

Hardware and Software
Engineered to Work Together



Oracle Database 12c: New Features for Administrators

Activity Guide – Volume I

D77758GC20

Edition 2.0 | November 2014 | D87750

Learn more from Oracle University at oracle.com/education/

Copyright © 2014, Oracle and/or its affiliates. All rights reserved.

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

Andy Rivenes, James Spiller, Donna Keesling, Maria Billings, Lachlan Williams, Peter Fusek, Mark Fuller, Gregg Christman, Dimpi Sarmah, Kevin Jernigan, Branislav Valny, Frank Fu, Joel Goodman, Gerlinde Frenzen, Harald Van Breederode, Hermann Baer, Jim Stenoish, Mark Drake, Beda Hammerschmidt, Prabhaker Gongloor, Patrick Wheeler, Maria Colgan, Jesse Kamp, Paul Needham, Pat Huey, Roy F Swonger, Ron Soltani, Sue Lee, Sharath Bhujani

Editors

Anwasha Ray, Malavika Jinka, Smita Kommini

Graphic Designers

Seema Bopaiah, Maheshwari Krishnamurthy

Publishers

Jobi Varghese, Pavithran Adka, Joseph Fernandez

This book was published using: Oracle Tutor

Table of Contents

Practices for Lesson 1: Introduction	1-1
Practices for Lesson 1	1-2
Practices for Lesson 2: Enterprise Manager Cloud Control and Other Tools	2-1
Practices for Lesson 2: Overview	2-2
Practice 2-1: Accessing Enterprise Manager	2-3
Practice 2-2: Adding a Database Instance as a New Target Monitored by EM Cloud Control	2-6
Practice 2-3: Creating New Named Credentials	2-9
Practice 2-4: Testing the Named Credential	2-10
Practices for Lesson 3: Basics of Multitenant Container Database and Pluggable Databases	3-1
Practices for Lesson 3: Overview	3-2
Practice 3-1: Exploring CDB Architecture and Structures	3-3
Practices for Lesson 4: Creating a Multitenant Container Database and Pluggable Databases	4-1
Practices for Lesson 4: Overview	4-2
Practice 4-1: Creating a New CDB	4-3
Practice 4-2: Exploring CDB and PDB Structures	4-7
Practice 4-3: Creating a PDB from Seed	4-18
Practice 4-4: Cloning PDB Within the Same CDB	4-26
Practice 4-5: Cloning a Non-CDB into a CDB	4-38
Practice 4-6: Merging All PDBs of CDBs into a Single CDB	4-45
Practices for Lesson 5: Managing a Multitenant Container Database and Pluggable Databases	5-1
Practices for Lesson 5: Overview	5-2
Practice 5-1: Shutdown and Startup of the CDB	5-3
Practice 5-2: Closing and Opening a PDB	5-8
Practice 5-3: Changing PDBs' Open Mode	5-14
Practice 5-4: Instance Parameter Changes: Impact on PDBs (Optional)	5-16
Practices for Lesson 6: Managing Tablespaces and Users in a CDB and PDBs	6-1
Practices for Lesson 6: Overview	6-2
Practice 6-1: Managing Tablespaces	6-3
Practice 6-2: Managing Common and Local Users	6-10
Practice 6-3: Managing Local and Common Roles	6-19
Practice 6-4: Managing Local and Common Privileges	6-27
Practices for Lesson 7: Backup, Recovery, Flashback CDB and PDBs	7-1
Practices for Lesson 7: Overview	7-2
Practice 7-1: Cold CDB Backup	7-3
Practice 7-2: RMAN Whole CDB Backup	7-5
Practice 7-3: RMAN CDB / PDB Backup	7-10
Practice 7-4: RMAN Recovery from PDB Data File Loss	7-12
Practice 7-5: SQL PDB Hot Backup (Optional)	7-17
Practice 7-6: SQL Control File Backup (Optional)	7-19
Practice 7-7: RMAN Recovery from Control File Loss (Optional)	7-21
Practice 7-8: RMAN Recovery from Redo Log File Member Loss (Optional)	7-26
Practice 7-9: RMAN Recovery from SYSTEM Root Data File Loss (Optional)	7-29
Practice 7-10: RMAN Recovery from Non-Essential Root Data File Loss (Optional)	7-34
Practice 7-11: PITR on PDB Tablespaces (Optional)	7-39
Practice 7-12: Flashback from Common User Drop (Optional)	7-52

Practices for Lesson 8: Heat Map, Automatic Data Optimization and Online Datafile Move.....	8-1
Practices for Lesson 8: Overview.....	8-2
Practice 8-1: Enabling Heat Map	8-3
Practice 8-2: Automatic Data Optimization – Creating a TIER Policy.....	8-5
Practice 8-3: Automatic Data Optimization – Creating a COMPRESS Policy	8-13
Practice 8-4: Cleanup ADO Policies and Heat Map Statistics.....	8-21
Practice 8-5: Moving Data File Online.....	8-23
Practices for Lesson 9: In-Database Archiving and Temporal Validity	9-1
Practices for Lesson 9: Overview.....	9-2
Practice 9-1: In-Database Archiving – Row-Archival	9-3
Practice 9-2: Temporal Validity	9-8
Practice 9-3: Collecting User Context in FDA History Tables (Optional).....	9-17
Practice 9-4: Cleaning Up FDA.....	9-23
Practices for Lesson 10: Auditing	10-1
Practices for Lesson 10: Overview.....	10-2
Practice 10-1: Enabling Unified Auditing	10-3
Practice 10-2: Auditing RMAN Backup and Recovery Operations	10-9
<i>Practice 10-3: Auditing SYS and End-Users (Optional).....</i>	<i>10-13</i>
<i>Practice 10-4: Excluding DBSNMP Login Events (Optional).....</i>	<i>10-18</i>
Practices for Lesson 11: Privileges	11-1
Practices for Lesson 11: Overview.....	11-2
Practice 11-1: Managing Password File with SYSBACKUP Entry	11-3
Practice 11-2: Capturing Privileges	11-8
<i>Practice 11-3: Capturing Privileges Used Through Roles (Optional).....</i>	<i>11-17</i>
<i>Practice 11-4: Capturing Privileges Used In Contexts (Optional)</i>	<i>11-21</i>
<i>Practice 11-5: Using INHERIT PRIVILEGES Privilege (Optional)</i>	<i>11-24</i>
<i>Practice 11-6: Using BEQUEATH Views (Optional)</i>	<i>11-28</i>
Practices for Lesson 12: Oracle Data Redaction.....	12-1
Practices for Lesson 12: Overview.....	12-2
Practice 12-1: Redacting Protected Column Values with FULL Redaction	12-3
Practice 12-2: Redacting Protected Column Values with PARTIAL Redaction (Optional).....	12-7
Practice 12-3: Cleaning Up Redaction Policies.....	12-10
Practice 12-4: Changing the Default Value for FULL Redaction (Optional)	12-12
Practices for Lesson 13: Recovery Manager - New Features and Temporal History Enhancements	13-1
Practices for Lesson 13: Overview.....	13-2
Practice 13-1: Using SYSBACKUP in RMAN	13-3
Practice 13-2: Recovering a Table by Using Table Recovery.....	13-8
Practices for Lesson 14: Real-Time Database Operation Monitoring	14-1
Practices for Lesson 14: Overview.....	14-2
Practice 14-1: Starting Enterprise Manager Database Express	14-3
Practice 14-2: Identifying and Starting Database Operations	14-7
<i>Practice 14-3: Identifying and Starting Database Load Operations (Optional)</i>	<i>14-14</i>
Practice 14-4: Cleaning Up.....	14-16
Practices for Lesson 15: Emergency Monitoring and Compare Period ADDM.....	15-1
Practices for Lesson 15: Overview.....	15-2
Practice 15-1: Using Emergency Monitoring.....	15-3
Practice 15-2: Cleaning Up.....	15-7
<i>Practice 15-3: Using Compare Period ADDM (Optional)</i>	<i>15-8</i>

Practices for Lesson 16: ADR and Network Enhancements	16-1
Practices for Lesson 16: Overview.....	16-2
Practice 16-1: Viewing ADR DDL Log File.....	16-3
Practices for Lesson 17: In-Memory Column Store	17-1
Practices for Lesson 17: Overview.....	17-2
Practice 17-1: Configuring IM Column Store.....	17-3
Practice 17-2: Configuring In-Memory Tables.....	17-6
Practice 17-3: Querying In-Memory Tables	17-12
Practice 17-4: Exporting and Importing In-Memory Tables (Optional)	17-20
Lesson 17-5: Using In-Memory Column Store (Demonstration).....	17-25
Practices for Lesson 18: In-Memory Caching	18-1
Practices for Lesson 18: Overview.....	18-2
Practice 18-1: Using Automatic Big Table Caching.....	18-3
Practice 18-2: Setting and Monitoring Force Full Database Caching	18-12
Practices for Lesson 19: SQL Tuning Enhancements	19-1
Practices for Lesson 19: Overview.....	19-2
Practice 19-1: Using Dynamic Plans	19-3
Practice 19-2: Using Re-Optimization	19-9
Practices for Lesson 20: Resource Manager and Other Performance Enhancements	20-1
Practices for Lesson 20: Overview.....	20-2
Practice 20-1: Using CDB Resource Manager Plans and Directives.....	20-3
Practice 20-2: Using Multi-Process Multi-Threaded Architecture	20-14
Practices for Lesson 21: Tables, Indexes and Online Operations	21-1
Practices for Lesson 21: Overview.....	21-2
Practice 21-1: Using Invisible Table Columns.....	21-3
Practice 21-2: Using Advanced Row Compression.....	21-7
Practices for Lesson 22: Oracle Data Pump, SQL*Loader, and External Tables.....	22-1
Practices for Lesson 22: Overview.....	22-2
Practice 22-1: Exporting/Importing Databases in FULL TRANSPORTABLE Mode.....	22-3
<i>Practice 22-2: Loading Data Using SQL*Loader Express Mode (Optional)</i>	<i>22-15</i>
Practices for Lesson 23: Partitioning Enhancements.....	23-1
Practices for Lesson 23: Overview.....	23-2
Practice 23-1: Local and Global Partial Indexing on Partitioned Tables	23-3
Practices for Lesson 24: SQL Enhancements and Migration Assistant for Unicode	24-1
Practices for Lesson 24: Overview.....	24-2
Practice 24-1: Using 32K VARCHAR2 Data Type.....	24-3
Practice 24-2: Querying a Table Using a SQL Row-Limiting Clause (Optional).....	24-7
Practices for Appendix C: Schema and Data Changes Management	25-1
Practices for Appendix C: Overview.....	25-2
Appendix C-1: Using Schema Change Plans	25-3

Practices for Lesson 1: Introduction

Chapter 1

Practices for Lesson 1

Practices Overview

There are no practices for this lesson.

Practices for Lesson 2: Enterprise Manager Cloud Control and Other Tools

Chapter 2

Practices for Lesson 2: Overview

Practices Overview

Your system currently has Oracle Database 12c 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 12c 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 12c functionalities through the different menus and options. And lastly, you will add the `orcl` database as a monitored target.

Practice 2-1: Accessing Enterprise Manager

Overview

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

Assumptions

You reviewed the Oracle Enterprise Manager 12c: 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 4
Copyright (c) 1996, 2014 Oracle Corporation. All rights
reserved.
```

```
Starting Oracle Management Server...
```

```
Starting WebTier...
```

```
WebTier Successfully Started
```

```
Oracle Management Server Already Started
```

```
Oracle Management Server is Up
```

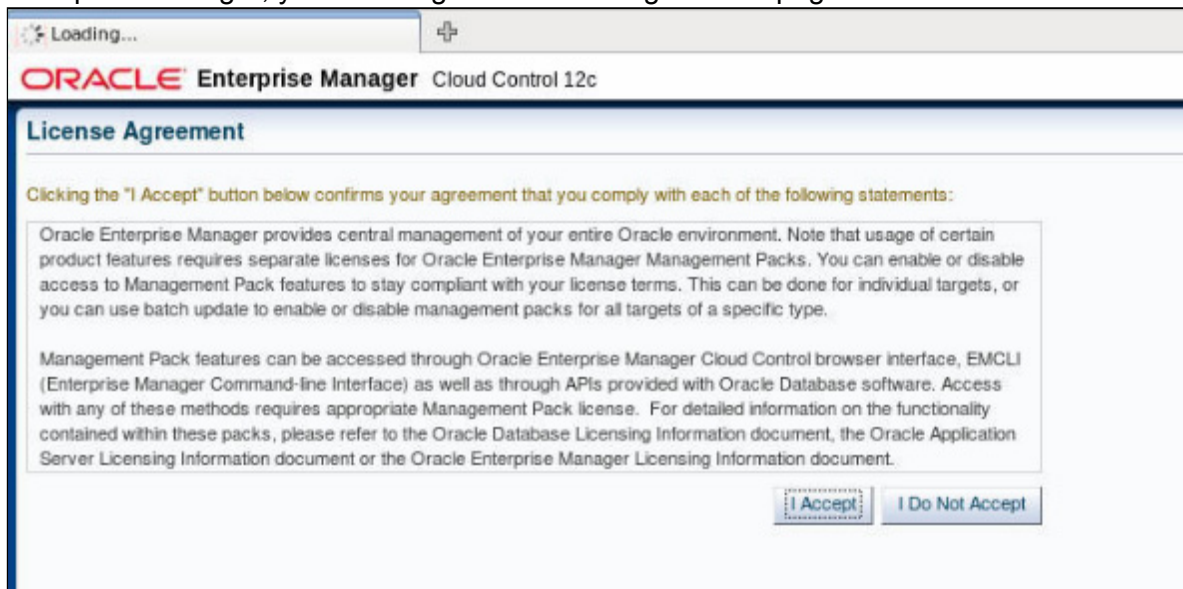
```
Starting BI Publisher Server ...
```

```
BI Publisher Server Already Started
```

```
BI Publisher Server is Up
```

```
$
```

3. Most probably, 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. Enter **sysman** in the User Name field and **orac1e123** in the Password field. Then click **Login**.
5. The Accessibility Preference page appears. The "Your accessibility preferences are presented because this is your first login. You can set these now, or at anytime by using Username menu." message appears. Click **I'll deal with this later**.
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.



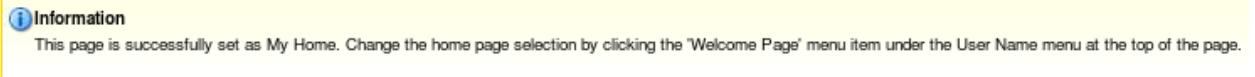
7. The “Welcome to Enterprise Manager Cloud Control 12c” page appears with choices, in the “Select Enterprise Manager Home page” section, such as:

- Welcome Page
- All Targets
- Sitemap
- Summary
- Databases
- Incidents
- SOA
- Middleware
- Composite Application
- Services
- Business Applications

In the **Select Enterprise Manager Home Page**, each choice has a preview and you can view any of the previews by clicking any image to get a larger preview. Then select a radio button to select your personal **Home Page**. The page also has global menus with the following choices: Enterprise Manager Overview, Latest Features, Learn More, and Getting Started.

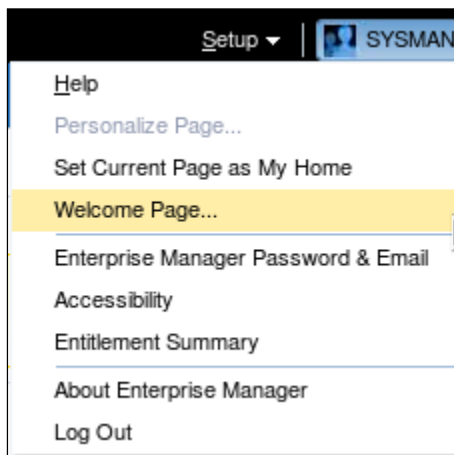
Preview any images that interest you.

8. Click the radio button next to the **Summary** choice. After being successfully set, it informs you how to change it.



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

Answer: SYSMAN > Welcome Page... menu.



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

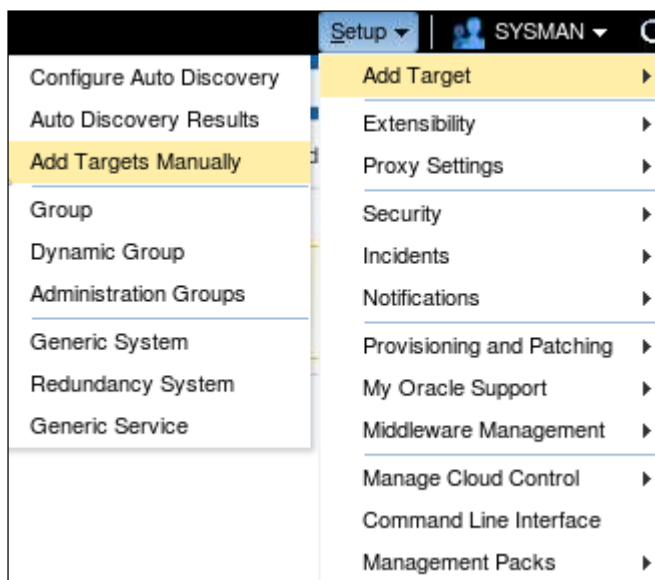
Assumptions

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

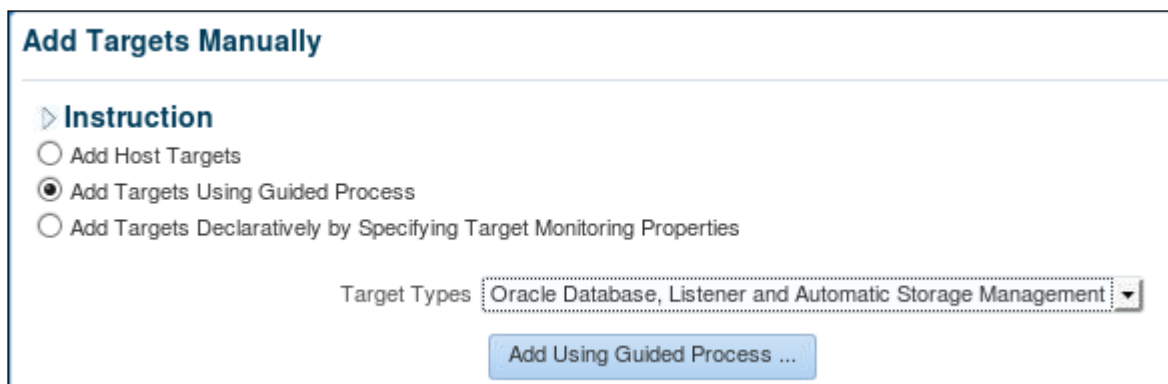
Tasks

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 Targets Using Guided Process”. Then in “Target Types”, choose “Oracle Database, Listener and Automatic Storage Management”. Click “Add Using Guided Process...” button.



- c. On the “Database Discovery : Search Criteria” page, in “Specify Host or Cluster”, click the magnifying glass to find your host. Select your host, then click “Next”.

- 1) On the “Database Discovery : Results” page, in the “Databases” list, select the **orcl** database.
- 2) 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 12.1.0.2.0 -
64bit
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
$
```

3) Enter oracle_4U for the “Monitor Password”.

Database Discovery: Results

Set Global Target Properties

Specify Group for Targets

Back

Databases

The following databases have been discovered on this host. Provide monitoring credentials and save the targets to start monitoring the database monitoring credentials for all the selected database targets using the 'Specify Common Monitoring Credentials' action. You can set Global Target targets or add them to a Target Group while saving the targets for monitoring.

View ▾

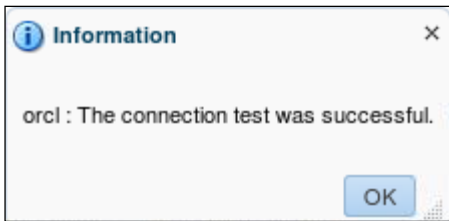
Specify Common Monitoring Credentials

Configure

Test Connection

	Target Name	Monitoring Credentials			Target Group
		Monitor Username	Monitor Password	Role	
<input type="checkbox"/>	cdb1 (Container Database)	db snmp		Normal ▾	
<input type="checkbox"/>	orcl2	db snmp		Normal ▾	
<input checked="" type="checkbox"/>	orcl	db snmp	*****	Normal ▾	
<input type="checkbox"/>	em12rep.example.com	db snmp		Normal ▾	

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



e. Click the “OK” button, then “Next” then “Save” buttons to complete the operation, and finally “Close”.

Practice 2-3: 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 Practice 2-2 that added 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:

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

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

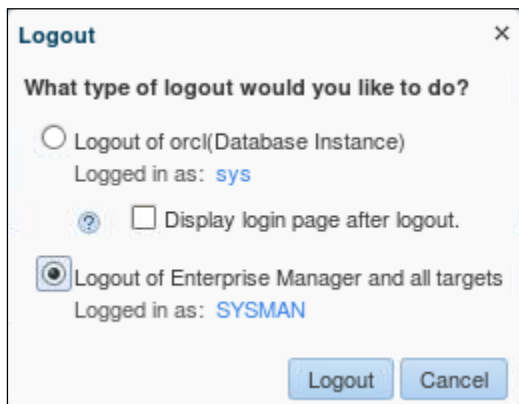
Practice 2-4: Testing the Named Credential

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`. Click `orcl`.
3. Click **Administration**, then **Storage** and then **Tablespaces**. The named credential `credorcl` is displayed in the **Database Login** page.
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. Under the SYSMAN menu at the top right hand, as soon as you click the Log Out button, the following screenshot is displayed.



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

Practices for Lesson 3: Basics of Multitenant Container Database and Pluggable Databases

Chapter 3

Practices for Lesson 3: Overview

Practices Overview

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

In Oracle Database 12c, 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 3-1: 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 `pgrep` UNIX command.

```
$ pgrep -lf cdb1
11567 ora_pmon_cdb1
11569 ora_psp0_cdb1
11571 ora_vktm_cdb1
11575 ora_gen0_cdb1
11577 ora_mman_cdb1
11581 ora_diag_cdb1
11583 ora_dbrm_cdb1
11585 ora_vkrn_cdb1
11587 ora_dia0_cdb1
11589 ora_dbw0_cdb1
11591 ora_lgwr_cdb1
11593 ora_ckpt_cdb1
11595 ora_lg00_cdb1
11597 ora_smon_cdb1
11599 ora_lg01_cdb1
11601 ora_reco_cdb1
11603 ora_lreg_cdb1
11605 ora_pxmn_cdb1
11607 ora_mmon_cdb1
11609 ora_mmln_cdb1
11611 ora_d000_cdb1
11613 ora_s000_cdb1
11635 ora_tmon_cdb1
11637 ora_tt00_cdb1
11639 ora_smco_cdb1
11641 ora_w000_cdb1
11643 ora_w001_cdb1
11645 ora_aqpc_cdb1
11649 ora_cjq0_cdb1
11651 ora_p000_cdb1
11653 ora_p001_cdb1
```

Copyright © 2014, Oracle and/or its affiliates. All rights reserved.

```

11655 ora_p002_cdb1
11657 ora_p003_cdb1
11659 ora_p004_cdb1
11661 ora_p005_cdb1
11663 ora_p006_cdb1
11665 ora_p007_cdb1
11974 ora_qm02_cdb1
11978 ora_q002_cdb1
11980 ora_q003_cdb1
12687 ora_w002_cdb1
13200 ora_w003_cdb1
13203 ora_w004_cdb1
13209 ora_w005_cdb1
13213 ora_w006_cdb1
13227 ora_w007_cdb1
30315 ora_p008_cdb1
30317 ora_p009_cdb1
30319 ora_p00a_cdb1
30321 ora_p00b_cdb1
30323 ora_p00c_cdb1
30325 ora_p00d_cdb1
32079 ora_j000_cdb1
32081 ora_j001_cdb1
$

```

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

```

$ . oraenv
ORACLE_SID = [orcl] ? cdb1
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 12.1.0.2.0 -
64bit Production
With the Partitioning, OLAP, Advanced Analytics and Real
Application Testing options
SQL>

```

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

```
SQL> SELECT name, cdb, con_id FROM v$databases;

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 12.1.0.2.0 - on 23-JAN-2014 03:58:05

Copyright (c) 1991, 2014, Oracle. All rights reserved.

Connecting to
(DESCRIPTION=(ADDRESS=(PROTOCOL=IPC)(KEY=EXTPROC1521)))
STATUS of the LISTENER
-----
Alias                     LISTENER
Version                   TNSLSNR for Linux: Version 12.1.0.2.0
- Production
Start Date                21-JAN-2014 10:07:53
Uptime                    1 days 17 hr. 50 min. 15 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/<Your_ServerName>/listener/alert/log.xml
Listening Endpoints Summary...
(DESCRIPTION=(ADDRESS=(PROTOCOL=tcp)(HOST=<Your_ServerName>)(PORT=1521)))
```

```
(DESCRIPTION=(ADDRESS=(PROTOCOL=ipc)(KEY=EXTPROC1521)))
(DESCRIPTION=(ADDRESS=(PROTOCOL=tcp)(HOST=<Your_ServerName>)(PORT=5500))(Presentation=HTTP)(Session=RAW))
DESCRIPTION=(ADDRESS=(PROTOCOL=tcp)(HOST=<Your_ServerName>)(PORT=5501))(Presentation=HTTP)(Session=RAW))
(DESCRIPTION=(ADDRESS=(PROTOCOL=tcp)(HOST=<Your_ServerName>)(PORT=5502))(Presentation=HTTP)(Session=RAW))
Services Summary...
Service "cdbl" has 1 instance(s).
  Instance "cdbl", status READY, has 1 handler(s) for this service...
Service "cdblXDB" has 1 instance(s).
  Instance "cdbl", status READY, has 1 handler(s) for this service...
Service "eml2rep" has 1 instance(s).
  Instance "eml2rep", status READY, has 1 handler(s) for this service...
Service "eml2repXDB" has 1 instance(s).
  Instance "eml2rep", 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 "cdbl", 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 12.1.0.2.0 - on 23-JAN-2014 04:03:36

Copyright (c) 1991, 2014, Oracle. All rights reserved.

Connecting to
 (DESCRIPTION=(ADDRESS=(PROTOCOL=IPC)(KEY=EXTPROC1521)))
The command completed successfully
$ lsnrctl status

LSNRCTL for Linux: Version 12.1.0.2.0 - on 23-JAN-2014 04:03:40

...
Listening Endpoints Summary...
  (DESCRIPTION=(ADDRESS=(PROTOCOL=ipc)(KEY=EXTPROC1521)))

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

b. Check services.

```
$ lsnrctl services

LSNRCTL for Linux: Version 12.1.0.2.0 - on 23-JAN-2014 04:05:15

Copyright (c) 1991, 2014, 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:0 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):
```

Copyright © 2014, Oracle and/or its affiliates. All rights reserved.

```

"D000" established:0 refused:0 current:0 max:1022
state:ready
    DISPATCHER <machine: <Your_ServerName>, pid: 8006>

    (ADDRESS=(PROTOCOL=tcp)(HOST=<Your_ServerName>)(PORT=48320))
    Service "eml2rep" has 1 instance(s).
        Instance "eml2rep", status READY, has 1 handler(s) for this
        service...
            Handler(s):
                "DEDICATED" established:2 refused:0 state:ready
                LOCAL SERVER
    Service "eml2repXDB" has 1 instance(s).
        Instance "eml2rep", status READY, has 1 handler(s) for this
        service...
            Handler(s):
                "D000" established:0 refused:0 current:0 max:1022
                state:ready
                DISPATCHER <machine: <Your_ServerName>, pid: 26154>

                (ADDRESS=(PROTOCOL=tcp)(HOST=<Your_ServerName>)(PORT=14734))
                Service "orcl" has 1 instance(s).
                    Instance "orcl", status READY, has 1 handler(s) for this
                    service...
                        Handler(s):
                            "DEDICATED" established:0 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:0 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:0 refused:0 current:0 max:1022
                            state:ready
                            DISPATCHER <machine: <Your_ServerName>, pid: 4564>

                            (ADDRESS=(PROTOCOL=tcp)(HOST=<Your_ServerName>)(PORT=55648))
                            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: <Your_ServerName>, pid: 787>

(ADDRESS=(PROTOCOL=tcp)(HOST=<Your_ServerName>)(PORT=48078))
Service "pdb1_1" has 1 instance(s).
  Instance "cdbl", status READY, has 1 handler(s) for this
  service...
    Handler(s):
      "DEDICATED" established:0 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 12.1.0.2.0 -
64bit
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
cdblXDB                             1
cdbl                                 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$pdb;
```

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 "CID"
FROM cdb_pdb;
```

PDB_ID	PDB_NAME	DBID	GUID	CID
3	PDB1_1	671895159	F07E7765AD9421C6E0438D23B98B67C2	3
2	PDB\$SEED	4169567781	F07E43584FB117D3E0438D23B98BA067	2

```
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
-----
/u01/app/oracle/oradata/cdb1/control01.ctl
/u01/app/oracle/fast_recovery_area/cdb1/control02.ctl

CON_ID
-----
      0
      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> set pages 100
SQL> col file_name format A60
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
FROM cdb_data_files ORDER BY con_id ;

FILE_NAME
TABLESPA FILE_ID CON_ID
-----
/u01/app/oracle/oradata/cdb1/system01.dbf
```

```

SYSTEM          1          1
/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
SYS_AUX         3          1
/u01/app/oracle/oradata/pdb1_1/system01.dbf
SYSTEM          8          3
/u01/app/oracle/oradata/pdb1_1/example01.dbf
EXAMPLE         11         3
/u01/app/oracle/oradata/pdb1_1/SAMPLE_SCHEMA_users01.dbf
USERS           10         3
/u01/app/oracle/oradata/pdb1_1/sysaux01.dbf
SYS_AUX         9          3

8 rows selected.

SQL>

```

2) With **ls** UNIX command:

```

SQL> !ls -l /u01/app/oracle/oradata/cdb1
total 2232756

```

```

-rw-rw---- 1 oracle oinstall 17973248 Apr 24 07:26
control01.ctl
drwxrwx--- 2 oracle oinstall      4096 Apr 24 03:26 pdb1_1
drwxrwx--- 2 oracle oinstall      4096 Apr 24 03:16 pdbseed
-rw-rw---- 1 oracle oinstall 52429312 Apr 24 05:02 redo01.log
-rw-rw---- 1 oracle oinstall 52429312 Apr 24 07:26 redo02.log
-rw-rw---- 1 oracle oinstall 52429312 Apr 24 05:01 redo03.log
-rw-rw---- 1 oracle oinstall 880812032 Apr 24 07:25 sysaux01.dbf
-rw-rw---- 1 oracle oinstall 838868992 Apr 24 07:25 system01.dbf
-rw-rw---- 1 oracle oinstall 203431936 Apr 24 07:26 temp01.dbf
-rw-rw---- 1 oracle oinstall 356524032 Apr 24 07:25
undotbs01.dbf
-rw-rw---- 1 oracle oinstall 5251072 Apr 24 05:07 users01.dbf

SQL> !ls -l /u01/app/oracle/oradata/cdb1/pdbseed
total 985064
-rw-rw---- 1 oracle oinstall 102768640 Apr 24 03:23
pdbseed_temp012014-04-24_03-16-22-AM.dbf
-rw-rw---- 1 oracle oinstall 744497152 Apr 24 03:36 sysaux01.dbf
-rw-rw---- 1 oracle oinstall 272637952 Apr 24 03:36 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
       from dba_data_files;
```

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

```
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
       from v$datafile d, v$tablespace ts
       where d.ts#=ts.ts#
       and   d.con_id=ts.con_id
       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 A55
SQL> select FILE_NAME, TABLESPACE_NAME, FILE_ID
       from cdb_temp_files;

FILE_NAME
-----
TABLESPA FILE_ID
-----
/u01/app/oracle/oradata/cdb1/temp01.dbf
TEMP          1

/u01/app/oracle/oradata/pdb1_1/pdb1_1_temp012014-04-24_
03-29-21-AM.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
       where username = 'SYSTEM';

USERNAME                                COM CON_ID
-----
SYSTEM                                YES     1
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
       where common = 'YES';

USERNAME
-----
SYSKM
XS$NULL
DIP
SPATIAL_CSW_ADMIN_USR
OLAPSYS
```



```
SYSTEM
ORDSYS
DBSNMP
ORDPLUGINS
GSMCATUSER
XDB
SYS
DVF
APEX_040200
MDSYS
FLOWS_FILES
GSMUSER
AUDSYS
DVSYS
OJVMSYS
APPQOSSYS
WMSYS
LBACSYS
ANONYMOUS
SI_INFORMTN_SCHEMA
SPATIAL_WFS_ADMIN_USR
SYSBACKUP
CTXSYS
OUTLN
ORACLE_OCM
GSMADMIN_INTERNAL
MDDATA
APEX_PUBLIC_USER
ORDDATA
SYSDG

35 rows selected.

SQL>
```

- c. List all local users of the CDB.

```
SQL> select distinct username, con_id from cdb_users
       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
      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	3
RESOURCE	YES	3
DBA	YES	3
AUDIT_ADMIN	YES	3
AUDIT_VIEWER	YES	3
SELECT_CATALOG_ROLE	YES	3
EXECUTE_CATALOG_ROLE	YES	3
DELETE_CATALOG_ROLE	YES	3
CAPTURE_ADMIN	YES	3
EXP_FULL_DATABASE	YES	3
IMP_FULL_DATABASE	YES	3
CDB_DBA	YES	3
PDB_DBA	YES	3
...		
DV_AUDIT_CLEANUP	YES	1
DV_DATAPUMP_NETWORK_LINK	YES	1
DV_REALM_RESOURCE	YES	1
DV_REALM_OWNER	YES	1
...		

168 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.

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
       from cdb_role_privs
       where grantee='SYSTEM';
```

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

```
SQL> EXIT
$
```


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

Chapter 4

Practices for Lesson 4: Overview

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`.
- Clone 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).

Practice 4-1: Creating a New CDB

Overview

In this practice, you will create a new empty CDB named `cdb2` with DBCA. An empty CDB is a CDB that contains no PDB except the seed.

Assumptions

The CDB `cdb1` has been created during the classroom setup.

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 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 12.1.0.2.0 -
64bit Production
With the Partitioning, OLAP, Advanced Analytics and Real
Application Testing options

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 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 12.1.0.2.0 -
64bit Production
```

With the Partitioning, OLAP, Advanced Analytics and Real Application Testing options

```
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 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 12.1.0.2.0 -
64bit Production
With the Partitioning, OLAP, Advanced Analytics and Real
Application Testing options

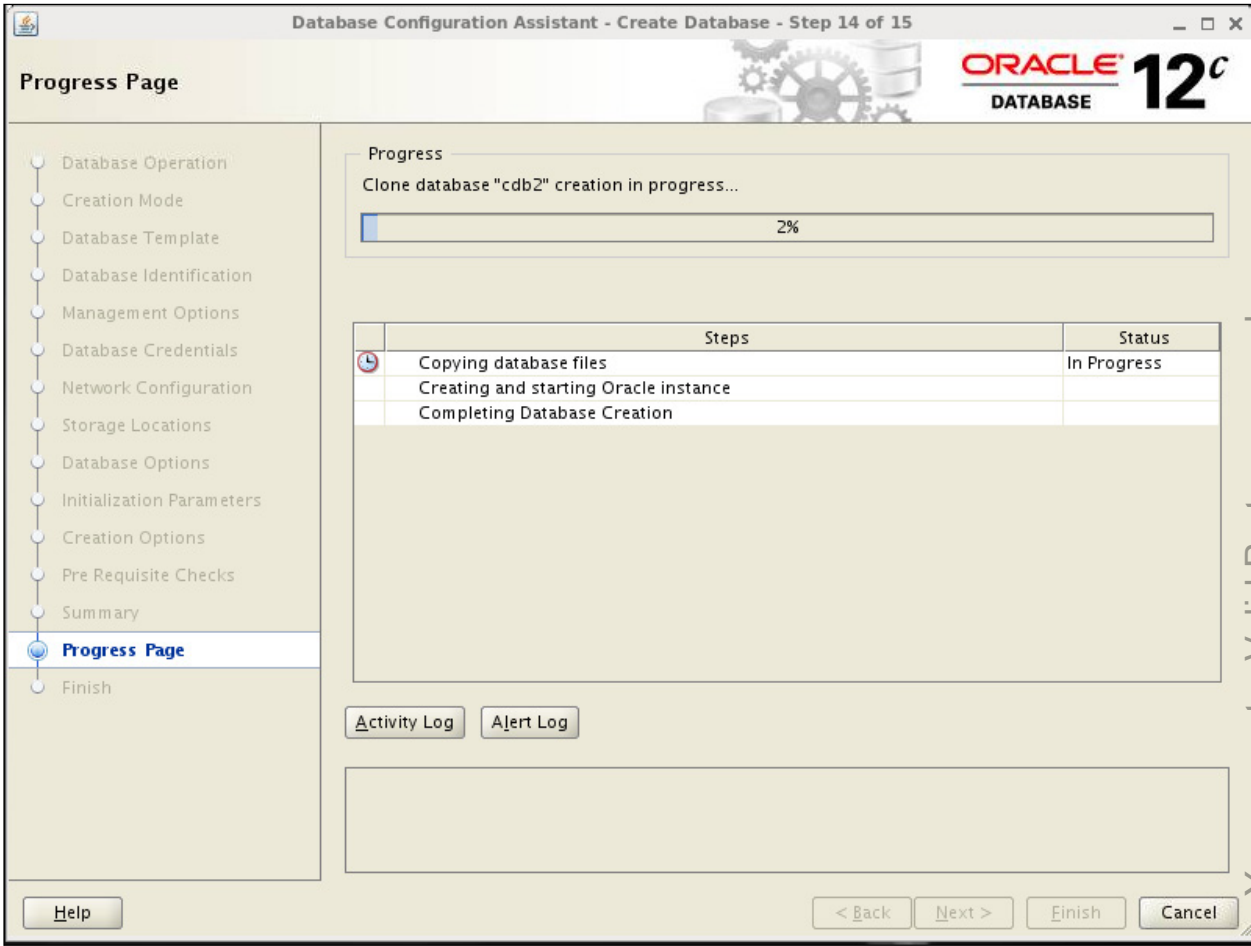
SQL> SHUTDOWN IMMEDIATE
Database closed.
Database dismounted.
ORACLE instance shut down.
SQL> EXIT
$
```

- 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

Step	Window/Page Description	Choices or Values
		Global Database Name: cdb2 SID: cdb2 Select “Create As Container Database” Select “Create An Empty Container Database” Click Next.
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 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	
o.	Step 15: Finish	Click Close.



Practice 4-2: 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 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 12.1.0.2.0 -
64bit Production
With the Partitioning, OLAP, Advanced Analytics and Real
Application Testing options

SQL>
```

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

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

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
...
Listening Endpoints Summary...

(DESCRIPTION=(ADDRESS=(PROTOCOL=tcp) (HOST=yourserver) (PORT=1521)
))
(DESCRIPTION=(ADDRESS=(PROTOCOL=ipc) (KEY=EXTPROC1521)))
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 12.1.0.2.0 -
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$pdb;
```

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> col PDB_ID format 999999
SQL> col PDB_NAME format a8
SQL> col CON_ID format 99
SQL> SELECT PDB_ID, PDB_NAME, DBID, GUID, CON_ID
        from cdb_pdb order by 1;
```

PDB_ID	PDB_NAME	DBID	GUID	CON_ID
2	PDB\$SEED	3646457277	F0A01B77954521B2E0438D23B98B6C48	

```
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 A50
SQL> col tablespace_name format A10
SQL> SELECT FILE_NAME, TABLESPACE_NAME, FILE_ID, con_id
       from   cdb_data_files
       order  by con_id ;

FILE_NAME                                                    TABLESPACE
-----
FILE_ID CON_ID
-----
/u01/app/oracle/oradata/cdb2/system01.dbf                  SYSTEM
      1      1
/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

SQL>
```

- 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
       from   dba_data_files;

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



```

/u01/app/oracle/oradata/cdb2/users01.dbf    USERS    6

SQL> EXIT
$

```

Notice that only root data files are listed.

- e. Start the cdb1 database.

```

$ . oraenv
ORACLE_SID = [cdb2] ? cdb1
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  788529152 bytes
Fixed Size                  2919328 bytes
Variable Size              314573920 bytes
Database Buffers           465567744 bytes
Redo Buffers                5468160 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 12.1.0.2.0 -
64bit Production
With the Partitioning, OLAP, Advanced Analytics and Real
Application Testing options

SQL> ALTER PLUGGABLE DATABASE pdb1_1 OPEN;

Pluggable database altered.

SQL>
```

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

```
SQL> CONNECT system@pdb1_1
Enter password: *****
Connected.
SQL> col file_name format A56
SQL> SELECT FILE_NAME, TABLESPACE_NAME, FILE_ID
       from   dba_data_files;

FILE_NAME
-----
TABLESPACE_NAME          FILE_ID
-----
/u01/app/oracle/oradata/pdb1_1/system01.dbf
SYSTEM                   8
/u01/app/oracle/oradata/pdb1_1/sysaux01.dbf
SYSAUX                   9
/u01/app/oracle/oradata/pdb1_1/SAMPLE_SCHEMA_users01.dbf
USERS                    10
```

```

/u01/app/oracle/oradata/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
       from v$datafile d, v$tablespace ts
       where d.ts#=ts.ts#
       and   d.con_id=ts.con_id
       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/pdb1_1/pdb1_1_temp012014-04-24_03-29-21-
AM.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-rw---- 1 oracle oinstall 7680 Apr 24 03:23 orapwcdb1
-rw-r----- 1 oracle oinstall 7680 Apr 24 08:10 orapwcdb2
-rw-rw---- 1 oracle oinstall 7680 Apr 24 00:31 orapwem12rep
-rw-r----- 1 oracle oinstall 7680 Apr 24 02:47 orapworcl
-rw-rw---- 1 oracle oinstall 7680 Apr 24 03:07 orapworcl2
-rw-rw---- 1 oracle oinstall 3584 Apr 24 08:43 spfilecdb1.ora
-rw-r----- 1 oracle oinstall 3584 Apr 24 08:15 spfilecdb2.ora
-rw-rw---- 1 oracle oinstall 3584 Apr 24 02:35 spfileem12rep.ora
-rw-rw---- 1 oracle oinstall 2560 Apr 24 03:25 spfileorcl2.ora
-rw-rw---- 1 oracle oinstall 2560 Apr 24 07:02 spfileorcl.ora
$
```

- k. Check ADR files, directories, and 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
...
CREATE PLUGGABLE DATABASE PDB$SEED AS CLONE USING
'/u01/app/oracle/product/12.1.0/dbhome_1/assistants/dbca/templat
es//pdbseed.xml' source_file_name_convert =
('/oradata/seedata/pdbseed/temp01.dbf','/u01/app/oracle/oradata
/cdb2/pdbseed/pdbseed_temp012014-04-24_08-03-57-AM.dbf',
'/oradata/seedata/pdbseed/system01.dbf','/u01/app/oracle/oradat
a/cdb2/pdbseed/system01.dbf',
'/oradata/seedata/pdbseed/sysaux01.dbf','/u01/app/oracle/oradat
a/cdb2/pdbseed/sysaux01.dbf') NOCOPY
Thu Apr 24 08:03:58 2014
*****
Pluggable Database PDB$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
*****
Database Characterset for PDB$SEED is US7ASCII
...
*****
Post plug operations are now complete.
Pluggable database PDB$SEED with pdb id - 2 is now marked as
NEW.
```

```

*****
Completed: CREATE PLUGGABLE DATABASE PDB$SEED AS CLONE USING
'/u01/app/oracle/product/12.1.0/dbhome_1/assistants/dbca/templat
es//pdbseed.xml' source_file_name_convert =
('/oradata/seeddata/pdbseed/temp01.dbf','/u01/app/oracle/oradat
a/cdb2/pdbseed/pdbseed_temp012014-04-24_08-03-57-AM.dbf',
'/oradata/seeddata/pdbseed/system01.dbf','/u01/app/oracle/oradat
a/cdb2/pdbseed/system01.dbf',
'/oradata/seeddata/pdbseed/sysaux01.dbf','/u01/app/oracle/oradat
a/cdb2/pdbseed/sysaux01.dbf') NOCOPY
alter pluggable database PDB$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 = [cdb1] ? 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 12.1.0.2.0 -
64bit Production
With the Partitioning, OLAP, Advanced Analytics 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
       where username ='SYSTEM';

USERNAME                                COM  CON_ID
-----
SYSTEM                                YES      1

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
      where common ='YES' order by 1;
```

```
USERNAME
```

```
-----
```

```
ANONYMOUS
```

```
APEX_040200
```

```
APEX_PUBLIC_USER
```

```
APPQOSSYS
```

```
AUDSYS
```

```
CTXSYS
```

```
DBSNMP
```

```
DIP
```

```
DVF
```

```
DVSY
```

```
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
      where common = 'NO';
```

```
no rows selected
```

```
SQL>
```

- e. List local users in root.

```
SQL> select distinct username from dba_users
      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 4-3: Creating a PDB from Seed

Overview

In this practice, you will create a new PDB `pdb2_1` in `cdb2` from `seed`. Use the `CREATE_FILE_DEST` clause to define the directory where the PDB files will be created.

Assumptions

The creation of the CDB `cdb2` is successful and completed during Practice 4-1.

Tasks

Either use DBCA or SQL Developer or SQL commands.

The creation using SQL commands 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 for
ORACLE_HOME=/u01/app/oracle/product/12.1.0/dbhome_1 is
/u01/app/oracle
$ mkdir $ORACLE_BASE/oradata/cdb2/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 12.1.0.1.0 -
64bit Production
With the Partitioning, OLAP, Advanced Analytics and Real
Application Testing options

SQL> CREATE PLUGGABLE DATABASE pdb2_1 ADMIN USER pdb2_1_admin
IDENTIFIED BY oracle_4U ROLES=(CONNECT)
CREATE_FILE_DEST='/u01/app/oracle/oradata/cdb2/pdb2_1';
2      3
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" 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@pdb2_1 AS SYSDBA
```

```
Enter password: *****
```

```
SQL>
```

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

```
SQL> !lsnrctl status
```

```
...
```

```
Listening Endpoints Summary...
```

```
(DESCRIPTION=(ADDRESS=(PROTOCOL=tcp)(HOST=yourserver)(PORT=1521))
```

```
))
```

```
(DESCRIPTION=(ADDRESS=(PROTOCOL=ipc)(KEY=EXTPROC1521)))
```

```
(DESCRIPTION=(ADDRESS=(PROTOCOL=tcp)(HOST=yourserver)(PORT=5502)
```

```
)(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\@localhost:1521\//pdb2_1 AS SYSDBA
Enter password: *****
Connected.
SQL> connect pdb2_1_admin@PDB2_1
Enter password: *****
Connected.

SQL> show con_name

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

7. List the data files created.

```
SQL> !ls -R $ORACLE_BASE/oradata/cdb2/pdb2_1/*
/u01/app/oracle/oradata/cdb2/pdb2_1/CDB2:
F7C6F982A216396EE0438D23B98BD187

/u01/app/oracle/oradata/cdb2/pdb2_1/CDB2/F7C6F982A216396EE0438D2
3B98BD187:
datafile

/u01/app/oracle/oradata/cdb2/pdb2_1/CDB2/F7C6F982A216396EE0438D2
3B98BD187/datafile:
o1_mf_sysaux_9okn7brb_.dbf  o1_mf_temp_9okn7brc_.dbf
o1_mf_system_9okn7br4_.dbf

SQL>
```

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

```
SQL> connect system@pdb2_1
Enter password: *****
Connected.
SQL> col name format A30
SQL> select name from v$services;

NAME
-----
pdb2_1

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
       from   cdb_data_files
       order  by con_id ;

FILE_NAME
-----
TABLESPA FILE_ID CON_ID
-----
/u01/app/oracle/oradata/cdb2/pdb2_1/CDB2/F7C6F982A216396EE0438D2
3B98BD187/datafile/o1_mf_system_9okn7br4_.dbf
SYSTEM          8      3

/u01/app/oracle/oradata/cdb2/pdb2_1/CDB2/F7C6F982A216396EE0438D2
3B98BD187/datafile/o1_mf_sysaux_9okn7brb_.dbf
SYSAux          9      3

SQL> select FILE_NAME, TABLESPACE_NAME, FILE_ID
       from   dba_data_files;

FILE_NAME
-----
TABLESPA FILE_ID
-----
/u01/app/oracle/oradata/cdb2/pdb2_1/CDB2/F7C6F982A216396EE0438D2
3B98BD187/datafile/o1_mf_system_9okn7br4_.dbf
SYSTEM          8

/u01/app/oracle/oradata/cdb2/pdb2_1/CDB2/F7C6F982A216396EE0438D2
3B98BD187/datafile/o1_mf_sysaux_9okn7brb_.dbf
SYSAux          9

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

FILE_NAME
-----
TABLESPA FILE_ID
-----
/u01/app/oracle/oradata/cdb2/pdb2_1/CDB2/F7C6F982A216396EE0438D2
3B98BD187/datafile/o1_mf_temp_9okn7brc_.dbf
TEMP          3

```

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

FILE_NAME
-----
TABLESPA FILE_ID
-----
/u01/app/oracle/oradata/cdb2/pdb2_1/CDB2/F7C6F982A216396EE0438D2
3B98BD187/datafile/o1_mf_temp_9okn7brc_.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
       from   cdb_data_files
       order by con_id, file_id ;

FILE_NAME
```

```
-----
TABLESPACE 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/pdb2_1/CDB2/F7C6F982A216396EE04
38D23B98BD187/datafile/o1_mf_system_9okn7br4_.dbf
```

```
SYSTEM          8          3
```

```
/u01/app/oracle/oradata/cdb2/pdb2_1/CDB2/F7C6F982A216396EE04
38D23B98BD187/datafile/o1_mf_sysaux_9okn7brb_.dbf
```

```
SYSAUX          9          3
```

```
SQL> COL file_name FORMAT A44
```

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

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

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

FILE_NAME	TABLESPACE	FILE_ID
/u01/app/oracle/oradata/cdb2/pdb2_1/CDB2/F7C6F982A216396EE0438D23B98BD187/datafile/o1_mf_temp_9okn7brc_.dbf	TEMP	3

Copyright © 2014, Oracle and/or its affiliates. All rights reserved.

```
/u01/app/oracle/oradata/cdb2/temp01.dbf          TEMP          1

SQL> select FILE_NAME, TABLESPACE_NAME, FILE_ID
       from dba_temp_files;

FILE_NAME                                TABLESPA FILE_ID
-----
/u01/app/oracle/oradata/cdb2/temp01.dbf          TEMP          1

SQL> EXIT
$
```

Practice 4-4: Cloning PDB Within the Same CDB

Overview

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

Assumptions

The pdb2_1 has been successfully created in Practice 4-3.

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 for
ORACLE_HOME=/u01/app/oracle/product/12.1.0/dbhome_1 is
/u01/app/oracle
$ mkdir $ORACLE_BASE/oradata/cdb2/pdb2_2
$
```

2. Run SQL*Plus and connect to the root as a user granted with CREATE PLUGGABLE DATABASE privilege to clone pdb2_2 from pdb2_1.

- 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. Clone pdb2_2 from pdb2_1.

```
SQL> CREATE PLUGGABLE DATABASE pdb2_2 FROM pdb2_1
      CREATE_FILE_DEST='/u01/app/oracle/oradata/cdb2/pdb2_2';
      2
Pluggable database created.

SQL>
```

3. Check the open mode of pdb2_2.

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

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

SQL>
```

4. Reopen PDB2_1 in READ WRITE mode if the PDB is still needed.

```
SQL> alter pluggable database PDB2_1 close;

Pluggable database altered.

SQL> alter pluggable database PDB2_1 open;

Pluggable database altered.

SQL>
```

5. Open PDB2_2 in READ WRITE mode.

```
SQL> alter pluggable database PDB2_2 open;

Pluggable database altered.

SQL> EXIT
$
```

6. Connect to PDB2_2 as SYSDBA.

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

```
$ netca
```

- b. On the Welcome page, select the “Local Net Service Name configuration” and click Next.
- c. On the Net Service Name Configuration page, accept Add and click Next.

- d. On the Net Service Name Configuration, Service Name page, enter `pdb2_2` as Service Name and click Next.
- e. On the Net Service Name Configuration, Select Protocols page, select TCP and click Next.
- f. 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.
- g. On the Net Service Name Configuration, Test page, select "No, do not test" and click Next.
- h. On the Net Service Name Configuration, Net Service Name page, accept `pdb2_2` as Net Service Name and click Next.
- i. On the Net Service Name Configuration, Another Net Service Name page, select No, and Next.
- j. On the Net Service Name Configuration Done page, click Next.
- k. When you are back on the Welcome page, click Finish.

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

```
Enter password: *****
```

```
Connected to:
```

```
Oracle Database 12c Enterprise Edition Release 12.1.0.2.0 -  
64bit Production
```

```
With the Partitioning, OLAP, Advanced Analytics and Real  
Application Testing options
```

```
SQL>
```

7. Check the open mode of the PDBs.

```
SQL> CONNECT / AS SYSDBA
```

```
Connected.
```

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

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

```
SQL>
```

8. Connect to PDB2_2 as the SYSTEM user.

```
SQL> connect system@PDB2_2
Enter password: *****
Connected.
SQL> show con_name
PDB2_2
SQL> EXIT
$
```

9. List the data files created.

```
$ ls -R $ORACLE_BASE/oradata/cdb2/pdb2_2
/u01/app/oracle/oradata/cdb2/pdb2_2:
CDB2

/u01/app/oracle/oradata/cdb2/pdb2_2/CDB2:
F7D54CD877C84F3BE0438D23B98B8665

/u01/app/oracle/oradata/cdb2/pdb2_2/CDB2/F7D54CD877C84F3BE0438D2
3B98B8665:
datafile

/u01/app/oracle/oradata/cdb2/pdb2_2/CDB2/F7D54CD877C84F3BE0438D2
3B98B8665/datafile:
o1_mf_sysaux_9omjb2m3_.dbf  o1_mf_temp_9omjb2m4_.dbf
o1_mf_system_9omjb2lz_.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 12.1.0.2.0 -
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 for
```

```
ORACLE_HOME=/u01/app/oracle/product/12.1.0/dbhome_1 is  
/u01/app/oracle
```

```
$ cd $ORACLE_BASE/oradata/cdb2
```

```
$ mkdir pdb2_2
```

```
$
```

3. Launch SQL Developer.

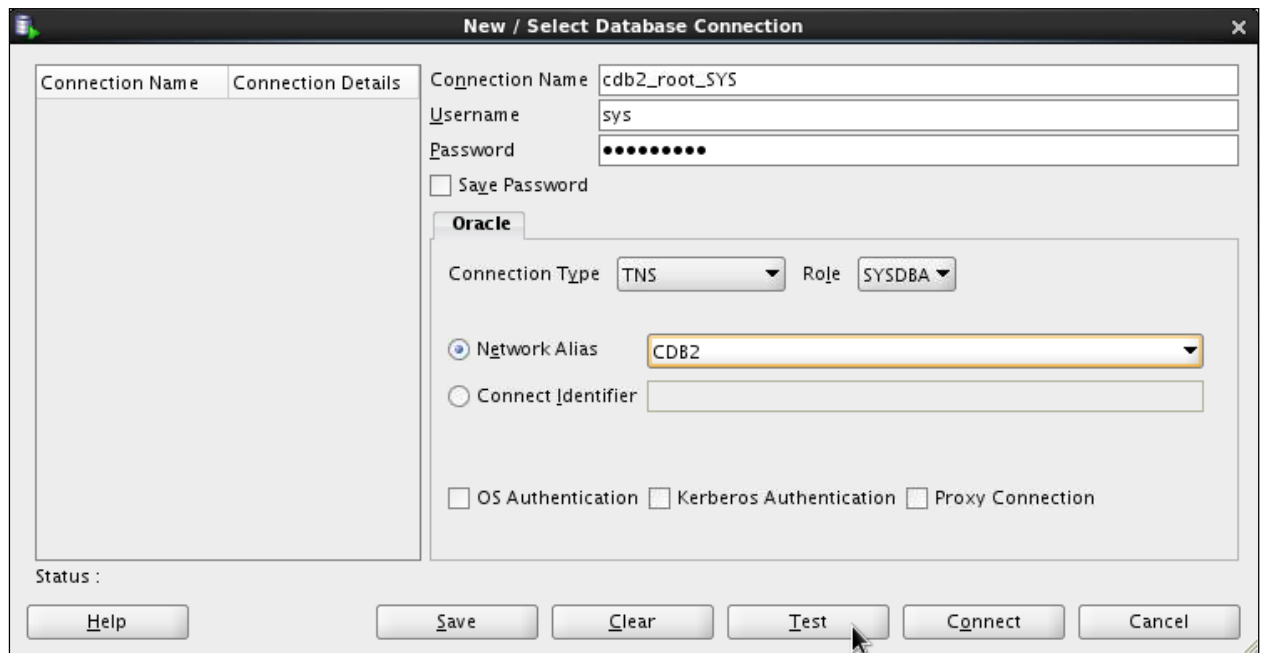
```
$ cd $ORACLE_HOME/sqldeveloper
```

```
$ ./sqldeveloper.sh
```

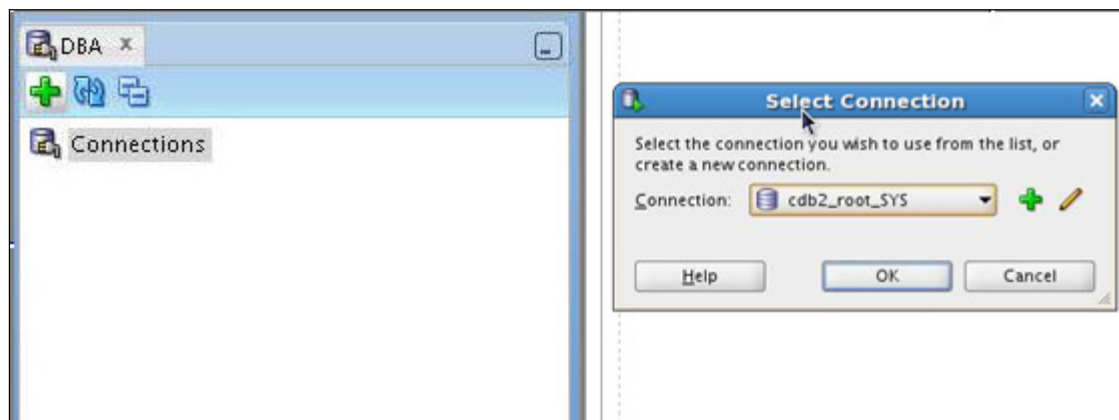
```
$
```

4. Create a connection as SYS in root cdb2.
5. 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

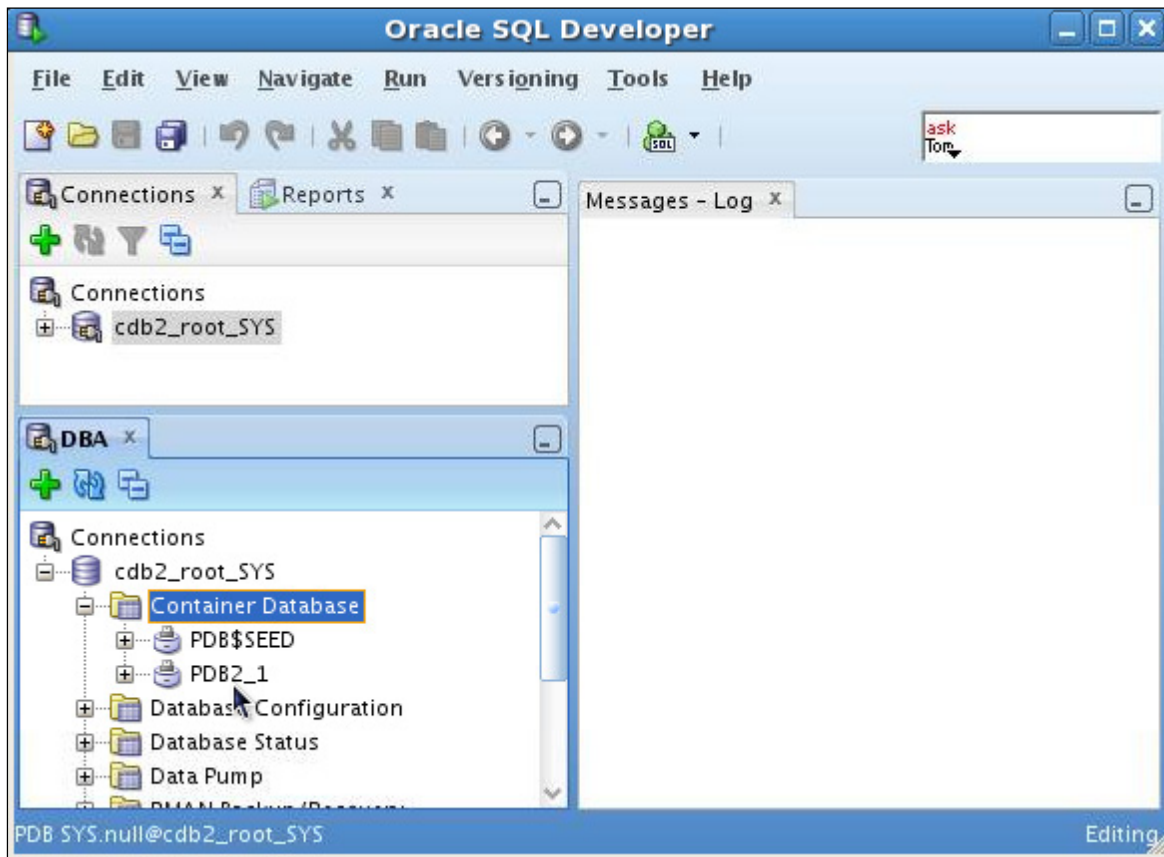


- e. Click Test.
- f. If the status is Success, click Save.
- g. Click Connect.
6. To manage the CDB and its PDBs:
 - a. Choose the View option.
 - b. Click DBA.
 - c. Click + in the left Connections pane to view an existing connection.
 - d. From the list of existing connections, choose the one you just created.



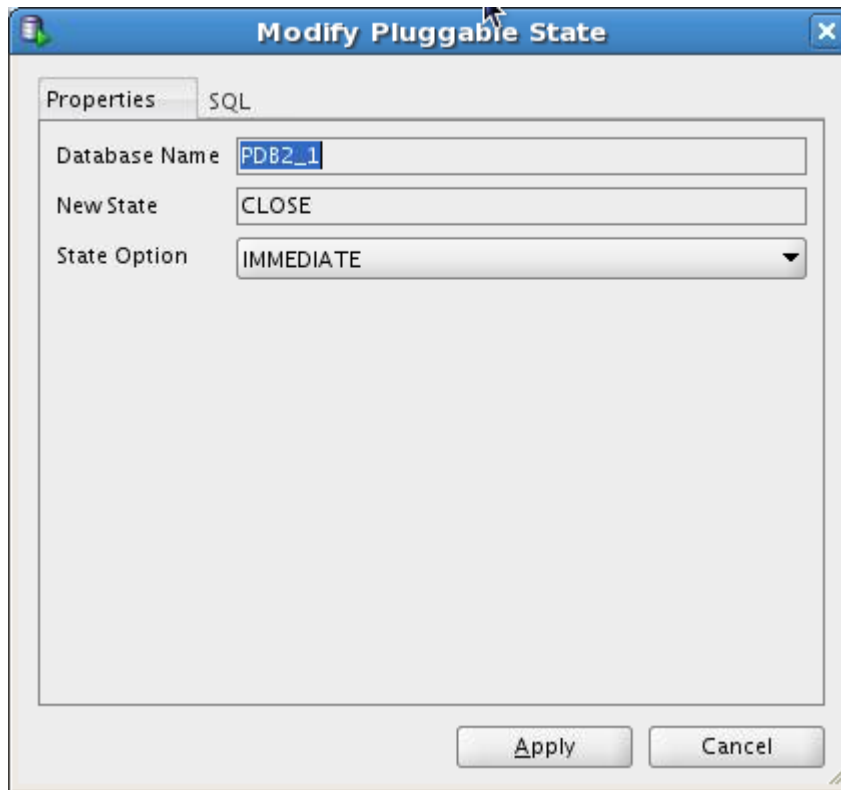
- e. Click OK.

- 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.



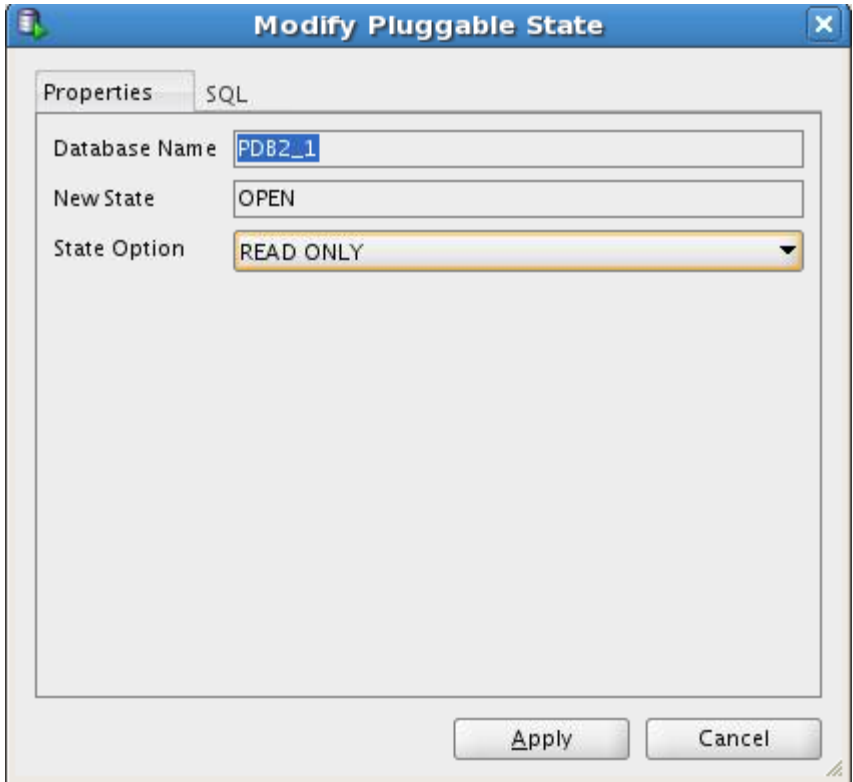
7. Open the source PDB in READ ONLY open mode.
 - a. Right-click pluggable database `PDB2_1` and choose Modify State to set it in READ ONLY open mode before cloning.

- b. First choose Close.



- c. Click Apply, then OK.
- d. Choose Modify State again.

- e. Set the State Option to READ ONLY.



- f. Click Apply then OK.

- 8. 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. Because you will use the CREATE_FILE_DEST clause to set the pdb2_2 files location, you switch to the SQL tab.

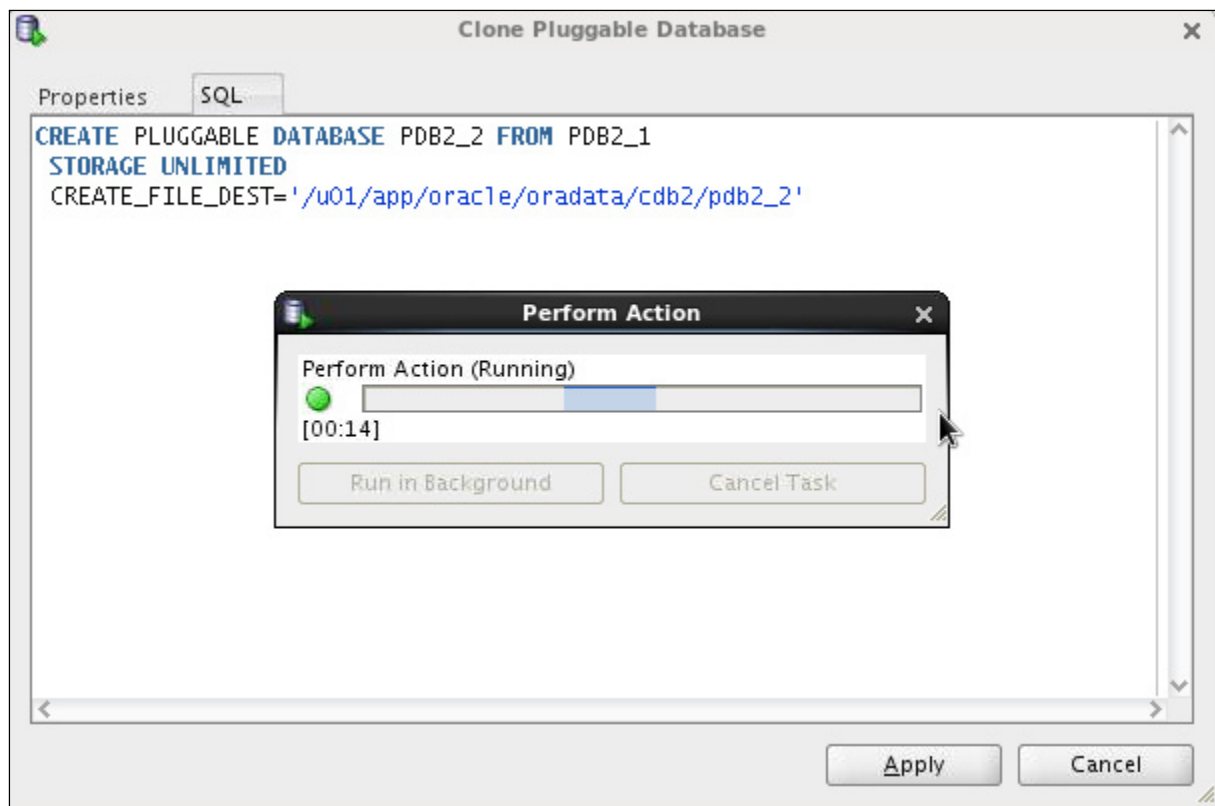
The screenshot shows the 'Clone Pluggable Database' dialog box with the following configuration:

- Database Name:** pdb2_2
- Source PDB:** PDB2_1
- Database Link:** NONE
- Storage:** Unlimited (checked)
- Total Size:** Unlimited (checked) 0 KB
- Temp Tablespace Usage:** Unlimited (checked) 0 KB
- File Name Conversions:** None

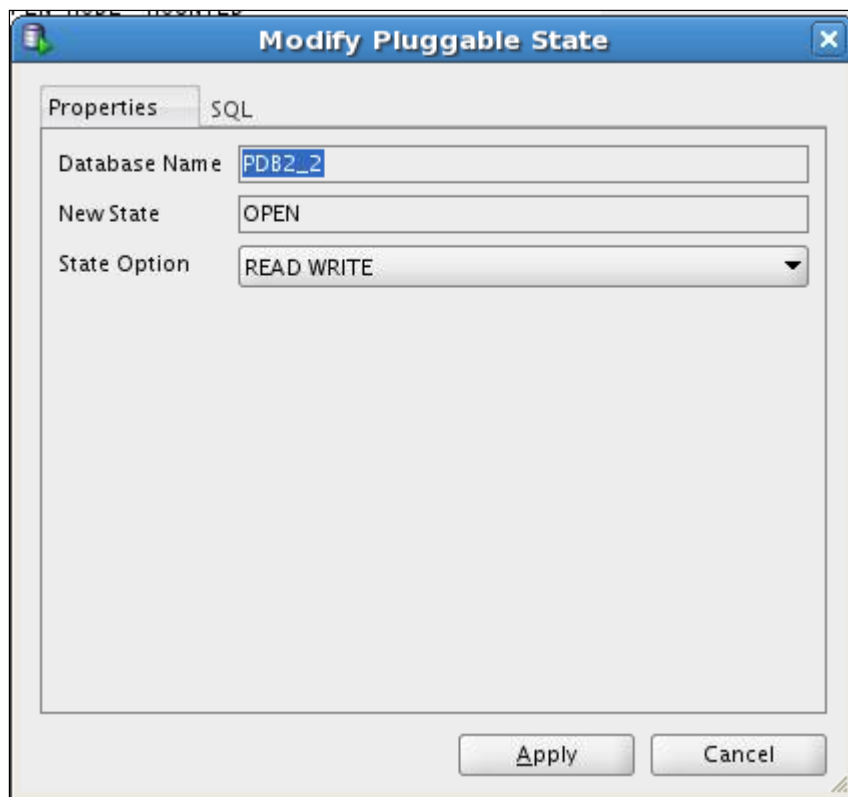
Below the 'File Name Conversions' dropdown, there is a note: "Select None if you have no NAME_CONVERT clause or if you are using OMF."

Buttons at the bottom: Apply, Cancel

- b. Update the `FILE_NAME_CONVERT` clause to `CREATE_FILE_DEST='/u01/app/oracle/oradata/cdb2/pdb2_2'` in the SQL statement.



- 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.



- 2) Click Apply, then OK.
 - e. Open pdb2_1 in READ WRITE mode. Right-click pdb2_1, click Modify State to close it first, click Apply, then OK. Repeat this operation to open it.
9. Leave SQL Developer.
- a. Click File.
 - b. Then click exit.

Practice 4-5: Cloning a Non-CDB into a CDB

Overview

In this practice, you will clone the non-CDB `orcl2` into the CDB `cdb2`.

Tasks

1. Open the non-CDB `orcl2` into READ ONLY mode.

```
$ . oraenv
ORACLE_SID = [cdb2] ? orcl2
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 mount
ORACLE instance started.

Total System Global Area  503316480 bytes
Fixed Size                  2916056 bytes
Variable Size              272630056 bytes
Database Buffers           222298112 bytes
Redo Buffers                5472256 bytes
Database mounted.
SQL> alter database open read only;

Database altered.
SQL> EXIT
$
```

2. Create a new PDB `pdb_orcl2` to clone the non-CDB `orcl2` into `cdb2`. This operation copies the `orcl2` datafiles to datafiles associated to the new `pdb_orcl2` PDB within the target CDB `cdb2`.
 - a. Create a new directory for the datafiles of the new PDB `pdb_orcl2`.

```
$ mkdir /u01/app/oracle/oradata/cdb1/pdb_orcl2
$
```

- b. In the CDB, create the database link that allows a connection to the remote non-CDB as a user with `CREATE PLUGGABLE DATABASE` privilege.

```
$ . 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 12.1.0.2.0 -
64bit Production
With the Partitioning, OLAP, Advanced Analytics and Real
Application Testing options

SQL> CREATE DATABASE LINK link_orcl2
      CONNECT TO system IDENTIFIED BY oracle_4U
      USING 'orcl2';

   2       3
Database link created.

SQL>
```

- c. Clone the source non-CDB into the CDB by using the database link.

```
SQL> create pluggable database PDB_ORCL2 from NON$CDB@link_orcl2
      create_file_dest='/u01/app/oracle/oradata/cdb1/pdb_orcl2';

   2

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. Create the `PDB_ORCL2` net service name.

- 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 to execute the `noncdb_to_pdb.sql` script before opening the PDB.

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

```
Enter password: *****
```

```
Connected to:
```

```
Oracle Database 12c Enterprise Edition Release 12.1.0.2.0 -  
64bit Production
```

```
With the Partitioning, OLAP, Advanced Analytics and Real  
Application Testing options
```

```
SQL>
```

- c. Execute the `noncdb_to_pdb.sql` script. Expect around 15 minutes to complete.

```
SQL> @$ORACLE_HOME/rdbms/admin/noncdb_to_pdb.sql
```

```
...
```

```
SQL> -- Step (III)
```

```
SQL> --
```

```
SQL> -- Invalidate views and synonyms which depend (directly or  
indirectly) on
```

```
SQL> -- invalid objects.
```

```
SQL> begin
```

```
2     loop
```

```
3         update sys.obj$ o_outer set status = 6
```

```
4         where      type# in (4, 5)
```

```
5                 and status not in (5, 6)
```

```
6                 and linkname is null
```

```

7          and ((subname is null) or (subname <>
'DBMS_DBUPGRADE_BABY'))
8          and exists (select o.obj# from sys.obj$ o,
sys.dependency$ d
9                      where      d.d_obj# = o_outer.obj#
10                             and d.p_obj# = o.obj#
11                             and (bitand(d.property, 1) = 1)
12                             and o.status > 1);
13      exit when sql%notfound;
14  end loop;
15 end;
16 /

```

PL/SQL procedure successfully completed.

SQL> commit;

Commit complete.

SQL>

SQL> alter system flush shared_pool;

System altered.

...

SQL> @@utlprp

...

SQL>

SQL> @@utlprp.sql 0

...

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

Session 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> WHENEVER SQLERROR CONTINUE;
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@PDB_ORCL2 as SYSDBA

Enter password: *****
Connected to:
Oracle Database 12c Enterprise Edition Release 12.1.0.2.0 -
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>

```


6. Verify that new datafiles are associated with the new PDB.

```
SQL> select name from v$datafile;

NAME
-----
-----
/u01/app/oracle/oradata/cdb2/undotbs01.dbf
/u01/app/oracle/oradata/cdb1/pdb_orcl2/CDB2/F9CDB58B5B42604FE043
8D23B98B4E40/datafile/o1_mf_system_9qomyrtg_.dbf

/u01/app/oracle/oradata/cdb1/pdb_orcl2/CDB2/F9CDB58B5B42604FE043
8D23B98B4E40/datafile/o1_mf_sysaux_9qomyrth_.dbf

/u01/app/oracle/oradata/cdb1/pdb_orcl2/CDB2/F9CDB58B5B42604FE043
8D23B98B4E40/datafile/o1_mf_users_9qomyrtj_.dbf

/u01/app/oracle/oradata/cdb1/pdb_orcl2/CDB2/F9CDB58B5B42604FE043
8D23B98B4E40/datafile/o1_mf_example_9qomyrtj_.dbf

SQL> EXIT
$
```

7. Because the non-CDB has been successfully cloned, you can now re-open the non-CDB in READ WRITE mode.

```
$ . oraenv
ORACLE_SID = [cdb2] ? orcl2
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 12.1.0.2.0 -
64bit Production
With the Partitioning, OLAP, Advanced Analytics and Real
Application Testing options

SQL> startup force
ORACLE instance started.

Total System Global Area  503316480 bytes
Fixed Size                  2917144 bytes
Variable Size              272633064 bytes
Database Buffers           222298112 bytes
```

```

Redo Buffers          5468160 bytes
Database mounted.
Database opened.
SQL> select name from v$datafile;

```

```

NAME
-----

```

```

/u01/app/oracle/oradata/orcl2/system01.dbf
/u01/app/oracle/oradata/orcl2/sysaux01.dbf
/u01/app/oracle/oradata/orcl2/undotbs01.dbf
/u01/app/oracle/oradata/orcl2/example01.dbf
/u01/app/oracle/oradata/orcl2/users01.dbf

```

```

SQL> EXIT
$

```

8. To release resource, shut down the orcl2 instance.

```

$ . oraenv
ORACLE_SID = [orcl] ? orcl2
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 12.1.0.2.0 -
64bit Production
With the Partitioning, OLAP, Advanced Analytics and Real
Application Testing options

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

```

Practice 4-6: 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 4-1.

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 = [orcl2] ? cdb1
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 12.1.0.2.0 -
64bit Production
With the Partitioning, OLAP, Advanced Analytics and Real
Application Testing options

SQL> select name, open_mode from v$pdb;

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';
```

2

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 or use the \$HOME/labs/CDB/compat.sql script:

```
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;
/
```

```

$ cd $HOME/labs/CDB
$ . oraenv
ORACLE_SID = [cdb1] ? 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 12.1.0.2.0 -
64bit Production
With the Partitioning, OLAP, Advanced Analytics and Real
Application Testing options
SQL> @compat
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
-----
CDB parameter sga_target mismatch: Previous 752M Current 4480M
Please check the parameter in the current CDB

CDB parameter _catalog_foreign_restore mismatch: Previous FALSE
Current TRUE
Please check the parameter in the current CDB

CDB parameter pga_aggregate_target mismatch: Previous 250M
Current 1490M
Please check the parameter in the current 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$pdb;
```

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 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 12.1.0.2.0 -
64bit Production
```

```
With the Partitioning, OLAP, Advanced Analytics and Real
Application Testing options
```

```
SQL> shutdown immediate
Database closed.
Database dismounted.
ORACLE instance shut down.
SQL> startup mount restrict
ORACLE instance started.

Total System Global Area  788529152 bytes
Fixed Size                  2276928 bytes
Variable Size              314573920 bytes
Database Buffers           465567744 bytes
Redo Buffers                5468160 bytes
Database mounted.
SQL> DROP DATABASE;

Database dropped.

SQL> EXIT
$
```

Remove archived logs and backups if necessary.

Practices for Lesson 5: Managing a Multitenant Container Database and Pluggable Databases

Chapter 5

Practices for Lesson 5: Overview

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 4-1.

`pdb2_1` is successfully created in `cdb2` after completion of Practice 4-3.

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

It is not necessary at this step to have successfully created `pdb_orcl2` (Practice 4-5).

Practice 5-1: 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 = [cdb1] ? 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 12.1.0.2.0 -
64bit Production
With the Partitioning, OLAP, Advanced Analytics and Real
Application Testing options

SQL> select name, cdb, con_id from v$databases;
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.

```
$ pgrep -lf 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 4697620480 bytes
Fixed Size                  2923760 bytes
Variable Size               989856528 bytes
Database Buffers            3690987520 bytes
Redo Buffers                 13852672 bytes
Database mounted.
Database opened.

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

SQL> EXIT
$
```

3. Explore the background processes.

```
$ pgrep -lf cdb2
1183 ora_pmon_cdb2
1185 ora_psp0_cdb2
1187 ora_vktm_cdb2
1191 ora_gen0_cdb2
1197 ora_mman_cdb2
1201 ora_diag_cdb2
1203 ora_dbrm_cdb2
1205 ora_vkrm_cdb2
1207 ora_dia0_cdb2
1209 ora_dbw0_cdb2
1211 ora_lgwr_cdb2
1213 ora_ckpt_cdb2
1215 ora_lg00_cdb2
1217 ora_smon_cdb2
1219 ora_lg01_cdb2
1221 ora_reco_cdb2
1223 ora_lreg_cdb2
```

```
1225 ora_pxmn_cdb2
1227 ora_mmon_cdb2
1229 ora_mmn1_cdb2
1231 ora_d000_cdb2
1233 ora_s000_cdb2
1256 ora_tmon_cdb2
1258 ora_tt00_cdb2
1260 ora_smco_cdb2
1262 ora_w000_cdb2
1264 ora_w001_cdb2
1291 ora_aqpc_cdb2
1295 ora_p000_cdb2
1297 ora_p001_cdb2
1299 ora_p002_cdb2
1301 ora_p003_cdb2
1303 ora_p004_cdb2
1305 ora_p005_cdb2
1309 ora_p006_cdb2
1311 ora_p007_cdb2
1419 ora_cjq0_cdb2
1463 ora_qm02_cdb2
1465 ora_qm03_cdb2
1467 ora_q002_cdb2
1469 ora_q003_cdb2
1473 ora_p008_cdb2
1475 ora_p009_cdb2
1477 ora_p00a_cdb2
1479 ora_p00b_cdb2
1483 ora_j000_cdb2
1485 ora_j001_cdb2
1487 ora_j002_cdb2
1489 ora_j003_cdb2
1491 ora_j004_cdb2
1493 ora_j005_cdb2
$
```

4. Explore the PDBs. They are all in MOUNTED state by default.

```
$ sqlplus / as sysdba
```

Connected to:

Oracle Database 12c Enterprise Edition Release 12.1.0.2.0 -
64bit Production

With the Partitioning, OLAP, Advanced Analytics and Real
Application Testing options

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

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$pdb;
```

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@PDB2_1 AS SYSDBA
Enter password: *****
Connected.
SQL> select CON_ID, NAME, OPEN_MODE from v$pdb;
```

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@PDB2_2 AS SYSDBA
Enter password: *****
Connected.
SQL> select CON_ID, NAME, OPEN_MODE from v$pdb;
```

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>
```

Practice 5-2: Closing and Opening a PDB

Overview

In this practice you close PDBs and open PDBs, and save the open state of PDBs to get PDBs automatically opened after CDB startup.

Tasks

1. Connect to the multitenant container database `cdb2` to close `PDB2_1`.
 - a. Start a DML transaction in another session.

```
$ . oraenv
ORACLE_SID = [cdb2] ? cdb2
The Oracle base for
ORACLE_HOME=/u01/app/oracle/product/12.1.0/dbhome_1 is
/u01/app/oracle
$ sqlplus sys@pdb2_1 as sysdba

Enter password: *****

Connected to:
Oracle Database 12c Enterprise Edition Release 12.1.0.2.0 -
64bit Production
With the Partitioning, OLAP, Advanced Analytics 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>
```


- b. In the first session, reconnect to the root as SYSDBA to close PDB2_1 in IMMEDIATE mode.

```
SQL> CONNECT / AS SYSDBA
Connected.
SQL> alter pluggable database pdb2_1 close immediate;
Pluggable database altered.

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

      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@pdb2_1
Enter password: *****
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>
```

2. 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@PDB2_1
Enter password: *****
Connected.
SQL> select * from system.mytab;

          C
-----
          1

SQL>
```

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

- a. Shut down the instance.

```
SQL> CONNECT / AS SYSDBA
Connected.
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  4697620480 bytes
Fixed Size                  2923760 bytes
Variable Size              989856528 bytes
Database Buffers          3690987520 bytes
Redo Buffers               13852672 bytes
SQL> select CON_ID, NAME, OPEN_MODE from v$pdb;

no rows selected.

SQL>
```

- c. Mount cdb2.

```
SQL> alter database mount;
```

Database altered.

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

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>
```

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

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$pdb;

  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>
```

4. Automate the PDBs opening after each CDB startup.

- a. Use the SAVE STATE clause.

```
SQL> alter pluggable database all open;

Pluggable database altered.

SQL> alter pluggable database all SAVE STATE;

Pluggable database altered.

SQL>
```

- b. Test by shutting down the instance and restarting it.

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

Total System Global Area  4697620480 bytes
Fixed Size                  2923760 bytes
Variable Size              989856528 bytes
Database Buffers           3690987520 bytes
Redo Buffers                13852672 bytes
Database mounted.
```

Database opened.

SQL>

- c. Check that the PDBs are all opened except the seed.

SQL> **select CON_ID, NAME, OPEN_MODE from v\$pdb;**

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 5-3: Changing PDBs' Open Mode

Overview

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

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@pdb2_1 as sysdba
Enter password: *****
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$pdb;
```

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$pdb;
```

CON_ID	NAME	OPEN_MODE	RES
3	PDB2	READ WRITE	NO

```
SQL>
```

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

Overview

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

Tasks

- Not all instance parameters are modifiable at the PDB level. A modifiable one, `OPTIMIZER_USE_SQL_PLAN_BASELINES`, has been chosen for the example so as to show how instance parameters behave at PDB and CDB level. Connect to `cdb2`.

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

- 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
$
```

- Connect to `pdb2` in `cdb2` and check the current value of the same instance parameter `OPTIMIZER_USE_SQL_PLAN_BASELINES`.

- Use `netca` to add the `PDB2` net service name for `pdb2` pluggable database of `cdb2` in the `tnsnames.ora` file.

```
$ netca
```

- On the Welcome page, select the “Local Net Service Name configuration” and click Next.
- On the Net Service Name Configuration page, accept Add and click Next.
- On the Net Service Name Configuration, Service Name page, enter `pdb2` as Service Name and click Next.
- On the Net Service Name Configuration, Select Protocols page, select TCP and click Next.
- On the Net Service Name Configuration, TCP/IP Protocol page, enter your complete host name, for example, `<yourservename>.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" 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.

Connect to pdb2 in cdb2.

```
$ sqlplus sys@pdb2 AS SYSDBA
Enter password: *****

Connected to:
Oracle Database 12c Enterprise Edition Release 12.1.0.2.0 -
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@pdb2_2 AS SYSDBA
Enter password: *****
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@pdb2 AS SYSDBA
Enter password: *****
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  4697620480 bytes
Fixed Size                  2923760 bytes
Variable Size              989856528 bytes
Database Buffers           3690987520 bytes
Redo Buffers                13852672 bytes
Database mounted.
Database opened.
SQL> col VALUE format a20
SQL> select CON_ID, VALUE from V$SYSTEM_PARAMETER
       where name ='optimizer_use_sql_plan_baselines';
  2
  CON_ID VALUE
-----
         0 TRUE
         3 FALSE

SQL> EXIT
$
```


Practices for Lesson 6: Managing Tablespaces and Users in a CDB and PDBs

Chapter 6

Practices for Lesson 6: Overview

Practices Overview

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

Assumptions

Practice 4-1 successfully created `cdb2`.

Practice 4-3 successfully created `pdb2_1`.

Practice 5-3 successfully renamed `pdb2_1` to `pdb2`.

Practice 6-1: Managing Tablespaces

Overview

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

Tasks

1. View permanent and temporary tablespace properties in cdb2.

```
$ . oraenv
ORACLE_SID = [cdb2] ? 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 12.1.0.2.0 -
64bit Production
With the Partitioning, OLAP, Advanced Analytics and Real
Application Testing options

SQL> col PROPERTY_NAME format a30
SQL> col PROPERTY_VALUE format a25
SQL> SELECT property_name, property_value
      FROM database_properties
      WHERE property_name LIKE 'DEFAULT_%TABLE%';
  2      3
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                                           3
SYSAUX                                           3
TEMP                                             3
SYSTEM                                           5
SYSAUX                                           5
UNDOTBS1                                         5
TEMP                                             5
USERS                                            5
```

Copyright © 2014, Oracle and/or its affiliates. All rights reserved.

```

EXAMPLE          5
SYSTEM           4
SYSAUX           4
TEMP             4
SYSTEM           6
SYSAUX           6
TEMP             6
USERS            6
EXAMPLE          6
SYSTEM           1
SYSAUX           1
UNDOTBS1         1
TEMP             1
USERS            1

22 rows selected.

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

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

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.

2. Manage permanent tablespaces.
 - a. Create permanent tablespace, `LDATA` in `PDB2`.

```

SQL> connect system@PDB2
Enter password: *****
Connected.
SQL> CREATE TABLESPACE ldata DATAFILE
      '/u01/app/oracle/oradata/cdb2/pdb2_1/ldata_01.dbf'
      SIZE 10M;

  2      3
Tablespace created.

```



```
SQL>
```

- b. Make the **LDATA** tablespace the default tablespace in the **PDB2** container.

```
SQL> ALTER PLUGGABLE DATABASE DEFAULT TABLESPACE LDATA;
```

```
Pluggable database altered.
```

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

```
2      3
PROPERTY_NAME                                PROPERTY_VALUE
-----
DEFAULT_TEMP_TABLESPACE                     TEMP
DEFAULT_PERMANENT_TABLESPACE                 LDATA
```

```
SQL>
```

3. Manage temporary tablespaces (**optional**).

- a. Create a temporary tablespace **TEMP_PDB2** in **PDB2**.

```
SQL> connect system@PDB2
```

```
Enter password: *****
```

```
Connected.
```

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

```
2      3
Tablespace created.
```

```
SQL>
```

- b. 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
       FROM database_properties
       WHERE property_name LIKE 'DEFAULT_%TABLE%';
```

```
2      3
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.

- c. Create a temporary tablespace MY_TEMP in PDB2.

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

```
2      3
```

```
Tablespace created.
```

```
SQL>
```

- d. Display default tablespaces of another PDB in cdb2.

```
SQL> connect system@PDB_ORCL2
```

```
Enter password: *****
```

```
Connected.
```

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

```
2      3
```

```
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
```

```
Enter password: *****
```

```
Connected.
```

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

```
User created.
```

```
SQL>
```

- b. View the default tablespace and temporary tablespace assignment for user C##U 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,
        temporary_tablespace, con_id
        FROM CDB_USERS
        WHERE username = 'C##U';
```

2	3	4	
USERNAME	DEFAULT_TABLESPACE	TEMPORARY_TABLESPACE	CON_ID
-----	-----	-----	-----
C##U	USERS	TEMP	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@PDB2
Enter password: *****
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
        FROM DBA_USERS
        WHERE username = 'LU';
```

2	3	
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
       FROM DBA_USERS
       WHERE username = 'LU';

 2      3
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
Enter password: *****
Connected.
SQL> col NAME format A12
SQL> select FILE#, ts.name, ts.ts#, ts.con_id
       from v$datafile d, v$tablespace ts
       where d.ts#=ts.ts#
       and   d.con_id=ts.con_id
       and   ts.name like 'UNDO%';

 2      3      4      5
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@PDB2
Enter password: *****
Connected.
SQL> CREATE UNDO TABLESPACE UNDO_PDB2 DATAFILE
      '/u01/app/oracle/oradata/cdb2/pdb2/undo_pdb2_01.dbf'
      SIZE 10M;
      2      3
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> !ls /u01/app/oracle/oradata/cdb2/pdb2/undo_pdb2_01.dbf
ls: cannot access
/u01/app/oracle/oradata/cdb2/pdb2/undo_pdb2_01.dbf: No such file
or directory

SQL>
```

Notice that the statement fails because an UNDO tablespace can only be set at CDB level, and moreover there is no UNDO tablespace created.

Practice 6-2: 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 a22
SQL> select USERNAME,COMMON,CON_ID from cdb_users;
```

USERNAME	COM	CON_ID
-----	---	---
ORACLE_OCM	YES	1
OJVMSYS	YES	1
SYSKM	YES	1
XS\$NULL	YES	1
C##U	YES	1
...		
BI	NO	6
PM	NO	6
IX	NO	6
SH	NO	6
OE	NO	6
HR	NO	6
SCOTT	NO	6
ORACLE_OCM	YES	6
OJVMSYS	YES	6
SYSKM	YES	6
XS\$NULL	YES	6
C##U	YES	6
...		
C##U	YES	4
...		

```
198 rows selected.

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

2
USERNAME          COM CON_ID
```

```

-----
SYSTEM          YES          1
SYSTEM          YES          3
SYSTEM          YES          4
SYSTEM          YES          5
SYSTEM          YES          6

5 rows selected.

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

 2
USERNAME
-----
SPATIAL_WFS_ADMIN_USR
ORACLE_OCM
APEX_PUBLIC_USER
ORDDATA
SYSBACKUP
CTXSYS
OUTLN
GSMADMIN_INTERNAL
MDDATA
SYSDG
XS$NULL
DIP
SPATIAL_CSW_ADMIN_USR
SYSTEM
ORDSYS
DVF
MDSYS
GSMUSER
AUDSYS
C##U
APPQOSSYS
WMSYS
ANONYMOUS
SI_INFORMTN_SCHEMA
SYSKM
OLAPSYS
DBSNMP
ORDPLUGINS

```

```
GSMCATUSER
XDB
SYS
APEX_040200
FLOWS_FILES
DVSYS
OJVMSYS
LBACSYS

36 rows selected.

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

USERNAME                                CON_ID
-----
PDB2_1_ADMIN                            4
IX                                       6
SH                                       6
PDBADMIN                                6
BI                                       6
OE                                       6
SCOTT                                    6
HR                                       6
PM                                       6
PDB2_1_ADMIN                            3
LU                                       3
IX                                       5
SH                                       5
BI                                       5
OE                                       5
SCOTT                                    5
HR                                       5
PM                                       5

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
      where username='C##_USER';

  2
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@pdb2
Enter password: *****
Connected.

SQL> connect c##_user@pdb2_2
Enter password: *****
Connected.

SQL> connect c##_user@cdb2
Enter password: *****
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
      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@PDB2 as sysdba
Enter password: *****
Connected.
SQL> col username format a25
SQL> select USERNAME,COMMON,CON_ID from cdb_users;

```

USERNAME	COM	CON_ID
PDB2_1_ADMIN	NO	3
ORACLE_OCM	YES	3
OJVMSYS	YES	3
SYSKM	YES	3
XS\$NULL	YES	3
C##U	YES	3
GSMCATUSER	YES	3
MDDATA	YES	3
SYSBACKUP	YES	3
DIP	YES	3
SYSDG	YES	3
APEX_PUBLIC_USER	YES	3
SPATIAL_CSW_ADMIN_USR	YES	3
SPATIAL_WFS_ADMIN_USR	YES	3
C##_USER	YES	3
GSMUSER	YES	3
AUDSYS	YES	3
FLows_FILES	YES	3
DVF	YES	3
MDSYS	YES	3
ORDSYS	YES	3
DBSNMP	YES	3
WMSYS	YES	3
APEX_040200	YES	3
APPQOSSYS	YES	3
GSMADMIN_INTERNAL	YES	3

Copyright © 2014, Oracle and/or its affiliates. All rights reserved.

ORDDATA	YES	3
CTXSYS	YES	3
ANONYMOUS	YES	3
XDB	YES	3
ORDPLUGINS	YES	3
DVSY	YES	3
SI_INFORMTN_SCHEMA	YES	3
OLAPSYS	YES	3
LBACSYS	YES	3
OUTLN	YES	3
SYSTEM	YES	3
SYS	YES	3
LU	NO	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
      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
      CONTAINER=CURRENT;
2
User created.

SQL> select USERNAME,COMMON,CON_ID from cdb_users
      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
DVSY	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_USR	YES	3
SPATIAL_WFS_ADMIN_USR	YES	3
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.		

Copyright © 2014, Oracle and/or its affiliates. All rights reserved.

```
SQL> grant create session to local_user_pdb2;

Grant succeeded.

SQL>
```

- d. Connect to PDB2 as LOCAL_USER_PDB2.

```
SQL> connect local_user_pdb2@PDB2
Enter password: *****
Connected.
SQL>
```

- e. Connect to PDB2_2 as LOCAL_USER_PDB2.

```
SQL> connect local_user_pdb2@PDB2_2
Enter password: *****
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@cdb2
Enter password: *****
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@PDB2_2 as sysdba
Enter password: *****
Connected.
SQL> col username format a22
SQL> select USERNAME, COMMON, CON_ID from cdb_users
       order by username;
```

USERNAME	COM	CON_ID
-----	----	-----
ANONYMOUS	YES	4
...		
PDB2_1_ADMIN	NO	4
...		

```

SYSTEM                YES                4
...
XS$NULL               YES                4

38 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
...
PDB2_1_ADMIN            NO
...
C##_USER                YES
...
SYSTEM                  YES
...
XS$NULL                 YES

38 rows selected.

SQL>

```

Notice that you view the same list.

Practice 6-3: 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 6-2 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	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	5
DBA	YES	6
...		
PDB_DBA	YES	3
PDB_DBA	YES	4
PDB_DBA	YES	6
...		
XS_SESSION_ADMIN	YES	6

```
420 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
APEX_GRANTS_FOR_NEW_USERS_ROLE     YES
AQ_ADMINISTRATOR_ROLE              YES
AQ_USER_ROLE                        YES
...
XS_RESOURCE                         YES
XS_SESSION_ADMIN                   YES

85 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@PDB2
Enter password: *****
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

86 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
...
C##_ROLE                           YES
...
PDB_DBA                            YES
...
XS_RESOURCE                        YES
XS_SESSION_ADMIN                   YES

86 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
...
LOCAL_ROLE_PDB2                     NO
...
XS_RESOURCE                         YES
XS_SESSION_ADMIN                    YES

87 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
       from cdb_role_privs where grantee='C##_USER';
  2
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
```

```

Enter password: *****
Connected.
SQL> select * from session_roles;

ROLE
-----
C##_ROLE

SQL>

```

```

SQL> connect c##_user@PDB2
Enter password: *****
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
       from cdb_role_privs where grantee='C##_USER';
  2
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

6 rows selected.

SQL>

```

```
SQL> connect c##_user
Enter password: *****
Connected.
SQL> select * from session_roles;

ROLE
-----
C##_ROLE

SQL>
```

```
SQL> connect c##_user@PDB2
Enter password: *****
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
Enter password: *****
Connected.
SQL> select * from session_roles;

ROLE
-----
C##_ROLE

SQL>
```

```
SQL> connect c##_user@PDB2
Enter password: *****
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@PDB2
Enter password: *****
Connected.
SQL> grant c##_role to local_user_PDB2;

Grant succeeded.

SQL> select GRANTEE, GRANTED_ROLE, COMMON, CON_ID
        from cdb_role_privs where grantee='LOCAL_USER_PDB2';
  2
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@PDB2
Enter password: *****
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@PDB2
Enter password: *****
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
       from cdb_role_privs where grantee='LOCAL_USER_PDB2';
  2
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@PDB2
Enter password: *****
Connected.
SQL> select * from session_roles;

ROLE
-----
C##_ROLE
LOCAL_ROLE_PDB2
SQL>

```

Practice 6-4: 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 6-2 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
Enter password: *****
Connected.
SQL> col grantee format a18
SQL> col privilege format a14
SQL> select GRANTEE, PRIVILEGE, COMMON, CON_ID
       from cdb_sys_privs
       where grantee in ('C##_USER', 'LOCAL_USER_PDB2');

  2      3
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@PDB2
Enter password: *****
Connected.
SQL> select GRANTEE, PRIVILEGE, COMMON
       from dba_sys_privs
       where grantee in ('C##_USER', 'LOCAL_USER_PDB2');

  2      3
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
Enter password: *****
Connected.
SQL> grant CREATE TABLE, UNLIMITED TABLESPACE to C##_USER
       CONTAINER=ALL;

  2
Grant succeeded.

SQL> col grantee format a12
SQL> col privilege format a30
SQL> select GRANTEE, PRIVILEGE, COMMON, CON_ID
       from cdb_sys_privs
       where grantee = 'C##_USER';

  2      3

```


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> grant CREATE SEQUENCE to C##_USER CONTAINER=CURRENT;			
Grant succeeded.			
SQL> select GRANTEE, PRIVILEGE, COMMON, CON_ID			
from cdb_sys_privs			
where grantee = 'C##_USER';			
2	3		
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@PDB2			
Enter password: *****			
Connected.			
SQL> grant CREATE SYNONYM to C##_USER CONTAINER=CURRENT;			
Grant succeeded.			
SQL> select GRANTEE, PRIVILEGE, COMMON, CON_ID			
from cdb_sys_privs			
where grantee = 'C##_USER';			
2	3		
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> 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.

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

```
SQL> connect system
Enter password: *****
Connected.
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.

- 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
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.

- 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@PDB2
Enter password: *****
Connected.
SQL> grant UNLIMITED TABLESPACE to LOCAL_USER_PDB2;

Grant succeeded.

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

2	3		
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 7: Backup, Recovery, Flashback CDB and PDBs

Chapter 7

Practices for Lesson 7: Overview

Practices Overview

In the following practices you will perform backup and recovery operations on the CDB and PDBs.

- RMAN `cdb2` backup
- RMAN whole and partial `pdb2` backup
- Recovery from `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 4-1.

`pdb2_1` is successfully created from previous Practice 4-3.

`pdb2_1` is successfully renamed to `pdb2` from previous Practice 5-3.

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

```
$ cd /home/oracle/recovery/catchup_07
$ ./cr_TABLESPACES.sh
$
```

Practice 7-1: 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 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 12.1.0.2.0 -
64bit Production
With the Partitioning, OLAP, Advanced Analytics 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  4697620480 bytes
Fixed Size                  2923760 bytes
```

```

Variable Size                989856528 bytes
Database Buffers            3690987520 bytes
Redo Buffers                 13852672 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  4697620480 bytes
Fixed Size                 2923760 bytes
Variable Size              989856528 bytes
Database Buffers          3690987520 bytes
Redo Buffers               13852672 bytes
Database mounted.
Database opened.
SQL> EXIT
$

```


Practice 7-2: 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 4-3 and 5-4.

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.
 - a. Check the current RMAN configuration.

```
RMAN> SHOW ALL;

using target database control file instead of recovery catalog
RMAN configuration parameters for database with db_unique_name
CDB2 are:
CONFIGURE RETENTION POLICY TO REDUNDANCY 1; # default
CONFIGURE BACKUP OPTIMIZATION OFF; # default
CONFIGURE DEFAULT DEVICE TYPE TO DISK; # default
CONFIGURE CONTROLFILE AUTOBACKUP ON; # default
CONFIGURE CONTROLFILE AUTOBACKUP FORMAT FOR DEVICE TYPE DISK TO
'%F'; # default
CONFIGURE DEVICE TYPE DISK PARALLELISM 1 BACKUP TYPE TO
BACKUPSET; # default
CONFIGURE DATAFILE BACKUP COPIES FOR DEVICE TYPE DISK TO 1; #
default
CONFIGURE ARCHIVELOG BACKUP COPIES FOR DEVICE TYPE DISK TO 1; #
default
CONFIGURE MAXSETSIZE TO UNLIMITED; # default
CONFIGURE ENCRYPTION FOR DATABASE OFF; # default
CONFIGURE ENCRYPTION ALGORITHM 'AES128'; # default
CONFIGURE COMPRESSION ALGORITHM 'BASIC' AS OF RELEASE 'DEFAULT'
OPTIMIZE FOR LOAD TRUE ; # default
CONFIGURE RMAN OUTPUT TO KEEP FOR 7 DAYS; # default
CONFIGURE ARCHIVELOG DELETION POLICY TO NONE; # default
```

Copyright © 2014, Oracle and/or its affiliates. All rights reserved.

```

CONFIGURE SNAPSHOT CONTROLFILE NAME TO
'/u01/app/oracle/product/12.1.0/dbhome_1/dbs/snapcf_cdb2.f'; #
default

RMAN>

```

The server parameter file (SPFILE) and the control files are automatically backed up in a multitenant configuration.

- b. Set the `db_recovery_file_dest_size` to 18 GB.

```

RMAN> ALTER SYSTEM SET db_recovery_file_dest_size=18G
SCOPE=both;

Statement processed

RMAN> BACKUP DATABASE PLUS ARCHIVELOG;

Starting backup at 20-05-2014 09:09:26
current log archived
allocated channel: ORA_DISK_1
channel ORA_DISK_1: SID=271 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=56 RECID=1 STAMP=848048967
channel ORA_DISK_1: starting piece 1 at 20-05-2014 09:09:28
channel ORA_DISK_1: finished piece 1 at 20-05-2014 09:09:29
piece
handle=/u01/app/oracle/fast_recovery_area/CDB2/backupset/2014_05
_20/o1_mf_annnn_TAG20140520T090928_9qp6y8l7_.bkp
tag=TAG20140520T090928 comment=NONE
channel ORA_DISK_1: backup set complete, elapsed time: 00:00:01
Finished backup at 20-05-2014 09:09:29

Starting backup at 20-05-2014 09:09:29
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=00015
name=/u01/app/oracle/oradata/cdb1/pdb_orcl2/CDB2/F9CDB58B5B42604
FE0438D23B98B4E40/datafile/o1_mf_example_9qomyrtj_.dbf
input datafile file number=00013
name=/u01/app/oracle/oradata/cdb1/pdb_orcl2/CDB2/F9CDB58B5B42604
FE0438D23B98B4E40/datafile/o1_mf_sysaux_9qomyrth_.dbf
input datafile file number=00012
name=/u01/app/oracle/oradata/cdb1/pdb_orcl2/CDB2/F9CDB58B5B42604
FE0438D23B98B4E40/datafile/o1_mf_system_9qomyrtg_.dbf

```

```

input datafile file number=00014
name=/u01/app/oracle/oradata/cdb1/pdb_orcl2/CDB2/F9CDB58B5B42604FE0438D23B98B4E40/datafile/o1_mf_users_9qomyrtj_.dbf
channel ORA_DISK_1: starting piece 1 at 20-05-2014 09:09:30
channel ORA_DISK_1: finished piece 1 at 20-05-2014 09:10:45
piece
handle=/u01/app/oracle/fast_recovery_area/CDB2/F9CDB58B5B42604FE0438D23B98B4E40/backupset/2014_05_20/o1_mf_nnndf_TAG20140520T090929_9qp6ybdx_.bkp tag=TAG20140520T090929 comment=NONE
channel ORA_DISK_1: backup set complete, elapsed time: 00:01:15
channel ORA_DISK_1: starting full datafile backup set
channel ORA_DISK_1: specifying datafile(s) in backup set
input datafile file number=00004
name=/u01/app/oracle/oradata/cdb2/undotbs01.dbf
input datafile file number=00003
name=/u01/app/oracle/oradata/cdb2/sysaux01.dbf
input datafile file number=00001
name=/u01/app/oracle/oradata/cdb2/system01.dbf
input datafile file number=00006
name=/u01/app/oracle/oradata/cdb2/users01.dbf
channel ORA_DISK_1: starting piece 1 at 20-05-2014 09:10:45
channel ORA_DISK_1: finished piece 1 at 20-05-2014 09:11:30
piece
handle=/u01/app/oracle/fast_recovery_area/CDB2/backupset/2014_05_20/o1_mf_nnndf_TAG20140520T090929_9qp70p4r_.bkp tag=TAG20140520T090929 comment=NONE
channel ORA_DISK_1: backup set complete, elapsed time: 00:00:45
channel ORA_DISK_1: starting full datafile backup set
channel ORA_DISK_1: specifying datafile(s) in backup set
input datafile file number=00019
name=/u01/app/oracle/oradata/pdb1_1/example01.dbf
input datafile file number=00017
name=/u01/app/oracle/oradata/pdb1_1/sysaux01.dbf
input datafile file number=00016
name=/u01/app/oracle/oradata/pdb1_1/system01.dbf
input datafile file number=00018
name=/u01/app/oracle/oradata/pdb1_1/SAMPLE_SCHEMA_users01.dbf
channel ORA_DISK_1: starting piece 1 at 20-05-2014 09:11:30
channel ORA_DISK_1: finished piece 1 at 20-05-2014 09:12:35
piece
handle=/u01/app/oracle/fast_recovery_area/CDB2/F9CC62B4C0D022FBE0438D23B98BDE28/backupset/2014_05_20/o1_mf_nnndf_TAG20140520T090929_9qp722w1_.bkp tag=TAG20140520T090929 comment=NONE
channel ORA_DISK_1: backup set complete, elapsed time: 00:01:05
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/CDB2/F9CD78E8102A5422E0
438D23B98B55B7/datafile/o1_mf_sysaux_9qolyw3y_.dbf
input datafile file number=00008
name=/u01/app/oracle/oradata/cdb2/pdb2_1/CDB2/F9CD78E8102A5422E0
438D23B98B55B7/datafile/o1_mf_system_9qolyw3r_.dbf
input datafile file number=00020
name=/u01/app/oracle/oradata/cdb2/pdb2_1/ldata_01.dbf
channel ORA_DISK_1: starting piece 1 at 20-05-2014 09:12:35
channel ORA_DISK_1: finished piece 1 at 20-05-2014 09:13:00
piece
handle=/u01/app/oracle/fast_recovery_area/CDB2/F9CD78E8102A5422E0
0438D23B98B55B7/backupset/2014_05_20/o1_mf_nnndf_TAG20140520T090
929_9qp7447g_.bkp tag=TAG20140520T090929 comment=NONE
channel ORA_DISK_1: backup set complete, elapsed time: 00:00:25
channel ORA_DISK_1: starting full datafile backup set
channel ORA_DISK_1: specifying datafile(s) in backup set
input datafile file number=00011
name=/u01/app/oracle/oradata/cdb2/pdb2_2/CDB2/F9CD9FE4AC155B62E0
438D23B98B355C/datafile/o1_mf_sysaux_9qommbgs_.dbf
input datafile file number=00010
name=/u01/app/oracle/oradata/cdb2/pdb2_2/CDB2/F9CD9FE4AC155B62E0
438D23B98B355C/datafile/o1_mf_system_9qommbgr_.dbf
channel ORA_DISK_1: starting piece 1 at 20-05-2014 09:13:01
channel ORA_DISK_1: finished piece 1 at 20-05-2014 09:13:26
piece
handle=/u01/app/oracle/fast_recovery_area/CDB2/F9CD9FE4AC155B62E0
0438D23B98B355C/backupset/2014_05_20/o1_mf_nnndf_TAG20140520T090
929_9qp74xqs_.bkp tag=TAG20140520T090929 comment=NONE
channel ORA_DISK_1: backup set complete, elapsed time: 00:00:25
channel ORA_DISK_1: starting full datafile backup set
channel ORA_DISK_1: specifying datafile(s) in backup set
input datafile file number=00007
name=/u01/app/oracle/oradata/cdb2/pdbseed/sysaux01.dbf
input datafile file number=00005
name=/u01/app/oracle/oradata/cdb2/pdbseed/system01.dbf
channel ORA_DISK_1: starting piece 1 at 20-05-2014 09:13:26
channel ORA_DISK_1: finished piece 1 at 20-05-2014 09:13:51
piece
handle=/u01/app/oracle/fast_recovery_area/CDB2/F9CD1C41E4EC4332E0
0438D23B98B4BBD/backupset/2014_05_20/o1_mf_nnndf_TAG20140520T090
929_9qp75pgo_.bkp tag=TAG20140520T090929 comment=NONE
channel ORA_DISK_1: backup set complete, elapsed time: 00:00:25
Finished backup at 20-05-2014 09:13:51

Starting backup at 20-05-2014 09:13:51

```

```
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=57 RECID=2 STAMP=848049232
channel ORA_DISK_1: starting piece 1 at 20-05-2014 09:13:52
channel ORA_DISK_1: finished piece 1 at 20-05-2014 09:13:53
piece
handle=/u01/app/oracle/fast_recovery_area/CDB2/backupset/2014_05_20/o1_mf_annnn_TAG20140520T091352_9qp76jxy_.bkp
tag=TAG20140520T091352 comment=NONE
channel ORA_DISK_1: backup set complete, elapsed time: 00:00:01
Finished backup at 20-05-2014 09:13:53

Starting Control File and SPFILE Autobackup at 20-05-2014
09:13:54
piece
handle=/u01/app/oracle/fast_recovery_area/CDB2/autobackup/2014_05_20/o1_mf_s_848049234_9qp76ncw_.bkp comment=NONE
Finished Control File and SPFILE Autobackup at 20-05-2014
09:13:57

RMAN>
```

Practice 7-3: 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 4-3 and 5-4.

Tasks

1. Perform a whole PDB backup.

A new RMAN command allows you to back up all data files of the pluggable database.

```
RMAN> BACKUP PLUGGABLE DATABASE pdb2;

Starting backup at 20-05-2014 10:03:06
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/CDB2/F9CD78E8102A5422E0
438D23B98B55B7/datafile/o1_mf_sysaux_9qolyw3y_.dbf
input datafile file number=00008
name=/u01/app/oracle/oradata/cdb2/pdb2_1/CDB2/F9CD78E8102A5422E0
438D23B98B55B7/datafile/o1_mf_system_9qolyw3r_.dbf
input datafile file number=00020
name=/u01/app/oracle/oradata/cdb2/pdb2_1/ldata_01.dbf
channel ORA_DISK_1: starting piece 1 at 20-05-2014 10:03:06
channel ORA_DISK_1: finished piece 1 at 20-05-2014 10:04:41
piece
handle=/u01/app/oracle/fast_recovery_area/CDB2/F9CD78E8102A5422E
0438D23B98B55B7/backupset/2014_05_20/o1_mf_nnndf_TAG20140520T220
306_9qqn8w05_.bkp tag=TAG20140520T220306 comment=NONE
channel ORA_DISK_1: backup set complete, elapsed time: 00:01:35
Finished backup at 20-05-2014 10:04:41

Starting Control File and SPFILE Autobackup at 20-05-2014
10:04:42
piece
handle=/u01/app/oracle/fast_recovery_area/CDB2/autobackup/2014_0
5_20/o1_mf_s_848095482_9qqncvbk_.bkp comment=NONE
Finished Control File and SPFILE Autobackup at 20-05-2014
10:04:45

RMAN>
```

2. Perform a partial PDB backup of the tablespace `ldata`.

A new RMAN command allows you to back up some data files of the pluggable database.

```

RMAN> BACKUP TABLESPACE pdb2:ldata;

Starting backup at 20-05-2014 10:05:02
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=00020
name=/u01/app/oracle/oradata/cdb2/pdb2_1/ldata_01.dbf
channel ORA_DISK_1: starting piece 1 at 20-05-2014 10:05:03
channel ORA_DISK_1: finished piece 1 at 20-05-2014 10:05:04
piece
handle=/u01/app/oracle/fast_recovery_area/CDB2/F9CD78E8102A5422E
0438D23B98B55B7/backupset/2014_05_20/o1_mf_nnndf_TAG20140520T220
502_9qqndhcb_.bkp tag=TAG20140520T220502 comment=NONE
channel ORA_DISK_1: backup set complete, elapsed time: 00:00:01
Finished backup at 20-05-2014 10:05:04

Starting Control File and SPFILE Autobackup at 20-05-2014
10:05:04
piece
handle=/u01/app/oracle/fast_recovery_area/CDB2/autobackup/2014_0
5_20/o1_mf_s_848095504_9qqndkok_.bkp comment=NONE
Finished Control File and SPFILE Autobackup at 20-05-2014
10:05:07

RMAN> EXIT

$
```

Practice 7-4: RMAN Recovery from PDB Data File Loss

Overview

In this practice, you will recover from a PDB data file loss.

Assumptions

The `LDATA` tablespace has been successfully created in Practice 6-1 and backed up in Practice 7-3.

Tasks

1. Remove a data file of the `LDATA` tablespace of `PDB2`.

```
$ sqlplus system@PDB2

Enter password: *****

Connected to:
Oracle Database 12c Enterprise Edition Release 12.1.0.2.0 -
64bit Production
With the Partitioning, OLAP, Advanced Analytics and Real
Application Testing options

SQL> select file_name from dba_data_files
       where tablespace_name='LDATA';
      2
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. Attempt a table creation in the tablespace.

```
$ sqlplus system@PDB2

Enter password: *****

Connected to:
Oracle Database 12c Enterprise Edition Release 12.1.0.2.0 -
64bit Production
With the Partitioning, OLAP, Advanced Analytics and Real
Application Testing options

SQL> CREATE TABLE test (c number) TABLESPACE ldata;
CREATE TABLE test (c number) TABLESPACE ldata
*
ERROR at line 1:
ORA-01116: error in opening database file 22
ORA-01110: data file 22:
'/u01/app/oracle/oradata/cdb2/pdb2_1/ldata_01.dbf'
ORA-27041: unable to open file
Linux-x86_64 Error: 2: No such file or directory
Additional information: 3

SQL>
```

- b. Put the tablespace in OFFLINE mode.

```
SQL> ALTER TABLESPACE ldata OFFLINE IMMEDIATE;

Tablespace altered.

SQL> exit
$
```

- c. Run RMAN to connect to cdb2 with a user with SYSDBA or SYSBACKUP privilege.

```
$ rman target /

connected to target database: CDB2 (DBID=534631279)
RMAN>
```

- d. Restore and recover the tablespace.

```

RMAN> RESTORE TABLESPACE pdb2:LDATA;

Starting restore at 20-05-2014 10:06:30
using target database control file instead of recovery catalog
allocated channel: ORA_DISK_1
channel ORA_DISK_1: SID=30 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 00020 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/F9CD78E8102A5422E0438D23
B98B55B7/backupset/2014_05_20/o1_mf_nnndf_TAG20140520T220502_9qq
ndhcb_.bkp
channel ORA_DISK_1: piece
handle=/u01/app/oracle/fast_recovery_area/CDB2/F9CD78E8102A5422E
0438D23B98B55B7/backupset/2014_05_20/o1_mf_nnndf_TAG20140520T220
502_9qqndhcb_.bkp tag=TAG20140520T220502
channel ORA_DISK_1: restored backup piece 1
channel ORA_DISK_1: restore complete, elapsed time: 00:00:01
Finished restore at 20-05-2014 10:06:33

RMAN> RECOVER TABLESPACE pdb2:LDATA;

Starting recover at 20-05-2014 10:06:50
using channel ORA_DISK_1

starting media recovery
media recovery complete, elapsed time: 00:00:01

Finished recover at 20-05-2014 10:06:53

RMAN> exit
$

```

- e. Put the tablespace back ONLINE.

```

$ sqlplus system@PDB2

Enter password: *****

Connected to:
Oracle Database 12c Enterprise Edition Release 12.1.0.2.0 -
64bit Production

```

With the Partitioning, OLAP, Advanced Analytics and Real Application Testing options

```
SQL> ALTER TABLESPACE ldata ONLINE;
Tablespace altered.
```

```
SQL>
```

- f. Re-attempt a table creation in the tablespace. Then drop the table.

```
SQL> CREATE TABLE test (c number) TABLESPACE ldata;
```

```
Table created.
```

```
SQL> DROP TABLE test;
```

```
Table dropped.
```

```
SQL> EXIT
```

```
$
```

3. If you do not intend to perform any further practices on CDBs, shut down the `cdb1` (if it was not dropped in Practice 4-6) and `cdb2` instances.

```
$ . oraenv
```

```
ORACLE_SID = [cdb2] ? cdb1
```

```
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 12.1.0.2.0 -
64bit Production
```

With the Partitioning, OLAP, Advanced Analytics 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
12.1.0.2.0 - 64bit Production
```

With the Partitioning, OLAP, Advanced Analytics and Real Application Testing options

```
$
```

```
$ . oraenv
ORACLE_SID = [cdb1] ? cdb2
The Oracle base for
ORACLE_HOME=/u01/app/oracle/product/12.1.0/dbhome_1 is
/u01/app/oracle
$ sqlplus / as sysdba

SQL*Plus: Release 12.1.0.2.0 Production on Thu Feb 5 09:15:29
2014

Copyright (c) 1982, 2014, Oracle. All rights reserved.

Connected to:
Oracle Database 12c Enterprise Edition Release 12.1.0.2.0 -
64bit Production
With the Partitioning, OLAP, Advanced Analytics and Real
Application Testing options

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

Practice 7-5: 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 4-3 and 5-4.

Tasks

1. List all data files belonging to PDB2 to be backed up.

```
$ sqlplus system@PDB2

Enter password: *****

Connected to:
Oracle Database 12c Enterprise Edition Release 12.1.0.2.0 -
64bit Production
With the Partitioning, OLAP, Advanced Analytics and Real
Application Testing options

SQL> select file_name from dba_data_files;

FILE_NAME
-----
FILE_NAME
-----
-----
/u01/app/oracle/oradata/cdb2/pdb2_1/CDB2/F9CD78E8102A5422E0438D2
3B98B55B7/datafile/o1_mf_system_9qolyw3r_.dbf

/u01/app/oracle/oradata/cdb2/pdb2_1/CDB2/F9CD78E8102A5422E0438D2
3B98B55B7/datafile/o1_mf_sysaux_9qolyw3y_.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 -r /u01/app/oracle/oradata/cdb2/pdb2_1 /home/oracle/backup
$
```

4. Deactivate the backup mode.

```
$ sqlplus system@PDB2

Enter password: *****

Connected to:
Oracle Database 12c Enterprise Edition Release 12.1.0.2.0 -
64bit Production
With the Partitioning, OLAP, Advanced Analytics and Real
Application Testing options

SQL> ALTER PLUGGABLE DATABASE pdb2 END BACKUP;

Pluggable database altered.

SQL>
```

Practice 7-6: 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.
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>
```

If you read the trace file located in `/u01/app/oracle/diag/rdbms/cdb2/cdb2/trace` directory, you will see all root and PDBs data files, and the CDB redo log files referred in the `CREATE CONTROL FILE` command.

```
CREATE CONTROLFILE REUSE DATABASE "CDB2" NORESETLOGS  ARCHIVELOG
    MAXLOGFILES 16
    MAXLOGMEMBERS 3
    MAXDATAFILES 1024
    MAXINSTANCES 8
    MAXLOGHISTORY 292
LOGFILE
  GROUP 1 '/u01/app/oracle/oradata/cdb2/redo01.log'  SIZE 50M
BLOCKSIZE 512,
  GROUP 2 '/u01/app/oracle/oradata/cdb2/redo02.log'  SIZE 50M
BLOCKSIZE 512,
  GROUP 3 '/u01/app/oracle/oradata/cdb2/redo03.log'  SIZE 50M
BLOCKSIZE 512
-- STANDBY LOGFILE
DATAFILE
  '/u01/app/oracle/oradata/cdb2/system01.dbf',
  '/u01/app/oracle/oradata/cdb2/sysaux01.dbf',
  '/u01/app/oracle/oradata/cdb2/undotbs01.dbf',
  '/u01/app/oracle/oradata/cdb2/pdbseed/system01.dbf',
  '/u01/app/oracle/oradata/cdb2/users01.dbf',
  '/u01/app/oracle/oradata/cdb2/pdbseed/sysaux01.dbf',
```

```

'/u01/app/oracle/oradata/cdb2/pdb2_1/CDB2/F9CD78E8102A5422E0438D
23B98B55B7/datafile/o1_mf_system_9qolyw3r_.dbf',

'/u01/app/oracle/oradata/cdb2/pdb2_1/CDB2/F9CD78E8102A5422E0438D
23B98B55B7/datafile/o1_mf_sysaux_9qolyw3y_.dbf',

'/u01/app/oracle/oradata/cdb2/pdb2_2/CDB2/F9CD9FE4AC155B62E0438D
23B98B355C/datafile/o1_mf_system_9qommbgr_.dbf',

'/u01/app/oracle/oradata/cdb2/pdb2_2/CDB2/F9CD9FE4AC155B62E0438D
23B98B355C/datafile/o1_mf_sysaux_9qommbgs_.dbf',

'/u01/app/oracle/oradata/cdb1/pdb_orcl2/CDB2/F9CDB58B5B42604FE04
38D23B98B4E40/datafile/o1_mf_system_9qomyrtg_.dbf',

'/u01/app/oracle/oradata/cdb1/pdb_orcl2/CDB2/F9CDB58B5B42604FE04
38D23B98B4E40/datafile/o1_mf_sysaux_9qomyrth_.dbf',

'/u01/app/oracle/oradata/cdb1/pdb_orcl2/CDB2/F9CDB58B5B42604FE04
38D23B98B4E40/datafile/o1_mf_users_9qomyrtj_.dbf',

'/u01/app/oracle/oradata/cdb1/pdb_orcl2/CDB2/F9CDB58B5B42604FE04
38D23B98B4E40/datafile/o1_mf_example_9qomyrtj_.dbf',
  '/u01/app/oracle/oradata/pdb1_1/system01.dbf',
  '/u01/app/oracle/oradata/pdb1_1/sysaux01.dbf',
  '/u01/app/oracle/oradata/pdb1_1/SAMPLE_SCHEMA_users01.dbf',
  '/u01/app/oracle/oradata/pdb1_1/example01.dbf',
  '/u01/app/oracle/oradata/cdb2/pdb2_1/ldata_01.dbf'
CHARACTER SET AL32UTF8
;

```


Practice 7-7: RMAN Recovery from Control File Loss (*Optional*)

Overview

In this practice, you will recover the CDB from the control file loss.

Assumptions

Practice 7-2 successfully completed the whole CDB backup of `cdb2`.

Tasks

1. Remove the control files of the CDB.

```
SQL> CONNECT / AS SYSDBA
Connected.
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> !rm /u01/app/oracle/fast_recovery_area/cdb2/control02.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      4697620480 bytes

Fixed Size                     2923760 bytes
Variable Size                  989856528 bytes
Database Buffers               3690987520 bytes
Redo Buffers                    13852672 bytes
```

```

RMAN> RESTORE CONTROLFILE FROM AUTOBACKUP;

```

```

Starting restore at 21-05-2014 12:42:41
using target database control file instead of recovery catalog
allocated channel: ORA_DISK_1
channel ORA_DISK_1: SID=12 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/2014_05_20/o1
_mf_s_848096121_9qqnztot_.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/2014_05_20/o1
_mf_s_848096121_9qqnztot_.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 21-05-2014 12:42:43

```

```

RMAN> ALTER DATABASE MOUNT;

```

```

Statement processed
released channel: ORA_DISK_1

```

```

RMAN> RECOVER DATABASE;

```

```

Starting recover at 21-05-2014 12:43:19
Starting implicit crosscheck backup at 21-05-2014 12:43:19
allocated channel: ORA_DISK_1
channel ORA_DISK_1: SID=12 device type=DISK
Crosschecked 13 objects
Finished implicit crosscheck backup at 21-05-2014 12:43:20

```

```

Starting implicit crosscheck copy at 21-05-2014 12:43:20
using channel ORA_DISK_1
Finished implicit crosscheck copy at 21-05-2014 12:43:20

```

```

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/2014_05_20/o1
_mf_s_848096121_9qqnztot_.bkp

using channel ORA_DISK_1

starting media recovery

archived log for thread 1 with sequence 64 is already on disk as
file /u01/app/oracle/oradata/cdb2/redo01.log
archived log file name=/u01/app/oracle/oradata/cdb2/redo01.log
thread=1 sequence=64
media recovery complete, elapsed time: 00:00:03
Finished recover at 21-05-2014 12:43:25

RMAN> ALTER DATABASE OPEN RESETLOGS;

Statement processed

RMAN> select name, open_mode from v$pdb;

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;

Starting backup at 21-05-2014 12:45:09
current log archived
...
Finished backup at 21-05-2014 12:50:02

Starting Control File and SPFILE Autobackup at 21-05-2014
12:50:02
piece
handle=/u01/app/oracle/fast_recovery_area/CDB2/autobackup/2014_0
5_21/o1_mf_s_848105402_9qqylvl1_.bkp comment=NONE
Finished Control File and SPFILE Autobackup at 21-05-2014
12:50:05

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
21/05/2014 12:45:44
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
Archive Log	1	20-05-2014 09:09:27	/u01/app/oracle/fast_recovery_area/CDB2/archivelog/2014_05_20/o1_mf_1_56_9qp6y73f_.arc
Backup Set	1	20-05-2014 09:09:29	

```
Backup Piece      1      20-05-2014 09:09:29
/u01/app/oracle/fast_recovery_area/CDB2/backupset/2014_05_20/o1_
mf_annnn_TAG20140520T090928_9qp6y817_.bkp

...

Do you really want to delete the above objects (enter YES or
NO)?

...

backup piece
handle=/u01/app/oracle/fast_recovery_area/CDB2/backupset/2014_05
_21/o1_mf_annnn_TAG20140521T004510_9qgxrxjd_.bkp RECID=17
STAMP=848105117
Deleted 29 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 7-8: 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

Enter password: *****

Connected to:
Oracle Database 12c Enterprise Edition Release 12.1.0.2.0 -
64bit Production
With the Partitioning, OLAP, Advanced Analytics 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>
```

2. Remove a redo log file member of the cdb2.

```
SQL> !rm /u01/app/oracle/oradata/cdb2/redo01.log
$
```

3. In the alert log file, read the warning message related to the absence of the redo log file member.
 - a. Switch the current redo log file as many times as necessary until the group 1 becomes the current one.

```
SQL> SELECT group#, status FROM v$log;

  GROUP# STATUS
-----
       1 INACTIVE
       2 INACTIVE
       3 CURRENT

SQL> alter system switch logfile;

System altered.

SQL> SELECT group#, status FROM v$log;

  GROUP# STATUS
-----
```

```

1 CURRENT
2 INACTIVE
3 ACTIVE

```

```

SQL> !
$

```

- b. Open the alert log file. At the end of the trace file, you should find a warning message like the one below. Then quit the text editor and return to the SQL prompt.

```

$ cd /u01/app/oracle/diag/rdbms/cdb2/cdb2/trace/
$ vi a*
Wed May 21 00:55:19 2014
Errors in file Errors in file
/u01/app/oracle/diag/rdbms/cdb2/cdb2/trace/cdb2_lg00_12140.trc:
ORA-00313: open failed for members of log group 1 of thread 1
ORA-00312: online log 1 thread 1:
'/u01/app/oracle/oradata/cdb2/redo01.log'
ORA-27037: unable to obtain file status
Linux-x86_64 Error: 2: No such file or directory
Additional information: 3
:q

exit
SQL>

```

4. 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).

```

SQL> ALTER DATABASE DROP LOGFILE MEMBER
      '/u01/app/oracle/oradata/cdb2/redo01.log';
2
Database altered.

SQL> ALTER DATABASE ADD LOGFILE MEMBER
      '/u01/app/oracle/oradata/cdb2/redo01.log' TO GROUP 1;
2
Database altered.

SQL> EXIT
$

```


Practice 7-9: 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. In the following practices, if you do not want the CDB startup to spend too much time on the automatic PDBs reopening, discard the open state for all PDBs.

```
$ sqlplus system

Enter password: *****

Connected to:
Oracle Database 12c Enterprise Edition Release 12.1.0.2.0 -
64bit Production
With the Partitioning, OLAP, Advanced Analytics and Real
Application Testing options

SQL> ALTER PLUGGABLE DATABASE ALL DISCARD STATE;

Pluggable database altered.

SQL>
```

2. Remove the SYSTEM data file from the root SYSTEM tablespace.

```
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
$
```

- Run RMAN to connect to `cdb2` with a user with `SYSDBA` or `SYSBACKUP` privilege. In case the error message does not appear when you connect to RMAN, execute a simple query that accesses the `SYSTEM` tablespace such as `SELECT count(*) FROM dba_users;`

```
$ 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 3
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
$
```

- 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 12.1.0.2.0 -
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 4697620480 bytes
Fixed Size                2923760 bytes
Variable Size             989856528 bytes
Database Buffers          3690987520 bytes
Redo Buffers               13852672 bytes
Database mounted.
SQL> EXIT
$
```

```

$ rman target /
connected to target database: CDB2 (DBID=562519177, not open)
RMAN> RESTORE TABLESPACE SYSTEM;

Starting restore at 21-05-2014 12:58:41
using target database control file instead of recovery catalog
allocated channel: ORA_DISK_1
channel ORA_DISK_1: SID=12 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/2014_05_21/o1_
mf_nnndf_TAG20140521T004518_9qgxvc3p_.bkp
channel ORA_DISK_1: piece
handle=/u01/app/oracle/fast_recovery_area/CDB2/backupset/2014_05
_21/o1_mf_nnndf_TAG20140521T004518_9qgxvc3p_.bkp
tag=TAG20140521T004518
channel ORA_DISK_1: restored backup piece 1
channel ORA_DISK_1: restore complete, elapsed time: 00:00:35
Finished restore at 21-05-2014 12:59:17

RMAN> RECOVER TABLESPACE SYSTEM;

Starting recover at 21-05-2014 12:59:24
using channel ORA_DISK_1

starting media recovery

archived log for thread 1 with sequence 2 is already on disk as
file
/u01/app/oracle/fast_recovery_area/CDB2/archivelog/2014_05_21/o1_
mf_1_2_9qqylro0_.arc
archived log for thread 1 with sequence 3 is already on disk as
file
/u01/app/oracle/fast_recovery_area/CDB2/archivelog/2014_05_21/o1_
mf_1_3_9qqy7b1g_.arc
archived log for thread 1 with sequence 4 is already on disk as
file
/u01/app/oracle/fast_recovery_area/CDB2/archivelog/2014_05_21/o1_
mf_1_4_9qqy7hbm_.arc
archived log for thread 1 with sequence 5 is already on disk as
file

```

```

/u01/app/oracle/fast_recovery_area/CDB2/archivelog/2014_05_21/o1
_mf_1_5_9qqy7os9_.arc
archived log for thread 1 with sequence 6 is already on disk as
file
/u01/app/oracle/fast_recovery_area/CDB2/archivelog/2014_05_21/o1
_mf_1_6_9qqy7rsp_.arc
archived log for thread 1 with sequence 7 is already on disk as
file
/u01/app/oracle/fast_recovery_area/CDB2/archivelog/2014_05_21/o1
_mf_1_7_9qqyc59z_.arc
archived log for thread 1 with sequence 8 is already on disk as
file
/u01/app/oracle/fast_recovery_area/CDB2/archivelog/2014_05_21/o1
_mf_1_8_9qqycfwo_.arc
archived log for thread 1 with sequence 9 is already on disk as
file
/u01/app/oracle/fast_recovery_area/CDB2/archivelog/2014_05_21/o1
_mf_1_9_9qqycq4y_.arc
archived log for thread 1 with sequence 10 is already on disk as
file
/u01/app/oracle/fast_recovery_area/CDB2/archivelog/2014_05_21/o1
_mf_1_10_9qqyg8q4_.arc
archived log file
name=/u01/app/oracle/fast_recovery_area/CDB2/archivelog/2014_05_
21/o1_mf_1_2_9qqylro0_.arc thread=1 sequence=2
archived log file
name=/u01/app/oracle/fast_recovery_area/CDB2/archivelog/2014_05_
21/o1_mf_1_3_9qqy7blg_.arc thread=1 sequence=3
archived log file
name=/u01/app/oracle/fast_recovery_area/CDB2/archivelog/2014_05_
21/o1_mf_1_4_9qqy7hbm_.arc thread=1 sequence=4
archived log file
name=/u01/app/oracle/fast_recovery_area/CDB2/archivelog/2014_05_
21/o1_mf_1_5_9qqy7os9_.arc thread=1 sequence=5
archived log file
name=/u01/app/oracle/fast_recovery_area/CDB2/archivelog/2014_05_
21/o1_mf_1_6_9qqy7rsp_.arc thread=1 sequence=6
archived log file
name=/u01/app/oracle/fast_recovery_area/CDB2/archivelog/2014_05_
21/o1_mf_1_7_9qqyc59z_.arc thread=1 sequence=7
archived log file
name=/u01/app/oracle/fast_recovery_area/CDB2/archivelog/2014_05_
21/o1_mf_1_8_9qqycfwo_.arc thread=1 sequence=8
media recovery complete, elapsed time: 00:00:03
Finished recover at 21-05-2014 12:59:28

RMAN> ALTER DATABASE OPEN;

```

```
Statement processed
```

```
RMAN>
```

5. Back up the CDB.

```
RMAN> BACKUP DATABASE PLUS ARCHIVELOG DELETE INPUT;
```

```
...
```

```
RMAN> EXIT
```

```
$
```

Practice 7-10: 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 12.1.0.2.0 -
64bit Production
With the Partitioning, OLAP, Advanced Analytics 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 Data Recovery Advisor commands to discover, restore and recover the failure. If the failure does not appear immediately when executing the `LIST FAILURE` statement, reiterate the statement a few seconds later.

- a. Discover the failure.

```
RMAN> LIST FAILURE;

using target database control file instead of recovery catalog
Database Role: PRIMARY
```

```

List of Database Failures
=====

Failure ID Priority Status      Time Detected      Summary
-----
262          HIGH      OPEN      20-05-2014 10:06:04 One or more
non-system datafiles are missing

RMAN>

```

- b. If you want more details about the failure, use the `DETAIL` clause in the same command.

```

RMAN> LIST FAILURE DETAIL;

Database Role: PRIMARY

List of Database Failures
=====

Failure ID Priority Status      Time Detected      Summary
-----
262          HIGH      OPEN      20-05-2014 10:06:04 One or more
non-system datafiles are missing
  Impact: See impact for individual child failures
  List of child failures for parent failure ID 262
  Failure ID Priority Status      Time Detected      Summary
  -----
    885          HIGH      OPEN      21-05-2014 01:07:06 Datafile 3:
'/u01/app/oracle/oradata/cdb2/sysaux01.dbf' is missing
    Impact: Some objects in tablespace SYSAUX might be
    unavailable

RMAN>

```

- c. Get advice from RMAN Data Recovery Advisor.

```

RMAN> ADVISE FAILURE;

Database Role: PRIMARY

List of Database Failures
=====

Failure ID Priority Status      Time Detected      Summary
-----

```

```

262          HIGH      OPEN      20-05-2014 10:06:04 One or more
non-system datafiles are missing
  Impact: See impact for individual child failures
  List of child failures for parent failure ID 262
  Failure ID Priority Status      Time Detected      Summary
  -----
  885          HIGH      OPEN      21-05-2014 01:07:06 Datafile 3:
'/u01/app/oracle/oradata/cdb2/sysaux01.dbf' is missing
  Impact: Some objects in tablespace SYSAUX might be
unavailable

analyzing automatic repair options; this may take some time
allocated channel: ORA_DISK_1
channel ORA_DISK_1: SID=32 device type=DISK
analyzing automatic repair options complete

Mandatory Manual Actions
=====
no manual actions available

Optional Manual Actions
=====
1. If file /u01/app/oracle/oradata/cdb2/sysaux01.dbf was
unintentionally renamed or moved, restore it

Automated Repair Options
=====
Option Repair Description
-----
1          Restore and recover datafile 3
  Strategy: The repair includes complete media recovery with no
data loss
  Repair script:
/u01/app/oracle/diag/rdbms/cdb2/cdb2/hm/reco_46185046.hm

RMAN>

```

- d. Preview the provided script to repair the failure.

```

RMAN> REPAIR FAILURE PREVIEW;

Strategy: The repair includes complete media recovery with no
data loss
Repair script:
/u01/app/oracle/diag/rdbms/cdb2/cdb2/hm/reco_46185046.hm

```



```

contents of repair script:
  # restore and recover datafile
  sql 'alter database datafile 3 offline';
  restore ( datafile 3 );
  recover datafile 3;
  sql 'alter database datafile 3 online';

```

```

RMAN>

```

- e. If the provided script satisfies you, repair the failure. This will execute the script.

```

RMAN> REPAIR FAILURE;

```

Strategy: The repair includes complete media recovery with no data loss

Repair script:

```

/u01/app/oracle/diag/rdbms/cdb2/cdb2/hm/reco_46185046.hm

```

contents of repair script:

```

  # restore and recover datafile
  sql 'alter database datafile 3 offline';
  restore ( datafile 3 );
  recover datafile 3;
  sql 'alter database datafile 3 online';

```

Do you really want to execute the above repair (enter YES or NO)? **YES**

executing repair script

sql statement: alter database datafile 3 offline

Starting restore at 21-05-2014 01:09:35

using channel ORA_DISK_1

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/2014_05_21/o1_
mf_nnndf_TAG20140521T010026_9qqyrcyz_.bkp

channel ORA_DISK_1: piece

handle=/u01/app/oracle/fast_recovery_area/CDB2/backupset/2014_05

```
_21/o1_mf_nnndf_TAG20140521T010026_9qqyrcyz_.bkp
tag=TAG20140521T010026
channel ORA_DISK_1: restored backup piece 1
channel ORA_DISK_1: restore complete, elapsed time: 00:00:35
Finished restore at 21-05-2014 01:10:10

Starting recover at 21-05-2014 01:10:11
using channel ORA_DISK_1

starting media recovery
media recovery complete, elapsed time: 00:00:01

Finished recover at 21-05-2014 01:10:12

sql statement: alter database datafile 3 online
repair failure complete

RMAN> EXIT
$
```

Practice 7-11: 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_PDB 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 4-4.

Tasks

1. Set the situation where deleted rows have been committed.
 - a. Create a TEST_PDB tablespace in PDB2_2, a local user LOCAL_TEST, and a table. Execute the \$HOME/labs/CDB/setup.sql script.

```
$ sqlplus sys@PDB2_2 as sysdba
Enter password: *****

SQL> @$HOME/labs/CDB/setup.sql
Pluggable database altered.

Tablespace created.

User created.

Grant succeeded.

Disconnected from Oracle Database 12c Enterprise Edition Release
12.1.0.2.0 - 64bit Production
With the Partitioning, OLAP, Advanced Analytics and Real
Application Testing options
$
```

- b. Back up the new tablespace.

```
$ rman target /

connected to target database: CDB2 (DBID=540373866)

RMAN> backup pluggable database pdb2_2;

Starting backup at 21-05-2014 01:11:26
using target database control file instead of recovery catalog
allocated channel: ORA_DISK_1
channel ORA_DISK_1: SID=32 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=00011
name=/u01/app/oracle/oradata/cdb2/pdb2_2/CDB2/F9CD9FE4AC155B62E0
438D23B98B355C/datafile/o1_mf_sysaux_9qommbgs_.dbf
input datafile file number=00010
name=/u01/app/oracle/oradata/cdb2/pdb2_2/CDB2/F9CD9FE4AC155B62E0
438D23B98B355C/datafile/o1_mf_system_9qommbgr_.dbf
input datafile file number=00021
name=/u01/app/oracle/oradata/cdb2/pdb2_2/test_pdb1.f
channel ORA_DISK_1: starting piece 1 at 21-05-2014 01:11:27
channel ORA_DISK_1: finished piece 1 at 21-05-2014 01:12:02
piece
handle=/u01/app/oracle/fast_recovery_area/CDB2/F9CD9FE4AC155B62E
0438D23B98B355C/backupset/2014_05_21/o1_mf_nnndf_TAG20140521T011
127_9qqz9zph_.bkp tag=TAG20140521T011127 comment=NONE
channel ORA_DISK_1: backup set complete, elapsed time: 00:00:35
Finished backup at 21-05-2014 01:12:02

Starting Control File and SPFILE Autobackup at 21-05-2014
01:12:02
piece
handle=/u01/app/oracle/fast_recovery_area/CDB2/autobackup/2014_0
5_21/o1_mf_s_848106722_9qqzc3jx_.bkp comment=NONE
Finished Control File and SPFILE Autobackup at 21-05-2014
01:12:05

RMAN> EXIT
$

```

- c. Create a table with 4 rows by using the `$HOME/labs/CDB/insert.sql` script, check the SCN, delete 2 rows and recheck the SCN.

```
$ sqlplus sys@PDB2_2 as sysdba

Enter password: *****

SQL> @$HOME/labs/CDB/insert.sql

Table created.

1 row created.

1 row created.

1 row created.

1 row created.

Commit complete.

SQL> select timestamp_to_scn(sysdate) from v$database;

TIMESTAMP_TO_SCN(SYSDATE)
-----
                2722628

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 discard the PDB OPEN state if this has not been completed in practice 7-9 task 1.**
- 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 = 2722628;
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 21-05-2014 01:17:19
allocated channel: ORA_DISK_1
channel ORA_DISK_1: SID=12 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 00010 to
/u01/app/oracle/oradata/cdb2/pdb2_2/CDB2/F9CD9FE4AC155B62E0438D2
3B98B355C/datafile/o1_mf_system_9qommbgr_.dbf
channel ORA_DISK_1: restoring datafile 00011 to
/u01/app/oracle/oradata/cdb2/pdb2_2/CDB2/F9CD9FE4AC155B62E0438D2
3B98B355C/datafile/o1_mf_sysaux_9qommbgs_.dbf
```

```

channel ORA_DISK_1: restoring datafile 00021 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/F9CD9FE4AC155B62E0438D23
B98B355C/backupset/2014_05_21/o1_mf_nnndf_TAG20140521T011127_9qq
z9zph_.bkp
channel ORA_DISK_1: piece
handle=/u01/app/oracle/fast_recovery_area/CDB2/F9CD9FE4AC155B62E
0438D23B98B355C/backupset/2014_05_21/o1_mf_nnndf_TAG20140521T011
127_9qqz9zph_.bkp tag=TAG20140521T011127
channel ORA_DISK_1: restored backup piece 1
channel ORA_DISK_1: restore complete, elapsed time: 00:00:25
Finished restore at 21-05-2014 01:17:45

Starting recover at 21-05-2014 01:17:46
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='sCCf'

initialization parameters used for automatic instance:
db_name=CDB2
db_unique_name=sCCf_pitr_pdb2_2_CDB2
compatible=12.1.0.2.0
db_block_size=8192
db_files=200
diagnostic_dest=/u01/app/oracle
_system_trig_enabled=FALSE
sga_target=2560M
processes=200
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

```

Oracle instance started

Total System Global Area 2684354560 bytes

Fixed Size 2919072 bytes

Variable Size 587203936 bytes

Database Buffers 2080374784 bytes

Redo Buffers 13856768 bytes

Automatic instance created

contents of Memory Script:

```
{
# set requested point in time
set until    scn 2722628;
# restore the controlfile
restore clone controlfile;

# mount the controlfile
sql clone 'alter database mount clone database';
}
```

executing Memory Script

executing command: SET until clause

Starting restore at 21-05-2014 01:19:05

allocated channel: ORA_AUX_DISK_1

channel ORA_AUX_DISK_1: SID=12 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/CDB2/autobackup/2014_05_21/o1_mf_s_848106722_9qqzc3jx_.bkp

channel ORA_AUX_DISK_1: piece

handle=/u01/app/oracle/fast_recovery_area/CDB2/autobackup/2014_05_21/o1_mf_s_848106722_9qqzc3jx_.bkp tag=TAG20140521T011202

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/oradata/CDB2/controlfile/o1_mf_9qqzrb4x_.ctl

Finished restore at 21-05-2014 01:19:06

sql statement: alter database mount clone database

contents of Memory Script:

```
{
# set requested point in time
set until   scn 2722628;
# switch to valid datafilecopies
switch clone datafile 10 to datafilecopy

"/u01/app/oracle/oradata/cdb2/pdb2_2/CDB2/F9CD9FE4AC155B62E0438D
23B98B355C/datafile/o1_mf_system_9qommbgr_.dbf";
switch clone datafile 11 to datafilecopy

"/u01/app/oracle/oradata/cdb2/pdb2_2/CDB2/F9CD9FE4AC155B62E0438D
23B98B355C/datafile/o1_mf_sysaux_9qommbgs_.dbf";
switch clone datafile 21 to datafilecopy
"/u01/app/oracle/oradata/cdb2/pdb2_2/test_pdb1.f";
# 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 datafile 6 to new;
# restore the tablespaces in the recovery set and the auxiliary
set
restore clone datafile 1, 4, 3, 6;

switch clone datafile all;
}
```

executing Memory Script

executing command: SET until clause

datafile 10 switched to datafile copy

input datafile copy RECID=3 STAMP=848107162 file
name=/u01/app/oracle/oradata/cdb2/pdb2_2/CDB2/F9CD9FE4AC155B62E0
438D23B98B355C/datafile/o1_mf_system_9qommbgr_.dbf

datafile 11 switched to datafile copy

input datafile copy RECID=4 STAMP=848107162 file
name=/u01/app/oracle/oradata/cdb2/pdb2_2/CDB2/F9CD9FE4AC155B62E0
438D23B98B355C/datafile/o1_mf_sysaux_9qommbgs_.dbf

datafile 21 switched to datafile copy

```

input datafile copy RECID=5 STAMP=848107162 file
name=/u01/app/oracle/oradata/cdb2/pdb2_2/test_pdb1.f

executing command: SET NEWNAME

executing command: SET NEWNAME

executing command: SET NEWNAME

executing command: SET NEWNAME

Starting restore at 21-05-2014 01:19:13
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/oradata/CDB2/datafile/o1_mf_system_%u_.dbf
channel ORA_AUX_DISK_1: restoring datafile 00004 to
/u01/app/oracle/oradata/CDB2/datafile/o1_mf_undotbs1_%u_.dbf
channel ORA_AUX_DISK_1: restoring datafile 00003 to
/u01/app/oracle/oradata/CDB2/datafile/o1_mf_sysaux_%u_.dbf
channel ORA_AUX_DISK_1: restoring datafile 00006 to
/u01/app/oracle/oradata/CDB2/datafile/o1_mf_users_%u_.dbf
channel ORA_AUX_DISK_1: reading from backup piece
/u01/app/oracle/fast_recovery_area/CDB2/backupset/2014_05_21/o1_
mf_nnndf_TAG20140521T010026_9qqyrcyz_.bkp
channel ORA_AUX_DISK_1: piece
handle=/u01/app/oracle/fast_recovery_area/CDB2/backupset/2014_05
_21/o1_mf_nnndf_TAG20140521T010026_9qqyrcyz_.bkp
tag=TAG20140521T010026
channel ORA_AUX_DISK_1: restored backup piece 1
channel ORA_AUX_DISK_1: restore complete, elapsed time: 00:01:05
Finished restore at 21-05-2014 01:20:19

datafile 1 switched to datafile copy
input datafile copy RECID=10 STAMP=848107219 file
name=/u01/app/oracle/oradata/CDB2/datafile/o1_mf_system_9qqzrlfx
_.dbf
datafile 4 switched to datafile copy
input datafile copy RECID=11 STAMP=848107219 file
name=/u01/app/oracle/oradata/CDB2/datafile/o1_mf_undotbs1_9qqzrl
do_.dbf
datafile 3 switched to datafile copy

```

```

input datafile copy RECID=12 STAMP=848107219 file
name=/u01/app/oracle/oradata/CDB2/datafile/o1_mf_sysaux_9qqzrl9f
_.dbf
datafile 6 switched to datafile copy
input datafile copy RECID=13 STAMP=848107219 file
name=/u01/app/oracle/oradata/CDB2/datafile/o1_mf_users_9qqzrlj4_
.dbf

contents of Memory Script:
{
# set requested point in time
set until   scn 2722628;
# 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";
sql clone 'PDB2_2' "alter database datafile
10 online";
sql clone 'PDB2_2' "alter database datafile
11 online";
sql clone 'PDB2_2' "alter database datafile
21 online";
sql clone "alter database datafile  6 online";
# recover pdb
recover clone database tablespace  "SYSTEM", "UNDOTBS1",
"SYSAUX", "USERS" pluggable database
'PDB2_2'    delete archivelog;
sql clone 'alter database open read only';
plsql <<<begin
    add_dropped_ts;
end; >>>;
plsql <<<begin
    save_pdb_clean_scn;
end; >>>;
# shutdown clone before import
shutdown clone abort
plsql <<<begin
    pdbpitr_inspect(pdbname =>  'PDB2_2');
end; >>>;
}
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

sql statement: alter database datafile 10 online

sql statement: alter database datafile 11 online

sql statement: alter database datafile 21 online

sql statement: alter database datafile 6 online

Starting recover at 21-05-2014 01:20:21
using channel ORA_AUX_DISK_1

starting media recovery

archived log for thread 1 with sequence 14 is already on disk as
file
/u01/app/oracle/fast_recovery_area/CDB2/archivelog/2014_05_21/o1
_mf_1_14_9qqzotsl_.arc
channel ORA_AUX_DISK_1: starting archived log restore to default
destination
channel ORA_AUX_DISK_1: restoring archived log
archived log thread=1 sequence=13
channel ORA_AUX_DISK_1: reading from backup piece
/u01/app/oracle/fast_recovery_area/CDB2/backupset/2014_05_21/o1
_mf_1_14_9qqzotsl_.bkp
channel ORA_AUX_DISK_1: piece
handle=/u01/app/oracle/fast_recovery_area/CDB2/backupset/2014_05
_21/o1_mf_1_14_9qqzotsl_.bkp
tag=TAG20140521T010519
channel ORA_AUX_DISK_1: restored backup piece 1
channel ORA_AUX_DISK_1: restore complete, elapsed time: 00:00:01
archived log file
name=/u01/app/oracle/oradata/1_13_848105025.dbf thread=1
sequence=13
channel clone_default: deleting archived log(s)
archived log file
name=/u01/app/oracle/oradata/1_13_848105025.dbf RECID=28
STAMP=848107223

```

```
archived log file
name=/u01/app/oracle/fast_recovery_area/CDB2/archivelog/2014_05_
21/o1_mf_1_14_9qqzotsl_.arc thread=1 sequence=14
media recovery complete, elapsed time: 00:00:07
Finished recover at 21-05-2014 01:20:31

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_9qqzrl9f_.dbf
deleted
auxiliary instance file
/u01/app/oracle/oradata/CDB2/controlfile/o1_mf_9qqzrb4x_.ctl
deleted
Finished recover at 21-05-2014 01:20:50

Statement processed

RMAN> EXIT
$
```

- c. Check the content of the `local_test.tab_test` table.

```
$ sqlplus sys@pdb2_2 as sysdba

Enter password: *****

Connected to:
Oracle Database 12c Enterprise Edition Release 12.1.0.2.0 -
64bit Production
With the Partitioning, OLAP, Advanced Analytics 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=29 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 12 objects

RMAN> BACKUP DATABASE PLUS ARCHIVELOG delete all input;
```

```
..  
RMAN> EXIT  
$
```

Practice 7-12: 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 6-2.

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 12.1.0.2.0 -
64bit Production
With the Partitioning, OLAP, Advanced Analytics 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 4697620480 bytes
Fixed Size                  2923760 bytes
Variable Size              989856528 bytes
Database Buffers          690987520 bytes
Redo Buffers               13852672 bytes
Database mounted.

SQL> ALTER SYSTEM SET
        DB_FLASHBACK_RETENTION_TARGET=2880 SCOPE=BOTH;
```

2


```
System altered.
```

```
SQL> ALTER DATABASE FLASHBACK ON;
```

```
Database altered.
```

```
SQL> ALTER DATABASE OPEN;
```

```
Database altered.
```

```
SQL>
```

2. Preserve the OPEN state for all PDBs.

```
SQL> ALTER PLUGGABLE DATABASE ALL OPEN;
```

```
Pluggable database altered.
```

```
SQL> ALTER PLUGGABLE DATABASE ALL SAVE STATE;
```

```
Pluggable database altered.
```

```
SQL>
```

3. 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';
```

```
2
```

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)
2724037

```
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>
```

4. 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 4697620480 bytes
Fixed Size                  2923760 bytes
Variable Size               989856528 bytes
Database Buffers           3690987520 bytes
Redo Buffers                 13852672 bytes
Database mounted.
SQL> FLASHBACK DATABASE TO SCN 2724037;

Flashback complete.

SQL>
```

- 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';

  2
USERNAME          COM      CON_ID
-----
C##_USER          YES        1

SQL>
```

- 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';

  2
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>
```

- 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  4697620480 bytes
Fixed Size                  2923760 bytes
```

```
Variable Size          989856528 bytes
Database Buffers      3690987520 bytes
Redo Buffers          13852672 bytes
Database mounted.
```

```
SQL> FLASHBACK DATABASE TO SCN 2724037;
```

```
Flashback complete.
```

```
SQL> ALTER DATABASE OPEN RESETLOGS;
```

```
Database altered.
```

```
SQL>
```

8. Check that the C##_USER can connect in each container.

```
SQL> connect C##_USER
```

```
Enter password: *****
```

```
Connected.
```

```
SQL> connect C##_USER@PDB2
```

```
Enter password: *****
```

```
Connected.
```

```
SQL> connect C##_USER@PDB2_2
```

```
Enter password: *****
```

```
Connected.
```

```
SQL> EXIT
```

```
$
```

9. Back up the CDB.

```
$ rman target /
```

```
connected to target database: CDB2 (DBID=534631279)
```

```
RMAN> BACKUP DATABASE PLUS ARCHIVELOG delete all input;
```

```
...
```

```
RMAN> EXIT
```

```
$
```

10. The following practices do not use the CDB. Release resources by shutting down the `cdb2` instance.

```
$ . oraenv
ORACLE_SID = [cdb2] ? 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 12.1.0.2.0 -
64bit Production
With the Partitioning, OLAP, Advanced Analytics 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
12.1.0.2.0 - 64bit Production
With the Partitioning, OLAP, Advanced Analytics and Real
Application Testing options
$
```


Practices for Lesson 8: Heat Map, Automatic Data Optimization and Online Datafile Move

Chapter 8

Practices for Lesson 8: Overview

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 familiarize yourself with moving datafiles online.

Assumptions

The environment is prepared beforehand; that is, 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 8-1: Enabling Heat Map

Overview

In this practice, you will enable activity tracking or heat map.

Tasks

1. Perform several operations to clean up 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 = [cdb2] ? orcl
The Oracle base for
ORACLE_HOME=/u01/app/oracle/product/12.1.0/dbhome_1 is
/u01/app/oracle
$
```

- c. Start up the orcl instance.

```
$ sqlplus / as sysdba

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

Total System Global Area      503316480 bytes
Fixed Size                     2916056 bytes
Variable Size                  272630056 bytes
Database Buffers               222298112 bytes
Redo Buffers                    5472256 bytes
Database mounted.
Database opened.
SQL> EXIT
$
```

- d. Run the `ADO_cleanup.sh` script to clean up any existing ADO policy and tablespaces.

```
$ ./ADO_cleanup.sh
alter table scott.employee ilm delete_all;
*
ERROR at line 1:
ORA - 00942: table or view does not exist

DROP TABLESPACE adotbs INCLUDING CONTENTS AND DATAFILES
*
ERROR at line 1:
ORA-00959: tablespace 'ADOTBS' does not exist

DROP TABLESPACE low_cost_store INCLUDING CONTENTS AND DATAFILES
*
ERROR at line 1:
ORA-00959: tablespace 'LOW_COST_STORE' does not exist
$
```

2. Set the `HEAT_MAP` instance parameter to `ON` at the instance scope.

```
$ sqlplus / as sysdba
Connected to:
Oracle Database 12c Enterprise Edition Release 12.1.0.2.0 -
64bit Production
With the Partitioning, OLAP, Advanced Analytics and Real
Application Testing options

SQL> ALTER SYSTEM SET heat_map=on SCOPE=BOTH;

System altered.

SQL> EXIT
$
```

Practice 8-2: 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_STORE` tablespace when the source `ADOTBS` 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
User altered.

Grant succeeded.

Grant succeeded.

Grant succeeded.

$
```

- b. Execute the `$HOME/labs/ADO/create_tbs.sql` script to create two tablespaces:
 - The `ADOTBS` tablespace to store the `SCOTT.EMPLOYEE` and insert rows into the `SCOTT.EMPLOYEE` table
 - The tablespace `LOW_COST_STORE` where the `SCOTT.EMPLOYEE` table may be moved to due to space pressure

```
$ sqlplus system

Enter password: *****

SQL> @$HOME/labs/ADO/create_tbs.sql

Tablespace created.

Tablespace created.

SQL>
```

2. Create and store the SCOTT.EMPLOYEE table on the ADOTBS tablespace and blow it up with about 3500 rows. The rows inserted should raise the percentage of empty space in ADOTBS 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. Execute the \$HOME/labs/ADO/emp.sql script to create and store the SCOTT.EMPLOYEE table on the ADOTBS tablespace and blow it up with about 3500 rows.

```
SQL> CONNECT scott
Enter password: *****
Connected.
SQL> @emp.sql
drop table employee purge
      *
ERROR at line 1:
ORA-00942: table or view does not exist

Table created.

14 rows created.

PL/SQL procedure successfully completed.

      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      4      5      6      7
OBJECT_NAME  Tracking Time          Seg write  Ful Loo
```

Copyright © 2014, Oracle and/or its affiliates. All rights reserved.

```

-----
EMPLOYEE      21-MAY-14 01:38:47          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      21-may-14 01:38:58 21-may-14 01:38:58

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      21-may-14 01:39:14 YES          NO          YES NO

SQL>

```

- c. Execute the @\$HOME/labs/ADO/check.sql script to 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> @check.sql
```

Tablespace	Size (MB)	Free (MB)	% Free	% Used
SYSTEM	810	.875	0	100
SYSAUX	980	41.625	4	96
EXAMPLE	1277.5	58.4375	5	95
USERS	5	3.3125	66	34
UNDOTBS1	145	114.6875	79	21
ADOTBS	10	8.6875	<u>87</u>	<u>13</u>
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
-----
P1           STORAGE      SEGMENT LOW_COST_STORE
```

```
SQL> SELECT policy_name, object_name, inherited_from, enabled
        FROM   user_ilmobjects;
```

POLICY_NAME	OBJECT_NAME	INHERITED_FROM	ENA
-----	-----	-----	----
P1	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
RETENTION TIME	30
JOB LIMIT	2
EXECUTION MODE	2
EXECUTION INTERVAL	15
TBS PERCENT USED	85
TBS PERCENT FREE	25
POLICY TIME	0

```
8 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
RETENTION TIME	30
JOB LIMIT	2
EXECUTION MODE	2
EXECUTION INTERVAL	15
TBS PERCENT USED	5
TBS PERCENT FREE	95
POLICY TIME	0
8 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 execute the @\$HOME/labs/ADO/ilm.sql script that uses the following PL/SQL block connected as the ADO policy owner:

```
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;
/
```

```
SQL> CONNECT scott
Enter password: *****
Connected.
SQL> @ilm.sql

Session altered.

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.

```
SQL> @check.sql
```

Tablespace	Size (MB)	Free (MB)	% Free	% Used
SYSTEM	810	.125	0	100
SYSAUX	990	51.0625	5	95
EXAMPLE	1277.5	58.4375	5	95
USERS	5	3.3125	66	34
UNDOTBS1	145	113.6875	78	22
ADOTBS	10	9	<u>90</u>	<u>10</u>
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
-----
      2 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
-----
      2 P1           EMPLOYEE SELECTED FOR EXECUTION ILMJOB366
```

```
SQL> SELECT task_id, job_name, job_state FROM user_ilmresults;
```

```
TASK_ID JOB_NAME                JOB_STATE
-----
      2 ILMJOB366                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 8-3: 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 at the row level so that blocks 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.

We will make the ADO evaluation interval short enough to be practical for the practice. The procedure to accomplish this is `DBMS_ILM_ADMIN.CUSTOMIZE_ILM` which changes the `POLICY TIME` to 1 to change the evaluation of days to seconds.

Assumptions

If you did execute the previous practice 8-2, ensure that you set the `HEAT_MAP` initialization parameter to `ON` at the instance scope as described in Practice 8-1 task 2.

Tasks

1. Execute the `$HOME/labs/ADO/comp.sql` script to 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.

```
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);
got varchar2(1) := '';
n_uncmp number :=0;
n_advanced number :=0;
n_other number :=0;
begin
  stmt := 'select dbms_compression.get_compression_type(';
  stmt := stmt||got||owner||got||','||got||tabname||got;
  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;
/

```

```

$ sqlplus / as sysdba
Connected.
SQL> @comp.sql

Procedure created.

Grant succeeded.

Synonym created.

SQL>

```

2. Create the SCOTT.EMPLOYEE table and insert rows. Execute the \$HOME/labs/ADO/emp2.sql script.

```

SQL> CONNECT scott
Enter password: *****
Connected.
SQL> @emp2.sql
drop table employee purge
*
ERROR at line 1:
ORA-00942: table or view does not exist

```

```

Table created.

14 rows created.

PL/SQL procedure successfully completed.

      COUNT(*)
-----
          3584

SQL>

```

3. Add a row-level compression policy on SCOTT.EMPLOYEE table. Use the following columns format.

```

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 A4

```

```

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 A4
SQL> ALTER TABLE scott.employee ILM ADD POLICY ROW STORE
COMPRESS ADVANCED ROW 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 ROW AFTER 30 DAYS OF NO MODIFICATION;
ALTER TABLE scott.employee ILM ADD POLICY ROW STORE COMPRESS
ADVANCED ROW AFTER 30 DAYS OF NO MODIFICATION
*
ERROR at line 1:
ORA-38342: heat map not enabled

SQL>
```

4. 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

POLI ACTION_TYPE SCOPE      COMPRESSION_LEVEL CONDITION_TYPE
CONDITION_DAYS
-----
P21  COMPRESSION ROW      ADVANCED              LAST MODIFICATION TIME
              30
```

SQL>

5. 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> COL object_name FORMAT A10
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';

OBJECT_NAM Uncomp Rows   Comp Rows HCC Comp Rows
-----
EMPLOYEE           3584             0             0

SQL>

```

6. You cannot wait until the 30 days delay is over. To indicate that the policy is specified in seconds rather than in days, set the `POLICY TIME` to 1 (seconds) instead of the default value 0 (days) to test ADO policy evaluation quickly instead of waiting for the policy duration.

```

SQL> connect / as sysdba
Connected.
SQL> exec
dbms_ilm_admin.customize_ilm(dbms_ilm_admin.POLICY_TIME,dbms_ilm
_admin.ILM_POLICY_IN_SECONDS)

PL/SQL procedure successfully completed.

SQL> COL name format A20
SQL> COL value format 9999
SQL> SELECT * FROM dba_ilmparameters;

NAME                                VALUE
-----
ENABLED                             1
RETENTION TIME                       30
JOB LIMIT                            2
EXECUTION MODE                       2

```

```
EXECUTION INTERVAL      15
TBS PERCENT USED        85
TBS PERCENT FREE        25
POLICY TIME              1
```

```
8 rows selected.
```

```
SQL>
```

7. Wait until 30 seconds (instead of 30 days) have passed without any modification on SCOTT.EMPLOYEE table. For the purpose of the demo, you will not wait until MMON evaluates the ADO policies. You launch the ADO policy evaluation and ADO task execution immediately by executing the \$HOME/labs/ADO/ilm.sql script.

```
SQL> connect scott
```

```
Enter password: *****
```

```
Connected.
```

```
SQL> @ilm.sql
```

```
Session altered.
```

```
PL/SQL procedure successfully completed.
```

```
SQL>
```

8. Display the result of the executed task. Use the following column formats.

```
COL task_id format 99999
```

```
COL task_owner format A8
```

```
COL policy_name format A4
```

```
COL job_name format A10
```

```
COL SELECTED_FOR_EXECUTION format A22
```

```
SQL> select task_id, task_owner, state
        from dba_ilmtasks where task_owner='SCOTT';
```

```
2
```

```
TASK_ID TASK_OWN STATE
```

```
-----
```

```
21 SCOTT ACTIVE
```

```
SQL>
```

9. If the STATE column displays ACTIVE, the task is still executing. Rerun the SELECT statement until STATE shows COMPLETED.

```
SQL> select task_id, task_owner, state
```

```
        from dba_ilmtasks where task_owner='SCOTT';
```

```
2
```



```

TASK_ID TASK_OWN STATE
-----
      21 SCOTT    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
-----
      21 P21  EMPLOYEE      SELECTED FOR EXECUTION ILMJOB446

SQL>

```

10. 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           : 524
Adv/basic compressed   : 3060
Others                  : 0

PL/SQL procedure successfully completed.

SQL> SELECT compression, compress_for
        FROM user_tables WHERE table_name = 'EMPLOYEE';
2
COMPRESS COMPRESS_FOR
-----
DISABLED

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
OBJECT_N Uncomp Blocks  Comp Rows Comp Blocks HCC Comp Blocks
-----
EMPLOYEE              13      3060              18              0

SQL>

```

Practice 8-4: 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. Note that the list below may display different rows than yours. This depends on whether you performed other queries or DML statements.

```
SQL> select OBJ#, TS#, TRACK_TIME from sys.heat_map_stat$;
```

OBJ#	TS#	TRACK_TIM
93782	6	18-APR-14
19704	1	21-MAY-14
19701	1	21-MAY-14
92680	6	21-MAY-14
17490	1	21-MAY-14
19050	1	21-MAY-14
17493	1	21-MAY-14
382	1	21-MAY-14
91231	1	21-MAY-14
18267	1	21-MAY-14
92558	6	21-MAY-14
19921	1	21-MAY-14
92574	6	21-MAY-14
8475	1	21-MAY-14
92687	6	21-MAY-14

```

        6027          1 21-MAY-14
        8695          1 21-MAY-14
         385          1 21-MAY-14
         591          1 21-MAY-14
         592          1 21-MAY-14
        6028          1 21-MAY-14
       19918          1 21-MAY-14
      18264          1 21-MAY-14
         -1         -1 21-MAY-14

24 rows selected.

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 21-MAY-14

SQL>
```

You notice that the procedure deletes all rows in HEAT_MAP_STAT\$ table except the dummy row.

Practice 8-5: 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

6 rows selected.

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
      '/u01/app/oracle/oradata/orcl/online_tbs01.dbf' TO
      '/u01/app/oracle/oradata/orcl/online/online_tbs01.dbf';
  2      3
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 10493952 May 21 02:08
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;
      2      3
Database altered.

SQL>
```

```
SQL> !ls -l /u01/app/oracle/oradata/orcl/online_tbs01.dbf
-rw-r----- 1 oracle oinstall 10493952 May 21 02:09
/u01/app/oracle/oradata/orcl/online_tbs01.dbf

SQL> !ls -l /u01/app/oracle/oradata/orcl/online/online*
-rw-r----- 1 oracle oinstall 10493952 May 21 02:09
/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;
      2      3
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 10493952 May 21 02:10  
/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 9: In-Database Archiving and Temporal Validity

Chapter 9

Practices for Lesson 9: Overview

Practices Overview

In this practice, you will familiarize yourself with new features like In-Database Archiving and Temporal Validity.

Practice 9-1: 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. Make sure you are in the `~/labs/VT` directory.

```
$ cd ~/labs/VT
$
```

2. Connected under `SYSTEM`, create the table `HR.EMP_ARCH` with `ROW ARCHIVAL` attribute.

```
$ . 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

Enter password: *****

Connected to:
Oracle Database 12c Enterprise Edition Release 12.1.0.2.0 -
64bit Production
With the Partitioning, OLAP, Advanced Analytics 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>

```

4. Execute the \$HOME/labs/VT/emp_arch.sql script to insert new rows in the table.

```

SQL> @emp_arch.sql

1 row created.

1 row created.

1 row created.

1 row created.

Commit complete.

SQL>

```

5. 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>

```

6. 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>
```

7. Update `ORA_ARCHIVE_STATE` column to reflect a non-active state for employee numbers 101 and 102, by using the `DBMS_ILM.ARCHIVESTATENAME` function or the value directly.

```
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>
```

8. 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>
```

9. 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>
```

10. 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>
```

11. Disable the row-archival attribute on `HR.EMP_ARCH` table.

```
SQL> ALTER TABLE hr.emp_arch NO ROW ARCHIVAL;

Table altered.
```

```
SQL>
```

12. 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 9-2: 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.

```
$ sqlplus system

Enter password: *****

Connected to:
Oracle Database 12c Enterprise Edition Release 12.1.0.2.0 -
64bit Production
With the Partitioning, OLAP, Advanced Analytics 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 end 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> col table_name format A10
SQL> col column_name format A20
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(VAID_TIME_S TO_CHAR(VAID_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 );
      2      3      4
Table created.

SQL>
```

```
SQL> DESC hr.emp

Name                               Null?      Type
-----
EMPNO                               NUMBER
SALARY                              NUMBER
DEPTID                              NUMBER
NAME                                VARCHAR2(100)

SQL>
```

```
SQL> @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 do not appear. Use the following column formats.

```
col name format A8
```

```
col empno format 999
```

```
col valid_time_start Format a35
```

```
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. Execute \$HOME/labs/VT/query1.sql to view all rows with explicit named valid-time columns.

```
SQL> @query1.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>

```

10. Execute `$HOME/labs/VT/query2.sql` to view rows using valid-time temporal flashback queries.

- a. Using an AS OF query:

```

SQL> @query2.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
JAMES        31-dec-1992          31-dec-1994

SQL>

```

Only employees who were still valid on the date '31-DEC-1992' are displayed.

- b. Execute `$HOME/labs/VT/query3.sql` to display only employees who were still valid on the date '01-JAN-2013'.

```

SQL> @query3.sql

NAME          Start                End
-----
JIM           01-jan-1992          31-dec-2013
TOM           01-jan-1994

SQL>

```

- c. Execute `$HOME/labs/VT/query4.sql` to display all employees whose `VALID_TIME_START` is less than or equal to '31-DEC-1992' and `VALID_TIME_END` greater than '31-DEC-1993'.

```

SQL> @query4.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
JAMES     31-dec-1992      31-dec-1994

```

```
6 rows selected.
```

```
SQL>
```

- d. Execute \$HOME/labs/VT/query5.sql to display all employees whose VALID_TIME_START is less than or equal to '31-DEC-2011' and VALID_TIME_END greater than or equal to '31-DEC-2012'.

```
SQL> @query5.sql
```

```

NAME      Start              End
-----
JIM        01-jan-1992      31-dec-2013
JEAN       01-jan-1992      31-dec-2012
TOM        01-jan-1994

```

```
SQL>
```

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
-----
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 9-3: 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 for
ORACLE_HOME=/u01/app/oracle/product/12.1.0/dbhome_1 is
/u01/app/oracle
$
```

- b. Execute the `$HOME/labs/FDA/test.sql` script to create and populate the `hr.test_table1` table.

```
$ sqlplus / as sysdba

Connected to:
Oracle Database 12c Enterprise Edition Release 12.1.0.2.0 -
64bit Production
With the Partitioning, OLAP, Advanced Analytics and Real
Application Testing options

SQL> @test.sql

DROP TABLE hr.test_table1
          *
ERROR at line 1:
ORA-00942: table or view does not exist

Table created.

1 row created.

1 row created.

1 row created.
```

```
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.TEST_TABLE1`. 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.TEST_TABLE1 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
Enter password: *****
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_93793      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. Use the following column formats.

```
COL rid FORMAT A18
COL name FORMAT A17
COL startscn FORMAT 9999999
COL endscn FORMAT 9999999
```

```
SQL> truncate table hr.test_table1;

Table truncated.

SQL> select * from hr.SYS_FBA_HIST_93793;

RID                                STARTSCN    ENDSCN  XID                                O  NUM
-----
NAME                                NOW
-----
AAAW5hAAGAAAADtAAA  2137865    2137871  0300190075070000 U    1
Premier test row  21-MAY-14

AAAW5hAAGAAAADtAAA                                2137865                                1
First test row   21-MAY-14

AAAW5hAAGAAAADtAAA  2137871    2137975  0A0010009E050000 U    1
Primero test row 21-MAY-14

AAAW5hAAGAAAADtAAB                                2137975                                2
Second test row  21-MAY-14

AAAW5hAAGAAAADtAAC                                2137975                                3
Third test row

SQL>
```

4. Collect the usernames of users who performed the `UPDATE` operations.

```
SQL> CONNECT / AS SYSDBA
Connected.
SQL> select DBMS_FLASHBACK_ARCHIVE.GET_SYS_CONTEXT
           ('0300190075070000', '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      8
User Name
-----
VERSIONS_XID
-----
VERSIONS_STARTTIME
-----
VERSIONS_ENDTIME
-----
NUM NAME
-----

0300190075070000
21-MAY-14 02.44.29.0000000000 AM
21-MAY-14 02.44.35.0000000000 AM
      1 Premier test row

21-MAY-14 02.44.29.0000000000 AM
      1 First test row

0A0010009E050000
21-MAY-14 02.44.35.0000000000 AM
21-MAY-14 02.45.14.0000000000 AM
      1 Primero test row

SQL>

```

5. Collect the module names of the users who performed the UPDATE operations.

```
SQL> COL "Module Name" FORMAT A30
SQL> select DBMS_FLASHBACK_ARCHIVE.GET_SYS_CONTEXT
        ('0300190075070000', '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 Premier test row
SQL*Plus                                1 First test row
SQL*Plus                                1 Primero test row

SQL> EXIT
--
```

Practice 9-4: Cleaning Up FDA

Overview

In this practice, you clean up the FDA tablespace.

Assumptions

You created the FDA tablespace during Practice 9-3.

Tasks

1. Reconnect as `SYSDBA` to execute the `$HOME/labs/FDA/FDA_cleanup.sql` script to disable flashback archive on the `HR.TEST_TABLE1` table and drop the `FDA_TBS` tablespace.

```
$ sqlplus / as sysdba

Connected to:
Oracle Database 12c Enterprise Edition Release 12.1.0.2.0 -
64bit Production
With the Partitioning, OLAP, Advanced Analytics and Real
Application Testing options

SQL> @FDA_cleanup.sql

Table altered.

Flashback archive dropped.

Tablespace dropped.

$
```


Practices for Lesson 10: Auditing

Chapter 10

Practices for Lesson 10: Overview

Practices Overview

In the practices for this lesson, you first enable unified audit, then configure audit policies to audit RMAN operations and finally configure audit policies to audit `SYS` and `SH` users while updating employees' salaries.

You then view the audited data in the `UNIFIED_AUDIT_TRAIL` view.

Practice 10-1: 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 for
ORACLE_HOME=/u01/app/oracle/product/12.1.0/dbhome_1 is
/u01/app/oracle
$
```

```
$ lsnrctl stop

LSNRCTL for Linux: Version 12.1.0.2.0 - on 21-MAY-2014 02:50:36

Copyright (c) 1991, 2014, Oracle. All rights reserved.

Connecting to
(DESCRIPTION=(ADDRESS=(PROTOCOL=IPC)(KEY=EXTPROC1521)))
The command completed successfully
$
```

- b. Shut down all instances.

```
$ pgrep -lf pmon
2464 ora_pmon_em12rep
24567 ora_pmon_orcl
$
```

- 1) Shut down the orcl instance.

```
$ sqlplus / as sysdba

Connected to:
Oracle Database 12c Enterprise Edition Release 12.1.0.2.0 -
64bit Production
With the Partitioning, OLAP, Advanced Analytics and Real
Application Testing options

SQL> shutdown immediate
Database closed.
```

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

2) 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 4
Copyright (c) 1996, 2014 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 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 12.1.0.2.0 -
64bit Production
With the Partitioning, OLAP, Advanced Analytics and Real
Application Testing options

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

3) Verify that all instances are down.

```
$ pgrep -lf 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 -Wl,--disable-new-dtags -
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 -lnnzst12 -lzt12 -
```

Copyright © 2014, Oracle and/or its affiliates. All rights reserved.

```

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 -lnnzst12 -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 -lserver12"; 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 -lnnzst12 -lzt12 -lxml12 -locr12 -
locrb12 -locrut12 -lhasgen12 -lskgxn2 -lnnzst12 -lzt12 -lxml12
-lgeneric12 -loraz -llzopro -lorabz2 -lipp_z -lipp_bz2 -
lippdcmerged -lippsemerged -lippdcmerged -lippsmmerged -
lippcore -lippcpmerged -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 -lasmc1nt12 -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 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 mount
ORACLE instance started.

Total System Global Area      503316480 bytes
Fixed Size                     2916056 bytes
Variable Size                 272630056 bytes
Database Buffers              222298112 bytes
Redo Buffers                   5472256 bytes
Database mounted.

SQL> ALTER DATABASE ARCHIVELOG;

Database altered.

SQL> ALTER DATABASE OPEN;

Database altered.

SQL> EXIT
Disconnected from Oracle Database 12c Enterprise Edition Release
12.1.0.2.0 - 64bit
With the Partitioning, OLAP, Advanced Analytics, Real
Application Testing and Unified Auditing options
$

```

You can see that the Unified Auditing option is enabled in the SQL*Plus banner.

b. Restart the listener.

```

$ lsnrctl start
...
Connecting to
(DESCRIPTION=(ADDRESS=(PROTOCOL=IPC) (KEY=EXTPROC1521)))
STATUS of the LISTENER
-----
Alias                     LISTENER
Version                   TNSLSNR for Linux: Version 12.1.0.2.0
-
Start Date                21-MAY-2014 03:14:21
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/<your_hostname>/listener/alert/log.
xml
Listening Endpoints Summary...
  (DESCRIPTION= (ADDRESS= (PROTOCOL=ipc) (KEY=EXTPROC1521)))

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


Practice 10-2: Auditing RMAN Backup and Recovery Operations

In this practice, you perform RMAN backups. 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. Restore and recovery operations are also audited.

Assumptions

Practice 10-1 successfully enabled unified audit.

Tasks

1. Perform an RMAN backup of the `USERS` tablespace.

```
$ rman target /

connected to target database: ORCL (DBID=1315477536)

RMAN> backup tablespace USERS;

Starting backup at 21-MAY-14
using target database control file instead of recovery catalog
allocated channel: ORA_DISK_1
channel ORA_DISK_1: SID=240 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 21-MAY-14
channel ORA_DISK_1: finished piece 1 at 21-MAY-14
piece
handle=/u01/app/oracle/fast_recovery_area/ORCL/backupset/2014_05
_21/o1_mf_nnndf_TAG20140521T031452_9qr6kdcz_.bkp
tag=TAG20140521T031452 comment=NONE
channel ORA_DISK_1: backup set complete, elapsed time: 00:00:01
Finished backup at 21-MAY-14

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 12.1.0.2.0 -
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 21-MAY-14
using target database control file instead of recovery catalog
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 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/2014_05_21/o1_
mf_nnndf_TAG20140521T031452_9qr6kdcz_.bkp
channel ORA_DISK_1: piece
handle=/u01/app/oracle/fast_recovery_area/ORCL/backupset/2014_05
_21/o1_mf_nnndf_TAG20140521T031452_9qr6kdcz_.bkp
tag=TAG20140521T031452
channel ORA_DISK_1: restored backup piece 1
channel ORA_DISK_1: restore complete, elapsed time: 00:00:01
Finished restore at 21-MAY-14

RMAN> recover tablespace USERS;

Starting recover at 21-MAY-14
using channel ORA_DISK_1

starting media recovery
media recovery complete, elapsed time: 00:00:01

Finished recover at 21-MAY-14

RMAN> exit
$

```

- d. Put the tablespace USERS back online.

```

$ sqlplus system

Enter password: *****

Connected to:
Oracle Database 12c Enterprise Edition Release 12.1.0.2.0 -
64bit Production
With the Partitioning, OLAP, Advanced Analytics, Real
Application Testing and Unified Auditing options

SQL> alter tablespace USERS online;

Tablespace altered.

SQL>

```

3. View the resulting audit data.

```
SQL> select DBUSERNAME, RMAN_OPERATION
       from   UNIFIED_AUDIT_TRAIL
       where  RMAN_OPERATION is not null;

 2      3

DBUSERNAME                                RMAN_OPERATION
-----
SYS                                           Backup
SYS                                           Restore
SYS                                           Recover

SQL> exit
$
```

Practice 10-3: Auditing SYS and End-Users (Optional)

Overview

In this practice, you will audit actions performed by the `SYS` user and `EMMA` end-user in `orcl`, which are not audited by the `ORA_SECURECONFIG` predefined audit policy. The `SYS` user is allowed to update employees' salary and `EMMA` is allowed to select employees' salary but not to update them. You want to control if the `SYS` user uses his privilege to update employees' salary and if `EMMA` attempts to do it. Furthermore, you will audit any employee row deletion except by `HR`.

Tasks

1. Audit any update action on the `HR.EMPLOYEES` table by either `SYS` or `EMMA` users.
 - a. Execute the `$HOME/labs/Security/emma.sql` script to create the user `EMMA` and grant her the `SELECT` privilege on `HR.EMPLOYEES` table.

```
$ sqlplus system

Enter password: *****

Connected to:
Oracle Database 12c Enterprise Edition Release 12.1.0.2.0 -
64bit Production
With the Partitioning, OLAP, Advanced Analytics, Real
Application Testing and Unified Auditing options

SQL> @emma.sql

User created.

Grant succeeded.

Grant succeeded.

SQL>
```

- b. Create and enable an audit policy that audits any update action on the `HR.EMPLOYEES` table by either `SYS` or `EMMA` users. You want to audit only the successful operations from `SYS`.

```
SQL> CREATE AUDIT POLICY aud_update_sal_pol
        ACTIONS update ON hr.employees;

2      3
Audit policy created.

SQL> AUDIT POLICY aud_update_sal_pol BY sys WHENEVER SUCCESSFUL;

Audit succeeded.

SQL> AUDIT POLICY aud_update_sal_pol BY emma;

Audit succeeded.

SQL>
```

2. Create and enable an audit policy that audits any employee row deletion by any user except `EMMA`.

```
SQL> CREATE AUDIT POLICY aud_delete_emp_pol
        ACTIONS delete ON hr.employees;

2      3
Audit policy created.

SQL> AUDIT POLICY aud_delete_emp_pol EXCEPT emma;

Audit succeeded.

SQL>
```

3. Display the audit policies.

```
SQL> COL user_name FORMAT A8
SQL> COL policy_name FORMAT A18
SQL> SELECT * FROM AUDIT_UNIFIED_ENABLED_POLICIES
        WHERE POLICY_NAME like 'AUD_%_POL';

2
USER_NAM POLICY_NAME          ENABLED_ SUC FAI
-----
SYS      AUD_UPDATE_SAL_POL BY      YES NO
EMMA     AUD_UPDATE_SAL_POL BY      YES YES
EMMA     AUD_DELETE_EMP_POL EXCEPT YES YES

SQL>
```

4. Connect as SYS and execute an update command on the HR.EMPLOYEES table.

```
SQL> CONNECT / AS SYSDBA
Connected.
SQL> UPDATE hr.employees SET salary=salary+100
      WHERE last_name='Me';
      2
0 rows updated.

SQL> UPDATE hr.employees SET salary=sa+100 ;
UPDATE hr.employees SET salary=sa+100
                        *
ERROR at line 1:
ORA-00904: "SA": invalid identifier

SQL>
```

5. Connect as EMMA and execute an update command on the HR.EMPLOYEES table.

```
SQL> CONNECT emma
Enter password: *****
Connected.
SQL> UPDATE hr.employees SET salary=salary+100
      WHERE last_name='Me';
      2
      *
ERROR at line 1:
ORA-01031: insufficient privileges

SQL>
```

6. Connect as SYS and delete Urman employee on the HR.EMPLOYEES table.

```
SQL> CONNECT / AS SYSDBA
Connected.
SQL> DELETE FROM hr.employees WHERE last_name='Urman';

1 row deleted.

SQL> ROLLBACK;

Rollback complete.

SQL>
```

7. Connect as EMMA and execute the same command.

```
SQL> CONNECT emma
Enter password: *****
Connected.
SQL> DELETE FROM hr.employees WHERE last_name='Urman';

1 row deleted.

SQL> ROLLBACK;

Rollback complete.

SQL>
```

8. Connect as SYSTEM to display the audited actions. Use the following column formats.

```
col dbusername format A10
col action_name format A12
col system_privilege_used FORMAT A30
col object_name format A10
```

```
SQL> CONNECT system
Enter password: *****
Connected.
SQL> SELECT dbusername, action_name, object_name,
           system_privilege_used, unified_audit_policies
FROM unified_audit_trail
WHERE UNIFIED_AUDIT_POLICIES like 'AUD_%_POL';

DBUSERNAME ACTION_NAME OBJECT_NAM SYSTEM_PRIVILEGE_USED
-----
UNIFIED_AUDIT_POLICIES
-----
EMMA          UPDATE          EMPLOYEES
AUD_UPDATE_SAL_POL

SYS           UPDATE          EMPLOYEES  SYSDBA
AUD_UPDATE_SAL_POL

SYS           DELETE          EMPLOYEES  SYSDBA
AUD_DELETE_EMP_POL

SQL>
```


You notice that:

- EMMA's attempt to update the salary of employees is recorded although she did not succeed because of the lack of privileges, whereas her attempt to delete an employee row was not as expected.
- SYS's operations are all recorded except the update operation that failed. This was also expected as you only wanted to record the successful operations.

9. Execute the `$HOME/labs/Security/noaud.sql` to disable and drop the audit policies.

```
SQL> @noaud.sql

Noaudit succeeded.

Noaudit succeeded.

Audit Policy dropped.

Noaudit succeeded.

Noaudit succeeded.

Audit Policy dropped.

SQL>
```

10. Drop the user EMMA.

```
SQL> DROP USER emma;

User dropped.

SQL> EXIT
$
```

Practice 10-4: Excluding DBSNMP Login Events (Optional)

Overview

In this practice, you will exclude all login events performed by the DBSNMP user which are audited by default by the `ORA_SECURECONFIG` predefined audit policy. This generates too many rows in the audit storage.

Tasks

1. Verify that the `ORA_SECURECONFIG` predefined audit policy is enabled by default. Use the following column formats.

```
COL USER_NAME          FORMAT A10
COL POLICY_NAME         FORMAT A20
COL ENABLED_OPT         FORMAT A8
COL SUCCESS             FORMAT A3
COL FAILURE             FORMAT A3
```

```
$ sqlplus system

Enter password: *****

Connected to:
Oracle Database 12c Enterprise Edition Release 12.1.0.2.0 -
64bit Production
With the Partitioning, OLAP, Advanced Analytics, Real
Application Testing and Unified Auditing options

SQL> SELECT POLICY_NAME, ENABLED_OPT, USER_NAME, SUCCESS,
           FAILURE
        FROM AUDIT_UNIFIED_ENABLED_POLICIES;

 2      3
POLICY_NAME          ENABLED_ USER_NAME  SUC FAI
-----
ORA_SECURECONFIG     BY        ALL USERS  YES YES
ORA_LOGON_FAILURES   BY        ALL USERS  NO  YES

SQL>
```

2. Update the `ORA_SECURECONFIG` predefined audit policy so that all login events performed are still recorded except those from the DBSNMP user.
 - a. Disable the `ORA_SECURECONFIG` audit policy.

```
SQL> NOAUDIT POLICY ORA_SECURECONFIG;

Noaudit succeeded.

SQL>
```

- b. Re-enable the `ORA_SECURECONFIG` audit policy excluding the `DBSNMP` user.

```
SQL> AUDIT POLICY ORA_SECURECONFIG EXCEPT dbsnmp;

Audit succeeded.

SQL>
```

- c. Verify.

```
SQL> SELECT POLICY_NAME, ENABLED_OPT, USER_NAME,
           SUCCESS, FAILURE
        FROM AUDIT_UNIFIED_ENABLED_POLICIES;

  2      3
POLICY_NAME          ENABLED_ USER_NAME  SUC FAI
-----
ORA_SECURECONFIG    EXCEPT  DBSNMP      YES YES
ORA_LOGON_FAILURES  BY          ALL USERS   NO  YES

SQL> EXIT
$
```


Practices for Lesson 11: Privileges

Chapter 11

Practices for Lesson 11: Overview

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 11-1: Managing 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 for
ORACLE_HOME=/u01/app/oracle/product/12.1.0/dbhome_1 is
/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 12.1.0.2.0 -
64bit Production
With the Partitioning, OLAP, Advanced Analytics, 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 the 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

234 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 `SYSPRIVILEGE` privilege in order to perform backup, restore, and recover operations, hence act as the `SYSPRIVILEGE` 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 `SYSPRIVILEGE` entry, it does not accept any `SYSPRIVILEGE` entry.

- a. Recreate the file in 12 format, compatible with the `SYSPRIVILEGE` 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 12.1.0.2.0 -
64bit Production
With the Partitioning, OLAP, Advanced Analytics, Real
Application Testing and Unified Auditing options
SQL> grant create session, SYSPRIVILEGE 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@orcl as SYSBACKUP
Enter password: *****
Connected.
SQL> SHOW USER
USER is "SYSBACKUP"
SQL> exit
$
```

- d. Test the remote connection in RMAN.

```
$ rman target john@orcl

Enter password: *****

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@orcl AS SYSBACKUP"'

Enter password: *****
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 11-2: 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 for
ORACLE_HOME=/u01/app/oracle/product/12.1.0/dbhome_1 is
/u01/app/oracle
$
```

2. Run the `priv_setup.sql` script to create JIM, TOM, U1, U2, and KATE users, HR_MGR and SALES_CLERK roles.

```
$ sqlplus system

Enter password: *****

Connected to:
Oracle Database 12c Enterprise Edition Release 12.1.0.2.0 -
64bit Production
With the Partitioning, OLAP, Advanced Analytics, 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
```

Copyright © 2014, Oracle and/or its affiliates. All rights reserved.

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 ul 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.

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, then as TOM who selects rows from the SH.SALES table and finally as U1 who creates a procedure selecting data from the U2.T1 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

Connected.

Procedure created.

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. Use the following column formats.

COL capture FORMAT A14

COL username FORMAT A10

COL object_owner FORMAT A12

COL object_name FORMAT A12

COL obj_priv FORMAT A12

```
SQL> SELECT  capture, username, object_owner, object_name,
              obj_priv
        FROM    dba_used_objprivs
       WHERE   username IN ('JIM', 'TOM', 'U1', 'U2')
       AND     object_name NOT IN
              ('DBMS_APPLICATION_INFO', 'PRODUCT_PRIVS', 'DUAL')
       ORDER BY username;
```

2 3 4 5 6

CAPTURE	USERNAME	OBJECT_OWNER	OBJECT_NAME	OBJ_PRIV

All_privs	JIM	HR	EMPLOYEES	DELETE
All_privs	TOM	SH	SALES	SELECT
All_privs	U1	U2	T1	SELECT
All_privs	U1	SYS	DBMS_OUTPUT	EXECUTE
ORA\$DEPENDENCY	U1	SYS	DBMS_OUTPUT	EXECUTE
ORA\$DEPENDENCY	U1	U2	T1	SELECT

6 rows selected.

SQL>

Notice the ORA\$DEPENDENCY capture that has been automatically generated. It contains all the privileges used for compiling dependency objects, such as the U2.T1 table accessed by U1 who creates the procedure selecting data from the U2.T1 table.

8. Display the system privileges used.

```
SQL> SELECT username, sys_priv FROM dba_used_sysprivs
      WHERE username IN ('JIM', 'TOM', 'U1', 'U2');
```

2

USERNAME	SYS_PRIV
TOM	CREATE SESSION
JIM	CREATE SESSION
U1	CREATE SESSION
U1	CREATE PROCEDURE

SQL>

9. Display the path of the privileges used if the privileges were granted to roles, and roles to users.

```
SQL> COL path FORMAT A32
```

```
SQL> SELECT username, obj_priv, object_name, path
      FROM dba_used_objprivs_path
      WHERE username IN ('TOM','JIM', 'U1', 'U2')
      AND object_name NOT IN
          ('DBMS_APPLICATION_INFO', 'PRODUCT_PRIVS', 'DUAL')
      ORDER BY username, object_name;
```

2 3 4

USERNA	OBJ_PRIV	OBJECT_NAME	PATH
JIM	DELETE	EMPLOYEES	GRANT_PATH('JIM', 'HR_MGR')
TOM	SELECT	SALES	GRANT_PATH('TOM', 'SALES_CLERK')
U1	EXECUTE	DBMS_OUTPUT	GRANT_PATH('PUBLIC')
U1	EXECUTE	DBMS_OUTPUT	GRANT_PATH('PUBLIC')
U1	SELECT	T1	GRANT_PATH('U1')

```

U1      SELECT      T1      GRANT_PATH('U1')

6 rows selected.

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. Use the following column formats.

```

COL username FORMAT A8
COL sys_priv FORMAT A8
COL obj_priv FORMAT A8

```

```

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_NAME  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
      TO hr_mgr_junior;

 2
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 shows that the `All_privs` capture has been stopped. Use the following column formats.

```
COL name FORMAT A14
COL type FORMAT A8
COL enabled FORMAT A2
COL roles FORMAT A26
COL context FORMAT A26
```

```
SQL> SELECT name, type, enabled, roles, context
       FROM dba_priv_captures;
```

NAME	TYPE	EN	ROLES
------	------	----	-------

CONTEXT			

ORA\$DEPENDENCY DATABASE N			
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
      FROM dba_unused_privs
      WHERE username IN ('JIM', 'TOM', 'U1', 'U2','KATE') ;
2      3
no rows selected

SQL>
```

Practice 11-3: Capturing 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 the `HR.EMPLOYEES` table and TOM who selects rows from the `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
```

```
Connected.
```

```
Procedure created.
```

```
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. Use the following column formats.

```
COL username FORMAT A10
COL object_owner FORMAT A12
COL object_name FORMAT A11
COL obj_priv FORMAT A10
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	OBJECT_OWNER	OBJECT_NAME	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. Use the following column formats.

```
COL username FORMAT A12
COL path FORMAT A32
COL object_name FORMAT A11
COL sys_priv FORMAT A10
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_NAME	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> SELECT name, type, enabled, roles, context			
FROM dba_priv_captures;			
2			
NAME	TYPE	EN	ROLES

CONTEXT			

ORA\$DEPENDENCY	DATABASE	N	
Role_privs	ROLE	N	ROLE_ID_LIST(120, 116)
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 11-4: Capturing 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 the `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

Connected.

Procedure created.

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> 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  OBJECT_OWNER OBJECT_NAME OBJ_PRIV   USED_ROLE
-----
TOM       SH            SALES      SELECT   SALES_CLERK

SQL>
```

6. Display the system privileges used.

```
SQL> SELECT sys_priv FROM dba_used_sysprivs
     WHERE  username='TOM' OR used_role='SALES_CLERK';

 2
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 11-5: Using INHERIT PRIVILEGES Privilege (Optional)

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 11-2.

Tasks

- The developer U1 creates an invoker's rights procedure that selects rows from the `U2.T1` table.

The user U1 is granted the `SELECT` privilege on the `U2.T1` table

- Connect as user U1 with password u1.

```
SQL> connect u1
Enter password: *****
Connected.
SQL>
```

- 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 with password kate and executes the procedure.

```
SQL> CONNECT kate
Enter password: *****
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> 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. Use the following column formats.

```
COL privilege FORMAT A20
COL type FORMAT A6
COL table_name FORMAT A10
COL grantee FORMAT A8
```

```
SQL> connect / as sysdba
Connected.
SQL> select PRIVILEGE, TYPE, TABLE_NAME, GRANTEE
       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 11-6: Using BEQUEATH Views (Optional)

Overview

In this practice you understand the different types of BEQUEATH views: the CURRENT_USER and the DEFINER views.

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 for
ORACLE_HOME=/u01/app/oracle/product/12.1.0/dbhome_1 is
/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

Enter password: *****

Connected to:
Oracle Database 12c Enterprise Edition Release 12.1.0.2.0 -
64bit Production
With the Partitioning, OLAP, Advanced Analytics, Real
Application Testing and Unified Auditing options

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
Enter password: *****
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 a 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
```

```
Enter password : *****
```

```
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 12: Oracle Data Redaction

Chapter 12

Practices for Lesson 12: Overview

Practices Overview

In the practice for this lesson, you use Oracle Data Redaction to redact values of shielded columns of the `HR.EMPLOYEES` table and also to change the default value for FULL redaction.

Practice 12-1: 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

Enter password: *****

Connected to:
Oracle Database 12c Enterprise Edition Release 12.1.0.2.0 -
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 Favieret                  9000
          110 Chen                      8200
          111 Sciarra                   7700
          112 Urman                     7800
          113 Popp                      6900

6 rows selected.

SQL>
```

- 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

PL/SQL procedure successfully completed.

SQL>
```

- 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. Use the following column formats.

```
COL object_owner FORMAT A12
COL object_name  FORMAT A12
COL policy_name  FORMAT A14
COL expression   FORMAT A12
COL enable       FORMAT A6
COL policy_description FORMAT A10
COL column_name  FORMAT A10
COL function_type FORMAT A17
COL function_parameters FORMAT A20
```



```
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> SELECT object_owner, object_name, column_name,
           function_type, function_parameters
       FROM redaction_columns;
```

2	3		
OBJECT_OWNER	OBJECT_NAME	COLUMN_NAME	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
Enter password: *****
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 12-2: 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 Favieret                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
Enter password: *****
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 12-3: Cleaning Up Redaction Policies

Overview

In this practice you clean up the redaction policy applied on the `HR.EMPLOYEES` table.

- 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
Enter password: *****
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>
```

- 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
```

```
$
```

Practice 12-4: Changing the Default Value for FULL Redaction (Optional)

Overview

In this practice, you use full redaction to redact the returned data to a fixed value.

You will modify the default value for full redaction of data of NUMBER data type to 10 for the commission percentage of all employees.

Tasks

1. Modify the default value to 10 for full redaction of the commission percentage of all employees.
 - a. Display the information from the data dictionary view before updating the default value.

```
$ sqlplus / AS SYSDBA

Connected to:
Oracle Database 12c Enterprise Edition Release 12.1.0.2.0 -
64bit Production
With the Partitioning, OLAP, Advanced Analytics, Real
Application Testing and Unified Auditing options

SQL> SELECT number_value FROM REDACTION_VALUES_FOR_TYPE_FULL;

NUMBER_VALUE
-----
              0

SQL>
```

- b. Modify the default value.

```
SQL> exec DBMS_REDACT.UPDATE_FULL_REDACTION_VALUES( -
          NUMBER_VAL => 10)

PL/SQL procedure successfully completed.

SQL>
```

- c. Display the information from the data dictionary view.

```
SQL> SELECT number_value FROM REDACTION_VALUES_FOR_TYPE_FULL;

NUMBER_VALUE
-----
              10

SQL>
```


- d. Add the `COMMISSION_PCT` column to the policy for full redaction using the following code.

```
BEGIN
  DBMS_REDACT.ADD_POLICY
    (object_schema => 'HR',
     object_name   => 'EMPLOYEES',
     policy_name   => 'EMPCOMM_POLICY',
     column_name   => 'COMMISSION_PCT',
     function_type => DBMS_REDACT.FULL,
     expression    => '1=1');
END;
```

```
/
SQL> BEGIN
  DBMS_REDACT.ADD_POLICY
    (object_schema => 'HR',
     object_name   => 'EMPLOYEES',
     policy_name   => 'EMPCOMM_POLICY',
     column_name   => 'COMMISSION_PCT',
     function_type => DBMS_REDACT.FULL,
     expression    => '1=1');
  END;
/
  2      3      4      5      6      7      8      9     10
PL/SQL procedure successfully completed.

SQL>
```

- e. The `REDACTION_COLUMNS` view shows the masking function defined on the `HR.EMPLOYEES` table. Use the following column formats.

```
COL object_owner FORMAT A12
COL object_name  FORMAT A12
COL policy_name  FORMAT A14
COL column_name  FORMAT A15
COL function_type FORMAT A17
COL function_parameters FORMAT A20
```

```
SQL> SELECT object_owner, object_name, column_name,
           function_type
        FROM redaction_columns;

  2      3
OBJECT_OWNER OBJECT_NAME  COLUMN_NAMEFUNCTION_TYPE
-----
HR            EMPLOYEES   COMMISSION_PCT  FULL REDACTION
```

```
SQL>
```

2. Display the values of the `COMMISSION_PCT` column of all employees.

```
SQL> CONNECT sh
```

```
Enter password: *****
```

```
Connected.
```

```
SQL> SELECT commission_pct, first_name FROM hr.employees  
       ORDER BY 1 DESC;
```

```
2
```

```

COMMISSION_PCT FIRST_NAME
-----
... rows deleted ...
           Shelley
           William
           0 John
           0 Allan
           0 Patrick
           0 Ellen
... rows deleted ...
           0 Sundar
           0 Charles
           0 Sundita
           0 Amit

83 rows selected.

SQL>

```

The result still displays the value 0.

3. After you modify a value, you must restart the database for it to take effect. If you only flush the buffer cache, the real value of the column will be displayed.

```

SQL> CONNECT / AS SYSDBA
Connected.
SQL> SHUTDOWN IMMEDIATE
Database closed.
Database dismounted.
ORACLE instance shut down.
SQL> STARTUP
ORACLE instance started.

Total System Global Area  503316480 bytes
Fixed Size                  2916056 bytes
Variable Size              272630056 bytes
Database Buffers           222298112 bytes
Redo Buffers                5472256 bytes
Database mounted.
Database opened.
SQL>

```

4. Display the values of the `COMMISSION_PCT` column of all employees.

```
SQL> CONNECT sh
Enter password: *****
Connected.
SQL> SELECT commission_pct, first_name FROM hr.employees
        ORDER BY 1 DESC;

      2
COMMISSION_PCT FIRST_NAME
-----
... rows deleted ...
              Shelley
              William
            10 John
            10 Allan
            10 Patrick
            10 Ellen
... rows deleted ...
            10 Sundar
            10 Charles
            10 Sundita
            10 Amit

83 rows selected.

SQL>
```

Notice that the default value is only applied to the values that are not `NULL`.

Question: When you updated the default value to a single, blank space for full redaction of the character data type, you did not restart the instance to get the right result.

Answer: The original default value is the same as the one you set. You just activated the default value for full redaction policies. Whereas, in this current case, the default value for the number data type is different from the original default value.

5. Drop the redaction policy.

```

BEGIN
  DBMS_REDACT.DROP_POLICY
  ( object_schema => 'HR',
    object_name   => 'EMPLOYEES',
    policy_name   => 'EMPCOMM_POLICY');
END;
/

```

```

SQL> CONNECT / AS SYSDBA
Connected.
SQL> BEGIN
      DBMS_REDACT.DROP_POLICY
      ( object_schema => 'HR',
        object_name   => 'EMPLOYEES',
        policy_name   => 'EMPCOMM_POLICY');
      END;
/
  2      3      4      5      6      7
PL/SQL procedure successfully completed.

SQL> SELECT object_owner, object_name, column_name,
           function_type
      FROM redaction_columns;
  2      3
no rows selected

SQL> SELECT * FROM redaction_policies;

no rows selected

SQL>

```

6. Reset the default values of full redaction for the NUMBER data type to the default.

```

SQL> exec SYS.DBMS_REDACT.UPDATE_FULL_REDACTION_VALUES( -
      NUMBER_VAL => 0)

PL/SQL procedure successfully completed.

SQL> SELECT varchar_value, number_value, date_value
      FROM REDACTION_VALUES_FOR_TYPE_FULL;
  2
V NUMBER_VALUE DATE_VALU
- - - - -

```

```
0 01-JAN-01
```

```
SQL>
```

7. *Question:* If you create another full redaction policy, which values are displayed for the full redacted columns of `HR.EMPLOYEES`?

Answer: The values displayed for the full redacted columns of `HR.EMPLOYEES` use the default values that you had set in the previous practices. They are still in effect until you restart the instance.

8. Restart the instance.

```
SQL> SHUTDOWN IMMEDIATE
```

```
Database closed.
```

```
Database dismounted.
```

```
ORACLE instance shut down.
```

```
SQL> STARTUP
```

```
ORACLE instance started.
```

```
Total System Global Area  501059584 bytes
```

```
Fixed Size                  2289968 bytes
```

```
Variable Size               264244944 bytes
```

```
Database Buffers            226492416 bytes
```

```
Redo Buffers                 8032256 bytes
```

```
Database mounted.
```

```
Database opened.
```

```
SQL>
```

9. Check that the values for the `SALARY` and `HIRE_DATE` columns are displayed without redaction.

```
SQL> CONNECT sh
Enter password: *****
Connected.
SQL> SELECT      first_name, last_name, salary, commission_pct,
                hire_date
      FROM      hr.employees
      WHERE      commission_pct is not null
      ORDER BY   4 DESC;
   2   3   4   5   6
FIRST_NAME  LAST_NAME          SALARY COMMISSION_PCT HIRE_DATE
-----
John        Russell              14000          .4 01-OCT-04
Allan       McEwen               9000          .35 01-AUG-04
...
Sundita     Kumar                6100          .1 21-APR-08
Amit        Banda                6200          .1 21-APR-08

35 rows selected.

SQL> EXIT
$
```


Practices for Lesson 13: Recovery Manager - New Features and Temporal History Enhancements

Chapter 13

Practices for Lesson 13: Overview

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 13-1: 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 for
ORACLE_HOME=/u01/app/oracle/product/12.1.0/dbhome_1 is
/u01/app/oracle
$ rman TARGET '/' AS SYSBACKUP''

Recovery Manager: Release 12.1.0.2.0 - Production on Wed May 21
03:40:14 2014

Copyright (c) 1982, 2014, 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;

new RMAN configuration parameters:
CONFIGURE CONTROLFILE AUTOBACKUP ON;
new RMAN configuration parameters are successfully stored

RMAN> BACKUP DATABASE PLUS ARCHIVELOG;

Starting backup at 21-05-2014 03:42:04
current log archived
```

```

allocated channel: ORA_DISK_1
channel ORA_DISK_1: SID=25 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=27 RECID=1 STAMP=848114840
input archived log thread=1 sequence=28 RECID=2 STAMP=848115725
channel ORA_DISK_1: starting piece 1 at 21-05-2014 03:42:06
channel ORA_DISK_1: finished piece 1 at 21-05-2014 03:42:09
piece
handle=/u01/app/oracle/fast_recovery_area/ORCL/backupset/2014_05_21/o1_mf_annnn_TAG20140521T034205_9qr84g4t_.bkp
tag=TAG20140521T034205 comment=NONE
channel ORA_DISK_1: backup set complete, elapsed time: 00:00:03
Finished backup at 21-05-2014 03:42:09

...
Starting Control File and SPFILE Autobackup at 21-05-2014
03:43:27
piece
handle=/u01/app/oracle/fast_recovery_area/ORCL/autobackup/2014_05_21/o1_mf_s_848115808_9qr870st_.bkp comment=NONE
Finished Control File and SPFILE Autobackup at 21-05-2014
03:43:31

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      810      SYSTEM              YES
/u01/app/oracle/oradata/orcl/system01.dbf
3     1000      SYSAUX                NO
/u01/app/oracle/oradata/orcl/sysaux01.dbf
4      145      UNDOTBS1              YES
/u01/app/oracle/oradata/orcl/undotbs01.dbf
5     1277      EXAMPLE              NO
/u01/app/oracle/oradata/orcl/example01.dbf
6        5      USERS                NO
/u01/app/oracle/oradata/orcl/users01.dbf

```

```

List of Temporary Files
=====
File Size(MB) Tablespace          Maxsize(MB) Tempfile Name
-----
1      197      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 05/21/2014
03:51:46
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
FORCE_FULL_DB_CACHING	VARCHAR2 (3)

RMAN> **EXIT**

 Recovery Manager complete.
 \$

Practice 13-2: 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. Execute the `$HOME/labs/FDA/test.sql` script to recreate the `HR.TEST_TABLE1` table and populate it with three rows.

```
$ sqlplus / as sysdba

Connected to:
Oracle Database 12c Enterprise Edition Release 12.1.0.2.0 -
64bit Production
With the Partitioning, OLAP, Advanced Analytics, Real
Application Testing and Unified Auditing options

SQL> @$HOME/labs/FDA/test.sql

Table dropped.

Table created.

1 row created.

1 row created.

1 row created.

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 21-05-2014 03:53:11
```



```
using target database control file instead of recovery catalog
allocated channel: ORA_DISK_1
channel ORA_DISK_1: SID=22 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 21-05-2014 03:53:12
channel ORA_DISK_1: finished piece 1 at 21-05-2014 03:53:13
piece
handle=/u01/app/oracle/fast_recovery_area/ORCL/backupset/2014_05_21/o1_mf_nnndf_TAG20140521T035312_9qr8s8kl_.bkp
tag=TAG20140521T035312 comment=NONE
channel ORA_DISK_1: backup set complete, elapsed time: 00:00:01
Finished backup at 21-05-2014 03:53:13

Starting Control File and SPFILE Autobackup at 21-05-2014
03:53:13
piece
handle=/u01/app/oracle/fast_recovery_area/ORCL/autobackup/2014_05_21/o1_mf_s_848116393_9qr8sbck_.bkp comment=NONE
Finished Control File and SPFILE Autobackup at 21-05-2014
03:53:16

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 12.1.0.2.0 -
64bit Production
With the Partitioning, OLAP, Advanced Analytics and Real
Application Testing options and Unified Auditing options

SQL> select timestamp_to_scn(current_timestamp)
        from v$database;

TIMESTAMP_TO_SCN(CURRENT_TIMESTAMP)
-----
                                2146235

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 /u01/app/oracle/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 2146235 AUXILIARY
DESTINATION '/u01/app/oracle/backup_test';

Starting recover at 21-05-2014 03:54:15
using target database control file instead of recovery catalog
current log archived
allocated channel: ORA_DISK_1
channel ORA_DISK_1: SID=27 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='DpB1'

initialization parameters used for automatic instance:

db_name=ORCL

db_unique_name=DpB1_pitr_ORCL

compatible=12.1.0.2.0

db_block_size=8192

db_files=200

diagnostic_dest=/u01/app/oracle

_system_trig_enabled=FALSE

sga_target=488M

processes=200

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	511705088 bytes
--------------------------	-----------------

Fixed Size	2917240 bytes
------------	---------------

Variable Size	155192456 bytes
---------------	-----------------

Database Buffers	348127232 bytes
------------------	-----------------

Redo Buffers	5468160 bytes
--------------	---------------

Automatic instance created

contents of Memory Script:

```
{
# set requested point in time
set until scn 2146235;
# 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 21-05-2014 03:55:36
allocated channel: ORA_AUX_DISK_1
channel ORA_AUX_DISK_1: SID=12 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/2014_05_21/o1_
mf_s_848116393_9qr8sbck_.bkp
channel ORA_AUX_DISK_1: piece
handle=/u01/app/oracle/fast_recovery_area/ORCL/autobackup/2014_0
5_21/o1_mf_s_848116393_9qr8sbck_.bkp tag=TAG20140521T035313
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_9qr8xvwn
_.ctl
Finished restore at 21-05-2014 03:55:40

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 2146235;
# 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 21-05-2014 03:55:48
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/2014_05_21/o1_
mf_nnndf_TAG20140521T034209_9qr84lhm_.bkp
channel ORA_AUX_DISK_1: piece
handle=/u01/app/oracle/fast_recovery_area/ORCL/backupset/2014_05
_21/o1_mf_nnndf_TAG20140521T034209_9qr84lhm_.bkp
tag=TAG20140521T034209
channel ORA_AUX_DISK_1: restored backup piece 1
channel ORA_AUX_DISK_1: restore complete, elapsed time: 00:00:35
Finished restore at 21-05-2014 03:56:23

```

```

datafile 1 switched to datafile copy
input datafile copy RECID=4 STAMP=848116583 file
name=/u01/app/oracle/backup_test/ORCL/datafile/o1_mf_system_9qr8
y4ks_.dbf
datafile 4 switched to datafile copy
input datafile copy RECID=5 STAMP=848116583 file
name=/u01/app/oracle/backup_test/ORCL/datafile/o1_mf_undotbs1_9q
r8y4m3_.dbf
datafile 3 switched to datafile copy
input datafile copy RECID=6 STAMP=848116583 file
name=/u01/app/oracle/backup_test/ORCL/datafile/o1_mf_sysaux_9qr8
y4go_.dbf

contents of Memory Script:
{
# set requested point in time
set until   scn 2146235;
# 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 21-05-2014 03:56:24
using channel ORA_AUX_DISK_1

starting media recovery

archived log for thread 1 with sequence 29 is already on disk as
file

```

```

/u01/app/oracle/fast_recovery_area/ORCL/archivelog/2014_05_21/o1_
_mf_1_29_9qr86xcc_.arc
archived log for thread 1 with sequence 30 is already on disk as
file
/u01/app/oracle/fast_recovery_area/ORCL/archivelog/2014_05_21/o1_
_mf_1_30_9qr8v80o_.arc
archived log file
name=/u01/app/oracle/fast_recovery_area/ORCL/archivelog/2014_05_
21/o1_mf_1_29_9qr86xcc_.arc thread=1 sequence=29
archived log file
name=/u01/app/oracle/fast_recovery_area/ORCL/archivelog/2014_05_
21/o1_mf_1_30_9qr8v80o_.arc thread=1 sequence=30
media recovery complete, elapsed time: 00:00:01
Finished recover at 21-05-2014 03:56:26

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_9qr8xvwn_.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      511705088 bytes

```

```

Fixed Size                2917240 bytes
Variable Size             159386760 bytes
Database Buffers          343932928 bytes
Redo Buffers               5468160 bytes

```

```

sql statement: alter system set control_files =
''/u01/app/oracle/backup_test/ORCL/controlfile/o1_mf_9qr8xvwn_.c
tl'' comment= ''RMAN set'' scope=spfile

```

Oracle instance shut down

```

connected to auxiliary database (not started)
Oracle instance started

```

```

Total System Global Area    511705088 bytes

```

```

Fixed Size                2917240 bytes
Variable Size             159386760 bytes
Database Buffers          343932928 bytes
Redo Buffers               5468160 bytes

```

```

sql statement: alter database mount clone database

```

contents of Memory Script:

```

{
# set requested point in time
set until scn 2146235;
# 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 21-05-2014 03:57:46


```

allocated channel: ORA_AUX_DISK_1
channel ORA_AUX_DISK_1: SID=12 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/DPBL_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/2014_05_21/o1_
mf_nnndf_TAG20140521T035312_9qr8s8kl_.bkp
channel ORA_AUX_DISK_1: piece
handle=/u01/app/oracle/fast_recovery_area/ORCL/backupset/2014_05_
21/o1_mf_nnndf_TAG20140521T035312_9qr8s8kl_.bkp
tag=TAG20140521T035312
channel ORA_AUX_DISK_1: restored backup piece 1
channel ORA_AUX_DISK_1: restore complete, elapsed time: 00:00:01
Finished restore at 21-05-2014 03:57:47

datafile 6 switched to datafile copy
input datafile copy RECID=8 STAMP=848116668 file
name=/u01/app/oracle/backup_test/DPBL_PITR_ORCL/datafile/o1_mf_u
sers_9qr91tv3_.dbf

contents of Memory Script:
{
# set requested point in time
set until scn 2146235;
# 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 21-05-2014 03:57:48
using channel ORA_AUX_DISK_1

```

```
starting media recovery
```

```
archived log for thread 1 with sequence 30 is already on disk as file
```

```
/u01/app/oracle/fast_recovery_area/ORCL/archivelog/2014_05_21/o1_mf_1_30_9qr8v80o_.arc
```

```
archived log file
```

```
name=/u01/app/oracle/fast_recovery_area/ORCL/archivelog/2014_05_21/o1_mf_1_30_9qr8v80o_.arc thread=1 sequence=30
```

```
media recovery complete, elapsed time: 00:00:01
```

```
Finished recover at 21-05-2014 03:57:50
```

```
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_DpBl_qoFx":
```

```
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$_ORDERS_QUEUE_
TABLE_S','IX',1,1,'12.01.00.02.00',newblock)
ORA-00376: file 5 cannot be read at this time
ORA-01110: data file 5:
'/u01/app/oracle/oradata/orcl/example01.dbf'
ORA-06512: at "SYS.DBMS_TRANSFORM_EXIMP", line 197
ORA-06512: at line 1
ORA-06512: at "SYS.DBMS_METADATA", line 10261
ORA-39127: unexpected error from call to export_string
:=SYS.DBMS_TRANSFORM_EXIMP.INSTANCE_INFO_EXP('AQ$_STREAMS_QUEUE_
TABLE_S','IX',1,1,'12.01.00.02.00',newblock)
ORA-00376: file 5 cannot be read at this time
ORA-01110: data file 5:
'/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 10261
EXPDP> . . exported "HR"."TEST_TABLE1"
6.015 KB          3 rows
EXPDP> Master table "SYS"."TSPITR_EXP_DpBl_qoFx" successfully
loaded/unloaded
EXPDP>
*****
*****
EXPDP> Dump file set for SYS.TSPITR_EXP_DpBl_qoFx is:
EXPDP> /u01/app/oracle/backup_test/tspitr_DpBl_17830.dmp
EXPDP> Job "SYS"."TSPITR_EXP_DpBl_qoFx" completed with 2
error(s) at Wed May 21 03:58:28 2014 elapsed 0 00:00:20
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_DpBl_Cord" successfully
loaded/unloaded
IMPDP> Starting "SYS"."TSPITR_IMP_DpBl_Cord":

```

Copyright © 2014, Oracle and/or its affiliates. All rights reserved.

```

IMPDP> Processing object type TABLE_EXPORT/TABLE/TABLE
IMPDP> Processing object type TABLE_EXPORT/TABLE/TABLE_DATA
IMPDP> . . imported "HR"."TEST_TABLE1"
6.015 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_DpBl_Cord" successfully
completed at Wed May 21 03:58:42 2014 elapsed 0 00:00:05
Import completed

Removing automatic instance
Automatic instance removed
auxiliary instance file
/u01/app/oracle/backup_test/ORCL/datafile/o1_mf_temp_9qr8zbtm_.t
mp deleted
auxiliary instance file
/u01/app/oracle/backup_test/DPBL_PITR_ORCL/onlinelog/o1_mf_3_9qr
921dy_.log deleted
auxiliary instance file
/u01/app/oracle/backup_test/DPBL_PITR_ORCL/onlinelog/o1_mf_2_9qr
91zrn_.log deleted
auxiliary instance file
/u01/app/oracle/backup_test/DPBL_PITR_ORCL/onlinelog/o1_mf_1_9qr
91yhd_.log deleted
auxiliary instance file
/u01/app/oracle/backup_test/DPBL_PITR_ORCL/datafile/o1_mf_users_
9qr91tv3_.dbf deleted
auxiliary instance file
/u01/app/oracle/backup_test/ORCL/datafile/o1_mf_sysaux_9qr8y4go_
.dbf deleted
auxiliary instance file
/u01/app/oracle/backup_test/ORCL/datafile/o1_mf_undotbs1_9qr8y4m
3_.dbf deleted
auxiliary instance file
/u01/app/oracle/backup_test/ORCL/datafile/o1_mf_system_9qr8y4ks_
.dbf deleted
auxiliary instance file
/u01/app/oracle/backup_test/ORCL/controlfile/o1_mf_9qr8xvwn_.ctl
deleted
auxiliary instance file tspitr_DpBl_17830.dmp deleted
Finished recover at 21-05-2014 03:58:45

RMAN>

```

5. Check that the table is fully recovered.

```
RMAN> select * from hr.test_table1;
```

NUM	NAME	NOW
1	First test row	21-MAY-14
2	Second test row	21-MAY-14
3	Third test row	

```
RMAN> exit
$
```


Practices for Lesson 14: Real-Time Database Operation Monitoring

Chapter 14

Practices for Lesson 14: 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 14-1: 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 12.1.0.2.0 -
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 12.1.0.2.0
- Production
Start Date                          21-MAY-2014 03:14:21
Uptime                              3 days 19 hr. 35 min. 2 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/tnslnr/youserver/listener/alert/log.xml
Listening Endpoints Summary...
(DESCRIPTION=(ADDRESS=(PROTOCOL=tcp) (HOST=<Your
hostname>) (PORT=1521)))
(DESCRIPTION=(ADDRESS=(PROTOCOL=ipc) (KEY=EXTPROC1521)))
(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 12.1.0.2.0 -
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.setHTTPPort(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
```

```
...
```

```
STATUS of the LISTENER
```

```
-----
```

```
Alias                LISTENER
Version              TNSLSNR for Linux: Version 12.1.0.2.0
- Production
Start Date           21-MAY-2014 04:24:21
Uptime               0 days 00 hr. 1 min. 2 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 <http://localhost:5500/em>.

Practice 14-2: 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 for
ORACLE_HOME=/u01/app/oracle/product/12.1.0/dbhome_1 is
/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 12.1.0.2.0 on Wed May 21 04:02:23 2014

Copyright (c) 1982, 2014, Oracle. All rights reserved.

Last Successful login time: Wed May 21 2014 04:01:38 +00:00

Connected to:
Oracle Database 12c Enterprise Edition Release 12.1.0.2.0 -
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
12.1.0.2.0 - 64bit Production
```

With the Partitioning, OLAP, Advanced Analytics, Real Application Testing and Unified Auditing options

\$

- Execute the \$HOME/labs/DBOps/DBOps1.sql script. The script starts a first database operation. The database operation is named 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
```

```
Enter password: *****
```

```
Connected to:
```

```
Oracle Database 12c Enterprise Edition Release 12.1.0.2.0 -
64bit Production
```

```
With the Partitioning, OLAP, Advanced Analytics, Real
Application Testing and Unified Auditing options
```

```
SQL> @DBOps1.sql
```

```
PL/SQL procedure successfully completed.
```

```
...
```

```
6889 rows selected.
```

```
...
```

```
27 rows selected.
```

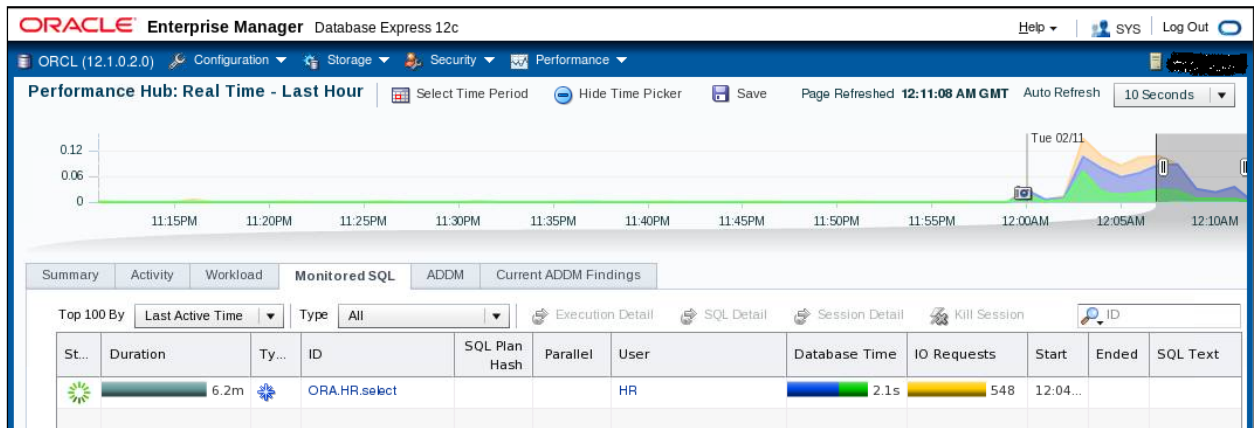
```
...
```

```
990025 rows selected.
```

```
PL/SQL procedure successfully completed.
```

```
SQL>
```

- 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.



Note: You could also use Enterprise Manager Cloud Control.

- In another terminal window, 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  503316480 bytes
Fixed Size                 2916056 bytes
Variable Size             272630056 bytes
Database Buffers          222298112 bytes
Redo Buffers              5472256 bytes
Database mounted.
Database opened.
SQL> EXIT
$
```

- 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 4
```

```
Copyright (c) 1996, 2014 Oracle Corporation. All rights reserved.
```

```
Starting Oracle Management Server...
```

```
Starting WebTier...
```

```
WebTier Successfully Started
```

```
Oracle Management Server Successfully Started
```

```
Oracle Management Server is Up
```

```
$
```

- 3) Use <https://localhost:7802/em> to get the Enterprise Manager Cloud Control console appear, enter **sysman** in the User Name field and **orac1e123** 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 a previous practice. 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 by executing the `$HOME/labs/DBOps/DBOps2.sql` script. The script names 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
```

```
Enter password: *****
```

```
Connected to:
```



```
Oracle Database 12c Enterprise Edition Release 12.1.0.2.0 -  
64bit Production
```

```
With the Partitioning, OLAP, Advanced Analytics, Real  
Application Testing and Unified Auditing options
```

```
SQL> @DBOps2.sql
```

```
PL/SQL procedure successfully completed.
```

```
...
```

```
990025 rows selected.
```

```
...
```

```
...
```

```
PL/SQL procedure successfully completed.
```

```
SQL>
```

Some statements take long and you can terminate them after sometime.

- 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.

Stat...	Duration	Ty...	ID	SQL Plan Hash	User	Parallel	Database Ti...	IO Requests	Start	Ended
	3.8m		ORA.SYSTEM.se...		SYSTEM		2.8s	128	12:39:58 ...	
	38.8m		ORA.HR.select		HR		2.1s	548	12:04:54 ...	
	5.0s		1mmqjwktwdbn	17722011...	SYS		5.3s	2,896	Mon Feb ...	Mon Fe
	47.0s		cnphq355f5rah		SYS		46.9s	11K	Mon Feb ...	Mon Fe

- Execute the \$HOME/labs/DBOps/DBOps3.sql script to start a third database operation in another session. The script names 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

Enter password: *****

Connected to:
Oracle Database 12c Enterprise Edition Release 12.1.0.2.0 -
64bit Production
With the Partitioning, OLAP, Advanced Analytics, Real
Application Testing and Unified Auditing options
```

```
SQL> @DBOps3.sql

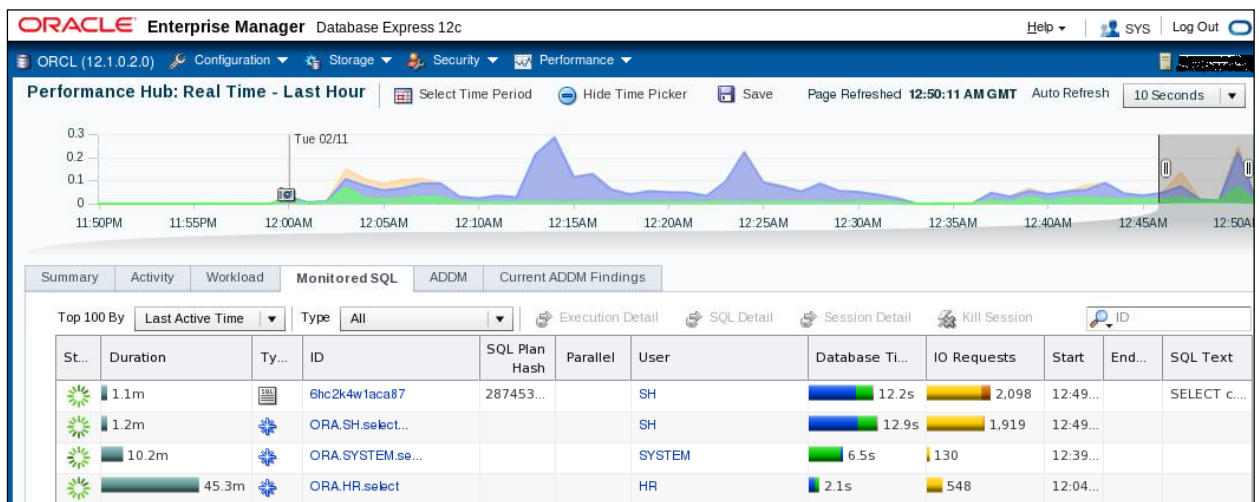
PL/SQL procedure successfully completed.

...

PL/SQL procedure successfully completed.

SQL> EXIT
$
```

8. View the new database operation using Enterprise Manager Database Express or Enterprise Manager Cloud Control.



Practice 14-3: 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. Execute the `$HOME/labs/DBOps/DBOps4.sql` script to start a new database operation. The script names 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
```

```
Enter password: *****
```

```
Connected to:
```

```
Oracle Database 12c Enterprise Edition Release 12.1.0.2.0 -
64bit Production
```

```
With the Partitioning, OLAP, Advanced Analytics, Real
Application Testing and Unified Auditing options
```

```
SQL> @DBOps4.sql
```

```
PL/SQL procedure successfully completed.
```

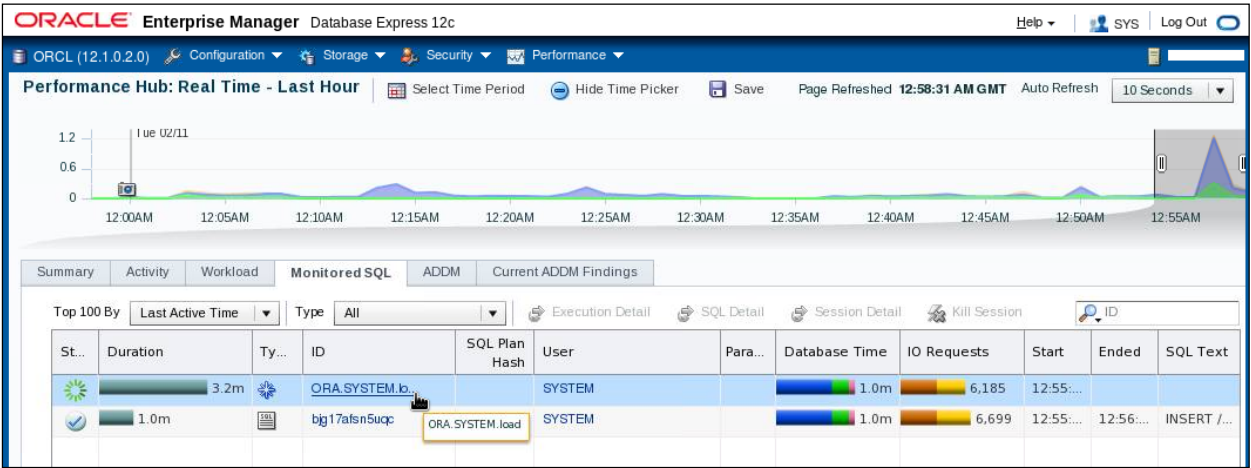
```
999 rows created.
```

```
919842 rows created.
```

```
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.



Practice 14-4: 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.

```
SQL> REVOKE SELECT ANY DICTIONARY FROM hr, sh;
```

```
Revoke succeeded.
```

```
SQL> EXIT
```

```
$
```