.cust_last_name, c.cust_first_name,
s.prod_id, p.prod_name, s.time_id
FROM   sh.sales s, sh.customers c, sh.products p
WHERE  s.cust_id = c.cust_id
AND    s.prod_id = p.prod_id
ORDER BY c.cust_id, s.time_id;
 …

SQL> EXEC DBMS_SQL_MONITOR.END_OPERATION('ORA.SH.select2',
:dbop_eid)

PL/SQL procedure successfully completed.

SQL> EXIT
$
```

8. View the new database operation using Enterprise Manager Database Express or Enterprise Manager Cloud Control.



Monitored SQL Executions

Page Refreshed 7:13:49 PM GMT+00 Auto Refresh 10 Second

Top 100 By Last Active Time ▼ □ SQL □ PL/SQL □ Database Operation

Status	Duration	Type	ID	SQL Plan Hash	User	Parallel	Database Time	IO Requests	Start	Ended
	13.6m		ORA.HR.select				1.2m		7:00:05 PM	
	14.0m		ORA.SYSTEM.select				2.2m	3	6:59:41 PM	
✓	4.0s		ORA.SH.select2		SH		4.6s	104	7:13:01 PM	7:13:05 PM

Practice 13-3: Identifying and Starting Database Load Operations *(Optional)*

Overview

In this practice, you will use Enterprise Manager Database Express to monitor the various database load operations started in the `orcl` database.

Tasks

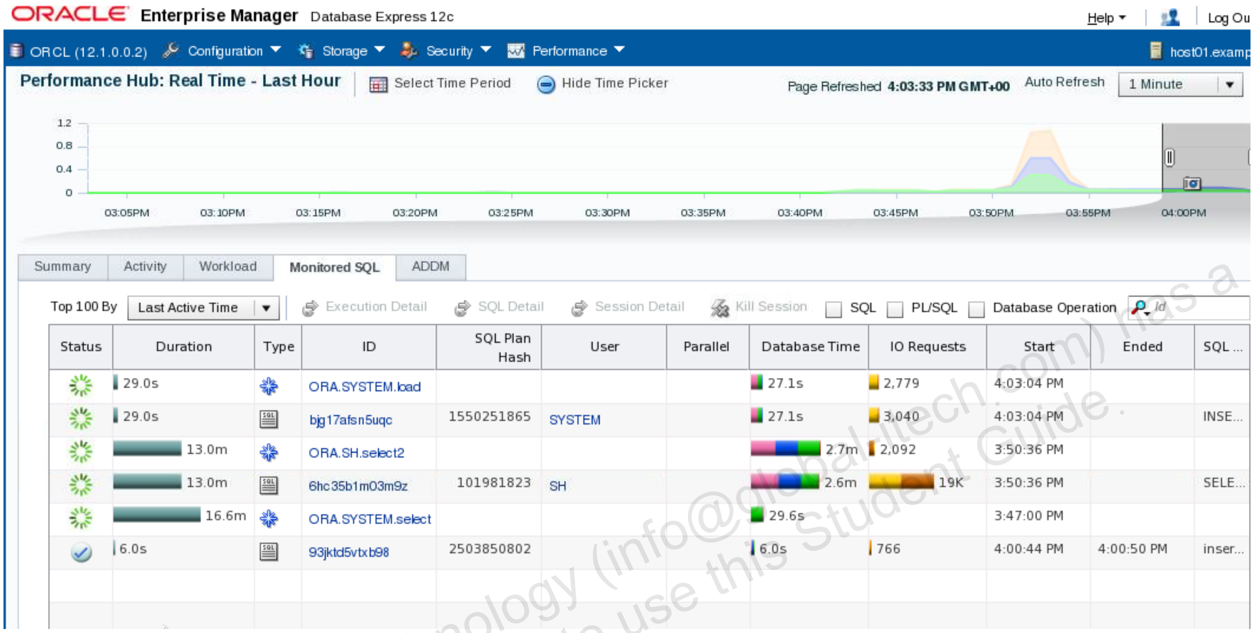
1. Start a new database operation. Name the database operation `ORA.SYSTEM.load`. The database operation performs several bulk-load statements as the `SYSTEM` user using the `DBMS_MONITOR.BEGIN_OPERATION` procedure to monitor.

```
VAR dbop_eid NUMBER;
EXEC :dbop_eid := DBMS_SQL_MONITOR.BEGIN_OPERATION
('ORA.SYSTEM.load', forced_tracking => 'Y')
INSERT /*+ append */ INTO sh.sales NOLOGGING SELECT * from
sh.sales WHERE ROWNUM < 1000;
INSERT /*+ append */ INTO sh.sales NOLOGGING SELECT * from
sh.sales;
EXEC DBMS_SQL_MONITOR.END_OPERATION('ORA.SYSTEM.load',
:dbop_eid)
```

```
$ sqlplus system/oracle_4U

Connected to:
Oracle Database 12c Enterprise Edition Release 12.1.0.0.2 -
64bit Production
With the Partitioning, OLAP, Data Mining, Real Application
Testing and Unified Auditing options

SQL> VAR dbop_eid NUMBER;
SQL> EXEC :dbop_eid := DBMS_SQL_MONITOR.BEGIN_OPERATION
('ORA.SYSTEM.load', forced_tracking => 'Y')

PL/SQL procedure successfully completed.
SQL> INSERT /*+ append */ INTO sh.sales NOLOGGING SELECT * from
sh.sales WHERE ROWNUM<1000;


999 rows created.


SQL> INSERT /*+ append */ INTO sh.sales NOLOGGING SELECT * from
sh.sales;


919842 rows created.


SQL> EXEC DBMS_SQL_MONITOR.END_OPERATION('ORA.SYSTEM.load',
:dbop_eid)
```

```
PL/SQL procedure successfully completed.

SQL>
```

2. View the new database operation currently executing using Enterprise Manager Database Express. From Enterprise Manager Database Express, refresh the list of database operations.



Performance Hub: Real Time - Last Hour | Select Time Period | Hide Time Picker | Page Refreshed 4:03:33 PM GMT+00 | Auto Refresh 1 Minute

Top 100 By Last Active Time ▾ | □ SQL □ PL/SQL □ Database Operation | Id

Status	Duration	Type	ID	SQL Plan Hash	User	Parallel	Database Time	IO Requests	Start	Ended	SQL ...
⟳	29.0s		ORA.SYSTEM.load				27.1s	2,779	4:03:04 PM		
⟳	29.0s		bjg17afsn5uqc	1550251865	SYSTEM		27.1s	3,040	4:03:04 PM		INSE...
⟳	13.0m		ORA.SH.select2				2.7m	2,092	3:50:36 PM		
⟳	13.0m		6hc35b1m03m9z	101981823	SH		2.6m	19K	3:50:36 PM		SELE...
⟳	16.6m		ORA.SYSTEM.select				29.6s		3:47:00 PM		
✓	6.0s		93jktd5vtxb98	2503850802			6.0s	766	4:00:44 PM	4:00:50 PM	inser...

Practice 13-4: Cleaning Up

Overview

In this practice, you revoke the `SELECT ANY DICTIONARY` privilege granted to `HR` and `SH` users for the purpose of these practices.

1. Revoke the `SELECT ANY DICTIONARY` privilege granted to `HR` and `SH` users.

```
$ sqlplus system/oracle_4U

Connected to:
Oracle Database 12c Enterprise Edition Release 12.1.0.0.2 -
64bit Production
With the Partitioning, OLAP, Data Mining and Real Application
Testing options

SQL> REVOKE SELECT ANY DICTIONARY FROM hr, sh;
Revoke succeeded.
SQL> EXIT
$
```

Practices for Lesson 14: Schema and Data Changes Management

Chapter 14

Practices for Lesson 14

Practices Overview

In the practice for this lesson, you will use the Schema Change Plans demo to understand and view the steps required during schema change plan usage between two databases to synchronize two databases together.

Practice 14-1: Using Schema Change Plans

Overview

In this practice you use a browser to execute the Schema Change Plans demonstration.

Tasks

1. Launch a browser and enter: file:////home/oracle/demos/Schema_Change_Plans/Schema_change_plan.html.

Practices for Lesson 15: SQL Tuning Enhancements

Chapter 15

Practices for Lesson 15

Practices Overview

This practice covers the dynamic plans part of the Adaptive Execution Plans feature in Oracle Database 12*c*.

Practice 15-1: Using Dynamic Plans

Overview

In this practice, you will use the dynamic plans part of the Adaptive Execution Plans feature.

Tasks

1. Unlock the OE account in ORCL database and then grant SELECT ANY DICTIONARY privilege to OE.

```
$ . oraenv
ORACLE_SID = [orcl] ? orcl
The Oracle base remains unchanged with value /u01/app/oracle
$ sqlplus / as sysdba

Connected to:
Oracle Database 12c Enterprise Edition Release 12.1.0.0.2 -
64bit Production
With the Partitioning, OLAP, Advanced Analytics, Real
Application Testing and Unified Auditing options

SQL> grant select any dictionary to oe;

Grant succeeded.

SQL> alter user oe account unlock;
User altered.

SQL> alter user oe identified by oracle_4U;

User altered.

SQL>
```

2. From the same SQL*Plus session, connect as user OE and show the execution plan of the following query without executing it:

```
select /*+ monitor*/ product_name
from order_items o, product_information p
where o.unit_price = 15
  and quantity > 1
  and p.product_id = o.product_id;
```

```
SQL> connect oe/oracle_4U
Connected.
SQL>
SQL> explain plan for
select /*+ monitor*/ product_name
```

```
from order_items o, product_information p
where o.unit_price = 15
  and quantity > 1
  and p.product_id = o.product_id;
  2    3    4    5    6
Explained.

SQL> set lines 300
SQL> select * from table(dbms_xplan.display());

PLAN_TABLE_OUTPUT
----------------------------------------------------------------------------------------------------
----------------------------------------------------------------------------------------------------
-------------------------------------------------------------------------------------------------
Plan hash value: 1255158658


----------------------------------------------------------------------------------------------------
--
| Id  | Operation                    | Name                   | Rows  | Bytes | Cost (%CPU)| Time
|
----------------------------------------------------------------------------------------------------
--
|   0 | SELECT STATEMENT             |                        |     4 |   128 |     7   (0)| 00:00:01
|
|   1 |  NESTED LOOPS                |                        |       |       |            |
|
|   2 |   NESTED LOOPS               |                        |     4 |   128 |     7   (0)| 00:00:01
|
|*  3 |    TABLE ACCESS FULL         | ORDER_ITEMS            |     4 |    48 |     3   (0)| 00:00:01
|
|*  4 |    INDEX UNIQUE SCAN         | PRODUCT_INFORMATION_PK |     1 |       |     0   (0)| 00:00:01
|
|   5 |   TABLE ACCESS BY INDEX ROWID| PRODUCT_INFORMATION    |     1 |    20 |     1   (0)| 00:00:01
|

PLAN_TABLE_OUTPUT
----------------------------------------------------------------------------------------------------
----------------------------------------------------------------------------------------------------
-------------------------------------------------------------------------------------------------
----------------------------------------------------------------------------------------------------
--

Predicate Information (identified by operation id):
---------------------------------------------------


   3 - filter("O"."UNIT_PRICE"=15 AND "QUANTITY">1)
   4 - access("P"."PRODUCT_ID"="O"."PRODUCT_ID")

18 rows selected.

SQL>
```

3. What do you observe?
   The plan is using a simple NESTED LOOP join.
4. Now, execute the same query:

```
SQL> select /*+ monitor*/ product_name
     from   order_items o, product_information p
     where  o.unit_price = 15
     and    quantity > 1
     and    p.product_id = o.product_id;
  2    3    4    5
PRODUCT_NAME
--------------------------------------------------
Screws <B.28.S>
Screws <B.28.S>
Screws <B.28.S>
Screws <B.28.S>
…
Screws <B.28.S>
Screws <B.28.S>

13 rows selected.

SQL>
```

5. Show the resulting execution plan:

```
SQL> select * from table(dbms_xplan.display_cursor());

PLAN_TABLE_OUTPUT
-----------------------------------------------------------------------------------------
-----------------------------------------------------------------------------------------
-----------------------------------------------------------------------------------------
-------------------------------------
SQL_ID  439uv1rqa5svb, child number 1
-------------------------------------
select /*+ monitor*/ product_name from order_items o,
product_information p where o.unit_price = 15   and quantity > 1   and
p.product_id = o.product_id

Plan hash value: 1553478007

-----------------------------------------------------------------------------------------
--
| Id  | Operation          | Name                 | Rows  | Bytes | Cost (%CPU)| Time
|
-----------------------------------------------------------------------------------------
--



|   0 | SELECT STATEMENT   |                      |       |       |     8 (100)|
|
|*  1 |  HASH JOIN         |                      |    13 |   416 |     8   (0)| 00:00:01
|
|*  2 |   TABLE ACCESS FULL| ORDER_ITEMS          |    13 |   156 |     3   (0)| 00:00:01
|
|   3 |   TABLE ACCESS FULL| PRODUCT_INFORMATION  |   288 |  5760 |     5   (0)| 00:00:01
|
-----------------------------------------------------------------------------------------
--
```

```
Predicate Information (identified by operation id):
---------------------------------------------------

   1 - access("P"."PRODUCT_ID"="O"."PRODUCT_ID")
   2 - filter(("O"."UNIT_PRICE"=15 AND "QUANTITY">1))

Note
-----
   - cardinality feedback used for this statement


27 rows selected.

SQL>
```

6. What do you observe and conclude?
   The actual plan used at execution was a `HASH_JOIN`.
   Why did the plan change?
   The plan changed because the optimizer realized during the execution that the number of rows actually returned from the `order_items` table was much larger than expected. Multiple single-column predicates on the `order_items` table caused the initial cardinality estimate to be incorrect. The misestimation can't be corrected by extended statistics because one of the predicates is a non-equality predicate.
7. How would you confirm the plan change was caused by dynamic plans?
   By looking in `V$SQL` and checking the value of the new column `IS_RESOLVED_ADAPTIVE_PLAN`.

```
SQL> column sql_text format a30
SQL> select sql_id, sql_text, is_resolved_adaptive_plan
     from   v$sql
     where  sql_text like 'select /*+ monitor*/ product_name%';
  2    3

SQL_ID        SQL_TEXT                       I
------------- ------------------------------ -
439uv1rqa5svb select /*+ monitor*/ product_n Y
              ame from order_items o, produc
              t_information p where o.unit_p
              rice = 15   and quantity > 1
               and p.product_id = o.product_
              id

SQL>
```

8. The information learnt via dynamic plans is persisted as a SQL directive. Check the SQL directives created for the previous execution.

```
SQL> connect / as sysdba
Connected.
SQL> set pages 9999
SQL> set lines 300
SQL> col state format a5
SQL> col subobject_name format a11
SQL> col col_name format a11
SQL> col object_name format a13
SQL> select d.directive_id, o.object_type, o.object_name,
            o.subobject_name col_name, d.type, d.state, d.reason
     from   dba_sql_plan_directives d, dba_sql_plan_dir_objects
o
     where  d.DIRECTIVE_ID=o.DIRECTIVE_ID
     and    o.object_name in ('ORDER_ITEMS')
     order by d.directive_id;

no rows selected

SQL> /

no rows selected

SQL>
```

9. **-- You have to wait for a while before it is persisted. MMON is responsible for the flush.**
   -- In Database 12*c* DML monitoring and column usage information flush has been transferred to MMON instead of SMON.

```
SQL> select d.directive_id, o.object_type, o.object_name,
            o.subobject_name col_name, d.type, d.state, d.reason
     from   dba_sql_plan_directives d, dba_sql_plan_dir_objects
o
     where  d.DIRECTIVE_ID=o.DIRECTIVE_ID
    and    o.object_name in
('ORDER_ITEMS','PRODUCT_INFORMATION')
     order by d.directive_id;

DIRECTIVE_ID OBJECT OBJECT_NAME   COL_NAME    TYPE
STATE            REASON
------------ ------ ------------- ----------- ---------------- -
----------------- ------------------------------------

  1.3208E+19 COLUMN ORDER_ITEMS   UNIT_PRICE  DYNAMIC_SAMPLING
MISSING STATS    SINGLE TABLE CARDINALITY MISESTIMATE
  1.3208E+19 TABLE  ORDER_ITEMS               DYNAMIC_SAMPLING
MISSING STATS    SINGLE TABLE CARDINALITY MISESTIMATE
  1.3208E+19 COLUMN ORDER_ITEMS   QUANTITY    DYNAMIC_SAMPLING
MISSING STATS    SINGLE TABLE CARDINALITY MISESTIMATE


SQL>
```

Practice 15-2: Using Re-optimization

Overview

In this practice, you discover how the re-optimization (Cardinality Feedback) part of the Adaptive Execution Plans feature in Oracle Database 12*c* works.

Tasks

1. Unlock the `SH` account in `ORCL` database and then grant `SELECT ANYDICTONARY` privilege to `SH`.

```
SQL> grant select any dictionary to sh;

Grant succeeded.

SQL> alter user sh identified by oracle_4U account unlock;

User altered.

SQL>
```

2. Execute the following query. Note the `gather_plan_statistics` hint is used to display the actual number of rows returned from each operation in the plan. This will allow you to compare the optimizer's estimates with the actual number of rows returned.

```
select /*+ gather_plan_statistics HINT1 */ c.cust_first_name,
c.cust_last_name, sum(s.amount_sold)
from  customers c, sales s
where c.cust_id=s.cust_id
  and   c.cust_city='Los Angeles'
  and   c.cust_state_province='CA'
  and   c.country_id=52790
  and   s.time_id='09-NOV-00'
group by c.cust_first_name, c.cust_last_name;
```

```
SQL> connect sh/oracle_4U
Connected.
SQL> set pages 9999
SQL> set lines 300
SQL> COL sql_text format a30
SQL> select /*+ gather_plan_statistics HINT1 */
          c.cust_first_name, c.cust_last_name,
sum(s.amount_sold)
     from  customers c, sales s
     where c.cust_id=s.cust_id
     and   c.cust_city='Los Angeles'
     and   c.cust_state_province='CA'
     and   c.country_id=52790
     and   s.time_id='09-NOV-00'
     group by c.cust_first_name, c.cust_last_name;

no rows selected

SQL>
```

3. Display the associated execution plan. What do you observe?
   There is a large difference between the estimated (E-Rows) and the actual number of rows returned (A-Rows). This statement looks like a candidate for re-optimization.

```
SQL> select * from
table(dbms_xplan.display_cursor(FORMAT=>'ALLSTATS LAST'));


PLAN_TABLE_OUTPUT
---------------------------------------------------------------------------------------------------------------------------------------------
---------------------------------------------------------------------------------------------------------------------------------------------
-----------------
SQL_ID  1s4x3nv00bub2, child number 0
-------------------------------------
select /*+ gather_plan_statistics HINT1 */ c.cust_first_name, c.cust_last_name,
sum(s.amount_sold) from  customers c, sales s where c.cust_id=s.cust_id
and   c.cust_city='Los Angeles' and   c.cust_state_province='CA' and
c.country_id=52790 and   s.time_id='09-NOV-00' group by
c.cust_first_name, c.cust_last_name

Plan hash value: 3910574683

---------------------------------------------------------------------------------------------------------------------------------------------
-----------------
| Id  | Operation                                      | Name           | Starts | E-Rows | A-Rows |   A-Time   | Buffers | Reads  |  OMem |
1Mem | Used-Mem |
---------------------------------------------------------------------------------------------------------------------------------------------
-----------------
|   0 | SELECT STATEMENT                               |                |      1 |        |      0 |00:00:00.33 |   35069 |   1282 |       |
|       |
|   1 |  HASH GROUP BY                                 |                |      1|       1 |      0 |00:00:00.33 |   35069 |   1282 |   909K|
909K|          |
|   2 |   MERGE JOIN                                   |                |      1 |      1 |      0 |00:00:00.33 |   35069 |   1282 |       |
|       |
|*  3 |    TABLE ACCESS BY INDEX ROWID                 | CUSTOMERS      |      1 |      1 |    482 |00:00:00.07 |   34981 |   1198 |       |
|       |
|   4 |     INDEX FULL SCAN                            | CUSTOMERS_PK   |      1 |  55500 |  35610 |00:00:00.05 |      74 |     70 |       |
|       |
|*  5 |    SORT JOIN                                   |                |    482 |   2152 |      0 |00:00:00.22 |      88 |     84 | 95232 |
95232 |83968  (0)|
|   6 |     PARTITION RANGE SINGLE                     |                |      1 |   2152 |   2152 |00:00:00.12 |      88 |     84 |       |
|       |
|   7 |      TABLE ACCESS BY LOCAL INDEX ROWID BATCHED| SALES          |      1 |   2152 |   2152 |00:00:00.11 |      88 |     84 |       |
|       |
|   8 |       BITMAP CONVERSION TO ROWIDS              |                |      1 |        |   2152 |00:00:00.09 |       2 |      2 |       |
|       |
|*  9 |        BITMAP INDEX SINGLE VALUE               | SALES_TIME_BIX |      1 |        |      1 |00:00:00.02 |       2 |      2 |       |
|       |
---------------------------------------------------------------------------------------------------------------------------------------------
-----------------

Predicate Information (identified by operation id):
---------------------------------------------------


   3 - filter(("C"."CUST_CITY"='Los Angeles' AND "C"."CUST_STATE_PROVINCE"='CA' AND "C"."COUNTRY_ID"=52790))
   5 - access("C"."CUST_ID"="S"."CUST_ID")
       filter("C"."CUST_ID"="S"."CUST_ID")
   9 - access("S"."TIME_ID"='09-NOV-00')



33 rows selected.

SQL>
```

4. How would you confirm this statement will be re-optimized?
   You can confirm that by checking the value of the `is_reoptimizable` column in `v$sql`. This column indicates that this statement will be re-parsed on the next execution and information learnt on the first execution about the actual number of rows returned will be used to generate a better plan.

```
SQL> select sql_id, child_number, sql_text, is_reoptimizable
     from   v$sql
     where  sql_text like '%+ gather_plan_statistics HINT1%';
2    3

SQL_ID        CHILD_NUMBER SQL_TEXT                       I
------------- ------------ ------------------------------ -
6wdhzmv5882s2            0 select sql_id, child_number, s N
                           ql_text, is_reoptimizable from
                           v$sql where  sql_text like
                           '%gather_plan_statistics HINT1
                           %'

c81f6n14144kd            0 select /*+ gather_plan_statisti Y
                           cs HINT1 */ c.cust_first_name,
                           c.cust_last_name, sum(s.amount
                           _sold)    from  customers c, s
                           ales s    where c.cust_id=s.cu
                           st_id     and   c.cust_city='L
                           os Angeles' and c.cust_s
                           tate_province='CA'     and c
                           .country_id=52790     and   s.
                           time_id='09-NOV-00'    group b
                           y c.cust_first_name, c.cust_la
                           st_name

SQL>
```

5. Confirm your guess is correct:

```
SQL> select /*+ gather_plan_statistics HINT1 */
          c.cust_first_name, c.cust_last_name,
sum(s.amount_sold)
     from  customers c, sales s
     where c.cust_id=s.cust_id
     and   c.cust_city='Los Angeles'
     and   c.cust_state_province='CA'
     and   c.country_id=52790
     and   s.time_id='09-NOV-00'
     group by c.cust_first_name, c.cust_last_name;
```

```
no rows selected

SQL> select * from
table(dbms_xplan.display_cursor(FORMAT=>'ALLSTATS LAST'));

PLAN_TABLE_OUTPUT
-----------------------------------------------------------------------------------------------------------------
-----------------------------------------------------------------------------------------------------------------
-----------------------------------------------------------------
SQL_ID  1s4x3nv00bub2, child number 1
-------------------------------------
select /*+ gather_plan_statistics HINT1 */ c.cust_first_name, c.cust_last_name,
sum(s.amount_sold) from  customers c, sales s where c.cust_id=s.cust_id
and   c.cust_city='Los Angeles' and   c.cust_state_province='CA' and
c.country_id=52790 and   s.time_id='09-NOV-00' group by
c.cust_first_name, c.cust_last_name

Plan hash value: 3910574683

-----------------------------------------------------------------------------------------------------------------
-------------------------------
| Id  | Operation                                | Name           | Starts | E-Rows | A-Rows |   A-Time   |
Buffers |  OMem |  1Mem | Used-Mem |
-----------------------------------------------------------------------------------------------------------------
-------------------------------
|   0 | SELECT STATEMENT                         |                |      1 |        |      0 |00:00:00.04 |
35065 |       |       |          |
|   1 |  HASH GROUP BY                           |                |      1 |      1 |      0 |00:00:00.04 |
35065 |   909K|   909K|          |
|   2 |   MERGE JOIN                             |                |      1 |    767 |      0 |00:00:00.04 |
35065 |       |       |          |
|*  3 |    TABLE ACCESS BY INDEX ROWID           | CUSTOMERS      |      1 |    482 |    482 |00:00:00.06 |
34981 |       |       |          |
|   4 |     INDEX FULL SCAN                      | CUSTOMERS_PK   |      1 |  35610 |  35610 |00:00:00.04 |
74 |       |       |          |
|*  5 |    SORT JOIN                             |                |    482 |   2152 |      0 |00:00:00.01 |
84 | 99328 | 99328 |88064  (0)|
|   6 |     PARTITION RANGE SINGLE               |                |      1 |   2152 |   2152 |00:00:00.01 |
84 |       |       |          |
|   7 |      TABLE ACCESS BY LOCAL INDEX ROWID BATCHED| SALES          |      1 |   2152 |   2152 |00:00:00.01 |
84 |       |       |          |
|   8 |       BITMAP CONVERSION TO ROWIDS        |                |      1 |        |   2152 |00:00:00.01 |
2 |       |       |          |
|*  9 |        BITMAP INDEX SINGLE VALUE         | SALES_TIME_BIX |      1 |        |      1 |00:00:00.01 |
2 |       |       |          |
-----------------------------------------------------------------------------------------------------------------
-------------------------------

Predicate Information (identified by operation id):
---------------------------------------------------

   3 - filter(("C"."CUST_CITY"='Los Angeles' AND "C"."CUST_STATE_PROVINCE"='CA' AND "C"."COUNTRY_ID"=52790))
   5 - access("C"."CUST_ID"="S"."CUST_ID")
       filter("C"."CUST_ID"="S"."CUST_ID")
   9 - access("S"."TIME_ID"='09-NOV-00')

Note
-----
   - cardinality feedback used for this statement


37 rows selected.

SQL>
```

6. Check that a new child cursor was created:

```
SQL> select sql_id, child_number, sql_text, is_reoptimizable
     from   v$sql
     where  sql_text like '%gather_plan_statistics HINT1%';
  2    3

SQL_ID        CHILD_NUMBER SQL_TEXT                       I
------------- ------------ ------------------------------ -
6wdhzmv5882s2            0 select sql_id, child_number, s N
                            ql_text, is_reoptimizable from
                            v$sql where  sql_text like
                            '%gather_plan_statistics HINT1
                            %'

c81f6n14144kd            0 select /*+ gather_plan_statisti N
                            cs HINT1 */ c.cust_first_name,
                            c.cust_last_name, sum(s.amount
                            _sold)    from  customers c, s
                            ales s    where c.cust_id=s.cu
                            st_id     and   c.cust_city='L
                            os Angeles' and c.cust_s
                            tate_province='CA'     and c
                            .country_id=52790     and   s.
                            time_id='09-NOV-00'    group b
                            y c.cust_first_name, c.cust_la
                            st_name

c81f6n14144kd            1 select /*+ gather_plan_statisti Y
                            cs HINT1 */ c.cust_first_name,
                            c.cust_last_name, sum(s.amount
                            _sold)    from  customers c, s
                            ales s    where c.cust_id=s.cu
                            st_id     and   c.cust_city='L
                            os Angeles' and c.cust_s
                            tate_province='CA'     and c
                            .country_id=52790     and   s.
                            time_id='09-NOV-00'    group b
                            y c.cust_first_name, c.cust_la
                            st_name

987h8zk1zax24            0 select sql_id, child_number, s N
                            ql_text, is_reoptimizable from
                            v$sql where  sql_text like
                            '%gather_plan_statistics HINT1
                            %'

SQL>
```

```
SQL>  select CHILD_NUMBER, USE_FEEDBACK_STATS
      from    v$sql_shared_cursor where SQL_ID='c81f6n14144kd';

CHILD_NUMBER USE_FEEDBACK_STATS
------------ ------------------------------
       0     N
       1     Y

SQL>
```

7. Force the SQL plan directive to be flushed and check to see it was persisted into the data dictionary:

```
SQL> connect / as sysdba
Connected.
SQL> col state format a5
SQL> col subobject_name format a11
SQL> col col_name format a11
SQL> col object_name format a13
SQL> col dir_id format a23
SQL> col owner format a5
SQL> col state format A20

SQL> set echo on
SQL>
SQL> exec dbms_spd.flush_sql_plan_directive

PL/SQL procedure successfully completed.

SQL> select to_char(d.directive_id) dir_id, o.owner,
         o.object_name, o.subobject_name col_name,
o.object_type,
         d.type, d.state, d.reason
     from   dba_sql_plan_directives d, dba_sql_plan_dir_objects
o
     where  d.DIRECTIVE_ID=o.DIRECTIVE_ID
     and    o.owner in ('SH')
     order by 1,2,3,4,5;
```

```
DIR_ID                  OWNER OBJECT_NAME   COL_NAME    OBJECT
TYPE             STATE        REASON
----------------------- ----- ------------- ----------- ------ -
--------------- ------------ --------------------------------
11713533193494921230    SH    CUSTOMERS                 TABLE
DYNAMIC_SAMPLING NEW JOIN CARDINALITY MISESTIMATE
11713533193494921230    SH    SALES                     TABLE
DYNAMIC_SAMPLING NEW JOIN CARDINALITY MISESTIMATE



SQL> EXIT
$
```

Practices for Lesson 16: Emergency Monitoring and Compare Period ADDM

Chapter 16

Practices for Lesson 16

Practices Overview

In these practices, you will use Emergency Monitoring to troubleshoot a hanging situation discovered in `orcl` database instance.
Then you can optionally run the Compare Period ADDM demonstration to know how to use this new feature.

Practice 16-1: Using Emergency Monitoring

Assumption

You are managing the target `orcl` database and are already connected to Enterprise Manager Cloud Control in the target `orcl` database. Make sure you restart Enterprise Manager Cloud Control (you stopped it in Practice 9-1 to enable Unified Auditing).

1. Restart the Enterprise Manager Repository Database `em12rep`.

```
$ . oraenv
ORACLE_SID = [orcl] ? em12rep
The Oracle base for
ORACLE_HOME=/u01/app/oracle/product/12.1.0/dbhome_1 is
/u01/app/oracle
$ sqlplus / as sysdba

Connected to an idle instance.
SQL> startup
ORACLE instance started.

Total System Global Area  400846848 bytes
Fixed Size                  2271568 bytes
Variable Size             339740336 bytes
Database Buffers           50331648 bytes
Redo Buffers                8503296 bytes
Database mounted.
Database opened.
SQL> EXIT
$
```

2. Restart the OMS.

```
$ export OMS_HOME=/u01/app/oracle/product/middleware/oms
$ $OMS_HOME/bin/emctl start oms
Oracle Enterprise Manager Cloud Control 12c Release 2
Copyright (c) 1996, 2012 Oracle Corporation.  All rights
reserved.
Starting Oracle Management Server...
Starting WebTier...
WebTier Successfully Started
Oracle Management Server Successfully Started
Oracle Management Server is Up
WARNING: Limit of open file descriptors is found to be 1024.
The OMS has been started but it may run out of descriptors under
heavy usage.
For proper functioning of OMS, please set "ulimit -n" to be at
least 4096.
```

```
$
```

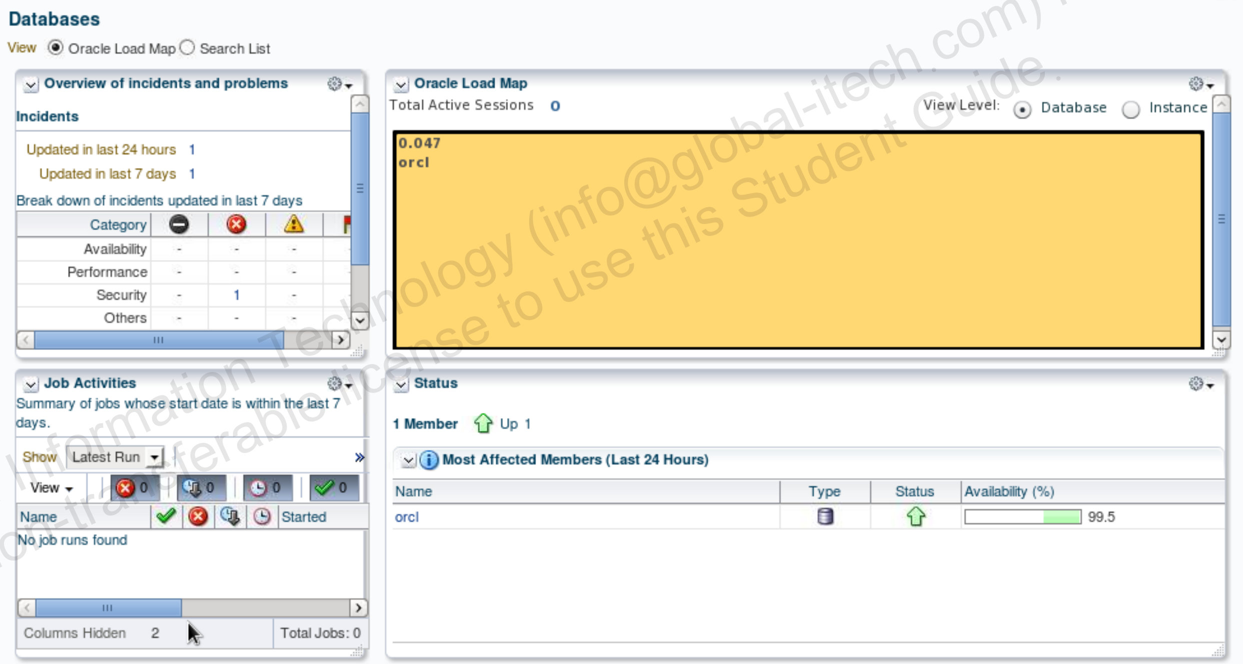
3. Use https://localhost:7802/em to get the Enterprise Manager Cloud Control Console appear, enter **sysman** in the User Name field and **Oracle123** in the Password field. Then click Login.
   The status of the orcl database agent might be in unreachable state because the oms was stopped in practice 9. However this has no incidence on other practices.

Overview

In this practice you will troubleshoot a hanging situation after users told you they could not connect to the orcl instance anymore.
You can use Emergency Monitoring only from Enterprise Manager Cloud Control.

Tasks

1. Make sure you are already connected to Enterprise Manager Cloud Control in the target orcl database with SYSDBA credentials.
   a. Connect to Enterprise Manager Cloud Control as sysman with Oracle123 password.
   b. After being connected, click “Targets” and then “Databases”.



Databases

View ● Oracle Load Map ○ Search List

Overview of incidents and problems

Incidents

Updated in last 24 hours 1
Updated in last 7 days 1

Break down of incidents updated in last 7 days

Category	⛔	✖	⚠
Availability	-	-	-
Performance	-	-	-
Security	-	1	-
Others	-	-	-

Job Activities

Summary of jobs whose start date is within the last 7 days.

Show Latest Run

View ✖ 0 | 0 | 0 | ✓ 0

Name	✓	✖			Started
No job runs found					

Columns Hidden 2 Total Jobs: 0

Oracle Load Map

Total Active Sessions 0

View Level: ● Database ○ Instance

0.047
orcl

Status

1 Member Up 1

Most Affected Members (Last 24 Hours)

Name	Type	Status	Availability (%)
orcl		⇧	99.5

   c. From the right pane, click the orcl database instance. You are still not connected to orcl.
   d. Therefore, go to any of the menus and try to execute any DBA operation, “Create a tablespace” or “Create a new user” for example.
   e. Because you created a named credential in Practice 1-3, the named credential is proposed to make a connection as SYSDBA to the instance. You can use it or create a new credential to log in. Log in.
2. In a terminal window, make sure you are at the ~/labs/Emergency directory and your environment points to the orcl instance.

```
$ cd ~/labs/Emergency
$ . oraenv
```

```
ORACLE_SID = [orcl] ? orcl
The Oracle base remains unchanged with value /u01/app/oracle
$
```

3. Execute the `Emergency_setup.sh` shell script.

```
$ ./Emergency_setup.sh
DROP TRIGGER trig_after_logon
*
ERROR at line 1:
ORA-04080: trigger 'TRIG_AFTER_LOGON' does not exist
```

   Leave the `Emergency_setup.sh` shell script pending.

4. Meanwhile you create a new user. From another terminal window, connect to the `orcl` database as `SYSTEM` with `oracle_4U` password.

```
$ sqlplus system/oracle_4U

SQL*Plus: Release 12.1.0.0.2 Production on Thu Sep 6 15:41:45
2012

Copyright (c) 1982, 2012, Oracle.  All rights reserved.
```

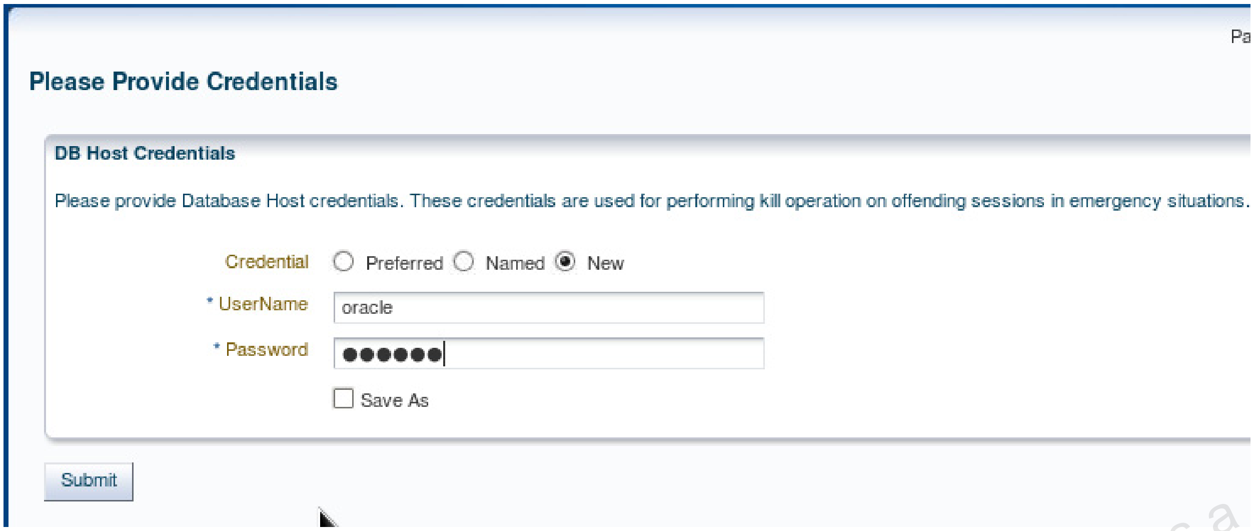
   But the session is pending.

5. From another terminal window, connect to the `orcl` database `AS SYSDBA` with `oracle_4U` password.

```
$ sqlplus / AS SYSDBA

SQL*Plus: Release 12.1.0.0.2 Production on Thu Sep 6 15:41:45
2012

Copyright (c) 1982, 2012, Oracle.  All rights reserved.
```

   But the session is pending.

6. You will use Emergency Monitoring to quickly solve the hanging situation. Click Performance from the menu, then Emergency Monitoring from the list of options.



Please Provide Credentials

DB Host Credentials

Please provide Database Host credentials. These credentials are used for performing kill operation on offending sessions in emergency situations.

Credential ○ Preferred ○ Named ● New

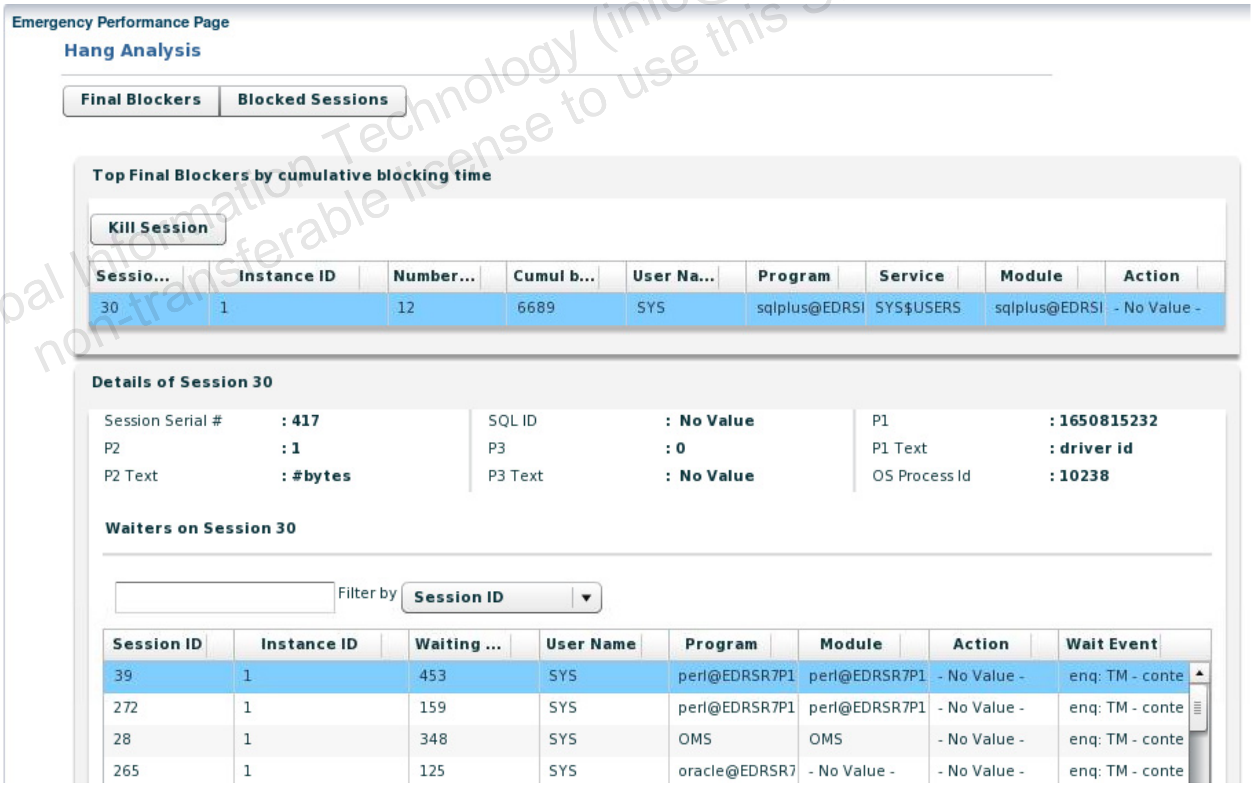
* UserName: oracle

* Password: ●●●●●●

□ Save As

In DB Host Credentials, choose New and enter `oracle` for both the UserName and the Password. Then click Submit.

7. The Hanging Analysis page is displayed. It shows the blockers and blocked sessions. To release the blocked sessions, kill the blocking session. Click the Kill Session button. And click YES to approve the operation. A message appears to inform that the session has been killed. Click OK.



Emergency Performance Page

Hang Analysis

Top Final Blockers by cumulative blocking time

Sessio...	Instance ID	Number...	Cumul b...	User Na...	Program	Service	Module	Action
30	1	12	6689	SYS	sqlplus@EDRSI	SYS$USERS	sqlplus@EDRSI	- No Value -

Details of Session 30

Session Serial #	: 417	SQL ID	: No Value	P1	: 1650815232
P2	: 1	P3	: 0	P1 Text	: driver id
P2 Text	: #bytes	P3 Text	: No Value	OS Process Id	: 10238

Waiters on Session 30

Filter by: Session ID

Session ID	Instance ID	Waiting ...	User Name	Program	Module	Action	Wait Event
39	1	453	SYS	perl@EDRSR7P1	perl@EDRSR7P1	- No Value -	enq: TM - conte
272	1	159	SYS	perl@EDRSR7P1	perl@EDRSR7P1	- No Value -	enq: TM - conte
28	1	348	SYS	OMS	OMS	- No Value -	enq: TM - conte
265	1	125	SYS	oracle@EDRSR7	- No Value -	- No Value -	enq: TM - conte

8. Check that the sessions that were hanging are now released.

```
$ sqlplus system/oracle_4U

SQL*Plus: Release 12.1.0.0.2 Production on Thu Sep 6 15:41:45
2012

Copyright (c) 1982, 2012, Oracle.  All rights reserved.

SQL> exit
$
```

From the other terminal window, the connection is also released.

```
$ sqlplus / AS SYSDBA

SQL*Plus: Release 12.1.0.0.2 Production on Thu Sep 6 15:41:45
2012

Copyright (c) 1982, 2012, Oracle.  All rights reserved.

SQL> exit
$
```

Practice 16-2: Cleaning Up

Overview

In this practice you clean up the environment of the `orcl` database.

Tasks

1. From the session where you launched the `Emergency_setup.sh` shell script, clean up the `orcl` database. If the `Emergency_setup.sh` shell script does not end up (because of the 1800 seconds sleep), interrupt it with a `CTRL C`.

```
$ ./Emergency_setup.sh
DROP TRIGGER trig_after_logon
*
ERROR at line 1:
ORA-04080: trigger 'TRIG_AFTER_LOGON' does not exist

CTRL C
$
```

2. From one of the released SQL*Plus session, drop the trigger.

```
SQL> ALTER SYSTEM SET "_system_trig_enabled"=FALSE;

System altered.

SQL> DROP TRIGGER trig_after_logon;

Trigger dropped.

SQL>
```

3. Shut down the `orcl` instance to release resources for the next practices.

```
SQL> SHUTDOWN IMMEDIATE
Database closed.
Database dismounted.
ORACLE instance shut down.
SQL> EXIT
$
```

Practice 16-3: Using Compare Period ADDM *(optional)*

Overview

In this demonstration, you will see how to use the Compare Period ADDM with Enterprise Manager Cloud Control. You discover that during the current period of time, the performance is decreasing. You need to understand why the performance changed, and the root causes of this change, so as to perform appropriate actions to solve the issue.

In this practice, you use a browser to execute the Compare Period ADDM demonstration.

Tasks

1. Launch a browser and enter in the file:////home/oracle/demos/Compare_Period_ADDM/Compare_Period_ADDM.html

Practices for Lesson 17: Resource Manager and Other Performance Enhancements

Chapter 17

Practices for Lesson 17

Practices Overview

In this practice, you create two CDB Resource Manager plans and associated directives to limit CPU resources used by two PDBs.

Practice 17-1: Using CDB Resource Manager Plans and Directives

Overview

In this practice, you create two CDB Resource Manager plans and associated directives to limit CPU resources used by two PDBs.

Tasks

1. Connect to the root of `cdb2` as `SYSDBA` and cleanup your environment by executing the `rsrc_cleanup.sql` script. The script will close all PDBs except `PDB1_1` and `PDB2_2`.
   a. Make sure you are in the `~/labs/RM` directory and your environment points to the `cdb2` instance.

```
$ cd ~/labs/RM
$ . oraenv
ORACLE_SID = [cdb1] ? cdb2
The Oracle base remains unchanged with value /u01/app/oracle
$
```

   b. Execute the `rsrc_cleanup.sql` script.
      1) Start up the multitenant container database instance if not already done.

```
$ sqlplus / as sysdba

Connected to an idle instance.

SQL> STARTUP
```

```
ORACLE instance started.

Total System Global Area 2455228416 bytes
Fixed Size                  2274024 bytes
Variable Size            1090522392 bytes
Database Buffers         1342177280 bytes
Redo Buffers               20254720 bytes
Database mounted.
Database opened.

SQL> @rsrc_cleanup.sql


Pluggable database altered.


Pluggable database altered.


NAME                               CON_ID OPEN_MODE
------------------------------ ---------- ----------
PDB$SEED                                2 READ ONLY
PDB1_1                                  6 READ WRITE
PDB2                                    3 MOUNTED
PDB2_2                                  4 READ WRITE
PDB_ORCL2                               5 MOUNTED


System altered.


NAME
--------------------------------
ORA$INTERNAL_CDB_PLAN


System altered.


PL/SQL procedure successfully completed.
```

```
PL/SQL procedure successfully completed.

BEGIN
DBMS_Resource_Manager.Delete_CDB_Plan_Directive('fairplan',
'pdb1_1'); END;

*
ERROR at line 1:
ORA-29358: resource plan FAIRPLAN does not exist
ORA-06512: at "SYS.DBMS_RMIN_SYS", line 3168
ORA-06512: at "SYS.DBMS_RESOURCE_MANAGER", line 1605
ORA-06512: at line 1



BEGIN
DBMS_Resource_Manager.Delete_CDB_Plan_Directive('fairplan',
'pdb2_2'); END;

*
ERROR at line 1:
ORA-29358: resource plan FAIRPLAN does not exist
ORA-06512: at "SYS.DBMS_RMIN_SYS", line 3168
ORA-06512: at "SYS.DBMS_RESOURCE_MANAGER", line 1605
ORA-06512: at line 1



BEGIN DBMS_Resource_Manager.Delete_CDB_Plan('fairplan'); END;

*
ERROR at line 1:
ORA-29358: resource plan FAIRPLAN does not exist
ORA-06512: at "SYS.DBMS_RMIN_SYS", line 2863
ORA-06512: at "SYS.DBMS_RESOURCE_MANAGER", line 1451
ORA-06512: at line 1



BEGIN
DBMS_Resource_Manager.Delete_CDB_Plan_Directive('unfairplan',
'pdb1_1'); END;

*
ERROR at line 1:
ORA-29358: resource plan UNFAIRPLAN does not exist
```

```
ORA-06512: at "SYS.DBMS_RMIN_SYS", line 3168
ORA-06512: at "SYS.DBMS_RESOURCE_MANAGER", line 1605
ORA-06512: at line 1



BEGIN
DBMS_Resource_Manager.Delete_CDB_Plan_Directive('unfairplan',
'pdb2_2'); END;

*
ERROR at line 1:
ORA-29358: resource plan UNFAIRPLAN does not exist
ORA-06512: at "SYS.DBMS_RMIN_SYS", line 3168
ORA-06512: at "SYS.DBMS_RESOURCE_MANAGER", line 1605
ORA-06512: at line 1



BEGIN DBMS_Resource_Manager.Delete_CDB_Plan('unfairplan'); END;

*
ERROR at line 1:
ORA-29358: resource plan UNFAIRPLAN does not exist
ORA-06512: at "SYS.DBMS_RMIN_SYS", line 2863
ORA-06512: at "SYS.DBMS_RESOURCE_MANAGER", line 1451
ORA-06512: at line 1




PL/SQL procedure successfully completed.


PL/SQL procedure successfully completed.

SQL> exit
$
```

2. Open a terminal window (it will be referred to as window1) to connect to `pdb1_1` in `cdb2` and create a PL/SQL procedure that burns CPU in `PDB1_1` as the `SYSTEM` user. You can use the `create_burn_cpu.sql` script to create the procedure after connecting to `PDB1_1`.

```
$ cd ~/labs/RM
$ . oraenv
ORACLE_SID = [cdb1] ? cdb2
The Oracle base remains unchanged with value /u01/app/oracle
$
$ sqlplus system/oracle_4U@localhost/pdb1_1

Connected to:
Oracle Database 12c Enterprise Edition Release 12.1.0.0.2 -
64bit Production
With the Partitioning, OLAP, Advanced Analytics and Real
Application Testing options

SQL> @create_burn_cpu.sql

Procedure created.

SQL>
```

3. Open a second terminal window (it will be referred to as window2) to connect to `pdb2_2` in `cdb2` and create a PL/SQL procedure that burns CPU in `PDB2_2` as the `SYSTEM` user. You can use the `create_burn_cpu.sql` script to create the procedure after connecting to `PDB2_2`.

```
$ cd ~/labs/RM
$ . oraenv
ORACLE_SID = [cdb1] ? cdb2
The Oracle base remains unchanged with value /u01/app/oracle
$
$ sqlplus system/oracle_4U@localhost/pdb2_2

Connected to:
Oracle Database 12c Enterprise Edition Release 12.1.0.0.2 -
64bit Production
With the Partitioning, OLAP, Advanced Analytics and Real
Application Testing options

SQL> @create_burn_cpu.sql

Procedure created.
```

```
SQL>
```

4. From window1, create two new CDB plans called `FAIRPLAN` and `UNFAIRPLAN`. `FAIRPLAN` should give one share to both `PDB1_1` and `PDB2_2`, and `UNFAIRPLAN` should give one share to `PDB1_1` and five shares to `PDB2_2`.

```
SQL> alter session set container = CDB$Root;

Session altered.

SQL> EXEC DBMS_Resource_Manager.Clear_Pending_Area();

PL/SQL procedure successfully completed.

SQL> EXEC DBMS_Resource_Manager.Create_Pending_Area();

PL/SQL procedure successfully completed.

SQL> EXEC DBMS_Resource_Manager.Create_CDB_Plan('fairplan', 'One
share each');

PL/SQL procedure successfully completed.

SQL> EXEC
DBMS_Resource_Manager.Create_CDB_Plan_Directive('fairplan',
'pdb1_1', shares => 1);

PL/SQL procedure successfully completed.

SQL> EXEC
DBMS_Resource_Manager.Create_CDB_Plan_Directive('fairplan',
'pdb2_2', shares => 1);

PL/SQL procedure successfully completed.

SQL> EXEC DBMS_Resource_Manager.Create_CDB_Plan('unfairplan',
'one share to pdb1_1 and five to pdb2_2');

PL/SQL procedure successfully completed.

SQL> EXEC
DBMS_Resource_Manager.Create_CDB_Plan_Directive('unfairplan',
'pdb1_1', shares => 1);

PL/SQL procedure successfully completed.
```

```
SQL> EXEC
DBMS_Resource_Manager.Create_CDB_Plan_Directive('unfairplan',
'pdb2_2', shares => 5);

PL/SQL procedure successfully completed.

SQL> EXEC DBMS_Resource_Manager.Validate_Pending_Area();

PL/SQL procedure successfully completed.

SQL> EXEC DBMS_Resource_Manager.Submit_Pending_Area();

PL/SQL procedure successfully completed.

SQL>
```

5. Still from window1, make sure both plans and associated directives were created correctly.

```
SQL> SELECT Plan from CDB_CDB_Rsrc_Plans
     WHERE Con_ID = 1 AND Plan IN ('FAIRPLAN', 'UNFAIRPLAN')
     ORDER BY 1;
  2   3
PLAN
---------------------------------------------------------------
FAIRPLAN
UNFAIRPLAN

SQL> select Plan, Pluggable_Database, Shares
     from   CDB_CDB_Rsrc_Plan_Directives
     where  Con_ID = 1
     and    Plan in ('FAIRPLAN', 'UNFAIRPLAN')
     and    Pluggable_Database in ('PDB1_1', 'PDB2_2')
     order by 1, 2;
  2    3    4    5    6
PLAN
----------------------------------------------------------------
----------------
PLUGGABLE_DATABASE
----------------------------------------------------------------
----------------
    SHARES
----------
FAIRPLAN
PDB1_1
         1
```

```

FAIRPLAN
PDB2_2
         1



UNFAIRPLAN
PDB1_1
         1

UNFAIRPLAN
PDB2_2

         5


SQL>
```

6. From window1, activate the CDB plan `FAIRPLAN`.

```
SQL> CONNECT / AS SYSDBA
Connected.
SQL> alter system set resource_manager_plan = fairplan;

System altered.

SQL> select Name from v$Rsrc_Plan where Con_ID = 1;

NAME
--------------------------------
FAIRPLAN

SQL>
```

7. From window1, connect as the `SYSTEM` user in `PDB1_1` and set `SERVEROUPUT` variable to `ON`.

```
SQL> CONNECT System/oracle_4U@localhost/pdb1_1
Connected.
SQL> set serveroutput on
SQL>
```

8. From window2, connect as the SYSTEM user in PDB2_2 and set SERVEROUPUT variable to ON.

```
SQL> CONNECT System/oracle_4U@localhost/pdb2_2
Connected.
SQL> set serveroutput on
SQL>
```

9. **DO NOT WAIT AND GO TO STEP 10 RIGHT AFTER:** From window1, execute the CPU burner procedure you created at step 2.

```
SQL> EXEC Burn_CPU_For_RM_Demo();

CPU:   139.9 Wall:  882.1 k: 2000000000

PL/SQL procedure successfully completed.

SQL>
```

10. From window2, execute the CPU burner procedure you created at step 3.

```
SQL> EXEC Burn_CPU_For_RM_Demo();

CPU:   140.9 Wall:  886.7 k: 2000000000

PL/SQL procedure successfully completed.

SQL>
```

11. What do you observe?
Both procedures finish their execution almost at the same time, and have both consumed almost the same CPU and wall-clock time during their execution.
This is expected because each PDB is receiving one share of CPU.

12. From window1, connect as user SYS in the root, and change the Resource Manager plan to UNFAIRPLAN.

```
SQL> CONNECT / AS SYSDBA
Connected.
SQL>
SQL> alter system set resource_manager_plan = unfairplan;
System altered.

SQL> select Name from v$Rsrc_Plan where Con_ID = 1;

NAME
--------------------------------
UNFAIRPLAN

SQL>
```

13. **DO NOT WAIT AND GO TO STEP 14 RIGHT AFTER:** From window1, connect as user `SYSTEM` in `PDB1_1` and execute the CPU burner procedure you created at step 2.

```
SQL> CONNECT System/oracle_4U@localhost/pdb1_1
Connected.
SQL>
SQL> set serveroutput on
SQL>
SQL> execute Burn_CPU_For_RM_Demo();
CPU:   150.5 Wall:  333.9 k: 2000000000

PL/SQL procedure successfully completed.

SQL>
```

14. From window2 execute the CPU burner procedure you created at step 3.

```
SQL> execute Burn_CPU_For_RM_Demo();
CPU:   154.2 Wall:  205.6 k: 2000000000

PL/SQL procedure successfully completed.

SQL> EXIT
$
```

15. What do you observe?
Now, execution of the CPU burner procedure takes much longer to execute in `PDB1_1` than in `PDB2_2`.
This is expected because `PDB2_2` is assigned five shares while `PDB1_1` only one.
However, the difference is not five times slower simply because once the procedure executed in `PDB2_2`, all CPU cycles goes to `PDB1_1`.

16. Make sure you set the CDB plan back to its default and open PDB2_2.

```
$ sqlplus / as sysdba

Connected to:
Oracle Database 12c Enterprise Edition Release 12.1.0.0.2 -
64bit Production
With the Partitioning, OLAP, Advanced Analytics and Real
Application Testing options

SQL> alter system set resource_manager_plan = '';

System altered.

SQL> select Name from v$Rsrc_Plan where Con_ID = 1;

NAME
--------------------------------
ORA$INTERNAL_CDB_PLAN

SQL> alter pluggable database all open;

Pluggable database altered.

SQL> exit
$
```

Practice 17-2: Using Multi-Process Multi-Threaded Architecture

Overview

In this practice, you switch cdb2 to use the multi-process multi-threaded architecture.

1. From a terminal window, connected as the oracle user, list all processes and threads used to run your cdb2 instance.

```
$ ps -eLo "pid tid comm args" |grep cdb2
25075 25075 ora_pmon_cdb2   ora_pmon_cdb2
25077 25077 ora_psp0_cdb2   ora_psp0_cdb2
25089 25089 ora_vktm_cdb2   ora_vktm_cdb2
25095 25095 ora_gen0_cdb2   ora_gen0_cdb2
25097 25097 ora_mman_cdb2   ora_mman_cdb2
25101 25101 ora_diag_cdb2   ora_diag_cdb2
25103 25103 ora_ofsd_cdb2   ora_ofsd_cdb2
25105 25105 ora_dbrm_cdb2   ora_dbrm_cdb2
25107 25107 ora_dia0_cdb2   ora_dia0_cdb2
25109 25109 ora_dbw0_cdb2   ora_dbw0_cdb2
25111 25111 ora_lgwr_cdb2   ora_lgwr_cdb2
25113 25113 ora_ckpt_cdb2   ora_ckpt_cdb2
25115 25115 ora_smon_cdb2   ora_smon_cdb2
25117 25117 ora_reco_cdb2   ora_reco_cdb2
25119 25119 ora_lreg_cdb2   ora_lreg_cdb2
25121 25121 ora_mmon_cdb2   ora_mmon_cdb2
25123 25123 ora_mmnl_cdb2   ora_mmnl_cdb2
25125 25125 ora_d000_cdb2   ora_d000_cdb2
25127 25127 ora_s000_cdb2   ora_s000_cdb2
25152 25152 ora_rvwr_cdb2   ora_rvwr_cdb2
25153 25153 oracle_25153_cd oraclecdb2
(DESCRIPTION=(LOCAL=YES)(ADDRESS=(PROTOCOL=beq)))
25156 25156 ora_tmon_cdb2   ora_tmon_cdb2
25158 25158 ora_arc0_cdb2   ora_arc0_cdb2
25160 25160 ora_arc1_cdb2   ora_arc1_cdb2
25162 25162 ora_arc2_cdb2   ora_arc2_cdb2
25164 25164 ora_arc3_cdb2   ora_arc3_cdb2
25166 25166 ora_tt00_cdb2   ora_tt00_cdb2
25169 25169 ora_smco_cdb2   ora_smco_cdb2
25172 25172 ora_fbda_cdb2   ora_fbda_cdb2
25178 25178 ora_w000_cdb2   ora_w000_cdb2
25360 25360 ora_aqpc_cdb2   ora_aqpc_cdb2
25372 25372 ora_p000_cdb2   ora_p000_cdb2
25374 25374 ora_p001_cdb2   ora_p001_cdb2
25376 25376 ora_p002_cdb2   ora_p002_cdb2
```

```
25378 25378 ora_p003_cdb2   ora_p003_cdb2
25415 25415 ora_qm02_cdb2   ora_qm02_cdb2
25417 25417 ora_q001_cdb2   ora_q001_cdb2
25419 25419 ora_q002_cdb2   ora_q002_cdb2
25487 25487 ora_cjq0_cdb2   ora_cjq0_cdb2
26061 26061 ora_w001_cdb2   ora_w001_cdb2
26612 26612 oracle_26612_cd oraclecdb2 (LOCAL=NO)
26822 26822 ora_j000_cdb2   ora_j000_cdb2
26824 26824 ora_j001_cdb2   ora_j001_cdb2
26871 26871 ora_j002_cdb2   ora_j002_cdb2
26880 26880 grep            grep cdb2
$
```

2. What do you observe?
   Each Oracle process is running into a different OS process.
3. Do the same as in step 1, but using Oracle Database dictionary views.

```
$ sqlplus / as sysdba

Connected to:
Oracle Database 12c Enterprise Edition Release 12.1.0.0.2 -
64bit Production
With the Partitioning, OLAP, Advanced Analytics and Real
Application Testing options

SQL> set linesize 300
SQL> set pages 100
SQL> col username for a10

SQL> select p.spid, p.stid,p.pname,s.username,s.program
from v$process p, v$session s
where s.paddr = p.addr
order by 1;
2    3    4

SPID             STID             PNAME USERNAME   PROGRAM
------------------------ ------------------------ ----- --------
-- ------------------------------------------------
25075                25075                PMON
oracle@YOURSERVER (PMON)
25077                25077                PSP0
oracle@YOURSERVER (PSP0)
25089                25089                VKTM
oracle@YOURSERVER (VKTM)
```

```
25095                 25095                  GEN0
oracle@YOURSERVER (GEN0)
25097                 25097                  MMAN
oracle@YOURSERVER (MMAN)
25101                 25101                  DIAG
oracle@YOURSERVER (DIAG)
25103                 25103                  OFSD
oracle@YOURSERVER (OFSD)
25105                 25105                  DBRM
oracle@YOURSERVER (DBRM)
25107                 25107                  DIA0
oracle@YOURSERVER (DIA0)
25109                 25109                  DBW0
oracle@YOURSERVER (DBW0)
25111                 25111                  LGWR
oracle@YOURSERVER (LGWR)
25113                 25113                  CKPT
oracle@YOURSERVER (CKPT)
25115                 25115                  SMON
oracle@YOURSERVER (SMON)
25117                 25117                  RECO
oracle@YOURSERVER (RECO)
25119                 25119                  LREG
oracle@YOURSERVER (LREG)
25121                 25121                  MMON
oracle@YOURSERVER (MMON)
25123                 25123                  MMNL
oracle@YOURSERVER (MMNL)
25152                 25152                  RVWR
oracle@YOURSERVER (RVWR)
25153                 25153                  SYS
sqlplus@YOURSERVER (TNS V1-V3)
25156                 25156                  TMON
oracle@YOURSERVER (TMON)
25158                 25158                  ARC0
oracle@YOURSERVER (ARC0)
25160                 25160                  ARC1
oracle@YOURSERVER (ARC1)
25162                 25162                  ARC2
oracle@YOURSERVER (ARC2)
25164                 25164                  ARC3
oracle@YOURSERVER (ARC3)
25166                 25166                  TT00
oracle@YOURSERVER (TT00)
25169                 25169                  SMCO
oracle@YOURSERVER (SMCO)
```

```
25172                 25172                  FBDA
oracle@YOURSERVER (FBDA)
25178                 25178                  W000
oracle@YOURSERVER (W000)
25360                 25360                  AQPC
oracle@YOURSERVER (AQPC)
25415                 25415                  QM02
oracle@YOURSERVER (QM02)
25417                 25417                  Q001
oracle@YOURSERVER (Q001)
25419                 25419                  Q002
oracle@YOURSERVER (Q002)
25487                 25487                  CJQ0
oracle@YOURSERVER (CJQ0)
26061                 26061                  W001
oracle@YOURSERVER (W001)
26612                 26612                  SYSTEM
sqlplus@YOURSERVER (TNS V1-V3)
26952                 26952                  SYS
sqlplus@YOURSERVER (TNS V1-V3)
27028                 27028                  M000
oracle@YOURSERVER (M000)

37 rows selected.

SQL>
```

4. Modify your `SPFILE` to prepare for MPMT architecture.

```
SQL> alter system  set threaded_execution=true scope=spfile;

System altered.

SQL>
```

5. Still connected from the same session, shut down your `cdb2` instance, and restart it again.

```
SQL> shutdown immediate
Database closed.
Database dismounted.
ORACLE instance shut down.
SQL>
SQL> startup
ORA-01017: invalid username/password; logon denied
SQL>
```

6. Why are you getting the " ORA-01017: invalid username/password; logon denied" error?

This is because when using MPMT architecture, you have to use password authentication for `SYSDBA` operations.

7. Start up your `cdb2` instance.

```
SQL> connect / as sysdba
Connected.

SQL> startup
ORA-01081: cannot start already-running ORACLE - shut it down
first
SQL> alter database mount;

Database altered.

SQL> alter database open;

Database altered.

SQL>
```

8. Still connected from your SQL*Plus session, list the OS processes and OS threads used to run your `cdb2` instance using OS commands.

```
SQL> ! ps -eLo "pid tid comm args" |grep cdb2
27375 27375 ora_pmon_cdb2   ora_pmon_cdb2
27377 27377 ora_psp0_cdb2   ora_psp0_cdb2
27379 27379 ora_vktm_cdb2   ora_vktm_cdb2
27385 27385 ora_scmn_cdb2   ora_u004_cdb2
27385 27386 oracle          ora_u004_cdb2
27385 27387 ora_gen0_cdb2   ora_u004_cdb2
27385 27388 ora_mman_cdb2   ora_u004_cdb2
27385 27398 ora_dbrm_cdb2   ora_u004_cdb2
27385 27402 ora_lgwr_cdb2   ora_u004_cdb2
27385 27403 ora_ckpt_cdb2   ora_u004_cdb2
27385 27404 ora_smon_cdb2   ora_u004_cdb2
27385 27406 ora_lreg_cdb2   ora_u004_cdb2
27385 27588 ora_rvwr_cdb2   ora_u004_cdb2
27391 27391 ora_scmn_cdb2   ora_u005_cdb2
27391 27392 oracle          ora_u005_cdb2
27391 27393 ora_diag_cdb2   ora_u005_cdb2
27391 27399 ora_dia0_cdb2   ora_u005_cdb2
27391 27405 ora_reco_cdb2   ora_u005_cdb2
27391 27407 ora_mmon_cdb2   ora_u005_cdb2
27391 27408 ora_mmnl_cdb2   ora_u005_cdb2
27391 27409 ora_d000_cdb2   ora_u005_cdb2
27391 27410 ora_s000_cdb2   ora_u005_cdb2
27391 27411 ora_n000_cdb2   ora_u005_cdb2
```

```
27391 27575 oracle_27575_cd ora_u005_cdb2
27391 27589 ora_tmon_cdb2   ora_u005_cdb2
27391 27590 ora_arc0_cdb2   ora_u005_cdb2
27391 27591 ora_arc1_cdb2   ora_u005_cdb2
27391 27592 ora_arc2_cdb2   ora_u005_cdb2
27391 27593 ora_arc3_cdb2   ora_u005_cdb2
27391 27594 ora_tt00_cdb2   ora_u005_cdb2
27391 27595 ora_smco_cdb2   ora_u005_cdb2
27391 27596 ora_fbda_cdb2   ora_u005_cdb2
27391 27597 ora_w000_cdb2   ora_u005_cdb2
27391 27598 ora_aqpc_cdb2   ora_u005_cdb2
27391 27610 ora_p000_cdb2   ora_u005_cdb2
27391 27611 ora_p001_cdb2   ora_u005_cdb2
27391 27612 ora_p002_cdb2   ora_u005_cdb2
27391 27613 ora_p003_cdb2   ora_u005_cdb2
27391 27620 ora_qm02_cdb2   ora_u005_cdb2
27391 27621 ora_q001_cdb2   ora_u005_cdb2
27391 27622 ora_q002_cdb2   ora_u005_cdb2
27391 27655 ora_q003_cdb2   ora_u005_cdb2
27391 27669 ora_cjq0_cdb2   ora_u005_cdb2
27391 27899 ora_j001_cdb2   ora_u005_cdb2
27391 27900 ora_j002_cdb2   ora_u005_cdb2
27391 27901 ora_j003_cdb2   ora_u005_cdb2
27391 27902 ora_j004_cdb2   ora_u005_cdb2
27391 27903 ora_j005_cdb2   ora_u005_cdb2
27391 27904 ora_j006_cdb2   ora_u005_cdb2
27391 27905 ora_j007_cdb2   ora_u005_cdb2
27391 27906 ora_j008_cdb2   ora_u005_cdb2
27391 27910 ora_j009_cdb2   ora_u005_cdb2
27391 27911 ora_j010_cdb2   ora_u005_cdb2
27391 27912 ora_j011_cdb2   ora_u005_cdb2
27395 27395 ora_scmn_cdb2   ora_u006_cdb2
27395 27396 oracle          ora_u006_cdb2
27395 27397 ora_ofsd_cdb2   ora_u006_cdb2
27401 27401 ora_dbw0_cdb2   ora_dbw0_cdb2
27855 27855 vim             vim alert_cdb2.log
27918 27918 bash            /bin/bash -c  ps -eLo "pid tid comm
args" |grep cdb2
27920 27920 grep            grep cdb2


SQL>
SQL> ! ps -ef | grep cdb2
oracle   27375     1  0 05:03 ?        00:00:00 ora_pmon_cdb2
oracle   27377     1  0 05:03 ?        00:00:00 ora_psp0_cdb2
oracle   27379     1  0 05:03 ?        00:00:01 ora_vktm_cdb2
oracle   27385     1  1 05:03 ?        00:00:02 ora_u004_cdb2
oracle   27391     1 11 05:03 ?        00:00:25 ora_u005_cdb2
oracle   27395     1  0 05:03 ?        00:00:00 ora_u006_cdb2
oracle   27401     1  0 05:03 ?        00:00:00 ora_dbw0_cdb2
oracle   27855 31546  0 05:06 pts/6    00:00:00 vim
alert_cdb2.log
oracle   27955 26951  0 05:07 pts/5    00:00:00 /bin/bash -c  ps
-ef | grep cdb2
oracle   27957 27955  0 05:07 pts/5    00:00:00 grep cdb2

SQL>
```

9. Do the same using SQL commands.

```
SQL> set linesize 300
SQL> col username for a10
SQL> select p.spid, p.stid,p.pname,s.username,s.program
     from   v$process p, v$session s
     where  s.paddr = p.addr
     order by 1;
2    3    4

SPID            STID              PNAME USERNAME   PROGRAM
------------------------ ------------------------ ----- --------
-- ------------------------------------------------
27375                 27375                  PMON
oracle@YOURSERVER (PMON)
27377                 27377                  PSP0
oracle@YOURSERVER (PSP0)
27379                 27379                  VKTM
oracle@YOURSERVER (VKTM)
27385                 27388                  MMAN
oracle@YOURSERVER (MMAN)
27385                 27385                  SCMN
oracle@YOURSERVER (SCMN)
27385                 27387                  GEN0
oracle@YOURSERVER (GEN0)
27385                 27406                  LREG
oracle@YOURSERVER (LREG)
27385                 27588                  RVWR
oracle@YOURSERVER (RVWR)
```

```
27385                 27404                  SMON
oracle@YOURSERVER (SMON)
27385                 27403                  CKPT
oracle@YOURSERVER (CKPT)
27385                 27402                  LGWR
oracle@YOURSERVER (LGWR)
27385                 27398                  DBRM
oracle@YOURSERVER (DBRM)
27391                 27399                  DIA0
oracle@YOURSERVER (DIA0)
27391                 27591                  ARC1
oracle@YOURSERVER (ARC1)
27391                 27393                  DIAG
oracle@YOURSERVER (DIAG)
27391                 27391                  SCMN
oracle@YOURSERVER (SCMN)
27391                 27620                  QM02
oracle@YOURSERVER (QM02)
27391                 27621                  Q001
oracle@YOURSERVER (Q001)
27391                 27622                  Q002
oracle@YOURSERVER (Q002)
27391                 27405                  RECO
oracle@YOURSERVER (RECO)
27391                 27669                  CJQ0
oracle@YOURSERVER (CJQ0)
27391                 27407                  MMON
oracle@YOURSERVER (MMON)
27391                 27408                  MMNL
oracle@YOURSERVER (MMNL)
27391                 27655                  Q003
oracle@YOURSERVER (Q003)
27391                 27589                  TMON
oracle@YOURSERVER (TMON)
27391                 27590                  ARC0
oracle@YOURSERVER (ARC0)
27391                 27575                     SYS
sqlplus@YOURSERVER (TNS V1-V3)
27391                 27592                  ARC2
oracle@YOURSERVER (ARC2)
27391                 27593                  ARC3
oracle@YOURSERVER (ARC3)
27391                 27594                  TT00
oracle@YOURSERVER (TT00)
27391                 27595                  SMCO
oracle@YOURSERVER (SMCO)
```

```
27391                 27596                   FBDA
oracle@YOURSERVER (FBDA)
27391                 27597                   W000
oracle@YOURSERVER (W000)
27391                 27598                   AQPC
oracle@YOURSERVER (AQPC)
27395                 27397                   OFSD
oracle@YOURSERVER (OFSD)
27395                 27395                   SCMN
oracle@YOURSERVER (SCMN)
27401                 27401                   DBW0
oracle@YOURSERVER (DBW0)


37 rows selected.


SQL>
```

10. What do you observe?
    Now, many Oracle processes run as threads inside a small amount of OS processes.
11. Establish a remote connection to your `cdb2` instance using SQL*Plus, and list again OS processes and threads used to run all Oracle processes.

```
SQL> connect sys/oracle_4U@cdb2 as sysdba
Connected.
SQL> set linesize 300
SQL> col username for a10
SQL> select p.spid, p.stid,p.pname,s.username,s.program
     from   v$process p, v$session s
     where  s.paddr = p.addr
     order by 1;
2    3    4

SPID             STID             PNAME USERNAME   PROGRAM
------------------------ ------------------------ ----- --------
-- ------------------------------------------------
27375                 27375                   PMON
oracle@YOURSERVER (PMON)
27377                 27377                   PSP0
oracle@YOURSERVER (PSP0)
27379                 27379                   VKTM
oracle@YOURSERVER (VKTM)
27385                 27388                   MMAN
oracle@YOURSERVER (MMAN)
27385                 27385                   SCMN
oracle@YOURSERVER (SCMN)
```

```
27385                 27387                  GEN0
oracle@YOURSERVER (GEN0)
27385                 27406                  LREG
oracle@YOURSERVER (LREG)
27385                 27588                  RVWR
oracle@YOURSERVER (RVWR)
27385                 27404                  SMON
oracle@YOURSERVER (SMON)
27385                 27403                  CKPT
oracle@YOURSERVER (CKPT)
27385                 27402                  LGWR
oracle@YOURSERVER (LGWR)
27385                 27398                  DBRM
oracle@YOURSERVER (DBRM)
27391                 27399                  DIA0
oracle@YOURSERVER (DIA0)
27391                 27591                  ARC1
oracle@YOURSERVER (ARC1)
27391                 27393                  DIAG
oracle@YOURSERVER (DIAG)
27391                 27391                  SCMN
oracle@YOURSERVER (SCMN)
27391                 27598                  AQPC
oracle@YOURSERVER (AQPC)
27391                 27620                  QM02
oracle@YOURSERVER (QM02)
27391                 27621                  Q001
oracle@YOURSERVER (Q001)
27391                 27622                  Q002
oracle@YOURSERVER (Q002)
27391                 27405                  RECO
oracle@YOURSERVER (RECO)
27391                 27669                  CJQ0
oracle@YOURSERVER (CJQ0)
27391                 27407                  MMON
oracle@YOURSERVER (MMON)
27391                 27408                  MMNL
oracle@YOURSERVER (MMNL)
27391                 27655                  Q003
oracle@YOURSERVER (Q003)
27391                 27589                  TMON
oracle@YOURSERVER (TMON)
27391                 27590                  ARC0
oracle@YOURSERVER (ARC0)
27391                 27592                  ARC2
oracle@YOURSERVER (ARC2)
```

```
27391                 27593                  ARC3
oracle@YOURSERVER (ARC3)
27391                 27594                  TT00
oracle@YOURSERVER (TT00)
27391                 27595                  SMCO
oracle@YOURSERVER (SMCO)
27391                 27596                  FBDA
oracle@YOURSERVER (FBDA)
27391                 27597                  W000
oracle@YOURSERVER (W000)
27395                 27397                  OFSD
oracle@YOURSERVER (OFSD)
27395                 27395                  SCMN
oracle@YOURSERVER (SCMN)
27401                 27401                  DBW0
oracle@YOURSERVER (DBW0)
28009                 28009                      SYS
sqlplus@YOURSERVER (TNS V1-V3)

37 rows selected.

SQL>
```

12. What do you observe?
    Your foreground process that runs your SQL*Plus connection is using one OS process, and not threads.
13. How would you make sure foreground processes are using threads?

```
SQL> exit
$
```

a. View the tnsnames.ora file content.

```
$ cat $ORACLE_HOME/network/admin/tnsnames.ora
# tnsnames.ora Network Configuration File:
/u01/app/oracle/product/12.1.0/dbhome_1/network/admin/tnsnames.o
ra
# Generated by Oracle configuration tools.


PDB_ORCL2 =
  (DESCRIPTION =
    (ADDRESS = (PROTOCOL = TCP)(HOST = host01.example.com)(PORT
= 1521))
    (CONNECT_DATA =
      (SERVER = DEDICATED)
      (SERVICE_NAME = pdb_orcl2)
    )
  )
```

```
PDB2_2 =
  (DESCRIPTION =
    (ADDRESS = (PROTOCOL = TCP)(HOST = host01.example.com)(PORT
= 1521))
    (CONNECT_DATA =
      (SERVER = DEDICATED)
      (SERVICE_NAME = pdb2_2)
    )
  )

PDB2_1 =
  (DESCRIPTION =
    (ADDRESS = (PROTOCOL = TCP)(HOST = host01.example.com)(PORT
= 1521))
    (CONNECT_DATA =
      (SERVER = DEDICATED)
      (SERVICE_NAME = pdb2_1)
    )
  )

CDB2 =
  (DESCRIPTION =
    (ADDRESS = (PROTOCOL = TCP)(HOST = host01.example.com)(PORT
= 1521))
    (CONNECT_DATA =
      (SERVER = DEDICATED)
      (SERVICE_NAME = cdb2)
    )
  )


CDB1 =
  (DESCRIPTION =
    (ADDRESS = (PROTOCOL = TCP)(HOST = host01.example.com)(PORT
= 1521))
    (CONNECT_DATA =
      (SERVER = DEDICATED)
      (SERVICE_NAME = cdb1)
    )
  )

ORCL2 =
  (DESCRIPTION =
    (ADDRESS = (PROTOCOL = TCP)(HOST = host01.example.com)(PORT
= 1521))
    (CONNECT_DATA =
      (SERVER = DEDICATED)
      (SERVICE_NAME = orcl2)
    )
  )

PDB1_1 =
  (DESCRIPTION =
    (ADDRESS = (PROTOCOL = TCP)(HOST = host01.example.com)(PORT
= 1521))
    (CONNECT_DATA =
      (SERVER = DEDICATED)
      (SERVICE_NAME = pdb1_1)
    )
  )
ORCL =
  (DESCRIPTION =
    (ADDRESS = (PROTOCOL = TCP)(HOST = host01.example.com)(PORT
= 1521))
    (CONNECT_DATA =
      (SERVER = DEDICATED)
      (SERVICE_NAME = orcl)
    )
  )
$
```

b. View the `listener.ora` file content.

```
$ cat $ORACLE_HOME/network/admin/listener.ora
# listener.ora Network Configuration File:
/u01/app/oracle/product/12.1.0/dbhome_1/network/admin/listener.o
ra
# Generated by Oracle configuration tools.

LISTENER =
  (DESCRIPTION_LIST =
    (DESCRIPTION =
      (ADDRESS = (PROTOCOL = IPC)(KEY = EXTPROC1521))
      (ADDRESS = (PROTOCOL = TCP)(HOST =
host01.example.com)(PORT = 1521))
    )
  )
```

```
ADR_BASE_LISTENER = /u01/app/oracle
$
```

c. Keep a copy of the `listener.ora` file.

```
$ cp $ORACLE_HOME/network/admin/listener.ora
$ORACLE_HOME/network/admin/listener.ora.bak
$
```

d. Add the following parameter in the `listener.ora` file.

```
$ echo DEDICATED_THROUGH_BROKER_LISTENER=on >>
$ORACLE_HOME/network/admin/listener.ora
$
```

e. Check that the `listener.ora` file is adequately modified.

```
$ cat $ORACLE_HOME/network/admin/listener.ora
# listener.ora Network Configuration File:
/u01/app/oracle/product/12.1.0/dbhome_1/network/admin/listener.o
ra
# Generated by Oracle configuration tools.

LISTENER =
  (DESCRIPTION_LIST =
    (DESCRIPTION =
      (ADDRESS = (PROTOCOL = IPC)(KEY = EXTPROC1521))
      (ADDRESS = (PROTOCOL = TCP)(HOST =
host01.example.com)(PORT = 1521))
    )
  )

ADR_BASE_LISTENER = /u01/app/oracle

DEDICATED_THROUGH_BROKER_LISTENER=on
$
```

f. Restart the listener.

```
$ lsnrctl stop

LSNRCTL for Linux: Version 12.1.0.0.2 - Production on 04-SEP-
2012 13:01:23

Copyright (c) 1991, 2012, Oracle.  All rights reserved.

Connecting to
(DESCRIPTION=(ADDRESS=(PROTOCOL=IPC)(KEY=EXTPROC1521)))
The command completed successfully
$
```

```
$ lsnrctl start

LSNRCTL for Linux: Version 12.1.0.0.2 - Production on 04-SEP-
2012 13:01:29

Copyright (c) 1991, 2012, Oracle.  All rights reserved.

Starting /u01/app/oracle/product/12.1.0/dbhome_1/bin/tnslsnr:
please wait...

TNSLSNR for Linux: Version 12.1.0.0.2 - Production
System parameter file is
/u01/app/oracle/product/12.1.0/dbhome_1/network/admin/listener.o
ra
Log messages written to
/u01/app/oracle/diag/tnslsnr/host01/listener/alert/log.xml
Listening on:
(DESCRIPTION=(ADDRESS=(PROTOCOL=ipc)(KEY=EXTPROC1521)))
Listening on:
(DESCRIPTION=(ADDRESS=(PROTOCOL=tcp)(HOST=host01.example.com)(PO
RT=1521)))

Connecting to
(DESCRIPTION=(ADDRESS=(PROTOCOL=IPC)(KEY=EXTPROC1521)))
STATUS of the LISTENER
------------------------
Alias                     LISTENER
Version                   TNSLSNR for Linux: Version 12.1.0.0.2
- Production
Start Date                04-SEP-2012 13:01:29
Uptime                    0 days 0 hr. 0 min. 0 sec
Trace Level               off
Security                  ON: Local OS Authentication
SNMP                      OFF
Listener Parameter File
/u01/app/oracle/product/12.1.0/dbhome_1/network/admin/listener.o
ra
Listener Log File
/u01/app/oracle/diag/tnslsnr/host01/listener/alert/log.xml
Listening Endpoints Summary...
  (DESCRIPTION=(ADDRESS=(PROTOCOL=ipc)(KEY=EXTPROC1521)))

(DESCRIPTION=(ADDRESS=(PROTOCOL=tcp)(HOST=host01.example.com)(PO
RT=1521)))
The listener supports no services
The command completed successfully
$
```

g. Set the `LOCAL_LISTENER` parameter to `cdb2`.

```
$ sqlplus sys/oracle_4U@cdb2 as sysdba

Connected to:
Oracle Database 12c Enterprise Edition Release 12.1.0.0.2 -
64bit Production
With the Partitioning, OLAP, Advanced Analytics and Real
Application Testing options

SQL> show parameter local_listener

NAME                                 TYPE        VALUE
------------------------------------ ----------- --------------
local_listener                       string       LISTENER_CDB2

SQL> alter system set local_listener=cdb2 scope=both;

System altered.

SQL> show parameter local_listener

NAME                                 TYPE        VALUE
------------------------------------ ----------- --------------
local_listener                       string      CDB2

SQL>
```

14. Check that what you did is working.
Tip: Looking at the `sqlplus` program, it should be run as a thread in an existing OS process.

```
SQL> connect sys/oracle_4U@cdb2 as sysdba
Connected.

SQL> set linesize 300
SQL> set pages 100
SQL> col username for a10
SQL> select p.spid, p.stid,p.pname,s.username,s.program
     from   v$process p, v$session s
     where  s.paddr = p.addr
     order by 1;
2    3    4
```

```
SPID            STID              PNAME USERNAME   PROGRAM
------------------------ ------------------------ ----- --------
-- ------------------------------------------------
27375                 27375                  PMON
oracle@YOURSERVER (PMON)
27377                 27377                  PSP0
oracle@YOURSERVER (PSP0)
27379                 27379                  VKTM
oracle@YOURSERVER (VKTM)
27385                 27387                  GEN0
oracle@YOURSERVER (GEN0)
27385                 27385                  SCMN
oracle@YOURSERVER (SCMN)
27385                 27398                  DBRM
oracle@YOURSERVER (DBRM)
27385                 27588                  RVWR
oracle@YOURSERVER (RVWR)
27385                 27406                  LREG
oracle@YOURSERVER (LREG)
27385                 27404                  SMON
oracle@YOURSERVER (SMON)
27385                 27402                  LGWR
oracle@YOURSERVER (LGWR)
27385                 27403                  CKPT
oracle@YOURSERVER (CKPT)
27385                 27388                  MMAN
oracle@YOURSERVER (MMAN)
27391                 27399                  DIA0
oracle@YOURSERVER (DIA0)
27391                 27391                  SCMN
oracle@YOURSERVER (SCMN)
27391                 27393                  DIAG
oracle@YOURSERVER (DIAG)
27391                 28582                  SYS
sqlplus@YOURSERVER (TNS V1-V3)
27391                 27405                  RECO
oracle@YOURSERVER (RECO)
27391                 27622                  Q002
oracle@YOURSERVER (Q002)
27391                 27621                  Q001
oracle@YOURSERVER (Q001)
27391                 27669                  CJQ0
oracle@YOURSERVER (CJQ0)
27391                 27407                  MMON
oracle@YOURSERVER (MMON)
```

```
27391                 27408                  MMNL
oracle@YOURSERVER (MMNL)
27391                 27589                  TMON
oracle@YOURSERVER (TMON)
27391                 27590                  ARC0
oracle@YOURSERVER (ARC0)
27391                 27591                  ARC1
oracle@YOURSERVER (ARC1)
27391                 27592                  ARC2
oracle@YOURSERVER (ARC2)
27391                 27593                  ARC3
oracle@YOURSERVER (ARC3)
27391                 27594                  TT00
oracle@YOURSERVER (TT00)
27391                 27595                  SMCO
oracle@YOURSERVER (SMCO)
27391                 27596                  FBDA
oracle@YOURSERVER (FBDA)
27391                 27597                  W000
oracle@YOURSERVER (W000)
27391                 27598                  AQPC
oracle@YOURSERVER (AQPC)
27391                 27620                  QM02
oracle@YOURSERVER (QM02)
27395                 27395                  SCMN
oracle@YOURSERVER (SCMN)
27395                 27397                  OFSD
oracle@YOURSERVER (OFSD)
27401                 27401                  DBW0
oracle@YOURSERVER (DBW0)

36 rows selected.

SQL>
```

15. Revert to the non-MPMT architecture. Keep the `cdb2` database closed for the moment so as to preserve resources on the machine.

```
SQL> alter system set threaded_execution=false scope=spfile;

System altered.

SQL> shutdown immediate
Database closed.
Database dismounted.
ORACLE instance shut down.
```

```
SQL> exit
$
```

Practices for Lesson 18: Tables, Indexes and Online Operations

Chapter 18

Practices for Lesson 18

Practices Overview

In the practice for this lesson, you will use the invisible/visible table columns and the Advanced Row compression.

Practice 18-1: Using Invisible Table Columns

Overview

In this practice, you will create a table with invisible columns. These columns are not necessarily useful for the current application but might become useful in a later application release.

Tasks

1. Make sure you are in the `~/labs/Tables` directory and your environment points to the `orcl` instance.

```
$ cd ~/labs/Tables
$ . oraenv
ORACLE_SID = [orcl] ? orcl
The Oracle base remains unchanged with value /u01/app/oracle
$
```

2. First start the instance up.

```
$ sqlplus / as sysdba

Connected to an idle instance.
SQL> startup
ORACLE instance started.

Total System Global Area  400846848 bytes
Fixed Size                  2271568 bytes
Variable Size             339740336 bytes
Database Buffers           50331648 bytes
Redo Buffers                8503296 bytes
Database mounted.
Database opened.
SQL>
```

3. Run the `invisible_setup.sql` script to create a new user `STATS`

```
SQL> @invisible_setup.sql
Connected.
DROP USER stats CASCADE
          *
ERROR at line 1:
ORA-01918: user 'STATS' does not exist

User created.



Grant succeeded.
SQL>
```

4. The STATS user creates a new table CENSUS. The table structure contains three columns, GENDER, COUNTRY, NUMBER and an invisible column REGION. The REGION column is not used by the application yet, but might become useful in a future application release.

```
SQL> CREATE TABLE stats.census (gender VARCHAR2(10), country
CHAR(2), nb NUMBER, region VARCHAR2(20) INVISIBLE );

Table created.

SQL>
```

5. Describe the structure of the CENSUS table.
   a. You see that the invisible column does not appear.

```
SQL> DESC stats.census
 Name                        Null?    Type
 -------------------------- -------- --------------------------
 GENDER                               VARCHAR2(10)
 COUNTRY                              CHAR(2)
 NB                                   NUMBER

SQL>
```

   b. Describe the structure of the CENSUS table so that the invisible column appears.

```
SQL> SET COLINVISIBLE ON
SQL> DESC stats.census
 Name                        Null?    Type
 -------------------------- -------- --------------------------
 GENDER                               VARCHAR2(10)
 COUNTRY                              CHAR(2)
 NB                                   NUMBER
 REGION (INVISIBLE)                   VARCHAR2(20)

SQL>
```

6. Insert rows into the CENSUS table.
   a. You cannot insert a value for the invisible column unless you define it in the projection list.

```
SQL> INSERT INTO stats.census VALUES ('BOY','BR', 10000,
'BAHIA');
INSERT INTO stats.census VALUES ('BOY','BR', 10000, 'BAHIA')
                  *
ERROR at line 1:
ORA-00913: too many values

SQL>
```

b. Insert a row with values for the three visible columns.

```
SQL> INSERT INTO stats.census VALUES ('BOY','BR',100000);

1 row created.

SQL> COMMIT;

Commit complete.

SQL> SELECT * FROM stats.census;

GENDER     CO         NB
---------- -- ----------
BOY        BR     100000

SQL>
```

c. Insert a row with a value for the invisible column.

```
SQL> INSERT INTO stats.census (gender, country, nb, region)
VALUES ('BOY','BR', 35000,'BAHIA');

1 row created.

SQL> COMMIT;

Commit complete.

SQL> SELECT gender, country, nb, region FROM stats.census;

GENDER     CO         NB REGION
---------- -- ---------- --------------------
BOY        BR     100000
BOY        BR      35000 BAHIA

SQL>
```

7. Make the invisible column visible.

```
SQL> ALTER TABLE stats.census MODIFY (region VISIBLE);

Table altered.

SQL> SELECT * FROM stats.census;

GENDER     CO         NB REGION
```

```
---------- -- ---------- --------------------
BOY        BR     100000
BOY        BR      35000 BAHIA

SQL> EXIT
$
```

Practice 18-2: Advanced Row Compression

Overview

In this practice, you will use the row store advanced compression. You will compare the storage requirements between a compressed table and an uncompressed table and verify the compression ratio between different tables with the same number of rows.

Tasks

1. Connect as SH. You first create two copies of the SH.SALES table, the first being compressed and the second being uncompressed.

```
SQL> CONNECT sh/oracle_4U
Connected.
SQL> drop table sales_nocompress purge;

Table dropped.

SQL> drop table sales_advcompress purge;

Table dropped.

SQL> set echo on
SQL> set timing on
SQL> CREATE TABLE sales_nocompress AS SELECT * FROM sales;

Table created.

Elapsed: 00:00:02.13
SQL> CREATE TABLE sales_advcompress ROW STORE COMPRESS ADVANCED
AS SELECT * FROM sales where 1=0;

Table created.

Elapsed: 00:00:00.05
SQL>

SQL> SELECT count(*) FROM sales_nocompress;

  COUNT(*)
----------
    918843

Elapsed: 00:00:00.44
SQL> SELECT count(*) FROM sales_advcompress;
```

```
  COUNT(*)
----------
         0

Elapsed: 00:00:00.00
SQL>
```

2. Load the SALES_ADVCOMPRESS table using the following PL/SQL block.

```
SQL> declare
commit_after integer := 0 ;
loop_variable integer ;
 cursor c_sales is
 select prod_id, cust_id, time_id, channel_id, promo_id,
quantity_sold, amount_sold
 from sales ;
 begin
 for r_sales in c_sales
 loop
 if commit_after = 0
 then
 loop_variable := 0 ;
 commit_after := round(dbms_random.value(1,1)) ;
 end if ;
 insert into sales_advcompress
  (prod_id, cust_id, time_id, channel_id, promo_id,
quantity_sold, amount_sold)
 values
  (r_sales.prod_id, r_sales.cust_id, r_sales.time_id,
r_sales.channel_id,
  r_sales.promo_id, r_sales.quantity_sold, r_sales.amount_sold)
;
  if loop_variable = commit_after
  then
  commit ;
  commit_after := 0 ;
  end if ;
  loop_variable := loop_variable + 1 ;
  end loop ;
 end ;
/
```

```
  2    3    4    5    6    7    8    9   10   11   12   13   14
15   16   17   18   19   20   21   22   23   24   25   26   27
28


PL/SQL procedure successfully completed.


Elapsed: 00:01:37.67
SQL>
```

3. Verify the number of rows in the two tables.

```
SQL> SELECT count(*) FROM sales_nocompress;

  COUNT(*)
----------
    918843

Elapsed: 00:00:00.02
SQL> SELECT count(*) FROM sales_advcompress;

  COUNT(*)
----------
    918843

Elapsed: 00:00:00.31
SQL>
```

4. Now you can compare the storage requirements between the two tables you just created.

```
SQL> COL segment_name FORMAT A30
SQL> select segment_name, sum(bytes)/1024/1024 mb
from dba_segments
where owner = 'SH' and segment_name in
('SALES_NOCOMPRESS','SALES_ADVCOMPRESS')
group by segment_name order by segment_name;
2    3    4

SEGMENT_NAME                   MB
----------------------- ----------
SALES_ADVCOMPRESS               14
SALES_NOCOMPRESS                36



Elapsed: 00:00:00.46
SQL>
```

5. Use the `DBMS_COMPRESSION` package to get the compression ratio of the `SALES_ADVCOMPRESS` table.

```
SQL> set serveroutput on
SQL> DECLARE
 blkcnt_cmp pls_integer;
 blkcnt_uncmp pls_integer;
 row_cmp pls_integer;
 row_uncmp pls_integer;
 cmp_ratio pls_integer;
 comptype_str varchar2(100);
 BEGIN
 DBMS_COMPRESSION.GET_COMPRESSION_RATIO (
      'USERS', 'SH', 'SALES_ADVCOMPRESS', NULL,
DBMS_COMPRESSION.COMP_ADVANCED,
      blkcnt_cmp, blkcnt_uncmp, row_cmp, row_uncmp, cmp_ratio,
comptype_str,1000,1);
 DBMS_OUTPUT.PUT_LINE('Table = SH.SALES_ADVCOMPRESS');
 DBMS_OUTPUT.PUT_LINE('Block count compressed = ' ||
blkcnt_cmp);
 DBMS_OUTPUT.PUT_LINE('Block count uncompressed = ' ||
blkcnt_uncmp);
 DBMS_OUTPUT.PUT_LINE('Row count per block compressed = ' ||
row_cmp);
 DBMS_OUTPUT.PUT_LINE('Row count per block uncompressed = ' ||
row_uncmp);
 DBMS_OUTPUT.PUT_LINE('Compression type = ' || comptype_str);
 DBMS_OUTPUT.PUT_LINE('Compression ratio =
'||blkcnt_uncmp/blkcnt_cmp||' to 1');
 DBMS_OUTPUT.PUT_LINE('Compression ratio org= '||cmp_ratio);
 END;
/
  2    3    4    5    6    7    8    9   10   11   12   13   14
15   16   17   18   19   20   21
Table = SH.SALES_ADVCOMPRESS
Block count compressed = 1628
Block count uncompressed = 4379
Row count per block compressed = 558
Row count per block uncompressed = 207
Compression type = "Compress Advanced"
Compression ratio = 2.68980343980343980343980343980343980344 to
1
Compression ratio org= 3


PL/SQL procedure successfully completed.
```

```

Elapsed: 00:00:13.19
SQL>
```

6. Analyze the tables and note the number of rows compressed in the compressed table and compare the ratio of compression with the non compressed table.

```
SQL> ANALYZE TABLE sh.sales_nocompress COMPUTE STATISTICS;

Table analyzed.

Elapsed: 00:00:05.30
SQL> ANALYZE TABLE sh.sales_advcompress COMPUTE STATISTICS;

Table analyzed.

Elapsed: 00:00:05.76
SQL>
```

```
SQL> COL object_name format A20
SQL> CONNECT / AS SYSDBA
Connected.
SQL> SELECT object_name ,  AVGROWSIZE_NC, AVGROWSIZE_C, NBLK_NC,
NBLK_ADVANCED, NROWS_NC, NROWS_ADVANCED
FROM sys.compression_stat$ c , DBA_OBJECTS o
WHERE c.obj# = o.object_id;
2    3

OBJECT_NAME          AVGROWSIZE_NC AVGROWSIZE_C    NBLK_NC
-------------------- ------------- ------------ ----------
NBLK_ADVANCED   NROWS_NC NROWS_ADVANCED
------------- ---------- --------------
SALES_ADVCOMPRESS             32.8        10.43          8
         1732      35010         883833

Elapsed: 00:00:00.27

SQL> SELECT avg_row_len, num_rows FROM dba_tables WHERE
table_name='SALES_NOCOMPRESS';



AVG_ROW_LEN   NUM_ROWS
----------- ----------
         33     918843
```

```
SQL>
```

The non compressed table contains 918843 uncompressed rows whereas the compressed table contains 883833 compressed rows and only 35010 uncompressed rows with an average row size 3 times smaller.

7. Cleanup the tables.

```
SQL> DROP TABLE sh.sales_nocompress PURGE;

Table dropped.

Elapsed: 00:00:00.62
SQL> DROP TABLE sh.sales_advcompress PURGE;

Table dropped.

Elapsed: 00:00:00.13

SQL> EXIT
$
```

Practices for Lesson 19: ADR and Network Enhancements

Chapter 19

Practices for Lesson 19

Practices Overview

In this practice, you will familiarize yourself with viewing an ADR DDL log file and content.

Practice 19-1: Viewing ADR DDL Log File

Overview

In this practice, you will find and view the ADR DDL log file.

Tasks

1. Set the `ENABLE_DDL_LOGGING` instance parameter to `TRUE` to activate the DDL logging.

```
$ . oraenv
ORACLE_SID = [orcl] ? orcl
The Oracle base remains unchanged with value /u01/app/oracle
$ sqlplus / as sysdba

Connected to:
Oracle Database 12c Enterprise Edition Release 12.1.0.0.2 -
64bit Production
With the Partitioning, OLAP, Data Mining and Real Application
Testing options

SQL> ALTER SYSTEM SET enable_ddl_logging=TRUE SCOPE=both;

System altered.

SQL> EXIT
$
```

2. The administrator is performing various administration tasks requiring DDL statements.

```
$ sqlplus system/oracle_4U

Connected to:
Oracle Database 12c Enterprise Edition Release 12.1.0.0.2 -
64bit Production
With the Partitioning, OLAP, Data Mining and Real Application
Testing options

SQL*Plus: Release 12.1.0.0.2 Production on Fri Aug 24 11:09:58
2012

Copyright (c) 1982, 2012, Oracle.  All rights reserved.

SQL> CREATE TABLE scott.test1 (c NUMBER);

Table created.

SQL> CREATE TABLE scott.test2 (c VARCHAR2(10));
```

```

Table created.

SQL> CREATE USER new_u1 IDENTIFIED BY oracle_4U;

User created.

SQL> DROP USER new_u1 CASCADE;

User dropped.

SQL> DROP TABLE scott.test2;

Table dropped.

SQL> DROP TABLE scott.test1;

Table dropped.
SQL> EXIT
$
```

3. Check the existence of the DDL log file in the ADR directory.

```
$ cd /u01/app/oracle/diag/rdbms/orcl/orcl/log/ddl
$ ls -ltr
total 4
-rw-r----- 1 oracle oinstall 1257 Sep 10 05:19 log.xml
$
```

4. Use ADRCI utility to view the content of the DLL log file.
   a. Launch the `adrci` utility and execute the `SHOW LOG` command.

```
$ adrci
ADRCI: Release 12.1.0.0.2 - Production on Fri Aug 24 11:50:36
2012

Copyright (c) 1982, 2012, Oracle and/or its affiliates.  All
rights reserved.

ADR base = "/u01/app/oracle"
adrci> SHOW LOG
```

   b. A vi editor page is displayed.

```
2012-09-10 05:18:58.322000 +00:00
CREATE TABLE scott.test1 (c NUMBER)
2012-09-10 05:19:02.625000 +00:00
CREATE TABLE scott.test2 (c VARCHAR2(10))
```

```
2012-09-10 05:19:36.456000 +00:00
DROP USER new_u1 CASCADE
2012-09-10 05:19:38.594000 +00:00
DROP TABLE scott.test2
2012-09-10 05:19:41.920000 +00:00
DROP TABLE scott.test1
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
"/tmp/utsout_22394_14071_1.ado" 20L, 675C
```

c. To quit the editor, use the `:q` vi command.

```
:q
ADR Home = /u01/app/oracle/diag/rdbms/orcl/orcl:
****************************************************************
*********
Output the results to file: /tmp/utsout_22424_13977_2.ado
adrci> EXIT
$
```

5. View the content of the DDL log file with a UNIX command.

```
$ more /u01/app/oracle/diag/rdbms/orcl/orcl/log/ddl/log.xml
<msg time='2012-09-10T05:18:58.322+00:00' org_id='oracle'
comp_id='rdbms'
 msg_id='opiexe:4142:2946163730' type='UNKNOWN' group='diag_adl'
 level='16' host_id='EDRSR7P1' host_addr='139.185.35.107'
 version='1'>
 <txt>CREATE TABLE scott.test1 (c NUMBER)
 </txt>
</msg>
<msg time='2012-09-10T05:19:02.625+00:00' org_id='oracle'
comp_id='rdbms'
 msg_id='opiexe:4142:2946163730' type='UNKNOWN' group='diag_adl'
 level='16' host_id='EDRSR7P1' host_addr='139.185.35.107'>
 <txt>CREATE TABLE scott.test2 (c VARCHAR2(10))
 </txt>
</msg>
<msg time='2012-09-10T05:19:36.456+00:00' org_id='oracle'
comp_id='rdbms'
 msg_id='opiexe:4142:2946163730' type='UNKNOWN' group='diag_adl'
 level='16' host_id='EDRSR7P1' host_addr='139.185.35.107'>
 <txt>DROP USER new_u1 CASCADE
 </txt>
</msg>
<msg time='2012-09-10T05:19:38.594+00:00' org_id='oracle'
comp_id='rdbms'
 msg_id='opiexe:4142:2946163730' type='UNKNOWN' group='diag_adl'
 level='16' host_id='EDRSR7P1' host_addr='139.185.35.107'>
 <txt>DROP TABLE scott.test2
 </txt>
</msg>
<msg time='2012-09-10T05:19:41.920+00:00' org_id='oracle'
comp_id='rdbms'
 msg_id='opiexe:4142:2946163730' type='UNKNOWN' group='diag_adl'
 level='16' host_id='EDRSR7P1' host_addr='139.185.35.107'>
 <txt>DROP TABLE scott.test1
 </txt>
</msg>
$
```

Practices for Lesson 20: Oracle Data Pump, SQL*Loader, and External Tables

Chapter 20

Practices for Lesson 20

Practices Overview

In these practices, you will create a new non-CDB named `orcl3` with DBCA. After the non-CDB `orcl3` creation is completed, you will perform a FULL TRANSPORTABLE from the non-CDB `orcl` into the non-CDB `orcl3`.

You will also perform a data load by using SQL*Loader Express Mode.

Practice 20-1: Creating a New non-CDB `orcl3`

Overview

In this practice, you will create a new non-CDB named `orcl3` with DBCA.

Assumptions

The created non-CDB `orcl` already exists.

Tasks

1. Create a non-CDB named `orcl3` using DBCA following the steps below. Be sure to select **Unicode** (**AL32UTF8**) as the character set.

```
$ export ORACLE_HOME=/u01/app/oracle/product/12.1.0/dbhome_1
$ dbca
```

Step	Window/Page Description	Choices or Values
a.	Step 1: Database Operation	Select “Create Database”. Click Next.
b.	Step 2: Creation Mode	Select “**Advanced Mode**”. Click Next.
c.	Step 3: Database Template	Select “**General Purpose or Transaction Processing**”. Click Next.
d.	Step 4: Database Identification	Enter Global Database Name: **`orcl3`** SID: **`orcl3`** **DON’T SELECT “Create as Container Database.** Click Next.
e.	Step 5: Management Options	ConfigureEnterprise Manager (EM) Database Express Click Next
f.	Step 6: Database Credentials	Select “Use same Administrative password…” Enter: Password: **`oracle_4U`** Confirm password: **`oracle_4U`** Click Next.
g.	Step 7: Network Configuration	Click Next.
h.	Step 8: Storage Locations	Confirm Storage type is “**File System.**” Select “**Use Common Location for All Database Files**”. Click Next.
i.	Step 9: Database Options	**DON’T SELECT** “**Sample Schemas**” Click Next.

Step	Window/Page Description	Choices or Values
j.	Step 10: Initialization Parameters	Select “Character Sets”. Select “Use **Unicode** (**AL32UTF8**)”. Click Next.
k.	Step 11: Creation Options	Select “Create Database” . Click Next.
l.	Step 12: Pre Requisite Checks	Click Next.
m.	Step 13: Summary	Click **Finish.**
n.	Step 14: Progress Page	On the Database Configuration Assistant page (for password management) click **Exit**. Click **Close.**

2. While the creation of the new `orcl3` database is processing, you can start preparing the source `orcl` database for FULL TRANSPORTABLE exportation. Move to Practice 20-2.

Practice 20-2: Exporting/Importing Databases in FULL TRANSPORTABLE Mode

Overview

In this practice, you export the `orcl` database and import into the `orcl3` database using the Oracle Data Pump FULL TRANSPORTABLE feature.

Tasks

1. Connect to the source database `orcl`.

```
$ . oraenv
ORACLE_SID = [cdb2] ? orcl
The Oracle base remains unchanged with value /u01/app/oracle

$ sqlplus / as sysdba

Connected to:
Oracle Database 12c Enterprise Edition Release 12.1.0.0.2 -
64bit Production
With the Partitioning, OLAP, Data Mining and Real Application
Testing options

SQL>
```

2. Put the user-defined tablespaces in the source database `orcl` in read-only mode.
   a. Create new tablespace to be transported with other tablespaces into `orcl3` database.

```
SQL> CREATE TABLESPACE test DATAFILE
'/u01/app/oracle/oradata/orcl/test01.dbf' size 5M;

Tablespace created.

SQL>
```

   b. Create a table `HR.TESTTAB` in the `TEST` tablespace, insert rows, and commit. You will check at the end of the FULL TRANSPORTABLE operation from `orcl` to `orcl3` database if the `HR.TESTTAB` has been transported in a `TEST` tablespace in the `orcl3` database.

```
SQL> CREATE TABLE hr.testtab ( id NUMBER, label VARCHAR2(10))
TABLESPACE test;

Table created.

SQL> INSERT INTO hr.testtab VALUES (10,'Skirt');

1 row created.
```

```
SQL> INSERT INTO hr.testtab VALUES (20,'Trousers');

1 row created.

SQL> COMMIT;

Commit complete.

SQL>
```

c. Find the list of user-defined tablespaces to be put in read-only mode.

```
SQL> SELECT tablespace_name FROM dba_tablespaces ORDER BY 1;

TABLESPACE_NAME
------------------------------
EXAMPLE
SYSAUX
SYSTEM
TEMP
TEST
UNDOTBS1
USERS

7 rows selected.

SQL>
```

d. The list may be different from yours according to the tablespaces created during the training session. Make all tablespaces except SYSTEM, SYSAUX, TEMP, and UNDOTBS1 read-only.

```
SQL> ALTER TABLESPACE example READ ONLY;

Tablespace altered.

SQL> ALTER TABLESPACE test READ ONLY;

Tablespace altered.

SQL> ALTER TABLESPACE users  READ ONLY;

Tablespace altered.

SQL>
```

e. Find the list of data files associated to the read-only tablespaces that need to be transported.

```
SQL> SELECT file_name FROM dba_data_files
  2  WHERE tablespace_name IN ('EXAMPLE',
'FDA_TBS','TEST','USERS');

FILE_NAME
---------------------------------------------------------------
/u01/app/oracle/oradata/orcl/users01.dbf
/u01/app/oracle/oradata/orcl/example01.dbf
/u01/app/oracle/oradata/orcl/test01.dbf

SQL> EXIT
$
```

3. Export the `orcl` database in full transportable mode.

```
$ rm /u01/app/oracle/admin/orcl/dpdump/expfull.dmp
$ expdp system/oracle_4U DUMPFILE=expfull.dmp FULL=Y
TRANSPORTABLE=ALWAYS LOGFILE=exp.log

Connected to: Oracle Database 12c Enterprise Edition Release
12.1.0.0.2 - 64bit Production
With the Partitioning, OLAP, Advanced Analytics, Real
Application Testing
and Unified Auditing options
Starting "SYSTEM"."SYS_EXPORT_FULL_01":  system/********
DUMPFILE=expfull.dmp FULL=Y TRANSPORTABLE=ALWAYS LOGFILE=exp.log
Estimate in progress using BLOCKS method...
Processing object type
DATABASE_EXPORT/PLUGTS_FULL/FULL/PLUGTS_TABLESPACE
Processing object type DATABASE_EXPORT/PLUGTS_FULL/PLUGTS_BLK
Processing object type
DATABASE_EXPORT/EARLY_OPTIONS/VIEWS_AS_TABLES/TABLE_DATA
Processing object type DATABASE_EXPORT/NORMAL_OPTIONS/TABLE_DATA
Processing object type
DATABASE_EXPORT/NORMAL_OPTIONS/VIEWS_AS_TABLES/TABLE_DATA
Processing object type DATABASE_EXPORT/SCHEMA/TABLE/TABLE_DATA
Total estimation using BLOCKS method: 2.703 MB
Processing object type
DATABASE_EXPORT/PRE_SYSTEM_IMPCALLOUT/MARKER
Processing object type
DATABASE_EXPORT/PRE_INSTANCE_IMPCALLOUT/MARKER
Processing object type DATABASE_EXPORT/TABLESPACE
Processing object type DATABASE_EXPORT/PROFILE
Processing object type DATABASE_EXPORT/SYS_USER/USER
Processing object type DATABASE_EXPORT/SCHEMA/USER
Processing object type DATABASE_EXPORT/ROLE
Processing object type DATABASE_EXPORT/RADM_FPTM
Processing object type
DATABASE_EXPORT/GRANT/SYSTEM_GRANT/PROC_SYSTEM_GRANT
Processing object type DATABASE_EXPORT/SCHEMA/GRANT/SYSTEM_GRANT
Processing object type DATABASE_EXPORT/SCHEMA/ROLE_GRANT
Processing object type DATABASE_EXPORT/SCHEMA/DEFAULT_ROLE
Processing object type DATABASE_EXPORT/SCHEMA/ON_USER_GRANT
Processing object type DATABASE_EXPORT/SCHEMA/TABLESPACE_QUOTA
Processing object type DATABASE_EXPORT/RESOURCE_COST
Processing object type DATABASE_EXPORT/TRUSTED_DB_LINK
Processing object type DATABASE_EXPORT/SCHEMA/SEQUENCE/SEQUENCE
Processing object type DATABASE_EXPORT/DIRECTORY/DIRECTORY
Processing object type
DATABASE_EXPORT/DIRECTORY/GRANT/OWNER_GRANT/OBJECT_GRANT
Processing object type DATABASE_EXPORT/SCHEMA/SYNONYM
Processing object type DATABASE_EXPORT/SCHEMA/TYPE/INC_TYPE
Processing object type DATABASE_EXPORT/SCHEMA/TYPE/TYPE_SPEC
Processing object type
DATABASE_EXPORT/SCHEMA/TYPE/GRANT/OWNER_GRANT/OBJECT_GRANT
Processing object type
DATABASE_EXPORT/SYSTEM_PROCOBJACT/PRE_SYSTEM_ACTIONS/PROCACT_SYS
TEM
Processing object type DATABASE_EXPORT/SYSTEM_PROCOBJACT/PROCOBJ
Processing object type
DATABASE_EXPORT/SYSTEM_PROCOBJACT/POST_SYSTEM_ACTIONS/PROCACT_SY
STEM
Processing object type DATABASE_EXPORT/SCHEMA/PROCACT_SCHEMA
Processing object type
DATABASE_EXPORT/EARLY_OPTIONS/VIEWS_AS_TABLES/TABLE
Processing object type
DATABASE_EXPORT/EARLY_POST_INSTANCE_IMPCALLOUT/MARKER
Processing object type
DATABASE_EXPORT/SCHEMA/XMLSCHEMA/XMLSCHEMA
Processing object type DATABASE_EXPORT/NORMAL_OPTIONS/TABLE
Processing object type
DATABASE_EXPORT/NORMAL_OPTIONS/VIEWS_AS_TABLES/TABLE
Processing object type DATABASE_EXPORT/NORMAL_OPTIONS/PROCEDURE
Processing object type
DATABASE_EXPORT/NORMAL_OPTIONS/OPTION_PACKAGE/PACKAGE_SPEC
Processing object type
DATABASE_EXPORT/NORMAL_OPTIONS/OPTION_PACKAGE/PACKAGE_BODY
Processing object type
DATABASE_EXPORT/NORMAL_POST_INSTANCE_IMPCALLOU/MARKER
```

```
Processing object type DATABASE_EXPORT/SCHEMA/TABLE/TABLE
Processing object type
DATABASE_EXPORT/SCHEMA/TABLE/GRANT/OWNER_GRANT/OBJECT_GRANT
Processing object type DATABASE_EXPORT/SCHEMA/TABLE/COMMENT
Processing object type
DATABASE_EXPORT/XS_SECURITY/SCHEMA/XS_SECURITY_CLASS
Processing object type DATABASE_EXPORT/SCHEMA/FUNCTION/FUNCTION
Processing object type
DATABASE_EXPORT/SCHEMA/PROCEDURE/PROCEDURE
Processing object type
DATABASE_EXPORT/SCHEMA/FUNCTION/ALTER_FUNCTION
Processing object type
DATABASE_EXPORT/SCHEMA/PROCEDURE/ALTER_PROCEDURE
Processing object type DATABASE_EXPORT/SCHEMA/TABLE/INDEX/INDEX
Processing object type
DATABASE_EXPORT/SCHEMA/TABLE/INDEX/FUNCTIONAL_INDEX/INDEX
Processing object type
DATABASE_EXPORT/SCHEMA/TABLE/CONSTRAINT/CONSTRAINT
Processing object type
DATABASE_EXPORT/SCHEMA/TABLE/INDEX/STATISTICS/INDEX_STATISTICS
Processing object type
DATABASE_EXPORT/SCHEMA/TABLE/INDEX/STATISTICS/FUNCTIONAL_INDEX/I
NDEX_STATISTICS
Processing object type DATABASE_EXPORT/SCHEMA/VIEW/VIEW
Processing object type
DATABASE_EXPORT/SCHEMA/VIEW/GRANT/OWNER_GRANT/OBJECT_GRANT
Processing object type DATABASE_EXPORT/SCHEMA/VIEW/COMMENT
Processing object type DATABASE_EXPORT/SCHEMA/TYPE/TYPE_BODY
Processing object type
DATABASE_EXPORT/SCHEMA/TABLE/CONSTRAINT/REF_CONSTRAINT
Processing object type
DATABASE_EXPORT/SCHEMA/TABLE/INDEX/BITMAP_INDEX/INDEX
Processing object type
DATABASE_EXPORT/SCHEMA/TABLE/INDEX/STATISTICS/BITMAP_INDEX/INDEX
_STATISTICS
Processing object type
DATABASE_EXPORT/SCHEMA/TABLE/STATISTICS/TABLE_STATISTICS
Processing object type DATABASE_EXPORT/STATISTICS/MARKER
Processing object type
DATABASE_EXPORT/SCHEMA/TABLE/INDEX/DOMAIN_INDEX/INDEX
Processing object type DATABASE_EXPORT/SCHEMA/TABLE/TRIGGER
Processing object type DATABASE_EXPORT/SCHEMA/VIEW/TRIGGER
Processing object type DATABASE_EXPORT/SCHEMA/MATERIALIZED_VIEW
Processing object type DATABASE_EXPORT/SCHEMA/DIMENSION
Processing object type DATABASE_EXPORT/END_PLUGTS_BLK
Processing object type
DATABASE_EXPORT/FINAL_POST_INSTANCE_IMPCALLOUT/MARKER
Processing object type
DATABASE_EXPORT/SCHEMA/TABLE/POST_INSTANCE/PROCACT_INSTANCE
Processing object type
DATABASE_EXPORT/SCHEMA/TABLE/POST_INSTANCE/PROCDEPOBJ
Processing object type
DATABASE_EXPORT/SCHEMA/POST_SCHEMA/PROCOBJ
Processing object type
DATABASE_EXPORT/SCHEMA/POST_SCHEMA/PROCACT_SCHEMA
Processing object type
DATABASE_EXPORT/AUDIT_UNIFIED/AUDIT_POLICY
Processing object type
DATABASE_EXPORT/AUDIT_UNIFIED/AUDIT_POLICY_ENABLE
Processing object type DATABASE_EXPORT/AUDIT
Processing object type
DATABASE_EXPORT/POST_SYSTEM_IMPCALLOUT/MARKER
. . exported "SYS"."KU$_USER_MAPPING_VIEW"                  6.218
KB      50 rows
. . exported "ORDDATA"."ORDDCM_DOCS"                        252.9
KB       9 rows
. . exported "SYS"."AUD$"                                   25.18
KB      24 rows
. . exported "LBACSYS"."OLS$AUDIT_ACTIONS"                  5.734
KB       8 rows
. . exported "LBACSYS"."OLS$DIP_EVENTS"                     5.515
KB       2 rows
. . exported "LBACSYS"."OLS$INSTALLATIONS"                  6.937
KB       2 rows
. . exported "LBACSYS"."OLS$PROPS"                          6.210
KB       5 rows
. . exported "SYS"."DAM_CONFIG_PARAM$"                      6.507
KB      14 rows
. . exported "SYS"."TSDP_PARAMETER$"                        5.929
KB       1 rows
. . exported "SYS"."TSDP_POLICY$"                           5.898
KB       1 rows
. . exported "SYS"."TSDP_SUBPOL$"                           6.304
KB       1 rows
. . exported "SYSTEM"."REDO_DB"                                23
KB       1 rows
. . exported "WMSYS"."WM$ENV_VARS$"                         6.054
KB       5 rows
. . exported "WMSYS"."WM$EVENTS_INFO$"                      5.789
KB      12 rows
. . exported "WMSYS"."WM$HINT_TABLE$"                       9.429
KB      75 rows
```

```
. . exported "WMSYS"."WM$NEXTVER_TABLE$"                      6.351
KB        1 rows
. . exported "WMSYS"."WM$VERSION_HIERARCHY_TABLE$"            5.960
KB        1 rows
. . exported "WMSYS"."WM$WORKSPACES_TABLE$"                   12.08
KB        1 rows
. . exported "WMSYS"."WM$WORKSPACE_PRIV_TABLE$"               6.539
KB        8 rows
. . exported "LBACSYS"."OLS$AUDIT"                                0
KB        0 rows
. . exported "LBACSYS"."OLS$COMPARTMENTS"                         0
KB        0 rows
. . exported "LBACSYS"."OLS$DIP_DEBUG"                            0
KB        0 rows
. . exported "LBACSYS"."OLS$GROUPS"                               0
KB        0 rows
. . exported "LBACSYS"."OLS$LAB"                                  0
KB        0 rows
. . exported "LBACSYS"."OLS$LEVELS"                               0
KB        0 rows
. . exported "LBACSYS"."OLS$POL"                                  0
KB        0 rows
. . exported "LBACSYS"."OLS$POLICY_ADMIN"                         0
KB        0 rows
. . exported "LBACSYS"."OLS$POLS"                                 0
KB        0 rows
. . exported "LBACSYS"."OLS$POLT"                                 0
KB        0 rows
. . exported "LBACSYS"."OLS$PROFILE"                              0
KB        0 rows
. . exported "LBACSYS"."OLS$PROFILES"                             0
KB        0 rows
. . exported "LBACSYS"."OLS$PROG"                                 0
KB        0 rows
. . exported "LBACSYS"."OLS$SESSINFO"                             0
KB        0 rows
. . exported "LBACSYS"."OLS$USER"                                 0
KB        0 rows
. . exported "LBACSYS"."OLS$USER_COMPARTMENTS"                    0
KB        0 rows
. . exported "LBACSYS"."OLS$USER_GROUPS"                          0
KB        0 rows
. . exported "LBACSYS"."OLS$USER_LEVELS"                          0
KB        0 rows
. . exported "SYS"."DAM_CLEANUP_EVENTS$"                          0
KB        0 rows
```

```
. . exported "SYS"."DAM_CLEANUP_JOBS$"                       0
KB       0 rows
. . exported "SYS"."TSDP_ASSOCIATION$"                       0
KB       0 rows
. . exported "SYS"."TSDP_CONDITION$"                         0
KB       0 rows
. . exported "SYS"."TSDP_FEATURE_POLICY$"                    0
KB       0 rows
. . exported "SYS"."TSDP_PROTECTION$"                        0
KB       0 rows
. . exported "SYS"."TSDP_SENSITIVE_DATA$"                    0
KB       0 rows
. . exported "SYS"."TSDP_SENSITIVE_TYPE$"                    0
KB       0 rows
. . exported "SYS"."TSDP_SOURCE$"                            0
KB       0 rows
. . exported "SYSTEM"."REDO_LOG"                             0
KB       0 rows
. . exported "WMSYS"."WM$BATCH_COMPRESSIBLE_TABLES$"         0
KB       0 rows
. . exported "WMSYS"."WM$CONSTRAINTS_TABLE$"                 0
KB       0 rows
. . exported "WMSYS"."WM$CONS_COLUMNS$"                      0
KB       0 rows
. . exported "WMSYS"."WM$LOCKROWS_INFO$"                     0
KB       0 rows
. . exported "WMSYS"."WM$MODIFIED_TABLES$"                   0
KB       0 rows
. . exported "WMSYS"."WM$MP_GRAPH_WORKSPACES_TABLE$"         0
KB       0 rows
. . exported "WMSYS"."WM$MP_PARENT_WORKSPACES_TABLE$"        0
KB       0 rows
. . exported "WMSYS"."WM$NESTED_COLUMNS_TABLE$"              0
KB       0 rows
. . exported "WMSYS"."WM$REMOVED_WORKSPACES_TABLE$"          0
KB       0 rows
. . exported "WMSYS"."WM$RESOLVE_WORKSPACES_TABLE$"          0
KB       0 rows
. . exported "WMSYS"."WM$RIC_LOCKING_TABLE$"                 0
KB       0 rows
. . exported "WMSYS"."WM$RIC_TABLE$"                         0
KB       0 rows
. . exported "WMSYS"."WM$RIC_TRIGGERS_TABLE$"                0
KB       0 rows
. . exported "WMSYS"."WM$UDTRIG_DISPATCH_PROCS$"             0
KB       0 rows
```

```
. . exported "WMSYS"."WM$UDTRIG_INFO$"                        0
KB       0 rows
. . exported "WMSYS"."WM$VERSION_TABLE$"                      0
KB       0 rows
. . exported "WMSYS"."WM$VT_ERRORS_TABLE$"                    0
KB       0 rows
. . exported "WMSYS"."WM$WORKSPACE_SAVEPOINTS_TABLE$"         0
KB       0 rows
. . exported "SYSTEM"."SCHEDULER_PROGRAM_ARGS"            9.484
KB      12 rows
. . exported "SYS"."AUDTAB$TBS$FOR_EXPORT"                5.929
KB       2 rows
. . exported "WMSYS"."WM$EXP_MAP"                         7.695
KB       3 rows
. . exported "SYS"."DBA_SENSITIVE_DATA"                       0
KB       0 rows
. . exported "SYS"."DBA_TSDP_POLICY_PROTECTION"               0
KB       0 rows
. . exported "SYS"."FGA_LOG$FOR_EXPORT"                       0
KB       0 rows
. . exported "SYS"."NACL$_HOST_EXP"                           0
KB       0 rows
. . exported "SYS"."NACL$_WALLET_EXP"                         0
KB       0 rows
. . exported "SYSTEM"."SCHEDULER_JOB_ARGS"                    0
KB       0 rows
Master table "SYSTEM"."SYS_EXPORT_FULL_01" successfully
loaded/unloaded
*****************************************************************
*
Dump file set for SYSTEM.SYS_EXPORT_FULL_01 is:
  /u01/app/oracle/admin/orcl/dpdump/expfull.dmp
*****************************************************************
*
Datafiles required for transportable tablespace EXAMPLE:
  /u01/app/oracle/oradata/orcl/example01.dbf
Datafiles required for transportable tablespace TEST:
  /u01/app/oracle/oradata/orcl/test01.dbf
Datafiles required for transportable tablespace USERS:
  /u01/app/oracle/oradata/orcl/users01.dbf
Job "SYSTEM"."SYS_EXPORT_FULL_01" successfully completed at Mon
Sep 10 05:40:45 2012 elapsed 0 00:09:40
$
```

4. View the log file `exp.log` to get the list of data files to be transported before the full transportable import.

   a. Find the `exp.log` file.

```
$ cd /u01/app/oracle/admin/orcl/dpdump
$ ls -ltr exp*
-rw-r--r-- 1 oracle oinstall    1624 Sep  9 23:52 export.log
-rw-r----- 1 oracle oinstall 6012928 Sep 10 05:40 expfull.dmp
-rw-r--r-- 1 oracle oinstall   12605 Sep 10 05:40 exp.log
$
```

b. View the last lines of the `exp.log` file.

```
$ tail -20 exp.log
. . exported "SYS"."DBA_SENSITIVE_DATA"                      0
KB       0 rows
. . exported "SYS"."DBA_TSDP_POLICY_PROTECTION"              0
KB       0 rows
. . exported "SYS"."FGA_LOG$FOR_EXPORT"                      0
KB       0 rows
. . exported "SYS"."NACL$_HOST_EXP"                          0
KB       0 rows
. . exported "SYS"."NACL$_WALLET_EXP"                        0
KB       0 rows
. . exported "SYSTEM"."SCHEDULER_JOB_ARGS"                   0
KB       0 rows
Master table "SYSTEM"."SYS_EXPORT_FULL_01" successfully
loaded/unloaded
****************************************************************
*
Dump file set for SYSTEM.SYS_EXPORT_FULL_01 is:
  /u01/app/oracle/admin/orcl/dpdump/expfull.dmp
****************************************************************
*
Datafiles required for transportable tablespace EXAMPLE:
  /u01/app/oracle/oradata/orcl/example01.dbf
Datafiles required for transportable tablespace TEST:
  /u01/app/oracle/oradata/orcl/test01.dbf
Datafiles required for transportable tablespace USERS:
  /u01/app/oracle/oradata/orcl/users01.dbf
Job "SYSTEM"."SYS_EXPORT_FULL_01" successfully completed at Mon
Sep 10 05:40:45 2012 elapsed 0 00:09:40
$
```

5. After the `orcl3` database is created, you can copy the data files to the target locations `/u01/app/oracle/oradata/orcl3` and the export dump file to `/u01/app/oracle/admin/orcl3/dpdump`. Before proceeding, check that there are not any tablespaces in the target `orcl3` database having the same names as the tablespaces in the source `orcl` database.

   a. Set your environment to the target database `orcl3`.

```
$ . oraenv
```

```
ORACLE_SID = [cdb2] ? orcl3
The Oracle base remains unchanged with value /u01/app/oracle

$ sqlplus / as sysdba

Connected to:
Oracle Database 12c Enterprise Edition Release 12.1.0.0.2 -
64bit Production
With the Partitioning, OLAP, Data Mining and Real Application
Testing options

SQL>
```

b. Select the tablespace names.

```
SQL> SELECT tablespace_name FROM dba_tablespaces;

TABLESPACE_NAME
------------------------------
SYSTEM
SYSAUX
UNDOTBS1
TEMP
USERS

SQL>
```

c. Rename the `USERS` tablespace to `USERS_NEW` and the data file `/u01/app/oracle/oradata/orcl3/users01.dbf` to `/u01/app/oracle/oradata/orcl3/users_new01.dbf`.

```
SQL> ALTER TABLESPACE users RENAME TO users_new;

Tablespace altered.

SQL> ALTER TABLESPACE users_new OFFLINE;

Tablespace altered.

SQL> EXIT
$
$ mv /u01/app/oracle/oradata/orcl3/users01.dbf
/u01/app/oracle/oradata/orcl3/users_new01.dbf
$
$ sqlplus / as sysdba

Connected to:
```

```
Oracle Database 12c Enterprise Edition Release 12.1.0.0.2 -
64bit Production
With the Partitioning, OLAP, Data Mining and Real Application
Testing options

SQL> ALTER DATABASE RENAME FILE
'/u01/app/oracle/oradata/orcl3/users01.dbf' TO
'/u01/app/oracle/oradata/orcl3/users_new01.dbf';

Database altered.

SQL> ALTER TABLESPACE users_new ONLINE;

Tablespace altered.

SQL> SELECT tablespace_name FROM dba_tablespaces ORDER BY 1;

TABLESPACE_NAME
------------------------------
SYSAUX
SYSTEM
TEMP
UNDOTBS1
USERS_NEW

SQL> EXIT
$
```

d. Now you can copy the data files to the target locations `/u01/app/oracle/oradata/orcl3` and the export dump file to `/u01/app/oracle/admin/orcl3/dpdump`.

```
$ cp /u01/app/oracle/oradata/orcl/test01.dbf
/u01/app/oracle/oradata/orcl/example01.dbf
/u01/app/oracle/oradata/orcl/users01.dbf
/u01/app/oracle/oradata/orcl3
$ cp /u01/app/oracle/admin/orcl/dpdump/expfull.dmp
/u01/app/oracle/admin/orcl3/dpdump/expfull.dmp
$
```

6. Import the `orcl` database into the `orcl3` database in full transportable mode. There are many errors due to existing objects in the target `orcl3` database. These errors can be ignored.

```
$ rm /u01/app/oracle/admin/orcl3/dpdump/import.log
$ impdp system/oracle_4U FULL=Y dumpfile=expfull.dmp
TRANSPORT_DATAFILES='/u01/app/oracle/oradata/orcl3/test01.dbf','
```

```
/u01/app/oracle/oradata/orcl3/users01.dbf','/u01/app/oracle/orad
ata/orcl3/example01.dbf' logfile=import.log

Connected to: Oracle Database 12c Enterprise Edition Release
12.1.0.0.2 - 64bit Production
With the Partitioning, OLAP, Data Mining and Real Application
Testing options
Master table "SYSTEM"."SYS_IMPORT_FULL_01" successfully
loaded/unloaded
Starting "SYSTEM"."SYS_IMPORT_FULL_01":  system/******** FULL=Y
dumpfile=expfull.dmp
TRANSPORT_DATAFILES=/u01/app/oracle/oradata/orcl3/test01.dbf,/u0
1/app/oracle/oradata/orcl3/users01.dbf,/u01/app/oracle/oradata/o
rcl3/example01.dbf logfile=import.log
Processing object type
DATABASE_EXPORT/PRE_SYSTEM_IMPCALLOUT/MARKER
Processing object type
DATABASE_EXPORT/PRE_INSTANCE_IMPCALLOUT/MARKER
Processing object type DATABASE_EXPORT/PLUGTS_FULL/PLUGTS_BLK
Processing object type DATABASE_EXPORT/TABLESPACE
ORA-31684: Object type TABLESPACE:"UNDOTBS1" already exists
ORA-31684: Object type TABLESPACE:"TEMP" already exists
Processing object type DATABASE_EXPORT/PROFILE
ORA-31684: Object type PROFILE:"MONITORING_PROFILE" already
exists
Processing object type DATABASE_EXPORT/SYS_USER/USER
Processing object type DATABASE_EXPORT/SCHEMA/USER
ORA-31684: Object type USER:"OUTLN" already exists
ORA-31684: Object type USER:"OLAPSYS" already exists
ORA-31684: Object type USER:"MDDATA" already exists
ORA-31684: Object type USER:"SPATIAL_WFS_ADMIN_USR" already
exists
ORA-31684: Object type USER:"SPATIAL_CSW_ADMIN_USR" already
exists
ORA-31684: Object type USER:"FLOWS_FILES" already exists
ORA-31684: Object type USER:"APEX_PUBLIC_USER" already exists
ORA-31684: Object type USER:"APEX_040100" already exists
ORA-31684: Object type USER:"SCOTT" already exists
Processing object type DATABASE_EXPORT/ROLE
ORA-31684: Object type ROLE:"SELECT_CATALOG_ROLE" already exists
ORA-31684: Object type ROLE:"EXECUTE_CATALOG_ROLE" already
exists
ORA-31684: Object type ROLE:"DELETE_CATALOG_ROLE" already exists
ORA-31684: Object type ROLE:"CAPTURE_ADMIN" already exists
ORA-31684: Object type ROLE:"CDB_DBA" already exists
```

```
ORA-31684: Object type ROLE:"DBFS_ROLE" already exists
ORA-31684: Object type ROLE:"AQ_ADMINISTRATOR_ROLE" already
exists
ORA-31684: Object type ROLE:"AQ_USER_ROLE" already exists
ORA-31684: Object type ROLE:"ADM_PARALLEL_EXECUTE_TASK" already
exists
ORA-31684: Object type ROLE:"PROVISIONER" already exists
ORA-31684: Object type ROLE:"XS_RESOURCE" already exists
ORA-31684: Object type ROLE:"XS_SESSION_ADMIN" already exists
ORA-31684: Object type ROLE:"XS_NSATTR_ADMIN" already exists
ORA-31684: Object type ROLE:"XS_CACHE_ADMIN" already exists
ORA-31684: Object type ROLE:"GSMUSER_ROLE" already exists
ORA-31684: Object type ROLE:"GATHER_SYSTEM_STATISTICS" already
exists
ORA-31684: Object type ROLE:"OPTIMIZER_PROCESSING_RATE" already
exists
ORA-31684: Object type ROLE:"RECOVERY_CATALOG_OWNER" already
exists
ORA-31684: Object type ROLE:"EM_EXPRESS_BASIC" already exists
ORA-31684: Object type ROLE:"EM_EXPRESS_ALL" already exists
ORA-31684: Object type ROLE:"SCHEDULER_ADMIN" already exists
ORA-31684: Object type ROLE:"HS_ADMIN_SELECT_ROLE" already
exists
ORA-31684: Object type ROLE:"HS_ADMIN_EXECUTE_ROLE" already
exists
ORA-31684: Object type ROLE:"HS_ADMIN_ROLE" already exists
ORA-31684: Object type ROLE:"GLOBAL_AQ_USER_ROLE" already exists
ORA-31684: Object type ROLE:"OEM_ADVISOR" already exists
ORA-31684: Object type ROLE:"OEM_MONITOR" already exists
ORA-31684: Object type ROLE:"XDBADMIN" already exists
ORA-31684: Object type ROLE:"XDB_SET_INVOKER" already exists
ORA-31684: Object type ROLE:"AUTHENTICATEDUSER" already exists
ORA-31684: Object type ROLE:"XDB_WEBSERVICES" already exists
ORA-31684: Object type ROLE:"XDB_WEBSERVICES_WITH_PUBLIC"
already exists
ORA-31684: Object type ROLE:"XDB_WEBSERVICES_OVER_HTTP" already
exists
ORA-31684: Object type ROLE:"WM_ADMIN_ROLE" already exists
ORA-31684: Object type ROLE:"JAVAUSERPRIV" already exists
ORA-31684: Object type ROLE:"JAVAIDPRIV" already exists
ORA-31684: Object type ROLE:"JAVASYSPRIV" already exists
ORA-31684: Object type ROLE:"JAVADEBUGPRIV" already exists
ORA-31684: Object type ROLE:"EJBCLIENT" already exists
ORA-31684: Object type ROLE:"JMXSERVER" already exists
```

```
ORA-31684: Object type ROLE:"JAVA_ADMIN" already exists
ORA-31684: Object type ROLE:"JAVA_DEPLOY" already exists
ORA-31684: Object type ROLE:"CTXAPP" already exists
ORA-31684: Object type ROLE:"ORDADMIN" already exists
ORA-31684: Object type ROLE:"OLAP_XS_ADMIN" already exists
ORA-31684: Object type ROLE:"OLAP_DBA" already exists
ORA-31684: Object type ROLE:"CWM_USER" already exists
ORA-31684: Object type ROLE:"OLAP_USER" already exists
ORA-31684: Object type ROLE:"SPATIAL_WFS_ADMIN" already exists
ORA-31684: Object type ROLE:"WFS_USR_ROLE" already exists
ORA-31684: Object type ROLE:"SPATIAL_CSW_ADMIN" already exists
ORA-31684: Object type ROLE:"CSW_USR_ROLE" already exists
ORA-31684: Object type ROLE:"LBAC_DBA" already exists
ORA-31684: Object type ROLE:"APEX_ADMINISTRATOR_ROLE" already
exists
ORA-31684: Object type ROLE:"APEX_GRANTS_FOR_NEW_USERS_ROLE"
already exists
ORA-31684: Object type ROLE:"DV_REALM_RESOURCE" already exists
ORA-31684: Object type ROLE:"DV_REALM_OWNER" already exists
ORA-31684: Object type ROLE:"DV_ACCTMGR" already exists
ORA-31684: Object type ROLE:"DV_OWNER" already exists
ORA-31684: Object type ROLE:"DV_ADMIN" already exists
ORA-31684: Object type ROLE:"DV_SECANALYST" already exists
ORA-31684: Object type ROLE:"DV_PUBLIC" already exists
ORA-31684: Object type ROLE:"DV_PATCH_ADMIN" already exists
ORA-31684: Object type ROLE:"DV_MONITOR" already exists
ORA-31684: Object type ROLE:"DV_STREAMS_ADMIN" already exists
ORA-31684: Object type ROLE:"DV_GOLDENGATE_ADMIN" already exists
ORA-31684: Object type ROLE:"DV_XSTREAM_ADMIN" already exists
ORA-31684: Object type ROLE:"DV_GOLDENGATE_REDO_ACCESS" already
exists
ORA-31684: Object type ROLE:"DV_AUDIT_CLEANUP" already exists
Processing object type DATABASE_EXPORT/RADM_FPTM
Processing object type
DATABASE_EXPORT/GRANT/SYSTEM_GRANT/PROC_SYSTEM_GRANT
Processing object type DATABASE_EXPORT/SCHEMA/GRANT/SYSTEM_GRANT
Processing object type DATABASE_EXPORT/SCHEMA/ROLE_GRANT
Processing object type DATABASE_EXPORT/SCHEMA/DEFAULT_ROLE
Processing object type DATABASE_EXPORT/SCHEMA/ON_USER_GRANT
Processing object type DATABASE_EXPORT/SCHEMA/TABLESPACE_QUOTA
Processing object type DATABASE_EXPORT/RESOURCE_COST
Processing object type DATABASE_EXPORT/TRUSTED_DB_LINK
Processing object type DATABASE_EXPORT/SCHEMA/SEQUENCE/SEQUENCE
Processing object type DATABASE_EXPORT/DIRECTORY/DIRECTORY
Processing object type
DATABASE_EXPORT/DIRECTORY/GRANT/OWNER_GRANT/OBJECT_GRANT
Processing object type DATABASE_EXPORT/SCHEMA/SYNONYM
Processing object type DATABASE_EXPORT/SCHEMA/TYPE/INC_TYPE
Processing object type DATABASE_EXPORT/SCHEMA/TYPE/TYPE_SPEC
Processing object type
DATABASE_EXPORT/SCHEMA/TYPE/GRANT/OWNER_GRANT/OBJECT_GRANT
Processing object type
DATABASE_EXPORT/SYSTEM_PROCOBJACT/PRE_SYSTEM_ACTIONS/PROCACT_SYS
TEM
Processing object type DATABASE_EXPORT/SYSTEM_PROCOBJACT/PROCOBJ
Processing object type
DATABASE_EXPORT/SYSTEM_PROCOBJACT/POST_SYSTEM_ACTIONS/PROCACT_SY
STEM
Processing object type DATABASE_EXPORT/SCHEMA/PROCACT_SCHEMA
Processing object type
DATABASE_EXPORT/EARLY_OPTIONS/VIEWS_AS_TABLES/TABLE
Processing object type
DATABASE_EXPORT/EARLY_OPTIONS/VIEWS_AS_TABLES/TABLE_DATA
. . imported "SYS"."KU$_EXPORT_USER_MAP"                     6.132
KB      43 rows
Processing object type
DATABASE_EXPORT/EARLY_POST_INSTANCE_IMPCALLOUT/MARKER
Processing object type
DATABASE_EXPORT/SCHEMA/XMLSCHEMA/XMLSCHEMA
Processing object type DATABASE_EXPORT/NORMAL_OPTIONS/TABLE
Processing object type DATABASE_EXPORT/NORMAL_OPTIONS/TABLE_DATA
. . imported "ORDDATA"."ORDDCM_DOCS_TRANSIENT"               252.9
KB       9 rows
. . imported "SYS"."AMGT$DP$DAM_CONFIG_PARAM$"               6.507
KB      14 rows
. . imported "SYS"."DP$TSDP_PARAMETER$"                      5.929
KB       1 rows
. . imported "SYS"."DP$TSDP_POLICY$"                         5.898
KB       1 rows
. . imported "SYS"."DP$TSDP_SUBPOL$"                         6.304
KB       1 rows
. . imported "SYSTEM"."REDO_DB_TMP"                          22.57
KB       1 rows
. . imported "WMSYS"."E$ENV_VARS$"                           6.054
KB       5 rows
. . imported "WMSYS"."E$EVENTS_INFO$"                        5.789
KB      12 rows
. . imported "WMSYS"."E$HINT_TABLE$"                         9.460
KB      76 rows
```

```
. . imported "WMSYS"."E$NEXTVER_TABLE$"                   6.351
KB       1 rows
. . imported "WMSYS"."E$VERSION_HIERARCHY_TABLE$"         5.960
KB       1 rows
. . imported "WMSYS"."E$WORKSPACES_TABLE$"                12.08
KB       1 rows
. . imported "WMSYS"."E$WORKSPACE_PRIV_TABLE$"            6.539
KB       8 rows
. . imported "LBACSYS"."OLS_DP$OLS$AUDIT"                     0
KB       0 rows
. . imported "LBACSYS"."OLS_DP$OLS$COMPARTMENTS"              0
KB       0 rows
. . imported "LBACSYS"."OLS_DP$OLS$GROUPS"                    0
KB       0 rows
. . imported "LBACSYS"."OLS_DP$OLS$LAB"                       0
KB       0 rows
. . imported "LBACSYS"."OLS_DP$OLS$LEVELS"                    0
KB       0 rows
. . imported "LBACSYS"."OLS_DP$OLS$POL"                       0
KB       0 rows
. . imported "LBACSYS"."OLS_DP$OLS$POLS"                      0
KB       0 rows
. . imported "LBACSYS"."OLS_DP$OLS$POLT"                      0
KB       0 rows
. . imported "LBACSYS"."OLS_DP$OLS$PROFILE"                   0
KB       0 rows
. . imported "LBACSYS"."OLS_DP$OLS$PROG"                      0
KB       0 rows
. . imported "LBACSYS"."OLS_DP$OLS$USER"                      0
KB       0 rows
. . imported "SYS"."AMGT$DP$AUD$"                             0
KB       0 rows
. . imported "SYS"."AMGT$DP$DAM_CLEANUP_EVENTS$"              0
KB       0 rows
. . imported "SYS"."AMGT$DP$DAM_CLEANUP_JOBS$"                0
KB       0 rows
. . imported "SYS"."AMGT$DP$FGA_LOG$"                         0
KB       0 rows
. . imported "SYS"."DP$TSDP_ASSOCIATION$"                     0
KB       0 rows
. . imported "SYS"."DP$TSDP_CONDITION$"                       0
KB       0 rows
. . imported "SYS"."DP$TSDP_FEATURE_POLICY$"                  0
KB       0 rows
. . imported "SYS"."DP$TSDP_PROTECTION$"                      0
KB       0 rows
```

```
. . imported "SYS"."DP$TSDP_SENSITIVE_DATA$"                        0
KB       0 rows
. . imported "SYS"."DP$TSDP_SENSITIVE_TYPE$"                        0
KB       0 rows
. . imported "SYS"."DP$TSDP_SOURCE$"                                0
KB       0 rows
. . imported "SYSTEM"."REDO_LOG_TMP"                                0
KB       0 rows
. . imported "WMSYS"."E$BATCH_COMPRESSIBLE_TABLES$"                 0
KB       0 rows
. . imported "WMSYS"."E$CONSTRAINTS_TABLE$"                         0
KB       0 rows
. . imported "WMSYS"."E$CONS_COLUMNS$"                              0
KB       0 rows
. . imported "WMSYS"."E$LOCKROWS_INFO$"                             0
KB       0 rows
. . imported "WMSYS"."E$MODIFIED_TABLES$"                           0
KB       0 rows
. . imported "WMSYS"."E$MP_GRAPH_WORKSPACES_TABLE$"                 0
KB       0 rows
. . imported "WMSYS"."E$MP_PARENT_WORKSPACES_TABLE$"                0
KB       0 rows
. . imported "WMSYS"."E$NESTED_COLUMNS_TABLE$"                      0
KB       0 rows
. . imported "WMSYS"."E$REMOVED_WORKSPACES_TABLE$"                  0
KB       0 rows
. . imported "WMSYS"."E$RESOLVE_WORKSPACES_TABLE$"                  0
KB       0 rows
. . imported "WMSYS"."E$RIC_LOCKING_TABLE$"                         0
KB       0 rows
. . imported "WMSYS"."E$RIC_TABLE$"                                 0
KB       0 rows
. . imported "WMSYS"."E$RIC_TRIGGERS_TABLE$"                        0
KB       0 rows
. . imported "WMSYS"."E$UDTRIG_DISPATCH_PROCS$"                     0
KB       0 rows
. . imported "WMSYS"."E$UDTRIG_INFO$"                               0
KB       0 rows
. . imported "WMSYS"."E$VERSION_TABLE$"                             0
KB       0 rows
. . imported "WMSYS"."E$VT_ERRORS_TABLE$"                           0
KB       0 rows
. . imported "WMSYS"."E$WORKSPACE_SAVEPOINTS_TABLE$"                0
KB       0 rows
Processing object type
DATABASE_EXPORT/NORMAL_OPTIONS/VIEWS_AS_TABLES/TABLE
```

```
Processing object type
DATABASE_EXPORT/NORMAL_OPTIONS/VIEWS_AS_TABLES/TABLE_DATA
. . imported "SYSTEM"."SCHEDULER_PROGRAM_ARGS_TMP"       9.484
KB      12 rows
. . imported "WMSYS"."E$EXP_MAP"                         7.695
KB       3 rows
. . imported "SYS"."DP$DBA_SENSITIVE_DATA"                   0
KB       0 rows
. . imported "SYS"."DP$DBA_TSDP_POLICY_PROTECTION"           0
KB       0 rows
. . imported "SYS"."NACL$_HOST_IMP"                          0
KB       0 rows
. . imported "SYS"."NACL$_NAME_MAP_IMP"                      0
KB       0 rows
. . imported "SYS"."NACL$_WALLET_IMP"                        0
KB       0 rows
. . imported "SYSTEM"."SCHEDULER_JOB_ARGS_TMP"               0
KB       0 rows
Processing object type DATABASE_EXPORT/NORMAL_OPTIONS/PROCEDURE
ORA-39342: Internal error - failed to import internal objects
tagged with APEX due to ORA-00955: name is already used by an
existing object.
ORA-31684: Object type PROCEDURE:"SYS"."VALIDATE_APEX" already
exists
Processing object type
DATABASE_EXPORT/NORMAL_OPTIONS/OPTION_PACKAGE/PACKAGE_BODY
ORA-31684: Object type PACKAGE_BODY:"SYS"."WWV_DBMS_SQL" already
exists
Processing object type
DATABASE_EXPORT/NORMAL_POST_INSTANCE_IMPCALLOU/MARKER
Processing object type DATABASE_EXPORT/SCHEMA/TABLE/TABLE
ORA-39151: Table "SCOTT"."DEPT" exists. All dependent metadata
and data will be skipped due to table_exists_action of skip
ORA-39151: Table "SCOTT"."EMP" exists. All dependent metadata
and data will be skipped due to table_exists_action of skip
ORA-39151: Table "SCOTT"."BONUS" exists. All dependent metadata
and data will be skipped due to table_exists_action of skip
ORA-39151: Table "SCOTT"."SALGRADE" exists. All dependent
metadata and data will be skipped due to table_exists_action of
skip
ORA-39083: Object type TABLE:"OE"."PURCHASEORDER" failed to
create with error:
ORA-31061: XDB error: DBMS_XDBZ.ENABLE_HIERARCHY
ORA-06512: at "XDB.DBMS_XDBZ0", line 131
ORA-06512: at "XDB.DBMS_XDBZ0", line 598
ORA-01031: insufficient privileges
```

```
Failing sql is:
BEGIN DBMS_XDBZ.ENABLE_HIERARCHY('"OE"','"PURCHASEORDER"'); END;
Processing object type
DATABASE_EXPORT/SCHEMA/TABLE/GRANT/OWNER_GRANT/OBJECT_GRANT
Processing object type DATABASE_EXPORT/SCHEMA/TABLE/COMMENT
Processing object type
DATABASE_EXPORT/XS_SECURITY/SCHEMA/XS_SECURITY_CLASS
ORA-39083: Object type XS_SECURITY_CLASS:"SYS" failed to create
with error:
ORA-06550: line 11, column 27:
PLS-00103: Encountered the symbol ";" when expecting one of the
following:

   . ( ) , * % & = - + < / > at in is mod remainder not rem =>
   <an exponent (**)> <> or != or ~= >= <= <> and or like like2
   like4 likec as between from using || multiset member
   submultiset
The symbol ")" was substituted for ";" to continue.
Failing sql is:

DECLARE
 priv_list        XS$PRIVILEGE_LIST;
BEGIN

 priv_lis
Processing object type DATABASE_EXPORT/SCHEMA/FUNCTION/FUNCTION
Processing object type
DATABASE_EXPORT/SCHEMA/PROCEDURE/PROCEDURE
Processing object type
DATABASE_EXPORT/SCHEMA/FUNCTION/ALTER_FUNCTION
Processing object type
DATABASE_EXPORT/SCHEMA/PROCEDURE/ALTER_PROCEDURE
Processing object type DATABASE_EXPORT/SCHEMA/TABLE/INDEX/INDEX
Processing object type
DATABASE_EXPORT/SCHEMA/TABLE/INDEX/FUNCTIONAL_INDEX/INDEX
Processing object type
DATABASE_EXPORT/SCHEMA/TABLE/CONSTRAINT/CONSTRAINT
Processing object type
DATABASE_EXPORT/SCHEMA/TABLE/INDEX/STATISTICS/INDEX_STATISTICS
Processing object type
DATABASE_EXPORT/SCHEMA/TABLE/INDEX/STATISTICS/FUNCTIONAL_INDEX/I
NDEX_STATISTICS
Processing object type DATABASE_EXPORT/SCHEMA/VIEW/VIEW
Processing object type
DATABASE_EXPORT/SCHEMA/VIEW/GRANT/OWNER_GRANT/OBJECT_GRANT
```

```
Processing object type DATABASE_EXPORT/SCHEMA/TYPE/TYPE_BODY
Processing object type
DATABASE_EXPORT/SCHEMA/TABLE/CONSTRAINT/REF_CONSTRAINT
Processing object type
DATABASE_EXPORT/SCHEMA/TABLE/INDEX/BITMAP_INDEX/INDEX
Processing object type
DATABASE_EXPORT/SCHEMA/TABLE/INDEX/STATISTICS/BITMAP_INDEX/INDEX
_STATISTICS
Processing object type
DATABASE_EXPORT/SCHEMA/TABLE/STATISTICS/TABLE_STATISTICS
Processing object type DATABASE_EXPORT/STATISTICS/MARKER
OE.ACTION_TABLE_MEMBERS : sqlerrm = ORA-20000: Unable to set
values for index ACTION_TABLE_MEMBERS: does not exist or
insufficient privileges
OE.LINEITEM_TABLE_MEMBERS : sqlerrm = ORA-20000: Unable to set
values for index LINEITEM_TABLE_MEMBERS: does not exist or
insufficient privileges
Importing statistics failed for 2 object(s);
Processing object type
DATABASE_EXPORT/SCHEMA/TABLE/INDEX/DOMAIN_INDEX/INDEX
Processing object type DATABASE_EXPORT/SCHEMA/TABLE/TRIGGER
Processing object type DATABASE_EXPORT/SCHEMA/VIEW/TRIGGER
Processing object type DATABASE_EXPORT/SCHEMA/MATERIALIZED_VIEW
Processing object type DATABASE_EXPORT/SCHEMA/DIMENSION
Processing object type DATABASE_EXPORT/END_PLUGTS_BLK
Processing object type
DATABASE_EXPORT/FINAL_POST_INSTANCE_IMPCALLOUT/MARKER
Processing object type
DATABASE_EXPORT/SCHEMA/TABLE/POST_INSTANCE/PROCACT_INSTANCE
Processing object type
DATABASE_EXPORT/SCHEMA/TABLE/POST_INSTANCE/PROCDEPOBJ
Processing object type
DATABASE_EXPORT/SCHEMA/POST_SCHEMA/PROCOBJ
ORA-39083: Object type
PROCOBJ:"APEX_040100"."ORACLE_APEX_PURGE_SESSIONS" failed to
create with error:
ORA-27477: "APEX_040100"."ORACLE_APEX_PURGE_SESSIONS" already
exists
Failing sql is:
BEGIN
dbms_scheduler.create_job('"ORACLE_APEX_PURGE_SESSIONS"',
job_type=>'STORED_PROCEDURE', job_action=>
'WWV_FLOW_CACHE.PURGE_SESSIONS'
, number_of_arguments=>0,
start_date=>TO_TIMESTAMP_TZ('04-JUL-2012 11.31.16.846009000 PM -
07:00','DD-MON-RRRR HH.MI.SSXFF AM
TZR','NLS_DATE_LANGUAGE=english'), repeat_interval=>
'FREQ
ORA-39083: Object type
PROCOBJ:"APEX_040100"."ORACLE_APEX_MAIL_QUEUE" failed to create
with error:
ORA-27477: "APEX_040100"."ORACLE_APEX_MAIL_QUEUE" already exists
Failing sql is:
BEGIN
dbms_scheduler.create_job('"ORACLE_APEX_MAIL_QUEUE"',
job_type=>'STORED_PROCEDURE', job_action=>
'WWV_FLOW_MAIL.PUSH_QUEUE_IMMEDIATE'
, number_of_arguments=>0,
start_date=>TO_TIMESTAMP_TZ('04-JUL-2012 11.31.17.024586000 PM -
07:00','DD-MON-RRRR HH.MI.SSXFF AM
TZR','NLS_DATE_LANGUAGE=english'), repeat_interval=>
'FREQ=MINUTE
ORA-39083: Object type
PROCOBJ:"APEX_040100"."ORACLE_APEX_WS_NOTIFICATIONS" failed to
create with error:
ORA-27477: "APEX_040100"."ORACLE_APEX_WS_NOTIFICATIONS" already
exists
Failing sql is:
BEGIN
dbms_scheduler.create_job('"ORACLE_APEX_WS_NOTIFICATIONS"',
job_type=>'STORED_PROCEDURE', job_action=>
'WWV_FLOW_WORKSHEET_API.DO_NOTIFY'
, number_of_arguments=>0,
start_date=>TO_TIMESTAMP_TZ('04-JUL-2012 11.31.17.027832000 PM -
07:00','DD-MON-RRRR HH.MI.SSXFF AM
TZR','NLS_DATE_LANGUAGE=english'), repeat_interval
ORA-39083: Object type
PROCOBJ:"APEX_040100"."ORACLE_APEX_DAILY_MAINTENANCE" failed to
create with error:
ORA-27477: "APEX_040100"."ORACLE_APEX_DAILY_MAINTENANCE" already
exists
Failing sql is:
BEGIN
dbms_scheduler.create_job('"ORACLE_APEX_DAILY_MAINTENANCE"',
job_type=>'STORED_PROCEDURE', job_action=>
'WWV_FLOW_MAINT.DAILY_MAINTENANCE'
, number_of_arguments=>0,
start_date=>TO_TIMESTAMP_TZ('04-JUL-2012 11.31.17.031100000 PM -
07:00','DD-MON-RRRR HH.MI.SSXFF AM
TZR','NLS_DATE_LANGUAGE=english'), repeat_inter
Processing object type
DATABASE_EXPORT/SCHEMA/POST_SCHEMA/PROCACT_SCHEMA
Processing object type
DATABASE_EXPORT/AUDIT_UNIFIED/AUDIT_POLICY_ENABLE
Processing object type DATABASE_EXPORT/AUDIT
Processing object type
DATABASE_EXPORT/POST_SYSTEM_IMPCALLOUT/MARKER
Job "SYSTEM"."SYS_IMPORT_FULL_01" completed with 97 error(s) at
Tue Aug 21 09:59:05 2012 elapsed 0 00:04:44


$
```

7. Check in the target `orcl3` database that the tablespaces `TEST`, `EXAMPLE`, and `USERS` have been plugged and that the `HR.TESTTAB` table contains two rows as in the source `orcl` database.

```
$ sqlplus / as sysdba

Connected to:
Oracle Database 12c Enterprise Edition Release 12.1.0.0.2 -
64bit Production
With the Partitioning, OLAP, Data Mining and Real Application
Testing options

SQL> SELECT tablespace_name FROM dba_tablespaces;

TABLESPACE_NAME
------------------------------
SYSTEM
SYSAUX
UNDOTBS1
TEMP
USERS_NEW
EXAMPLE
TEST
USERS

8 rows selected.

SQL> SELECT name FROM v$datafile;

NAME
----------------------------------------------------------------
```

```
/u01/app/oracle/oradata/orcl3/system01.dbf
/u01/app/oracle/oradata/orcl3/example01.dbf
/u01/app/oracle/oradata/orcl3/sysaux01.dbf
/u01/app/oracle/oradata/orcl3/undotbs01.dbf
/u01/app/oracle/oradata/orcl3/test01.dbf
/u01/app/oracle/oradata/orcl3/users_new01.dbf
/u01/app/oracle/oradata/orcl3/users01.dbf

7 rows selected.

SQL> SELECT * FROM hr.testtab;

        ID LABEL
---------- ----------
        10 Skirt
        20 Trousers

SQL> EXIT
$
```

8. Put the user-defined tablespaces in the source database `orcl` back in read-write mode in order to let users work.

```
$ . oraenv
ORACLE_SID = [cdb2] ? orcl
The Oracle base remains unchanged with value /u01/app/oracle

$ sqlplus / as sysdba

Connected to:
Oracle Database 12c Enterprise Edition Release 12.1.0.0.2 -
64bit Production
With the Partitioning, OLAP, Data Mining and Real Application
Testing options

SQL> ALTER TABLESPACE example READ WRITE;

Tablespace altered.

SQL> ALTER TABLESPACE test READ WRITE;

Tablespace altered.

SQL> ALTER TABLESPACE users  READ WRITE;
```

```
Tablespace altered.

SQL> EXIT
$
```

Practice 20-3: Loading Data Using SQL*Loader Express Mode *(Optional)*

Overview

In this practice, you will load records from an `emp.dat` file into the `HR.TAB1` table using SQL*Loader in Express Mode.

Tasks

1. Make sure you are in the `~/labs/Load` directory and your environment points to the `orcl` instance.

```
$ cd ~/labs/Load
$ . oraenv
ORACLE_SID = [orcl] ? orcl
The Oracle base remains unchanged with value /u01/app/oracle
$
```

2. Create an `HR.EMP` table as follows.

```
$ sqlplus / as sysdba

Connected to:
Oracle Database 12c Enterprise Edition Release 12.1.0.0.2 -
64bit Production
With the Partitioning, OLAP, Data Mining and Real Application
Testing options

SQL> DROP TABLE hr.tab1 PURGE;
DROP TABLE hr.tab1 PURGE
              *
ERROR at line 1:
ORA-00942: table or view does not exist

SQL> CREATE TABLE hr.tab1 (id NUMBER, prod_name VARCHAR2(10));

Table created.

SQL> INSERT INTO hr.tab1 VALUES (10, 'Skirt');

1 row created.

SQL> INSERT INTO hr.tab1 VALUES (20, 'Trousers');

1 row created.

SQL> COMMIT;
```

```
Commit complete.
SQL> EXIT
$
```

3. Display the `$HOME/labs/Load/tab1.dat` file. It contains 5 records.

```
$ more tab1.dat
30, Shirt
40, Socks
50, Cap
60, Gloves
70, Tie
$
```

4. Load the five records into the `HR.TAB1` table using SQL*Loader in Express Mode.

```
$ sqlldr system/oracle_4U TABLE=hr.tab1

SQL*Loader: Release 12.1.0.0.2 - Production on Wed Aug 22
08:31:01 2012

Copyright (c) 1982, 2012, Oracle and/or its affiliates.  All
rights reserved.

Express Mode Load, Table: HR.TAB1
Path used:      External Table, DEGREE_OF_PARALLELISM=AUTO

Table HR.TAB1:
  5 Rows successfully loaded.

Check the log files:
  hr.log
  hr_%p.log_xt
for more information about the load.
$
```

5. Verify the existence of the log files.

```
$ ls -l hr*
-rw-r--r-- 1 oracle oinstall 1036 Aug 22 08:31 hr_782.log_xt
-rw-r--r-- 1 oracle oinstall 2241 Aug 22 08:31 hr.log
$
```

6. View the `hr.log` file.

```
$ more hr.log

SQL*Loader: Release 12.1.0.0.2 - Production on Wed Aug 22
08:31:01 2012

```

```
Copyright (c) 1982, 2012, Oracle and/or its affiliates.  All
rights reserved.

Express Mode Load, Table: HR.TAB1
Data File:      tab1.dat
  Bad File:     tab1.bad
  Discard File:  none specified

 (Allow all discards)

Number to load: ALL
Number to skip: 0
Errors allowed: 50
Continuation:    none specified
Path used:      External Table

Table HR.TAB1, loaded from every logical record.
Insert option in effect for this table: APPEND

   Column Name                  Position   Len  Term Encl
Datatype
------------------------------ ---------- ----- ---- ---- ------
---------------
ID                                  FIRST     *   ,
CHARACTER
PROD_NAME                            NEXT     *   ,
CHARACTER

Generated control file for possible reuse:
OPTIONS(EXTERNAL_TABLE=EXECUTE, TRIM=LRTRIM)
LOAD DATA
INFILE '(null)'
APPEND
INTO TABLE HR.TAB1
FIELDS TERMINATED BY ","
(
  ID,
  PROD_NAME
)
End of generated control file for possible reuse.

created temporary directory object SYS_SQLLDR_XT_TMPDIR_00000
for path /home/oracle/labs/Load
```

```
enable parallel DML: ALTER SESSION ENABLE PARALLEL DML

creating external table "SYS_SQLLDR_X_EXT_TAB1"

CREATE TABLE "SYS_SQLLDR_X_EXT_TAB1"
(
  "ID" NUMBER,
  "PROD_NAME" VARCHAR2(10)
)
ORGANIZATION external
(
  TYPE oracle_loader
  DEFAULT DIRECTORY SYS_SQLLDR_XT_TMPDIR_00000
  ACCESS PARAMETERS
  (
    RECORDS DELIMITED BY NEWLINE CHARACTERSET US7ASCII
    BADFILE 'SYS_SQLLDR_XT_TMPDIR_00000':'tab1.bad'
    LOGFILE 'hr_%p.log_xt'
    READSIZE 1048576
    FIELDS TERMINATED BY "," LRTRIM
    REJECT ROWS WITH ALL NULL FIELDS
    (
      "ID" CHAR(255),
      "PROD_NAME" CHAR(255)
    )
  )
  location
  (
    'tab1.dat'
  )
)REJECT LIMIT UNLIMITED

executing INSERT statement to load database table HR.TAB1

INSERT /*+ append parallel(auto) */ INTO HR.TAB1
(
  ID,
  PROD_NAME
)
SELECT
  "ID",
  "PROD_NAME"
FROM "SYS_SQLLDR_X_EXT_TAB1"

dropping external table "SYS_SQLLDR_X_EXT_TAB1"

Table HR.TAB1:
  5 Rows successfully loaded.

Run began on Wed Aug 22 08:31:01 2012
Run ended on Wed Aug 22 08:31:02 2012

Elapsed time was:     00:00:00.58
CPU time was:         00:00:00.01
$
```

7. View the `hr_782.log_xt` file.

```
$ more hr_782.log_xt

 LOG file opened at 08/22/12 08:31:01

Field Definitions for table SYS_SQLLDR_X_EXT_TAB1
  Record format DELIMITED BY NEWLINE
  Data in file has same endianness as the platform
  Reject rows with all null fields

  Fields in Data Source:

    ID                              CHAR (255)
      Terminated by ","
      Trim whitespace from left and right
    PROD_NAME                       CHAR (255)
      Terminated by ","
      Trim whitespace from left and right



 LOG file opened at 08/22/12 08:31:01

KUP-05004:   Warning: Intra source concurrency disabled because
parallel select was not request
ed.

Field Definitions for table SYS_SQLLDR_X_EXT_TAB1
  Record format DELIMITED BY NEWLINE
  Data in file has same endianness as the platform
  Reject rows with all null fields

  Fields in Data Source:

    ID                              CHAR (255)
      Terminated by ","
      Trim whitespace from left and right
    PROD_NAME                       CHAR (255)
      Terminated by ","
      Trim whitespace from left and right
$
```

8. Verify that the records have been inserted into the `HR.TAB1` table.

```
$ sqlplus / as sysdba

Connected to:
Oracle Database 12c Enterprise Edition Release 12.1.0.0.2 -
64bit Production
With the Partitioning, OLAP, Advanced Analytics and Real
Application Testing options

SQL> SELECT * FROM hr.tab1;

        ID PROD_NAME
---------- ----------
        10 Skirt
        20 Trousers
        30  Shirt
        40  Socks
        50  Cap
        60  Gloves
        70  Tie

SQL>
```

9. Drop the `HR.TAB1` table.

```
SQL> DROP TABLE hr.tab1 PURGE;

Table dropped.

SQL> EXIT
$
```

Practices for Lesson 21: Partitioning Enhancements

Chapter 21

Practices for Lesson 21

Practices Overview

In this practice, you will familiarize yourself with using partial local and global indexes.

Practice 21-1: Local and Global Partial Indexing on Partitioned Tables

Overview

In this practice, you will create a partitioned table with five partitions: A local partitioned index indexing only two partitions of the table and therefore composed of three index partitions, and a global index indexing the rows of only two partitions of the table.

Tasks

1. Create the partitioned table HR.TAB_PART1 with five partitions and only three local index partitions.
   a. Connect to the source database orcl.

```
$ . oraenv
ORACLE_SID = [cdb2] ? orcl
The Oracle base remains unchanged with value /u01/app/oracle

$ sqlplus / as sysdba

Connected to:
Oracle Database 12c Enterprise Edition Release 12.1.0.0.2 -
64bit Production
With the Partitioning, OLAP, Data Mining and Real Application
Testing options

SQL>
```

   b. To be sure that the segments, tables, partitions and indexes are created without inserting rows, set the deferred_segment_creation parameter to FALSE.

```
SQL> ALTER SYSTEM SET deferred_segment_creation=FALSE;

System altered.

SQL>
```

   c. Execute the CREATE TABLE statement as follows.

```
CREATE TABLE hr.tab_part1
 (
 order_id NUMBER(12),
 order_date DATE CONSTRAINT order_date_nn NOT NULL,
 order_mode VARCHAR2(8),
 customer_id NUMBER(6) CONSTRAINT order_cust_id_nn NOT NULL,
 order_status NUMBER(2),
 order_total NUMBER(8,2),
 sales_rep_id NUMBER(6),
 promotion_id NUMBER(6),
 CONSTRAINT order_mode_lov CHECK (order_mode in
('direct','online')),
 CONSTRAINT order_total_min CHECK (order_total >= 0)
)
INDEXING OFF
 PARTITION BY RANGE (ORDER_DATE)
 (PARTITION ord_p1 VALUES LESS THAN (TO_DATE('01-MAR-1999','DD-
MON-YYYY')) INDEXING ON,
  PARTITION ord_p2 VALUES LESS THAN (TO_DATE('01-JUL-1999','DD-
MON-YYYY')) INDEXING OFF,
  PARTITION ord_p3 VALUES LESS THAN (TO_DATE('01-OCT-1999','DD-
MON-YYYY')) INDEXING ON,
  PARTITION ord_p4 VALUES LESS THAN (TO_DATE('01-MAR-2000','DD-
MON-YYYY')),
  PARTITION ord_p5 VALUES LESS THAN (TO_DATE('01-MAR-2010','DD-
MON-YYYY')))
/
```

```
SQL> DROP TABLE hr.tab_part1 PURGE;

DROP TABLE hr.tab_part1 PURGE
              *
ERROR at line 1:
ORA-00942: table or view does not exist

SQL> CREATE TABLE hr.tab_part1
 (
 order_id NUMBER(12),
 order_date DATE CONSTRAINT order_date_nn NOT NULL,
 order_mode VARCHAR2(8),
 customer_id NUMBER(6) CONSTRAINT order_cust_id_nn NOT NULL,
 order_status NUMBER(2),
 order_total NUMBER(8,2),
 sales_rep_id NUMBER(6),
 promotion_id NUMBER(6),
 CONSTRAINT order_mode_lov CHECK (order_mode in
('direct','online')),
 CONSTRAINT order_total_min CHECK (order_total >= 0)
)
INDEXING OFF
 PARTITION BY RANGE (ORDER_DATE)
 (PARTITION ord_p1 VALUES LESS THAN (TO_DATE('01-MAR-1999','DD-
MON-YYYY')) INDEXING ON,
  PARTITION ord_p2 VALUES LESS THAN (TO_DATE('01-JUL-1999','DD-
MON-YYYY')) INDEXING OFF,
  PARTITION ord_p3 VALUES LESS THAN (TO_DATE('01-OCT-1999','DD-
MON-YYYY')) INDEXING ON,
  PARTITION ord_p4 VALUES LESS THAN (TO_DATE('01-MAR-2000','DD-
MON-YYYY')),
  PARTITION ord_p5 VALUES LESS THAN (TO_DATE('01-MAR-2010','DD-
MON-YYYY')))
/
  2    3    4    5    6    7    8    9   10   11   12   13   14
15   16   17   18   19   20   21

Table created.

SQL>
```

2. Check the default indexing value of the table created.

```
SQL> SELECT def_indexing FROM dba_part_tables
  2  WHERE table_name='TAB_PART1';


DEF
---
OFF


SQL>
```

3. Create a partial global index as follows.

```
SQL> CREATE INDEX hr.tab_part1_gidx_ordermode
             ON hr.tab_part1 (order_mode)
             GLOBAL INDEXING PARTIAL;
  2    3
Index created.

SQL>
```

4. Create a partial local partitioned index as follows.

```
SQL> CREATE INDEX hr.tab_part1_lidx_orderdate
             ON hr.tab_part1 (order_date)
             LOCAL INDEXING PARTIAL;
  2    3
Index created.

SQL>
```

5. Check the indexing type and status of the indexes.

```
SQL> col INDEX_NAME format a26
SQL> SELECT indexing, index_name, status
     FROM   dba_indexes
     WHERE  index_name like 'TAB_PART1%';
  2   3
```

```
INDEXING INDEX_NAME                 STATUS
-------- -------------------------- --------
PARTIAL  TAB_PART1_LIDX_ORDERDATE   N/A
PARTIAL  TAB_PART1_GIDX_ORDERMODE   VALID


SQL>
```

6. Check the status of the index partitions of the partial local index.

```
SQL> col PARTITION_NAME format a20
SQL> SELECT index_name, partition_name, status
     FROM dba_ind_partitions
     WHERE INDEX_NAME like 'TAB_PART1%';
  2   3
INDEX_NAME                 PARTITION_NAME       STATUS
-------------------------- -------------------- --------
TAB_PART1_LIDX_ORDERDATE   ORD_P5               UNUSABLE
TAB_PART1_LIDX_ORDERDATE   ORD_P4               UNUSABLE
TAB_PART1_LIDX_ORDERDATE   ORD_P3               USABLE
TAB_PART1_LIDX_ORDERDATE   ORD_P2               UNUSABLE
TAB_PART1_LIDX_ORDERDATE  ORD_P1                USABLE

SQL>
```

7. Insert rows into the table.

```
SQL> INSERT INTO hr.tab_part1
VALUES (10, TO_DATE('01-FEB-1999','DD-MON-YYYY'),
'direct',200,0,20,201,222);
1 row created.

SQL> INSERT INTO hr.tab_part1
VALUES (11, TO_DATE('01-MAY-1999','DD-MON-YYYY'),
'online',300,0,30,301,333);


1 row created.


SQL> INSERT INTO hr.tab_part1
VALUES (12, TO_DATE('01-SEP-1999','DD-MON-YYYY'),
'direct',400,0,40,401,444);


1 row created.


SQL> INSERT INTO hr.tab_part1
VALUES (13, TO_DATE('01-FEB-2000','DD-MON-YYYY'),
'direct',500,0,50,501,555);
```

```
1 row created.

SQL> COMMIT;

Commit complete.

SQL>
```

8. Check that the partial global index TAB_PART1_GIDX_ORDERMODE is used while performing queries on the highly selective column ORDER_MODE to access a single value such as direct. The optimizer has to rely on a full scan of the non-indexed partitions for the value such as online stored in rows in partitions that are not indexed.

   a. Collect statistics for the table.

```
SQL> EXEC dbms_stats.gather_table_stats ('HR','TAB_PART1')

PL/SQL procedure successfully completed.

SQL>
```

   b. Generate a plan for a query accessing the value direct of the ORDER_MODE in the table where there are few rows.

```
SQL> EXPLAIN PLAN FOR
  2  SELECT order_mode, order_status FROM hr.tab_part1
  3  WHERE order_mode='direct';

Explained.

SQL> select * from table(dbms_xplan.display) ;

PLAN_TABLE_OUTPUT
----------------------------------------------------------------
-
Plan hash value: 3230465927

----------------------------------------------------------------
-
---------------------------------------------------------


| Id  | Operation                                    | Name
| Rows   | Bytes | Cost (%CPU)| Time     | Pstart| Pstop |

----------------------------------------------------------------
----------------
---------------------------------------------------------
```

```
|   0 | SELECT STATEMENT                             |
|      2 |    18 |    30   (0)| 00:00:01 |       |       |

|   1 |  VIEW                                        | VW_TE_2
|      3 |    57 |    30   (0)| 00:00:01 |       |       |

|   2 |   UNION-ALL                                  |
|        |       |            |          |       |       |

|*  3 |    TABLE ACCESS BY GLOBAL INDEX ROWID BATCHED| TAB_PART1
|      1 |    17 |     3   (0)| 00:00:01 | ROWID | ROWID |

|*  4 |     INDEX RANGE SCAN                         |
TAB_PART1_GIDX_ORDERMODE
|      2 |       |     1   (0)| 00:00:01 |       |       |

|   5 |    PARTITION RANGE OR                        |
|      2 |    34 |    27   (0)| 00:00:01 |KEY(OR)|KEY(OR)|

|*  6 |     TABLE ACCESS FULL                        | TAB_PART1
|      2 |    34 |    27   (0)| 00:00:01 |KEY(OR)|KEY(OR)|

----------------------------------------------------------------
----------------
---------------------------------------------------------



Predicate Information (identified by operation id):
---------------------------------------------------


   3 - filter("TAB_PART1"."ORDER_DATE"<TO_DATE(' 1999-10-01
00:00:00', 'syyyy-mm-dd hh24:mi:ss') AND
              "TAB_PART1"."ORDER_DATE">=TO_DATE(' 1999-07-01
00:00:00', 'syyyy-mm-dd hh24:mi:ss') OR
"TAB_PART1"."ORDER_DATE"<TO_DATE('
              1999-03-01 00:00:00', 'syyyy-mm-dd hh24:mi:ss'))
   4 - access("ORDER_MODE"='direct')
   6 - filter("ORDER_MODE"='direct' AND
("TAB_PART1"."ORDER_DATE">=TO_DATE(' 1999-10-01 00:00:00',
'syyyy-mm-dd hh24:mi:ss') AND
              "TAB_PART1"."ORDER_DATE"<TO_DATE(' 2010-03-01
00:00:00', 'syyyy-mm-dd hh24:mi:ss') OR
"TAB_PART1"."ORDER_DATE"<TO_DATE('
              1999-07-01 00:00:00', 'syyyy-mm-dd hh24:mi:ss')
AND "TAB_PART1"."ORDER_DATE">=TO_DATE(' 1999-03-01 00:00:00',
'syyyy-mm-ddhh24:mi:ss')))


25 rows selected.


SQL>
```

The partial global index is used to access the two partitions of the table where the rows containing the `direct` value in the `ORDER_MODE` column are stored.

9. Load the `ord_p1` partition. This partition is indexed. So both the partial global and local indexes will be updated.

```
SQL> INSERT INTO hr.tab_part1 PARTITION (ord_p1)
  2  SELECT * FROM hr.tab_part1 PARTITION (ord_p1);

1 row created.
SQL> /

2 rows created.
SQL> /

4 rows created.
SQL> /

8 rows created.
SQL> /

16 rows created.

SQL> /

32 rows created.
SQL> /

64 rows created.

SQL> /

128 rows created.

SQL> /

256 rows created.
```

```
SQL> /

512 rows created.

SQL> /

1024 rows created.

SQL> COMMIT;

Commit complete.

SQL>
```

10. Generate the plan for a query by using the key of the partial global index.

```
SQL> EXEC dbms_stats.gather_table_stats ('HR','TAB_PART1')

PL/SQL procedure successfully completed.

SQL>
```

```
SQL> EXPLAIN PLAN FOR
  2  SELECT order_mode, order_status FROM hr.tab_part1 WHERE
order_mode='direct';

Explained.

SQL>
```

11. You see that the partial local index is used to access the `ord_p3` partition.

```
SQL> select * from table(dbms_xplan.display) ;

PLAN_TABLE_OUTPUT
----------------------------------------------------------------
-
Plan hash value: 1639651856


----------------------------------------------------------------
-
| Id  | Operation                                             | Name
| Rows   | Bytes | Cost (%CPU)| Time     | Pstart| Pstop |
----------------------------------------------------------------
-
|   0 | SELECT STATEMENT                                      |
|  2050 | 18450 |    43   (0)| 00:00:01 |       |       |
|   1 |  VIEW                                         | VW_TE_2
|  3600 | 68400 |    43   (0)| 00:00:01 |       |       |
|   2 |   UNION-ALL                                   |
|       |       |            |          |       |       |
|   3 |    CONCATENATION                              |
|       |       |            |          |       |       |
|   4 |     PARTITION RANGE SINGLE                    |
|  2048 | 34816 |    14   (0)| 00:00:01 |     1 |     1 |
|*  5 |      TABLE ACCESS FULL                        |
TAB_PART1                 |  2048 | 34816 |    14   (0)| 00:00:01
|     1 |     1 |
|   6 |     PARTITION RANGE SINGLE                    |
|     1 |    17 |     2   (0)| 00:00:01 |     3 |     3 |
|*  7 |      TABLE ACCESS BY LOCAL INDEX ROWID BATCHED|
TAB_PART1                 |     1 |    17 |     2   (0)| 00:00:01
|     3 |     3 |
|*  8 |       INDEX RANGE SCAN                        |
TAB_PART1_LIDX_ORDERDATE |     1 |       |     1   (0)| 00:00:01
|     3 |     3 |
|   9 |    PARTITION RANGE OR                         |
|  1551 | 26367 |    27   (0)| 00:00:01 |KEY(OR)|KEY(OR)|
|* 10 |     TABLE ACCESS FULL                         |
TAB_PART1                 |  1551 | 26367 |    27   (0)| 00:00:01
|KEY(OR)|KEY(OR)|
----------------------------------------------------------------
-


Predicate Information (identified by operation id):
---------------------------------------------------


   5 - filter("ORDER_MODE"='direct' AND
"TAB_PART1"."ORDER_DATE"<TO_DATE(' 1999-03-01 00:00:00', 'syyyy-
mm-dd hh24:mi:ss'))
   7 - filter("ORDER_MODE"='direct')
   8 - access("TAB_PART1"."ORDER_DATE">=TO_DATE(' 1999-07-01
00:00:00', 'syyyy-mm-dd hh24:mi:ss') AND
              "TAB_PART1"."ORDER_DATE"<TO_DATE(' 1999-10-01
00:00:00', 'syyyy-mm-dd hh24:mi:ss'))
       filter(LNNVL("TAB_PART1"."ORDER_DATE"<TO_DATE(' 1999-03-
01 00:00:00', 'syyyy-mm-dd hh24:mi:ss')))
  10 - filter(("TAB_PART1"."ORDER_DATE">=TO_DATE(' 1999-10-01
00:00:00', 'syyyy-mm-dd hh24:mi:ss') AND
              "TAB_PART1"."ORDER_DATE"<TO_DATE(' 2010-03-01
00:00:00', 'syyyy-mm-dd hh24:mi:ss') OR
"TAB_PART1"."ORDER_DATE"<TO_DATE('
              1999-07-01 00:00:00', 'syyyy-mm-dd hh24:mi:ss')
AND "TAB_PART1"."ORDER_DATE">=TO_DATE(' 1999-03-01 00:00:00',
'syyyy-mm-dd
              hh24:mi:ss')) AND "ORDER_MODE"='direct')


30 rows selected.


SQL>
```

12. Generate the plan for a query on the non-indexed partition `ord_p2`. A table access full is performed on the partition.

```
SQL> EXPLAIN PLAN FOR
     SELECT order_mode, order_status FROM hr.tab_part1
     WHERE order_date >=TO_DATE('1999-03-01 00:00:00','syyyy-mm-
dd hh24:mi:ss')
     AND   order_date <TO_DATE('1999-07-01 00:00:00','syyyy-mm-
dd hh24:mi:ss');


Explained.


SQL> select * from table(dbms_xplan.display) ;


PLAN_TABLE_OUTPUT
-----------------------------------------------------------------
-
Plan hash value: 2198373237


-----------------------------------------------------------------
-| Id  | Operation              | Name       | Rows   | Bytes |
Cost (%CPU)| Time      | Pstart| Pstop |
-----------------------------------------------------------------
-|   0 | SELECT STATEMENT       |            |      1 |    17 |
14   (0)| 00:00:01 |       |       |
|   1 |  PARTITION RANGE SINGLE|            |      1 |    17 |
14   (0)| 00:00:01 |     2 |     2 |
|   2 |   TABLE ACCESS FULL    | TAB_PART1 |      1 |    17 |
14   (0)| 00:00:01 |     2 |     2 |
-----------------------------------------------------------------
-


9 rows selected.


SQL>
```

13. Generate the plan for a query on the indexed partition `ord_p1` and `ord_p3`. A full scan is performed on the large partition `ord_p1` and the partial local index is used to access the `ord_p3` partition rows.

```
SQL> EXPLAIN PLAN FOR
  2  SELECT order_mode, order_status FROM hr.tab_part1
  3  WHERE order_date <TO_DATE('1999-03-01 00:00:00','syyyy-mm-
dd hh24:mi:ss')
  4   OR (order_date between TO_DATE('1999-07-01
00:00:00','syyyy-mm-dd hh24:mi:ss') AND TO_DATE('1999-10-01
00:00:00', 'syyyy-mm-dd hh24:mi:ss'));


Explained.


SQL> select * from table(dbms_xplan.display) ;


PLAN_TABLE_OUTPUT
----------------------------------------------------------------
-
Plan hash value: 2109685904
----------------------------------------------------------------
-
| Id  | Operation                                        | Name
| Rows   | Bytes | Cost (%CPU)| Time      | Pstart| Pstop |
----------------------------------------------------------------
-
|   0 | SELECT STATEMENT                                 |
|  3337 | 56729 |    30   (0)| 00:00:01 |       |       |
|   1 |  VIEW                                            | VW_TE_2
|  6621 |  122K|    30   (0)| 00:00:01 |       |       |
|   2 |   UNION-ALL                                      |
|       |       |            |          |       |       |
|   3 |    CONCATENATION                                 |
|       |       |            |          |       |       |
|   4 |     PARTITION RANGE SINGLE                       |
|     1 |    17 |     2   (0)| 00:00:01 |     3 |     3 |
|   5 |      TABLE ACCESS BY LOCAL INDEX ROWID BATCHED|
TAB_PART1                |     1 |    17 |     2   (0)| 00:00:01
|     3 |     3 |
|*  6 |       INDEX RANGE SCAN                           |
TAB_PART1_LIDX_ORDERDATE |     1 |       |     1   (0)| 00:00:01
|     3 |     3 |
|   7 |     PARTITION RANGE SINGLE                       |
|  4095 | 69615 |    14   (0)| 00:00:01 |     1 |     1 |
|*  8 |      TABLE ACCESS FULL                           |
TAB_PART1                |  4095 | 69615 |    14   (0)| 00:00:01
|     1 |     1 |
|   9 |     PARTITION RANGE OR                         |
|  2525 | 42925 |    14   (0)| 00:00:01 |KEY(OR)|KEY(OR)|
|* 10 |      TABLE ACCESS FULL                         |
TAB_PART1                   |  2525 | 42925 |    14   (0)| 00:00:01
|KEY(OR)|KEY(OR)|
----------------------------------------------------------------
-
Predicate Information (identified by operation id):
---------------------------------------------------


   6 - access("TAB_PART1"."ORDER_DATE">=TO_DATE(' 1999-07-01
00:00:00', 'syyyy-mm-dd hh24:mi:ss') AND
              "TAB_PART1"."ORDER_DATE"<TO_DATE(' 1999-10-01
00:00:00', 'syyyy-mm-dd hh24:mi:ss'))
       filter("ORDER_DATE"<TO_DATE(' 1999-03-01 00:00:00',
'syyyy-mm-dd hh24:mi:ss') OR "ORDER_DATE">=TO_DATE(' 1999-07-01
              00:00:00', 'syyyy-mm-dd hh24:mi:ss') AND
"ORDER_DATE"<=TO_DATE(' 1999-10-01 00:00:00', 'syyyy-mm-dd
hh24:mi:ss'))
   8 - filter("TAB_PART1"."ORDER_DATE"<TO_DATE(' 1999-03-01
00:00:00', 'syyyy-mm-dd hh24:mi:ss') AND ("ORDER_DATE"<TO_DATE('
              1999-03-01 00:00:00', 'syyyy-mm-dd hh24:mi:ss') OR
"ORDER_DATE">=TO_DATE(' 1999-07-01 00:00:00', 'syyyy-mm-dd
hh24:mi:ss') AND
              "ORDER_DATE"<=TO_DATE(' 1999-10-01 00:00:00',
'syyyy-mm-dd hh24:mi:ss')) AND
(LNNVL("TAB_PART1"."ORDER_DATE">=TO_DATE('
              1999-07-01 00:00:00', 'syyyy-mm-dd hh24:mi:ss'))
OR LNNVL("TAB_PART1"."ORDER_DATE"<TO_DATE(' 1999-10-01
00:00:00', 'syyyy-mm-dd
              hh24:mi:ss'))))
  10 - filter(("TAB_PART1"."ORDER_DATE">=TO_DATE(' 1999-10-01
00:00:00', 'syyyy-mm-dd hh24:mi:ss') AND
              "TAB_PART1"."ORDER_DATE"<TO_DATE(' 2010-03-01
00:00:00', 'syyyy-mm-dd hh24:mi:ss') OR
"TAB_PART1"."ORDER_DATE"<TO_DATE('
              1999-07-01 00:00:00', 'syyyy-mm-dd hh24:mi:ss')
AND "TAB_PART1"."ORDER_DATE">=TO_DATE(' 1999-03-01 00:00:00',
'syyyy-mm-dd
              hh24:mi:ss')) AND ("ORDER_DATE">=TO_DATE(' 1999-
07-01 00:00:00', 'syyyy-mm-dd hh24:mi:ss') AND
"ORDER_DATE"<=TO_DATE(' 1999-10-01
              00:00:00', 'syyyy-mm-dd hh24:mi:ss') OR
"ORDER_DATE"<TO_DATE(' 1999-03-01 00:00:00', 'syyyy-mm-dd
hh24:mi:ss')))


35 rows selected.
```

```
SQL> EXIT
$
```

Practices for Lesson 22: SQL Enhancements and Migration Assistant for Unicode

Chapter 22

Practices for Lesson 22

Practices Overview

In the practice for this lesson, you use the extended data type column to create columns of 32767 bytes long and the row-limiting clause to limit the rows resulting from queries.

Practice 22-1: Using 32K VARCHAR2 Data Type

Overview

In this practice, you create a new table with a column of data type VARCHAR2(32767).

Tasks

1. Connect to the source database orcl.

```
$ . oraenv
ORACLE_SID = [cdb2] ? orcl
The Oracle base remains unchanged with value /u01/app/oracle

$ sqlplus / as sysdba

Connected to:
Oracle Database 12c Enterprise Edition Release 12.1.0.0.2 -
64bit Production
With the Partitioning, OLAP, Data Mining and Real Application
Testing options

SQL>
```

2. Create a table LONG_VARCHAR with a column VARCHAR2(32767).

```
SQL> CREATE TABLE long_varchar(id NUMBER,vc VARCHAR2(32767));
CREATE TABLE long_varchar(id NUMBER,vc VARCHAR2(32767))
                                                    *
ERROR at line 1:
ORA-00910: specified length too long for its datatype

SQL>
```

3. Set the instance parameter MAX_STRING_SIZE to EXTENDED.

```
SQL> alter system set MAX_STRING_SIZE =EXTENDED;
alter system set MAX_STRING_SIZE =EXTENDED
*
ERROR at line 1:
ORA-02097: parameter cannot be modified because specified value
is invalid
ORA-14694: database must in UPGRADE mode to begin
MAX_STRING_SIZE migration

SQL>
```

4. Configure the database to be compatible with extended data type columns.

   a. Restart the database instance.

```
SQL> shutdown immediate
Database closed.
Database dismounted.
ORACLE instance shut down.
SQL>
```

.

```
SQL> startup upgrade
ORACLE instance started.

Total System Global Area 1686925312 bytes
Fixed Size                  2261160 bytes
Variable Size             989859672 bytes
Database Buffers          687865856 bytes
Redo Buffers                6938624 bytes
Database mounted.
Database opened.
SQL>
```

   b. Set the instance parameter MAX_STRING_SIZE to the EXTENDED value.

```
SQL> ALTER SYSTEM SET MAX_STRING_SIZE = EXTENDED;

System altered.

SQL>
```

   c. Execute the $ORACLE_HOME/rdbms/admin/utl32k.sql script as SYSDBA. Be aware that the script may last very long (around one hour). Although, at the very last step when the function is being dropped and the script stuck, you can start another session as SYSDBA and perform task 5 to check if the creation of the table is possible with a VARCHAR2(32767) data type column.

```
SQL> @$ORACLE_HOME/rdbms/admin/utl32k.sql


Session altered.


DOC>#################################################################
###########
DOC>#################################################################
###########
DOC>   The following statement will cause an "ORA-01722: invalid
number"
DOC>   error if the database has not been opened for UPGRADE.
DOC>
DOC>   Perform a "SHUTDOWN ABORT"  and
DOC>   restart using UPGRADE.
DOC>############################################################
###########
DOC>############################################################
###########
DOC>#

no rows selected

DOC>############################################################
###########
DOC>############################################################
###########
DOC>   The following statement will cause an "ORA-01722: invalid
number"
DOC>   error if the database does not have compatible >= 12.0.0
DOC>
DOC>   Set compatible >= 12.0.0 and retry.
DOC>############################################################
###########
DOC>############################################################
###########
DOC>#

PL/SQL procedure successfully completed.

Session altered.

0 rows updated.

Commit complete.

1671 rows updated.

Commit complete.

System altered.

PL/SQL procedure successfully completed.

Commit complete.

System altered.
```

```
Session altered.


PL/SQL procedure successfully completed.


No errors.


Session altered.


Package altered.


TIMESTAMP
----------------------------------------------------------------
----------------
COMP_TIMESTAMP UTLRP_BGN  2012-07-11 08:29:59


DOC>   The following PL/SQL block invokes UTL_RECOMP to
recompile invalid
DOC>   objects in the database. Recompilation time is
proportional to the
DOC>   number of invalid objects in the database, so this
command may take
DOC>   a long time to execute on a database with a large number
of invalid
DOC>   objects.
DOC>
DOC>   Use the following queries to track recompilation
progress:
DOC>
DOC>   1. Query returning the number of invalid objects
remaining. This
DOC>      number should decrease with time.
DOC>         SELECT COUNT(*) FROM obj$ WHERE status IN (4, 5,
6);
DOC>
DOC>   2. Query returning the number of objects compiled so far.
This number
DOC>      should increase with time.
DOC>         SELECT COUNT(*) FROM UTL_RECOMP_COMPILED;
DOC>
DOC>   This script automatically chooses serial or parallel
recompilation
DOC>   based on the number of CPUs available (parameter
cpu_count) multiplied
DOC>   by the number of threads per CPU (parameter
parallel_threads_per_cpu).
```

```
DOC>   On RAC, this number is added across all RAC nodes.
DOC>
DOC>   UTL_RECOMP uses DBMS_SCHEDULER to create jobs for
parallel
DOC>   recompilation. Jobs are created without instance affinity
so that they
DOC>   can migrate across RAC nodes. Use the following queries
to verify
DOC>   whether UTL_RECOMP jobs are being created and run
correctly:
DOC>
DOC>   1. Query showing jobs created by UTL_RECOMP
DOC>         SELECT job_name FROM dba_scheduler_jobs
DOC>            WHERE job_name like 'UTL_RECOMP_SLAVE_%';
DOC>
DOC>   2. Query showing UTL_RECOMP jobs that are running
DOC>         SELECT job_name FROM dba_scheduler_running_jobs
DOC>            WHERE job_name like 'UTL_RECOMP_SLAVE_%';
DOC>#

PL/SQL procedure successfully completed.

TIMESTAMP
----------------------------------------------------------------
----------------
COMP_TIMESTAMP UTLRP_END  2012-07-11 08:32:14

DOC> The following query reports the number of objects that have
compiled
DOC> with errors.
DOC>
DOC> If the number is higher than expected, please examine the
error
DOC> messages reported with each object (using SHOW ERRORS) to
see if they
DOC> point to system misconfiguration or resource constraints
that must be
DOC> fixed before attempting to recompile these objects.
DOC>#

OBJECTS WITH ERRORS
-------------------
                  4
```

```
DOC> The following query reports the number of errors caught
during
DOC> recompilation. If this number is non-zero, please query the
error
DOC> messages in the table UTL_RECOMP_ERRORS to see if any of
these errors
DOC> are due to misconfiguration or resource constraints that
must be
DOC> fixed before objects can compile successfully.
DOC>#

ERRORS DURING RECOMPILATION
---------------------------
                          4

Function created.

PL/SQL procedure successfully completed.

Function dropped.

...Database user "SYS", database schema "APEX_040200", user#
"98" 06:52:09
...Compiled 0 out of 2992 objects considered, 0 failed
compilation 06:52:09
...263 packages
...255 package bodies
...453 tables
...11 functions
...16 procedures
...3 sequences
...458 triggers
...1316 indexes
...207 views
...0 libraries
...6 types
...0 type bodies
...0 operators
...0 index types
...Begin key object existence check 06:52:09
...Completed key object existence check 06:52:10
...Setting DBMS Registry 06:52:10
...Setting DBMS Registry Complete 06:52:10
```

```
...Exiting validate 06:52:10

PL/SQL procedure successfully completed.



0 rows updated.



Commit complete.
SQL>
```

d. Restart the database in normal mode.

```
SQL> shutdown immediate
Database closed.
Database dismounted.
ORACLE instance shut down.
SQL> startup
ORACLE instance started.

Total System Global Area 1686925312 bytes
Fixed Size                  2261160 bytes
Variable Size             989859672 bytes
Database Buffers          687865856 bytes
Redo Buffers                6938624 bytes
Database mounted.
Database opened.
SQL>
```

e. Verify that the `MAX_STRING_SIZE` is set to `EXTENDED`.

```
SQL> show parameter MAX_STRING_SIZE

NAME                                 TYPE        VALUE
------------------------------------ ----------- --------------
max_string_size                      string      EXTENDED
SQL>
```

5. Create a table with an extended data type column of 32767 bytes.

```
SQL> CREATE TABLE long_varchar(id NUMBER,vc VARCHAR2(32767));
Table created.

SQL> DESC  long_varchar
 Name                               Null?    Type
 --------------------------------- -------- --------------------
 ID                                          NUMBER
 VC                                          VARCHAR2(32767)

SQL>
```

Practice 22-2: Quering a Table Using a SQL Row-Limiting Clause

Overview

In this practice, you limit the number or rows returned by a query that orders data.

Tasks

1. Count the number of rows in the HR.EMPLOYEES table.

```
SQL> select count(*) from hr.employees;

  COUNT(*)
----------
       83

SQL>
```

2. Select the EMPLOYEE_ID and LAST_NAME of the first 10 employees ordered by their last name.

```
SQL> SELECT employee_id, last_name FROM hr.employees
  2  ORDER BY last_name
  3  FETCH FIRST 10 ROWS ONLY;

EMPLOYEE_ID LAST_NAME
----------- -------------------------
    174     Abel
    166     Ande
    105    Austin
    204     Baer
    167     Banda
    172     Bates
    192     Bell
    151     Bernstein
    129     Bissot
    169     Bloom

10 rows selected.

SQL>
```

You see the first 10 employees ordered by their last name. The first of them has a name starting with letter A.

3. Select the EMPLOYEE_ID, LAST_NAME and HIRE_DATE of the first 10 employees ordered by their hire date.

```
SQL> SELECT employee_id,last_name, hire_date FROM hr.employees
  2  ORDER BY hire_date
  3  FETCH FIRST 10 ROWS ONLY;

EMPLOYEE_ID LAST_NAME                 HIRE_DATE
----------- ------------------------- ---------
        102 De Haan                   13-JAN-01
        203 Mavris                    07-JUN-02
        206 Gietz                     07-JUN-02
        205 Higgins                   07-JUN-02
        204 Baer                      07-JUN-02
        109 Faviet                    16-AUG-02
        108 Greenberg                 17-AUG-02
        114 Raphaely                  07-DEC-02
        122 Kaufling                  01-MAY-03
        115 Khoo                      18-MAY-03

10 rows selected.

SQL>
```

You see the first 10 employees ordered by their hire date. The first hired one was the employee De Haan.

4. Select the EMPLOYEE_ID, LAST_NAME and HIRE_DATE of the next 5 employees ordered by their hire date coming after the previous ones.

```
SQL> SELECT employee_id,last_name, hire_date FROM hr.employees
  2  ORDER BY hire_date
  3  OFFSET 10 ROWS FETCH NEXT 5 ROWS ONLY;

EMPLOYEE_ID LAST_NAME                 HIRE_DATE
----------- ------------------------- ---------
        100 King                      17-JUN-03
        137 Ladwig                    14-JUL-03
        200 Whalen                    17-SEP-03
        141 Rajs                      17-OCT-03
        184 Sarchand                  27-JAN-04


SQL>
```

5. Select the EMPLOYEE_ID and LAST_NAME of the first 8% of the employees ordered by their employee identifier.

```
SQL> SELECT employee_id, last_name FROM hr.employees
  2  ORDER BY employee_id
  3  FETCH FIRST 8 PERCENT ROWS ONLY;

EMPLOYEE_ID LAST_NAME
----------- -------------------------
        100 King
        101 Kochhar
        102 De Haan
        103 Hunold
        104 Ernst
        105 Austin
        106 Pataballa

7 rows selected.

SQL> EXIT
$
```

You see the first seven employees ordered by their employee identifier. The first one is the employee 100. 8 % of 83 rows correspond to approximately 7 rows.