



Integrated Cloud Applications & Platform Services

# Oracle Database 12c R2: New Features for 12c R1 Administrators

Activity Guide – Volume II

D93517GC10

Edition 1.0 | December 2016 | D98180

Learn more from Oracle University at [education.oracle.com](http://education.oracle.com)

The Oracle logo, consisting of the word "ORACLE" in a white, sans-serif font with a registered trademark symbol, set against a red rectangular background.

**Copyright © 2016, 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, Mark Fuller and James Spiller

#### **Technical Contributors and Reviewers**

Krishnanjani Chitta, Randall Richeson, Gerlinde Frenzen, Sailaja Pasupuleti, Anita Mukundan, Harald van Breederode, Jim Stenoish and Branislav Valny

**This book was published using: Oracle Tutor**

# Table of Contents

<b>Course Practice Environment: Security Credentials .....</b>	<b>I-1</b>
Course Practice Environment: Security Credentials.....	I-2
<b>Practices for Lesson 1: Introduction.....</b>	<b>1-1</b>
Practices for Lesson 1: Overview.....	1-2
Practice 1-1: Starting Enterprise Manager Database Express .....	1-3
Practice 1-2: Configuring Enterprise Manager Cloud Control Targets .....	1-8
Practice 1-3: Creating and Testing New Named Credentials .....	1-13
<b>Practices for Lesson 2: Application Containers and Applications.....</b>	<b>2-1</b>
Practices for Lesson 2: Overview.....	2-2
Practice 2-1: Installing an Application in an Application Container.....	2-3
Practice 2-2: Upgrading an Application .....	2-12
Practice 2-3: Opening and Closing Application PDBs.....	2-18
<b>Practices for Lesson 3: Security in CDB, Application Containers, and PDBs.....</b>	<b>3-1</b>
Practices for Lesson 3: Overview.....	3-2
Practice 3-1: Managing Common Entities and Shared Data in Application Containers.....	3-3
Practice 3-2: Querying Data Across Application PDBs in CDB .....	3-34
Practice 3-3: Applying Recorded Statements in Application PDBs .....	3-43
Practice 3-4: Managing PDB Lockdown Profiles.....	3-51
Practice 3-5: Auditing Operations in PDBs.....	3-56
<i>Practice 3-6: Protecting Application Common Objects with DV Common Realms (Optional) .....</i>	<i>3-68</i>
Practice 3-7: Unplugging and Plugging Encrypted PDBs (Optional) .....	3-80
<b>Practices for Lesson 4: Creation of PDBs Using New Methods .....</b>	<b>4-1</b>
Practices for Lesson 4: Overview.....	4-2
Practice 4-1: Unplugging and Plugging Application Containers Using Archive File (Optional).....	4-3
<i>Practice 4-2: Renaming Services (Optional) .....</i>	<i>4-10</i>
<i>Practice 4-3: Converting a Regular PDB to an Application PDB (Optional) .....</i>	<i>4-14</i>
Practice 4-4: Cloning Remote PDBs in Hot Mode and Automatic Refreshing.....	4-20
Practice 4-5: Relocating PDBs.....	4-30
Practice 4-6: Querying Data Across CDBs Using Proxy PDBs .....	4-37
Practice 4-7: Dropping Unnecessary PDBs .....	4-45
<b>Practices for Lesson 5: Recovery and Flashback of PDBs .....</b>	<b>5-1</b>
Practices for Lesson 5: Overview.....	5-2
Practice 5-1: Recovering from Essential PDB Datafiles Damage.....	5-3
Practice 5-2: Flashing Back an Application Container from the Loss of Application Common Users (Optional) .....	5-15
Practice 5-3: Flashing Back Using Restore Points (Optional) .....	5-29
<b>Practices for Lesson 6: Managing Performance in CDBs and PDBs .....</b>	<b>6-1</b>
Practices for Lesson 6: Overview.....	6-2
Practice 6-1: Monitoring Performance at CDB and PDB Levels.....	6-3
Practice 6-2: Getting Performance Recommendations at CDB and PDB Levels .....	6-8
Practice 6-3: Monitoring and Tuning SQL Executions at PDB Level.....	6-20
Practice 6-4 Using Performance Profiles to Limit CPU Between Application PDBs.....	6-35
<b>Practices for Lesson 7: Upgrade and Other Operations in CDBs and PDBs.....</b>	<b>7-1</b>
Practices for Lesson 7: Overview.....	7-2
Practice 7-1: Upgrading a 12.1.0.2 Regular PDB to a 12.2.0.1 Application PDB .....	7-3
Practice 7-2: Plugging Remote PDBs Through XTTS.....	7-28

<i>Practice 7-3: Upgrading a 12.1.0.2 CDB to a 12.2.0.1 CDB (Optional)</i> .....	7-38
<b>Practices for Lesson 9: Managing Privileges and User Profiles</b> .....	<b>9-1</b>
Practices for Lesson 9: Overview .....	9-2
Practice 9-1: Migrating the Password File .....	9-3
Practice 9-2: Managing Administrative Privileged Users Connections .....	9-4
<i>Practice 9-3: Using the INHERIT REMOTE PRIVILEGES Privilege (Optional)</i> .....	9-8
Practice 9-4: Using Privilege Analysis Runs and CBAC Roles .....	9-11
<b>Practices for Lesson 10: Auditing</b> .....	<b>10-1</b>
Practices for Lesson 10: Overview .....	10-2
Practice 10-1: Auditing Users Using Roles .....	10-3
<i>Practice 10-2: Auditing Tables with VPD Policies (Optional)</i> .....	10-9
<b>Practices for Lesson 11: Oracle Data Redaction</b> .....	<b>11-1</b>
Practices for Lesson 11: Overview .....	11-2
Practice 11-1: Using Data Redaction Formats in Data Redaction Policies .....	11-3
Practice 11-2: Managing Policy Expressions .....	11-12
<b>Practices for Lesson 12: Data Encryption</b> .....	<b>12-1</b>
Practices for Lesson 12: Overview .....	12-2
Practice 12-1: Encrypting Essential and Temporary Tablespaces .....	12-3
Practice 12-2: Decrypting Existing Tablespaces Online in PDBs .....	12-6
<b>Practices for Lesson 13: Transparent Sensitive Data Protection</b> .....	<b>13-1</b>
Practices for Lesson 13: Overview .....	13-2
Practice 13-1: Managing TSDP Policies with Unified Auditing Settings .....	13-3
<i>Practice 13-2: Managing TSDP Policies with FGA Settings (Optional)</i> .....	13-10
<i>Practice 13-3: Managing TSDP Policies with TDE Settings (Optional)</i> .....	13-15
<b>Practices for Lesson 14: Data Availability</b> .....	<b>14-1</b>
Practices for Lesson 14: Overview .....	14-2
Practice 14-1: Recovering Database Until Available Redo .....	14-3
Practice 14-2: Recovering Using New Commands .....	14-13
Practice 14-3: Recovering Tables Across Schemas .....	14-19
Practice 14-4: Moving and Compressing Tables Online .....	14-28
<b>Practices for Lesson 15: Oracle Data Pump, SQL*Loader and External Tables</b> .....	<b>15-1</b>
Practices for Lesson 15: Overview .....	15-2
Practice 15-1: Querying Partitioned External Tables .....	15-3
<b>Practices for Lesson 16: In-Memory Column Store</b> .....	<b>16-1</b>
Practices for Lesson 16: Overview .....	16-2
Practice 16-1: Configuring IM Column Store .....	16-3
Practice 16-2: Optimizing Queries by Using Join Groups .....	16-13
Practice 16-3: Optimizing Queries by Capturing Expressions in IM Column Store .....	16-16
Practice 16-4: Evicting Tables of IM Column Store Based on Heat Map Statistics .....	16-24
<b>Practices for Lesson 17: SQL Tuning Enhancements</b> .....	<b>17-1</b>
Practices for Lesson 17: Overview .....	17-2
Practice 17-1: Configuring SPM to Capture Specific Statement Plans .....	17-3
Practice 17-2: Improving Statistics Gathering Quality .....	17-8
Practice 17-3: Disabling Database Triggers During SPA Task Execution .....	17-29
<b>Practices for Lesson 18: Resource Manager and Other Enhancements</b> .....	<b>18-1</b>
Practices for Lesson 18: Overview .....	18-2
Practice 18-1: Avoiding Excessive PGA Memory Usage .....	18-3

Practice 18-2: Using Cursor Invalidations with DDL Statements.....	18-7
Practice 18-3: Using Advanced Index Compression .....	18-13
Practice 18-4: Creating Real-Time MVs with OQC .....	18-18
Practice 18-5: Creating On-Statement MVs on MJVs (Optional).....	18-23
<b>Practices for Lesson 19: Partitioning Enhancements .....</b>	<b>19-1</b>
Practices for Lesson 19: Overview.....	19-2
Practice 19-1: Creating Auto-list Partitioned Tables.....	19-3
Practice 19-2: Converting Manual Partitioned Tables to Automatic Partitioned Tables.....	19-7
Practice 19-3: Partitioning Tables with Multi-column List Partitioning .....	19-10
Practice 19-4: Protecting Partitions Data from DML.....	19-13
Practice 19-5: Applying Filter Conditions to Partition Operation.....	19-17
<b>Practices for Lesson 20: Real-Time Database Operation Monitoring .....</b>	<b>20-1</b>
Practices for Lesson 20: Overview.....	20-2
Practice 20-1: Monitoring Sessions.....	20-3
<b>Practices for Lesson 21: Diagnosability .....</b>	<b>21-1</b>
Practices for Lesson 21: Overview.....	21-2
Practice 21-1: Using ADR Advisor to Set Appropriate ADR Target Size.....	21-3
Practice 21-2: Configuring TFA Collector and Analyzing Collections.....	21-9
<b>Practices for Lesson 22: Oracle Database Public Cloud Services .....</b>	<b>22-1</b>
Practices for Lesson 22: Overview.....	22-2
Practice 22-1: Creating a Database Deployment .....	22-3
Practice 22-2: Connecting to the Database Deployment Compute Node.....	22-6
Practice 22-3: Backing Up the Database Deployment .....	22-8
Practice 22-4: Monitoring the Database Deployment.....	22-30
<b>Practices for Lesson 23: SQL*Plus and SQLcl.....</b>	<b>23-1</b>
Practices for Lesson 23: Overview.....	23-2
Practice 23-1: Using SQL*Plus and SQLcl Commands .....	23-3
Practice 23-2: Identifying Nonconvertible Columns.....	23-20



## **Practices for Lesson 9: Managing Privileges and User Profiles**

### **Chapter 9**

## Practices for Lesson 9: Overview

---

### Practices Overview

In these practices, you will handle new features related to security.

- Migrate the password file to 12.2 format.
- Manage administrative privileged users connections.
- Lock inactive common users.
- Allow a definer's rights procedure owner to perform database operations on a remote database.
- Compare two run results of privilege analysis for roles, including CBAC roles.
- Capture privileges for all privileges used by users and activate PL/SQL compilation capture (*Optional*).

Before starting the practice, execute the `$HOME/labs/admin/glogin_9.sh`, `$HOME/labs/APP/cleanup_CDB1.sh`, and `$HOME/labs/APP/cleanup_apps.sh` shell scripts. The first one sets formatting for all columns selected in queries, the second one drops `cdb1`, and the last one drops all application PDBs in `ORCL`.

**Be careful to execute the `$HOME/labs/APP/cleanup_CDB1.sh` script only if the upgrade operation has completed in Practice 7.**

```
$ $HOME/labs/admin/glogin_9.sh
$ $HOME/labs/APP/cleanup_CDB1.sh
...
$ $HOME/labs/APP/cleanup_apps.sh
...
$
```



## Practice 9-1: Migrating the Password File

---

### Overview

In this practice, you will migrate the 12.1 format password file of ORCL to the new 12.2 format.

### Tasks

1. Before converting the ORCL password file from 12.1 format to 12.2 format, check if it is still under 12.1 format.

**Note:** format=12 refers to a 12.1 format.

```
$ . oraenv
ORACLE_SID = [ORCL] ? ORCL
The Oracle base has been set to /u01/app/oracle
$ orapwd describe file=$ORACLE_HOME/dbs/orapwORCL
Password file Description : format=12
$
```

2. Convert the ORCL password file from 12.1 format to 12.2 format.

```
$ cd $ORACLE_HOME/dbs
$ mv orapwORCL orapwORCL.12
$
$ orapwd file=orapwORCL input_file=orapwORCL.12 format=12.2
$
$ orapwd describe file=orapwORCL
Password file Description : format=12.2
$
```

3. Verify that you can still connect to ORCL using the password authentication.

```
$ sqlplus sys@ORCL AS SYSDBA

Enter password: *****

Connected to:
SQL> EXIT
$
```

## Practice 9-2: Managing Administrative Privileged Users Connections

### Overview

In this practice, you will secure administrative users connections by enforcing the associated profile's password limits.

### Tasks

1. Create a profile for c##junior\_dba in ORCL so that the account will be locked for one day after two consecutive unsuccessful login attempts.

```
$ sqlplus / AS SYSDBA
Connected.
SQL> CREATE USER c##junior_dba IDENTIFIED BY oracle_4U
        CONTAINER=ALL;

2
User created.

SQL> GRANT sysdba, create session TO c##junior_dba
CONTAINER=ALL;

Grant succeeded.

SQL> SELECT username, account_status, password_profile,
        last_login, common, con_id
        FROM v$pwfile_users WHERE sysdba = 'TRUE';

2      3
USERNAME      ACCOUNT_STATUS PASSWORD_PROFILE
-----
LAST_LOGIN                                COM CON_ID
-----
SYS              OPEN
24-MAR-16 12.30.44.0000000000 AM +00:00 YES      0

C##JUNIOR_DBA OPEN              DEFAULT
                                YES      0

SQL> CREATE PROFILE c##prof_junior_dba
        LIMIT failed_login_attempts 2 password_lock_time 1
        CONTAINER=all;

2      3
Profile created.

SQL> ALTER USER c##junior_dba PROFILE c##prof_junior_dba
        CONTAINER=ALL;
```

```

2
User altered.

SQL> CONNECT c##junior_dba@ORCL AS SYSDBA
Enter password: *****
Connected.
SQL> SELECT username, account_status, password_profile,
           last_login, common, con_id
       FROM v$pwfile_users WHERE sysdba = 'TRUE';
2      3

USERNAME          ACCOUNT_STATUS PASSWORD_PROFILE
-----
LAST_LOGIN                                COM CON_ID
-----
SYS              OPEN
24-MAR-16 12.30.44.0000000000 AM +00:00 YES      0

C##JUNIOR_DBA OPEN                      C##PROF_JUNIOR_DBA
24-MAR-16 12.32.10.0000000000 AM +00:00 YES      0

SQL>

```

2. The junior DBA does not remember his password and, for security purposes, no one else knows it.

```

SQL> CONNECT c##junior_dba@ORCL AS SYSDBA
Enter password:
ERROR:
ORA-01017: invalid username/password; logon denied

SQL> CONNECT c##junior_dba@ORCL AS SYSDBA
Enter password:
ERROR:
ORA-01017: invalid username/password; logon denied

SQL> CONNECT c##junior_dba@ORCL AS SYSDBA
Enter password:
ERROR:
ORA-28000: the account is locked

SQL>

```

*Q/ Can you have this behavior in Oracle Database 12.1?*

**A/ No. An administrative privileged user can be under profile limits only in Oracle Database 12.2.**

3. Unlock the junior DBA's account and tell him to remember his password.

```
SQL> CONNECT / AS SYSDBA
Connected.
SQL> ALTER USER c##junior_dba IDENTIFIED BY oracle_4U ACCOUNT
UNLOCK;

User altered.

SQL> CONNECT c##junior_dba@ORCL AS SYSDBA
Enter password: *****
Connected.
SQL>
```

4. Your security procedures now certify that if the c##junior\_dba common user account does not log on for 1 minute (for the purpose of the practice, you will not wait for thirty days), the associated user account will be automatically locked. The account needs to be unlocked by an administrator before it is available for use again.

```
SQL> CONNECT / AS SYSDBA
Connected.
SQL> SELECT distinct profile, limit FROM CDB_PROFILES
      WHERE resource_name = 'INACTIVE_ACCOUNT_TIME';

 2
PROFILE                                LIMIT
-----
C##PROF_JUNIOR_DBA                     DEFAULT
ORA_STIG_PROFILE                       35
DEFAULT                                UNLIMITED

SQL> SELECT distinct profile, limit FROM CDB_PROFILES
      WHERE resource_name = 'INACTIVE_ACCOUNT_TIME'
      AND   profile = 'DEFAULT';

 2   3
PROFILE                                LIMIT
-----
DEFAULT                                UNLIMITED

SQL> ALTER PROFILE c##prof_junior_dba
      LIMIT inactive_account_time 1 CONTAINER=all;

 2   3
ALTER PROFILE c##prof_junior_dba
*
ERROR at line 1:
```

```
ORA-02377: invalid profile limit INACTIVE_ACCOUNT_TIME
```

```
SQL>
```

*Q/ Why is the limit not updatable?*

***A/ Use the simple troubleshooting tool:***

```
$ oerr ora 2377
```

```
02377, 00000, "invalid profile limit %s"
```

```
// *Cause: A value of 0 or lower was specified for the limit.
```

```
// *Action: Specify a limit greater than 0. For password  
profile parameters, some additional restrictions apply:
```

```
// * For the INACTIVE_ACCOUNT_TIME profile  
parameter, the specified limit cannot be less than 15 days.
```

```
// * For the PASSWORD_GRACE_TIME profile parameter,  
0 is allowed as a permissible value.
```

```
SQL> ALTER PROFILE c##prof_junior_dba  
LIMIT inactive_account_time 15 CONTAINER=all;
```

```
2
```

```
Profile altered.
```

```
SQL> SELECT distinct profile, limit FROM CDB_PROFILES  
WHERE resource_name = 'INACTIVE_ACCOUNT_TIME';
```

```
2
```

PROFILE	LIMIT
ORA_STIG_PROFILE	35
DEFAULT	UNLIMITED
C##PROF_JUNIOR_DBA	15

```
SQL>
```

**Note:** Unfortunately, because the limit cannot be set under 15 days, you cannot test it within the course time.

```
SQL> DROP PROFILE c##prof_junior_dba CASCADE;
```

```
Profile dropped.
```

```
SQL> DROP USER c##junior_dba CASCADE;
```

```
User dropped.
```

```
SQL>
```

## Practice 9-3: Using the INHERIT REMOTE PRIVILEGES Privilege (Optional)

---

### Overview

In this practice, you will use new system privileges to allow definer's rights procedures to execute database operations on a remote database when logged in as the definer rights procedure user through a current user database link.

### Tasks

1. Before starting the practice, execute the `$HOME/labs/SEC/setup_users.sh` shell script. The script creates `pdb_orcl` in `ORCL`, `pdb2` in `cdb2`, and the `dev` user that develops procedures in `pdb2`. These procedures execute in `pdb2` and access tables in `pdb_orcl`.

```
$ $HOME/labs/SEC/setup_users.sh
...
$
```

2. Create a database link in `pdb_orcl` to access tables in `pdb2`. The creator of the procedure is not the owner of the procedure.

```
$ . oraenv
ORACLE_SID = [ORCL] ? cdb2
The Oracle base has been set to /u01/app/oracle
$ sqlplus dba_junior@pdb2
Enter password: *****
Connected.
SQL> DROP PUBLIC DATABASE LINK pdb_orcl_link;

Database link dropped.

SQL> CREATE PUBLIC DATABASE LINK pdb_orcl_link USING 'pdb_orcl';

Database link created.

SQL> CREATE OR REPLACE PROCEDURE dev.proc1 (CODE in varchar2)
      AS v_code number;
      BEGIN
          SELECT code INTO v_code FROM u1.t1@pdb_orcl_link;
          dbms_output.put_line('Code is: '||v_code);
      END PROC1;
/
  2      3      4      5      6      7
Warning: Procedure created with compilation errors.

SQL>
```

*Q/ Why does the procedure creation fail?*

**A/ Show the errors.**

```
SQL> SHOW errors
Errors for PROCEDURE DEV.PROC1:

LINE/COL
-----
ERROR
-----
0/0
ORA-04052: error occurred when looking up remote object
U1.T1@PDB_ORCL_LINK
ORA-00604: error occurred at recursive SQL level 1
ORA-25433: User DEV does not have INHERIT REMOTE PRIVILEGES
privilege on connected user DBA_JUNIOR.

SQL>
```

3. According to the error message, allow the current user to use a current user database link from within the definer's rights procedure.

```
SQL> CONNECT system@pdb2
Enter password: *****
Connected.
SQL> GRANT inherit remote privileges ON USER dba_junior TO dev;

Grant succeeded.

SQL> CONNECT dba_junior@pdb2
Enter password: *****
Connected.
SQL> CREATE OR REPLACE PROCEDURE dev.procl (CODE in varchar2)
      AS v_code number;
      BEGIN
          SELECT code INTO v_code FROM u1.t1@pdb_orcl_link;
          dbms_output.put_line('Code is: '||v_code);
      END PROC1;
/
2      3      4      5      6      7
Procedure created.

SQL> SELECT grantee, table_name, grantor, privilege, type
      FROM   user_tab_privs WHERE grantee = 'DEV';
```

GRANTEE	TABLE_NAME	GRANTOR	PRIVILEGE	TYPE
-----	-----	-----	-----	-----
DEV	DBA_JUNIOR	DBA_JUNIOR	INHERIT REMOTE PRIVILEGES	USER

SQL> **EXIT**

\$

*Q/ Which other solution would be workable?*

***A/ A simple solution is to create the procedure while connected under the owner of the procedure.***



## Practice 9-4: Using Privilege Analysis Runs and CBAC Roles

### Overview

In this practice, you will use Privilege Analysis to analyze CBAC roles usage, start different runs, compare two run results to verify if CBAC roles are all used, and decide which CBAC roles can be dropped. You will also use a new view that shows information about unused grants to help reduce privilege grants, without the risk of breaking applications' functionalities.

### Tasks

1. Before starting the practice, execute the `$HOME/labs/SEC/setup_CBAC.sh` shell script. The script creates `app` and `u1` users, the `app.ivproc` invoker's rights procedure, the `role1` and `role3` CBAC roles, and the `role2` role in `pdb2`. The script also creates a simple path between `r1_app` and `r2_app` roles. `r1_app` is granted `SELECT` privilege on `app.t3` and then granted to `r2_app`. `r2_app` is then granted to `u1`.

```
$ $HOME/labs/SEC/setup_CBAC.sh
...
$
```

2. Before `u1` connects to execute `app.ivproc`, create the capture analysis to analyze CBAC roles usage.

```
$ . oraenv
ORACLE_SID = [ORCL] ? cdb2
The Oracle base has been set to /u01/app/oracle
$ sqlplus system@pdb2
Enter password: *****
Connected to:
SQL> exec SYS.DBMS_PRIVILEGE_CAPTURE.DROP_CAPTURE ( -
        name => 'CBAC_capture')
> BEGIN SYS.DBMS_PRIVILEGE_CAPTURE.DROP_CAPTURE (          name =>
'CBAC_capture'); END;

*
ERROR at line 1:
ORA-47931: Privilege capture CBAC_capture does not exist.
ORA-06512: at "SYS.DBMS_PRIVILEGE_CAPTURE", line 51
ORA-06512: at line 1

SQL> exec SYS.DBMS_PRIVILEGE_CAPTURE.CREATE_CAPTURE ( -
        name          => 'CBAC_capture', -
        description => 'Privileges used by PUBLIC', -
        type         => dbms_privilege_capture.g_role, -
        roles        => role_name_list(-
                        'ROLE1', 'ROLE2', 'ROLE3', 'R1_APP', 'R2_APP'))
```

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

SQL>
```

3. Start the first run of the analysis.

```
SQL> exec SYS.DBMS_PRIVILEGE_CAPTURE.ENABLE_CAPTURE ( -
          name          => 'CBAC_capture', -
          run_name      => 'CBAC_first_run')

> >
PL/SQL procedure successfully completed.

SQL>
```

4. While the analysis is started, u1 executes the app.ivproc procedure, using the role1 CBAC role.
  - a. When executing the app.ivproc procedure, the role1 CBAC role is used. When selecting rows from the app.t3 table, both r1\_app and r2\_app are being used.

```
SQL> CONNECT u1@pdb2
Enter password: *****
Connected.
SQL> SELECT * FROM session_roles;

ROLE
-----
ROLE2
R2_APP
R1_APP

SQL> SET SERVEROUTPUT ON
SQL> EXEC app.ivproc(1)
Code is from Invoker right procedure: 1

PL/SQL procedure successfully completed.

SQL> SELECT * FROM app.t3;

COL3
-----
Col3

SQL>
```

Notice that the active roles at login time are *role2*, *r1\_app*, and *r2\_app*, while the CBAC *role1* can also be used.

5. Stop the first run of the analysis.

```
SQL> CONNECT system@pdb2
Enter password: *****
Connected to:
SQL> exec SYS.DBMS_PRIVILEGE_CAPTURE.DISABLE_CAPTURE ( -
          name          => 'CBAC_capture')
>
PL/SQL procedure successfully completed.

SQL>
```

6. Generate the analysis result of the first run.

```
SQL> exec SYS.DBMS_PRIVILEGE_CAPTURE.GENERATE_RESULT ( -
          name          => 'CBAC_capture', -
          run_name      => 'CBAC_first_run')
> >
PL/SQL procedure successfully completed.

SQL>
```

7. Analyze the use of the roles.

```
SQL> SELECT capture, run_name, grantee, rolename
       FROM   dba_unused_grants
       WHERE  run_name = 'CBAC_FIRST_RUN'
       AND    rolename IN ('ROLE1', 'ROLE2', 'ROLE3');

  2      3      4
CAPTURE      RUN_NAME      GRANTEE ROLENAME
-----
CBAC_capture CBAC_FIRST_RUN SYS      ROLE3
CBAC_capture CBAC_FIRST_RUN U1      ROLE2

SQL>
```

**Q1/ Why do *role2* and *role3* appear in the list?**

**A1/ *role2* allows selecting rows from the *app.t2* table and *role3* allows deleting rows from the *app.t2* table. The table is not accessed during step 4.**

**Q2/ Why does *role1* not appear in the list?**

```
SQL> SELECT capture, run_name, obj_priv, object_owner owner,
          object_name obj, used_role, path
       FROM   dba_used_objprivs_path
       WHERE  run_name = 'CBAC_FIRST_RUN';

  2      3      4
```

```

CAPTURE          RUN_NAME          OBJ_PRIV OWN OBJ USED_ROLE
-----
PATH
-----
CBAC_capture CBAC_FIRST_RUN  SELECT  APP T1  ROLE1
GRANT_PATH('APP.IVPROC', 'ROLE1')

CBAC_capture CBAC_FIRST_RUN  SELECT  APP T3  R1_APP
GRANT_PATH('U1', 'R2_APP', 'R1_APP')

SQL>

```

**A2/ role1 allows selecting rows from the app.t1 table through the app.ivproc procedure executed in step 4. role1 is a CBAC role: you can observe that the Privilege Analysis captured the use of the CBAC role.**

**Q3/ Which role path did the query on app.t3 use?**

**A3/ r1\_app is granted SELECT privilege on app.t3, r1\_app is granted to r2\_app. r2\_app is then granted to u1.**

8. Start the second run of the analysis.

```

SQL> exec SYS.DBMS_PRIVILEGE_CAPTURE.ENABLE_CAPTURE ( -
          name          => 'CBAC_capture', -
          run_name       => 'CBAC_second_run')
> >
PL/SQL procedure successfully completed.

SQL>

```

9. While the second run of the analysis is started, u1 reexecutes the app.ivproc procedure, using the role1 CBAC role and select rows from app.t2.

```

SQL> CONNECT u1@pdb2
Enter password: *****
Connected.
SQL> SET SERVEROUTPUT ON
SQL> EXEC app.ivproc(1)
Code is from Invoker right procedure: 1

PL/SQL procedure successfully completed.

```

```

SQL> SELECT * FROM app.t2;

LABEL
-----
Label1

```

```
SQL>
```

10. Stop the first run of the analysis.

```
SQL> CONNECT system@pdb2
Enter password: *****
Connected.
SQL> exec SYS.DBMS_PRIVILEGE_CAPTURE.DISABLE_CAPTURE ( -
          name          => 'CBAC_capture')
>
PL/SQL procedure successfully completed.

SQL>
```

11. Generate the analysis result of the first run.

```
SQL> exec SYS.DBMS_PRIVILEGE_CAPTURE.GENERATE_RESULT ( -
          name          => 'CBAC_capture', -
          run_name      => 'CBAC_second_run')
> >
PL/SQL procedure successfully completed.

SQL>
```

12. Analyze the use of the roles.

```
SQL> SELECT capture, run_name, grantee, rolename
        FROM   dba_unused_grants
        WHERE  run_name LIKE 'CBAC_%_RUN'
        AND    rolename IN ('ROLE1', 'ROLE2', 'ROLE3');
 2      3      4
CAPTURE      RUN_NAME          GRANTEE ROLENAME
-----
CBAC_capture CBAC_FIRST_RUN    SYS     ROLE3
CBAC_capture CBAC_FIRST_RUN    U1      ROLE2
CBAC_capture CBAC_SECOND_RUN   SYS     ROLE3

SQL>
```

*Q1/ Why does `role2` not appear in the list of the second run?*

```
SQL> SELECT capture, run_name, obj_priv, object_owner owner,
          object_name obj, used_role, path
        FROM   dba_used_objprivs_path
        WHERE  run_name LIKE 'CBAC_%_RUN' ORDER BY 2;
 2      3      4
CAPTURE      RUN_NAME          OBJ_PRIV OWN OBJ USED_ROLE
-----

```

```

PATH
-----
CBAC_capture CBAC_FIRST_RUN    SELECT    APP T1    ROLE1
GRANT_PATH('APP.IVPROC', 'ROLE1')

CBAC_capture CBAC_FIRST_RUN    SELECT    APP T3    R1_APP
GRANT_PATH('U1', 'R2_APP', 'R1_APP')

CBAC_capture CBAC_SECOND_RUN   SELECT    APP T1    ROLE1
GRANT_PATH('APP.IVPROC', 'ROLE1')

CBAC_capture CBAC_SECOND_RUN   SELECT    APP T2    ROLE2
GRANT_PATH('U1', 'ROLE2')

SQL> EXIT
$

```

**A1/ Selecting rows from `app.t2` used the `SELECT` privilege on `app.t2`, which is granted directly to `role2`.**

**Q2/ What can you envisage after comparing the first and second runs of the Privilege Analysis?**

**A2/ You could consider dropping `role3`. The role is not being used.**

14. Clean up all runs and captures using the `$HOME/labs/SEC/cleanup_AP.sh` shell script. The script disables and drops the Privilege Analysis policy.

```

$ $HOME/labs/SEC/cleanup_AP.sh
...
$

```

# **Practices for Lesson 10: Auditing**

## **Chapter 10**

## Practices for Lesson 10: Overview

---

### Practices Overview

During auditing operations, you will audit users using specific roles and not only specific privileges, and also audit operations executed under Virtual Private Database (VPD) policies and predicates.



## Practice 10-1: Auditing Users Using Roles

### Overview

In this practice, you will create an audit policy to audit all users in `toys_root` application PDBs to whom common roles are granted directly in the application root.

### Task

1. Before starting the practice, execute the `$HOME/labs/admin/glogin_10.sh` and `$HOME/labs/APP/setup_toys_app.sh` shell scripts. The first one sets formatting for all columns selected in queries, and the second one creates the `toys_root` application container and the `robots` and `dolls` application PDBs in `ORCL`.

```
$ $HOME/labs/admin/glogin_10.sh
$ $HOME/labs/APP/setup_toys_app.sh
...
$
```

2. Suppose a common role is granted directly to the `toys_owner` user, owner of the `toys_app` application, allowing `toys_owner` to delete rows from a shared table. You want to audit all delete actions on this particular shared table performed by `toys_owner`. Execute the `$HOME/labs/SEC/setup_toys_owner.sql` script that grants the `role_delete` common role to `toys_owner` in the `toys_root` application container commonly.

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

```
$ sqlplus system@toys_root
Enter password: *****
Connected to:
SQL> @$HOME/labs/SEC/setup_toys_owner.sql
...
SQL>
```

3. Create the audit policy to control the `delete` object privilege usage by `toys_owner`.

```
SQL> EXEC
dbms_audit_mgmt.clean_audit_trail(dbms_audit_mgmt.audit_trail_un
ified, false, dbms_audit_mgmt.container_current)

PL/SQL procedure successfully completed.

SQL> CREATE AUDIT POLICY delete_pol
          ACTIONS delete ON test.test CONTAINER = ALL;
2
Audit policy created.

SQL> AUDIT POLICY delete_pol BY toys_owner;
```

Audit succeeded.

```
SQL> SELECT entity_name, entity_type, enabled_option
       FROM audit_unified_enabled_policies
       WHERE policy_name = 'DELETE_POL';
```

2 3

```
ENTITY_NAME ENTITY_ ENABLED_OPTION
-----
TOYS_OWNER USER BY USER
```

```
SQL> SELECT policy_name, common, inherited, audit_option
       FROM audit_unified_policies
       WHERE policy_name = 'DELETE_POL';
```

2 3

```
POLICY_NAME COM INH AUDIT_OPTION
-----
DELETE_POL YES NO DELETE
```

SQL>

4. Let the common user `toys_owner` perform audited operations in the application root on the shared table.

```
SQL> CONNECT toys_owner@toys_root
Enter password: *****
Connected.
SQL> DELETE FROM test.test WHERE testcol='Test1';
```

1 row deleted.

```
SQL> ROLLBACK;
```

Rollback complete.

SQL>

5. Check whether the use of object `DELETE` privilege is audited.

```
SQL> CONNECT system@toys_root
Enter password: *****
Connected.
SQL> SELECT dbusername, action_name, object_name, pdb_name
       FROM cdb_unified_audit_trail u, cdb_pdbs p
```

```

        WHERE u.dbid = p.dbid
        AND   UNIFIED_AUDIT_POLICIES = 'DELETE_POL';
2      3      4      5

```

```

DBUSERNAME ACTION_NAME  OBJECT_NAM PDB_NAME
-----
TOYS_OWNER DELETE        TEST        TOYS_ROOT

SQL>

```

*Q/ What happens when you create a new user? Is the new user audited?*

```

SQL> CREATE USER test2 IDENTIFIED BY oracle_4U;

User created.

SQL> GRANT create session TO test2;

Grant succeeded.

SQL> GRANT select, delete ON test.test TO test2;

Grant succeeded.

SQL> CONNECT test2@toys_root
Enter password: *****
Connected.

```

```

SQL> DELETE FROM test.test WHERE testcol='Test1';

```

```

1 row deleted.

SQL> ROLLBACK;

Rollback complete.

SQL> CONNECT system@toys_root
Enter password: *****
Connected.

SQL> SELECT dbusername, action_name, object_name, pdb_name
        FROM   cdb_unified_audit_trail u, cdb_pdfs p
        WHERE  u.dbid = p.dbid
        AND    UNIFIED_AUDIT_POLICIES = 'DELETE_POL';
2      3      4      5

```

DBUSERNAME	ACTION_NAME	OBJECT_NAM	PDB_NAME
TOYS_OWNER	DELETE	TEST	TOYS_ROOT

SQL>

***A/ The policy controls toys\_owner usage of the delete privilege. The new user is not audited. The new enhancement allows you to enable an audit policy on a role and get it effective for all users to whom the role is granted now or later.***

***If you looked at the ENABLED\_OPTION in the audit\_unified\_enabled\_policies view in step 3, you would have observed that the enabled option was set to BY USER and not BY GRANTED ROLE.***

6. Drop the audit policy and retest after recreating the audit policy on roles usage.
  - a. Execute the \$HOME/labs/SEC/cleanup\_audit\_pol.sql script to drop the audit policy.

```
SQL> @$HOME/labs/SEC/cleanup_audit_pol.sql
...
SQL> CREATE AUDIT POLICY delete_pol
        ACTIONS delete ON test.test CONTAINER = ALL;

2
Audit policy created.

SQL> AUDIT POLICY delete_pol BY USERS
        WITH GRANTED ROLES role_delete;

2
Audit succeeded.

SQL> SELECT entity_name, entity_type, enabled_option
        FROM   audit_unified_enabled_policies
        WHERE  policy_name = 'DELETE_POL';

2      3
ENTITY_NAME ENTITY_ ENABLED_OPTION
-----
ROLE_DELETE ROLE      BY GRANTED ROLE

SQL>
```

- b. Let the application common user, toys\_owner, perform audited operations in the application root on common tables.

```
SQL> CONNECT toys_owner@toys_root
Enter password: *****
Connected.
SQL> DELETE FROM test.test WHERE testcol='Test2';

1 row deleted.
```

```
SQL> ROLLBACK;
```

```
Rollback complete.
```

```
SQL>
```

- c. Check whether the use of privilege is audited.

```
SQL> CONNECT system@toys_root
```

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

```
Connected.
```

```
SQL> SELECT dbusername, action_name, object_name, pdb_name
        FROM   cdb_unified_audit_trail u, cdb_pdbas p
        WHERE  u.dbid = p.dbid
        AND    UNIFIED_AUDIT_POLICIES = 'DELETE_POL';
```

2	3	4	
DBUSERNAME	ACTION_NAME	OBJECT_NAM	PDB_NAME
-----	-----	-----	-----
TOYS_OWNER	DELETE	TEST	TOYS_ROOT

```
SQL>
```

- d. Create a new user and grant `role_delete` to the new user. Connected under the new user, delete rows from the table and check whether the action is audited.

```
SQL> CREATE USER test3 IDENTIFIED BY oracle_4U;
```

```
User created.
```

```
SQL> GRANT create session TO test3;
```

```
Grant succeeded.
```

```
SQL> GRANT role_delete TO test3;
```

```
Grant succeeded.
```

```
SQL> CONNECT test3@toys_root
```

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

```
Connected.
```

```
SQL> DELETE FROM test.test WHERE testcol='Test2';
```

```
DELETE FROM test.test WHERE testcol='Test2'
```

```
*
```

```
ERROR at line 1:
```

```
ORA-01031: insufficient privileges
```

```

SQL> CONNECT system@toys_root
Enter password: *****
Connected.
SQL> SELECT dbusername, action_name, object_name, pdb_name
       FROM   cdb_unified_audit_trail u, cdb_pdb$ p
       WHERE  u.dbid = p.dbid
       AND    UNIFIED_AUDIT_POLICIES = 'DELETE_POL';

```

2	3	4	
DBUSERNAME	ACTION_NAME	OBJECT_NAME	PDB_NAME
TOYS_OWNER	DELETE	TEST	TOYS_ROOT
TEST3	DELETE	TEST	TOYS_ROOT

```

SQL>

```

**Observe that any new user to whom the role is granted is automatically audited.**

7. Clean up the audit policy and test users.

```

SQL> @$HOME/labs/SEC/cleanup_audit_pol.sql
...
SQL> EXIT
$

```

## Practice 10-2: Auditing Tables with VPD Policies (Optional)

### Overview

In this practice, you will audit operations executed under Virtual Private Database (VPD) policies and predicates.

### Tasks

1. In `pdb_orcl`, implement two VPD policies on the `app.employees` table, one policy limiting the rows returned from `app.employees` queried by users to those containing the user name and another one limiting the rows to users' salary. The rows returned follow two predicates defined in the VPD policies. Execute the `$HOME/labs/SEC/setup_VPD.sh` shell script to create `app.employees`, create the `sec.policy_pkg` functions, and implement the `sec.employee_limit_pol1` and `sec.employee_limit_pol2` VPD policies using the `sec.policy_pkg` functions to define the restrictive predicates.

```
$ $HOME/labs/SEC/setup_VPD.sh
...
$
```

2. Test the `limit_user1` and `limit_user2` policy functions.

```
$ sqlplus sec@pdb_orcl
Enter password: *****
Connected to:
SQL> SELECT policy_pkg.limit_user1('a', 'b') FROM DUAL;

POLICY_PKG.LIMIT_USER('A','B')
-----
first_name = SYS_CONTEXT('userenv', 'session_user')

SQL> SELECT policy_pkg.limit_user2('a', 'b') FROM DUAL;

POLICY_PKG.LIMIT_USER2('A','B')
-----
sal < 3000

SQL>
```

3. Implement the `employee_limit_pol` VPD policies with the following characteristics:  
The policy limits the rows that are selected from the `app.employees` table.  
The functions that are used to return predicates are `SEC.POLICY_PKG.LIMIT_USER1` and `SEC.POLICY_PKG.LIMIT_USER2`.

```
SQL> EXEC dbms_rls.drop_policy('APP', -
'EMPLOYEES', 'EMPLOYEE_LIMIT_POL1')
> BEGIN dbms_rls.drop_policy('APP',
'EMPLOYEES', 'EMPLOYEE_LIMIT_POL1'); END;
```

```

*
ERROR at line 1:
ORA-28102: policy does not exist
ORA-06512: at "SYS.DBMS_RLS_INT", line 126
ORA-06512: at "SYS.DBMS_RLS", line 122
ORA-06512: at line 1

SQL> EXEC dbms_rls.add_policy('APP','EMPLOYEES', -
        'EMPLOYEE_LIMIT_POL1', 'SEC', -
        'POLICY_PKG.LIMIT_USER1','SELECT')
> >
PL/SQL procedure successfully completed.

SQL> EXEC dbms_rls.drop_policy('APP', -
        'EMPLOYEES','EMPLOYEE_LIMIT_POL2')
> BEGIN dbms_rls.drop_policy('A2P',
'EMPLOYEES','EMPLOYEE_LIMIT_POL1'); END;

*
ERROR at line 1:
ORA-28102: policy does not exist
ORA-06512: at "SYS.DBMS_RLS_INT", line 126
ORA-06512: at "SYS.DBMS_RLS", line 122
ORA-06512: at line 1

SQL> EXEC dbms_rls.add_policy('APP','EMPLOYEES', -
        'EMPLOYEE_LIMIT_POL2', 'SEC', -
        'POLICY_PKG.LIMIT_USER2','SELECT')
> >
PL/SQL procedure successfully completed.

SQL>

```

4. Connect as `kate` and then as `jim` and check if users can display all rows of the `app.employees` table.

```

SQL> CONNECT kate@pdb_orcl
Enter Password: *****
Connected.
SQL> SELECT * FROM app.employees;

no rows selected.

SQL> CONNECT jim@pdb_orcl

```



Enter Password: \*\*\*\*\*

Connected.

SQL> **SELECT \* FROM app.employees;**

EMP_ID	FIRST_NAME	HIREDATE	SAL	JOB
100	JIM	01-JAN-00	1000	DEVELOPER

SQL>

*Q/ Which rows can users display from the app.employees table?*

SQL> **CONNECT system@pdb\_orcl**

Enter Password: \*\*\*\*\*

Connected.

SQL> **SELECT distinct predicate, sql\_text**  
**FROM v\$vpd\_policy p, v\$sql s**  
**WHERE s.child\_address = p.address**  
**AND sql\_text = 'SELECT \* FROM app.employees';**

2 3 4

PREDICATE

SQL\_TEXT

**first\_name = SYS\_CONTEXT('userenv', 'session\_user')**

**SELECT \* FROM app.employees**

**sal < 3000**

**SELECT \* FROM app.employees**

SQL>

**A/ The users can display only the rows restricted by the predicates defined in the VPD policies. The predicate appended to the query statement is 'WHERE first\_name = SYS\_CONTEXT('userenv', 'session\_user') AND sal < 3000'.**

5. Audit users selecting rows from the app.employees table and check if the VPD predicate is stored in unified\_audit\_trail.
  - a. Create and enable the audit policy.

SQL> **EXEC**

**dbms\_audit\_mgmt.clean\_audit\_trail(dbms\_audit\_mgmt.audit\_trail\_unified, false, dbms\_audit\_mgmt.container\_current)**

PL/SQL procedure successfully completed.

SQL> **CREATE AUDIT POLICY select\_vpd\_pol**

```

                ACTIONS select ON app.employees;

2
Audit policy created.

SQL> AUDIT POLICY select_VPD_pol;

Audit succeeded.

SQL> SELECT entity_name, entity_type, enabled_option
        FROM   audit_unified_enabled_policies
        WHERE  policy_name = 'SELECT_VPD_POL';

2      3

ENTITY_NAME  ENTITY_  ENABLED_OPTION
-----
ALL USERS    USER      BY USER

SQL> SELECT policy_name, common, inherited, audit_option
        FROM   audit_unified_policies
        WHERE  policy_name = 'SELECT_VPD_POL';

2      3

POLICY_NAME          COM INH AUDIT_OPTION
-----
SELECT_VPD_POL      NO  NO  SELECT

SQL>

```

- b. Let jim perform audited operations on the table.

```

SQL> CONNECT jim@pdb_orcl
Enter password: *****
Connected.

```

```

SQL> SELECT * FROM app.employees;

```

EMP_ID	FIRST_NAME	HIREDATE	SAL	JOB
100	JIM	01-JAN-00	1000	DEVELOPER

```

SQL>

```

- c. Check if the VPD predicate is stored in unified\_audit\_trail.

```
SQL> CONNECT system@pdb_orcl
Enter password: *****
Connected.
SQL> SET long 20000
SQL> SELECT dbusername, action_name, object_name, rls_info
        FROM unified_audit_trail
        WHERE UNIFIED_AUDIT_POLICIES = 'SELECT_VPD_POL';
 2      3

DBUSERNAME ACTION_NAME OBJECT_NAME
-----
RLS_INFO
-----
JIM          SELECT          EMPLOYEES
((POLICY_TYPE=[3] 'VPD'), (POLICY_SCHEMA=[3] 'SEC'), (POLICY_NAME=[1
9] 'EMPLOYEE_LIMIT_POL2'), (PREDICATE=[10] 'sal <
3000'))); ((POLICY_TYPE=[3] 'VPD'), (POLICY_SCHEMA=[3] 'SEC'), (POLICY
_NAME=[19] 'EMPLOYEE_LIMIT_POL1'), (PREDICATE=[51] 'first_name =
SYS_CONTEXT('userenv', 'session_user')')));

SQL> EXIT
$
```

*Q/ Why is there a difference between the two results above (from Step 4 and Step 5c)?*

*A/ The v\$VPD\_POLICY and v\$SQL views display information from memory. It is volatile, whereas the UNIFIED\_AUDIT\_TRAIL view displays information from the AUDSYS schema user table. The two VPD policies predicates are stored in the RLS\_INFO column.*

6. Clean up the audit and VPD policies.

```
$ $HOME/labs/SEC/cleanup_VPD.sh
...
$
```



# **Practices for Lesson 11: Oracle Data Redaction**

## **Chapter 11**

## Practices for Lesson 11: 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 in `pdb_orcl`.

You will benefit from the new Data Redaction formats library, where Data Redaction formats are available to create policies. You will also use multiple policy expressions in the same policy on a table.

## Practice 11-1: Using Data Redaction Formats in Data Redaction Policies

---

### Overview

In this practice, you will create a new Data Redaction format in the Data Redaction library, to later create a Data Redaction policy using the format.

### Tasks

1. Before starting the practice, execute the `$HOME/labs/admin/glogin_11.sh` and `$HOME/labs/SEC/setup_pdb_orcl.sh` shell scripts. The first one sets formatting for all columns selected in queries, and the second one creates a regular PDB `pdb_orcl` in `orcl` and the `hr.employees` table in `pdb_orcl`.

```
$ $HOME/labs/admin/glogin_11.sh
$ $HOME/labs/SEC/setup_pdb_orcl.sh
...
$
```

2. 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 hr@pdb_orcl

Enter password: *****
Connected to:
SQL> SELECT * FROM hr.employees;

      EMP_ID LAST_NAME      ADDRESS                SALARY
-----
      100 Smith          10 Malcolm Drive        1000
      200 Adam           20 Market Street        2000
      300 Hardy           30 Third Avenue          3000
      400 Laurel          40 Champs-Elysees        4000

SQL>
```

3. Before creating the redaction policy on the `hr.employees` table to redact the `address` column by masking the street number, create a redaction format for any address of any table in any target database.
  - a. Connect to Enterprise Manager Cloud Control to the `pdb_orcl` target database. If an error appears, you must first start the OMS; else proceed directly with step b.
    - 1) Start the Enterprise Manager Repository Database `cdbe` if it is not already started.

```
$ . oraenv
ORACLE_SID = [ORCL] ? cdbem
The Oracle base remains unchanged with value /u01/app/oracle
$ sqlplus / AS SYSDBA

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

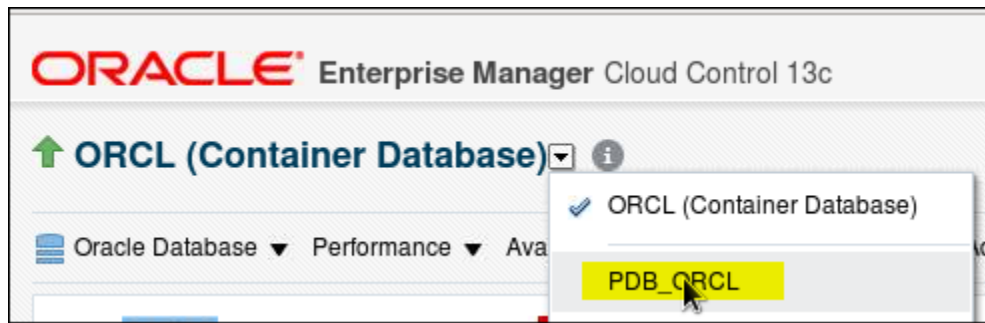
Total System Global Area  400846848 bytes
Fixed Size                  2271568 bytes
Variable Size              339740336 bytes
Database Buffers           50331648 bytes
Redo Buffers                8503296 bytes
Database mounted.
Database opened.
SQL> EXIT
$
```

- 2) Restart the OMS.

```
$ export OMS_HOME=/u01/app/oracle/product/middleware/oms
$ $OMS_HOME/bin/emctl start oms
Oracle Enterprise Manager Cloud Control 13c Release 1
Copyright (c) 1996, 2015 Oracle Corporation. All rights reserved.
Starting Oracle Management Server...
WebTier Successfully Started
Oracle Management Server Already Started
Oracle Management Server is Up
JVMD Engine is Up
$
```

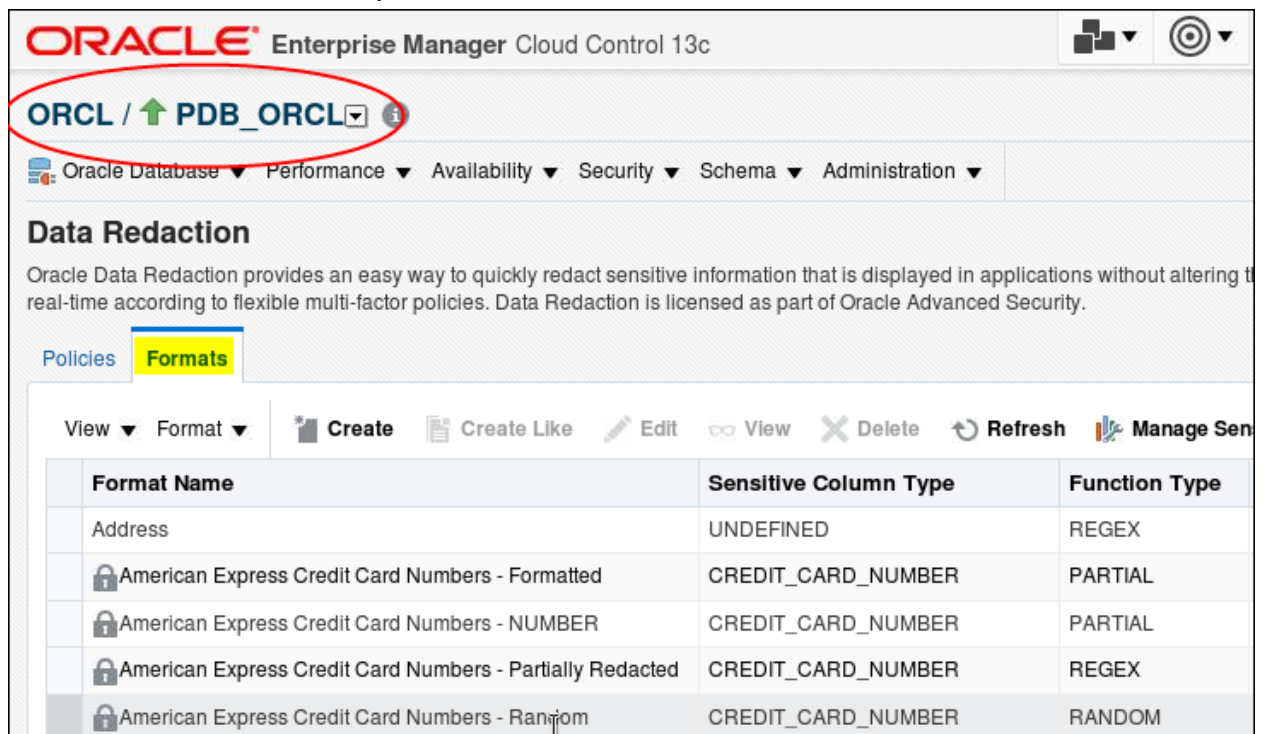


- 3) If the `pdb_orcl` target database does not appear in the list of monitored targets in `orcl`, proceed as described in Practice 1-2, task 8.



b. Create the Data Redaction format.

- 1) Click Security, and then Data Redaction. The Database Login page asks to confirm the `ORCL_SYS` credentials usage to log in to `pdb_orcl`. Click Login.
- 2) Click the Formats tab. The list of the predefined redaction formats is displayed. Select the “American Express Credit Card Numbers – Random” format.



3) Click View to read the attributes of the format. When you have finished, click OK.

**View** ✕

* <b>Format Name</b>	American Express Credit Card Numbers - Random
* <b>Description</b>	Redact the American Express Credit Card Number by replacing all digits with random digits
<b>Sensitive Column Type</b>	CREDIT_CARD_NUMBER
* <b>Redaction Function</b>	RANDOM

**Random Redaction.** The redacted data presented to the querying user appears as randomly-generated values each time it is displayed, depending on the data type of the column.

OK

- 4) Create the `street_number` format so that the street number in any address is systematically redacted to XX. Still in the Formats tab, click the Create button. Then fill the fields as defined in the screenshot below. Then click OK.

* Format Name	street_number
* Description	set Number displayed as XX
Sensitive Column Type	UNDEFINED
* Redaction Function	REGEX

**Regular Expression Based Redaction.** Specifies a regular expression that represents the column data that will be redacted.

**Function Attributes**

* Pattern	(\d\d)
-----------	--------

Specifies the regular expression pattern to be searched.  
Example: `*\d\d\d\d\d\d678` for number like `'012345678'`

* Replace String	XX
------------------	----

Example: Use `'XXXXXX\3'` (replace string) to redact `'012345678'` (actual value) which matches `'(\d\d\d) (\d\d\d) (\d\d\d)'` (regexp pattern) to `'XXXXXX678'` (redacted value). Note that the `\3` in the replace string preserves the actual data in the third set of parentheses in the pattern.

* Position	1
------------	---

Specifies the starting position of the string search. The default is 1, meaning it begins the search from the first character of column data.

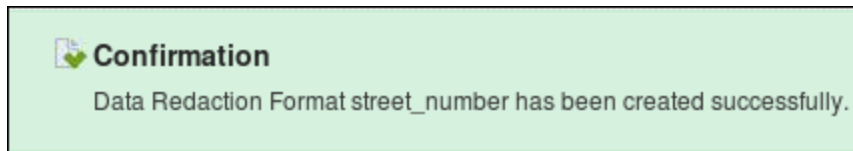
* Occurrence	0
--------------	---

Specifies how to perform the search and replace operation. Zero means it replaces all occurrences. Positive integer `'n'` would replace nth occurrence of the string.

Match Parameter	Ignore case
-----------------	-------------

Specifies the matching parameters for the REGEX redaction function.

- 5) The format creation is completed when you receive the following confirmation message:



6) You can see the new format in the format library.

UK National Insurance Numbers - VARCHAR	NATIONAL_INSURANCE_NUMBER	PARTIAL	Redact the UK National Insurance Num
UPC Numbers - Random	UNIVERSAL_PRODUCT_CODE	RANDOM	Redact the UPC Number by replacing
street_number	UNDEFINED	REGEX	Street Number dispayed as XX

4. Create the redaction policy on the `hr.employees` table to redact the address column by masking the street number. Use the new `street_number` format to redact the address column.
  - a. Click the Policies tab. In the Policies section, click the Create button. Fill the fields as defined in the screenshot below. Keep `EXPRESSION` to `1=1` since redaction should always be performed under any condition.

The screenshot shows the 'Create Data Redaction Policy' form in the Oracle Database interface. The form includes the following fields:

- Schema:** HR
- Table/View:** EMPLOYEES
- Policy Name:** Redact\_Street\_Number
- Policy Expression:** 1=1

The interface also shows navigation tabs for Oracle Database, Performance, Availability, Security, and Schema.

- b. In the Object Columns section, click the Add button. Choose the ADDRESS column to redact. Select the street\_number format to apply on the column. The attributes from the selected format are automatically displayed. Click OK.

Add

\* Column

ADDRESS

Column Datatype

VARCHAR2

Sensitive Column Type

UNDEFINED

Redaction Format

street\_number

\* Redaction Function

REGEX

Regular Expression Based Redaction. Specifies a regular expression that represents the column data that will be redacted.

Function Attributes

\* Pattern

(\d\d)

Specifies the regular expression pattern to be searched.  
Example: '\d\d\d\d\d\d678' for number like '012345678'

\* Replace String

XX

Example: Use 'XXXXXX\3' (replace string) to redact '012345678' (actual value) which matches '(\d\d\d)(\d\d\d)(\d\d\d)' (regexp pattern) to 'XXXXXX678' (redacted value). Note that the '\3' in the replace string preserves the actual data in the third set of parentheses in the pattern.

\* Position

1

Specifies the starting position of the string search. The default is 1, meaning it begins the search from the first character of column data.

\* Occurrence

0

Specifies how to perform the search and replace operation. Zero means it replaces all occurrences. Positive integer 'n' would replace nth occurrence of the string.

Match Parameter

Ignore case

- c. Before completing the `Redact_Street_Number` policy creation and application on the table column, display the PL/SQL block. Click OK. Click OK again.

**Create Data Redaction Policy: Redact\_Street\_Number**

Schema: HR  
Table/View: EMPLOYEES  
Policy Name: **Redact\_Street\_Number**  
Policy Expression: 1=1

**Show SQL**

```
BEGIN
  BEGIN DBMS_REDACT.ADD_POLICY (OBJECT_SCHEMA => 'HR',
    object_name => 'EMPLOYEES', policy_name => 'Redact_Street_Number', expression
    => '1=1'); END;

  BEGIN DBMS_REDACT.ALTER_POLICY (OBJECT_SCHEMA => 'HR',
    object_name => 'EMPLOYEES', policy_name => 'Redact_Street_Number', action =>
    DBMS_REDACT.ADD_COLUMN, column_name => "ADDRESS", function_type =>
    DBMS_REDACT.REGEXP, regexp_pattern => ('d/d),regexp_replace_string =>
    XX,regexp_position => 1,regexp_occurrence => 0,regexp_match_parameter =>
    i); END;

END;
```

**Object Columns**

Column	Column Datatype	Redaction Function	Function Attributes
ADDRESS	VARCHA2	REGEX	('d/d),XX,1,0,i

- d. The policy creation is completed when you receive the following confirmation message:

**Confirmation**  
Policy Redact\_Street\_Number has been created successfully.

5. Verify that the street number in the `address` column in the `hr.employees` table is displayed as XX.

```
SQL> CONNECT u1@pdb_orcl
Enter password: *****
Connected.
SQL> SELECT * FROM hr.employees;
```

EMP_ID	LAST_NAME	ADDRESS	SALARY
100	Smith	XX Malcolm Drive	1000
200	Adam	XX Market Street	2000
300	Hardy	XX Third Avenue	3000
400	Laurel	XX Champs-Elysees	4000

```
SQL>
```

*Q/ Was it necessary to connect to the PDB where the redaction policy was created to use the redaction format?*

***A/ No, it was not necessary to connect to the PDB where the redaction policy was created because the redaction format used is stored and available in Enterprise Manager repository. In Enterprise Manager Cloud Control, connect to the cdb2 target. Click Security, and then Data Redaction. In the Formats tab, the street\_number format appears in the list and is available for any new policy created in any database.***

## Practice 11-2: Managing Policy Expressions

### Overview

In this practice, you will use policy expressions to allow columns to be redacted in specific situations and not necessarily in the default situation defined by the initial expression of the policy.

### Tasks

1. Redact the `salary` column in the `hr.employees` table to 0. Using full redaction on a `NUMBER` data type column automatically uses the default 0 value.

```
SQL> CONNECT system@pdb_orcl
Enter password: *****
Connected.
SQL> SELECT number_value, varchar_value
       FROM REDACTION_VALUES_FOR_TYPE_FULL;

2

NUMBER_VALUE V
-----
0
```

SQL>

- a. Add the `salary` column to a policy for full redaction. Be aware that policies are case sensitive.

```
SQL> EXEC DBMS_REDACT.DROP_POLICY ( -
        policy_name    => 'Redact_Street_Number', -
        object_schema => 'HR', object_name    => 'EMPLOYEES')
> >

PL/SQL procedure successfully completed.

SQL> EXEC DBMS_REDACT.DROP_POLICY ( -
        policy_name    => 'REDACT_POLICY_EMP', -
        object_schema => 'HR', object_name    => 'EMPLOYEES')
> > BEGIN DBMS_REDACT.DROP_POLICY (        policy_name    =>
'REDACT_POLICY_EMP',        object_schema => 'HR', object_name
=> 'EMPLOYEES'); END;

*
ERROR at line 1:
ORA-28068: The object "EMPLOYEES" does not have a data redaction
policy.
ORA-06512: at "SYS.DBMS_REDACT_INT", line 57
ORA-06512: at "SYS.DBMS_REDACT", line 116
```



ORA-06512: at line 1

```
SQL> EXEC DBMS_REDACT.ADD_POLICY (object_schema => 'HR',-
                                object_name   => 'EMPLOYEES',-
                                policy_name  => 'REDACT_POLICY_EMP',-
                                column_name  => 'SALARY',-
                                function_type => DBMS_REDACT.FULL,-
                                expression   => '1=1')
```

> > > >

PL/SQL procedure successfully completed.

SQL>

- b. Query the `hr.employees` table again and note that the value of the `salary` column is 0 for all rows displayed.

```
SQL> CONNECT ul@pdb_orcl
```

Enter password: \*\*\*\*\*

Connected.

```
SQL> SELECT last_name, address, salary FROM hr.employees;
```

LAST_NAME	ADDRESS	SALARY
Smith	10 Malcolm Drive	0
Adam	20 Market Street	0
Hardy	30 Third Avenue	0
Laurel	40 Champs-Elysees	0

SQL>

2. Display the last name and address of all employees.

```
SQL> CONNECT laurel@pdb_orcl
```

Enter password: \*\*\*\*\*

Connected.

```
SQL> SELECT last_name, address, salary FROM hr.employees;
```

LAST_NAME	ADDRESS	SALARY
Smith	10 Malcolm Drive	0
Adam	20 Market Street	0
Hardy	30 Third Avenue	0
Laurel	40 Champs-Elysees	0

SQL>

- a. Add the `last_name` column to the policy for full redaction except for the `Laurel` user.

```
SQL> CONNECT system@pdb_orcl
Enter password: *****
Connected.
SQL> EXEC DBMS_REDACT.ALTER_POLICY (object_schema => 'HR', -
      object_name    => 'EMPLOYEES', -
      policy_name    => 'REDACT_POLICY_EMP', -
      action         => DBMS_REDACT.ADD_COLUMN, -
      column_name    => 'LAST_NAME', -
      expression     => -
      'SYS_CONTEXT(''USERENV'', 'SESSION_USER') != 'LAUREL''')
> > > >
PL/SQL procedure successfully completed.

SQL>
```

- b. The `redaction_columns` view shows masking functions defined on the `hr.employees` table.

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

 2      3
OBJECT_OWNER OBJECT_NAME  COLUMN_NAME  FUNCTION_TYPE
-----
HR            EMPLOYEES    SALARY       FULL REDACTION
HR            EMPLOYEES    LAST_NAME    FULL REDACTION

SQL>
```

- c. Display the values of the `last_name` and `salary` columns. First connect as `u1` and then as `laurel`.

```
SQL> CONNECT u1@pdb_orcl
Enter password: *****
Connected.
SQL> SELECT last_name, salary FROM hr.employees;

LAST_NAME      SALARY
-----
0
0
0
0

SQL> CONNECT laurel@pdb_orcl
```

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

```
Connected.
```

```
SQL> /
```

LAST_NAME	SALARY
-----	-----
	0
	0
	0
	0

```
SQL>
```

*Q/ Why is the result not fully the expected one?*

*A/ The default value for full redaction applies to the values of all rows omitting the expression of the policy. The expression of the redaction policy is still set to 1=1.*

d. Modify the expression.

```
SQL> CONNECT system@pdb_orcl
```

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

```
Connected.
```

```
SQL> EXEC DBMS_REDACT.ALTER_POLICY (object_schema => 'HR',-  
      object_name => 'EMPLOYEES',-  
      policy_name => 'REDACT_POLICY_EMP',-  
      action      => DBMS_REDACT.MODIFY_EXPRESSION,-  
      expression  =>  
      'SYS_CONTEXT(''USERENV'', 'SESSION_USER') != 'LAUREL'')
```

```
> > >
```

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

```
SQL> SELECT policy_name, expression FROM redaction_policies;
```

POLICY_NAME	EXPRESSION
-----	-----
REDACT_POLICY_EMP	
SYS_CONTEXT('USERENV', 'SESSION_USER') != 'LAUREL'	

```
SQL>
```

e. Retest.

```
SQL> CONNECT laurel@pdb_orcl
```

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

```
Connected.
```

```
SQL> SELECT last_name, salary FROM hr.employees;
```

```

LAST_NAME          SALARY
-----
Smith              1000
Adam               2000
Hardy              3000
Laurel             4000

```

```

SQL>
SQL> CONNECT u1@pdb_orcl
Enter password: *****
Connected.
SQL> /

```

```

LAST_NAME          SALARY
-----
                  0
                  0
                  0
                  0

```

```
SQL>
```

*Q1/ Which salary values did you expect Laurel to see?*

**A1/ Laurel should see a salary value of 0, and not the salary of other employees. Laurel can unfortunately see the salary for all the rows because there is only one expression for the redaction policy.**

*Q2/ Why doesn't U1 see the salary values?*

**A2/ U1 cannot see the employees' names because the updated expression applied. This is the normal behavior. He cannot see the employees' salaries because the default number value is used.**

*Q3/ How can you circumvent the issue?*

**A3/ Use policy expressions and apply policy expressions to specific columns.**

3. Reset the default expression for the EMP\_POLICY redaction policy.

```

SQL> CONNECT system@pdb_orcl
Enter password: *****
Connected.
SQL> EXEC DBMS_REDACT.ALTER_POLICY (object_schema => 'HR', -
      object_name => 'EMPLOYEES', -
      policy_name => 'REDACT_POLICY_EMP', -
      action      => DBMS_REDACT.MODIFY_EXPRESSION, -
      expression  => '1=1')

> > > >
PL/SQL procedure successfully completed.

```

```

SQL>
SQL> SELECT policy_name, expression FROM redaction_policies;

POLICY_NAME          EXPRESSION
-----
REDACT_POLICY_EMP 1=1

SQL>

```

4. Create a policy expression to apply on the `last_name` column only, so that the policy applies only when the user connected is not Laurel.
  - a. Create the policy expression that can be used for any column of any policy.

```

SQL> EXEC DBMS_REDACT.CREATE_POLICY_EXPRESSION ( -
          policy_expression_name => 'USER_EXPR', -
          expression              => -
            'SYS_CONTEXT(''USERENV'', 'SESSION_USER') != ''LAUREL'', -
          policy_expression_description => 'Apply to last_name col')
> > > >
PL/SQL procedure successfully completed.

```

```

SQL> SELECT policy_expression_name, expression
       FROM   redaction_expressions;

```

```

POLICY_EXPRESSION_NAME
-----
EXPRESSION
-----
USER_EXPR
SYS_CONTEXT('USERENV','SESSION_USER') != 'LAUREL'

SQL>

```

- b. Apply the new policy expression to the `last_name` column for the policy for full redaction except for Laurel.

```

SQL> EXEC DBMS_REDACT.APPLY_POLICY_EXPR_TO_COL (-
          object_schema => 'HR', -
          object_name    => 'EMPLOYEES', -
          column_name    => 'LAST_NAME', -
          policy_expression_name => 'USER_EXPR')
> > > >
PL/SQL procedure successfully completed.

SQL>

```

*Q/ How can we see that the policy expression is applied to the `last_name` column only?*

*A/ Some columns in the `REDACTION_EXPRESSIONS` view now display the column on which the policy expression is applied.*

```
SQL> SELECT policy_expression_name, expression, object_name,
           column_name
        FROM redaction_expressions;
```

```
POLICY_EXPRESSION_NAME
-----
EXPRESSION                                     OBJECT_NAME
-----
COLUMN_NAME
-----
USER_EXPR
SYS_CONTEXT('USERENV','SESSION_USER') != 'LAUREL' EMPLOYEES
LAST_NAME

SQL>
```

5. Retest.

```
SQL> CONNECT laurel@pdb_orcl
Enter password: *****
Connected.
SQL> SELECT last_name, salary FROM hr.employees;
```

LAST_NAME	SALARY
Smith	0
Adam	0
Hardy	0
Laurel	0

```
SQL> CONNECT ul@pdb_orcl
Enter password: *****
Connected.
SQL> /
```

LAST_NAME	SALARY
	0
	0
	0

SQL>

Q1/ Which values do Laurel and U1 see now in the *salary* column?

**A1/ The default FULL redaction value for any number. The default expression of the redaction policy is applied for both users connected on the *salary* column.**

Q2/ Which values do Laurel and U1 see now in the *last\_name* column?

**A2/ The policy expression applied on the *last\_name* column displays the default FULL redaction value (NULL for any varchar) for any user connected except Laurel.**

6. You have to modify the policy expression so that users allowed to see the *last\_name* column include not only Laurel but also Adam. Modify the *USER\_EXPR* policy expression.

```
SQL> CONNECT system@pdb_orcl
Enter password: *****
Connected.
SQL> EXEC DBMS_REDACT.UPDATE_POLICY_EXPRESSION (-
            policy_expression_name => 'USER_EXPR',-
            expression              => -
            'SYS_CONTEXT(''USERENV'', 'SESSION_USER') != ''LAUREL'' -
            AND -
            SYS_CONTEXT(''USERENV'', 'SESSION_USER') != ''ADAM''')
> > > > >

PL/SQL procedure successfully completed.
```

```
SQL> SELECT policy_expression_name, expression, object_name,
            column_name
        FROM redaction_expressions;
```

```

 2      3
POLICY_EXPRESSION_NAME
-----
EXPRESSION                                OBJECT_NAME
-----
COLUMN_NAME
-----
USER_EXPR
SYS_CONTEXT(''USERENV'', 'SESSION_USER') != ''LAUREL''      EMPLOYEES
AND SYS_CONTEXT(''USERENV'', 'SESSION_USER') != ''ADAM''
LAST_NAME

SQL>
```

*Q/ Will the policy expression update be dynamic and apply to the `last_name` column automatically?*

```
SQL> CONNECT laurel@pdb_orcl
Enter password: *****
Connected.
SQL> SELECT last_name, salary FROM hr.employees;
```

LAST_NAME	SALARY
Smith	0
Adam	0
Hardy	0
Laurel	0

```
SQL> CONNECT adam@pdb_orcl
Enter password: *****
Connected.
SQL> SELECT last_name, salary FROM hr.employees;
```

LAST_NAME	SALARY
Smith	0
Adam	0
Hardy	0
Laurel	0

```
SQL>
```

***A/ A policy expression update automatically updates all policies using that named expression.***

7. Clean up the redaction policies applied on the `hr.employees` table.

```
SQL> CONNECT system@pdb_orcl
Enter password: *****
Connected.
SQL> EXEC DBMS_REDACT.DROP_POLICY ( object_schema => 'HR', -
                                object_name   => 'EMPLOYEES', -
                                policy_name   => 'REDACT_POLICY_EMP')
> >
PL/SQL procedure successfully completed.

SQL>
```

*Q1/ Are policy expressions dropped with policies?*

```
SQL> SELECT policy_expression_name, expression, object_name,
```



```

                column_name
FROM    redaction_expressions;

2      3
POLICY_EXPRESSION_NAME
-----
EXPRESSION                                OBJECT_NAME
-----
COLUMN_NAME
-----
USER_EXPR
SYS_CONTEXT('USERENV','SESSION_USER')!='LAUREL'
AND SYS_CONTEXT('USERENV','SESSION_USER')!='ADAM'

SQL>

```

**A1/ Policy expressions are independent of policies.**

**Q2/ What is the difference between the result in Step 6 and the same query above?**

**A2/ The *OBJECT\_NAME* column does not contain any table name value after removing the policy from the table.**

**Q3/ Are policy expressions dropped with tables?**

```

SQL> DROP TABLE hr.employees;

Table dropped.

SQL> SELECT policy_expression_name, expression, object_name,
           column_name
FROM    redaction_expressions;

2      3
POLICY_EXPRESSION_NAME
-----
EXPRESSION                                OBJECT_NAME
-----
COLUMN_NAME
-----
USER_EXPR
SYS_CONTEXT('USERENV','SESSION_USER')!='LAUREL'
AND SYS_CONTEXT('USERENV','SESSION_USER')!='ADAM'

SQL> EXIT
$

```

**A3/ Policy expressions are also independent of table columns and can therefore be reapplied on columns in other tables, like data redaction formats can be applied on any table columns in any policy.**

8. To release resources for the next practices, execute the `$HOME/labs/SEC/cleanup_CDB2.sh` shell script to drop the `cdb2` database.

```
$ $HOME/labs/SEC/cleanup_CDB2.sh
...
$
```

# **Practices for Lesson 12: Data Encryption**

## **Chapter 12**

## Practices for Lesson 12: Overview

---

### Practices Overview

In the practices for this lesson, you use Oracle Transparent Data Encryption to encrypt essential tablespaces and decrypt non-essential tablespaces online.

## Practice 12-1: Encrypting Essential and Temporary Tablespaces

### Overview

In this practice, you will encrypt the essential tablespaces in databases, namely `system` and `temp`.

### Task

1. Before starting the practice, execute the `$HOME/labs/admin/glogin_12.sh` shell script. The script sets formatting for all columns selected in queries.

```
$ $HOME/labs/admin/glogin_12.sh
$
```

2. If you did not complete the optional practice 3-7 “Unplugging and Plugging Encrypted PDBs”, execute the `$HOME/labs/APP/enable_TDE_in_ORCL.sh` shell script to set up transparent data encryption (TDE) at the CDB root level and to create the master encryption key for `pdb_orcl`.

- a. Check first if the master encryption key for `pdb_orcl` exists.

```
$ . oraenv
ORACLE_SID = [ORCL] ? ORCL
The Oracle base remains unchanged with value /u01/app/oracle
$ sqlplus sys@pdb_orcl AS SYSDBA
Enter password: *****
Connected.
SQL> SELECT KEY_ID, KEY_USE, ACTIVATING_DBNAME,
          ACTIVATING_PDBNAME
        FROM V$ENCRYPTION_KEYS;
 2          3
KEY_ID                                KEY_USE
-----
ACTIVATING_DBNAME ACTIVATING_PDBNAME
-----
AXToIexeUU8iv5U8ZjQT29wAAAAAAAAAAAAAAAAAAAAAAAAAAAA TDE IN PDB
ORCL                                PDB_ORCL

SQL>
```

- b. If the result displayed no rows, execute the `$HOME/labs/APP/enable_TDE_in_ORCL.sh` shell script.

```
$ $HOME/labs/APP/enable_TDE_in_ORCL.sh
...
$
```

3. Encrypt the `pdb_orcl` system tablespace.
  - a. Launch the tablespace encryption.

```
SQL> COL name FORMAT A68
SQL> SELECT name FROM v$datafile;

NAME
-----
/u02/app/oracle/oradata/ORCL/pdb_orcl/system01.dbf
/u02/app/oracle/oradata/ORCL/pdb_orcl/sysaux01.dbf
/u02/app/oracle/oradata/ORCL/pdb_orcl/undotbs01.dbf

SQL> ALTER DATABASE DATAFILE
'/u02/app/oracle/oradata/ORCL/pdb_orcl/system01.dbf' ENCRYPT;
*
ERROR at line 1:
ORA-28440: cannot offline encrypt or decrypt data file 9 - file
is in use or recovery
ORA-01110: data file 9:
'/u02/app/oracle/oradata/ORCL/pdb_orcl/system01.dbf'
SQL>
```

*Q/ Why did it fail?*

***A/ The PDB must be in mounted mode to encrypt essential tablespaces like system, undo, and sysaux.***

```
SQL> SHUTDOWN IMMEDIATE
Pluggable database closed.
SQL>
```

- b. Restart the tablespace encryption.

```
SQL> ALTER DATABASE DATAFILE
'/u02/app/oracle/oradata/ORCL/pdb_orcl/system01.dbf' ENCRYPT;

Database altered.

SQL>
```

- c. When the encryption is complete, reopen `pdb_orcl`.

```
SQL> STARTUP
Pluggable Database opened.
SQL> SELECT encrypted FROM dba_tablespaces

WHERE tablespace_name = 'SYSTEM';

ENC
---
YES
```

```
SQL>
```

4. Encrypt the existing temp tablespace.

```
SQL> SELECT name FROM v$tempfile;

NAME
-----
/u02/app/oracle/oradata/ORCL/pdb_orcl/temp012016-03-23_01-28-14-
104-AM.dbf

SQL> ALTER DATABASE TEMPFILE
'/u02/app/oracle/oradata/ORCL/pdb_orcl/temp012016-03-23_01-28-
14-104-AM.dbf' ENCRYPT;
ALTER DATABASE TEMPFILE
'/u02/app/oracle/oradata/ORCL/pdb_orcl/temp012016-03-23_01-28-
14-104-AM.dbf' ENCRYPT
*
ERROR at line 1:
ORA-01916: keyword ONLINE, OFFLINE, RESIZE, AUTOEXTEND, END,
DROP, ENABLE, REMOVE, SUSPEND, ENCRYPT, or DECRYPT expected

SQL>
```

*Observe that it is not possible to encrypt an existing temporary tablespace.*

*Q/ Is it possible to create a temporary tablespace as encrypted?*

```
SQL> CREATE TEMPORARY TABLESPACE temp_enc
      TEMPFILE
      '/u02/app/oracle/oradata/ORCL/pdb_orcl/temp_enc01.dbf'
      SIZE 10m ENCRYPTION ENCRYPT;
2      3

Tablespace created.

SQL> SELECT tablespace_name, encrypted FROM dba_tablespaces
      WHERE encrypted = 'YES';
2
TABLESPACE_NAME          ENC
-----
SYSTEM                   YES
TEMP_ENC                  YES

SQL> EXIT
$
```

***A/ One can create encrypted temporary tablespaces in Oracle Database 12.2, but cannot convert an existing temporary tablespace.***

## Practice 12-2: Decrypting Existing Tablespaces Online in PDBs

### Overview

In this practice, you will encrypt and decrypt a non-essential tablespace online.

### Tasks

1. Before starting the practice, execute the `$HOME/labs/SEC/setup_NOENCTBS.sh` script that creates the non-encrypted `noenctbs` tablespace.

```
$ $HOME/labs/SEC/setup_NOENCTBS.sh
...
$
```

2. Encrypt the tablespace.

```
$ sqlplus sys@pdb_orcl AS SYSDBA
Enter password: *****
Connected to:
SQL> ALTER TABLESPACE noenctbs ENCRYPTION ENCRYPT;

Tablespace altered.

SQL>
```

If you encounter the following error message:

```
ALTER TABLESPACE noenctbs ENCRYPTION ENCRYPT
*
```

ERROR at line 1:

ORA-28425: missing a valid FILE\_NAME\_CONVERT clause

the FILE\_NAME\_CONVERT clause is required unless Oracle Managed Files are being used. In this case, use the following two statements:

```
SQL> !mkdir /u02/app/oracle/oradata/ORCL/pdb_orcl_enc
SQL>
SQL> ALTER TABLESPACE noenctbs ENCRYPTION ENCRYPT
      FILE_NAME_CONVERT=
              ('/u02/app/oracle/oradata/ORCL/pdb_orcl',
              '/u02/app/oracle/oradata/ORCL/pdb_orcl_enc');
2      3      4
```

```
ALTER TABLESPACE noenctbs ENCRYPTION ENCRYPT
*
ERROR at line 1:
ORA-28426: must FINISH a tablespace encrypt, decrypt or rekey
command first

SQL>
```

*Q/ Why does the operation not complete?*



**A/ The error message is clear. You must use the *FINISH* clause to complete the started operation.**

```
SQL> ALTER TABLESPACE noenctbs ENCRYPTION ONLINE
      FINISH ENCRYPT
      FILE_NAME_CONVERT =
          ('/u02/app/oracle/oradata/ORCL/pdb_orcl',
           '/u02/app/oracle/oradata/ORCL/pdb_orcl_enc');
2      3      4      5

Tablespace altered.

SQL>
```

```
SQL> SELECT encrypted FROM dba_tablespaces
      WHERE tablespace_name = 'NOENCTBS';
2
ENC
---
```

```
SQL>
```

**Q1/ Was the tablespace encrypted online or offline?**

**A1/ Online encryption is the default.**

**Q2/ What are the drawbacks of the online method?**

**A2/ Online uses extra storage and the operation is slower than offline.**

**Q3/ Where are the files located after encryption?**

**A3/ Datafiles are encrypted in the target destination defined in the *FILE\_NAME\_CONVERT* clause or in the *OMF* destination.**

3. You decide to stop encryption on the `noenctbs` tablespace while there are active transactions in tables in the tablespace.
  - a. From another window (*window2*), execute `$HOME/labs/SEC/transactions.sql`. The shell script creates tables in the tablespace, inserts rows, and starts update statements.

```
$ sqlplus sys@pdb_orcl AS SYSDBA
Enter password: *****
Connected to:
SQL> @$HOME/labs/SEC/transactions.sql
...
SQL>
```

- b. From *window1*, decrypt the tablespace.

```
SQL> ALTER TABLESPACE noenctbs ENCRYPTION DECRYPT;
```

```
Tablespace altered.
```

```
SQL> SELECT encrypted FROM dba_tablespaces  
       WHERE tablespace_name = 'NOENCTBS';
```

```
2
```

```
ENC
```

```
---
```

```
NO
```

```
SQL> EXIT
```

```
$
```

*Q/ What is your conclusion?*

***A/ Non-essential tablespaces can be decrypted online.***

- c. From *window2*, complete the update statement.

```
SQL> COMMIT;
```

```
Commit complete.
```

```
SQL>
```

*Q/ Was the tablespace decrypted online or offline?*

***A/ The tablespace was decrypted online. There are two online tablespace encryption/decryption operations. Nothing in the columns can describe if it was encryption or decryption.***

4. Execute the `$HOME/labs/SEC/disable_TDE_in_ORCL.sh` shell script to disable encryption in `ORCL` datafiles. Change the datafile directory in the shell script with the directory of your datafile.

```
$ $HOME/labs/SEC/disable_TDE_in_ORCL.sh
```

```
...
```

```
$
```

# **Practices for Lesson 13: Transparent Sensitive Data Protection**

## **Chapter 13**

## Practices for Lesson 13: Overview

---

### Practices Overview

In these practices, you will use TSDP to enforce protection on all sensitive columns of number data type, like employees salary and customers credit card number, through audit policies control.

## Practice 13-1: Managing TSDP Policies with Unified Auditing Settings

### Overview

In this practice, you create a TSDP policy to protect HR and OE sensitive columns of number data type, like employees salary and customers credit card number, against update actions on the sensitive columns by using an audit policy in `pdb_orcl`.

### Tasks

1. Before starting the practice, execute the `$HOME/labs/admin/glogin_13.sh` shell script. The script sets formatting for all columns selected in queries.

```
$ $HOME/labs/admin/glogin_13.sh
$
```

2. Use the `$HOME/labs/SEC/setup_TSDP.sh` shell script. The script performs the following operations:

- Creates tables: `oe.customers_info` including the `CCN` column (customers credit card number) and `hr.employees` including the `SALARY` column (employees salary)

```
$ . oraenv
ORACLE_SID = [ORCL] ? ORCL
The Oracle base remains unchanged with value /u01/app/oracle
$ $HOME/labs/SEC/setup_TSDP.sh
...
$
```

3. You will now create the `Sensitive_Numbers` and `Income` TSDP sensitive types.

```
$ sqlplus sys@pdb_orcl AS SYSDBA
...
Connected.
SQL> exec DBMS_TSDP_MANAGE.DROP_SENSITIVE_TYPE (-
        sensitive_type    => 'Sensitive_Numbers')
>
PL/SQL procedure successfully completed.

SQL> exec DBMS_TSDP_MANAGE.DROP_SENSITIVE_TYPE (-
        sensitive_type    => 'Income')
>
PL/SQL procedure successfully completed.

SQL> exec DBMS_TSDP_MANAGE.ADD_SENSITIVE_TYPE (-
        sensitive_type    => 'Sensitive_Numbers',-
        user_comment      => 'Type for credit card numbers' )
> >
PL/SQL procedure successfully completed.

SQL> exec DBMS_TSDP_MANAGE.ADD_SENSITIVE_TYPE (-
```

Copyright © 2016, Oracle and/or its affiliates. All rights reserved.

```

        sensitive_type => 'Income', -
        user_comment   => 'Type for salary' )
> >
PL/SQL procedure successfully completed.

SQL>

```

5. You will now associate the `CCN` column to the `Sensitive_Numbers` TSDP sensitive type, and associate the `SALARY` column to the `Income` TSDP sensitive type.

```

SQL> exec DBMS_TSDP_MANAGE.ADD_SENSITIVE_COLUMN(-
        schema_name      => 'OE', -
        table_name       => 'CUSTOMERS_INFO', -
        column_name      => 'CCN', -
        sensitive_type    => 'Sensitive_Numbers')
> > > >
PL/SQL procedure successfully completed.

SQL> exec DBMS_TSDP_MANAGE.ADD_SENSITIVE_COLUMN(-
        schema_name      => 'HR', -
        table_name       => 'EMPLOYEES', -
        column_name      => 'SALARY', -
        sensitive_type    => 'Income')
> > > >
PL/SQL procedure successfully completed.

SQL>

```

6. Display the list of sensitive columns associated to sensitive column types.

```

SQL> SELECT schema_name, table_name, column_name,
        sensitive_type
FROM    dba_sensitive_data;

```

```

  2      3
SCHEMA_NAM TABLE_NAME      COLUMN_NAME SENSITIVE_TYPE
-----
OE          CUSTOMERS_INFO CCN          Sensitive_Numbers
HR          EMPLOYEES      SALARY      Income

SQL>

```

7. You will create and configure a TSDP policy with Unified Audit settings.
- Create the TSDP policy. Define the audit parameters so that any update by `scott` on the sensitive columns will be audited.

```
SQL> DECLARE
    unified_audit_options DBMS_TSDP_PROTECT.FEATURE_OPTIONS;
    policy_conditions DBMS_TSDP_PROTECT.POLICY_CONDITIONS;
BEGIN
    unified_audit_options ('EVALUATE_PER') := 'STATEMENT';
    unified_audit_options ('ACTION_AUDIT_OPTIONS') := 'UPDATE';
    unified_audit_options ('ENTITY_NAME') := 'SCOTT';
    unified_audit_options ('ENABLE_OPTION') := 'BY';
    DBMS_TSDP_PROTECT.ADD_POLICY (
        POLICY_NAME          => 'Audit_nbrs',
        SECURITY_FEATURE      => DBMS_TSDP_PROTECT.UNIFIED_AUDIT,
        POLICY_ENABLE_OPTIONS => unified_audit_options,
        POLICY_APPLY_CONDITION => policy_conditions);
END;
/
 2      3      4      5      6      7      8      9     10     11     12     13     14
15
PL/SQL procedure successfully completed.

SQL>
```

- Display all information related to the new TSDP policy, like parameters and features.

```
SQL> SELECT policy_name, security_feature
        FROM   dba_tsdp_policy_feature
        WHERE  policy_name = 'Audit_nbrs';
 2      3
POLICY_NAME          SECURITY_FEATU
-----
Audit_nbrs          AUDIT

SQL> SELECT policy_name, parameter, value
        FROM   dba_tsdp_policy_parameter
        WHERE  policy_name = 'Audit_nbrs';
 2      3
POLICY_NAME          PARAMETER          VALUE
-----
Audit_nbrs          ACTION_AUDIT_OPTIONS UPDATE
Audit_nbrs          ENABLE_OPTION        BY
Audit_nbrs          ENTITY_NAME          SCOTT
Audit_nbrs          EVALUATE_PER          STATEMENT
```

```
SQL>
```

8. Associate the TSDP policy with the 'Sensitive\_Numbers' and 'Income' sensitive types.

```
SQL> exec DBMS_TSDP_PROTECT.ASSOCIATE_POLICY( -
        policy_name      => 'Audit_nbrs', -
        sensitive_type    => 'Sensitive_Numbers', -
        associate         => TRUE)

> > >
PL/SQL procedure successfully completed.

SQL> exec DBMS_TSDP_PROTECT.ASSOCIATE_POLICY( -
        policy_name      => 'Audit_nbrs', -
        sensitive_type    => 'Income', -
        associate         => TRUE)

> > >
PL/SQL procedure successfully completed.

SQL> SELECT * FROM dba_tsdp_policy_type
        WHERE policy_name = 'Audit_nbrs';
  2
POLICY_NAME      SENSITIVE_TYPE
-----
Audit_nbrs      Sensitive_Numbers
Audit_nbrs      Income

SQL>
```

9. Enable the TSDP policy protections at the sensitive type level.

```
SQL> exec DBMS_TSDP_PROTECT.ENABLE_PROTECTION_TYPE( -
        sensitive_type    => 'Sensitive_Numbers')

>
PL/SQL procedure successfully completed.

SQL> exec DBMS_TSDP_PROTECT.ENABLE_PROTECTION_TYPE( -
        sensitive_type    => 'Income')

>
PL/SQL procedure successfully completed.

SQL>
```

*Q/ What does the TSDP policy activation operation do?*

***A/ Enabling the TSDP policy automatically generates and enables an audit policy using the settings defined in the unified\_audit\_options argument.***



10. Disable the default TSDP policy REDACT\_AUDIT. It is enabled by default on all sensitive columns.

```
SQL> exec DBMS_TSDP_PROTECT.DISABLE_PROTECTION_COLUMN( -
          policy => 'REDACT_AUDIT')
>
PL/SQL procedure successfully completed.

SQL>
```

11. Display the protected columns.

```
SQL> SELECT schema_name, table_name,
           column_name, tsdp_policy,
           security_feature, security_feature_policy
       FROM dba_tsdp_policy_protection;
 2      3      4
SCHEMA_NAM TABLE_NAME      COLUMN_NAME TSDP_POLICY
SECURITY_FEATU
-----
SECURITY_FEATURE_POLICY
-----
OE          CUSTOMERS_INFO CCN          Audit_nbrs      AUDIT
ORA$UNIFIED_AUDIT_PKSksDwgLlBMLVAwnhazAq
apy7NRLCwWf4prSsjp5YA5txvniE9Ioxbw3pW03B
YOSEi2AL1rVHKc5ZU5xBtesvvgQixgaQxFahXN1H
xEYBLB7g

HR          EMPLOYEES        SALARY      Audit_nbrs      AUDIT
ORA$UNIFIED_AUDIT_da7IBpxxt63DbsyWQW6GP7
2qAkOUZvMYdCiDDtX32NHmzFaloQeHHvP0J0LnWB
H8oT3mkbCw4xXJocWz2qFLzg7QARWJSQ8esCLlrA

SQL>
```

12. Display the Unified Audit policies created.

```
SQL> SELECT policy_name, audit_option
       FROM audit_unified_policies
       WHERE object_name IN ('EMPLOYEES', 'CUSTOMERS_INFO');
 2      3
POLICY_NAME      AUDIT_OPTION
-----
ORA$UNIFIED_AUDI UPDATE
T_PKSksDWGLLBMLV
AWNHAZAQAPY7NRLC
WWF4PRSSJP5YA5TX
```

```

VNIE9IOXBW3PWO3B
YOSEI2AL1RVHKC5Z
U5XBTESVVGQIXGAQ
XFAHXN1HXEYBLB7G

ORA$UNIFIED_AUDI UPDATE
T_DA7IBPXXT63DBS
YWQW6GP72QAKOUZV
MYDCIDDTX32NHMZ
ALOQEHHP0J0LNWB
H8OT3MKBCW4XXJOC
WZ2QFLZG7QARWJSQ
8ESCLLRAF0N3VJPW

SQL>

```

13. Verify that the TSDP policy protects the two columns identified as sensitive types and that the Unified Auditing associated policy controls any update performed by `scott`.
- Connect as `PETER` and then as `scott`.

```

SQL> CONNECT peter@pdb_orcl
Enter password: *****
Connected.
SQL> UPDATE oe.customers_info SET ccn = 0;

10 rows updated.

SQL> UPDATE hr.employees SET salary = salary / 2;

7 rows updated.

SQL>
SQL> CONNECT scott@pdb_orcl
Enter password: *****
Connected.
SQL> UPDATE oe.customers_info SET ccn = 0;

10 rows updated.

SQL> UPDATE hr.employees SET salary = salary / 2;

7 rows updated.

SQL>

```

Q/ Are all actions audited?

```
SQL> CONNECT system@pdb_orcl
Enter password: *****
Connected.
SQL> SELECT dbusername, object_schema, object_name,
           action_name, sql_text
        FROM unified_audit_trail
        WHERE dbusername IN ('PETER','SCOTT');
 2      3      4
DBUSERNAME OBJECT_SCHEMA OBJECT_NAME ACTION_NAME
-----
SQL_TEXT
-----
SCOTT      OE            CUSTOMERS_INFO UPDATE
UPDATE oe.customers_info SET ccn = 0

SCOTT      HR            EMPLOYEES      UPDATE
UPDATE hr.employees SET salary = salary / 2

SQL> EXIT
$
```

**A/ Only actions performed by *scott* are audited because the audit condition is satisfied when *scott* is connected and updating sensitive columns.**

14. Drop the TSDP policy. Execute the `$HOME/labs/SEC/cleanup_TSDP.sh` shell script.

```
$ $HOME/labs/SEC/cleanup_TSDP.sh
...
$
```

## Practice 13-2: Managing TSDP Policies with FGA Settings (Optional)

### Overview

In this practice, you will create a TSDP policy to protect an HR sensitive column of number data type, like employees salary, using a Fine-Grained Audit policy in `pdb_orcl`. An FGA policy allows you to be more accurate on the actions audited. For example, in this case, you want to audit any select action on employees salary greater than 50000.

### Tasks

1. Use the `$HOME/labs/SEC/setup_TSDP.sh` shell script. The script performs the following operations:

- Creates the `hr.employees` table including the `SALARY` column (employees salary)
- Creates the `Income` TSDP sensitive type
- Associates the `SALARY` column to the `Income` TSDP sensitive type

```
$ . oraenv
ORACLE_SID = [ORCL] ? ORCL
The Oracle base remains unchanged with value /u01/app/oracle
$ $HOME/labs/SEC/setup_TSDP.sh
...
$
```

2. Display the list of sensitive columns associated to sensitive column types.

```
$ sqlplus system@pdb_orcl
Enter password: *****
Connected.
```

```
SQL> SELECT schema_name, table_name, column_name,
           sensitive_type
        FROM dba_sensitive_data;
```

```
  2      3
SCHEMA_NAM TABLE_NAME      COLUMN_NAME SENSITIVE_TYPE
-----
OE          CUSTOMERS_INFO CCN          Sensitive_Numbers
HR          EMPLOYEES       SALARY      Income

SQL>
```

3. You will create and configure a TSDP policy with FGA settings.
  - b. Create the TSDP policy. Define the FGA parameters so that any select on the salary sensitive column will be audited.

```
SQL> CONNECT sys@pdb_orcl AS SYSDBA
Enter password: *****
Connected.
SQL> DECLARE
  FGA_feature_options DBMS_TSDP_PROTECT.FEATURE_OPTIONS;
```

```

policy_conditions DBMS_TSDP_PROTECT.POLICY_CONDITIONS;
BEGIN
  FGA_feature_options('AUDIT_CONDITION'):= ' > 50000';
  FGA_feature_options('STATEMENT_TYPES'):= 'SELECT';
  DBMS_TSDP_PROTECT.ADD_POLICY(
    POLICY_NAME          => 'FGA_nbrs',
    SECURITY_FEATURE      => DBMS_TSDP_PROTECT.FGA,
    POLICY_ENABLE_OPTIONS => FGA_feature_options,
    POLICY_APPLY_CONDITION => policy_conditions);
END;
/
  2      3      4      5      6      7      8      9     10     11     12     13
PL/SQL procedure successfully completed.

SQL>

```

- c. Display all information related to the new TSDP policy, like parameters and features.

```

SQL> SELECT policy_name, security_feature
      FROM dba_tsdp_policy_feature
      WHERE policy_name = 'FGA_nbrs';

  2      3
POLICY_NAME          SECURITY_FEATU
-----
FGA_nbrs              FGA

SQL> SELECT policy_name, parameter, value
      FROM dba_tsdp_policy_parameter
      WHERE policy_name = 'FGA_nbrs';

  2      3
POLICY_NAME          PARAMETER          VALUE
-----
FGA_nbrs              AUDIT_CONDITION      > 50000
FGA_nbrs              STATEMENT_TYPES      SELECT

SQL>

```

4. Associate the TSDP policy with the 'Income' sensitive type.

```

SQL> exec DBMS_TSDP_PROTECT.ASSOCIATE_POLICY( -
      policy_name          => 'FGA_nbrs', -
      sensitive_type       => 'Income', -
      associate            => TRUE)

> > >
PL/SQL procedure successfully completed.

```

```
SQL> SELECT * FROM dba_tsdp_policy_type
      WHERE policy_name = 'FGA_nbrs';
2
POLICY_NAME          SENSITIVE_TYPE
-----
FGA_nbrs             Income

SQL>
```

5. Enable the TSDP policy protections at the sensitive type level.

```
SQL> exec DBMS_TSDP_PROTECT.ENABLE_PROTECTION_TYPE( -
      sensitive_type    => 'Income')
>
PL/SQL procedure successfully completed.

SQL>
```

*Q/ What does the TSDP policy activation operation do?*

***A/ Enabling the TSDP policy automatically generates and enables an FGA policy using the settings defined in the FGA\_feature\_options argument.***

6. Disable the default TSDP policy REDACT\_AUDIT. It is enabled by default on all sensitive columns.

```
SQL> exec DBMS_TSDP_PROTECT.DISABLE_PROTECTION_COLUMN( -
      policy => 'REDACT_AUDIT')
>
PL/SQL procedure successfully completed.

SQL>
```

7. Display the protected columns.

```
SQL> SELECT schema_name, table_name, column_name,
      tsdp_policy, security_feature_policy
      FROM dba_tsdp_policy_protection ;
2      3
SCHEMA_NAM TABLE_NAME          COLUMN_NAME TSdp_POLICY
-----
SECURITY_FEATURE_POLICY
-----
HR          EMPLOYEES           SALARY      FGA_nbrs
ORA$FGA_G1H4GPVJePYxRs3lGe1MK1FB5BxP8Heq
w97fHaihHLsehC2IvidLS2ikOnMvD5jKzLlJt23T
IqPtOhuvQAAgF2EFLAHGq0m5cHGWpAmbHvWV4Yr0
Sgqm1va8

SQL>
```

8. Display the FGA policies created.

```
SQL> SELECT object_name, policy_text, sel, ins, upd, del
      FROM dba_audit_policies
      WHERE policy_name LIKE 'ORA$FGA%';

2      3
OBJECT_NAME      POLICY_TEXT      SEL INS UPD DEL
-----
EMPLOYEES        SALARY > 50000 YES NO  NO  NO

SQL>
```

9. Verify that the TSDP policy protects the column identified as of sensitive type and that the associated FGA policy controls any select performed on the `salary` column whose value is greater than 50000.
- a. Connect as `PETER` and then as `SCOTT`.

```
SQL> CONNECT peter@pdb_orcl
Enter password: *****
Connected.
SQL> SELECT salary FROM hr.employees WHERE salary > 50001;

      SALARY
-----
      54320
     110000
     954323

SQL>
SQL> CONNECT scott@pdb_orcl
Enter password: *****
Connected.
SQL> SELECT salary FROM hr.employees
      WHERE emp_id IN (104, 105, 106);

2
      SALARY
-----
      44324
      24325
      33326

SQL>
```

*Q/ Are the actions audited?*

```
SQL> CONNECT system@pdb_orcl
Enter password: *****
```

Connected.

```
SQL> SELECT dbusername, object_schema, object_name,
           action_name, sql_text
       FROM unified_audit_trail
       WHERE dbusername IN ('PETER','SCOTT');
 2      3      4
DBUSERNAME OBJECT_SCHEMA OBJECT_NAME ACTION_NAME
-----
SQL_TEXT
-----
PETER      HR            EMPLOYEES SELECT
SELECT salary FROM hr.employees WHERE salary > 50001
```

```
SQL> EXIT
$
```

***A/ The select actions are audited when the audit condition is satisfied, specifically when rows with salary greater than 50000 are selected.***

10. Drop the TSDP policy. Execute the `$HOME/labs/SEC/cleanup_TSDP.sh` shell script.

```
$ $HOME/labs/SEC/cleanup_TSDP.sh
...
$
```



## Practice 13-3: Managing TSDP Policies with TDE Settings (Optional)

### Overview

In this practice, you create a TSDP policy to protect an `OE` sensitive column of number data type, like customers credit card number, using encryption in `pdb_orcl`.

### Tasks

1. Use the `$HOME/labs/SEC/setup_TSDP.sh` shell script. The script performs the following operations:

- Creates the `oe.customers_info` table including the `CCN` column (customers credit card number)
- Creates the `Sensitive_Numbers` and `Income` TSDP sensitive types
- Associates the `CCN` column to the `Sensitive_Numbers` TSDP sensitive type

```
$ . oraenv
ORACLE_SID = [ORCL] ? ORCL
The Oracle base remains unchanged with value /u01/app/oracle
$ $HOME/labs/SEC/setup_TSDP.sh
...
$
```

2. Display the list of sensitive columns associated to sensitive column types.

```
$ sqlplus system@pdb_orcl
Enter password: *****
Connected.
```

```
SQL> SELECT schema_name, table_name, column_name,
           sensitive_type
        FROM dba_sensitive_data;
2      3
```

SCHEMA_NAM	TABLE_NAME	COLUMN_NAME	SENSITIVE_TYPE
OE	CUSTOMERS_INFO	CCN	Sensitive_Numbers
HR	EMPLOYEES	SALARY	Income

```
SQL>
```

3. You will create and configure a TSDP policy with TDE settings.
  - a. Create the TSDP policy. Define the TDE parameters so that any insert or update on the credit card number sensitive column will need encryption and any select on the credit card number sensitive column will need decryption.

```
SQL> CONNECT sys@pdb_orcl AS SYSDBA
Enter password: *****
Connected.
SQL> DECLARE
    TDE_feature_options DBMS_TSDP_PROTECT.FEATURE_OPTIONS;
```

```

policy_conditions DBMS_TSDP_PROTECT.POLICY_CONDITIONS;
BEGIN
  TDE_feature_options('encrypt_algorithm') := 'AES192';
  TDE_feature_options('integrity_algorithm') := 'NOMAC';
  TDE_feature_options('salt') := 'NO SALT';
  DBMS_TSDP_PROTECT.ADD_POLICY(
    POLICY_NAME => 'TDE_nbrs',
    SECURITY_FEATURE => DBMS_TSDP_PROTECT.COLUMN_ENCRYPTION,
    POLICY_ENABLE_OPTIONS => TDE_feature_options,
    POLICY_APPLY_CONDITION => policy_conditions);
END;
/
  2      3      4      5      6      7      8      9     10     11     12     13     14
PL/SQL procedure successfully completed.

SQL>

```

- b. Display all information related to the new TSDP policy, like parameters and features.

```

SQL> SELECT policy_name, security_feature
      FROM dba_tsdp_policy_feature
      WHERE policy_name = 'TDE_nbrs';
  2      3
POLICY_NAME          SECURITY_FEATU
-----
TDE_nbrs             COL ENCRYPT

SQL> SELECT policy_name, parameter, value
      FROM dba_tsdp_policy_parameter
      WHERE policy_name = 'TDE_nbrs';
  2      3
POLICY_NAME          PARAMETER          VALUE
-----
TDE_nbrs             encrypt_algorithm  AES192
TDE_nbrs             integrity_algorithm NOMAC
TDE_nbrs             salt               NO SALT

SQL>

```

4. Associate the TSDP policy with the 'Sensitive\_numbers' sensitive type.

```

SQL> exec DBMS_TSDP_PROTECT.ASSOCIATE_POLICY( -
      policy_name      => 'TDE_nbrs', -
      sensitive_type    => 'Sensitive_Numbers', -
      associate         => TRUE)
> > >

```

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

```
SQL> SELECT * FROM dba_tsdp_policy_type
      WHERE policy_name = 'TDE_nbrs';
2
POLICY_NAME          SENSITIVE_TYPE
-----
TDE_nbrs             Sensitive_Numbers

SQL>
```

5. Enable the TSDP policy protections at the sensitive type level.

```
SQL> exec DBMS_TSDP_PROTECT.ENABLE_PROTECTION_TYPE( -
      sensitive_type  => 'Sensitive_Numbers')
> BEGIN DBMS_TSDP_PROTECT.ENABLE_PROTECTION_TYPE(
sensitive_type      => 'Sensitive_Numbers'); END;

*
ERROR at line 1:
ORA-45618: Policy enforcement on one or more columns failed.
ORA-06512: at "SYS.DBMS_TSDP_PROTECT", line 425
ORA-06512: at line 1

SQL> EXIT
$
```

Q1/ If you encounter this error, how do you find the root cause of the error?

**A1/ Read the trace file generated in**

**/u01/app/oracle/diag/rdbms/orcl/ORCL/trace/ORCL\_ora\_<your\_spid>.trc. (Find the SPID from your session in V\$PROCESS view). It reveals that the wallet required for encryption is not available:**

**\*\*\* 2016-02-23T10:52:14.289909+00:00 (PDB\_ORCL(3))**

**Error : 45618 - TSDP Policy Enforcement failed**

**KZDP OCI Error -ORA-28365: wallet is not open**

- 1) Execute the \$HOME/labs/SEC/open\_wallet.sh script.

```
$ $HOME/labs/SEC/open_wallet.sh
...
$
```

- 2) Reattempt to enable the TSDP policy protections at the sensitive type level.

```
$ sqlplus sys@pdb_orcl AS SYSDBA
Enter password: *****
Connected.
SQL> exec DBMS_TSDP_PROTECT.ENABLE_PROTECTION_TYPE( -
      sensitive_type  => 'Sensitive_Numbers')
```

```
>
PL/SQL procedure successfully completed.

SQL>
```

*Q2/ What does the TSDP policy activation operation do?*

**A2/ Enabling the TSDP policy automatically alters the table column using the TSDP encryption settings defined in the TDE\_feature\_options argument.**

6. Display the protected columns.

```
SQL> SELECT schema_name, table_name,
           column_name, tsdp_policy,
           security_feature, security_feature_policy
FROM   dba_tsdp_policy_protection
WHERE  security_feature <> 'REDACT_AUDIT';

 2      3      4      5
SCHEMA_NAM TABLE_NAME      COLUMN_NAME TSDP_POLICY
SECURITY_FEATU
-----
SECURITY_FEATURE_POLICY
-----
OE          CUSTOMERS_INFO CCN          TDE_nbrs      COL ENCRYPT
ORA$TDECE_VwDO0BsZAH5BQAXAhKvJCL9W7UMFV7
lIFwBtTkutxlBY5g2KtaYaFN6daRrXP5b78b0A9q
UFz1M5xYW4NPYKNmmcRg7cdsFgjACTRu1uhqAhfh
bu8m5rbQ

SQL>
```

7. Verify that the table `ccn` column is assigned the `encrypt` attribute.

```
SQL> DESC oe.customers_info

Name                                     Null?      Type
-----
CUSTOMER_ID                             NOT NULL   NUMBER(6)
CUST_FIRST_NAME                          VARCHAR2(20)
CUST_LAST_NAME                           VARCHAR2(20)
CCN_TYPE                                 VARCHAR2(6)
CCN                                       NUMBER(30) ENCRYPT

SQL>
```

8. Disable the default TSDP policy `REDACT_AUDIT`. It is enabled by default on all sensitive columns.

```
SQL> exec DBMS_TSDP_PROTECT.DISABLE_PROTECTION_COLUMN( -
           policy => 'REDACT_AUDIT')

>
```

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

```
SQL>
```

9. Verify that the TSDP policy encrypts the column identified as of sensitive type and that the associated TDE policy encrypts any insert performed on the `ccn` column and decrypts any select performed on the `ccn` column.

- a. Connect as `PETER`.

```
SQL> CONNECT peter@pdb_orcl
Enter password: *****
Connected.
SQL> INSERT INTO oe.customers_info VALUES (
          100, 'Bryn', 'A', 'MASTER', 512222222222220);
      2
1 row created.

SQL> COMMIT;

Commit complete.

SQL>
```

*Q/ How can you ensure that the inserted value is encrypted?*

```
SQL> SELECT ccn FROM oe.customers_info WHERE customer_id = 100;

          CCN
-----
512222222222220

SQL> ALTER SYSTEM CHECKPOINT;

System altered.

SQL>
```

- 1) Dump the data blocks to verify that the data stored in the file is not readable. Execute the `$HOME/labs/SEC/dump_blocks.sql` script to find the database address of the `oe.customers_info` table. The script determines the `file#` and `block#` for your table (*these numbers vary*).

```
SQL> @$HOME/labs/SEC/dump_blocks.sql
...
SQL> Set ECHO ON
SQL> connect sys/oracle_4U@//localhost/pdb_orcl as sysdba
Connected.
```

```

SQL>
SQL> SELECT file_id FROM dba_data_files
      2 WHERE RELATIVE_FNO =
      3      (SELECT distinct dbms_rowid.ROWID_RELATIVE_FNO(rowid)
FILE#
      4      FROM    oe.customers_info);

      FILE_ID
-----
          99

SQL>
SQL> SELECT distinct dbms_rowid.rowid_block_number(rowid) BLOCK#
      2 FROM    oe.customers_info;

      BLOCK#
-----
        29961

SQL>

```

- 2) Set the `TRACEFILE_IDENTIFIER` initialization parameter so that the trace file can be found more easily.

```

SQL> ALTER SESSION SET TRACEFILE_IDENTIFIER = my_dump;

Session altered.

SQL>

```

- 3) Dump the data block to a trace file. Substituting the `file#` and `block#` that you recorded with the previous command, execute the following command:

```

SQL> ALTER SYSTEM DUMP DATAFILE 99 BLOCK 29961;

System altered.

SQL>

```

- 4) Find the trace file. In this listing, the block dump is in the `ORCL_ora_<pid>_MY_DUMP.trc` file.

```

SQL> !
$ cd /u01/app/oracle/diag/rdbms/orcl/ORCL/trace
$ ls *MY_DUMP*
ORCL_ora_18543_MY_DUMP.trc  ORCL_ora_18543_MY_DUMP.trm
$

```

- 5) View the dump file. The `less` utility enables you to scroll up and down the file to find data of interest. Note that the credit card numbers are clearly visible.

```
$ less ORCL_ora_18543_MY_DUMP.trc
/* Rows deleted */
...
7F8BC1203D00 00000000 00000000 00000000 0205012C
[.....,....]
7F8BC1203D10 420402C2 016E7972 414D0641 52455453
[...Bryn.A.MASTER]
7F8BC1203D20 A0979E10 0A2B04C6 8E6F2CD7 DE0C4819
[.....+...o..H..]
7F8BC1203D30 05022C5D 0202C203 6E6F430A 6E617473
[,.....Constan]
7F8BC1203D40 016E6974 414D0647 52455453 8DB97910
[tin.G.MASTER.y..]
7F8BC1203D50 079A63F6 1192F013 2203CCE6 05022CF1
[.c.....".,...]
7F8BC1203D60 0302C203 72614808 6F736972 0446016E
[.....Harrison.F.]
7F8BC1203D70 44524143 80DE6510 27879DD5 A2F3E21B
[CARD.e.....'....]
7F8BC1203D80 30818B2D 05022C67 0402C203 6E614D07  [-
..0g,.....Man]
7F8BC1203D90 61687369 41044501 1058454D 376F7634
[isha.E.AMEX.4vo7]
7F8BC1203DA0 EE052AC0 4185EA81 5C98CE08 0305022C
[.*.....A...\,...]
7F8BC1203DB0 080502C2 72726148 6E6F7369 56044401
[.....Harrison.D.V]
7F8BC1203DC0 10415349 19C6B52F 6F409099 48DE0D80
[ISA./.....@o...H]
...
q
$ exit
SQL> EXIT
$
```

***A/ The inserted value is encrypted because the data stored on disk in blocks is not readable.***

10. Drop the TSDP policy. Execute the `$HOME/labs/SEC/cleanup_TSDP.sh` shell script.

```
$ $HOME/labs/SEC/cleanup_TSDP.sh
...
$
```





# **Practices for Lesson 14: Data Availability**

## **Chapter 14**

## Practices for Lesson 14: Overview

---

### Practices Overview

In these practices, you will use new RMAN recovery features and new online operations.

## Practice 14-1: Recovering Database Until Available Redo

### Overview

In this practice, you will have to recover `ORCL` in an incomplete manner because one of the `SYSAUX` datafiles is missing and unfortunately one of the archive logs required for the database recovery is also missing. Ask Oracle to automatically find the last available archive redo log and recover the database until the last available redo.

### Tasks

1. Before starting the practice, execute the `$HOME/labs/admin/glogin_14.sh` and `$HOME/labs/RMAN/backup_ORCL.sh` shell scripts. The first one sets formatting for all columns selected in queries and the second one performs the following actions:

- Creates the `users` tablespace in `pdb_orcl` and the `hr.employees` table in the `users` tablespace
- Backs up `ORCL` and the archivelog files

```
$ . oraenv
ORACLE_SID = [ORCL] ? ORCL
The Oracle base remains unchanged with value /u01/app/oracle
$ $HOME/labs/admin/glogin_14.sh
$ $HOME/labs/RMAN/backup_ORCL.sh
...
$
```

2. Set the CDB in a situation where you will have to perform an incomplete recovery. After having inserted rows in `hr.employees` many times, the database is regularly archiving redo log files. Suddenly someone informs you that the database is not available. You discover that the datafile of the `SYSAUX` tablespace is lost.

- a. Insert entries in `hr.employees` and a redo log file archiving takes place.

```
$ sqlplus hr@pdb_orcl
Enter password: *****
Connected.
SQL> SELECT count(*) FROM hr.employees;

COUNT (*)
-----
          7

SQL> INSERT INTO hr.employees VALUES (107,'Jim','Doe',22326);

1 row created.

SQL> COMMIT;

Commit complete.
```

```
SQL> CONNECT / AS SYSDBA
Connected.
SQL> ALTER SYSTEM SWITCH LOGFILE;

System altered.

SQL> EXIT
$
```

- b. Repeat the previous operation two times by executing the `$HOME/labs/RMAN/crash.sh` shell script.

```
$ $HOME/labs/RMAN/crash.sh
```

```
...
```

```
$
```

- c. Remove the datafile of the `sysaux` tablespace.

```
$ sqlplus / AS SYSDBA
Connected.
SQL> SELECT file_name FROM dba_data_files
        WHERE tablespace_name = 'SYSAUX';
```

```
FILE_NAME
-----
/u02/app/oracle/oradata/ORCL/sysaux01.dbf

SQL>
```

```
SQL> ! rm /u02/app/oracle/oradata/ORCL/sysaux01.dbf

SQL>
```

- d. Remove the penultimate archive log file, which corresponds to penultimate archived log. In the list below, it corresponds to `/u03/app/oracle/fast_recovery_area/ORCL/archivelog/2016_03_24/o1_mf_1_141_ch7ndfyf_.arc`.

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

STATUS          SEQUENCE#
-----
ACTIVE          142
CURRENT         143
ACTIVE          141

SQL> SELECT name FROM V$ARCHIVED_LOG ORDER BY completion_time;
```

```

NAME
-----
...
/u03/app/oracle/fast_recovery_area/ORCL/archivelog/2016_03_24/o1
_mf_1_139_ch7n6zvl_.arc

/u03/app/oracle/fast_recovery_area/ORCL/archivelog/2016_03_24/o1
_mf_1_140_ch7nd2wb_.arc

/u03/app/oracle/fast_recovery_area/ORCL/archivelog/2016_03_24/o1
_mf_1_141_ch7ndfyf_.arc

/u03/app/oracle/fast_recovery_area/ORCL/archivelog/2016_03_24/o1
_mf_1_142_ch7ndk92_.arc

141 rows selected.

SQL> EXIT
$

```

```

$ rm
/u03/app/oracle/fast_recovery_area/ORCL/archivelog/2016_03_24/o1
_mf_1_141_ch7ndfyf_.arc
$

```

3. A user complains that he cannot query the `hr.employees` table.

```

$ sqlplus hr@pdb_orcl
Enter password: *****
Connected.
SQL> SELECT * FROM hr.employees;
SELECT * FROM hr.employees
*
ERROR at line 1:
ORA-01116: error in opening database file 124
ORA-01110: data file 124:
'/u02/app/oracle/oradata/ORCL/pdb_orcl/users01.dbf'
ORA-27041: unable to open file
Linux-x86_64 Error: 2: No such file or directory
Additional information: 3

SQL>

```

*The user may still view the list of employees because data blocks are still in the buffer cache. In this case, flush the buffer cache.*

```
SQL> SELECT * FROM hr.employees;
```

EMP_ID	FIRST_NAME	LAST_NAME	SALARY
100	Adam	Xue	54320
101	Joe	Yard	24321
102	Miles	Wagon	110000
103	Peter	Allan	954323
104	Scott	British	44324
105	Luis	Cole	24325
106	John	Dan	33326
107	Jim	Doe	22326
108	Kyle	Smith	22326
109	Amanda	Jude	12345

```
10 rows selected.
```

```
SQL> ALTER SYSTEM FLUSH buffer_cache;
```

```
System altered.
```

```
SQL> SELECT * FROM hr.employees;
```

```
SELECT * FROM hr.employees
      *
```

```
ERROR at line 1:
```

```
ORA-01116: error in opening database file 78
```

```
ORA-01110: data file 78:
```

```
'/u02/app/oracle/oradata/ORCL/pdb_orcl/users01.dbf'
```

```
ORA-27041: unable to open file
```

```
Linux-x86_64 Error: 2: No such file or directory
```

```
Additional information: 3
```

```
SQL>
```

4. Then force the current redo log to be archived.

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

```
Connected.
```

```
SQL> ALTER SYSTEM SWITCH LOGFILE;
```

```
System altered.
```

```
SQL> EXIT
```

5. You have to recover the `sysaux` tablespace.

```
$ rman target /

connected to target database: ORCL (DBID=1434985951)

RMAN> LIST FAILURE ALL;

using target database control file instead of recovery catalog
Failure ID Priority Status      Time Detected Summary
-----
662          HIGH      OPEN      24-MAR-16      One or more non-
system datafiles are missing

RMAN>
```

```
RMAN> ADVISE FAILURE ALL;
Database Role: PRIMARY

List of Database Failures
=====

Failure ID Priority Status      Time Detected Summary
-----
662          HIGH      OPEN      24-MAR-16      One or more non-
system datafiles are missing

analyzing automatic repair options; this may take some time
allocated channel: ORA_DISK_1
channel ORA_DISK_1: SID=870 device type=DISK
analyzing automatic repair options complete

Mandatory Manual Actions
=====

1. If file /u02/app/oracle/oradata/ORCL/sysaux01.dbf was
unintentionally renamed or moved, restore it
2. If file /u02/app/oracle/oradata/ORCL/pdb_orcl/users01.dbf was
unintentionally renamed or moved, restore it
3. If you have an export of tablespace USERS, then drop and re-
create the tablespace and import the data.
4. Contact Oracle Support Services if the preceding
recommendations cannot be used, or if they do not fix the
failures selected for repair
```

```
Optional Manual Actions
=====
no manual actions available

Automated Repair Options
=====
no automatic repair options available

RMAN>
```

***Proceed manually.***

```
RMAN> SHUTDOWN ABORT

Oracle instance shut down

RMAN> STARTUP MOUNT

connected to target database (not started)
Oracle instance started
database mounted

Total System Global Area      3774873600 bytes

Fixed Size                     4599896 bytes
Variable Size                  771755944 bytes
Database Buffers               2986344448 bytes
Redo Buffers                   12173312 bytes

RMAN> RESTORE DATABASE;

Starting restore at 24-MAR-16
allocated channel: ORA_DISK_1
channel ORA_DISK_1: SID=376 device type=DISK

skipping datafile 5; already restored to file
/u02/app/oracle/oradata/ORCL/pdbseed/system01.dbf
skipping datafile 6; already restored to file
/u02/app/oracle/oradata/ORCL/pdbseed/sysaux01.dbf
skipping datafile 8; already restored to file
/u02/app/oracle/oradata/ORCL/pdbseed/undotbs01.dbf
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
/u02/app/oracle/oradata/ORCL/system01.dbf
channel ORA_DISK_1: restoring datafile 00003 to
/u02/app/oracle/oradata/ORCL/sysaux01.dbf
channel ORA_DISK_1: restoring datafile 00004 to
/u02/app/oracle/oradata/ORCL/undotbs01.dbf
channel ORA_DISK_1: restoring datafile 00007 to
/u02/app/oracle/oradata/ORCL/users01.dbf
channel ORA_DISK_1: reading from backup piece
/u03/app/oracle/fast_recovery_area/ORCL/backupset/2016_03_24/o1_
mf_nnndf_TAG20160324T113632_ch7nb1xo_.bkp
...
channel ORA_DISK_1: restoring datafile 00121 to
/u02/app/oracle/oradata/ORCL/pdb_orcl/system01.dbf
channel ORA_DISK_1: restoring datafile 00122 to
/u02/app/oracle/oradata/ORCL/pdb_orcl/sysaux01.dbf
channel ORA_DISK_1: restoring datafile 00123 to
/u02/app/oracle/oradata/ORCL/pdb_orcl/undotbs01.dbf
channel ORA_DISK_1: restoring datafile 00124 to
/u02/app/oracle/oradata/ORCL/pdb_orcl/users01.dbf
channel ORA_DISK_1: reading from backup piece
/u03/app/oracle/fast_recovery_area/ORCL/2ECAD150169B4364E0532633
960A3BCD/backupset/2016_03_24/o1_mf_nnndf_TAG20160324T113632_ch7
nc72g_.bkp
channel ORA_DISK_1: piece
handle=/u03/app/oracle/fast_recovery_area/ORCL/2ECAD150169B4364E
0532633960A3BCD/backupset/2016_03_24/o1_mf_nnndf_TAG20160324T113
632_ch7nc72g_.bkp tag=TAG20160324T113632
channel ORA_DISK_1: restored backup piece 1
channel ORA_DISK_1: restore complete, elapsed time: 00:00:08
Finished restore at 24-MAR-16

RMAN> RECOVER DATABASE;
archived log for thread 1 with sequence 148 is already on disk
as file
/u03/app/oracle/fast_recovery_area/ORCL/archivelog/2016_03_24/o1_
mf_1_148_ch7op9gr_.arc
archived log for thread 1 with sequence 149 is already on disk
as file
/u03/app/oracle/fast_recovery_area/ORCL/archivelog/2016_03_24/o1_
mf_1_149_ch7oqg6w_.arc
archived log for thread 1 with sequence 150 is already on disk
as file
/u03/app/oracle/fast_recovery_area/ORCL/archivelog/2016_03_24/o1_
mf_1_150_ch7osm0w_.arc
RMAN-00571: =====
RMAN-00569: ===== ERROR MESSAGE STACK FOLLOWS =====

```

```

RMAN-00571: =====
RMAN-03002: failure of recover command at 03/24/2016 12:09:08
RMAN-06053: unable to perform media recovery because of missing log
RMAN-06025: no backup of archived log for thread 1 with sequence 141 and starting SCN of 53625526 found to restore

RMAN>

```

*Q/ What does the last error message mean?*

***A/ An archived log file required to perform a complete recovery is missing. Therefore the only solution is to perform an incomplete recovery.***

6. Use the new feature that is able to automatically retrieve the last available archive redo log and complete the incomplete recovery.

```

RMAN> RECOVER DATABASE UNTIL AVAILABLE REDO;

archived log for thread 1 with sequence 140 is already on disk
as file
/u03/app/oracle/fast_recovery_area/ORCL/archivelog/2016_03_24/o1
_mf_1_140_ch7nd2wb_.arc
archived log for thread 1 with sequence 142 is already on disk
as file
/u03/app/oracle/fast_recovery_area/ORCL/archivelog/2016_03_24/o1
_mf_1_142_ch7ndk92_.arc
archived log for thread 1 with sequence 143 is already on disk
as file
/u03/app/oracle/fast_recovery_area/ORCL/archivelog/2016_03_24/o1
_mf_1_143_ch7oldpj_.arc
archived log for thread 1 with sequence 144 is already on disk
as file
/u03/app/oracle/fast_recovery_area/ORCL/archivelog/2016_03_24/o1
_mf_1_144_ch7o2prg_.arc
archived log for thread 1 with sequence 145 is already on disk
as file
/u03/app/oracle/fast_recovery_area/ORCL/archivelog/2016_03_24/o1
_mf_1_145_ch7oj489_.arc
archived log for thread 1 with sequence 146 is already on disk
as file
/u03/app/oracle/fast_recovery_area/ORCL/archivelog/2016_03_24/o1
_mf_1_146_ch7oj556_.arc
archived log for thread 1 with sequence 147 is already on disk
as file
/u03/app/oracle/fast_recovery_area/ORCL/archivelog/2016_03_24/o1
_mf_1_147_ch7oj6sq_.arc
archived log for thread 1 with sequence 148 is already on disk
as file
/u03/app/oracle/fast_recovery_area/ORCL/archivelog/2016_03_24/o1
_mf_1_148_ch7op9gr_.arc

```

```

archived log for thread 1 with sequence 149 is already on disk
as file
/u03/app/oracle/fast_recovery_area/ORCL/archivelog/2016_03_24/o1_
mf_1_149_ch7oqg6w_.arc
archived log for thread 1 with sequence 150 is already on disk
as file
/u03/app/oracle/fast_recovery_area/ORCL/archivelog/2016_03_24/o1_
mf_1_150_ch7osm0w_.arc
archived log for thread 1 with sequence 151 is already on disk
as file /u04/app/oracle/redo/redo01.log
archived log file
name=/u03/app/oracle/fast_recovery_area/ORCL/archivelog/2016_03_
24/o1_mf_1_140_ch7nd2wb_.arc thread=1 sequence=140
warning: attempt media recovery until thread 1, sequence 141
Finished recover at 24-MAR-16

RMAN>

```

*Q/ What does the last warning message mean?*

***A/ An attempt to recover by using the archived log file sequence 141 was interrupted because the file is missing to perform a complete recovery. Therefore the recovery operation performed an automatic incomplete recovery.***

```

RMAN> ALTER DATABASE OPEN RESETLOGS;

Statement processed

RMAN> ALTER PLUGGABLE DATABASE pdb_orcl OPEN;

Statement processed

RMAN> EXIT

$

```

7. Verify now that the number of rows in the `hr.employees` table is the number before the datafile was removed during the execution of the `$HOME/labs/RMAN/crash.sh` shell script.

```

$ sqlplus hr@pdb_orcl
Enter password: *****
Connected.
SQL> SELECT count(*) FROM hr.employees;

COUNT (*)
-----
          9

SQL> SELECT * FROM employees;

```

EMP_ID	FIRST_NAME	LAST_NAME	SALARY
100	Adam	Xue	54320
101	Joe	Yard	24321
102	Miles	Wagon	110000
103	Peter	Allan	954323
104	Scott	British	44324
105	Luis	Cole	24325
106	John	Dan	33326
107	Jim	Doe	22326
108	Kyle	Smith	22326

9 rows selected.

SQL> **EXIT**

\$

8. After an incomplete CDB recovery, back up ORCL.

```
$ $HOME/labs/RMAN/backup_ORCL.sh
...
$
```

## Practice 14-2: Recovering Using New Commands

### Overview

In this practice, you will recover lost datafiles using new RMAN REPAIR commands.

### Tasks

1. After the incomplete CDB recovery, back up ORCL if this has not been completed at the end of the previous practice.

```
$ $HOME/labs/RMAN/backup_ORCL.sh
...
$
```

2. Set the CDB in a situation where you will have to perform a complete recovery. After having inserted rows in hr.employees many times in pdb\_orcl, suddenly someone informs you that the hr.employees table is not available anymore. You discover that the datafile of the USERS tablespace disappeared.

```
$ $HOME/labs/RMAN/crash.sh
...
$
```

*Users may still be able to connect but fail to insert rows into the table or may not be able to connect because the PDB is in shutdown progress.*

```
$ sqlplus hr@pdb_orcl
Enter password: *****
Connected.
SQL> SELECT * FROM hr.employees;

  EMP_ID FIRST_NAME      LAST_NAME      SALARY
-----
  100 Adam              Xue            54320
  101 Joe                Yard            24321
  102 Miles              Wagon          110000
  103 Peter              Allan           954323
  104 Scott              British         44324
  105 Luis                Cole            24325
  106 John                Dan             33326
  108 Kyle                Smith           22326
  109 Amanda              Jude            12345

9 rows selected.

SQL> INSERT INTO employees VALUES (110, 'Ruth', 'A1',81818);
INSERT INTO employees VALUES (110, 'Ruth', 'A1',81818);
*
ERROR at line 1:
```

```

ORA-01116: error in opening database file 38
ORA-01110: data file 38:
'/u01/app/oracle/oradata/ORCL/pdb_orcl/users01.dbf'
ORA-27041: unable to open file
Linux-x86_64 Error: 2: No such file or directory
Additional information: 3

SQL> EXIT
$

```

*The user may still be able to insert employees. In this case, flush the buffer cache and force a checkpoint if necessary until Oracle detects that the file is missing.*

```

SQL> INSERT INTO employees VALUES (110, 'Ruth', 'A1',81818);

1 row created.

SQL> ALTER SYSTEM FLUSH BUFFER_CACHE;

System altered.

SQL> ALTER SYSTEM CHECKPOINT;

System altered.

SQL> INSERT INTO employees VALUES (110, 'Ruth', 'A1',81818);
INSERT INTO employees VALUES (110, 'Ruth', 'A1',81818)
*
ERROR at line 1:
ORA-01116: error in opening database file 136
ORA-01110: data file 136:
'/u02/app/oracle/oradata/ORCL/pdb_orcl/users01.dbf'
ORA-27041: unable to open file
Linux-x86_64 Error: 2: No such file or directory
Additional information: 3

SQL> EXIT
$

```

*The HR user may also not be able to connect anymore. The PDB could be shut down due to the datafile loss.*

```

$ sqlplus hr@pdb_orcl
Enter password: *****
ERROR:
ORA-01033: ORACLE initialization or shutdown in progress

```

```
Process ID: 0
Session ID: 0 Serial number: 0
```

3. Use the new RMAN REPAIR DATAFILE command to restore and recover the missing datafile.

```
$ rman target /

connected to target database: ORCL (DBID=651773304)

RMAN> LIST FAILURE DETAIL;

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

List of Database Failures
=====

Failure ID Priority Status      Time Detected Summary
-----
662          HIGH      OPEN      24-MAR-16      One or more non-
system datafiles are missing
  Impact: See impact for individual child failures
  List of child failures for parent failure ID 662
  Failure ID Priority Status      Time Detected Summary
  -----
  1334          HIGH      OPEN      24-MAR-16      Datafile 136:
  '/u02/app/oracle/oradata/ORCL/pdb_orcl/users01.dbf' is missing
  Impact: Some objects in tablespace USERS might be
  unavailable

RMAN> REPAIR DATAFILE 136;

Starting restore at 24-MAR-16
allocated channel: ORA_DISK_1
channel ORA_DISK_1: SID=873 device type=DISK
Executing: alter database datafile 136 offline

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 00136 to
/u02/app/oracle/oradata/ORCL/pdb_orcl/users01.dbf
channel ORA_DISK_1: reading from backup piece
/u03/app/oracle/fast_recovery_area/ORCL/2ED3B35D05973BAFE0532633
```

```

960A5336/backupset/2016_03_24/o1_mf_nnndf_TAG20160324T220959_ch8
sgxs8_.bkp
channel ORA_DISK_1: piece
handle=/u03/app/oracle/fast_recovery_area/ORCL/2ED3B35D05973BAFE
0532633960A5336/backupset/2016_03_24/o1_mf_nnndf_TAG20160324T220
959_ch8sgxs8_.bkp tag=TAG20160324T220959
channel ORA_DISK_1: restored backup piece 1
channel ORA_DISK_1: restore complete, elapsed time: 00:00:02
Finished restore at 24-MAR-16

Starting recover at 24-MAR-16
using channel ORA_DISK_1

starting media recovery

archived log for thread 1 with sequence 16 is already on disk as
file
/u03/app/oracle/fast_recovery_area/ORCL/archivelog/2016_03_24/o1
_mf_1_16_ch8sh6vw_.arc
archived log for thread 1 with sequence 17 is already on disk as
file
/u03/app/oracle/fast_recovery_area/ORCL/archivelog/2016_03_24/o1
_mf_1_17_ch8sjgj0_.arc
archived log for thread 1 with sequence 18 is already on disk as
file
/u03/app/oracle/fast_recovery_area/ORCL/archivelog/2016_03_24/o1
_mf_1_18_ch8sjl0o_.arc
archived log for thread 1 with sequence 19 is already on disk as
file
/u03/app/oracle/fast_recovery_area/ORCL/archivelog/2016_03_24/o1
_mf_1_19_ch8sjpw9_.arc
archived log file
name=/u03/app/oracle/fast_recovery_area/ORCL/archivelog/2016_03_
24/o1_mf_1_16_ch8sh6vw_.arc thread=1 sequence=16
archived log file
name=/u03/app/oracle/fast_recovery_area/ORCL/archivelog/2016_03_
24/o1_mf_1_17_ch8sjgj0_.arc thread=1 sequence=17
media recovery complete, elapsed time: 00:00:03
Executing: alter database datafile 136 online
Finished recover at 24-MAR-16

RMAN>

```

*Q/ Which operations does the command complete?*

*A/ The REPAIR command performs both the restore and recover operations.*



## 4. Open the PDB.

```

RMAN> ALTER PLUGGABLE DATABASE pdb_orcl OPEN;

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 10/10/2016
04:33:14
ORA-01113: file 104 needs media recovery
ORA-01110: data file 104:
'/u02/app/oracle/oradata/ORCL/pdb_orcl/pdbseed_i1_undo.dbf'

RMAN>

```

Repair other datafiles.

- a. Use **LIST FAILURE ALL;**
- b. Use **ADVISE FAILURE ALL;**
- c. Use **REPAIR FAILURE PREVIEW;**  
*Q/ Do you see a list of datafiles to be recovered?*  
**A/ Yes.**
- d. Use **REPAIR DATAFILE <list\_of\_datafile\_numbers>;**
- e. Now open the PDB.

```

RMAN> ALTER PLUGGABLE DATABASE pdb_orcl OPEN;

Statement processed

RMAN> EXIT
$

```

5. Inform users that they can now insert rows into `hr.employees`.

```

$ sqlplus hr@pdb_orcl

Enter password: *****
Connected to:

SQL> SELECT * FROM hr.employees;

  EMP_ID FIRST_NAME      LAST_NAME      SALARY
-----
    100      Adam      Xue           54320
    101       Joe      Yard           24321
    102     Miles     Wagon          110000
    103     Peter     Allan          954323

```

104	Scott	British	44324
105	Luis	Cole	24325
106	John	Dan	33326
108	Kyle	Smith	22326
109	Amanda	Jude	12345

9 rows selected.

```
SQL> INSERT INTO hr.employees VALUES (110, 'Ruth', 'A1', 81818);
```

1 row created.

```
SQL> COMMIT;
```

Commit complete.

```
SQL> EXIT
```

\$

6. After the PDB recovery, back up PDB\_ORCL.

```
$ $HOME/labs/RMAN/backup_pdb_orcl.sh
```

...

```
$
```

## Practice 14-3: Recovering Tables Across Schemas

### Overview

In this practice, you perform a table recovery of `hr.employees` after having inadvertently deleted rows from the table.

### Tasks

1. Execute the `$HOME/labs/RMAN/employees.sh` shell script to recreate the `hr.employees` table, populate it with seven rows in `pdb_orcl`, and create the unique index `hr.i_un_names`.

```
$ $HOME/labs/RMAN/employees.sh
...
$
```

2. Back up the pluggable database where the table is stored, considering that the CDB has been backed up in a previous practice with `$HOME/labs/RMAN/backup_ORCL.sh` shell script. If this is not the case, back up the `ORCL` CDB by executing the `BACKUP DATABASE` command.

```
$ export NLS_DATE_FORMAT='DD-MM-YYYY HH:MI:SS'
$ rman TARGET /

connected to target database: ORCL (DBID=1434985951)

RMAN> BACKUP PLUGGABLE DATABASE pdb_orcl;

Starting backup at 30-03-2016 01:43:08
using target database control file instead of recovery catalog
allocated channel: ORA_DISK_1
channel ORA_DISK_1: SID=866 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=00190
name=/u02/app/oracle/oradata/ORCL/pdb_orcl/users01.dbf
input datafile file number=00188
name=/u02/app/oracle/oradata/ORCL/pdb_orcl/sysaux01.dbf
input datafile file number=00187
name=/u02/app/oracle/oradata/ORCL/pdb_orcl/system01.dbf
channel ORA_DISK_1: starting piece 1 at 30-03-2016 01:43:09
channel ORA_DISK_1: finished piece 1 at 30-03-2016 01:43:34
piece
handle=/u03/app/oracle/fast_recovery_area/ORCL/2F3B3B64F4EA348DE
0532633960A51C5/backupset/2016_03_30/o1_mf_nnndf_TAG20160330T014
308_chpcssp3_.bkp tag=TAG20160330T014308 comment=NONE
channel ORA_DISK_1: backup set complete, elapsed time: 00:00:25
Finished backup at 30-03-2016 01:43:34
```

```

Starting Control File and SPFILE Autobackup at 30-03-2016
01:43:34
piece
handle=/u03/app/oracle/fast_recovery_area/ORCL/autobackup/2016_0
3_30/o1_mf_s_907811014_chpct9vw_.bkp comment=NONE
Finished Control File and SPFILE Autobackup at 30-03-2016
01:43:41

RMAN> EXIT
$

```

3. You inadvertently delete rows from the table.

- a. Before the delete action, select the current `sysdate`. This date will help you recover the table back to the time before the rows were deleted.

```

$ sqlplus hr@pdb_orcl
Enter password: *****
Connected.
SQL> SELECT timestamp_to_scn(current_timestamp)
        FROM v$database;

2
TIMESTAMP_TO_SCN(CURRENT_TIMESTAMP)
-----
                                55982421

SQL>

```

- b. Delete rows from the table.

```

SQL> DELETE FROM hr.employees WHERE salary < 50000;

4 rows deleted.

SQL> COMMIT;

Commit complete.

SQL> EXIT
$

```

4. Perform the table recovery as the `SYS` user using `/home/oracle/labs/backup_pdb_orcl` as the auxiliary destination using your own SCN number retrieved in step 3.a.

```

$ rm -rf /home/oracle/labs/backup_pdb_orcl
$ mkdir -p /home/oracle/labs/backup_pdb_orcl
$ rman TARGET /

```

```

connected to target database: ORCL (DBID=651773304)

RMAN> RECOVER TABLE hr.employees
        UNTIL SCN 55982421
        AUXILIARY DESTINATION '/home/oracle/labs/backup_pdb_orcl';
Starting recover at 30-03-2016 01:36:32
current log archived
using channel ORA_DISK_1
RMAN-05026: warning: presuming following set of tablespaces
applies to specified point-in-time

List of tablespaces expected to have UNDO segments
Tablespace SYSTEM
Tablespace PDB_ORCL:SYSTEM
Tablespace UNDOTBS1
Tablespace PDB_ORCL:UNDO_1

Creating automatic instance, with SID='beep'
...
auxiliary instance file
/home/oracle/labs/backup_pdb_orcl/ORCL/controlfile/o1_mf_ch8tjzff_.ctl deleted
RMAN-00571: =====
RMAN-00569: ===== ERROR MESSAGE STACK FOLLOWS =====
RMAN-00571: =====
RMAN-03002: failure of recover command at 30-03-2016 01:39:32
RMAN-05063: Cannot recover specified tables
RMAN-05057: Table HR.EMPLOYEES not found

RMAN> EXIT
$

```

*Q/ Why is the table not found?*

***A/ The connection in RMAN connects to the CDB root and not to the PDB where the table exists.***

```

$ rman TARGET sys@pdb_orcl

target database Password: *****
connected to target database: ORCL:PDB_ORCL (DBID=1248312368)

RMAN> RECOVER TABLE hr.employees UNTIL SCN 55982421
        AUXILIARY DESTINATION '/home/oracle/labs/backup_pdb_orcl';

2> 3>

```

```
Starting recover at 24-MAR-16
```

```
RMAN-00571: =====
RMAN-00569: ===== ERROR MESSAGE STACK FOLLOWS =====
RMAN-00571: =====
RMAN-03002: failure of recover command at 30-03-2016 01:40:32
RMAN-07536: command not allowed when connected to a Pluggable
Database

RMAN> EXIT
$
```

*Q/ How will you circumvent the restriction?*

***A/ If you read the “Database Backup and Recovery User's Guide” documentation, it specifies that certain restrictions apply when you connect directly to a pluggable database (PDB). The following operations are not available when you connect as target directly to a PDB:***

- ***Back up archived logs***
- ***Delete archived logs***
- ***Delete archived log backups***
- ***Restore archived logs (RMAN does restore archived logs when required during media recovery.)***
- ***Point-in-time recovery (PITR)***
- ***TSPITR***
- ***Table recovery***
- ***Duplicate database***
- ***Flashback operations***
- ***Running Data Recovery Advisor***
- ***Report/delete obsolete***
- ***Register database***
- ***Import catalog***
- ***Reset database***
- ***Configuring the RMAN environment (using the CONFIGURE command)***

***As the command is not allowed from a PDB, connect to the CDB root and use the following syntax:***

```
$ rman target /
connected to target database: ORCL (DBID=1434985951)

RMAN> RECOVER TABLE hr.employees OF PLUGGABLE DATABASE pdb_orcl
      UNTIL SCN 55982421
      AUXILIARY DESTINATION '/home/oracle/labs/backup_pdb_orcl';
2> 3>

Starting recover at 30-03-2016 01:42:32
```

```

using target database control file instead of recovery catalog
RMAN-00571: =====
RMAN-00569: ===== ERROR MESSAGE STACK FOLLOWS =====
RMAN-00571: =====
RMAN-03002: failure of recover command at 30-03-2016 01:42:56
RMAN-05063: Cannot recover specified tables
RMAN-05112: table "HR"."EMPLOYEES" already exists

RMAN> EXIT
$

```

**Recovering tables or table partitions into a different schema enables you to avoid name conflicts that may be caused by table, constraint, index, or trigger names that already exist in the source schema. The `hr.employees` table will be imported into the `new_hr` schema during the recovery operation. The `new_hr` schema must exist at the time of the recovery operation. The table name could also be renamed during the recovery operation.**

```

$ sqlplus system@pdb_orcl
Enter password: *****
Connected.
SQL> CREATE USER new_hr IDENTIFIED BY oracle_4U
      DEFAULT TABLESPACE users;
      2      3
User created.

SQL> GRANT create session, create table, unlimited tablespace
      TO new_hr;
      2

Grant succeeded.

SQL> EXIT
$

```

```

$ rman target /

connected to target database: ORCL (DBID=1434985951)

RMAN> RECOVER TABLE hr.employees OF PLUGGABLE DATABASE pdb_orcl
      UNTIL SCN 55982421
      AUXILIARY DESTINATION '/home/oracle/labs/backup_pdb_orcl'
      REMAP TABLE hr.employees:new_hr.employees;

Starting recover at 30-03-2016 01:45:32

```

```

using target database control file instead of recovery catalog
current log archived
allocated channel: ORA_DISK_1
channel ORA_DISK_1: SID=866 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 PDB_ORCL:SYSTEM
Tablespace UNDOTBS1
Tablespace PDB_ORCL:UNDO_1

Creating automatic instance, with SID='CEtk'
...
Performing export of tables...
  EXPDP> Starting "SYS"."TSPITR_EXP_hjft_zbld":
  EXPDP> Processing object type TABLE_EXPORT/TABLE/TABLE_DATA
  EXPDP> Processing object type
TABLE_EXPORT/TABLE/STATISTICS/TABLE_STATISTICS
  EXPDP> Processing object type TABLE_EXPORT/TABLE/TABLE
  EXPDP> . . exported "HR"."EMPLOYEES"
6.531 KB          7 rows
  EXPDP> Master table "SYS"."TSPITR_EXP_hjft_zbld" successfully
loaded/unloaded
  EXPDP>
*****
  EXPDP> Dump file set for SYS.TSPITR_EXP_hjft_zbld is:
  EXPDP>
/home/oracle/labs/backup_pdb_orcl/tspitr_hjft_10867.dmp
  EXPDP> Job "SYS"."TSPITR_EXP_hjft_zbld" successfully
completed at Wed Mar 30 04:27:39 2016 elapsed 0 00:00:35
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_hjft_sswj" successfully
loaded/unloaded
IMPDP> Starting "SYS"."TSPITR_IMP_hjft_sswj":
IMPDP> Processing object type TABLE_EXPORT/TABLE/TABLE
IMPDP> Processing object type TABLE_EXPORT/TABLE/TABLE_DATA
IMPDP> . . imported "NEW_HR"."EMPLOYEES"
6.531 KB          7 rows
IMPDP> Processing object type
TABLE_EXPORT/TABLE/STATISTICS/TABLE_STATISTICS
IMPDP> Job "SYS"."TSPITR_IMP_hjft_sswj" successfully
completed at Wed Mar 30 04:27:44 2016 elapsed 0 00:00:02
Import completed

Removing automatic instance
Automatic instance removed
auxiliary instance file
/home/oracle/labs/backup_pdb_orcl/ORCL/2F3B3B64F4EA348DE05326339
60A51C5/datafile/o1_mf_temp_chpo0cnw_.tmp deleted
auxiliary instance file
/home/oracle/labs/backup_pdb_orcl/ORCL/datafile/o1_mf_temp_chpnz
cdw_.tmp deleted
auxiliary instance file
/home/oracle/labs/backup_pdb_orcl/HJFT_PITR_PDB_ORCL_ORCL/online
log/o1_mf_6_chpobjgl_.log deleted
auxiliary instance file
/home/oracle/labs/backup_pdb_orcl/HJFT_PITR_PDB_ORCL_ORCL/online
log/o1_mf_5_chpobjgl_.log deleted
auxiliary instance file
/home/oracle/labs/backup_pdb_orcl/HJFT_PITR_PDB_ORCL_ORCL/online
log/o1_mf_4_chpobjgl_.log deleted
auxiliary instance file
/home/oracle/labs/backup_pdb_orcl/HJFT_PITR_PDB_ORCL_ORCL/2F3B3B
64F4EA348DE0532633960A51C5/datafile/o1_mf_users_chpo7jov_.dbf
deleted
auxiliary instance file
/home/oracle/labs/backup_pdb_orcl/ORCL/2F3B3B64F4EA348DE05326339
60A51C5/datafile/o1_mf_sysaux_chplmw3o_.dbf deleted
auxiliary instance file
/home/oracle/labs/backup_pdb_orcl/ORCL/2F3B3B64F4EA348DE05326339
60A51C5/datafile/o1_mf_system_chplmw9q_.dbf deleted
auxiliary instance file
/home/oracle/labs/backup_pdb_orcl/ORCL/datafile/o1_mf_sysaux_chp
lh2qs_.dbf deleted
auxiliary instance file
/home/oracle/labs/backup_pdb_orcl/ORCL/datafile/o1_mf_undotbs1_c
hplh2z4_.dbf deleted

```

```

auxiliary instance file
/home/oracle/labs/backup_pdb_orcl/ORCL/datafile/o1_mf_system_chp
lh2jj_.dbf deleted
auxiliary instance file
/home/oracle/labs/backup_pdb_orcl/ORCL/controlfile/o1_mf_chplfdc
q_.ctl deleted
auxiliary instance file tspitr_hjft_10867.dmp deleted
Finished recover at 30-03-2016 04:27:465

RMAN> EXIT
$

```

5. Check that the table is fully recovered.

```

$ sqlplus system@pdb_orcl
Enter password: *****

Connected to:

SQL> SELECT * FROM new_hr.employees;

   EMP_ID FIRST_NAME      LAST_NAME      SALARY
-----
    100   Adam          Xue             54320
    101   Joe           Yard             24321
    102   Miles         Wagon          110000
    103   Peter         Allan           954323
    104   Scott         British         44324
    105   Luis           Cole            24325
    106   John          Dan             33326

7 rows selected.

SQL> SELECT * FROM hr.employees;

   EMP_ID FIRST_NAME      LAST_NAME      SALARY
-----
    100   Adam          Xue             54320
    102   Miles         Wagon          110000
    103   Peter         Allan           954323

SQL>

```

*Observe that the original table still exists.*

*Q/ What happened to the unique index `hr.i_un_names`?*

```
SQL> SELECT index_name, table_owner
```

FROM dba_indexes WHERE table_name = 'EMPLOYEES';	
2	3
INDEX_NAME	TABLE_OWNER
-----	-----
PK_EMP_ID	HR
I_UN_NAMES	HR
SQL> EXIT	
\$	

***A/ When you use the **REMAP** clause, named constraints and indexes are not imported. This is to avoid name conflicts with existing tables.***

## Practice 14-4: Moving and Compressing Tables Online

### Overview

In this practice, you will move and compress big tables online.

### Tasks

1. Before starting the practice, execute the `$HOME/labs/RMAN/setup_bigtable.sh` shell script. The script creates in `pdb_orcl` the `test.bigtab` table.

```
$ $HOME/labs/RMAN/setup_bigtable.sh
...
$
```

2. The `test.bigtab` table uses more and more space. Compress it and move it to the `tbs_forctest` tablespace used for test purposes. Users should still be able to work on the table during the operation.
  - a. In another terminal window (*Window2*), connect as `test` and update rows.

```
$ . oraenv
ORACLE_SID = [ORCL] ? ORCL
The Oracle base remains unchanged with value /u01/app/oracle
$ sqlplus test@pdb_orcl
```

```
Enter password: *****
Connected.
SQL> UPDATE test.bigtab SET label = 'DATA FROM bigtab';

10000 rows updated.

SQL>
```

- b. In the first window, move and compress the table.

```
$ sqlplus system@pdb_orcl
Enter password: *****
Connected.
SQL> SELECT bytes, tablespace_name
      FROM dba_segments
      WHERE segment_name = 'BIGTAB';

 2      3
      BYTES TABLESPACE_NAME
-----
393216 USERS

SQL> ALTER TABLE test.bigtab MOVE TABLESPACE tbs_forcompress
      COMPRESS
      ONLINE UPDATE INDEXES;

 2      3
```

- c. Back in Window2, finish the transaction.

```
SQL> COMMIT;

Commit complete.

SQL> EXIT
$
```

- d. In the first window, the DDL statement completes.

```
Table altered.

SQL> SELECT bytes, tablespace_name
       FROM   dba_segments
       WHERE  segment_name = 'BIGTAB';
  2      3
    BYTES TABLESPACE_NAME
-----
196608 TBS_FORCOMPRESS

SQL> EXIT
$
```

*Q/ What is the compress ratio?*

*A/ The compress ratio is around 1 for 2.*



## **Practices for Lesson 15: Oracle Data Pump, SQL\*Loader and External Tables**

### **Chapter 15**

## Practices for Lesson 15: Overview

---

### Practices Overview

In this practice, you will administer and query partitioned external tables.



## Practice 15-1: Querying Partitioned External Tables

---

### Overview

In this practice, you will administrate and query partitioned external tables.

### Tasks

1. Before starting the practice, change the working directory to DP and execute the `$HOME/labs/admin/glogin_15.sh` shell script. The shell script sets formatting for all columns selected in queries.

```
$ cd $HOME/labs/DP
$ $HOME/labs/admin/glogin_15.sh
$
```

2. You received new files containing records about sales. The sales records are dispatched in two files according to the year of the sales.

- `$HOME/labs/DP/sales_1998.dat`
- `$HOME/labs/DP/sales_1999.dat`

In `pdb_orcl`, create the `sh.sales_ext_range` external table partitioned ranged by `time_id`.

```
$ . oraenv
ORACLE_SID = [ORCL] ? ORCL
The Oracle base remains unchanged with value /u01/app/oracle
$ sqlplus system@pdb_orcl
Enter password: *****
Connected.
SQL> CREATE DIRECTORY ext_dir AS '/home/oracle/labs/DP/';

Directory created.

SQL> CREATE DIRECTORY ext_dir2 AS '/home/oracle/labs/DP2/';

Directory created.

SQL> CREATE USER sh IDENTIFIED by oracle_4U;

User created.

SQL> GRANT create session, create table to sh;

Grant succeeded.

SQL> GRANT read, write on DIRECTORY ext_dir to sh;

Grant succeeded.
```

```
SQL> GRANT read, write on DIRECTORY ext_dir2 to sh;

Grant succeeded.

SQL>
```

Use the following code:

```
CREATE TABLE sh.sales_ext_range
(
    time_id          DATE NOT NULL,
    prod_id          INTEGER NOT NULL,
    cust_id          INTEGER NOT NULL,
    channel_id       INTEGER NOT NULL,
    promo_id         INTEGER NOT NULL,
    quantity_sold    NUMBER(10,2),
    amount_sold      NUMBER(10,2)
)
ORGANIZATION EXTERNAL
(
    TYPE ORACLE_LOADER
    DEFAULT DIRECTORY ext_dir
    ACCESS PARAMETERS
    (
        RECORDS DELIMITED BY NEWLINE
        BADFILE 'sh%a_%p.bad'
        LOGFILE 'sh%a_%p.log'
        FIELDS TERMINATED BY ','
        MISSING FIELD VALUES ARE NULL
    )
)
PARALLEL
REJECT LIMIT UNLIMITED
PARTITION by range (time_id)
(
    PARTITION year1998 VALUES LESS THAN (TO_DATE('31-12-1998',
'DD-MM-YYYY')) LOCATION ('sales_1998.dat'),
    PARTITION year1999 VALUES LESS THAN (TO_DATE('31-12-1999',
'DD-MM-YYYY')) LOCATION (ext_dir2:'sales_1999.dat')
);
```

**Q/ Which directories does the external table use?**

**A/ The partitions of the external table use two directories. The default directory for any partition created is `ext_dir`. The last partition uses another directory, `ext_dir2`, which corresponds to the active files for the current sales.**

```

SQL> DROP TABLE sh.sales_ext_range;

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

SQL> CREATE TABLE sh.sales_ext_range
(   time_id          DATE NOT NULL,
    prod_id          INTEGER NOT NULL,
    cust_id          INTEGER NOT NULL,
    channel_id       INTEGER NOT NULL,
    promo_id         INTEGER NOT NULL,
    quantity_sold    NUMBER(10,2),
    amount_sold      NUMBER(10,2)
)
ORGANIZATION EXTERNAL
(
  TYPE ORACLE_LOADER
  DEFAULT DIRECTORY ext_dir
  ACCESS PARAMETERS
  (
    RECORDS DELIMITED BY NEWLINE
    BADFILE 'sh%a_%p.bad'
    LOGFILE 'sh%a_%p.log'
    FIELDS TERMINATED BY ','
    MISSING FIELD VALUES ARE NULL
  )
)
PARALLEL
REJECT LIMIT UNLIMITED
PARTITION by range (time_id)
(
  PARTITION year1998 VALUES LESS THAN (TO_DATE('31-12-1998',
'DD-MM-YYYY')) LOCATION ('sales_1998.dat'),
  PARTITION year1999 VALUES LESS THAN (TO_DATE('31-12-1999',
'DD-MM-YYYY')) LOCATION (ext_dir2:'sales_1999.dat')
);

```

2	3	4	5	6	7	8	9	10	11	12	13	14
15	16	17	18	19	20	21	22	23	24	25	26	27
28	29											

```

Table created.

```

```
SQL>
```

*Q/ How can you check that the locations are correctly set for the partitions?*

*A/ Use the DBA\_XTERNAL\_LOC\_PARTITIONS view.*

```
SQL> SELECT table_name, partition_name, location,
           directory_name
        FROM DBA_XTERNAL_LOC_PARTITIONS;
```

```
2      3
```

TABLE_NAME	PARTITION_NAME	LOCATION	DIRECTORY_NAME
SALES_EXT_RANGE	YEAR1998	sales_1998.dat	
SALES_EXT_RANGE	YEAR1999	sales_1999.dat	EXT_DIR2

```
SQL>
```

*Q/ How do you find the name of the directory of the first file for the first partition of the partitioned external table?*

*A/ Use the DBA\_XTERNAL\_PART\_TABLES view.*

```
SQL> SELECT table_name, type_name, default_directory_name,
           access_parameters
        FROM dba_xternal_part_tables;
```

```
2      3
```

TABLE_NAME	TYPE_NAME	DEFAULT_DIRECTORY_NAME
SALES_EXT_RANGE	ORACLE_LOADER	EXT_DIR
RECORDS	DELIMITED BY NEWLINE	
BADFILE	'sh%a_%p.bad'	
LOGFILE	'sh%a_%p.'	

```
SQL>
```

3. Find the sales of year 1998 and 1999 and all together.

```
SQL> SELECT count(*)
        FROM sh.sales_ext_range PARTITION (year1998);
```

```
2
```

```
COUNT (*)
```

```
-----
357668
```

```
SQL> SELECT count(*)
      FROM sh.sales_ext_range PARTITION (year1999);

2

COUNT (*)
-----
495890

SQL> SELECT count(*) FROM sh.sales_ext_range;

COUNT (*)
-----
853558

SQL>
```

*Q1/ Are you sure that the number of 853558 is the total number of sales over the past two years?*

```
$ wc -l sales_1998.dat
357675 sales_1998.dat
$ wc -l ../DP2/sales_1999.dat
495899 sales_1999.dat
$
```

**A1/ There is a discrepancy between the total number of records read from the two files (853574) and the total number of rows selected.**

*Q2/ Are there any log files generated?*

```
$ ls -l *.log
-rw-r--r-- 1 oracle oinstall 4325 Apr  1 08:04 sh000_4018.log
-rw-r--r-- 1 oracle oinstall 5250 Apr  1 08:04 sh001_4020.log
-rw-r--r-- 1 oracle oinstall 3151 Apr  1 08:04 sh002_4022.log
-rw-r--r-- 1 oracle oinstall 1572 Apr  1 08:04 sh003_4024.log
-rw-r--r-- 1 oracle oinstall 1708 Apr  1 08:04 sh004_4026.log
$
```

**A2/ The information written in the log files is:**

**error processing column TIME\_ID in row 50000 for datafile  
/home/oracle/labs/DP/sales\_1998.dat**

**ORA-01400: cannot insert NULL into (TIME\_ID)error processing  
column TIME\_ID in row 100000 for datafile  
/home/oracle/labs/DP/sales\_1998.dat**

**ORA-01400: cannot insert NULL into (TIME\_ID)**

**A number of records cannot be read because the record contains null values.**

4. A new file with sales for 2000 has arrived. You will add a new partition to the table.

```
SQL> ALTER TABLE sh.sales_ext_range ADD PARTITION year2000
      VALUES LESS THAN (TO_DATE('31-12-2000', 'DD-MM-YYYY'))
      LOCATION (ext_dir2:'sales_2000.dat');
2      3
Table altered.

SQL>
```

5. Count the number of sales for 2000.

```
SQL> SELECT count(*)
      FROM sh.sales_ext_range PARTITION (year2000);
2

COUNT (*)
-----
984128

SQL> host wc -l ../DP2/sales_2000.dat
984256 ../DP2/sales_2000.dat

SQL>
```

*Q1/ Are you sure that the number of 984128 corresponds to the total number of sales for 2000, apart from the issue reported above that some of null rows may be discarded?*

```
SQL> SELECT count(*)
      FROM sh.sales_ext_range
      WHERE time_id <= TO_DATE('31-12-2000', 'DD-MM-YYYY')
      AND    time_id >= TO_DATE('01-01-2000', 'DD-MM-YYYY');
2      3      4
COUNT (*)
-----
465292

SQL> EXIT
$
```

**A2/ No.**

*Q2/ Is this discrepancy related to the same issue as the one reported above?*

**A2/ Reading the content of the /home/oracle/labs/DP2/sales\_2000.dat file, you observe that some of the records do not contain sales for 2000. Be aware that in Oracle Database 12.2, row validation is not supported yet. You must ensure that the records satisfy the partitioning conditions.**

**In this case, create two distinct files, one for 2000 sales and another one for 2001 sales and then add another partition for 2001 sales.**

## **Practices for Lesson 16: In-Memory Column Store**

### **Chapter 16**

## Practices for Lesson 16: Overview

---

### Practices Overview

In the practices for this lesson, you will manage the IM column store size and use new Oracle Database 12.2 performance enhancements for queries on in-memory tables.



## Practice 16-1: Configuring IM Column Store

### Overview

In this practice, you create and load `oe` schema tables in `pdb_orcl`. The IM column store is already configured. You check whether the IM column store is appropriately sized to populate the `oe` in-memory tables.

### Tasks

1. Before starting the practice, execute the `$HOME/labs/admin/glogin_16.sh` shell script. The shell script sets formatting for all columns selected in queries.

```
$ $HOME/labs/admin/glogin_16.sh
$
```

2. Run the `$HOME/labs/IMDB/setup_tables.sh` script that creates in `pdb_orcl` the `oe` user and the `lineorder`, `supplier`, and `date_dim` tables, and finally loads the tables in the IM column store. The `lineorder` table contains 1000000 rows.

```
$ $HOME/labs/IMDB/setup_tables.sh
...
$
```

*It may take some time (5 minutes). Once SQL\*Loader started loading, you can proceed with the next step from another terminal window.*

3. `oe` tables all have a priority set to `CRITICAL`. Check whether all `oe` tables are populated into the IM column store. Compare the `BYTES_NOT_POPULATED` value to the `BYTES` value. A segment is fully populated into the IM column store if `BYTES_NOT_POPULATED` value is 0. When loading is completed, recheck whether all `oe` tables are populated into the IM column store.

```
$ sqlplus system@pdb_orcl
Enter password: *****
Connected.
SQL> SELECT segment_name, bytes, inmemory_size,
           bytes_not_populated, populate_status
        FROM v$im_segments;
 2      3
SEGMENT_NAME      BYTES INMEMORY_SIZE BYTES_NOT_POPULATED
-----
POPULATE_STATUS
-----
CUSTOMER          7708672      2359296              0
COMPLETED

LINEORDER         935067648      91226112      699752448
OUT OF MEMORY

SUPPLIER          2015232      1310720              0
```

```
COMPLETED
```

```
DATE_DIM          352256          1310720          0
COMPLETED
```

```
PART              7798784          2359296          0
COMPLETED
```

```
SQL>
```

*Q1/ Why are the segments not entirely populated into the IM column store?*

**A1/ The result means that the size allocated for the IM column store is not large enough.**

*Q2/ Can you increase the IM column store without shutting down the database instance?*

**A2/ In Oracle Database 12.2, the size of the in-memory area can be dynamically increased after instance startup but not decreased.**

*Q3/ You know that only `pdb_orcl` stores in-memory segments. Is it possible to configure the size for IM column store for `pdb_orcl` so that this PDB can benefit from the maximum allocated size?*

**A3/ Yes, it is possible to configure the size for IM column store for `pdb_orcl` only. The parameter can be set per-PDB to limit the maximum size used by each PDB. You can therefore disable IM column store usage in the other PDBs of the CDB and set the maximum available for `pdb_orcl`.**

5. You want to increase the size of the IM column store in `pdb_orcl`.

*Q/ Which utility can help you get recommendations on IM column store size?*

**A/ The In-Memory Advisor is an interesting tool for two types of recommendations:**

- **Sizing the IM column store**
- **Finding the best candidates for in-memory population**

- a. Use the IM Advisor to get recommendations. The `instimadv.sql` script prompts you for **answers** described in bold below (The password is the same as the one used for all database users).

```
SQL> CONNECT sys@pdb_orcl AS SYSDBA
```

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

```
Connected.
```

```
SQL> @$HOME/labs/IMDB/instimadv
```

```
Welcome to the Oracle Database In-Memory Advisor
(DBMS_INMEMORY_ADVISOR) installation.
```

```
DBMS_INMEMORY_ADVISOR uses Active Session History (ASH),
Automatic Workload Repository (AWR) and optionally SQL Tuning
Sets (STS) to determine which tables, partitions and
subpartitions to place In Memory for optimized analytics
processing performance. DBMS_INMEMORY_ADVISOR produces a
```

recommendation report and a SQLPlus script to implement its recommendations.

This installation script will create user IMADVISON and add object definitions to the schema including the DBMS\_INMEMORY\_ADVISOR package. This installation script creates user IMADVISON using the IDENTIFIED BY password method. If you prefer to use either the IDENTIFIED EXTERNALLY or IDENTIFIED GLOBALLY method, abort this installation by pressing ^C. Then create user IMADVISON using your preferred method. Add no objects to the IMADVISON schema. Then run this installation script again.

These actions will be taken on the database to which you are currently connected.

Please enter the connection ID for the current database?

**PDB\_ORCL**

This installation script creates a new Oracle database user and schema named IMADVISON for the operation of DBMS\_INMEMORY\_ADVISOR...

Please enter the password for user IMADVISON? **\*\*\*\*\***

For confirmation, please re-enter the password for user IMADVISON? **\*\*\*\*\***

Available tablespaces:

TABLESPACE\_NAME

-----

SYSAUX

SYSTEM

TBS\_INMEM

TEMP

Please enter the default, permanent tablespace name for user IMADVISON? **TBS\_INMEM**

Please enter the temporary tablespace name for user IMADVISON?

**TEMP**

The In-Memory Advisor uses the Oracle directory object IMADVISON\_DIRECTORY by default.

If you wish to create the IMADVISON\_DIRECTORY object, please enter an OS host directory path for the IMADVISON\_DIRECTORY object.

If not, please press ENTER to continue.

```
? /home/oracle/labs/IMDB
```

```
Connecting to IMADVISOR @ PDB_ORCL..
```

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

```
Connected.
```

```
No errors.
```

```
No errors.
```

```
No errors.
```

```
No errors.
```

```
No errors.
```

```
No errors.
```

```
No errors.
```

```
No errors.
```

```
No errors.
```

```
No errors.
```

```
No errors.
```

```
No errors.
```

```
No errors.
```

```
No errors.
```

```
No errors.
```

```
DBMS_INMEMORY_ADVISOR installation successful.
```

```
Users who will use the DBMS_INMEMORY_ADVISOR package must be  
GRANTED EXECUTE on the DBMS_INMEMORY_ADVISOR package.
```

```
Please enter a comma separated list of Oracle Database users to  
whom you wish EXECUTE on the DBMS_INMEMORY_ADVISOR package to be  
GRANTED?
```

```
SYS,SYSTEM
```

```
GRANT EXECUTE ON dbms_inmemory_advisor TO SYS
```

```
GRANT EXECUTE ON dbms_inmemory_advisor TO SYSTEM
```

```
While logged in as IMADVISOR or with sufficient privileges, you  
can GRANT EXECUTE ON DBMS_INMEMORY_ADVISOR to additional users  
as needed.
```

```
DBMS_INMEMORY_ADVISOR installation and setup complete.
```

```
Disconnected from Oracle Database 12c Enterprise Edition Release  
12.2.0.1.0- 64bit Production
```

```
$
```

- b. Run the IM Advisor using the `imadvisor_analyze_and_report` SQL script as a user with the privilege to execute the `DBMS_INMEMORY_ADVISOR` package. The script prompts you for **three answers**.

```
$ sqlplus sys@pdb_orcl AS SYSDBA
Enter password: *****
Connected.
SQL> @$HOME/labs/IMDB/imadvisor_analyze_and_report

Specify the IM task name
The IM Advisor generates a report as imadvisor_<taskname>.html
file in the current working directory
The sql file is generated as imadvisor_sql_<taskname>.sql
Enter value for im_task_name: TaskIMADV_1
IM Task name Specified: TaskIMADV_1

Enter begin time for report:

--      Valid input formats:
--      To specify absolute begin time:
--          [MM/DD[/YY]] HH24:MI[:SS]
--      Examples: 02/23/03 14:30:15
--                  02/23 14:30:15
--                  14:30:15
--                  14:30
--      To specify relative begin time: (start with '-' sign)
--          -[HH24:]MI
--      Examples: -1:15  (SYSDATE - 1 Hr 15 Mins)
--                  -25   (SYSDATE - 25 Mins)

Enter value for begin_time: -4:00
Report begin time specified: -4:00

old 102:  lftime_in  := nvl('&&begin_time', '-60');
new 102:  lftime_in  := nvl('-4:00', '-60');
old 104:  :btime := to_char( begin_time,
'&&imadvisor_time_format' );
new 104:  :btime := to_char( begin_time, 'YYYY-MON-DD
HH24:MI:SS.FF' );

PL/SQL procedure successfully completed.

Enter duration in minutes starting from begin time:
```

```

Defaults to SYSDATE - begin_time

Enter value for duration: 5
Report duration specified: 5
old 8:  begin_time := to_timestamp(:btime,
'&&imadvisor_time_format');
new 8:  begin_time := to_timestamp(:btime, 'YYYY-MON-DD
HH24:MI:SS.FF');
old 18:  duration := nvl('&&duration', since_begin_time);
new 18:  duration := nvl('5', since_begin_time);
old 29:  :etime := to_char( end_time,
'&&imadvisor_time_format' );
new 29:  :etime := to_char( end_time, 'YYYY-MON-DD
HH24:MI:SS.FF' );

PL/SQL procedure successfully completed.

Using 2016-APR-05 21:42:08.000000000 as report begin time
Using 2016-APR-05 21:47:08.000000000 as report end time

1 row selected.

IM Advisor: Adding Statistics..
DECLARE
*
ERROR at line 1:
ORA-13605: The specified task or object SQLACCESS_GENERAL does
not exist for the current user.
...
ORA-06512: at line 9

Disconnected from Oracle Database 12c Enterprise Edition Release
12.2.0.1.0- 64bit Production
$

```

**Q1/ Which output files are generated?**

**A1/ None are produced because there was no activity in the PDB and therefore recorded in the AWR. Two output files would have been generated if there would have been enough AWR data: the report and the DDL script.**

**But the execution could have completed with the following:**

```

IMADVISOR: Finished Adding Statistics
IMADVISOR: Finished Executing the task
IM Advisor: Generating Recommendations..
imadvisor_TaskIMADV_1.html

```

```

imadvisor_sql_TaskIMADV_1.html
imadvisor_object_TaskIMADV_1.html
imadvisor_TaskIMADV_1.sql
Enter value for recommendation_directory: /home/oracle/labs/IMDB
Placing recommendation files in directory /home/oracle/labs/IMDB
IM Advisor generated report in imadvisor_TaskIMADV_1.html
IM Advisor generated DDL script in imadvisor_TaskIMADV_1.sql

```

*Q2/ What could the report suggest?*

**A2/ The report could suggest the following:**

*At the top is a summary of the Total Database Time analyzed in the report, and which percentage of that Database Time can be ascribed to Analytics Processing.*

*The section labeled 'In-Memory Sizes' contains a table giving possible In-Memory sizes and the estimated benefit from each of those In-Memory sizes.*

*Below this section there is a table summarizing the recommendations of the top objects to place in memory with their recommended compression type and estimated benefit.*

*Finally, at the bottom of the report there is a table of information about the database where the Advisor was run.*

*The main report links to an additional page which lists all the objects recommended to be placed In-Memory with the recommended compression type and estimated benefit.*

*Q3/ What does the DDL script suggest?*

**A3/ The DDL script contains SQL statements which can be run on the target database to modify the objects recommended to be placed in the in-memory column store along with the recommended compression types. You can modify the DDL to fine tune non-default parameters or select a different compression method.**

c. Increase the IM column store.

```

$ sqlplus sys@pdb_orcl AS SYSDBA
Enter password: *****
Connected.
SQL> ALTER SYSTEM SET inmemory_size=1500M;
ALTER SYSTEM SET inmemory_size=1500M
*
ERROR at line 1:
ORA-02097: parameter cannot be modified because specified value
is invalid
ORA-02095: specified initialization parameter cannot be modified
SQL>

```

*Q1/ Does the error message say that the specified initialization parameter is not modifiable?*

**A1/ No. It says that the value is invalid and therefore the parameter cannot be modified.**

*Q2/ Why is the value invalid? Check the IM size allocated for the CDB.*

```
SQL> CONNECT / AS SYSDBA
Connected.
SQL> SHOW PARAMETER inmemory_size
NAME                                TYPE                                VALUE
-----
inmemory_size                       big integer 128M
SQL>
```

**A2/ The IM size allocated for the CDB is 128M.**

**Q3/ What do you conclude?**

**A3/ The IM size per-PDB cannot be higher than the IM size allocated for the CDB.**

```
SQL> ALTER SYSTEM SET inmemory_size=1500M SCOPE=BOTH;

System altered.

SQL> CONNECT sys@pdb_orcl AS SYSDBA
Enter password: *****
Connected.
SQL> SHOW PARAMETER inmemory_size
NAME                                TYPE                                VALUE
-----
inmemory_size                       big integer 1500M
SQL>
```

**Observe that the PDB inherits the CDB parameter value.**

**Q4/ Can the sum of IM sizes of all PDBs be higher than the IM size allocated for the CDB?**

```
SQL> SHOW PDBs
CON_ID CON_NAME                                OPEN MODE RESTRICTED
-----
3 PDB_ORCL                                READ WRITE NO
SQL> CONNECT / AS SYSDBA
Connected.
SQL> CREATE PLUGGABLE DATABASE pdb_im
      ADMIN USER admin IDENTIFIED BY oracle_4U
      ROLES=(CONNECT)
      FILE_NAME_CONVERT=
        ('/u02/app/oracle/oradata/ORCL/pdbseed',
         '/u02/app/oracle/oradata/ORCL/pdb_im');
2      3      4      5      6
Pluggable database created.

SQL> CONNECT sys@pdb_im AS SYSDBA
```



```
Enter password: *****
Connected.
SQL> SHOW PARAMETER inmemory_size
```

NAME	TYPE	VALUE
inmemory_size	big integer	1500M

```
SQL>
```

**A4/ The sum of all PDBs can be greater than the IM size allocated to the CDB but if a PDB uses all the allocated IM space allocated for the CDB, then other PDBs will not be able to use any IM space because there will not be any left. By default, each PDB inherits the CDB value.**

- d. Once there is enough space in the IMCS, you do not need to re-execute the query to have the remaining bytes left out the IMCS populated into the IM column store. If the background process in charge of repopulation does not wake up fast, reselect from oe.lineorder. (SELECT count(\*) FROM oe.Lineorder;)

```
SQL> CONNECT sys@pdb_orcl AS SYSDBA
```

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

```
Connected.
```

```
SQL> SELECT segment_name, bytes, inmemory_size,
           bytes_not_populated
       FROM v$im_segments;
```

2	3			
SEGMENT_NAME	BYTES	INMEMORY_SIZE	BYTES_NOT_POPULATED	
PART	7798784	2359296	0	
CUSTOMER	7708672	2359296	0	
LINEORDER	935067648	266207232	233963520	
SUPPLIER	2015232	1310720	0	
DATE_DIM	352256	1310720	0	

```
SQL> SELECT segment_name, bytes, inmemory_size,
           bytes_not_populated
       FROM v$im_segments;
```

2	3			
SEGMENT_NAME	BYTES	INMEMORY_SIZE	BYTES_NOT_POPULATED	
PART	7798784	2359296	0	
CUSTOMER	7708672	2359296	0	
LINEORDER	935075840	311492608	116981760	
SUPPLIER	2015232	1310720	0	
DATE_DIM	352256	1310720	0	

```

SQL> SELECT segment_name, bytes, inmemory_size,
           bytes_not_populated
       FROM v$sql_segments;

```

2	3			
SEGMENT_NAME	BYTES	INMEMORY_SIZE	BYTES_NOT_POPULATED	
-----	-----	-----	-----	
PART	7798784	2359296	0	
CUSTOMER	7708672	2359296	0	
LINEORDER	935067648	356581376	0	
SUPPLIER	2015232	1310720	0	
DATE_DIM	352256	1310720	0	

```

SQL> EXIT
$

```

## Practice 16-2: Optimizing Queries by Using Join Groups

### Overview

In this practice, you will optimize queries on in-memory tables by creating join groups on columns frequently used in join.

### Tasks

1. You are aware of performance issues on the following query on two in-memory tables:

```
$ sqlplus oe@pdb_orcl
Enter password: *****
Connected.
SQL> SELECT /*+ monitor*/
          SUM(lo_extendedprice * lo_discount) revenue
        FROM   oe.lineorder l, oe.date_dim d
        WHERE  l.lo_orderdate = d.d_datekey;
 2          3          4
          REVENUE
-----
113585171676607

SQL>
```

*Q/ Which is the Oracle Database 12.2 new feature that helps improve the performance in queries on joined columns?*

*A/ Creating a join group with the columns frequently joined. A new SQL object is created that lists columns belonging to the same domain.*

2. Create the join group on the two frequently joined columns.

```
SQL> CREATE INMEMORY JOIN GROUP j1
          (lineorder(lo_orderdate), date_dim(d_datekey));
 2
Join group created.

SQL> SELECT joingroup_name, table_name, column_name, flags
        FROM   dba_joingroups;
 2
JOINGROUP_NAME    TABLE_NAME    COLUMN_NAME    FLAGS
-----
J1                LINEORDER     LO_ORDERDATE  MASTER
J1                DATE_DIM      D_DATEKEY
```

*Q1/ What can the `FLAGS` column values mean?*

*A1/ Master value indicates the primary column.*

Q2/ According to the number of values in both tables, which table is the most appropriate one to be the master table in the join group?

**A2/ The date\_dim table is a dimension table typically holding many fewer date values than the lineorder table. Therefore the date\_dim table would be more appropriate to be the master table.**

```
SQL> DROP INMEMORY JOIN GROUP j1;

Join group deleted.

SQL> CREATE INMEMORY JOIN GROUP j1
      (date_dim(d_datekey), lineorder(lo_orderdate));
2
Join group created.

SQL> SELECT joingroup_name, table_name, column_name, flags
      FROM   dba_joingroups;
2
```

JOINGROUP_NAME	TABLE_NAME	COLUMN_NAME	FLAGS
J1	LINEORDER	LO_ORDERDATE	
J1	DATE_DIM	D_DATEKEY	MASTER

```
SQL>
```

Q3/ If a third column is added to the join group, will it be the primary column?

**A3/ No. There is only one primary column because there is only one master table defined when the join group is created.**

- Verify that the join group is used in the execution plan.

```
SQL> SELECT /*+ monitor*/
      SUM(lo_extendedprice * lo_discount) revenue
      FROM   oe.lineorder l, oe.date_dim d
      WHERE  l.lo_orderdate = d.d_datekey;
2      3      4
      REVENUE
-----
113585171676607

SQL>
```

- Drop the join group.

Q1/ Is the join group removed if all columns participating in the join groups are removed from the join group?

```
SQL> ALTER INMEMORY JOIN GROUP j1 REMOVE
      (date_dim(d_datekey));
2
```

Join group altered.

```
SQL> ALTER INMEMORY JOIN GROUP j1 REMOVE
                                     (lineorder(lo_orderdate)) ;
*
ERROR at line 1:
ORA-00944: insufficient number of clustered columns

SQL>
```

**A1/ A join group must contain at least one column, the primary column.**

**Q2/ If you remove a column from the join group, does it necessarily have to be the primary column?**

**A2/ No. Any column participating to the join group can be removed. The last one being the single one becomes the primary column.**

```
SQL> SELECT joingroup_name, table_name, column_name, flags
       FROM   dba_joingroups;
  2

```

JOINGROUP_NAME	TABLE_NAME	COLUMN_NAME	FLAGS
J1	LINEORDER	LO_ORDERDATE	MASTER

```
SQL>
```

Now, drop the join group.

```
SQL> DROP INMEMORY JOIN GROUP j1;
```

Join group deleted.

```
SQL> SELECT * FROM dba_joingroups;
```

no rows selected

```
SQL> SELECT * FROM sys.im_domain$;
```

no rows selected

```
SQL>
```

**Observe that the domains created with the join groups are automatically dropped.**

## Practice 16-3: Optimizing Queries by Capturing Expressions in IM Column Store

### Overview

In this practice, you will optimize queries on in-memory tables that frequently use the same expressions.

### Tasks

1. You are aware of performance issues on the following query on two in-memory tables using expressions:

```
SQL> CONNECT oe@pdb_orcl
Enter password: *****
Connected.
SQL> SELECT lo_shipmode, sum(lo_quantity) as sum_qty,
           sum(lo_extendedprice) as sum_base_price,
           sum(lo_extendedprice * (1 - lo_discount))
           as sum_disc_price,
           sum(lo_extendedprice*(1-lo_discount)*(1+lo_tax))
           as charge,
           avg(lo_quantity) as avg_qty,
           avg(lo_extendedprice) as avg_price,
           avg(lo_discount) as avg_disc,
           count(*) as count_order
FROM   oe.lineorder l , oe.date_dim d
WHERE  l.lo_orderdate = d.d_datekey
AND    to_date(d.d_datekey , 'YYYY-MM-DD')
      <= to_date('1998-12-31', 'YYYY-MM-DD')-90
GROUP BY lo_shippriority, lo_shipmode
ORDER BY lo_shippriority, lo_shipmode;
```

LO_SHIPMOD	SUM_QTY	SUM_BASE_PRICE	SUM_DISC_PRICE	CHARGE	AVG_QTY	AVG_PRICE	AVG_DISC	COUNT_ORDER
AIR	21583356	3238404549310	-12952791362931					
-2820183473762731	25.6122986	3842914.16	5.0017266					842695
FOB	21636535	3248099653561	-13002216333227					
-2828946332277405	25.477797	3824753.99	5.0022067					849231
MAIL	21598723	3237642563785	-13005162207675					

```

-2832405750949590 25.5536937 3830491.58 5.01135905      845229

RAIL              21670268      3254571198827 -12999270864911
-2830962014776165 25.5996021 3844702.23 4.99965269      846508

REG AIR          21599383      3238494700156 -12941651260852
-2816661705717778 25.5436857 3829882.12 4.99100505      845586

SHIP             21659340      3247673682221 -12981537074098
-2830694440122772 25.5218109 3826825.45  4.993769      848660

TRUCK            21608396      3236770512921 -12978828891218
-2828998093912544 25.5043063 3820347.73 5.00818771      847245

7 rows selected.

SQL>

```

*Q1/ Which is the Oracle Database 12.2 new feature that helps improve the performance in queries on joined columns?*

***A1/ Caching expressions results that are computationally intensive to evaluate, in IM column store, is a good way to improve query performance.***

*Q2/ How are in-memory expressions (IMEs) considered as good candidates?*

***A2/ Statistics such as frequency of execution, cost of evaluation on a per-segment basis, regularly maintained by the optimizer and stored in the Expression Statistics Store (ESS) help in identification and evaluation of IM expressions in the IM column store.***

2. Configure the CDB so that the optimizer detects the query expressions as good IMEs and store the results in in-memory expression units (IMEUs).

Specifies a snapshot that defines the period of time from which expression statistics are considered. You can specify either of the following values:

- **CUMULATIVE:** The database considers all expression statistics since the creation of the database.
- **CURRENT:** The database considers only expressions statistics from the past 24 hours.

```

SQL> CONNECT / AS SYSDBA
Connected.
SQL> EXEC DBMS_INMEMORY_ADMIN.IME_CAPTURE_EXPRESSIONS('CURRENT')

PL/SQL procedure successfully completed.

SQL>

```

- Execute the queries several times and check whether the expressions results are stored in the In-Memory column store.

```
SQL> CONNECT oe@pdb_orcl
Enter password: *****
Connected.
SQL> SELECT lo_shipmode, sum(lo_quantity) as sum_qty,
           sum(lo_extendedprice) as sum_base_price,
           sum(lo_extendedprice * (1 - lo_discount))
           as sum_disc_price,
           sum(lo_extendedprice*(1-lo_discount)*(1+lo_tax))
           as charge,
           avg(lo_quantity) as avg_qty,
           avg(lo_extendedprice) as avg_price,
           avg(lo_discount) as avg_disc,
           count(*) as count_order
FROM   oe.lineorder l , oe.date_dim d
WHERE  l.lo_orderdate = d.d_datekey
AND    to_date(d.d_datekey , 'YYYY-MM-DD')
       <= to_date('1998-12-31','YYYY-MM-DD')-90
GROUP BY lo_shippriority, lo_shipmode
ORDER BY lo_shippriority, lo_shipmode;
```

LO_SHIPMOD	SUM_QTY	SUM_BASE_PRICE	SUM_DISC_PRICE	CHARGE	AVG_QTY	AVG_PRICE	AVG_DISC	COUNT_ORDER
AIR	21583356	3238404549310	-12952791362931					
-2820183473762731	25.6122986	3842914.16	5.0017266					842695
FOB	21636535	3248099653561	-13002216333227					
-2828946332277405	25.477797	3824753.99	5.0022067					849231
MAIL	21598723	3237642563785	-13005162207675					
-2832405750949590	25.5536937	3830491.58	5.01135905					845229
RAIL	21670268	3254571198827	-12999270864911					
-2830962014776165	25.5996021	3844702.23	4.99965269					846508
REG AIR	21599383	3238494700156	-12941651260852					
-2816661705717778	25.5436857	3829882.12	4.99100505					845586



```

SHIP          21659340      3247673682221  -12981537074098
-2830694440122772 25.5218109 3826825.45   4.993769      848660

TRUCK          21608396      3236770512921  -12978828891218
-2828998093912544 25.5043063 3820347.73  5.00818771     847245

7 rows selected.

SQL> SELECT expression_text FROM user_expression_statistics;

EXPRESSION_TEXT
-----
"D_DATEKEY"
"LO_EXTENDEDPRICE"
"LO_DISCOUNT"
"LO_EXTENDEDPRICE"*"LO_DISCOUNT"
"LO_ORDERDATE"

SQL>

```

**Remark:** The background process may wake up in a few minutes. Be patient.

```

SQL> SELECT * FROM user_im_expressions;

no rows selected

SQL>

```

*Q/ Is there any IM expressions attached to the `LINEORDER` table?*

***A/ No. There are some statistics but not enough of these to consider the expressions as internal virtual columns associated to the table.***

4. Because you cannot wait for the detection of eligible expressions based on ESS statistics due the short period of time when the query and expressions were used, you decide to control the creation of virtual columns for each expressions.
  - a. Set the appropriate instance parameter to control virtual columns usage.

```

SQL> CONNECT / AS SYSDBA
Connected.
SQL> ALTER SYSTEM SET inmemory_virtual_columns = ENABLE
SCOPE=both;

2
System altered.

SQL>

```

- b. To better explain the virtual columns usage, you will create a simple test case.

```
SQL> CONNECT oe@pdb_orcl
Enter password: *****
Connected.
SQL> CREATE TABLE oe.test (
        c1 number, c2 number, vc1 AS (c1+c2), vc2 AS (c1*2))
        INMEMORY;
      2      3
Table created.

SQL> INSERT INTO oe.test (c1, c2) VALUES (1,2);

1 row created.

SQL> INSERT INTO oe.test (c1, c2) VALUES (2,3);

1 row created.

SQL> COMMIT;

Commit complete.

SQL>
```

- c. Query the in-memory tables.

```
SQL> SELECT lo_shipmode, sum(lo_quantity) as sum_qty,
        sum(lo_extendedprice) as sum_base_price,
        sum(lo_extendedprice * (1 - lo_discount))
        as sum_disc_price,
        sum(lo_extendedprice*(1-lo_discount)*(1+lo_tax))
        as charge,
        avg(lo_quantity) as avg_qty,
        avg(lo_extendedprice) as avg_price,
        avg(lo_discount) as avg_disc,
        count(*) as count_order
  FROM   oe.lineorder l , oe.date_dim d
 WHERE  l.lo_orderdate = d.d_datekey
 AND    to_date(d.d_datekey , 'YYYY-MM-DD')
        <= to_date('1998-12-31', 'YYYY-MM-DD')-90
 GROUP BY lo_shippriority, lo_shipmode
 ORDER BY lo_shippriority, lo_shipmode;
      2      3      4      5      6      7      8      9     10     11     12     13     14
15     16
```

LO_SHIPMOD	SUM_QTY	SUM_BASE_PRICE	SUM_DISC_PRICE	
CHARGE	AVG_QTY	AVG_PRICE	AVG_DISC	COUNT_ORDER
AIR	21583356	3238404549310	-12952791362931	
-2820183473762731	25.6122986	3842914.16	5.0017266	842695
FOB	21636535	3248099653561	-13002216333227	
-2828946332277405	25.477797	3824753.99	5.0022067	849231
MAIL	21598723	3237642563785	-13005162207675	
-2832405750949590	25.5536937	3830491.58	5.01135905	845229
RAIL	21670268	3254571198827	-12999270864911	
-2830962014776165	25.5996021	3844702.23	4.99965269	846508
REG AIR	21599383	3238494700156	-12941651260852	
-2816661705717778	25.5436857	3829882.12	4.99100505	845586
SHIP	21659340	3247673682221	-12981537074098	
-2830694440122772	25.5218109	3826825.45	4.993769	848660
TRUCK	21608396	3236770512921	-12978828891218	
-2828998093912544	25.5043063	3820347.73	5.00818771	847245

7 rows selected.

SQL> **SELECT \* FROM oe.test;**

C1	C2	VC1	VC2
1	2	3	2
2	3	5	4

SQL>

5. Check if IMEs results are now captured.

- Execute the \$HOME/labs/IMDB/loop.sql SQL script that executes the same queries several times.

SQL> @\$HOME/labs/IMDB/loop

...

SQL>

- b. You will have to wait until w00 background slaves flush the ESS statistics from SGA to disk unless you execute the `dbms_stats.flush_database_monitoring_info`.

```
SQL> EXEC dbms_stats.flush_database_monitoring_info

PL/SQL procedure successfully completed.

SQL> SELECT column_name, sql_expression
       FROM   v$sql_mecol_cu;
2
COLUMN_NAME  SQL_EXPRESSION
-----
VC1          "C1"+"C2"
VC2          "C1"*2

SQL> SELECT expression_text, evaluation_count
       FROM   user_expression_statistics;
2
EXPRESSION_TEXT                                EVALUATION_COUNT
-----
"D_DATEKEY"                                     118640
TO_DATE(TO_CHAR("D_DATEKEY"), 'YYYY-MM-DD')    56764
"LO_SHIPMODE"                                   137387050
"LO_EXTENDEDPRICE"*(1-"LO_DISCOUNT")          136278542
"LO_QUANTITY"                                   137387050
"LO_EXTENDEDPRICE"                             153406490
SYS_OP_BLOOM_FILTER(:BF0000,"LO_ORDERDATE")    137387050
"LO_TAX"                                         137387050
"LO_ORDERDATE"                                  16076204
1+"LO_TAX"                                       136278542
"LO_DISCOUNT"                                 153406490
"LO_SHIPPRIORITY"                              137387050
"LO_EXTENDEDPRICE"*"LO_DISCOUNT"              11852572
"C1"*2                                           30
"C1"                                              60
"C2"                                              60
"C1"+"C2"                                         30

17 rows selected.

SQL>
```

*Q/ What does the list contain?*

**A/ The list contains the references to all columns participating to expressions in the query above. For example, `lo_tax` belongs to the `(1+lo_tax)` expression,**

*lo\_extendedprice \* (1 - lo\_discount) belongs to  
sum(lo\_extendedprice \* (1 - lo\_discount)) and to  
sum(lo\_extendedprice\*(1-lo\_discount)\*(1+lo\_tax)).*

```
SQL> SELECT * FROM user_im_expressions;
```

no rows selected

```
SQL>
```

There is still not enough statistics to consider the expressions as eligible IM expressions.

- c. Look at new statistics related to IMEs and more precisely to expression units (IMEUs).

```
SQL> SELECT display_name, value
        FROM   v$mystat m, v$statname n
        WHERE  m.statistic# = n.statistic#
        AND    display_name like 'IM scan EU %';
```

2	3	4	
DISPLAY_NAME			VALUE
IM scan EU bytes in-memory			2296
IM scan EU bytes uncompressed			168
IM scan EU rows			28

```
SQL> EXIT
```

```
$
```

## Practice 16-4: Evicting Tables of IM Column Store Based on Heat Map Statistics

---

### Overview

You are very busy with performance tuning and cannot manually manage the content of the In-Memory Column Store.

In this practice, you will use automatic data optimization (ADO) to allow Oracle automatic eviction of in-memory tables from the IM column store that are not accessed for some time. This enhancement will help you a lot in automatic management of the content of the In-Memory Column Store.

### Tasks

1. The only thing you have to set is to enable heat map statistics collection. First, clear up all heat map statistics that may exist on any segment.

```
$ sqlplus / AS SYSDBA
Enter password: *****
Connected.
SQL> EXEC dbms_ilm_admin.clear_heat_map_all

PL/SQL procedure successfully completed.

SQL> ALTER SYSTEM SET heat_map = ON;

System altered.

SQL>
```

2. Enable an ADO policy on the `oe.lineorder` in-memory table so that there is an automatic eviction of the table from the IM column store when the table is not accessed after 1 day.

```
SQL> CONNECT oe@pdb_orcl
Enter password: *****
Connected.
SQL> ALTER TABLE oe.supplier ILM DELETE_ALL;

Table altered.

SQL> ALTER TABLE oe.supplier ILM ADD POLICY
        NO INMEMORY SEGMENT AFTER 1 DAYS OF NO ACCESS;
        2
Table altered.

SQL>
```

*Q/ Which were the possible automatic actions handled by ADO until Oracle Database 12.2?*

**A/ The automatic actions were either *STORAGE* or *COMPRESSION*. A new automatic action can be *EVICT* which means *INMEMORY STORE EVICTION*.**

```
SQL> SELECT action_type, condition_type, object_name
      FROM dba_ilmdatamovementpolicies m, dba_ilmobjects o
      WHERE m.policy_name = o.policy_name;
```

```
2      3
ACTION_TYPE CONDITION_TYPE          OBJECT_NAME
-----
EVICT          LAST ACCESS TIME      SUPPLIER
```

```
SQL> SELECT segment_name, bytes, inmemory_size,
      bytes_not_populated
      FROM v$im_segments
      WHERE segment_name = 'SUPPLIER';
```

```
2      3      4
SEGMENT_NAME          BYTES INMEMORY_SIZE BYTES_NOT_POPULATED
-----
SUPPLIER              2015232          1310720              0
```

```
SQL>
```

3. You cannot wait until the 1 day delay is over. Indicate that the policy is specified in seconds rather than in days with the following customization.

```
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>
```

4. **Wait until two minutes have passed without any modification on the `oe.supplier` table and check that the table is evicted from the IM column store.**
  - a. For the purpose of the demo, you will not wait until MMON evaluates the ADO policies. Execute the `$HOME/labs/IMDB/ado.sql` script to force evaluation of the policy and immediate execution of an ADO task. Wait a few minutes or seconds that the background process in charge of the execution wakes up.

```
SQL> @$HOME/labs/IMDB/ado.sql
...
SQL> SELECT task_id, state
      FROM dba_ilmtasks WHERE task_owner = 'OE';
```

```
2
TASK_ID STATE
-----
3 COMPLETED
```

```
SQL> SELECT task_id, object_name, selected_for_execution
      FROM dba_ilmevaluationdetails
      WHERE object_name = 'SUPPLIER';
```

```
2      3
      TASK_ID OBJECT_NAME      SELECTED_FOR_EXECUTION
-----
      3 SUPPLIER      SELECTED FOR EXECUTION
```

```
SQL>
```

**Q1/ Can the segment be repopulated in the IM column store?**

```
SQL> SELECT inmemory, table_name FROM user_tables;
```

```
INMEMORY TABLE_NAME
-----
ENABLED   LINEORDER
ENABLED   PART
DISABLED SUPPLIER
ENABLED   CUSTOMER
ENABLED   DATE_DIM
ENABLED   TEST
```

```
6 rows selected.
```

```
SQL>
```

**A1/ Even if a user executes queries on the segment, the segment will not be repopulated in the IM column store.**

**Q2/ How can you enable the segment to be repopulated into the IM column store?**

**A2/ The ADO eviction policy disabled the attribute on the segment. Reenable the INMEMORY attribute for the segment.**

```
SQL> ALTER TABLE oe.supplier INMEMORY PRIORITY HIGH;
```

```
Table altered.
```

```
SQL> SELECT count(*) FROM oe.supplier;
```



```

COUNT (*)
-----
      16000

SQL> SELECT segment_name, bytes, inmemory_size,
           bytes_not_populated
        FROM v$sql_segments;

SEGMENT_NAME          BYTES INMEMORY_SIZE BYTES_NOT_POPULATED
-----
CUSTOMER              7708672      2359296              0
LINEORDER             935067648     464584704             0
SUPPLIER              2015232      1310720              0
PART                  7798784      2359296              0
DATE_DIM              352256       1310720              0

SQL> EXIT
$

```



# **Practices for Lesson 17: SQL Tuning Enhancements**

## **Chapter 17**

## Practices for Lesson 17: Overview

---

### Practices Overview

In the practices for this lesson, you will use the Optimizer Statistics Advisor to improve the quality of statistics gathering. You will also use new SPA parameters.

## Practice 17-1: Configuring SPM to Capture Specific Statement Plans

### Overview

SQL Plan Management (SPM) is an Oracle Database 11g feature that provides controlled execution plan evolution. With SPM, the optimizer automatically manages execution plans and ensures that only known or verified plans are used. When a new plan is found for a SQL statement, it will not be used until it has been verified to have comparable or better performance than the current plan. There are three steps for a baseline to be accepted: capture, selection, and verification.

Oracle Database 12.2 introduces new configuration parameters for capture setting.

### Tasks

1. Before starting the practice, execute the `$HOME/labs/admin/glogin_17.sh` shell script. The shell script sets formatting for all columns selected in queries. In a previous practice, you ran the `$HOME/labs/IMDB/setup_tables.sh` script that creates the `pdb_orcl`, then the `oe` user, the `lineorder`, `supplier`, and `date_dim` tables and finally loads the tables with data. Execute the `$HOME/labs/TUNING/setup_sh.sh` shell script. The script creates the `sh.sales` partitioned table in `pdb_orcl`.

```
$ $HOME/labs/admin/glogin_17.sh
$ $HOME/labs/TUNING/setup_sh.sh
...
$
```

2. The first component of SPM is Plan Capture. There are two main ways to capture plans: automatically (on the fly), or bulk load. In this practice you turn on automatic plan capture so that the SPM repository is automatically populated for any repeatable SQL statement. Turn on automatic plan capture by setting the initialization parameter `optimizer_capture_sql_plan_baselines` to true at the instance level.

```
$ . oraenv
ORACLE_SID = [ORCL] ? ORCL
The Oracle base has been set to /u01/app/oracle
$ sqlplus system@pdb_orcl
Enter password: *****
Connected.
SQL> ALTER SYSTEM SET
          optimizer_capture_sql_plan_baselines = TRUE;
          2
System altered.

SQL>
```

3. Execute the `$HOME/labs/TUNING/query1.sql` script. The script executes a repetitive query on `oe` tables and then a repetitive query on `sh` tables.

```
SQL> @$HOME/labs/TUNING/query1.sql
SQL> SELECT /*LOAD_AUTO*/ lo_shipmode, sum(lo_quantity) as
sum_qty,
```

```

2          sum(lo_extendedprice) as sum_base_price,
3          sum(lo_extendedprice * (1 - lo_discount))
4          as sum_disc_price,
5          sum(lo_extendedprice*(1-
lo_discount)*(1+lo_tax))
6          as charge,
7          avg(lo_quantity) as avg_qty,
8          avg(lo_extendedprice) as avg_price,
9          avg(lo_discount) as avg_disc,
10         count(*) as count_order
11 FROM    oe.lineorder l , oe.date_dim d
12 WHERE   l.lo_orderdate = d.d_datekey
13 AND     to_date(d.d_datekey , 'YYYY-MM-DD')
14         <= to_date('1998-12-31','YYYY-MM-DD')-90
15 GROUP BY lo_shippriority, lo_shipmode
16 ORDER BY lo_shippriority, lo_shipmode;

```

LO_SHIPMOD	SUM_QTY	SUM_BASE_PRICE	SUM_DISC_PRICE	CHARGE
-----	-----	-----	-----	-----
AVG_QTY	AVG_PRICE	AVG_DISC	COUNT_ORDER	
-----	-----	-----	-----	
AIR	21583356	3.2384E+12	-1.295E+13	-2.820E+15
25.6122986	3842914.16	5.0017266	842695	
FOB	21636535	3.2481E+12	-1.300E+13	-2.829E+15
25.477797	3824753.99	5.0022067	849231	
MAIL	21598723	3.2376E+12	-1.301E+13	-2.832E+15
25.5536937	3830491.58	5.01135905	845229	
RAIL	21670268	3.2546E+12	-1.300E+13	-2.831E+15
25.5996021	3844702.23	4.99965269	846508	
REG AIR	21599383	3.2385E+12	-1.294E+13	-2.817E+15
25.5436857	3829882.12	4.99100505	845586	
SHIP	21659340	3.2477E+12	-1.298E+13	-2.831E+15
25.5218109	3826825.45	4.993769	848660	
TRUCK	21608396	3.2368E+12	-1.298E+13	-2.829E+15
25.5043063	3820347.73	5.00818771	847245	

7 rows selected.

...

```
SQL> SELECT /*LOAD_AUTO*/ count(*)
      2      FROM    sh.sales
      3      WHERE   time_id <= TO_DATE('1998-12-31', 'SYYYY-MM-
DD');
```

COUNT (\*)

-----

178834

...

SQL>

*Q1/ Did the capture load plans for both queries?*

```
SQL> SELECT sql_text, sql_handle, origin
      FROM    dba_sql_plan_baselines
      WHERE   sql_text like 'SELECT /*LOAD_AUTO*/%';
```

2 3

SQL\_TEXT

-----

SQL\_HANDLE ORIGIN

-----

SELECT /\*LOAD\_AUTO\*/ count(\*)

FROM sh.sales

WHERE time\_id <= TO\_DAT

SQL\_0ae456f7f2328903 AUTO-CAPTURE

SELECT /\*LOAD\_AUTO\*/ lo\_shipmode, sum(lo\_quantity) as sum\_qty,

sum(l

SQL\_82f54579a9471356 AUTO-CAPTURE

SQL>

**A1/ Yes. It did. This is the default behavior of automatic capturing.**

*Q2/ How would you set automatic capture to capture plans that query sh.sales table only?*

**A2/ There are three new parameters to filter the capture. The one needed for this case is `AUTO_CAPTURE_SQL_TEXT`. Multiple filters are permitted for each filter type, and adding a new filter will not override previous ones. If multiple `allow = TRUE`, filters are specified for a single filter type, only one must be satisfied. All `allow = FALSE` filters must be satisfied.**

```

SQL> variable cnt number;
SQL> EXEC :cnt :=
dbms_spm.drop_sql_plan_baseline('SQL_0ae456f7f2328903')

PL/SQL procedure successfully completed.

SQL> EXEC :cnt :=
dbms_spm.drop_sql_plan_baseline('SQL_82f54579a9471356')

PL/SQL procedure successfully completed.

SQL> EXEC dbms_spm.configure(-
                        'AUTO_CAPTURE_SQL_TEXT', '%sh.sales%', TRUE)
>
PL/SQL procedure successfully completed.

SQL> SELECT parameter_name, last_modified
        FROM    dba_sql_management_config;
2
PARAMETER_NAME                                LAST_MODIFIED
-----
SPACE_BUDGET_PERCENT
PLAN_RETENTION_WEEKS
AUTO_CAPTURE_PARSING_SCHEMA_NAME
AUTO_CAPTURE_MODULE
AUTO_CAPTURE_ACTION
AUTO_CAPTURE_SQL_TEXT      17-APR-16 03.20.35.000000 AM

6 rows selected.

SQL>

```

5. Reexecute the `$HOME/labs/TUNING/query1.sql` script. The script executes a repetitive query on `oe` tables and then a repetitive query on `sh` tables

```

SQL> @$HOME/labs/TUNING/query1.sql
...
SQL>

```

*Q/ Did SPM capture only query plans on `sh.sales` table?*

```

SQL> SELECT sql_text, sql_handle, origin
        FROM    dba_sql_plan_baselines
        WHERE    sql_text like 'SELECT /*LOAD_AUTO*/%';
2      3
SQL_TEXT

```



```

-----
SQL_HANDLE          ORIGIN
-----
SELECT /*LOAD_AUTO*/ count(*)
      FROM    sh.sales
      WHERE   time_id <= TO_DAT
SQL_0ae456f7f2328903 AUTO-CAPTURE

SQL>

```

***A/ Yes. Query plans on oe tables were not captured.***

6. Remove existing filters, if any, on SQL text and check that all plans would have been captured.

```

SQL> EXEC dbms_spm.configure(-
              'AUTO_CAPTURE_SQL_TEXT', NULL, NULL)
>
PL/SQL procedure successfully completed.

SQL> EXIT
$

```

## Practice 17-2: Improving Statistics Gathering Quality

### Overview

The advisor task runs automatically in the maintenance window, but you can also run it on demand. If the advisor makes findings and then recommendations, then in some cases you can run system-generated scripts to implement them. Optimizer statistics play a significant part in determining the execution plan for queries. Therefore, it is critical for the optimizer to gather and maintain accurate and up-to-date statistics. All findings are derived from rules, but not all rules generate findings. You can influence the rules used or not used in the AST (Automatic SQL Tuning).

### Tasks

1. Your goal is to disable recommendations for all objects except those in the OE schema.
  - a. Create an object filter for an Optimizer Advisor Task using the code:

```
CREATE OR REPLACE PROCEDURE sh_obj_filter(p_tname IN VARCHAR2)
IS v_retc CLOB;
BEGIN
    v_retc :=
DBMS_STATS.CONFIGURE_ADVISOR_OBJ_FILTER(p_tname, 'EXECUTE', NULL, N
ULL, NULL, 'DISABLE');
    v_retc :=
DBMS_STATS.CONFIGURE_ADVISOR_OBJ_FILTER(p_tname, 'EXECUTE', NULL, '
OE', NULL, 'ENABLE');
END;
/
```

```
$ sqlplus sys@pdb_orcl AS SYSDBA
Enter password: *****
Connected.
SQL> CREATE OR REPLACE PROCEDURE sh_obj_filter(
        p_tname IN VARCHAR2) IS
        v_retc CLOB;
        BEGIN
            v_retc :=
DBMS_STATS.CONFIGURE_ADVISOR_OBJ_FILTER(p_tname, 'EXECUTE', NULL, N
ULL, NULL, 'DISABLE');
            v_retc :=
DBMS_STATS.CONFIGURE_ADVISOR_OBJ_FILTER(p_tname, 'EXECUTE', NULL, '
OE', NULL, 'ENABLE');
        END;
        /
        2      3      4      5      6      7      8
Procedure created.

SQL>
```

- b. Create and execute the advisor task.

```
SQL> DECLARE
  v_tname    VARCHAR2(128) := 'my_task';
  v_ename    VARCHAR2(128) := NULL;
  v_report   CLOB := null;
  v_script   CLOB := null;
  v_implementation_result CLOB;
BEGIN
  v_tname := DBMS_STATS.CREATE_ADVISOR_TASK(v_tname);
  sh_obj_filter(v_tname);
  v_ename := DBMS_STATS.EXECUTE_ADVISOR_TASK(v_tname);
END;
/
2      3      4      5      6      7      8      9      10     11     12
PL/SQL procedure successfully completed.

SQL>
```

- c. Verify that the procedure completed.

```
SQL> SELECT advisor_name, execution_type, last_execution,
           status
        FROM user_advisor_tasks
       WHERE task_name = 'MY_TASK';
2      3      4
ADVISOR_NAME          EXECUTION_TYPE LAST_EXECUTION STATUS
-----
Statistics Advisor STATISTICS          EXEC_11      COMPLETED

SQL> SELECT task_name, execution_name, execution_end,
           execution_type AS type, status
        FROM user_advisor_executions;
2      3

2      3
TASK_NAME          EXECUTION_NAME EXECUTION_END      TYPE
-----
-----
STATUS
-----
MY_TASK          EXEC_11          13-OCT-16
STATISTICS
COMPLETED
```

```

AUTO_STATS_ADVISOR_TASK EXEC_1          03-OCT-16
STATISTICS
COMPLETED

SQL>

```

- d. Generate the report.

```

SQL> VAR b_report CLOB
SQL> DECLARE
    v_tname  VARCHAR2(32767);
BEGIN

```

```

    v_tname   := 'my_task';
    :b_report := dbms_stats.report_advisor_task(v_tname, type =>
'TEXT', section=>'ALL', level=>'ALL');
END;
/

```

```

2      3      4      5      6      7

```

PL/SQL procedure successfully completed.

SQL>

```

SQL> DECLARE

```

```

    v_len    NUMBER(10);
    v_offset NUMBER(10) :=1;
    v_amount  NUMBER(10) :=10000;

```

```

BEGIN

```

```

    v_len := DBMS_LOB.getlength(:b_report);
    WHILE (v_offset < v_len)
    LOOP

```

```

DBMS_OUTPUT.PUT_LINE(DBMS_LOB.SUBSTR(:b_report,v_amount,v_offset
));

```

```

    v_offset := v_offset + v_amount;

```

```

    END LOOP;

```

```

END;

```

```

/

```

```

2      3      4      5      6      7      8      9     10     11     12     13

```

PL/SQL procedure successfully completed.

SQL>

- e. Display the findings and recommendations for each of them.

```

SQL> SELECT f.finding_id, f.message, r.benefit_type
       FROM   user_advisor_findings f,
              user_advisor_recommendations r

```

```

WHERE f.finding_id = r.finding_id
AND   f.task_name = 'MY_TASK'
AND   f.execution_name = 'EXEC_11';

```

2      3      4      5      6

FINDING\_ID

-----

MESSAGE

-----

BENEFIT\_TYPE

-----

1

The CONCURRENT preference is not used.  
Set the CONCURRENT preference.

2

There are 5 object(s) with no statistics.  
Gather Statistics on those objects with no statistics.

1

The CONCURRENT preference is not used.  
Set parameter job\_queue\_processes to 1 or higher.

2

There are 5 object(s) with no statistics.  
Set parameter \_enable\_automatic\_maintenance to 1.

SQL>

- f. Generate the script before a possible implementation.

```

SQL> SET SERVEROUTPUT ON
SQL> VARIABLE b_script CLOB
SQL> DECLARE
    v_tname VARCHAR2(32767);
BEGIN
    v_tname := 'my_task';
    :b_script := DBMS_STATS.SCRIPT_ADVISOR_TASK(v_tname);
END;
/

```

2      3      4      5      6      7

PL/SQL procedure successfully completed.

SQL>

```

SQL> DECLARE
  v_len      NUMBER(10);
  v_offset NUMBER(10) :=1;
  v_amount NUMBER(10) :=10000;
BEGIN
  v_len := DBMS_LOB.getlength(:b_report);
  WHILE (v_offset < v_len)
  LOOP
    DBMS_OUTPUT.PUT_LINE(DBMS_LOB.SUBSTR(:b_script,v_amount,v_offset
));
    v_offset := v_offset + v_amount;
  END LOOP;
END;
/
  2      3      4      5      6      7      8      9     10     11     12     13
PL/SQL procedure successfully completed.
-- Script generated for the recommendations from execution
EXEC_11
-- in the statistics advisor task MY_TASK
-- Script version 12.2
-
-
-- No scripts will be provided for the rule USEAUTOJOB.Please
check the report for more details.
-- No scripts will be provided for the rule COMPLETEAUTOJOB.
Please check the report for more details.
-- No scripts will be provided for the rule
MAINTAINSTATSHISTORY. Please check the report for more details.
-- No scripts will be provided for the rule
TURNONSQLPLANDIRECTIVE. Please check the report for more
details.
-- No scripts will be provided for the rule AVOIDSETPROCEDURES.
Please check the report for more details.
-- No scripts will be provided for the rule USEDEFAULTPARAMS.
Please check the report for more details.
-- No scripts will be provided for the rule
USEGATHERSCHEMASTATS. Please check the report for more details.
-- No scripts will be provided for the rule
AVOIDINEFFICIENTSTATSOPRSEQ. Please check the report for
more details.
-- No scripts will be provided for the rule
AVOIDUNNECESSARYSTATSCOLLECTION. Please check the report for
more details.
-- No scripts will be provided for the rule

```

```

GATHERSTATSAFTERBULKDML. Please check the report for more
details.
--
No scripts will be provided for the rule AVOIDDROPRECREATE.
Please check the report for more details.
-- No scripts will be provided for the rule AVOIDOUTOFRANGE.
Please check the report for more details.
-- No scripts will be provided for the rule AVOIDANALYZETABLE.
Please check the report for more details.
-- No scripts will be provided for the rule USEAUTOJOB. Please
check the report for more details.
-- No scripts will be provided for the rule COMPLETEAUTOJOB.
Please check the report for more details.
-- No scripts will be provided for the rule
MAINTAINSTATSHISTORY. Please check the report for more details.
-- No scripts will be provided for the rule
TURNONSQLPLANDIRECTIVE. Please check the report for more
details.
-- No scripts will be provided for the rule AVOIDSETPROCEDURES.
Please check the report for more details.
-- No scripts will be provided for the rule USEDEFAULTPARAMS.
Please check the report for more details.
-- No scripts will be provided for the rule
USEGATHERSCHEMASTATS. Please check the report for more details.
-- No scripts will be provided for the rule
AVOIDINEFFICIENTSTATSOPRSEQ. Please check the report for more
details.
-- No scripts will be provided for the rule
AVOIDUNNECESSARYSTATSCOLLECTION. Please check the report for more
details.
-- No scripts will be provided for the rule
GATHERSTATSAFTERBULKDML. Please check the report for more
details.
--
No scripts will be provided for the rule AVOIDDROPRECREATE.
Please check the report for more details.
-- No scripts will be provided for the rule AVOIDOUTOFRANGE.
Please check the report for more details.
-- No scripts will be provided for the rule AVOIDANALYZETABLE.
Please check the report for more details.
--

Scripts for rule USECONCURRENT
-- Rule Description: Use Concurrent preference for Statistics
Collection

```

```

-- Set the concurrent preference begin
dbms_stats.set_global_prefs('CONCURRENT','AUTOMATIC'); END;;

-- No scripts will be provided for the rule USEAUTOJOB. Please
check the report for more details.
-- No scripts will be provided for the rule COMPLETEAUTOJOB.
Please check the report for more details.
-- No scripts will be provided for the rule
MAINTAINSTATSHISTORY. Please check the report for more details.
-- No scripts will be provided for the rule
TURNONSQLPLANDIRECTIVE. Please check the report for more
details.
-- No scripts will be provided for the rule AVOIDSETPROCEDURES.
Please check the report for more details.
-- No scripts will be provided for the rule USEDEFAULTPARAMS.
Please check the report for more details.
-- No scripts will be provided for the rule
USEGATHERSCHEMASTATS. Please check the report for more details.
-- No scripts will be provided for the rule
AVOIDINEFFICIENTSTATSOPRSEQ. Please check the report for more
details.
-- No scripts will be provided for the rule
AVOIDUNNECESSARYSTATSCOLLECTION. Please check the report for
more details.
-- No scripts will be provided for the rule
GATHERSTATSAFTERBULKDML. Please check the report for more
details.
-- No scripts will be provided for the rule AVOIDDROPRECREATE.
Please check the report for more details.
-- No scripts will be provided for the rule AVOIDOUTOFRANGE.
Please check the report for more details.
-- No scripts will be provided for the rule AVOIDANALYZETABLE.
Please check the report for more details.

-- Scripts for rule USEDEFAULTPREFERENCE
-- Rule Description: Use Default Preference for Stats Collection
-- Set global preferences to default values.

-- Scripts for rule USEDEFAULTOBJECTPREFERENCE
-- Rule Description: Use Default Object Preference for
statistics Collection
-- Setting object-level preferences to default values
-- setting CASCADE to default value for object level preference
-- setting ESTIMATE_PERCENT to default value for object level
preference
-- setting METHOD_OPT to default value for object level

```



```
preference
-- setting GRANULARITY to default value for object level
preference
-- setting NO_INVALIDATE to default value for object
level preference

-- Scripts for rule USEINCREMENTAL
-- Rule Description: Statistics should be maintained
incrementally when it is beneficial
-- Turn on the incremental option for those objects for which
using incremental is helpful.

-- Scripts for rule UNLOCKNONVOLATILETABLE
-- Rule Description: Statistics for objects with non-volatile
should not be locked
-- Unlock statistics for objects that are not volatile.

-- Scripts for rule LOCKVOLATILETABLE
-- Rule Description: Statistics for objects with volatile data
should be locked
-- Lock statistics for volatile objects.

-- Scripts for rule NOTUSEINCREMENTAL
-- Rule Description: Statistics should not be maintained
incrementally when it is not beneficial
-- Turn off incremental option for those objects for which using
incremental is not helpful.

-- Scripts for rule USEAUTODEGREE
-- Rule Description: Use Auto Degree for statistics collection
-- Turn on auto degree for those objects for which using auto
degree is helpful.

-- Scripts for rule AVOIDSTALESTATS
-- Rule Description: Avoid objects with stale or no statistics
-- Gather statistics for those objects that are missing or have
no statistics.
-- Scripts for rule MAINTAINSTATSCONSISTENCY
-- Rule Description: Statistics of dependent objects should be
```

```

consistent
-- Gather statistics for those objects that are missing or have
no statistics.
declare
    obj_filter_list
dbms_stats.ObjectTab;
    obj_filter          dbms_stats.ObjectElem;

obj_cnt          number := 0;
begin
    obj_filter_list := dbms_stats.ObjectTab();
    obj_filter.ownname := 'OE';
    obj_filter.objtype := 'TABLE';
    obj_filter.objname := 'CUSTOMER';
    obj_filter_list.extend();
    obj_cnt := obj_cnt + 1;

    obj_filter_list(obj_cnt) := obj_filter;
    obj_filter.ownname := 'OE';
    obj_filter.objtype := 'TABLE';
    obj_filter.objname := 'DATE_DIM';
    obj_filter_list.extend();
    obj_cnt := obj_cnt + 1;

    obj_filter_list(obj_cnt) := obj_filter;
    obj_filter.ownname := 'OE';
    obj_filter.objtype := 'TABLE';
    obj_filter.objname := 'LINEORDER';
    obj_filter_list.extend();
    obj_cnt := obj_cnt + 1;

    obj_filter_list(obj_cnt) := obj_filter;
    obj_filter.ownname := 'OE';
    obj_filter.objtype := 'TABLE';
    obj_filter.objname := 'PART';
    obj_filter_list.extend();
    obj_cnt := obj_cnt + 1;

    obj_filter_list(obj_cnt) := obj_filter;
    obj_filter.ownname := 'OE';
    obj_filter.objtype := 'TABLE';
    obj_filter.objname := 'SUPPLIER';
    obj_filter_list.extend();

```

Copyright © 2016, Oracle and/or its affiliates. All rights reserved.

```

obj_cnt := obj_cnt + 1;

obj_filter_list(obj_cnt) := obj_filter;

dbms_stats.gather_database_stats(

obj_filter_list=>obj_filter_list);
end;
/

PL/SQL procedure successfully completed.

SQL>

```

*Q/ What would you do if you agree with the recommendations?*

***A/ You could either execute the generated SQL script or use the DBMS\_STATS.IMPLEMENT\_ADVISOR\_TASK.***

```

SQL> VARIABLE b_ret CLOB
SQL> DECLARE
    v_tname VARCHAR2(32767);
BEGIN
    v_tname := 'my_task';
    :b_ret := DBMS_STATS.IMPLEMENT_ADVISOR_TASK(v_tname);
END;
/

2      3      4      5      6      7
PL/SQL procedure successfully completed.

SQL>

```

- g. Drop the task.

```

SQL> exec DBMS_STATS.DROP_ADVISOR_TASK('my_task')

PL/SQL procedure successfully completed.

SQL>

```

3. Your goal is now to create a rule filter for an Optimizer Advisor Task. You configure a filter using `CONFIGURE_ADVISOR_RULE_FILTER`, specifying that task execution should exclude all rules but `AvoidStaleStats`, honoring only this rule which was one of the recommendations of the script generated above.

- a. Find all the existing rules.

```

SQL> SELECT name, rule_type, description
       FROM v$sqlstats_advisor_rules
       ORDER BY NAME;

```

2	3
NAME	
-----	
RULE_TYPE	
-----	
DESCRIPTION	
-----	
AvoidUnnecessaryStatsCollection	
OBJECT	
Avoid unnecessary statistics collection	
UseDefaultObjectPreference	
OBJECT	
Use Default Object Preference for statistics collection	
UseAutoDegree	
OBJECT	
Use Auto Degree for statistics collection	
AvoidOutOfRange	
OBJECT	
Avoid Out of Range Histogram endpoints	
NotUseIncremental	
OBJECT	
Statistics should not be maintained incrementally when it is not	
UseIncremental	
OBJECT	
Statistics should be maintained incrementally when it is benefic	
AvoidDropRecreate	
OBJECT	
Avoid drop and recreate object sequences	
MaintainStatsConsistency	
OBJECT	
Statistics of dependent objects should be consistent	
UnlockNonVolatileTable	
OBJECT	
Statistics for objects with non-volatile should not be locked	

LockVolatileTable

OBJECT

Statistics for objects with volatile data should be locked

GatherStatsAfterBulkDML

OBJECT

Do not gather statistics right before bulk DML

AvoidAnalyzeTable

OBJECT

Avoid using analyze table commands for statistics collection

### **AvoidStaleStats**

OBJECT

Avoid objects with stale or no statistics

UseGatherSchemaStats

OPERATION

Use gather\_schema\_stats procedure

AvoidInefficientStatsOprSeq

OPERATION

Avoid inefficient statistics operation sequences

AvoidSetProcedures

OPERATION

Avoid Set Statistics Procedures

UseDefaultParams

OPERATION

Use Default Parameters in Statistics Collection Procedures

CompleteAutoJob

SYSTEM

Auto Statistics Gather Job should complete successfully

UseAutoJob

SYSTEM

Use Auto Job for Statistics Collection

MaintainStatsHistory

```

SYSTEM
Maintain Statistics History

UseConcurrent
SYSTEM
Use Concurrent preference for Statistics Collection


SYSTEM


TurnOnSQLPlanDirective
SYSTEM
SQL Plan Directives should not be disabled


UseDefaultPreference
SYSTEM
Use Default Preference for Stats Collection


24 rows selected.

SQL>

```

- b. Configure a filter that enables only the `AvoidStaleStats` rule.

```

SQL> CREATE OR REPLACE PROCEDURE rule_filter(
      p_tname IN VARCHAR2) IS
  v_retc CLOB;
BEGIN
  v_retc := DBMS_STATS.CONFIGURE_ADVISOR_RULE_FILTER(
    task_name          => 'opt_adv_task1',
    stats_adv_opr_type => 'EXECUTE',
    rule_name          => 'AvoidStaleStats',
    action              => 'ENABLE');
END;
/
  2      3      4      5      6      7      8      9     10     11
Procedure created.

SQL>

```

- c. Create and execute the advisor task.

```
SQL> DECLARE
  v_tname  VARCHAR2(128) := 'opt_adv_task1';
  v_ename  VARCHAR2(128) := NULL;
  v_report CLOB := null;
  v_script CLOB := null;
  v_implementation_result CLOB;
BEGIN
  v_tname := DBMS_STATS.CREATE_ADVISOR_TASK(v_tname);
  rule_filter(v_tname);
  v_ename := DBMS_STATS.EXECUTE_ADVISOR_TASK(v_tname);
  v_report := DBMS_STATS.REPORT_ADVISOR_TASK(v_tname);
END;
/
  2      3      4      5      6      7      8      9     10     11     12     13     14
15
PL/SQL procedure successfully completed.

SQL>
```

- d. Verify that the procedure completed.

```
SQL> SELECT advisor_name, execution_type, last_execution,
           status
       FROM user_advisor_tasks
       WHERE task_name = 'OPT_ADV_TASK1';
  2      3      4
ADVISOR_NAME          EXECUTION_ LAST_EXECUTION STATUS
-----
Statistics Advisor STATISTICS EXEC_21          COMPLETED

SQL> SELECT task_name, execution_name, execution_end,
           execution_type AS type, status
       FROM user_advisor_executions;
  2      3
TASK_NAME          EXECUTION_NAME EXECUTION_END          TYPE
-----
-----
STATUS
-----
OPT_ADV_TASK1          EXEC_21          13-OCT-16
STATISTICS
COMPLETED
```

```

AUTO_STATS_ADVISOR_TASK EXEC_1          03-OCT-16
STATISTICS
COMPLETED

SQL>

```

e. Generate the report.

```

SQL> VAR b_report CLOB
SQL> DECLARE
    v_tname  VARCHAR2(32767);
BEGIN
    v_tname  := 'opt_adv_task1';
    :b_report := dbms_stats.report_advisor_task(v_tname, type =>
'TEXT', section=>'ALL', level=>'ALL');
END;
/
 2      3      4      5      6      7

PL/SQL procedure successfully completed.

SQL>
SQL> DECLARE
    v_len    NUMBER(10);
    v_offset NUMBER(10) :=1;
    v_amount  NUMBER(10) :=10000;
BEGIN
    v_len := DBMS_LOB.getlength(:b_report);
    WHILE (v_offset < v_len)
    LOOP
        DBMS_OUTPUT.PUT_LINE(DBMS_LOB.SUBSTR(:b_report,v_amount,v_offset
));
        v_offset := v_offset + v_amount;
    END LOOP;
END;
/
 2      3      4      5      6      7      8      9     10     11     12     13
...
Recommendation:  Regather statistics on objects with stale
statistics.
Example:
-- Gathering statistics for tables with stale or no statistics
in schema, SH:
exec dbms_stats.gather_schema_stats('SH', options =>
'GATHER AUTO')

```



Rationale: Stale statistics or no statistics will result in bad plans.

-----  
Rule Name: UseIncremental

Rule Description: Statistics should be maintained incrementally when it is beneficial

**Finding: Incremental option should be used on 1 object(s) for Statistics gathering.**

**Schema: SH**

**Objects: SALES**

**Recommendation: Use the incremental option for statistics gathering on these objects.**

Example:

-- Turn on the incremental option for 'SH.SALES':

```
dbms_stats.set_table_prefs('SH', 'SALES', 'INCREMENTAL',  
'TRUE');
```

Rationale: Using the incremental option reduces the time it takes to gather statistics on partitioned tables. However, it does store additional information, which takes up additional space.

-----  
PL/SQL procedure successfully completed.

SQL>

- f. Display the findings and recommendations for each of them.

```
SQL> SELECT f.finding_id, f.message, r.benefit_type  
       FROM   user_advisor_findings f,  
              user_advisor_recommendations r  
       WHERE  f.finding_id = r.finding_id  
       AND    f.task_name = 'OPT_ADV_TASK1'  
       AND    f.execution_name = 'EXEC_21';
```

2      3      4      5      6

1

**There are 10 object(s) with stale statistics.**

Regather statistics on objects with stale statistics.

2

Incremental option should be used on 1 object(s) for statistics gathering.

Use the incremental option for statistics gathering on these objects.

1

There are 10 object(s) with stale statistics.

Set parameter job\_queue\_processes to 1 or higher.

2

Incremental option should be used on 1 object(s) for statistics gathering.

Set parameter \_enable\_automatic\_maintenance to 1.

SQL>

- g. Generate the script before a possible implementation.

```
SQL> SET SERVEROUTPUT ON
```

```
SQL> VARIABLE b_script CLOB
```

```
SQL> DECLARE
```

```
    v_tname VARCHAR2(32767);
```

```
BEGIN
```

```
    v_tname := 'opt_adv_task1';
```

```
    :b_script := DBMS_STATS.SCRIPT_ADVISOR_TASK(v_tname);
```

```
END;
```

```
/
```

```
2      3      4      5      6      7
```

PL/SQL procedure successfully completed.

```
SQL>
```

```
SQL> DECLARE
```

```
    v_len      NUMBER(10);
```

```
    v_offset   NUMBER(10) :=1;
```

```
    v_amount   NUMBER(10) :=10000;
```

```
BEGIN
```

```
    v_len := DBMS_LOB.getlength(:b_report);
```

```
    WHILE (v_offset < v_len)
```

```
    LOOP
```

```
    DBMS_OUTPUT.PUT_LINE(DBMS_LOB.SUBSTR(:b_script,v_amount,v_offset));
```

```

        v_offset := v_offset + v_amount;
    END LOOP;
END;
/
2      3      4      5      6      7      8      9     10     11     12     13
PL/SQL procedure successfully completed.
-- Script generated for the recommendations from execution
EXEC_21
-- in the statistics advisor task OPT_ADV_TASK1
-- Script version 12.2
-
-
...
-- Scripts for rule USEINCREMENTAL
-- Rule Description: Statistics should be maintained
incrementally when it is beneficial
-- Turn on the incremental option for those objects
for which using incremental is helpful.
begin
dbms_stats.set_table_prefs('SH', 'SALES', 'INCREMENTAL',
'TRUE');
end;
/
...
-- Scripts for rule AVOIDSTALESTATS
-- Rule Description: Avoid objects with stale or no statistics
-- Gather statistics for those objects that are missing or have
no statistics.
-- Scripts for rule MAINTAINSTATSCONSISTENCY
-- Rule Description: Statistics of dependent objects should be
consistent
-- Gather statistics for those objects that are missing or have
no statistics.
declare
    obj_filter_list
dbms_stats.ObjectTab;
    obj_filter      dbms_stats.ObjectElem;

obj_cnt            number := 0;
begin
    obj_filter_list := dbms_stats.ObjectTab();
    obj_filter.ownname := 'AUDSYS';
    obj_filter.objtype := 'TABLE';

```

```

obj_filter.objname := 'AUD$UNIFIED';
obj_filter_list.extend();
obj_cnt := obj_cnt + 1;

obj_filter_list(obj_cnt) := obj_filter;
obj_filter.ownname := 'MDSYS';
obj_filter.objtype := 'TABLE';
obj_filter.objname := 'SDO_COORD_OP_PARAM_VALS';
obj_filter_list.extend();
obj_cnt := obj_cnt + 1;

obj_filter_list(obj_cnt) := obj_filter;
obj_filter.ownname := 'MDSYS';
obj_filter.objtype := 'TABLE';
obj_filter.objname := 'SDO_GEOR_XMLSCHEMA_TABLE';
obj_filter_list.extend();
obj_cnt := obj_cnt + 1;

obj_filter_list(obj_cnt) := obj_filter;
obj_filter.ownname := 'MDSYS';
obj_filter.objtype := 'TABLE';
obj_filter.objname := 'SDO_STYLES_TABLE';
obj_filter_list.extend();
obj_cnt := obj_cnt + 1;

obj_filter_list(obj_cnt) := obj_filter;
obj_filter.ownname := 'MDSYS';
obj_filter.objtype := 'TABLE';
obj_filter.objname := 'SDO_XML_SCHEMAS';
obj_filter_list.extend();
obj_cnt := obj_cnt + 1;

obj_filter_list(obj_cnt) := obj_filter;
obj_filter.ownname := 'MDSYS';
obj_filter.objtype := 'TABLE';
obj_filter.objname := 'SDO_XSD_TABLE';
obj_filter_list.extend();
obj_cnt := obj_cnt + 1;

obj_filter_list(obj_cnt) := obj_filter;
obj_filter.ownname := 'SYSTEM';
obj_filter.objtype := 'TABLE';

```

```

obj_filter.objname := 'AQ$_QUEUES';
obj_filter_list.extend();
obj_cnt := obj_cnt + 1;

obj_filter_list(obj_cnt) := obj_filter;
obj_filter.ownname := 'WMSYS';
obj_filter.objtype := 'TABLE';
obj_filter.objname := 'WM$ENV_VARS$';
obj_filter_list.extend();
obj_cnt := obj_cnt + 1;

obj_filter_list(obj_cnt) := obj_filter;
obj_filter.ownname := 'XDB';
obj_filter.objtype := 'TABLE';
obj_filter.objname := 'XDB$RESOURCE';
obj_filter_list.extend();
obj_cnt := obj_cnt + 1;

obj_filter_list(obj_cnt) := obj_filter;
obj_filter.ownname := 'XDB';
obj_filter.objtype := 'TABLE';
obj_filter.objname := 'XDB$XDB_READY';
obj_filter_list.extend();
obj_cnt := obj_cnt + 1;

obj_filter_list(obj_cnt) := obj_filter;

dbms_stats.gather_database_stats(
obj_filter_list=>obj_filter_list);
end;
/

PL/SQL procedure successfully completed.

SQL>

```

*Q/ What would you do if you agree with the recommendations?*

***A/ You could either execute the generated SQL script or use the DBMS\_STATS.IMPLEMENT\_ADVISOR\_TASK. Observe that the 10 objects with no statistics are listed in the report, and the incremental option recommended for those objects for which using incremental would be helpful is suggested for SH.SALES table.***

```
SQL> VARIABLE b_ret CLOB
```

```

SQL> DECLARE
    v_tname VARCHAR2(32767);
BEGIN
    v_tname := 'opt_adv_task1';
    :b_ret := DBMS_STATS.IMPLEMENT_ADVISOR_TASK(v_tname);
END;
/
  2      3      4      5      6      7
PL/SQL procedure successfully completed.

SQL>

```

h. Drop the task.

```

SQL> exec DBMS_STATS.DROP_ADVISOR_TASK('opt_adv_task1')

PL/SQL procedure successfully completed.

SQL> EXIT
$

```

## Practice 17-3: Disabling Database Triggers During SPA Task Execution

### Overview

In this practice, you will use the new SPA (SQL Performance Analysis) task parameter that disables database triggers during SPA tasks execution. As SPA tasks complete performance tests on queries, they must not update, insert, or delete data through triggers for example. This is not the default behavior before Oracle Database 12.2.

### Tasks

1. Before starting the practice, execute the `$HOME/labs/TUNING/setup_trig.sh` shell script. The script creates a database trigger on `sh.sales` partitioned table in `pdb_orcl`.

```
create or replace trigger trig_insert_sales
before update on app_sh.sales
begin
insert into app_sh.sales select * from app_sh.sales;
end;
/
```

```
$ $HOME/labs/TUNING/setup_trig.sh
```

```
...
$
```

2. Due to performance query issues reported by users, you have to execute a performance analysis task on `sh.sales` table to analyze the queries while updates are executed on `sh.sales` table and captured in an STS. Any update statement triggers an insert in the table.
  - a. Create an STS (SQL Tuning Set) using the `$HOME/labs/TUNING/script_sts.sql` SQL script. The STS will be populated when the following update statement will be executed from another session.

```
UPDATE sh.sales SET amount_sold = amount_sold+1 WHERE
amount_sold = 7;
```

```
$ . oraenv
ORACLE_SID = [ORCL] ? ORCL
The Oracle base remains unchanged with value /u01/app/oracle
$ sqlplus sh@pdb_orcl
Enter password: *****
Connected.
SQL> SELECT count(*) FROM sh.sales WHERE amount_sold = 7;

COUNT (*)
-----
          55

SQL> @$HOME/labs/TUNING/script_sts.sql
...
```

```
SQL>
```

- b. Now that the STS contains the update statement execution, create and execute the SPA task analysis.

```
SQL> EXEC :tname := dbms_sqlpa.create_analysis_task(-
                        sqlset_name => :sname, -
                        task_name => :tname);

> >
PL/SQL procedure successfully completed.

SQL> EXEC dbms_sqlpa.execute_analysis_task(-
                        task_name      => :tname, -
                        execution_type => 'test execute', -
                        execution_name => :ename)

> > >
PL/SQL procedure successfully completed.

SQL>
```

*Q/ What happened? Did the trigger execute?*

```
SQL> SELECT rows_processed FROM dba_advisor_sqlstats
       WHERE task_name = :tname AND execution_name = :ename;

2
ROWS_PROCESSED
-----
                55

SQL>
```

***A/ No. The trigger did not insert 55 more rows into sh.sales table. The old behavior in previous Oracle Database versions of SPA would have executed the trigger.***

3. You want the trigger to be executed during the test analysis task to see the performance impact of the insert statement. You can configure the SPA task so that this happens.
- a. Set the appropriate parameter for the SPA task before executing the task.

```
SQL> EXEC dbms_sqlpa.set_analysis_task_parameter (:tname, -
                        'EXECUTE_TRIGGERS', 'TRUE')

>
PL/SQL procedure successfully completed.

SQL>
```

- b. Reexecute a new SPA task analysis with the new setting.

```
SQL> EXEC :ename := 'my_exe2';
SQL> EXEC dbms_sqlpa.execute_analysis_task(-
                        task_name      => :tname, -
```



```

                                execution_type => 'test execute', -
                                execution_name => :ename)

> > >
PL/SQL procedure successfully completed.

SQL>

```

*Q/ What happened? Was the trigger executed?*

```

SQL> SELECT rows_processed FROM dba_advisor_sqlstats
      WHERE task_name = :tname AND execution_name = :ename;

2
ROWS_PROCESSED
-----
                110

SQL>

```

**A/ Yes. The trigger inserted 55 more rows into the *sh.sales* table.**

4. Execute the `$HOME/labs/TUNING/drop_SPA.sql` SQL script to drop the SPA task, the STS, and the trigger.

```

SQL> @$HOME/labs/TUNING/drop_SPA.sql
...
SQL> EXIT
$

```



# **Practices for Lesson 18: Resource Manager and Other Enhancements**

## **Chapter 18**

## Practices for Lesson 18: Overview

---

### Practices Overview

In the practices for this lesson, you will benefit from Resource Manager new directives to avoid excessive PGA consumption, benefit from cursor deferred invalidations in DDL statements, use OQC (On Query Computation) for direct MV access, and benefit from on-statement materialized views on MJVs.

## Practice 18-1: Avoiding Excessive PGA Memory Usage

### Overview

In this practice, you will use Resource Manager to avoid excessive PGA memory consumption in `pdb_orcl`.

### Tasks

1. Before starting the practice, execute the `$HOME/labs/admin/glogin_18.sh` shell script. The shell script sets formatting for all columns selected in queries.

```
$ $HOME/labs/admin/glogin_18.sh
$
```

2. Some users informed you that their query performance decreased in `pdb_orcl` when `oe` was generating some reporting queries on `lineorder` table. You don't know yet which limit to set on this user.
  - a. In a terminal window (*window1*), start the reporting query on `oe.lineorder` table.

```
$ . oraenv
ORACLE_SID = [ORCL] ? ORCL
The Oracle base has been set to /u01/app/oracle
$ sqlplus oe@pdb_orcl
Enter password: *****
Connected.
SQL> SELECT * FROM oe.lineorder
        ORDER BY LO_ORDERKEY,LO_CUSTKEY,LO_PARTKEY,LO_SUPPKEY;
...
SQL>
```

- b. Meanwhile, in another terminal window (*window2*), check the PGA memory used by `oe`.

```
$ sqlplus system@pdb_orcl
Enter password: *****
Connected.
SQL> SELECT username, name, value
        FROM   v$sesstat m, v$statname a, v$session s
        WHERE  m.statistic# = a.statistic# AND m.sid = s.sid
        AND    a.name LIKE 'session pga memory%'
        AND    username = 'OE';

USERNAME      NAME                                VALUE
-----
OE            session pga memory                  132408552
OE            session pga memory max              132867304

SQL>
```

*Q/ What is the PGA memory maximum used by oe?*

*A/ The maximum PGA used is about 126Mb.*

- c. Use the Resource Manager new directive to limit the PGA memory usage for `oe` in `pdb_orcl` to 100Mb.

- 1) Still in *window2*, use the `$HOME/labs/TUNING/script_rm.sql` SQL script that creates the `pga_plan` resource manager plan, the `reporting_users` consumer group and associates `oe` to the `reporting_users` consumer group.

```
SQL> @$HOME/labs/TUNING/script_rm.sql
...
PLAN          GROUP_OR_SUBPLAN          SESSION_PGA_LIMIT
-----
PGA_PLAN      OTHER_GROUPS                200

SQL>
```

- 2) Now set the PGA limit for the `reporting_users` and activate the plan in `pdb_orcl`.

```
SQL> EXEC dbms_resource_manager.clear_pending_area()
>
PL/SQL procedure successfully completed.

SQL> EXEC dbms_resource_manager.create_pending_area()
>
PL/SQL procedure successfully completed.

SQL> EXEC dbms_resource_manager.create_plan_directive(-
          'PGA_plan', 'Reporting_Users', session_pga_limit => 80)
>
PL/SQL procedure successfully completed.

SQL> EXEC dbms_resource_manager.validate_pending_area()
>
PL/SQL procedure successfully completed.

SQL> EXEC dbms_resource_manager.submit_pending_area()
>
PL/SQL procedure successfully completed.

SQL> ALTER SYSTEM SET resource_manager_plan='PGA_plan';

System altered.

SQL> ALTER SYSTEM FLUSH buffer_cache;
```

System altered.

```
SQL> SELECT plan, group_or_subplan, session_pga_limit
      FROM dba_rsrc_plan_directives
      WHERE plan = 'PGA_PLAN';
```

2	3	
PLAN	GROUP_OR_SUBPLAN	SESSION_PGA_LIMIT
-----	-----	-----
PGA_PLAN	OTHER_GROUPS	200
PGA_PLAN	REPORTING_USERS	80

```
SQL>
```

- d. Back in *window1* in *oe* session, restart the query on *oe.lineorder* table.

```
SQL> SELECT * FROM oe.lineorder
      ORDER BY LO_ORDERKEY,LO_CUSTKEY,LO_PARTKEY,LO_SUPPKEY;
```

```
2 SELECT * FROM oe.lineorder
```

```
*
```

```
ERROR at line 1:
```

```
ORA-10260: PGA limit (80 MB) exceeded - process terminated
```

```
SQL> EXIT
```

```
$
```

- e. Back in *window2* in *oe* session, check the PGA memory limit by *oe*.

```
SQL> SELECT username, name, value
      FROM v$sesstat m, v$statname a, v$session s
      WHERE m.statistic# = a.statistic# AND m.sid = s.sid
      AND a.name LIKE 'session pga memory%'
      AND username = 'OE';
```

2	3	4	5
USERNAME	NAME		VALUE
-----	-----	-----	-----
OE	session pga memory		83855496
OE	session pga memory max		104817896

```
SQL> EXIT
```

```
$
```

3. Clean up the Resource Manager plan.

```
$ $HOME/labs/TUNING/cleanup_rm.sh  
...  
$
```



## Practice 18-2: Using Cursor Invalidations with DDL Statements

### Overview

In this practice, you have to perform DDL statements on tables while avoiding cursor invalidation to reduce the impact of DDL cursor invalidations.

### Tasks

1. In a previous practice, you ran the `$HOME/labs/TUNING/setup_sh.sh` shell script that created the `sh` schema in `pdb_orcl`. If this is not the case, reexecute it. If errors appear during import, ignore them.

```
$ $HOME/labs/TUNING/setup_sh.sh
...
$
```

2. Your goal is to create an index on `sh.t1` table without invalidating the cursors opened on the table.
  - a. Use the `$HOME/labs/TUNING/script_t1.sql` SQL script to create the table.

```
$ sqlplus sh@pdb_orcl
Enter password: *****
Connected.
```

```
SQL> @$HOME/labs/TUNING/script_t1.sql
```

```
...
SQL>
```

*Q/ Is there any invalidation on the cursor of the SELECT statement on the table?*

```
SQL> SELECT object_status, ddl_no_invalidate,
           is_rolling_invalid, is_rolling_refresh_invalid,
           invalidations, sql_text
        FROM v$sql
        WHERE sql_text like 'SELECT%sh.t1%'
        AND    sql_text not like '%ddl%';
 2      3      4      5      6
OBJECT_STATUS          D I I INVALIDATIONS  SQL_TEXT
-----
VALID                  N N N                0 SELECT * FROM sh.t1

SQL>
```

*A/ There has been no DDL statement since the SELECT statement on the table and therefore no cursor invalidation.*

- b. Create a new index on the table.

```
SQL> CREATE INDEX sh.i ON sh.t1 (c1);

Index created.

SQL> SELECT object_status, ddl_no_invalidate,
           is_rolling_invalid, is_rolling_refresh_invalid,
           invalidations, sql_text
        FROM v$sql
        WHERE sql_text like 'SELECT%sh.t1%'
        AND    sql_text not like '%ddl%';
 2      3      4      5      6
OBJECT_STATUS          D I I INVALIDATIONS SQL_TEXT
-----
VALID                  N Y N                  0 SELECT * FROM sh.t1

SQL>
```

Q1/ What does `IS_ROLLING_INVALID` column value set to `Y` mean?

**A1/ `IS_ROLLING_INVALID` column value set to `Y` means that this cursor is rolling invalidated, but the cursor has not executed in this state yet. There are still no invalidations.**

Q2/ What did lead it to be set to `Y` whereas no `DEFERRED INVALIDATION` clause has been defined in the `CREATE INDEX DDL` statement?

```
SQL> SHOW PARAMETER cursor_invalidation

NAME                                TYPE        VALUE
-----
cursor_invalidation                string      DEFERRED

SQL>
```

**A2/ `CURSOR_INVALIDATION` parameter value is set to `DEFERRED`.**

- c. Reexecute the query using the cursor.

```
SQL> SELECT * FROM sh.t1;

      C1      C2
-----
      1      2

SQL> SELECT object_status, ddl_no_invalidate,
           is_rolling_invalid, is_rolling_refresh_invalid,
           invalidations, sql_text
        FROM v$sql
        WHERE sql_text like 'SELECT%sh.t1%'
```

```

      AND      sql_text not like '%ddl%';
2      3      4      5      6
OBJECT_STATUS      D I I INVALIDATIONS SQL_TEXT
-----
VALID              N X N              0 SELECT * FROM sh.t1

SQL>

```

**Q/** What does `IS_ROLLING_INVALID` column value set to `X` mean and what did lead it to be set to `X`?

**A/** `IS_ROLLING_INVALID` column value set to `X` means that this cursor is rolling invalidated, and the cursor has executed in this state. There are still no invalidations.

3. Your goal is to be able to truncate the `sh.t1` table without invalidating the cursors opened.
  - a. Truncate the table without invalidating the cursor opened on the table.

```

SQL> TRUNCATE TABLE sh.t1 DEFERRED INVALIDATION;

Table truncated.

SQL> SELECT object_status, ddl_no_invalidate,
           is_rolling_invalid, is_rolling_refresh_invalid,
           invalidations, sql_text
      FROM v$sql
     WHERE sql_text like 'SELECT%sh.t1%'
     AND   sql_text not like '%ddl%';
2      3      4      5      6
OBJECT_STATUS      D I I INVALIDATIONS SQL_TEXT
-----
INVALID_UNAUTH     N N N              1 SELECT * FROM sh.t1

SQL>

```

**Q1/** What do `DDL_NO_INVALIDATE`, `IS_ROLLING_INVALID`, and `IS_ROLLING_REFRESH_INVALID` column values set to `N` mean?

**A1/** The column values set to `N` mean that cursor deferred invalidation did not work. Remember that cursor deferred invalidation used for `TRUNCATE` statement is effective on partitioned tables only.

**Q2/** Which other columns can tell that cursor deferred invalidation did not work?

**A2/** The `OBJECT_STATUS` and `INVALIDATIONS` columns have been updated. There is now one invalidation on the current cursor.

- b. Reexecute the query using the cursor.

```
SQL> SELECT * FROM sh.t1;
```

```
no rows selected
```

```
SQL> SELECT object_status, ddl_no_invalidate,
           is_rolling_invalid, is_rolling_refresh_invalid,
           invalidations, sql_text
        FROM v$sql
        WHERE sql_text like 'SELECT%sh.t1%'
        AND    sql_text not like '%ddl%';
```

2	3	4	5	6	
OBJECT_STATUS		D	I	I	INVALIDATIONS SQL_TEXT
-----					
VALID		N	N	N	1 SELECT * FROM sh.t1

```
SQL>
```

*Q/ Is the cursor still invalid?*

**A/ No. The OBJECT\_STATUS has been updated to VALID. There has been one invalidation on the child cursor.**

4. Your goal is to be able to truncate the partitioned sh.sales table without invalidating the cursors opened.
- a. Query the old SALES\_Q1\_2001 partition of the table.

```
SQL> SELECT * FROM sh.sales partition (SALES_Q1_2001);
```

```
...
```

```
SQL>
```

*Q/ Is there any invalidation on the cursor of the SELECT statement on the table?*

```
SQL> SELECT object_status, ddl_no_invalidate,
           is_rolling_invalid, is_rolling_refresh_invalid,
           invalidations, sql_text
        FROM v$sql
        WHERE sql_text like 'SELECT%sh.sales partition%'
        AND    sql_text not like '%ddl%';
```

2	3	4	5	6	
OBJECT_STATUS		D	I	I	INVALIDATIONS SQL_TEXT
-----					
VALID		N	N	N	0 SELECT * FROM sh.sal es partition (SALES_ Q1_2001)

```
SQL>
```

**A/ There has been no DDL statement since the SELECT statement on the table and therefore no cursor invalidation.**

- b. Truncate the table without invalidating the cursor opened on the table.

```
SQL> TRUNCATE TABLE sh.sales DEFERRED INVALIDATION;
```

Table truncated.

```
SQL> SELECT object_status, ddl_no_invalidate,
           is_rolling_invalid, is_rolling_refresh_invalid,
           invalidations, sql_text
       FROM v$sql
       WHERE sql_text like 'SELECT%sh.sales partition%'
       AND    sql_text not like '%ddl%';
```

2	3	4	5	6	
OBJECT_STATUS		D	I	I	INVALIDATIONS SQL_TEXT
-----					
VALID		N	N	Y	0 SELECT * FROM sh.sal es partition (SALES_ Q1_2001)

```
SQL>
```

**Q/ What does IS\_ROLLING\_REFRESH\_INVALID column value set to Y mean?**

**A/ IS\_ROLLING\_REFRESH\_INVALID column value set to Y means that this cursor is rolling invalidated and requires execution-time refresh. Some metadata is stale. So the metadata is refreshed at cursor start time, and the cursor can continue to be used with possibly suboptimal plan. The recompilation is deferred.**

- c. Reexecute the query using the cursor.

```
SQL> SELECT * FROM sh.sales partition (SALES_Q1_2001);
```

no rows selected

```
SQL> SELECT object_status, ddl_no_invalidate,
           is_rolling_invalid, is_rolling_refresh_invalid,
           invalidations, sql_text
       FROM v$sql
       WHERE sql_text like 'SELECT%sh.sales partition%'
       AND    sql_text not like '%ddl%';
```

2	3	4	5	6	
OBJECT_STATUS		D	I	I	INVALIDATIONS SQL_TEXT
-----					
VALID		N	N	X	0 SELECT * FROM sh.sal

```
es partition (SALES_  
Q1_2001)
```

```
SQL> EXIT  
$
```

*Q/ What does `IS_ROLLING_REFRESH_INVALID` column value set to X mean?*

*A/ `IS_ROLLING_INVALID` column value set to x means that this cursor is rolling invalidated, and the cursor has executed in this state. There are still no invalidations.*

## Practice 18-3: Using Advanced Index Compression

---

### Overview

In this practice, you will use Advanced Index Compression to reduce the storage for indexes.

### Tasks

1. Before starting the practice, execute the `$HOME/labs/TUNING/setup_index.sh` shell script that creates an index with low compression on the `sh.test` table in `pdb_orcl`.

```
$ $HOME/labs/TUNING/setup_index.sh
...
SQL> SELECT index_name , compression from dba_indexes
        WHERE index_name = 'I_TEST';
  2
INDEX_NAME                COMPRESSION
-----
I_TEST                    ADVANCED LOW
...
$
```

2. Verify the storage used by the low compressed index using the following PL/SQL block:

```
DECLARE
blkcnt_cmp pls_integer;
blkcnt_uncmp pls_integer;
row_cmp pls_integer;
row_uncmp pls_integer;
cmp_ratio pls_integer;
comptype_str varchar2(100);
BEGIN
DBMS_COMPRESSION.GET_COMPRESSION_RATIO
(
  scratchtbsname => 'USERS',
  ownname => 'SH',
  objname => 'I_TEST',
  subobjname => NULL,
  comptype => dbms_compression.COMP_INDEX_ADVANCED_LOW,
  blkcnt_cmp => blkcnt_cmp,
  blkcnt_uncmp => blkcnt_uncmp,
  row_cmp => row_cmp,
  row_uncmp => row_uncmp,
  cmp_ratio => cmp_ratio,
  comptype_str => comptype_str,
  subset_numrows => dbms_compression.COMP_RATIO_MINROWS,
  objtype => dbms_compression.OBJTYPE_INDEX
);
```

```

DBMS_OUTPUT.PUT_LINE('Block count compressed = ' || blkcnt_cmp);
DBMS_OUTPUT.PUT_LINE('Block count uncompressed = ' ||
blkcnt_uncmp);
DBMS_OUTPUT.PUT_LINE('Row count per block compressed = ' ||
row_cmp);
DBMS_OUTPUT.PUT_LINE('Row count per block uncompressed = ' ||
row_uncmp);
DBMS_OUTPUT.PUT_LINE('Compression type = ' || comptype_str);
DBMS_OUTPUT.PUT_LINE('Compression ratio org= '||cmp_ratio);
END;
/

```

```

$ sqlplus sh@pdb_orcl
Enter password: *****
Connected.
SQL> SET serveroutput on
SQL> DECLARE
blkcnt_cmp pls_integer;
blkcnt_uncmp pls_integer;
row_cmp pls_integer;
row_uncmp pls_integer;
cmp_ratio pls_integer;
comptype_str varchar2(100);
BEGIN
DBMS_COMPRESSION.GET_COMPRESSION_RATIO
(
scratchtbsname => 'USERS',
ownname => 'SH',
objname => 'I_TEST',
subobjname => NULL,
comptype => dbms_compression.COMP_INDEX_ADVANCED_LOW,
blkcnt_cmp => blkcnt_cmp,
blkcnt_uncmp => blkcnt_uncmp,
row_cmp => row_cmp,
row_uncmp => row_uncmp,
cmp_ratio => cmp_ratio,
comptype_str => comptype_str,
subset_numrows => dbms_compression.COMP_RATIO_MINROWS,
objtype => dbms_compression.OBJTYPE_INDEX
);
DBMS_OUTPUT.PUT_LINE('Block count compressed = ' || blkcnt_cmp);
DBMS_OUTPUT.PUT_LINE('Block count uncompressed = ' ||
blkcnt_uncmp);

```



```

DBMS_OUTPUT.PUT_LINE('Row count per block compressed = ' ||
row_cmp);
DBMS_OUTPUT.PUT_LINE('Row count per block uncompressed = ' ||
row_uncmp);
DBMS_OUTPUT.PUT_LINE('Compression type = ' || comptype_str);
DBMS_OUTPUT.PUT_LINE('Compression ratio org= ' || cmp_ratio);
END;
/
  2      3      4      5      6      7      8      9      10     11     12     13     14
15     16     17     18     19     20     21     22     23     24     25     26     27
28     29     30     31     32
Block count compressed = 809
Block count uncompressed = 1029
Row count per block compressed = 648
Row count per block uncompressed = 510
Compression type = "Compress Advanced Low"
Compression ratio org= 1

PL/SQL procedure successfully completed.

SQL>

```

### 3. Use now Advanced High Compression.

```

SQL> DECLARE
blkcnt_cmp pls_integer;
blkcnt_uncmp pls_integer;
row_cmp pls_integer;
row_uncmp pls_integer;
cmp_ratio pls_integer;
comptype_str varchar2(100);
BEGIN
DBMS_COMPRESSION.GET_COMPRESSION_RATIO
(
scratchtbsname => 'USERS',
ownname => 'SH',
objname => 'I_TEST',
subobjname => NULL,
comptype => dbms_compression.COMP_INDEX_ADVANCED_HIGH,
blkcnt_cmp => blkcnt_cmp,
blkcnt_uncmp => blkcnt_uncmp,
row_cmp => row_cmp,
row_uncmp => row_uncmp,
cmp_ratio => cmp_ratio,
comptype_str => comptype_str,

```

```

subset_numrows => dbms_compression.COMP_RATIO_MINROWS,
objtype => dbms_compression.OBJTYPE_INDEX
);
DBMS_OUTPUT.PUT_LINE('Block count compressed = ' || blkcnt_cmp);
DBMS_OUTPUT.PUT_LINE('Block count uncompressed = ' ||
blkcnt_uncmp);
DBMS_OUTPUT.PUT_LINE('Row count per block compressed = ' ||
row_cmp);
DBMS_OUTPUT.PUT_LINE('Row count per block uncompressed = ' ||
row_uncmp);
DBMS_OUTPUT.PUT_LINE('Compression type = ' || comptype_str);
DBMS_OUTPUT.PUT_LINE('Compression ratio org= ' || cmp_ratio);
END;
/
  2      3      4      5      6      7      8      9      10     11     12     13     14
15     16     17     18     19     20     21     22     23     24     25     26     27
28     29     30     31     32
Block count compressed = 130
Block count uncompressed = 1029
Row count per block compressed = 4033
Row count per block uncompressed = 510
Compression type = "Compress Advanced High"
Compression ratio org= 8

PL/SQL procedure successfully completed.
SQL>

```

Q1/ Which of the two compression ratios is better?

**A1/ The Advanced High Compression ratio is much better than the Advanced Low Compression ratio.**

Q2/ Which operation would you be inclined to perform?

```

SQL> SELECT blocks FROM dba_segments
      WHERE segment_name='I_TEST';

      BLOCKS
-----
          896

SQL>

```

**A2/ Rebuild the index with Advanced High Compression.**

```

SQL> ALTER INDEX sh.i_test REBUILD COMPRESS ADVANCED HIGH;

Index altered.

```

```
SQL> SELECT blocks FROM dba_segments
        WHERE segment_name='I_TEST';
```

```

        BLOCKS
-----
        256
```

```
SQL> SELECT index_name, compression from dba_indexes
        WHERE index_name = 'I_TEST';
```

```

2
INDEX_NAME                                COMPRESSION
-----
I_TEST                                    ADVANCED HIGH
```

```
SQL>
```

*Q3/ What if the index was not compressed?*

```
SQL> ALTER INDEX sh.i_test REBUILD NOCOMPRESS;
```

```
Index altered.
```

```
SQL> SELECT blocks FROM dba_segments
        WHERE segment_name='I_TEST';
```

```

        BLOCKS
-----
        1152
```

```
SQL> SELECT index_name, compression from dba_indexes
        WHERE index_name = 'I_TEST';
```

```

2
INDEX_NAME                                COMPRESSION
-----
I_TEST                                    DISABLED
```

```
SQL> EXIT
```

```
$
```

**A3/ The gain of blocks is quite significant with Advanced High Compression.**

## Practice 18-4: Creating Real-Time MVs with OQC

### Overview

In this practice, you will create “On-Query Computation” (OQC) MVs to provide up-to-date result to the users, even if the current content is not in sync with the base tables.

### Tasks

1. In the previous practice, you truncated the partitioned sh.sales table. the \$HOME/labs/TUNING/setup\_sh.sh shell script that created the sh schema in pdb\_orcl. If this is not the case, reexecute it. If import errors appear, ignore them.

```
$ $HOME/labs/TUNING/setup_sh.sh
...
$
```

2. Create an MV log on sh.sales table and then an OQC MV on sh.sales.

```
$ sqlplus sh@pdb_orcl
Enter password: *****
Connected.
SQL> CREATE MATERIALIZED VIEW LOG ON sh.sales
      WITH ROWID, SEQUENCE (prod_id) INCLUDING NEW VALUES;
      2
Materialized view log created.

SQL> CREATE MATERIALIZED VIEW sh.mav_sum_sales
      REFRESH FAST ON COMMIT
      WITH PRIMARY KEY
      ENABLE ON QUERY COMPUTATION
      AS
      SELECT prod_id, sum(quantity_sold) as sum_qty,
             count(quantity_sold) as cnt_qty,
             sum(amount_sold) sum_amt,
             count(amount_sold) cnt_amt,
             count(*) as cnt_star
      FROM sh.sales
      GROUP BY prod_id;
      2      3      4      5      6      7      8      9     10     11
CREATE MATERIALIZED VIEW sh.mav_sum_sales
*
ERROR at line 1:
ORA-32360: cannot ENABLE ON QUERY COMPUTATION for a
refresh-on-commit materialized view

SQL>
```

Q/ Why does the operation fail?

SQL> !oerr ora 32360

32360, 00000, "cannot ENABLE ON QUERY COMPUTATION for a refresh-on-commit materialized view"

// \*Cause: The ENABLE ON QUERY COMPUTATION option was specified when the

// ON COMMIT REFRESH option was also specified.

// \*Action: Remove the ENABLE ON QUERY COMPUTATION option or specify refresh

// options other than ON COMMIT REFRESH.

**A/ You requested the data to be committed before refreshed in MVs during the MV definition. It is incompatible with OQC.**

3. Create an MV log on the oe.supplier table and then an OQC MV on oe.supplier.

```
SQL> CREATE MATERIALIZED VIEW LOG ON oe.supplier;
```

Materialized view log created.

```
SQL> CREATE MATERIALIZED VIEW mv_supplier
      REFRESH FAST ON DEMAND
      ENABLE ON QUERY COMPUTATION
      AS SELECT * FROM oe.supplier;
```

```
      2      3      4
AS SELECT * FROM oe.supplier
      *
```

ERROR at line 4:

ORA-32361: cannot ENABLE ON QUERY COMPUTATION for the materialized view

SQL>

Q/ Why does the operation fail?

!oerr ora 32361

32361, 00000, "cannot ENABLE ON QUERY COMPUTATION for the materialized view"

// \*Cause: On-query computation was not feasible for the materialized view

// being created or altered.

// \*Action: Remove the ENABLE ON QUERY COMPUTATION option or make the

// materialized view capable of on-query computation.

4. In the following example, you will use a very simple table so that you can see the difference between an OQC MV and a refresh fast MV.

- a. Create the simple table sh.t with four rows. Execute the

\$HOME/labs/TUNING/script\_t.sql SQL script.

```
SQL> @$HOME/labs/TUNING/script_t.sql
```

...

SQL>

- b. Create a fast refresh MV and an OQC MV on sh.t.

```
SQL> CREATE MATERIALIZED VIEW mv1
      REFRESH FAST
      AS
      SELECT a, sum(c) sumc, count(c) cntc, count(*) cntstar
      FROM sh.t
      GROUP BY a;
      2      3      4      5      6
Materialized view created.

SQL>
```

```
SQL> CREATE MATERIALIZED VIEW mv2
      REFRESH FAST ENABLE ON QUERY COMPUTATION
      AS
      SELECT a, sum(c) sumc, count(c) cntc, count(*) cntstar
      FROM sh.t
      GROUP BY a;
      2      3      4      5      6
Materialized view created.

SQL>
```

- c. Complete various DML statement on sh.t. Execute the \$HOME/labs/TUNING/script\_DML\_t.sql SQL script.

```
SQL> @$HOME/labs/TUNING/script_DML_t.sql
...
SQL>
```

- d. Compare data queried from the fast refresh MV and an OQC MV.

```
SQL> SELECT * FROM sh.mv1 ORDER BY a;

      A      SUMC      CNTC      CNTSTAR
-----
      1          1          1          1
      2          1          1          1
      3          1          1          1
      4          1          1          1

SQL> SELECT * FROM sh.mv2 ORDER BY a;

      A      SUMC      CNTC      CNTSTAR
-----
```

1	1	1	1
2	1	1	1
3	1	1	1
4	1	1	1

SQL>

Q1/ Is there any difference?

**A1/ No.**

Q2/ What is required to perform a direct access to the fresh data?

**A2/ Use the hint.**

SQL> SELECT /\*+ FRESH\_MV \*/ \* FROM sh.mv2 ORDER BY a;

A	SUMC	CNTC	CNTSTAR
-----	-----	-----	-----
2	2	2	2
3	2	2	2
4	2	2	2
5	2	2	2

SQL>

Q3/ Is the hint useful for a non-OQC MV?

SQL> SELECT /\*+ FRESH\_MV \*/ \* FROM sh.mv1 ORDER BY a;

A	SUMC	CNTC	CNTSTAR
-----	-----	-----	-----
1	1	1	1
2	1	1	1
3	1	1	1
4	1	1	1

SQL>

**A3/ No. The MV was not defined to provide direct access to the fresh data.**

Q4/ What would you have to do to get refreshed data from MV1?

**A4/ Refresh the base tables with MV.**

SQL> EXEC dbms\_mview.refresh('mv1', 'f')

PL/SQL procedure successfully completed.

SQL> SELECT \* FROM sh.mv1 ORDER BY a;

A	SUMC	CNTC	CNTSTAR
---	------	------	---------

```
-----
      2      2      2      2
      3      2      2      2
      4      2      2      2
      5      2      2      2

SQL> SELECT mview_name, on_query_computation
      FROM dba_mviews
      WHERE mview_name like 'MV%';
2      3

MVIEW_NAME      ON_QUERY_COMPUTATION
-----
MV1              N
MV2              Y

SQL> EXIT
$
```



## Practice 18-5: Creating On-Statement MVs on MJVs (Optional)

### Overview

In this practice, you want to continue using the MJVs and avoid the overhead introduced by MV logging on each of the base tables in case of fast refresh and the time required to complete the commit, slightly longer because of the extra processing involved. Nevertheless, you still want the MV to be refreshed all the time.

### Tasks

1. Before starting the practice, install the `sh` sample schema in `pdb_orcl` so as to have several tables to join in the MJV creation. If import errors appear, ignore them.

```
$ $HOME/labs/TUNING/setup_full_sh.sh
...
$
```

2. Create an ordinary “on commit” MJV on three `sh` tables.

```
$ sqlplus sh@pdb_orcl
Enter password: *****
Connected.
SQL> CREATE MATERIALIZED VIEW sh.mv_on_commit
      REFRESH FORCE ON COMMIT
      AS
      SELECT s.rowid sales_rid, c.cust_first_name first_name,
             c.cust_last_name last_name, p.prod_name prod_name,
             s.quantity_sold quantity_sold, s.amount_sold amount_sold
      FROM sh.sales s, sh.customers c, sh.products p
      WHERE s.cust_id = c.cust_id AND s.prod_id = p.prod_id;
2      3      4      5      6      7      8
Materialized view created.

SQL>
```

3. Create an “on statement” MJV on the same three `sh` tables.

```
SQL> CREATE MATERIALIZED VIEW sh.mv_on_statement
      REFRESH FAST ON STATEMENT
      USING TRUSTED CONSTRAINT AS
      SELECT s.rowid sales_rid, c.cust_first_name first_name,
             c.cust_last_name last_name, p.prod_name prod_name,
             s.quantity_sold quantity_sold, s.amount_sold amount_sold
      FROM sh.sales s, sh.customers c, sh.products p
      WHERE s.cust_id = c.cust_id AND s.prod_id = p.prod_id;
2      3      4      5      6      7      8
Materialized view created.
```

```
SQL>
```

4. Verify the type of MVs created.

```
SQL> SELECT mview_name, refresh_mode, refresh_method,
           last_refresh_type
        FROM dba_mviews
        WHERE mview_name like 'MV_ON%';
 2      3      4
MVIEW_NAME                REFRESH_M REFRESH_ LAST_REF
-----
MV_ON_STATEMENT            STATEMENT FAST      COMPLETE
MV_ON_COMMIT               COMMIT    FORCE     FAST_PCT

SQL>
```

5. Compare the behavior after performing a DML statement on one of the base tables, between the “on commit” MV and the “on statement” MV.

```
SQL> INSERT INTO sh.sales
      VALUES (1,100,TO_DATE('2003-07-01 00:00:00',
                           'YYYY-MM-DD HH24:MI:SS'),1,1,10,10);

1 row created.

SQL> SELECT * FROM sh.mv_on_commit
      WHERE quantity_sold=10 AND amount_sold=10;
 2

no rows selected

SQL> SELECT * FROM sh.mv_on_statement
      WHERE quantity_sold=10 AND amount_sold=10;
 2

SALES_RID                FIRST_NAME
-----
LAST_NAME
-----
PROD_NAME                                QUANTITY_SOLD
-----
AMOUNT_SOLD
-----
AAASbcAAqAAATWIAAB Yvette
Mossberg
P1
```

10

```

10

SQL> COMMIT;

Commit complete.

SQL> SELECT * FROM sh.mv_on_commit
      WHERE quantity_sold=10 AND amount_sold=10;
2

SALES_RID          FIRST_NAME
-----
LAST_NAME
-----
PROD_NAME          QUANTITY_SOLD
-----
AMOUNT_SOLD
-----
AAASbcAAqAAATWIAAB Yvette
Mossberg
P1                                10
10

SQL>

```

*Q/ What would happen to the data in MJV if the transaction is rolled back?*

```

SQL> INSERT INTO sh.sales
      VALUES (1,100,TO_DATE('2003-07-01 00:00:00',
                             'SYYYY-MM-DD HH24:MI:SS'),1,1,20,20);

1 row created.

SQL> SELECT * FROM sh.mv_on_statement
      WHERE quantity_sold=20 AND amount_sold=20;
2

SALES_RID          FIRST_NAME
-----
LAST_NAME
-----
PROD_NAME          QUANTITY_SOLD
-----
AMOUNT_SOLD
-----

```

```

-----
AAASbcAAqAAATWIAAB Yvette
Mossberg
P1                                     20
                20

SQL> ROLLBACK;

Rollback complete.

SQL> SELECT * FROM sh.mv_on_statement
      WHERE quantity_sold=20 AND amount_sold=20;
2

no rows selected

SQL> EXIT
$

```

***A/ Data is rolled back in the “on statement” MJV too.***

# **Practices for Lesson 19: Partitioning Enhancements**

## **Chapter 19**

## Practices for Lesson 19: Overview

---

### Practices Overview

In the practices for this lesson, you will use the new partitioning auto-list strategy as well as the multi-column list partitioning strategy. You will also practice the conversion of existing partitioned tables that require manual intervention when adding partitions to partitioned tables that will benefit from automatic partitions creation. And finally, you will filter data during partition maintenance operations.

## Practice 19-1: Creating Auto-list Partitioned Tables

### Overview

In this practice, you create auto-list partitioned tables. You will observe that as data is loaded into tables, the database creates new partitions if the loaded partitioning key values do not correspond to any of the existing partitions.

### Tasks

1. Before starting the practice, execute the `$HOME/labs/admin/glogin_19.sh` shell script. The shell script sets formatting for all columns selected in queries. In a previous practice, you ran the `$HOME/labs/IMDB/setup_tables.sh` script that created the `pdb_orcl`, then the `oe` user, the `lineorder`, `supplier`, and `date_dim` tables and finally loads the tables with data. You also ran the `$HOME/labs/TUNING/setup_sh.sh` shell script that created the `sh.sales` partitioned table in `pdb_orcl`. Execute the `$HOME/labs/PART/setup_app_sh.sh` shell script. It creates the `app_sh` user in `pdb_orcl`.

```
$ $HOME/labs/admin/glogin_19.sh
$ $HOME/labs/PART/setup_app_sh.sh
...
$
```

2. Your goal is to create the `sales` table in `app_sh` schema so that when a sale is inserted into the table, the row is stored into a partition corresponding to the state where the sale was committed. You do not have the time to create all partitions for all the states of America.
  - a. Create the auto-list partitioned table.

```
$ . oraenv
ORACLE_SID = [ORCL] ? ORCL
The Oracle base has been set to /u01/app/oracle
$ sqlplus app_sh@pdb_orcl
Enter password: *****
Connected.
SQL> CREATE TABLE app_sh.sales
      ( SALESMAN_ID  NUMBER(5),
        SALESMAN_NAME VARCHAR2(30),
        SALES_STATE  VARCHAR2(20),
        SALES_AMOUNT NUMBER(10),
        SALES_DATE   DATE
      )
      PARTITION BY LIST (SALES_STATE) AUTOMATIC
      (PARTITION P_CAL VALUES ('CALIFORNIA'));
  2      3      4      5      6      7      8      9
Table created.

SQL> SELECT owner, table_name, partitioning_type AS type,
```

```

        autolist, interval, interval_subpartition,
        autolist_subpartition
FROM    dba_part_tables
WHERE   owner IN ('SH','APP_SH') ;

```

2	3	4	5		
OWNER	TABLE_NAME		TYPE	AUTOLIST	INTERVAL
	INTERVAL_SUB		AUTOLIST_SUB		
APP_SH	SALES		LIST	YES	
		NO			
SH	COSTS		RANGE	NO	
		NO			
SH	SALES		RANGE	NO	
		NO			

```

SQL>

```

- b. Recreate the auto-list partitioned table to provide a default.

```

SQL> DROP TABLE app_sh.sales PURGE;

Table dropped.

SQL> CREATE TABLE app_sh.sales
      ( SALESMAN_ID  NUMBER(5),
        SALESMAN_NAME VARCHAR2(30),
        SALES_STATE  VARCHAR2(20),
        SALES_AMOUNT NUMBER(10),
        SALES_DATE   DATE
      )
      PARTITION BY LIST (SALES_STATE) AUTOMATIC
      (PARTITION P_CAL VALUES ('CALIFORNIA'),
       PARTITION state_unknown VALUES (DEFAULT));

```

2	3	4	5	6	7	8	9	10	11
CREATE TABLE app_sh.sales									
*									
ERROR at line 1:									
ORA-14851: DEFAULT [sub]partition cannot be specified for									
AUTOLIST [sub]partitioned objects.									

```

SQL>

```



Q/ Why is the operation not possible whereas the same operation completes with manual auto-list partitioning?

```
SQL> CREATE TABLE app_sh.sales
      ( SALESMAN_ID  NUMBER(5) ,
        SALESMAN_NAME VARCHAR2(30) ,
        SALES_STATE   VARCHAR2(20) ,
        SALES_AMOUNT  NUMBER(10) ,
        SALES_DATE    DATE
      )
      PARTITION BY LIST (SALES_STATE)
      (PARTITION P_CAL VALUES ('CALIFORNIA') ,
       PARTITION state_unknown VALUES (DEFAULT)) ;
2      3      4      5      6      7      8      9      10
Table created.

SQL>
```

***A/ Because an automatic partition will be created when a new key value is inserted in the table, there is no need to have a default partition where to put unknown values. This becomes a contradiction with the concept.***

```
SQL> DROP TABLE app_sh.sales PURGE;

Table dropped.

SQL> CREATE TABLE app_sh.sales
      ( SALESMAN_ID  NUMBER(5) ,
        SALESMAN_NAME VARCHAR2(30) ,
        SALES_STATE   VARCHAR2(20) ,
        SALES_AMOUNT  NUMBER(10) ,
        SALES_DATE    DATE
      )
      PARTITION BY LIST (SALES_STATE) AUTOMATIC
      (PARTITION P_CAL VALUES ('CALIFORNIA')) ;
2      3      4      5      6      7      8      9
Table created.

SQL>
```

c. Insert a sale made in Florida.

```
SQL> INSERT INTO app_sh.sales VALUES
      (1, 'Smith', 'FLORIDA', 100, sysdate);
2
1 row created.

SQL> COMMIT;
```

Commit complete.

SQL>

- d. Display the partitions created.

```
SQL> SELECT table_name, composite, partition_name, high_value
       FROM   user_tab_partitions;
```

2

TABLE_NAME	COM	PARTITION_NAME	HIGH_VALUE
SALES	NO	P_CAL	'CALIFORNIA'
SALES	NO	SYS_P268	'FLORIDA'

SQL>

## Practice 19-2: Converting Manual Partitioned Tables to Automatic Partitioned Tables

### Overview

In this practice, you will convert existing partitioned and subpartitioned tables so that you do not have to complete any manual intervention when new rows inserted require new partitions that do not exist.

### Tasks

1. Your goal is to convert the existing list partitioned tables into auto-list partitioned tables so that you do not have to perform any manual intervention when a new partition is required for sales being committed in other states.
  - a. Create a manual list partitioned table as it used to be before Oracle Database 12.2

```
SQL> CREATE TABLE app_sh.sales_country
      (SALES_ID NUMBER(5), COUNTRY VARCHAR2(20))
      PARTITION BY LIST (COUNTRY)
      ( PARTITION P_USA VALUES ('USA'),
        PARTITION P_CANADA VALUES ('CANADA'),
        PARTITION P_JAPAN VALUES ('JAPAN'));
```

2      3      4      5      6  
Table created.

```
SQL>
SQL> SELECT partitioning_type AS type, autolist
      FROM   user_part_tables
      WHERE  table_name = 'SALES_COUNTRY';
```

2      3      4  
TYPE   AUTOLIST  
-----  
LIST   NO

```
SQL> SELECT table_name, composite, partition_name, high_value
      FROM   user_tab_partitions
      WHERE  table_name = 'SALES_COUNTRY';
```

2      3

TABLE_NAME	COM	PARTITION_NAME	HIGH_VALUE
SALES_COUNTRY	NO	P_CANADA	'CANADA'
SALES_COUNTRY	NO	P_JAPAN	'JAPAN'
SALES_COUNTRY	NO	P_USA	'USA'

```
SQL>
```

- b. Insert sales committed in France and Germany.

```
SQL> @$HOME/labs/PART/insert3.sql
Connected.
INSERT INTO app_sh.sales_country VALUES (10, 'FRANCE')
          *
ERROR at line 1:
ORA-14400: inserted partition key does not map to any partition

INSERT INTO app_sh.sales_country VALUES (12, 'GERMANY')
          *
ERROR at line 1:
ORA-14400: inserted partition key does not map to any partition

Commit complete.
SQL>
```

- c. It fails. Convert the table to an auto list partitioned table.

```
SQL> ALTER TABLE app_sh.sales_country
      SET PARTITIONING AUTOMATIC;

2
Table altered.

SQL> SELECT partitioning_type AS type, autolist
      FROM   user_part_tables
      WHERE  table_name = 'SALES_COUNTRY';

2      3      4
TYPE    AUTOLIST
-----
LIST    YES

SQL>
```

- d. Insert sales committed in France and Germany.

```
SQL> @$HOME/labs/PART/insert3.sql
...
SQL>
```

- e. Check that new partitions are created.

```
SQL> SELECT table_name, composite, partition_name, high_value
      FROM   user_tab_partitions
      WHERE  table_name = 'SALES_COUNTRY';

2      3
TABLE_NAME      COM PARTITION_NAME HIGH_VALUE
-----
SALES_COUNTRY   NO  P_CANADA          'CANADA'
```

SALES_COUNTRY	NO	P_JAPAN	'JAPAN'
SALES_COUNTRY	NO	P_USA	'USA'
SALES_COUNTRY	NO	SYS_P306	'FRANCE'
SALES_COUNTRY	NO	SYS_P307	'GERMANY'

SQL>

***If you need to convert the same table back to manual, use ALTER TABLE  
<table\_name> SET PARTITIONING MANUAL;***

## Practice 19-3: Partitioning Tables with Multi-column List Partitioning

### Overview

In this practice, you will use multi-column auto-list partitioning.

### Tasks

1. Create a partitioned table using a multi-column list strategy. The partition key values rely on the state and channel of each sale.

```
SQL> CREATE TABLE app_sh.sales_by_region_and_channel
      ( deptno number, deptname varchar2(20),
        quarterly_sales number(10, 2),
        state varchar2(2),
        channel varchar2(1))
PARTITION BY LIST (state, channel)
(PARTITION q1_northwest_direct VALUES (('OR', 'D'),
                                         ('WA', 'D')),
 PARTITION q1_northwest_indirect VALUES (('OR', 'I'),
                                           ('WA', 'I')),
 PARTITION q1_southwest_direct VALUES (('AZ', 'D'),
                                         ('UT', 'D'),
                                         ('NM', 'D')),
 PARTITION q1_ca_direct VALUES ('CA', 'D'),
 PARTITION direct VALUES (DEFAULT, 'D'),
 PARTITION rest VALUES (DEFAULT));
 2      3      4      5      6      7      8      9     10     11     12     13     14
15     16          PARTITION direct VALUES (DEFAULT, 'D'),
                                     *
ERROR at line 15:
ORA-14318: DEFAULT partition must be last partition specified

SQL>
```

*Q/ Why does it fail?*

***A/ If DEFAULT is specified, no other value can be specified.***

```
SQL> CREATE TABLE app_sh.sales_by_region_and_channel
      ( deptno number, deptname varchar2(20),
        quarterly_sales number(10, 2),
        state varchar2(2),
        channel varchar2(1))
PARTITION BY LIST (state, channel)
(PARTITION q1_northwest_direct VALUES (('OR', 'D'),
                                         ('WA', 'D')),
 PARTITION q1_northwest_indirect VALUES (('OR', 'I'),
                                           ('WA', 'I')),
 PARTITION q1_southwest_direct VALUES (('AZ', 'D'),
                                         ('UT', 'D'),
                                         ('NM', 'D')),
 PARTITION q1_ca_direct VALUES ('CA', 'D'),
 PARTITION direct VALUES (DEFAULT, 'D'),
 PARTITION rest VALUES (DEFAULT));
```

```

        PARTITION q1_southwest_direct VALUES (('AZ', 'D'),
                                                ('UT', 'D'),
                                                ('NM', 'D')),
        PARTITION q1_ca_direct VALUES ('CA', 'D'),
        PARTITION rest VALUES (DEFAULT));
2      3      4      5      6      7      8      9     10     11     12     13     14
15
Table created.

SQL>

```

2. Insert rows.

```

SQL> @$HOME/labs/PART/insert4.sql
...
SQL>

```

3. Read the rows.

```

SQL> SELECT state, channel
      FROM sales_by_region_and_channel
      PARTITION (q1_northwest_direct);

2      3
STATE   CHANNEL
-----
WA      D

SQL> SELECT state, channel
      FROM sales_by_region_and_channel
      PARTITION (q1_northwest_indirect);

2      3
no rows selected

SQL> SELECT state, channel
      FROM sales_by_region_and_channel
      PARTITION (q1_southwest_direct);

2      3
no rows selected

SQL> SELECT state, channel
      FROM sales_by_region_and_channel
      PARTITION (q1_ca_direct);

2      3
STATE   CHANNEL
-----
CA      D

```

```
SQL> SELECT state, channel
      FROM sales_by_region_and_channel
      PARTITION (rest);
```

```

  2      3
STATE    CHANNEL
-----
OR       W
CA       I
FL
TX       W
        W
```

```
SQL>
```

4. List the partitions and their values.

```
SQL> COL partition_name FORMAT A21
SQL> COL high_value FORMAT A44
SQL> SELECT partition_name, high_value
      FROM user_tab_partitions
      WHERE table_name = 'SALES_BY_REGION_AND_CHANNEL';
```

```

  2      3
PARTITION_NAME          HIGH_VALUE
-----
Q1_CA_DIRECT            ( 'CA', 'D' )
Q1_NORTHWEST_DIRECT     ( 'OR', 'D' ), ( 'WA', 'D' )
Q1_NORTHWEST_INDIRECT   ( 'OR', 'I' ), ( 'WA', 'I' )
Q1_SOUTHWEST_DIRECT     ( 'AZ', 'D' ), ( 'UT', 'D' ), ( 'NM', 'D'
)
REST                     DEFAULT
```

```
SQL>
```



## Practice 19-4: Protecting Partitions Data from DML

### Overview

In this practice, you will manage the data in partitioned tables so that archived data cannot be updated before being moved to another storage tier by ADO (Automatic Data Optimization).

### Tasks

1. You decide that all partitions of `app_sh.sales3` that contain sales committed in California and older than 2002 should not be updatable.
  - a. Create the partitioned table.

```
SQL> CREATE TABLE app_sh.sales3
( SALESMAN_ID  NUMBER(5),
  SALESMAN_NAME VARCHAR2(30),
  SALES_STATE   VARCHAR2(20),
  SALES_AMOUNT  NUMBER(10),
  SALES_DATE    DATE )
PARTITION BY LIST (SALES_STATE)
SUBPARTITION BY RANGE (SALES_DATE)
SUBPARTITION TEMPLATE
(SUBPARTITION SP1_2002 values less than (TO_DATE('1-1-
2002', 'dd-mm-yyyy')),
 SUBPARTITION SP2_2010 values less than (TO_DATE('1-1-
2010', 'dd-mm-yyyy')),
 SUBPARTITION SP2_2016 VALUES LESS THAN (MAXVALUE))
( PARTITION P_CA VALUES ('CALIFORNIA'),
  PARTITION P_PA VALUES ('PENNSYLVANIA'));
```

Table created.

```
SQL> COL high_value FORMAT A78
```

```
SQL> SELECT partition_name, subpartition_name, high_value
FROM   user_tab_subpartitions
WHERE  table_name = 'SALES3';
```

```
2      3
PARTITION_NAME  SUBPARTITION_
-----
HIGH_VALUE
-----
P_CA            P_CA_SP1_2002
TO_DATE(' 2002-01-01 00:00:00', 'SYYYY-MM-DD HH24:MI:SS',
'NLS_CALENDAR=GREGORIA
P_CA            P_CA_SP2_2010
```

```

TO_DATE(' 2010-01-01 00:00:00', 'SYYYY-MM-DD HH24:MI:SS',
'NLS_CALENDAR=GREGORIA

P_CA                P_CA_SP2_2016
MAXVALUE

P_PA                P_PA_SP1_2002
TO_DATE(' 2002-01-01 00:00:00', 'SYYYY-MM-DD HH24:MI:SS',
'NLS_CALENDAR=GREGORIA

P_PA                P_PA_SP2_2010
TO_DATE(' 2010-01-01 00:00:00', 'SYYYY-MM-DD HH24:MI:SS',
'NLS_CALENDAR=GREGORIA

P_PA                P_PA_SP2_2016
MAXVALUE

6 rows selected.

SQL>

```

2. Insert rows into the table.

```

SQL> @$HOME/labs/PART/insert2.sql
...
SQL>

```

3. Select the subpartitions, make them read only, and verify that no sales in California older than 2002 can be modified nor inserted.

```

SQL> ALTER TABLE app_sh.sales3
      MODIFY SUBPARTITION P_CA_SP1_2002 READ ONLY;

2
Table altered.

SQL> ALTER TABLE app_sh.sales3
      MODIFY SUBPARTITION P_PA_SP1_2002 READ ONLY;

2
Table altered.

SQL> SELECT sales_state, sales_date
      FROM   app_sh.sales3 SUBPARTITION (p_ca_sp1_2002);

2
SALES_STATE      SALES_DAT
-----
CALIFORNIA      01-JAN-01

```

```

SQL> UPDATE app_sh.sales3 SET sales_amount=200
      WHERE sales_state = 'CALIFORNIA';
UPDATE app_sh.sales3 SET sales_amount=200
      *
ERROR at line 1:
ORA-14466: Data in a read-only partition or subpartition cannot
be modified.

SQL> INSERT INTO app_sh.sales3
      VALUES (14,'Smith','CALIFORNIA',100,TO_DATE('2001-12-31
00:00:00','SYYYY-MM-DD HH24:MI:SS','NLS_CALENDAR=GREGORIAN'));
  2  INSERT INTO app_sh.sales3
      *
ERROR at line 1:
ORA-14466: Data in a read-only partition or subpartition cannot
be modified.

SQL> INSERT INTO app_sh.sales3
      VALUES (15,'Smart','CALIFORNIA',100,TO_DATE('2005-01-01
00:00:00','SYYYY-MM-DD HH24:MI:SS','NLS_CALENDAR=GREGORIAN'));
  2
1 row created.

SQL> COMMIT;

Commit complete.

SQL>

```

*Q/ Did the last insert statement insert rows in the read only subpartition?*

***A/ No. The insert statement either inserted the row in an existing subpartition or created a new subpartition.***

```

SQL> SELECT partition_name, subpartition_name, read_only
      FROM   user_tab_subpartitions
      WHERE  table_name = 'SALES3';
  2      3
PARTITION_NAME SUBPARTITION_ REA
-----
P_CA           P_CA_SP1_2002 YES
P_CA           P_CA_SP2_2010 NO
P_CA           P_CA_SP2_2016 NO
P_PA           P_PA_SP1_2002 YES
P_PA           P_PA_SP2_2010 NO

```

```
P_PA          P_PA_SP2_2016 NO
```

```
6 rows selected.
```

```
SQL> EXIT
```

```
$
```

## Practice 19-5: Applying Filter Conditions to Partition Operation

### Overview

In this practice, you want to move old orders of the partitions of the `sh.sales` table related to year 1998 (partition `SALES_Q1_1998`) whose amount sold did not reach 8 dollars. These orders are moved to a new partition in another tablespace where compression takes place.

### Tasks

1. Before starting the practice, execute the `$HOME/labs/RMAN/setup_partition.sh` shell script. The script creates the `sh.sales` partitioned table in `pdb_orcl` using Data Pump import. Ignore possible errors that may show during the import.

```
$ $HOME/labs/RMAN/setup_partition.sh
...
$
```

2. Move the orders of the partition of the `sh.sales` table related to the first quarter of year 1998 whose amount sold did not reach 8 dollars to a tablespace where old records get compressed.
  - a. First, count the number of rows concerned by the criteria.

```
$ sqlplus system@pdb_orcl
Enter password: *****
Connected.
SQL> SELECT count(*)
      FROM    sh.sales
      WHERE   time_id <= TO_DATE('1998-03-31', 'SYYYY-MM-DD');

  2      3
COUNT (*)
-----
      43687

SQL> SELECT count(*) FROM sh.sales PARTITION (SALES_Q1_1998);

COUNT (*)
-----
      43687

SQL>
```

- b. Move the filtered rows to a new partition in the `TBS_FORCOMPRESS` tablespace.

```
SQL> ALTER TABLE sh.sales
      MOVE PARTITION SALES_Q1_1998
      TABLESPACE tbs_forcompress COMPRESS
      INCLUDING ROWS WHERE amount_sold < 8
      ONLINE UPDATE INDEXES;

  2      3      4      5
```

Table altered.

SQL>

- c. Count the rows moved in the `history_1998` partition.

```
SQL> SELECT count(*)
      FROM    sh.sales
      WHERE   time_id <= TO_DATE('1998-03-31', 'SYYYY-MM-DD');

2
```

```
COUNT (*)
-----
          330
```

```
SQL> SELECT count(*) FROM sh.sales PARTITION (SALES_Q1_1998);

COUNT (*)
-----
          330

SQL>
```

```
SQL> EXIT
$
```

*Q/ What happened to the rows that did not match the criteria?*

***A/ Records that did not match the filter condition when the partition maintenance started are not preserved. They were automatically deleted.***

# **Practices for Lesson 20: Real-Time Database Operation Monitoring**

## **Chapter 20**

## Practices for Lesson 20: Overview

---

### Practices Overview

In the practices for this lesson, you monitor database operations in sessions in `pdb_orcl` executed by `sh`. You will use `DBMS_SQL_MONITOR` to start and end monitoring operations in sessions and use Enterprise Manager Database Express to monitor database operations execution in sessions.



## Practice 20-1: Monitoring Sessions

### Overview

In this practice, you will use Enterprise Manager Database Express to test various database operations executed in `pdb_orcl` in various sessions.

### Tasks

1. Before starting the practice, check if Enterprise Manager Database Express is started and accessible in `pdb_orcl` directly.
  - a. Configure the port number for Enterprise Manager Database Express for `pdb_orcl` PDB. (This terminal window will be called *System\_Window*).

```
$ . oraenv
ORACLE_SID = [ORCL] ? ORCL
The Oracle base remains unchanged with value /u01/app/oracle
$ sqlplus system@pdb_orcl
Enter password: *****

SQL> SHOW PARAMETER dispatchers
NAME                                TYPE                                VALUE
-----
dispatchers                        string                            (PROTOCOL=TCP) (SERVICE=ORCLXDB)
max_dispatchers                    integer
SQL>
```

- b. Select a port number used for Enterprise Manager Database Express for `pdb_orcl`.

```
SQL> SELECT dbms_xdb_config.gethttpsport FROM DUAL;

GETHTTPSPO
-----
0

SQL> EXEC dbms_xdb_config.sethttpsport(5530)

PL/SQL procedure successfully completed.

SQL>
```

- c. Verify that the listener is running and listens to the localhost (*yourserver*) using TCP protocol, the port 5530, the http presentation with RAW session data.

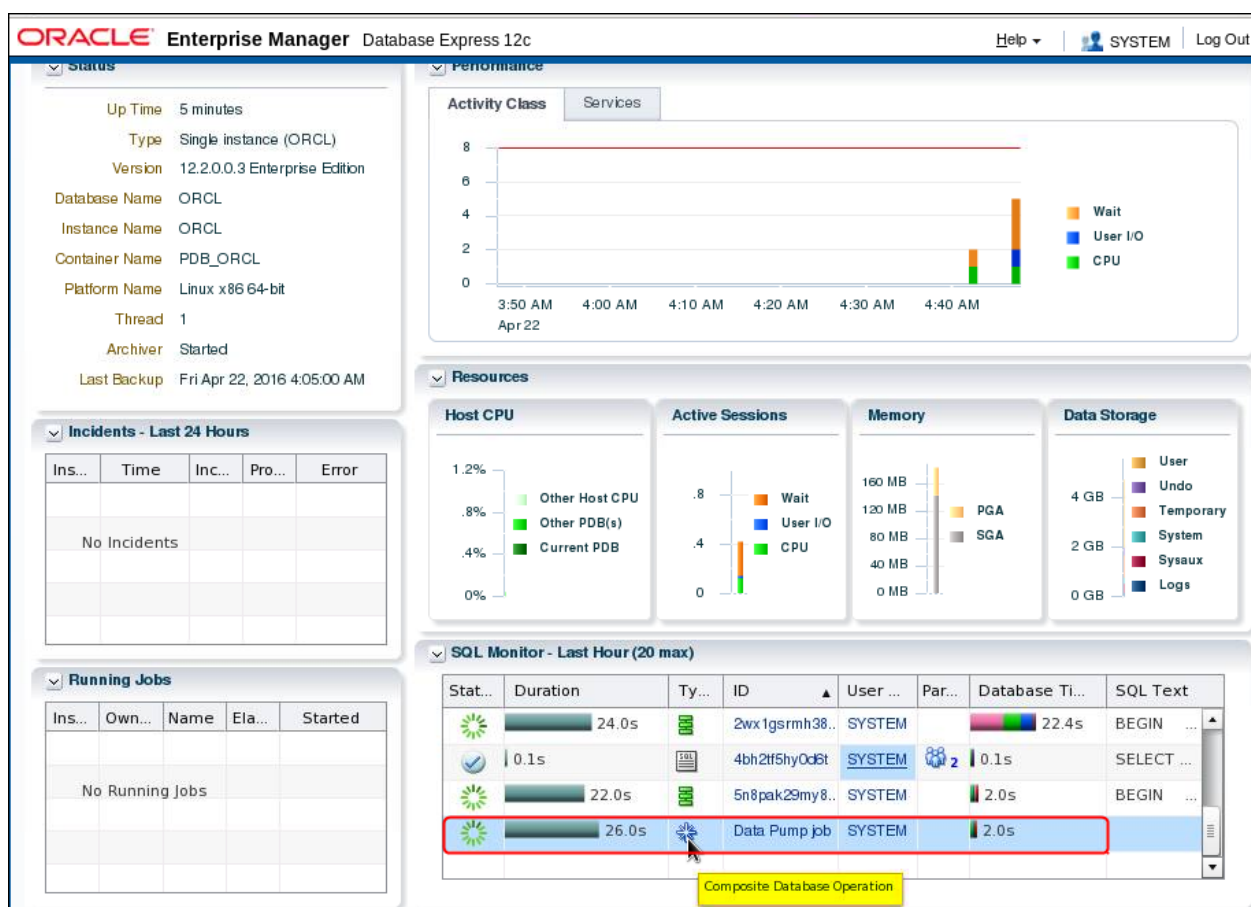
```
SQL> !lsnrctl status

...
Listening Endpoints Summary...
  (DESCRIPTION=(ADDRESS=(PROTOCOL=tcp) (HOST=<your
hostname>) (PORT=1521)))
  (DESCRIPTION=(ADDRESS=(PROTOCOL=tcps) (HOST=<your
hostname>) (PORT=5530)) (Security=(my_wallet_directory=/u01/app/or
acle/admin/ORCL/xdb_wallet)) (Presentation=HTTP) (Session=RAW))
...
SQL>
```

- d. Launch a browser and use the following URL <https://localhost:5530/em>.
- e. Most probably, you receive a Secure Connection Failed message and you need to add a security exception. At the end of the alert box, click **I Understand the Risks**.
- f. At the bottom of the page, click **Add Exception**.
- g. Confirm that “Permanently store this exception” is selected in your training environment and click **Confirm Security Exception**.
- h. Enter **system** in the User Name field. Enter the password in the Password field. Then click **Login**.
2. Before continuing the practice, execute the `$HOME/labs/admin/glogin_20.sh` and the `$HOME/labs/TUNING/setup_full_sh.sh` shell scripts. The first shell script sets formatting for all columns selected in queries and the second shell script recreates the `sh` schema in `pdb_orcl`. **While the second script is executing, go to Enterprise Manager Database Express home page.**

```
$ $HOME/labs/admin/glogin_20.sh
$ $HOME/labs/TUNING/setup_full_sh.sh
...
$
```

*Q/ Is there any DB operation monitored whereas you did not start any DB operation monitoring?*



**A/ Yes. The Data Pump import is automatically monitored. It would be also the case for any Data Pump export.**

- Open a new terminal window (*SH\_window1*) and connect to *pdb\_orcl* as *sh*.

```
$ . oraenv
ORACLE_SID = [ORCL] ? ORCL
The Oracle base remains unchanged with value /u01/app/oracle
$ sqlplus sh@pdb_orcl
Enter password: *****

SQL> SELECT sid,serial# FROM v$session where username='SH';

      SID      SERIAL#
-----
      627         7145

SQL>
```

4. Open another new terminal window (*SH\_window2*) and connect to *pdb\_orcl* as *sh*.

```
$ . oraenv
ORACLE_SID = [ORCL] ? ORCL
The Oracle base remains unchanged with value /u01/app/oracle
$ sqlplus sh@pdb_orcl
Enter password: *****

SQL> SELECT sid,serial# FROM v$session where username='SH';

      SID      SERIAL#
-----
      374      62954
      627      7145

SQL>
```

5. From the *System\_Window*, start monitoring the session connected in *SH\_window1* and the session connected in *SH\_window2*.

```
SQL> VAR dbop_eid1 NUMBER;
SQL> EXEC :dbop_eid1 :=
DBMS_SQL_MONITOR.BEGIN_OPERATION('ORA.session1', forced_tracking
=> 'Y', session_id => 627, session_serial => 7145);

PL/SQL procedure successfully completed.

SQL> VAR dbop_eid2 NUMBER;
SQL> EXEC :dbop_eid2 :=
DBMS_SQL_MONITOR.BEGIN_OPERATION('ORA.session2', forced_tracking
=> 'Y', session_id => 374, session_serial => 62954);

PL/SQL procedure successfully completed.

SQL>
```

6. View the DB operations monitored using Enterprise Manager Database Express. From Enterprise Manager Database Express home page, click the “Performance” menu, then the “Performance Hub” option, then the “Monitored SQL” tab. The list of database operations monitored appears.

Activity Workload <b>Monitored SQL</b>										
Top 100 By Last Active Time Type All Execution Detail SQL Detail Session Detail Kill Session										
Status	Duration	Ty...	ID	SQL Plan Hash	User Name	SQLMON_CONSUMER_GROUP	P...	Database Time	IO Requests	Start
	1.5m		ORA.session1		SH					4:51:48 AM
	1.5m		ORA.session2		SH					4:51:48 AM
Composite Database Operation										

**Note:** You could also use Enterprise Manager Cloud Control.

Q/ Which other mean do you have to view the various monitored sessions?

A/ The **v\$sql\_monitor** view provides information displayed in GUI tools.

```
SQL> SELECT dbop_name, dbop_exec_id AS id, status, cpu_time,
           buffer_gets AS buffers,
           physical_read_bytes AS physical_reads,
           to_char(FIRST_REFRESH_TIME, 'dd-mon-yy hh:mi')
             AS time_start,
           to_char(LAST_REFRESH_TIME , 'dd-mon-yy hh:mi')
             AS time_end
FROM   v$sql_monitor
WHERE  dbop_name IS NOT NULL
ORDER BY dbop_exec_id;
```

2	3	4	5	6	7	8	9	10
DBOP_NAME		ID	STATUS			CPU_TIME	BUFFERS	PHYSICAL_READS
TIME_START			TIME_END					
Data Pump job	1	DONE				1945000	806	11128832
22-apr-16 04:47		22-apr-16 04:48						
ORA.session1	2	EXECUTING				0	0	0
22-apr-16 04:51		22-apr-16 04:58						
ORA.session2	3	EXECUTING				0	0	0
22-apr-16 04:51		22-apr-16 04:58						

```
SQL>
```

Q/ What do you observe in both DB operations monitored?

A/ **None of them consume any resources because no operations have been started in any sessions. They are in EXECUTING status whereas the Data Pump import is in DONE status having consumed cpu and physical reads resources.**

- In **SH\_window1**, execute the `$HOME/labs/OP/queries_in_1.sql` SQL script. The script performs several **SELECT** statements as **SH** user. **Move on to the next step while the statements are running.** You will have time to view the database operation **ORA.session1** in EM Database Express until you execute **EXEC DBMS\_SQL\_MONITOR.END\_OPERATION.**

```
SQL> @$HOME/labs/OP/queries_in_1.sql
...
...
918843 rows selected.

SQL>
```

8. In *SH\_window2*, execute the `$HOME/labs/OP/queries_in_2.sql` SQL script. The script performs one `SELECT` statement as `SH` user. **Move on to the next step while the statements are running in both sessions, to monitor with EM Database Express the database operations in both sessions.** You will have time to view the database operation `ORA.session1` in EM Database Express until you execute `EXEC DBMS_SQL_MONITOR.END_OPERATION`. Set the time refresh to 5 seconds.

```
SQL> @$HOME/labs/OP/queries_in_2.sql
...
918843 rows selected.

SQL>
```

*Q/ What does the DB operation monitoring clearly show here?*

Monitored SQL										
Top 100 By		Last Active Time	Type	All	Execution Detail	SQL Detail	Session Detail	Kill Session		
Status	Duration	Ty...	ID	SQL Plan Hash	User Name	SQLMON_CONSUMER_GROUP	P...	Database Time	IO Requests	Start
	11.3m		ORA.session1		SH			2.4s	3	4:51:48 AM
	11.3m		ORA.session2		SH			2.3s		4:51:48 AM

***A/ Session1 consumed more IO requests than session2 and in fact session2 did not consume any IO requests. If you read the queries executed by both sessions, you see that they both executed the same query. Therefore the first query execution had to fetch the blocks into the buffer cache whereas the second query execution could benefit from the blocks being already in cache. This is clearly stated in the view below too.***

```
SQL> SELECT dbop_name, dbop_exec_id AS id, status, cpu_time,
           buffer_gets AS buffers,
           physical_read_bytes AS physical_reads,
           to_char(FIRST_REFRESH_TIME, 'dd-mon-yy hh:mi')
              AS time_start,
           to_char(LAST_REFRESH_TIME, 'dd-mon-yy hh:mi')
              AS time_end
FROM   v$sql_monitor
WHERE  dbop_name IS NOT NULL
ORDER BY dbop_exec_id;
```

2	3	4	5	6	7	8	9	10	
DBOP_NAME	ID	STATUS			CPU_TIME		BUFFERS	PHYSICAL_READS	
-----					-----				
TIME_START		TIME_END							
-----		-----							
Data Pump job	1	DONE			1945000		806	11128832	
22-apr-16 04:47		22-apr-16 04:48							
ORA.session1	2	EXECUTING			2446000		7	24576	
22-apr-16 04:51		22-apr-16 05:10							

```

ORA.session2      3 EXECUTING    3278000          0          0
22-apr-16 04:51 22-apr-16 05:10

SQL>

```

9. Once you consider having completed monitoring and possibly tuned the DB operations in session1 and session2, you can stop the DB operations monitoring.  
From the *System\_Window*, stop monitoring the DB operations in *SH\_window1* and in *SH\_window2*.

```

SQL> EXEC DBMS_SQL_MONITOR.END_OPERATION('ORA.session1',
:dbop_eid1)

PL/SQL procedure successfully completed.

SQL> EXEC DBMS_SQL_MONITOR.END_OPERATION('ORA.session2',
:dbop_eid2)

PL/SQL procedure successfully completed.

SQL> SELECT dbop_name, dbop_exec_id AS id, status, cpu_time,
           buffer_gets AS buffers,
           physical_read_bytes AS physical_reads,
           to_char(FIRST_REFRESH_TIME, 'dd-mon-yy hh:mi')
              AS time_start,
           to_char(LAST_REFRESH_TIME , 'dd-mon-yy hh:mi')
              AS time_end
FROM   v$sql_monitor
WHERE  dbop_name IS NOT NULL
ORDER BY dbop_exec_id;

```

2	3	4	5	6	7	8	9	10
DBOP_NAME	ID	STATUS		CPU_TIME		BUFFERS	PHYSICAL_READS	
-----	---	-----		-----		-----	-----	-----
TIME_START		TIME_END						
-----		-----						
Data Pump job	1	DONE		1945000		806	11128832	
22-apr-16 04:47		22-apr-16 04:48						
ORA.session1	2	EXECUTING		2446000		7	24576	
22-apr-16 04:51		22-apr-16 05:12						
ORA.session2	3	EXECUTING		3278000		0	0	
22-apr-16 04:51		22-apr-16 05:12						

```
SQL>
```

10. Refresh Enterprise Manager Database Express. The list of database operations monitored will disappear.

*Q/ Why do the DB operations still appear although you stopped monitoring?*

**A/ Even if the DB operations ran queries only, to complete the DB operation monitoring, the users in their respective sessions should commit (or even disconnect).**

- a. In *SH\_window1*, the user should commit.

```
SQL> COMMIT;
```

```
Commit complete.
```

```
SQL>
```

- b. In *SH\_window2*, the user should commit.

```
SQL> COMMIT;
```

```
Commit complete.
```

```
SQL>
```

Monitored SQL										
Top 100 By Last Active Time Type All Execution Detail SQL Detail Session Detail Kill Session										
Status	Duration	Ty...	ID	SQL Plan Hash	User Name	SOLMON_CONSUMER_GROUP	P...	Database Time	IO Requests	Start
	21.4m		ORA.session2		SH			3.3s		4:51:48 AM
	21.2m		ORA.session1		SH			2.4s	3	4:51:48 AM

- c. From the *System\_Window*, display the *V\$SQL\_MONITOR* view.

```
SQL> SELECT dbop_name, dbop_exec_id AS id, status, cpu_time,
           buffer_gets AS buffers,
           physical_read_bytes AS physical_reads,
           to_char(FIRST_REFRESH_TIME, 'dd-mon-yy hh:mi')
              AS time_start,
           to_char(LAST_REFRESH_TIME, 'dd-mon-yy hh:mi')
              AS time_end
FROM   v$sql_monitor
WHERE  dbop_name IS NOT NULL
ORDER BY dbop_exec_id;
```

2	3	4	5	6	7	8	9	10
DBOP_NAME		ID	STATUS		CPU_TIME		BUFFERS	PHYSICAL_READS
-----								
TIME_START		TIME_END						
-----								



```

Data Pump job    1 DONE          1945000      806      11128832
22-apr-16 04:47 22-apr-16 04:48

ORA.session1     2 DONE          2446000        7        24576
22-apr-16 04:51 22-apr-16 05:13

ORA.session2     3 DONE          3278000        0          0
22-apr-16 04:51 22-apr-16 05:13

SQL> EXIT
$

```



# **Practices for Lesson 21: Diagnosability**

## **Chapter 21**

## Practices for Lesson 21: Overview

---

### Practices Overview

In the practices for this lesson, you manage ADR space usage by using the ADR advisor that helps set the appropriate size retention policy. Then you will diagnose sudden issues by using the new TFA Collector utility.

## Practice 21-1: Using ADR Advisor to Set Appropriate ADR Target Size

### Overview

In this practice, you will have to configure the ADR space so that ADR diagnostics files do not exceed the disk space. The system administrator restricts the usage for ADR diagnostics files for all databases, `cdb1` and `ORCL`, to 200Mb. (This is a very small value used to demonstrate what happens in the case of the training session.)

### Tasks

1. Before starting the practice, execute the `$HOME/labs/admin/glogin_21.sh` script. The shell script sets formatting for all columns selected in queries. In a previous practice, you ran the `$HOME/labs/IMDB/setup_tables.sh` script that created the `pdb_orcl`, then the `oe` user, the `lineorder`, `supplier`, and `date_dim` tables and finally loads the tables with data. If this is not the case, execute the script now.

```
$ $HOME/labs/admin/glogin_21.sh
$ $HOME/labs/IMDB/setup_tables.sh
...
$
```

2. Limit the target size for ADR `cdb1` diagnostics files to 80Mb. If you upgraded `cdb1` to 12.2 in practice 7-3 and dropped it, or executed the `$HOME/labs/APP/cleanup_CDB1.sh` shell script at the beginning of practices 9, skip this task and go directly to the next task. If you did not upgrade `cdb1`, start it up.

```
$ . oraenv
ORACLE_SID = [ORCL] ? cdb1
The Oracle base remains unchanged with value /u01/app/oracle
$ adrci
ADRCI: Release 12.1.0.2.0 - Production on Mon Apr 25 23:15:14
2016

Copyright (c) 1982, 2014, Oracle and/or its affiliates. All
rights reserved.

ADR base = "/u01/app/oracle"
adrci> SET HOMEPATH diag/rdbms/cdb1/cdb1
adrci> SET CONTROL (SIZEP_POLICY = 84000000)
DIA-48314: Invalid ADR Control parameter [SIZEP_POLICY]
adrci> EXIT
$
```

*Q/ Why does it fail?*

*A/ The `SIZEP_POLICY` retention period parameter in the existing `SET CONTROL` command does not exist in the `ADRCI` tool 12.1 version.*

3. Use the Oracle Database 12.2 version of ADRCI.

```
$ . oraenv
ORACLE_SID = [ORCL] ? ORCL
The Oracle base remains unchanged with value /u01/app/oracle
$ adrci
ADRCI: Release 12.2.0.1.0 - Production on Mon Apr 25 23:47:37
2016

Copyright (c) 1982, 2016, Oracle and/or its affiliates. All
rights reserved.

ADR base = "/u01/app/oracle"
adrci> SET HOMEPATH diag/rdbms/cdb1/cdb1
adrci> SET CONTROL (SIZEP_POLICY = 84000000)
DIA-48322: Relation [ADR_CONTROL_AUX] of ADR V[2] incompatible
with V[2] tool
DIA-48210: Relation Not Found
DIA-48166: error with opening ADR block file because file does
not exist
[/u01/app/oracle/diag/rdbms/cdb1/cdb1/metadata/ADR_CONTROL_AUX.a
ms] [0]
adrci>
```

*Q/ Why does it still fail?*

*A/ Use the oerr utility to diagnose the error.*

**\$ oerr DIA 48322**

**48322, 00000, "Relation [%s] of ADR V[%s] incompatible with V[%s] tool"**

**// \*Document: YES**

**// \*Cause: the tool version is incompatible with the ADR version**

**// \*Action: use another version of tool and retry**

**As the error message says, the tool version (ADRCI: Release 12.2.0.1.0) is incompatible with the ADR version (the ADR for cdb1 is of Oracle 12.1.0.2.0 Database version). Therefore this setting is incompatible with any Oracle 12.1.0.2.0 Database version.**

4. Limit the target size for ADR ORCL diagnostics files to 200Mb as requested at the beginning.

```
adrci> SET HOMEPATH diag/rdbms/orcl/ORCL
adrci> SET CONTROL (SIZEP_POLICY = 200000000)
adrci>
```

*Q/ How can you verify the SIZEP\_POLICY value set?*

```
adrci> SELECT sizep_policy FROM adr_control_aux;

ADR Home = /u01/app/oracle/diag/rdbms/orcl/ORCL:
*****
```

```

SIZEP_POLICY
-----
200000000
1 row fetched

adrci> SELECT shortp_policy, longp_policy FROM adr_control;

ADR Home = /u01/app/oracle/diag/rdbms/orcl/ORCL:
*****
SHORTP_POLICY          LONGP_POLICY
-----
720                    8760
1 row fetched

adrci>

```

*Observe that the values for `SHORTP_POLICY` and `LONGP_POLICY` is in hour, which are still the default values (30 days for `SHORTP_POLICY` and 365 days for `LONGP_POLICY`).*

***A/ `V$DIAG_ADR_CONTROL` and `V$DIAG_ADR_CONTROL_AUX` views can also display the three retention parameters.***

- Back to the ADRCI session, predict how much space is going to be required to store the diagnostics for `cdb1` for a short retention period set to 8 days.

```

adrci> SET HOMEPATH diag/rdbms/cdb1/cdb1
adrci> ESTIMATE (SHORTP_POLICY = 192)
Estimate
Short Policy Hours: 0
Long Policy Hours: 0
Size Policy Bytes: 0
adrci>

```

*Q/ Why does it return 0 values for all three retention parameters?*

***A/ You are trying to use a new feature on a 12.1 ADR. The ADR advisor considers that there is no existing diagnostics to extrapolate information from to make space prediction.***

- Predict how much space is going to be required to store the diagnostics for `ORCL` for a short retention period set to 8 days.

```

adrci> SET HOMEPATH diag/rdbms/orcl/ORCL
adrci> ESTIMATE (SHORTP_POLICY = 192)
Estimate
Short Policy Hours: 192
Long Policy Hours: 8760
Size Policy Bytes: 8115316
adrci>

```

*Q1/ What does the advisor predict?*

**A1/ The ADR advisor says that for a short retention policy set to 8 days and a long retention policy kept to 365 days, only 7.73 Mb would be required. This prediction relies on the current ADR diagnostics files space used.**

**Q2/ What would the advisor predict for a long retention period set to 90 days?**

```
adrci> ESTIMATE (SHORTP_POLICY = 192, LONGP_POLICY = 2160)
Estimate
Short Policy Hours: 192
Long Policy Hours: 2160
Size Policy Bytes: 5805237
adrci>
```

**A2/ The ADR advisor says that for a short retention policy set to 8 days and a long retention policy kept to 90 days, only 5.53 Mb would be required. This prediction relies on the current ADR diagnostics files space used.**

**Q3/ Which short and long retention periods would the advisor predict for a ADR size target of 200Mb?**

```
adrci> ESTIMATE (SIZEP_POLICY = 209715200)
Estimate
Short Policy Hours: 25627
Long Policy Hours: 311798
Size Policy Bytes: 209715200
adrci>
```

**A3/ The ADR advisor says that for a size target set to 200Mb, the possible long retention could be set to 12991 days and the short retention could be set to 1067 days.**

7. Set the ADR target size to 100Mb.

```
adrci> SET CONTROL (sizep_policy = 104857600)
adrci>
```

8. Purge the ADR files down to 5Mb, without permanently changing the current retention policy settings.
- a. From another terminal window, check if in the ORCL ADR directory of your own database, the space used by ADR files exceeds 5Mb. If this is not the case, find an appropriate value so that you can verify later that sufficient ADR files were removed from the ADR to reach the target.

```
$ . oraenv
ORACLE_SID = [ORCL] ? ORCL
The Oracle base remains unchanged with value /u01/app/oracle
$ cd $ORACLE_BASE/diag/rdbms/orcl/ORCL
$ du -b -s .
10152727 .
$ du -h -s .
10M .
$
```



- b. The current space used by ADR files in this case exceeds 5Mb. It reaches 10Mb. More precisely purge the ADR log files down to 5Mb.

```
adrci> PURGE -size 5000000 -type log
adrci>
```

*Q1/ Did the command remove ADR files to release space in ADR?*

```
$ du -b -s .
10152727 .
$ du -h -s .
10M .
$
```

**A1/ No. The space used by ADR is still the same.**

*Q2/ If you do not understand why the `PURGE` command which completed successfully but did not achieve what was expected, how can you get some help?*

**A2/ Use the `HELP` command in `ADRCI`.**

```
adrci> HELP PURGE

Usage: PURGE [[-i <id1> | <id1> <id2>] |
              [[-age <mins>] |
              [-size <bytes>] |
              [-type
{ALERT|INCIDENT|TRACE|CDUMP|HM|UTSCDMP|LOG} ]]]

Purpose: Purge the diagnostic data in the current ADR home. If
no option is specified, the default purging policy will be used.

Options:

  [-i id1 | id1 id2]: Purge a single specified incident, or a
range of incidents.

  [-age <mins>]: Purge diagnostic data older than <mins> from
the ADR home, if the data is purgable.

  [-size <bytes>]: Purge diagnostic data from the ADR home
until the size of the home reaches <bytes> bytes.

  [-type ALERT|INCIDENT|TRACE|CDUMP|HM|UTSCDMP|LOG]: Purge a
specific type of data.

Notes:

  When purging by size, only INCIDENT, TRACE, CDUMP and
UTSCDMP data is considered.
```

Some data cannot be purged (such as incidents in the 'tracked' state), which means that the specified target size may not be reached in all cases.

Examples:

```
purge
purge -i 123 456
purge -age 60 -type incident
purge -size 10000000
```

adrci>

*Observe that when purging by size, only INCIDENT, TRACE, CDUMP, and UTSCDMP data is considered.*

```
adrci> PURGE -size 5000000
adrci> exit
$
```

*Q3/ Did the PURGE command achieve what was expected?*

```
$ du -h -s .
8.3M .
$
```

***A3/ It removed ADR files but could not reach the target size requested. Some ADR files cannot be removed. ADR contains a lot of different files: logs, traces, dumps, reports, metadata, internal repository files (such as lock files and relation files). Those files are purged in different ways. Some are always present, and cannot be purged at all. Others are purged by simply deleting the entire file. Some files, like relation files, are "purged" by purging parts of their internal structure.***

## Practice 21-2: Configuring TFA Collector and Analyzing Collections

### Overview

In this practice, you will encounter issues. You will therefore use the TFA Collector utility to collect and analyze the issues.

### Tasks

1. Before starting the practice, open another terminal window under `root` (it will be called *Root window*), patch TFA and execute the `/home/oracle/labs/TFA/setup_tfa.sh`. The patch may issue some errors. Ignore them. The shell script sets conditions under which you will discover how to handle the configuration of the utility.

```
$ su -
Password: *****
# cd /u01/app/oracle/tfa/host01/tfa_home/bin
# ./patchtfa.sh
...
# /home/oracle/labs/TFA/setup_tfa.sh
...
#
```

**Note:** If TFA was not installed at all, you can still install TFA by executing the `root.sh` script, and then patch it with `/u01/app/oracle/tfa/host01/tfa_home/bin/patchtfa.sh` shell script.

```
# pgrep -lf tfa
1264 /bin/sh /etc/init.d/init.tfa run
#
# /u01/app/oracle/product/12.2.0/dbhome_1/root.sh
Performing root user operation.

The following environment variables are set as:
    ORACLE_OWNER= oracle
    ORACLE_HOME= /u01/app/oracle/product/12.2.0/dbhome_1

Enter the full pathname of the local bin directory:
[/usr/local/bin]:
The contents of "dbhome" have not changed. No need to overwrite.
The contents of "oraenv" have not changed. No need to overwrite.
The contents of "coraenv" have not changed. No need to
overwrite.

Entries will be added to the /etc/oratab file as needed by
Database Configuration Assistant when a database is created
Finished running generic part of root script.
Now product-specific root actions will be performed.
```

```

Do you want to setup Oracle Trace File Analyzer (TFA) now ?
yes|no] : yes
Installing Oracle Trace File Analyzer (TFA) .
Log File:
/u01/app/oracle/product/12.2.0/dbhome_1/install/root_<your_hostn
ame>_2016-04-29_06-14-04.log
Finished installing Oracle Trace File Analyzer (TFA)
#

```

2. Ensure that TFA Collector runs.

a. Check the process.

```

# pgrep -lf tfa
24081 /bin/sh /etc/init.d/init.tfa run
24134 /u01/app/oracle/tfa/host01/tfa_home/jre/bin/java -Xms128m
-Xmx512m oracle.rat.tfa.TFAMain
/u01/app/oracle/tfa/host01/tfa_home
#

```

b. In the *Oracle window*, use TFA Collector to make the same verification.

```

$ /u01/app/oracle/tfa/bin/tfactl -help
User oracle does not have keys to run TFA. Please check with TFA
Admin(root)
$

```

*Q/ The TFA Collector tool is known and installed. Why does it not run?*

**A/ The oracle user is not authorized to use it. Connect as root and give access to oracle user to use TFA Collector.**

c. Switch to the *Root window* and launch the TFA Collector.

```

# /u01/app/oracle/tfa/bin/tfactl -help
Usage : /u01/app/oracle/tfa/bin/tfactl <command> [options]

commands:diagcollect|collection|analyze|ips|run|start|stop|enabl
e|disable|status|print|access|purge|directory|host|receiver|set|
toolstatus|uninstall|diagnosetfa
For help with a command: /u01/app/oracle/tfa/bin/tfactl
<command> -help
#

```

d. Add the `oracle` user as another user who can perform diagnostic files collection and analysis.

*Q1/ How do you proceed to get help on adding users to the TFA Collector configuration?*

**A1/ You can see from the help list that there is a access command to “Add or Remove or List TFA Users and Groups”.**

```

# /u01/app/oracle/tfa/bin/tfactl
tfactl> access -help
Add or Remove or List TFA Users and Groups

```

```
Usage : /u01/app/oracle/tfa/bin/tfactl access <command>
[options]
```

```
commands:lsusers|add|remove|block|unblock|enable|disable|reset|r
emoveall
```

For detailed help on each command use:

```
/u01/app/oracle/tfa/bin/tfactl access <command> -help
```

```
tfactl> access lsusers -help
```

List all the TFA Users

```
Usage : /u01/app/oracle/tfa/bin/tfactl lsusers [ -local ]
```

Examples:

```
/u01/app/oracle/tfa/bin/tfactl access lsusers
```

```
/u01/app/oracle/tfa/bin/tfactl access lsusers -local
```

```
tfactl> access lsusers
```

No Users in TFA Access Manager list in <your\_hostname>.

```
tfactl>
```

```
tfactl> access add -user oracle
```

Successfully added 'oracle' to TFA Access list.

```
.-----.
```

TFA Users in <your_hostname>		
User Name	User Type	Status
oracle	USER	Allowed

```
'-----'
```

```
tfactl>
```

*Q2/ Is the `oracle` user now able to use the TFA Collector? Switch to the Oracle window.*

```
$ /u01/app/oracle/tfa/bin/tfactl -help
```

```
Usage : /u01/app/oracle/tfa/bin/tfactl <command> [options]
```

```
commands:diagcollect|collection|analyze|ips|run|start|stop|print
|directory|toolstatus
```

For detailed help on each command use:

```
/u01/app/oracle/tfa/bin/tfactl <command> -help
```

\$

**A2/ The *oracle* user is now able to use the TFA Collector.**

**Q3/ Can the *oracle* user perform the same operations as *root* does?**

**A3/ From the commands listed in the help, the *oracle* user is limited in TFA Collector commands usage. The *oracle* user cannot stop and restart the TFA Collector, disable and enable the TFA autostart, update the configuration, uninstall TFA Collector, and provide access to other users.**

**Q4/ Find the command to display the nodes on which TFA Collector is running.**

**A4/ You can use the *print* command.**

```
$ /u01/app/oracle/tfa/bin/tfactl
```

```
tfactl> print -help
```

Print requested details.

Usage : /u01/app/oracle/tfa/bin/tfactl print <command> [options]

```
commands:status|components|config|directories|hosts|receivers|co
llectors|robjects|actions|repository|runmode|suspendedips|protoc
ols
```

For detailed help on each command use:

```
/u01/app/oracle/tfa/bin/tfactl print <command> -help
```

```
tfactl> print status
```

```
.------.
| Host      | Status of TFA | PID    | Port | Version      | Build
ID          | Inventory Status |        |      |              | ID
+-----+-----+-----+-----+-----+-----+
|<your_host>| RUNNING       | 17284  | 5000 | 12.2.0.1.0   | 
12201020160330005427 | COMPLETE      |        |      |              | 
'-----+'-----+'-----+'-----+'-----+'-----'
```

```
tfactl>
```

**Q5/ How do you find the list of the components for which the TFA Collector is able to collect diagnostics files?**

**A5/ Use the same *print* command.**

```
tfactl> print components
```

```
.------.
| Host      | Status of TFA | PID    | Port | Version      | Build
ID          | Inventory Status |        |      |              | ID
+-----+-----+-----+-----+-----+-----+
|<your_host>| RUNNING       | 17284  | 5000 | 12.2.0.1.0   | 
12201020160330005427 | COMPLETE      |        |      |              | 
'-----+'-----+'-----+'-----+'-----+'-----'
```

XML Components	
Field	Value
Name	RDBMS
Description	Database logs
Comp. Types	collection action
Configuration	all
Subcomponents	name:database required: default: name:instance required: default:
Name	ASM
Description	ASM logs
Comp. Types	collection
Configuration	all
Subcomponents	name:instance required: default:
Also collect	TNS ASMPROXY ASMIO
Name	DBWLM
Description	DBWLM logs
Comp. Types	collection
Configuration	all

Name	TNS
Description	TNS logs
Comp. Types	collection
Configuration	all
-----+	
Name	RHP
Description	RHP logs
Comp. Types	collection
Configuration	all
-----+	
Name	CRS
Description	CRS logs
Comp. Types	collection
Configuration	all
Also collect	OS
	RACDBCLOUD
-----+	
Name	WLS
Description	WLS logs
Comp. Types	collection
Configuration	all
-----+	
Name	EMAGENT
Description	EMAGENT logs
Comp. Types	collection



Configuration	all
-----	
Name	OMS
Description	OMS logs
Comp. Types	collection
Configuration	all
-----	
Name	OCM
Description	OCM logs
Comp. Types	collection
Configuration	all
-----	
Name	EMPLUGINS
Description	EMPLUGINS logs
Comp. Types	collection
Configuration	all
-----	
Name	EM
Description	EM logs
Comp. Types	collection
Configuration	all
Also collect	EMAGENT
	OCM
	OMS
	WLS

	EMPLUGINS
-----+	
Name	ACFS
Description	ACFS logs
Comp. Types	collection
Configuration	all
Also collect	CRS
	ASM
	OS
	ASMPROXY
-----+	
Name	INSTALL
Description	Oracle Installation related files
Comp. Types	collection
Configuration	all
-----+	
Name	CFGTOOLS
Description	CFGTOOLS logs
Comp. Types	collection
Configuration	all
Also collect	INSTALL
-----+	
Name	OS
Description	OS files such as /var/log/messages
Comp. Types	collection

Configuration	all
-----+	
Name	IPS
Description	Incident Packaging Service
Comp. Types	collection
Configuration	all
-----+	
Name	ODADOM0
Description	ODADOM0 logs
Comp. Types	collection action
Configuration	ODADom0
Also collect	OS
-----+	
Name	ODA
Description	ODA/OS logs
Comp. Types	collection
Configuration	ODA
Also collect	OS
	ODADOM0
-----+	
Name	EXADATA
Description	EXADATA
Comp. Types	collection
Configuration	EXADATA
-----+	

Name	COMPUTENODE
Description	COMPUTENODE
Comp. Types	collection
Configuration	EXADATA
+-----+	
Name	RACDBCLOUD
Description	Data and Logs for RAC DB Cloud Service
Comp. Types	collection action
Configuration	RACDBCLOUD
Also collect	OS
+-----+	
Name	ASMIO
Description	ASMIO
Comp. Types	collection
Configuration	ALL
+-----+	
Name	ASMPROXY
Description	ASMPROXY
Comp. Types	collection
Configuration	ALL
Also collect	ASM
+-----+	
Name	SUNDIAG
Description	sundiag logs
Comp. Types	action

Configuration	EXADATA
-----+	
Name	CHMOS
Description	CHMOS files (Note that this can be large for longer durations)
Comp. Types	action
Configuration	ALL
-----+	
Name	STORAGEMETRICS
Description	I/O metrics from Exadata storage cell(s)
Comp. Types	action
Configuration	EXADATA
-----+	
Name	AWRHTML
Description	AWRHTML logs
Comp. Types	action
Configuration	ALL
-----+	
Name	AWRTEXT
Description	AWRTEXT logs
Comp. Types	action
Configuration	ALL
-----+	
Name	EXADOM0
Description	EXADOM0 logs
Comp. Types	collection action

```

| Configuration | EXADATA
|
+-----+-----+
| Name          | ZDLRA
|
| Description    | ZDLRA logs
|
| Comp. Types   | collection action
|
| Configuration | EXADATA
|
| Also collect  | SUNDIAG
|
|                | CELL
|
|                | RDBMS
|
+-----+-----+
| Name          | ODA STORAGE
|
| Description    | ODA Storage logs and Data
|
| Comp. Types   | action
|
| Configuration | ODA
|
| Also collect  | ODA
|
|                | OS
|
|                | ASM
|
+-----+-----+
tfactl>

```

**Q6/ How do you find the current TFA Collector configuration?**

```

tfactl> print config
.-----+-----+
|                                     <your_hostname>
+-----+-----+
| Configuration Parameter              | Value
+-----+-----+
| TFA Version                          | 12.2.0.1.0
| Automatic Diagnostic Collection       | true
| Alert Log Scan                       | true

```

Trimming of files during diagcollection	true
Inventory Trace level	1
Collection Trace level	1
Scan Trace level	1
Other Trace level	1
Repository current size (MB)	17
Repository maximum size (MB)	10240
Max Size of TFA Log (MB)	50
Max Number of TFA Logs	10
Max Size of Core File (MB)	20
Max Collection Size of Core Files (MB)	200
Minimum Free Space to enable Alert Log Scan (MB)	500
Time interval between consecutive Disk Usage Snapshot(minutes)	60
Automatic Purging	true
Age of Purging Collections (Hours)	12
TFA IPS Pool Size	5

```
'-----+-----'
tfactl>
```

**A6/ Using the same *print* command, you can observe the automatic diagnostic collection is enabled.**

3. Change the “Age of Purging Collections” to 48 hours.

```
tfactl> help set
```

Turn ON/OFF or Modify various TFA features

```
Usage : /u01/app/oracle/tfa/bin/tfactl set [
autodiagcollect=<ON|OFF> | trimfiles=<ON|OFF> |
tracelevel=<COLLECT|SCAN|INVENTORY|OTHER>:<1|2|3|4> |
reposizeMB=<n> | repositorydir=<dir> [-force] | logsize=<n> [-
local] | logcount=<n> [-local] | maxcorefilesize=<n> [-local] |
maxcorecollectionsize=<n> [-local] autopurge=<ON|OFF> |
minSpaceForRTScan=<n> | rtscan=<ON|OFF> | diskUsageMon=<ON|OFF> |
| diskUsageMonInterval=<n> | minagetopurge=<n> |
tfaIpsPoolSize=<n> ] [-c]
```

```
autodiagcollect    allow for automatic diagnostic
collection when an event
```

```
is observed (default ON)
```

```
trimfiles          allow trimming of files during
diagcollection (default ON)
```

```
tracelevel         control the trace level of log files in
/u01/app/oracle/tfa/host01/log
```

```

                                (default 1 for all facilities)
    reposizeMB=<n>                set the maximum size of diagcollection
repository to <n>MB
    repositorydir=<dir>          set the diagcollection repository to
<dir>
    -force                        skip initial checks while changing
repository (Not Recommended)
    logsize=<n>                  set the maximum size of each TFA log to
<n>MB (default 50 MB)
    logcount=<n>                 set the maximum number of TFA logs to
<n> (default 10)
    maxcorefilesize=<n>          set the maximum size of Core File to
<n>MB (default 20 MB )
    maxcorecollectionsize=<n>    set the maximum collection size of
Core Files to <n>MB (default 200 MB )
    autopurge                    allow automatic purging of collections
when less space
                                is observed in repository (default OFF)
    minSpaceForRTScan=<n>        Minimum space required to run RT
Scanning(default 500)
    rtscan                      allow Alert Log Scanning
    diskUsageMon                allow Disk Usage Monitoring
    diskUsageMonInterval=<n>    Time interval between consecutive
Disk Usage Snapshot(default 60 minutes)
    minagetopurge               set the age in hours for collections to
be skipped by
                                AutoPurge (default 12 Hours)
    tfaIpsPoolSize              set the TFA IPS pool size
    -c                          set the value on all nodes (Does not
apply to repository
                                settings)
    -local                      set the value on the local node (if
option is not included
                                the value will be set on all the nodes)

```

#### Examples:

```

/u01/app/oracle/tfa/bin/tfactl set autodiagcollect=ON
/u01/app/oracle/tfa/bin/tfactl set autopurge=ON
/u01/app/oracle/tfa/bin/tfactl set tracelevel=INVENTORY:3
/u01/app/oracle/tfa/bin/tfactl set reposizeMB=20480
/u01/app/oracle/tfa/bin/tfactl set logsize=100

```

```
tfactl> set minagetopurge = 48
```

Access Denied: Only TFA Admin can run this command



\$

Switch to the *Root window*.

```
tfactl> set minagetopurge = 48
```

Invalid option =

Turn ON/OFF or Modify various TFA features

Usage : /u01/app/oracle/tfa/bin/tfactl set [  
autodiagcollect=<ON|OFF> | trimfiles=<ON|OFF> |  
tracelevel=<COLLECT|SCAN|INVENTORY|OTHER>:<1|2|3|4> |  
resizeMB=<n> | repositorydir=<dir> [-force] | logsize=<n> [-  
local] | logcount=<n> [-local] | maxcorefilesize=<n> [-local] |  
maxcorecollectionsize=<n> [-local] autopurge=<ON|OFF> |  
minSpaceForRTScan=<n> | rtscan=<ON|OFF> | diskUsageMon=<ON|OFF>  
| diskUsageMonInterval=<n> | minagetopurge=<n> |  
tfaIpsPoolSize=<n> ] [-c]

autodiagcollect allow for automatic diagnostic  
collection when an event

is observed (default ON)

trimfiles allow trimming of files during  
diagcollection (default ON)

tracelevel control the trace level of log files in  
/u01/app/oracle/tfa/host01/log  
(default 1 for all facilities)

resizeMB=<n> set the maximum size of diagcollection  
repository to <n>MB

repositorydir=<dir> set the diagcollection repository to  
<dir>

-force skip initial checks while changing  
repository (Not Recommended)

logsize=<n> set the maximum size of each TFA log to  
<n>MB (default 50 MB)

logcount=<n> set the maximum number of TFA logs to  
<n> (default 10)

maxcorefilesize=<n> set the maximum size of Core File to  
<n>MB (default 20 MB )

maxcorecollectionsize=<n> set the maximum collection size of  
Core Files to <n>MB (default 200 MB )

autopurge allow automatic purging of collections  
when less space

is observed in repository (default OFF)

minSpaceForRTScan=<n> Minimum space required to run RT  
Scanning(default 500)

rtscan allow Alert Log Scanning

diskUsageMon allow Disk Usage Monitoring

```

    diskUsageMonInterval=<n>   Time interval between consecutive
    Disk Usage Snapshot(default 60 minutes)

    minagetopurge              set the age in hours for collections to
    be skipped by

                                AutoPurge (default 12 Hours)

    tfaIpsPoolSize             set the TFA IPS pool size

    -c                         set the value on all nodes (Does not
    apply to repository

                                settings)

    -local                     set the value on the local node (if
    option is not included

                                the value will be set on all the nodes)

Examples:
    /u01/app/oracle/tfa/bin/tfactl set autodiagcollect=ON
    /u01/app/oracle/tfa/bin/tfactl set autopurge=ON
    /u01/app/oracle/tfa/bin/tfactl set tracelevel=INVENTORY:3
    /u01/app/oracle/tfa/bin/tfactl set resizeMB=20480
    /u01/app/oracle/tfa/bin/tfactl set logsize=100

tfactl>

```

Observe the syntax carefully. There is no space between the = and the value.

```

tfactl> set minagetopurge=48
Successfully set minFileAgeToPurge=48
.-----
|                                     <your_hostname>
+-----+-----+
| Configuration Parameter              | Value
+-----+-----+
| TFA Version                          | 12.2.0.1.0
| Automatic Diagnostic Collection       | true
| Alert Log Scan                       | true
| Trimming of files during diagcollection | true
| Inventory Trace level                | 1
| Collection Trace level               | 1
| Scan Trace level                    | 1
| Other Trace level                   | 1
| Repository current size (MB)         | 232
| Repository maximum size (MB)        | 10240
| Max Size of TFA Log (MB)             | 50
| Max Number of TFA Logs               | 10
| Max Size of Core File (MB)          | 20
| Max Collection Size of Core Files (MB) | 200
| Minimum Free Space to enable Alert Log Scan (MB) | 500

```

Time interval between consecutive Disk Usage Snapshot (minutes)	60
Automatic Purging	true
<b>Age of Purging Collections (Hours)</b>	<b>48</b>
TFA IPS Pool Size	5
'-----+-----'	

tfactl>

4. Run a collection under oracle.

- a. Switch to the *Oracle window* and collect diagnostics files related to ORCL database only generated within the last 5 days.

```
$ /u01/app/oracle/tfa/bin/tfactl

tfactl> diagcollect -help

Collect logs from across nodes in cluster

Usage : /u01/app/oracle/tfa/bin/tfactl diagcollect [ -all |
[component_name1] [component_name2] ... [component_nameN] ] [-
node <all|local|n1,n2,...>] [-tag <description>] [-z <filename>]
[-since <n><h|d>] [-from <time> -to <time>] [-for <time>] [-
nocopy] [-notrim] [-silent] [-nocores][collectalldirs][-
collectdir <dir1,dir2...>][-examples]

    components:-ips|-database|-asm|-crsclient|-dbclient|-dbwlm|-
tns|-rhp|-procinfo|-afd|-crs|-wls|-emagent|-oms|-ocm|-
emplugins|-em|-acfs|-install|-cfgtools|-os|-ips|-ashhtml|-
ashtext|-awrhtml|-awrtext

    -node                Specify comma separated list of host names
for collection

    -tag <tagname>       The files will be collected into tagname
directory inside

                        repository
    -z <zipname>          The files will be collected into tagname
directory with the

                        specified zipname
    -since <n><h|d>       Files from past 'n' [d]ays or 'n' [h]ours
    -from                "Mon/dd/yyyy hh:mm:ss"      From <time>
                        or "yyyy-dd-mm hh:mm:ss"
                        or "yyyy-dd-mmThh:mm:ss"
                        or "yyyy-dd-mm"
    -to                  "Mon/dd/yyyy hh:mm:ss"      To <time>
                        or "yyyy-dd-mm hh:mm:ss"
                        or "yyyy-dd-mmThh:mm:ss"
                        or "yyyy-dd-mm"
    -for                 "Mon/dd/yyyy"                For <date>.
```

```

                                or "yyyy-dd-mm"
    -nocopy                      Does not copy back the zip files to
initiating node from all nodes
    -notrim                     Does not trim the files collected
    -silent                     This option is used to submit the
diagcollection as a background
                                process
    -nocores                    Do not collect Core files when it would
normally have been
                                collected
    -collectalldirs             Collect all files from a directory marked
"Collect All"
                                flag to true
    -collectdir                 Specify comma separated list of directories
and collection will
                                include all files from these irrespective of
type and time constraints
                                in addition to components specified
    -examples                   Show diagcollect usage examples
For detailed help on each component use:
    /u01/app/oracle/tfa/bin/tfactl diagcollect [component_name1]
[component_name2] ... [component_nameN] -help

tfactl>

```

```

tfactl> diagcollect -database ORCL -since 5d
Collecting data for all nodes

Collection Id : 20160428102932<your_hostname>

Detailed Logging at :
/u01/app/oracle/tfa/repository/collection_Fri_Aug_05_11_20_46_GM
T_2016_node_all/diagcollect_20160805112046_host01.log
2016/08/05 11:20:50 GMT : Collection Name :
tfa_Fri_Aug_05_11_20_46_GMT_2016.zip
2016/08/05 11:20:50 GMT : Collecting diagnostics from hosts :
[host01]
2016/08/05 11:20:50 GMT : Scanning of files for Collection in
progress...
2016/08/05 11:20:50 GMT : Collecting additional diagnostic
information...
2016/08/05 11:20:55 GMT : Getting list of files satisfying time
range [07/31/2016 11:20:50 GMT, 08/05/2016 11:20:55 GMT]
2016/08/05 11:21:06 GMT : Collecting ADR incident files...

```

```

2016/08/05 11:21:16 GMT : Completed collection of additional
diagnostic information...
2016/08/05 11:21:18 GMT : Completed Local Collection

.------.
|           Collection Summary           |
+-----+-----+-----+-----+
| Host   | Status   | Size   | Time   |
+-----+-----+-----+-----+
| host01 | Completed | 5.5MB  | 28s    |
+-----+-----+-----+-----+

Logs are being collected to:
/u01/app/oracle/tfa/repository/collection_Fri_Aug_05_11_20_46_GMT_2016_node_all
/u01/app/oracle/tfa/repository/collection_Fri_Aug_05_11_20_46_GMT_2016_node_all/host01.tfa_Fri_Aug_05_11_20_46_GMT_2016.zip

tfactl> exit
$

```

- b. Check the collection files generated.

```

$ cd /u01/app/oracle/tfa/repository/
$ ls -l collection*
collection_Fri_Aug_05_11_20_46_GMT_2016_node_all:
total 5624
-rw-r--r-- 1 oracle oinstall 2412 Aug  5 11:21
diagcollect_20160805112046_host01.log
-rw-r--r-- 1 oracle oinstall 742 Aug  5 11:21
diagcollect_console_20160805112046_host01.log
-rw-r--r-- 1 oracle oinstall 5733698 Aug  5 11:21
host01.tfa_Fri_Aug_05_11_20_46_GMT_2016.zip
-rw-r--r-- 1 oracle oinstall 685 Aug  5 11:21
host01.tfa_Fri_Aug_05_11_20_46_GMT_2016.zip.txt
$

```

*Q1/ What does the first log file show?*

```

2016/08/05 11:20:50 GMT : Collection Name :
tfa_Fri_Aug_05_11_20_46_GMT_2016.zip
2016/08/05 11:20:50 GMT : Collecting diagnostics from hosts :
[host01]
2016/08/05 11:20:50 GMT : Scanning of files for Collection in
progress...
2016/08/05 11:20:50 GMT : Collecting additional diagnostic
information...
2016/08/05 11:20:55 GMT : Getting list of files satisfying time
range [07/31/201

```

```

6 11:20:50 GMT, 08/05/2016 11:20:55 GMT]
2016/08/05 11:21:06 GMT : Collecting ADR incident files...
2016/08/05 11:21:16 GMT : Completed collection of additional
diagnostic informat
ion...
2016/08/05 11:21:18 GMT : Completed Local Collection
2016/08/05 11:21:18 GMT : Summary!host01!5.5MB!28s!Completed
2016/08/05 11:21:18 GMT : Completed collection of zip files.

```

**A1/ The log file provides information about the types and number of diagnostics files collected.**

**Q2/ What does the first text file show?**

```

TFA Version : 12.2.0.1.0
Build ID : 12201020160330005427

Collection ID : 20160428102932<your_hostname>
Zip file name : tfa_Thu_Apr_28_10_29_32_GMT_2016.zip
Zip file location :
/u01/app/oracle/tfa/repository/collection_Thu_Apr_28_10
_29_32_GMT_2016_node_all/<your_hostname>.tfa_Thu_Apr_28_10_29_32
_GMT_2016.zip
Zip file creation date : Thu Apr 28 2016 10:29:34 GMT
Host name : <your_hostname>
Duration of Diagnostics :
    Start date : Sat Apr 23 2016 10:29:34 GMT
    End date : Thu Apr 28 2016 10:29:34 GMT
Component(s) in zip file : RDBMS = [orcl]
User initiated
Directory Structure in zip file :
<your_hostname>/
<your_hostname>/diag/rdbms/orcl/ORCL/trace

To check full contents of the zip file, Please run "unzip -l
<zipfile>"

```

**A2/ The text file provides information about the zip file generated containing all diagnostic files created during the last five days about ORCL database.**

c. List the files collected.

```

$ unzip -l
/u01/app/oracle/tfa/repository/collection_Thu_Apr_28_10_29_32_GM
T_2016_node_all/<your_hostname>.tfa_Thu_Apr_28_10_29_32_GMT_2016
.zip
Archive:  host01.tfa_Fri_Aug_05_11_20_46_GMT_2016.zip

```

Length	Date	Time	Name
-----	-----	-----	----
3559	08-04-2016	08:36	host01/diag/rdbms/orcl/ORCL/trace/ORCL_aqpc_11715.trc
1349	08-05-2016	08:01	host01/diag/rdbms/orcl/ORCL/trace/ORCL_aqpc_14151.trc
...			
777865	08-05-2016	11:21	host01.zip_inventory.xml
4593054	08-04-2016	13:08	host01/diag/rdbms/orcl/ORCL/incident/incdir_69722/ORCL_dbrm_28263_i69722.trc
4003907	08-04-2016	22:19	host01/diag/rdbms/orcl/ORCL/incident/incdir_72122/ORCL_dbrm_23113_i72122.trc
3611954	08-05-2016	08:21	host01/diag/rdbms/orcl/ORCL/incident/incdir_88922/ORCL_dbrm_17200_i88922.trc
685	08-05-2016	11:21	host01.tfa_Fri_Aug_05_11_20_46_GMT_2016.zip.txt
26	08-05-2016	11:21	TFA.txt
693	08-05-2016	11:21	host01/host01_summary
24559	08-05-2016	11:21	host01/host01_OPATCH_DBHOMES
2295	08-05-2016	11:21	host01/host01_collection.log
-----			
29815760			923 files
\$			

5. A user informs you about an ORA-00600 that raised in `pdb_orcl` database. Run a collection under `root`.
- a. The user was connected in a terminal window (*User window*) under SQL\*Plus and executed the `/home/oracle/labs/TFA/myquery.sql` SQL script.

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

```
$ sqlplus oe@pdb_orcl
Enter password: *****
Connected.
SQL> @$HOME/labs/TFA/myquery
...
SQL> SELECT /*+ monitor*/
2          SUM(lo_extendedprice * lo_discount) revenue
3 FROM    oe.lineorder l, oe.date_dim d
4 WHERE   l.lo_orderdate = d.d_datekey;
FROM    oe.lineorder l, oe.date_dim d
        *
```

```

ERROR at line 3:
ORA-00600: internal error code, arguments: [kds_verif_alloc:
unsupported verif
mode], [], [], [], [], [], [], [], [], [], [], []

SQL>

```

- a. Switch to the *Root window* and collect diagnostics files related to the incident in `pdb_orcl`. Because it is an ORA-600 error, you will have to send the diagnostic files to My Oracle Support.

1) First retrieve the incident number.

```

tfactl> ips show incidents
...

ADR Home = /u01/app/oracle/diag/rdbms/orcl/ORCL:
*****

INCIDENT_ID          PROBLEM_KEY
CREATE_TIME
-----
-----

46706                ORA 600 [kds_verif_alloc: unsupported verif
mode]                2016-05-08 23:46:57.170000 +00:00

tfactl>

```

2) Collect the diagnostic files for the incident and zip them.

```

tfactl> diagcollect -ips -incident 46706 -manageips

Collecting data for incident 46706 ...
Collecting data for all nodes
Creating ips package in master node ...
Trying to use ADR homopath diag/rdbms/orcl/ORCL ...
Submitting request to generate package for ADR homopath
/u01/app/oracle/diag/rdbms/orcl/ORCL
Master package completed for ADR homopath
/u01/app/oracle/diag/rdbms/orcl/ORCL
Created package 1 based on incident id 46706, correlation level
basic

TFA IPS collection is now paused for package manipulation.
Once completed please run, tfactl diagcollect -resumeips
20160508234906ipscoll_<your_servername>
#

```

**Q1/** What does the last message tell?

**A1/** The last message that you should execute another operation to complete the IPS zip file generation.



```
# /u01/app/oracle/tfa/bin/tfactl

tfactl> diagcollect -resumeips
20160508235601lipscoll_<your_servername>
Creating ips package in master node ...
Trying ADR basepath /u01/app/oracle
Trying to use ADR homopath diag/rdbms/orcl/ORCL ...
Submitting request to generate package for ADR homopath
/u01/app/oracle/diag/rdbms/orcl/ORCL
Master package completed for ADR homopath
/u01/app/oracle/diag/rdbms/orcl/ORCL

Collection Id : 20160508235601<your_servername>

Detailed Logging at :
/u01/app/oracle/tfa/repository/collection_Tue_Oct_10_13_33_01_GMT_2016_node_all/diagcollect_20160508235601_<your_servername>.log
2016/10/13 33:33:31 GMT : Collection Name :
tfa_Tue_Oct_10_13_33_01_GMT_2016.zip
2016/10/13 33:33:31 GMT : Collecting diagnostics from hosts :
[<your_servername>]
2016/10/13 33:33:31 GMT : Collecting additional diagnostic
information...
2016/10/13 33:33:31 GMT : Scanning of files for Collection in
progress...
2016/10/13 33:33:36 GMT : Getting list of files satisfying time
range [05/08/2016 11:57:31 GMT, 05/08/2016 23:57:36 GMT]
2016/10/13 33:33:38 GMT : Collecting ADR incident files...
2016/10/13 33:33:44 GMT : Completed collection of additional
diagnostic information...
2016/10/13 33:33:48 GMT : Completed Local Collection

.------.
|           Collection Summary           |
+-----+-----+-----+-----+
| Host           | Status      | Size      | Time      |
+-----+-----+-----+-----+
| <your_servername> | Completed  | 1.1MB    | 17s      |
+-----+-----+-----+-----+

Logs are being collected to:
/u01/app/oracle/tfa/repository/collection_Tue_Oct_13_06_33_01_GMT_2016_node_all
```

```
/u01/app/oracle/tfa/repository/collection_Tue_Oct_13_06_33_01_GMT_2016_node_all/<your_servername>.tfa_Tue_Oct_13_06_33_01_GMT_2016.zip
```

```
tfactl> exit
```

```
#
```

**Q2/ Switch to the Oracle window. Can the *oracle* user analyze the diagnostic files collected by *root*?**

```
$ ls -l /u01/app/oracle/tfa/repository/collection*
```

```
ls: cannot open directory
```

```
/u01/app/oracle/tfa/repository/collection_2016_09_19T09_06_06_node_<your_server>: Permission denied
```

```
/u01/app/oracle/tfa/repository/collection_Thu_Oct_06_33_25_GMT_2016_node_all
```

```
Total 132
```

```
-rw-r--r-- 1 oracle oinstall      1890 Oct 13 06:33
```

```
diagcollect_20161013332501_<your_servername>.log
```

```
-rw-r--r-- 1 oracle oinstall        742 Oct 13 06:33
```

```
diagcollect_console_20161013063325_<your_servername>.log
```

```
-rw-r--r-- 1 oracle oinstall    118653 Oct 13 06:33
```

```
<your_servername>.tfa_Thu_Oct_13_06_33_25_GMT_2016.zip
```

```
-rw-r--r-- 1 oracle oinstall        685 Oct 13 06:33
```

```
<your_servername>.tfa_Thu_Oct_13_06_33_25_GMT_2016.zip.txt
```

```
ls: cannot open directory
```

```
/u01/app/oracle/tfa/repository/collection_Tue_Oct_13_06_41_01_GMT_2016_node_all: Permission denied
```

```
$
```

**A2/ It is consistent that the *oracle* user cannot read the collections generated by *root*. However, the *root* user can read the collections generated by *oracle*.**

**Q3/ What does the log file report?**

```
# cd /u01/app/oracle/tfa/repository/collection*
```

```
# ls
```

```
diagcollect_20161013332501_<your_servername>.log
```

```
diagcollect_console_20161013332501_<your_servername>.log
```

```
<your_servername>.tfa_Thu_Oct_13_06_33_25_GMT_2016.zip
```

```
<your_servername>.tfa_Thu_Oct_13_06_33_25_GMT_2016.zip.txt
```

```
# more diagcollect_20161013332501_<your_servername>.log
```

```
2016/10/13 33:33:31 GMT : Collection Name :
```

```
tfa_Tue_Oct_13_3_25_01_GMT_2016.zip
```

```
2016/10/13 33:33:31 GMT : Collecting diagnostics from hosts :
```

```
[<your_servername>]
```

```
2016/10/13 33:33:31 GMT : Collecting additional diagnostic information...
```

```
2016/10/13 33:33:31 GMT : Scanning of files for Collection in progress...
```

```

2016/10/13 33:33:36 GMT : Getting list of files satisfying time
range [05/08/2016 11:57:31 GMT, 05/08/2016 23:57:36 GMT]
2016/10/13 33:33:37 GMT : Starting Thread to identify stored
files to collect
2016/10/13 33:33:37 GMT : Getting List of Files to Collect
2016/10/13 33:33:37 GMT : Finished Getting List of Files to
Collect
2016/10/13 33:33:38 GMT : Collecting ADR incident files...
2016/10/13 33:33:38 GMT : Waiting for collection of additional
diagnostic information
2016/10/13 33:33:44 GMT : Completed collection of additional
diagnostic information...
2016/10/13 33:33:48 GMT : Completed Zipping of all files
2016/10/13 33:33:48 GMT : Cleaning up temporary files
2016/10/13 33:33:48 GMT : Finished Cleaning up temporary files
2016/10/13 33:33:48 GMT : Finalizing the Collection Zip File
2016/10/13 33:33:48 GMT : Finished Finalizing the Collection Zip
File
2016/10/13 33:33:48 GMT : Total Number of Files checked : 3205
2016/10/13 33:33:48 GMT : Total Size of all Files Checked : 12GB
2016/10/13 33:33:48 GMT : Number of files containing required
range : 0
2016/10/13 33:33:48 GMT : Total Size of Files containing
required range : 0kB
2016/10/13 33:33:48 GMT : Number of files trimmed : 0
2016/10/13 33:33:48 GMT : Total Size of data prior to zip :
8.1MB
2016/10/13 33:33:48 GMT : Saved 0kB by trimming files
2016/10/13 33:33:48 GMT : Zip file size : 1.1MB
2016/10/13 33:33:48 GMT : Total time taken : 17s
2016/10/13 33:33:48 GMT : Completed Local Collection
2016/10/13 33:33:48 GMT :
Summary!<your_servername>!1.1MB!17s!Completed
2016/10/13 33:33:48 GMT : Completed collection of zip files.
[root@<your_servername>
#

```

**A3/ The log file reports how IPS scanned diagnostic files related to the reported incident to generate the zip file.**

**Q4/ What does the text file report?**

```

# more
<your_servername>.tfa_Thu_Oct_13_06_33_25_GMT_2016.zip.txt
TFA Version : 12.2.0.1.0
Build ID : 12201020160421014806

```

```

Collection ID : 20160508235601<your_servername>
Zip file name : tfa_Thu_Oct_13_06_33_25_GMT_2016.zip
Zip file location :
/u01/app/oracle/tfa/repository/collection_Tue_Oct_13_23_56_01_GM
T_2016_node_all/<your_servername>.tfa_Tue_Oct_13_23_56_2016.zip
Zip file creation date : Tue Oct 13 2016 33:57:31 GMT
Host name : <your_servername>
Duration of Diagnostics :
  Start date : Tue Oct 13 2016 33:57:31 GMT
  End date : Tue Oct 13 2016 34:57:31 GMT
Component(s) in zip file : IPS
User initiated

<your_servername>//u01/app/oracle/tfa/repository/suptools/ips/us
er_root/20160508235601lipscoll_<your_servername>

Directory Structure in zip file :
<your_servername>/

To check full contents of the zip file, Please run "unzip -l
<zipfile>"
#

```

**A4/ The text file reports who and when the zip file was generated, and how to unzip the zip file.**

**Q5/ What does the zip file contain?**

```

# unzip -l <your_servername>.tfa_Oct_13_06_33_25_GMT_2016.zip
Archive:  <your_servername>.tfa_Tue_Oct_13_33_25_01_GMT_2016.zip
  Length      Date    Time    Name
  -----
  10679  13-10-2016  23:57
<your_servername>/diag/rdbms/orcl/ORCL/manifest.xml
    239  13-10-2016  23:57
<your_servername>/diag/rdbms/orcl/ORCL/trace/ORCL_ora_2745.trm
  312151  13-10-2016  23:57
<your_servername>/diag/rdbms/orcl/ORCL/trace/alert_ORCL.log
    1242  13-10-2016  23:57
<your_servername>/diag/rdbms/orcl/ORCL/trace/ORCL_ora_2745.trc
  1337142  13-10-2016  23:57
<your_servername>/diag/rdbms/orcl/ORCL/alert/log.xml
    711  13-10-2016  23:57
<your_servername>/diag/rdbms/orcl/ORCL/hm/HMREPORT_HM_RUN_141.hm
    1326  13-10-2016  23:57
<your_servername>/diag/rdbms/orcl/ORCL/log/debug/log.xml

```

```

5909273 13-10-2016 23:57
<your_servername>/diag/rdbms/orcl/ORCL/incident/incdir_46706/ORCL_ora_2745_i46706.trc
1759 13-10-2016 23:57
<your_servername>/diag/rdbms/orcl/ORCL/incident/incdir_46706/ORCL_m000_2750_i46706_a.trc
219368 13-10-2016 23:57
<your_servername>/diag/rdbms/orcl/ORCL/incident/incdir_46706/ORCL_ora_2745_i46706.trm
96503 13-10-2016 23:57
<your_servername>/diag/rdbms/orcl/ORCL/incident/incdir_46706/incident_sig.xml
228 13-10-2016 23:57
<your_servername>/diag/rdbms/orcl/ORCL/incident/incdir_46706/ORCL_m000_2750_i46706_a.trm
476 13-10-2016 23:57
<your_servername>/diag/rdbms/orcl/ORCL/incpkg/pkg_3/seq_1/export/IPS_PACKAGE.dmp
193 13-10-2016 23:57
<your_servername>/diag/rdbms/orcl/ORCL/incpkg/pkg_3/seq_1/export/IPS_PACKAGE_INCIDENT.dmp
384 13-10-2016 23:57
<your_servername>/diag/rdbms/orcl/ORCL/incpkg/pkg_3/seq_1/export/INCIDENT_FILE.dmp
613 13-10-2016 23:57
<your_servername>/diag/rdbms/orcl/ORCL/incpkg/pkg_3/seq_1/export/HM_RUN.dmp
280 13-10-2016 23:57
<your_servername>/diag/rdbms/orcl/ORCL/incpkg/pkg_3/seq_1/export/IPS_PACKAGE_HISTORY.dmp
251 13-10-2016 23:57
<your_servername>/diag/rdbms/orcl/ORCL/incpkg/pkg_3/seq_1/export/DDE_USER_ACTION_PARAMETER.dmp
708 13-10-2016 23:57
<your_servername>/diag/rdbms/orcl/ORCL/incpkg/pkg_3/seq_1/export/DDE_USER_ACTION_PARAMETER_DEF.dmp
244 13-10-2016 23:57
<your_servername>/diag/rdbms/orcl/ORCL/incpkg/pkg_3/seq_1/export/DDE_USER_ACTION.dmp
2998 13-10-2016 23:57
<your_servername>/diag/rdbms/orcl/ORCL/incpkg/pkg_3/seq_1/export/IPS_FILE_METADATA.dmp
452 13-10-2016 23:57
<your_servername>/diag/rdbms/orcl/ORCL/incpkg/pkg_3/seq_1/export/PROBLEM.dmp
908 13-10-2016 23:57
<your_servername>/diag/rdbms/orcl/ORCL/incpkg/pkg_3/seq_1/export/DDE_USER_ACTION_DEF.dmp

```

```

353 13-10-2016 23:57
<your_servername>/diag/rdbms/orcl/ORCL/incpkg/pkg_3/seq_1/export
/DDE_USER_INCIDENT_TYPE.dmp
498 13-10-2016 23:57
<your_servername>/diag/rdbms/orcl/ORCL/incpkg/pkg_3/seq_1/export
/INCCKEY.dmp
214 13-10-2016 23:57
<your_servername>/diag/rdbms/orcl/ORCL/incpkg/pkg_3/seq_1/export
/IPS_FILE_COPY_LOG.dmp
166 13-10-2016 23:57
<your_servername>/diag/rdbms/orcl/ORCL/incpkg/pkg_3/seq_1/export
/DDE_USER_INCIDENT_ACTION_MAP.dmp
749 13-10-2016 23:57
<your_servername>/diag/rdbms/orcl/ORCL/incpkg/pkg_3/seq_1/export
/INCIDENT.dmp
1158 13-10-2016 23:57
<your_servername>/diag/rdbms/orcl/ORCL/incpkg/pkg_3/seq_1/export
/IPS_PACKAGE_FILE.dmp
207 13-10-2016 23:57
<your_servername>/diag/rdbms/orcl/ORCL/incpkg/pkg_3/seq_1/export
/EM_USER_ACTIVITY.dmp
2930 13-10-2016 23:57
<your_servername>/diag/rdbms/orcl/ORCL/incpkg/pkg_3/seq_1/export
/IPS_CONFIGURATION.dmp
96503 13-10-2016 23:57
<your_servername>/diag/rdbms/orcl/ORCL/incpkg/pkg_3/seq_1/config
.xml
406 13-10-2016 23:56
<your_servername>/diag/rdbms/orcl/ORCL/keyfile.xml
555 13-10-2016 23:56
<your_servername>/u01/app/oracle/tfa/repository/suptools/ips/use
r_root/20160508235601ipscoll_<your_servername>/state.log
101 13-10-2016 23:57
<your_servername>.zip_inventory.xml
750 13-10-2016 23:57
<your_servername>.tfa_Sun_May_08_23_56_01_GMT_2016.zip.txt
26 13-10-2016 23:57 TFA.txt
23416 13-10-2016 23:57
<your_servername>/<your_servername>_OPATCH_DBHOMES
2801 13-10-2016 23:57
<your_servername>/<your_servername>_collection.log
3168 13-10-2016 23:57
<your_servername>/<your_servername>_LSMOD
63409 13-10-2016 23:57
<your_servername>/<your_servername>_RPMQA
54144 13-10-2016 23:57
<your_servername>/<your_servername>_PROC DIRINFO

```

```
      882  13-10-2016 23:57
<your_servername>/<your_servername>_oratab
      1211  13-10-2016 23:57
<your_servername>/<your_servername>_VARTMPORACLE
-----
      8151775                                44 files
#
```

**A5/ The zip file contains what IPS traditionally collects for a reported issue.**

3) Send the zipped collection to *My Oracle Support*.





# **Practices for Lesson 22: Oracle Database Public Cloud Services**

## **Chapter 22**

## Practices for Lesson 22: Overview

---

### Practices Overview

In these practices, you will create a database deployment, then check that backups are automatically completed, perform an on-demand backup, and finally monitor the database deployment using DBaaS Monitor.

#### **Important note:**

Currently we do not have hands-on lab for this lesson. Currently the four practices can only be demonstrated if the instructor has an available Oracle Public Cloud DBCS account or viewed using this recorded video.

([http://education.oracle.com/pls/web\\_prod-plq-dad/db\\_pages.getpage?page\\_id=904&get\\_params=cloudId:384,objectId:15739](http://education.oracle.com/pls/web_prod-plq-dad/db_pages.getpage?page_id=904&get_params=cloudId:384,objectId:15739))

The video has voice over explanation for these practices. It is best viewed at your convenience with audio enabled.

## Practice 22-1: Creating a Database Deployment

---

### Overview

In this practice, you will create a database deployment, considering the following attributes:

- The database deployment must be created with an Oracle Database 12c environment installed with no particular option, named `<student_initials>DBCS` with a pre-created database named `MYORCL` and a PDB named `MYPDB1`, with the same administrator password for the `SYS` and `SYSTEM` database users (The password is provided by your instructor). For the purpose of ease, the database deployment in all practices will be referred to as `MYDBCS`.
- The service level required is "Oracle Database Cloud Service" and not "Oracle Database Cloud Service - Virtual Image" because you want the database instance to be created for you.
- You choose a monthly billing frequency.
- You need **1 CPU** and 7.5 GB RAM, 20 GB for the database.
- You want your automated database backups be stored on an Oracle Storage Cloud container and on the local compute node associated to your database deployment.

### Assumptions

In the training environment, you have an Oracle Cloud account assigned. All information related to your Oracle Cloud account is provided by the instructor.

Record those items here:

Domain ID: \_\_\_\_\_

User name: \_\_\_\_\_

Password: \_\_\_\_\_

At this point, there is no deployment database.

### Tasks

1. Sign in to Oracle Cloud using a web address (URL) provided by your Oracle Cloud account administrator (the instructor). First enter the identity domain, then the user name and password.  
For details on accessing the console, see the documentation on *Accessing the Database Cloud Service Console* (<http://www.oracle.com/pls/topic/lookup?ctx=cloud&id=CSDBI3331>).
2. Among the services ('list' in old format - 'tile' in new format), click the "Oracle Database Cloud Service" or "Database".
  - a. Click the Open Service console.
  - b. Click Create Service to start the database deployment process.
  - c. Then fill in the fields in each page of the database deployment creation until you confirm its creation. Refer to the instructions listed in the overview of the practice and review the summary in the last step to ensure you created the appropriate database deployment.

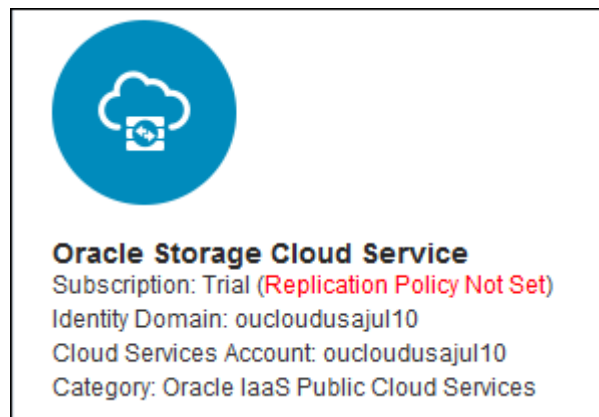
If you need details on each step of the creation process, follow *Creating a Database Deployment* (<http://www.oracle.com/pls/topic/lookup?ctx=cloud&id=CSDBI3299>).

**Note:** If you want an environment installed with the In-Memory Database option, clicking Details next to Enterprise Edition in one of the steps displays the Pricing tab to get the information about which Service you will need.

- 1) **If you used** the wizard to create a public/private key pair for you, **you may download the keys to your local client. Thus, a zip file named `sshkeybundle.zip` containing the `publicKey` and `privateKey` files in `open_ssh` format is generated, both files respectively containing the public key and private key. For security reasons, move the zip file to `~/.ssh`, and then unzip. Change the permissions on the `privateKey` file to owner access only with `chmod 600 privateKey`.**

**Note:** If you prefer to create the key pair manually, create the secure shell (SSH) private/public key pair before step 2.b, with copies of the private and public key files on your local computer. You will point to the SSH public key when you create the database deployment. For details on generating an SSH private/public key pair, refer to the *Generating a Secure Shell (SSH) Private/Public Key Pair* section of the *Using Oracle Database Cloud Service (Database as a Service)* documentation.

- 2) Check first in the Storage Cloud Service detail page. You may see the red message: Replication Policy Not Set. This message appears when your domain spans multiple sites (Check in the Compute Service console). For the details of replication see *Selecting a Replication Policy for Oracle Storage Cloud Service* (<http://www.oracle.com/pls/topic/lookup?ctx=cloud&id=CSSTO-GUID-5D53C11F-3D9E-43E4-8D1D-DDBB95DEC715>)



If this is the case, set the replication. If this is not the case, go to the next step. For this practice, Replication may be set to None.

### Set Replication Policy

Select the data center (DC) and georeplication policy for your service instance.

- \* ☐ Primary DC: Chicago (us2); Georeplication DC: None
- ☐ Primary DC: Ashburn (us6); Georeplication DC: None
- ☐ Primary DC: Ashburn (us6); Georeplication DC: Chicago (us2)
- ☒ Primary DC: Chicago (us2); Georeplication DC: Ashburn (us6)

**Caution:** Once set, the replication policy cannot be changed for the service instance.

Set
Cancel

- 3) Create the Cloud storage container by selecting the "Create Cloud Storage Container" check box provided during the database deployment process in the "Backup and Recovery Configuration " section. Provide the following information:
  - a) The storage container name structured as  
`https|http://<datacenter_code>/<version>/<schema name>/<container name>.`  
 To find the `<datacenter_code>/<version>/<schema name>`, go to the Oracle Database Backup Cloud service and retrieve the REST endpoint field value. Append to this value the name of the container you want to create for your backups.  
**Warning:** Ensure you provide the right `datacenter_code`. If not, the database deployment gets created but with no storage container available for your backups. Any further backup will fail.  
 Example: If the REST end point is `https://em2.storage.oraclecloud.com` and if you want to create JLScontainer in the identity domain oucloudusajul10, enter the following value:  
`https://em2.storage.oraclecloud.com/v1/Storage-oucloudusajul10/JLScontainer`
  - b) The name of the owner of the container: the user name of your Cloud account
  - c) The password of the owner of the container: the password of your Cloud account
3. Before confirming the database deployment, you should get a summary displaying all the characteristics that you defined.
4. The database deployment is in progress and then later completed.

## Practice 22-2: Connecting to the Database Deployment Compute Node

### Overview

In this practice, you will configure the connection for `opc` user and another one for `oracle` user.

### Assumption

As you are working in an Oracle Classroom, certain modifications must be made.

Edit the file `~/.ssh/config` to reflect the IP address of compute node:

```
Host <IP address of compute node>
ProxyCommand nc -X connect -x ges-proxy.us.oracle.com:80 %h %p
GSSAPIAuthentication no
```

### Tasks

1. Retrieve the IP Address of the compute node assigned to your database deployment. Read *Viewing Detailed Information for a Database Deployment* (<http://www.oracle.com/pls/topic/lookup?ctx=cloud&id=CSDBI3302>).

If you signed out, sign in to Oracle Cloud using your Cloud user account. First enter the identity domain, then the user name and password.

You access the My Services - Dashboard console.

2. Among the list of services, click the Oracle Database Cloud Service.
3. Click the Open Service Console.
4. Click the name of your database deployment. The public IP Address is displayed.

If you are using Windows to connect to your compute node, use PuTTY. First use PuTTY Key Generator to load the system generated private key file and save the file in PuTTY's own format. Then use PuTTY to create a new session where the host name is the public IP Address of the compute node, the Connection Data uses `opc` as the Auto-Login username, and the Connection SSH Auth uses the private key file converted in the PuTTY format.

If you are using a Linux client to connect to your compute node, use `ssh`. Read *Connecting to a Compute Node Using the ssh Utility on UNIX and UNIX-Like Platforms* (<http://www.oracle.com/pls/topic/lookup?ctx=cloud&id=CSDBI3450>).

This operation opens the session to the compute node. You are now connected to the compute node as `opc`.

```
$ ssh -i your_private_key_file opc@your_compute_node_IP_Address
[opc@MYDBCS ~]$
```

**Question:** Did the `opc` user provide any password to log in the compute node?

**Answer:** No. The `opc` user authentication is completed with the SSH private/public keys, the SSH private key file that pairs with the public key used during the database deployment creation process.

5. Perform the same operation to create a new session as `oracle` to the compute node. You are now connected to the compute node as `oracle`.

```
[opc@MYDBCS ~]$ exit
logout
Connection to your_compute_node_IP_Address closed.
$ ssh -i your_private_key_file oracle@your_compute_node_IP_Address
[oracle@MYDBCS ~]$ exit
$
```

## Practice 22-3: Backing Up the Database Deployment

### Overview

In this practice, you will back up your database deployment manually.

*Q/ Is the database of your database deployment automatically backed up?*

**A/ Yes. During the database deployment, you selected the backup destination to “Both Cloud Storage and Local Storage”.**

- **None:** No backup configuration is created.
- **Local Storage Only:** Backups are configured to be created automatically and stored only on local storage on the compute nodes associated with the deployment. (Not available with the UI Cloud interface, but only with the `oracle-dbcs-cli` utility)
- **Cloud Storage Only:** Backups are configured to be created automatically and stored on an Oracle Storage Cloud Service container. The container must have been created before creating the Database as a Service database deployment.
- **Both Cloud Storage and Local Storage:** Backups are configured to be created automatically and stored both on local compute node storage and on an Oracle Storage Cloud Service container. The container must have been created before creating the Database as a Service database deployment.

### Tasks

1. Log in to the database deployment compute node as `opc` and switch as `root`.

```
[opc@MYDBCS ~]$ sudo -s
[root@MYDBCS ~]#
```

2. Display the content of the database files configuration file (`/home/oracle/bkup/<DBNAME>/dbcfg.spec`) and the system files configuration file (`/home/oracle/bkup/<DBNAME>/oscfg.spec`) to know which files are backed up during the automated database backup.

*Q/ Did you find the system files configuration file (`/home/oracle/bkup/<DBNAME>/oscfg.spec`)?*

**A/ Yes.**

3. Execute a manual backup with the `bkup_api` utility. Read *Creating an On-Demand Backup by Using the Oracle Database Cloud Service Console* (<http://www.oracle.com/pls/topic/lookup?ctx=cloud&id=CSDBI-GUID-019BCAF6-63D6-409E-85F2-FA5098D4076D>).
  - a. Disable the scheduled backup configuration (`/etc/crontab`). Edit the `crontab` file to set the `obkup` command in comment.

```
# 35 0 * * * root /var/opt/oracle/bkup_api/bkup_api bkup_start -
-dbname=MYORCL
# 0,30 * * * * root /home/oracle/bkup/MYORCL/obkup -
dbname=MYORCL -archivelog
```



- b. Backup the database deployment, including the database and the files defined in the configuration file described in step 2.

```
[root@MYDBCS MYORCL]# /var/opt/oracle/bkup_api/bkup_api bkup_start
DBaaS Backup API V1.5 @2015 Multi-Oracle home
-> Action : bkup_start
-> logfile: /var/opt/oracle/bkup_api/log/bkup_api.log
** process started with PID: 21230
** see log file for monitor progress
-----
[root@MYDBCS MYORCL]#
```

- c. Follow up the backup operation.

1) Imagine that the status displayed the following information.


```
[root@MYDBCS MYORCL]# /var/opt/oracle/bkup_api/bkup_api bkup_status
DBaaS Backup API V1.5 @2015 Multi-Oracle home
-> Action : bkup_status
-> logfile: /var/opt/oracle/bkup_api/log/bkup_api.log
Warning: unable to get current configuration of: catalog
* Current backup settings:
* Last registered Bkup: 07-25 16:00 API::31085:: Starting dbaas backup
process
* Bkup state: finished
*****
* API History: API steps
  API:: NEW PROCESS 31085
  API:: Starting dbaas backup process
* DBaaS Error trace:
  -> API::ERROR Cannot complete the backup to disk storage
  -> API::ERROR Please check the log to get more error information
  -> API::ERROR UNKNOWN ERROR
  -> API::ERROR -Detail
  -> API::ERROR OBKUP:: ..... FAIL
  -> API::ERROR RMAN-00569: === ERROR MESSAGE STACK FOLLOWS ==
  -> API::ERROR RMAN-03002: failure of recover command at 07/25/2016
16:00:36
  -> Oracle Error Codes found:
      ORA-01858: a non-numeric character was found where a numeric
was expected
*
* RETURN CODE:1
#####
[root@MYDBCS MYORCL]#
```

**Q1/ What would the error message mean?**

**A1/ “Cannot complete the backup to disk storage” means that there is not enough disk space to store the backupsets and other files created for the backup.**

Q2/ Which operation would you perform to circumvent the issue?

**A2/ Scale up the storage capacity of the pre-created Oracle database instance MYORCL. The CDB should get 10 GB more to store application data.**

- 1) You would open the Oracle Database Cloud Service console.
- 2) You would click the MYDBCS link.
- 3) From the  menu icon, You would select Scale Up/Down.

**You would see the following storage scale up options:**

**Create New Storage Volume:** adds a new storage volume to the database deployment and mounts it as the next available `/u0n` mount point.

**Extend Data Storage Volume:** adds the storage volume to the existing Linux LVM disk group (or Oracle ASM disk group on deployments that use Oracle Real Application Clusters) for database data storage.

**Extend Backup Storage Volume:** adds the storage volume to the existing Linux LVM disk group (or Oracle ASM disk group on deployments that use Oracle Real Application Clusters) for backup and FRA storage.

Read *Scaling Up the Storage for a Database Deployment*  
(<http://www.oracle.com/pls/topic/lookup?ctx=cloud&id=CSDBI3339>).


Q3/ Which storage scale up option would you choose to increase the application storage volume for the database?

**A3/ Extend Data Storage Volume would be the appropriate solution.**

When the scaling operation would begin, the deployment would be in Maintenance status and would be unavailable while the scaling operation would be in progress. The deployment would be shutdown and rebooted during this operation.

```
[root@MYDBCS MYORCL]#
Broadcast message from root@MYDBCS.compute-
ous1sdbatest01.oraclecloud.internal
      (unknown) at 16:14 ...

The system is going down for power off NOW!
```

- 4) You would switch to the Oracle Compute Cloud Service console to view details about the storage extension of your database deployment.
- 5) You would click the Open Service console.
- 6) You would click View from the  menu icon for your database deployment. The storage volume added would be 10 GB.

- 7) Then you would re-execute the manual backup with the `bkup_api` utility.

```
[root@MYDBCS ~]# /var/opt/oracle/bkup_api/bkup_api bkup_start
DBaaS Backup API V1.5 @2015 Multi-Oracle home
-> Action : bkup_start
-> logfile: /var/opt/oracle/bkup_api/log/bkup_api.log
** process started with PID: 28634
** see log file for monitor progress
-----
[root@MYDBCS MYORCL]#
```

- 8) Imagine that the status displayed the following information:

```
[root@MYDBCS MYORCL]# /var/opt/oracle/bkup_api/bkup_api bkup_status
DBaaS Backup API V1.5 @2015 Multi-Oracle home
-> Action : bkup_status
-> logfile: /var/opt/oracle/bkup_api/log/bkup_api.log
Warning: unable to get current configuration of: catalog
* Current backup settings:
* Last registered Bkup: 07-26 07:54 API::28634:: Starting dbaas backup
process
* Bkup state: finished
*****
* API History: API steps
  API:: NEW PROCESS 28634
  API:: Starting dbaas backup process
* DBaaS Error trace:
  -> API::ERROR Cannot complete the backup to disk storage
  -> API::ERROR FRA DISK SPACE ERROR
  -> API::ERROR FRA space error. No enough disk space left
  -> API::ERROR Insufficient disk space at FRA, as configured in
db_recovery_file_dest_size.
  -> API::ERROR Please increase FRA size or delete unwanted content.
  -> API::ERROR -Detail
  -> API::ERROR OBKUP:: ..... FAIL
  -> API::ERROR RMAN-00569: ==== ERROR MESSAGE STACK FOLLOWS ====
  -> API::ERROR RMAN-03002: failure of backup plus archivelog command
at 07/26/2016 07:54:57
  -> API::ERROR RMAN-03002: failure of recover command at 07/26/2016
07:54:49
  -> Oracle Error Codes found:
      ORA-19804: cannot reclaim 217291264 bytes disk space from
6442450944 limit
      ORA-19809: limit exceeded for recovery files
      ORA-01858: a non-numeric character was found where a numeric
was expected
*
* RETURN CODE:1
#####
[root@MYDBCS MYORCL]#
```

*Q/ What would the new error messages mean?*

***A/ The error messages would mean that even if there is now enough disk space, the MYORCL database does not allow enough space for RMAN to store the backupsets.***

9) You would increase the FRA size. You would open a connection as oracle.

```
[oracle@MYDBCS ~]$ sqlplus / as sysdba

SQL*Plus: Release 12.1.0.2.0 Production on Tue Jul 26 08:52:09 2016

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 Oracle Label Security option

SQL> SHOW PARAMETER db_reco

NAME                                TYPE                                VALUE
-----
db_recovery_file_dest                string
/u03/app/oracle/fast_recovery_area
db_recovery_file_dest_size           big integer 6G
SQL> ALTER SYSTEM SET db_recovery_file_dest_size=12G SCOPE=BOTH;

System altered.

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

Total System Global Area 3221225472 bytes
Fixed Size                  2929552 bytes
Variable Size              1862274160 bytes
Database Buffers           1342177280 bytes
Redo Buffers                13844480 bytes
Database mounted.
Database opened. SQL> EXIT
Disconnected from Oracle Database 12c Enterprise Edition Release
12.1.0.2.0 - 64bit Production
With the Oracle Label Security option
[oracle@MYDBCS ~]$
```

- 10) You would re-execute the manual backup with the `bkup_api` utility from the `opc` session.

```
[root@MYDBCS opc]# /var/opt/oracle/bkup_api/bkup_api bkup_start
DBaaS Backup API V1.5 @2015 Multi-Oracle home
-> Action : bkup_start
-> logfile: /var/opt/oracle/bkup_api/log/bkup_api.log
** process started with PID: 3319
** see log file for monitor progress
-----
[root@MYDBCS opc]#
```

```
[root@MYDBCS opc]# /var/opt/oracle/bkup_api/bkup_api bkup_status
DBaaS Backup API V1.5 @2015 Multi-Oracle home
-> Action : bkup_status
-> logfile: /var/opt/oracle/bkup_api/log/bkup_api.log
Warning: unable to get current configuration of: catalog
* Current backup settings:
* Last registered Bkup: 07-26 09:37 API::3319:: Starting dbaas backup
process
* Bkup state: running
*****
* API History: API steps
  API:: NEW PROCESS 3319
  API:: Starting dbaas backup process
*****
* Backup steps
  -> API:: Oracle database state is up and running
*
* RETURN CODE:0
#####
[root@MYDBCS opc]#
[root@MYDBCS opc]#
```

```
[root@MYDBCS opc]# /var/opt/oracle/bkup_api/bkup_api bkup_status
DBaaS Backup API V1.5 @2015 Multi-Oracle home
-> Action : bkup_status
-> logfile: /var/opt/oracle/bkup_api/log/bkup_api.log
Warning: unable to get current configuration of: catalog
* Current backup settings:
* Last registered Bkup: 07-26 09:37 API::3319:: Starting dbaas backup
process
* Bkup state: finished
*****
* API History: API steps
  API:: NEW PROCESS 3319
```

```

API:: Starting dbaas backup process
case API::WARNING : your partition /u03 has 100% used space
*
* RETURN CODE:1
#####
[root@MYDBCS opc]#

```

*Q/ What would the warning message mean?*

***A/ The partition /u03 dedicated for the FRA would be now full. You should either remove existing backups if any, or scale up the storage for the database for the next backup.***

11) To get more details about the root cause of the issue, you would find and read the backup logs. Below is an example.

```

[root@MYDBCS ~]$ cd /home/oracle/bkup/<DBNAME>/log
[root@MYDBCS log]$ ls -ltr
...
lrwxrwxrwx 1 oracle oinstall      57 Sep  6 09:43 obkup.log ->
/home/oracle/bkup/MYORCL/log/obkup2016-09-06_09:43:24.log
-rw-r--r-- 1 oracle oinstall 10629 Sep  6 09:44 obkup2016-09-
06_09:43:24.log
[root@MYDBCS log]$
[root@MYDBCS log]$ more obkup2016-09-06_09:43:24.log
OBKUP:: DB instance: MYORCL
OBKUP:: Determining if the filesystem is not full
Case OK : Your partition /u03 has 33% used space
Case OK : Your partition /u01 has 14% used space
Case OK : Your partition /u02 has 20% used space
OBKUP:: ..... OK
OBKUP:: Performing backup to local storage (primary backup)
OBKUP:: Executing rman instructions
Connection string:
/u01/app/oracle/product/12.1.0/dbhome_1/bin/rman msgno target
/
-> set encryption off;
-> RECOVER COPY OF DATABASE WITH TAG 'dbaas_incr_backup' UNTIL
TIME 'SYSDATE -';
-> BACKUP INCREMENTAL LEVEL 1 FOR RECOVER OF COPY WITH TAG
'dbaas_incr_backup' DATABASE plus archivelog;
-> DELETE FORCE NOPROMPT ARCHIVELOG ALL BACKED UP 1 TIMES TO
DISK COMPLETED BEFORE 'SYSDATE - 1';
-> DELETE FORCE NOPROMPT OBSOLETE;
-> RMAN OUTPUT:

Recovery Manager: Release 12.1.0.2.0 - Production on Tue
Sep 6 09:43:28 2016

```

```

...
    RMAN> set echo off;
    RMAN-03030: echo set off
    RMAN>
    Recovery Manager complete.
OBKUP:: ..... FAIL
OBKUP::ERROR Cannot complete the backup to disk storage
OBKUP::ERROR-Detail
OBKUP::ERROR UNKNOWN ERROR
OBKUP::ERROR Please check the log to get more error information
OBKUP:: Message sent to DB alertlog.
OBKUP:: Message sent to System log.
OBKUP:: Clean MOTD.
OBKUP:: Message sent to MOTD.
[root@MYDBCS2 log]#

```

12) You would read the content of the alert log. Below is an example.

```

...
OBKUP::ERROR Cannot complete the backup to disk storage
Tue Sep 06 09:58:03 2016
Errors in file
/u01/app/oracle/diag/rdbms/myorcl/MYORCL/trace/MYORCL_m000_24847
.trc:
ORA-19815: WARNING: db_recovery_file_dest_size of 6442450944
bytes is 95.04% used, and has 319583744 remaining bytes
available.
*****
You have following choices to free up space from recovery area:
1. Consider changing RMAN RETENTION POLICY. If you are using
Data Guard,
    then consider changing RMAN ARCHIVELOG DELETION POLICY.
2. Back up files to tertiary device such as tape using RMAN
BACKUP RECOVERY AREA command.
3. Add disk space and increase db_recovery_file_dest_size
parameter to reflect the new space.
4. Delete unnecessary files using RMAN DELETE command. If an
operating system command was used to delete files, then use RMAN
CROSSCHECK and DELETE EXPIRED commands.

```

13) You would re-execute the manual backup with the `bkup_api` utility from the `opc` session and check the status.

```

[root@MYDBCS MYORCL]# /var/opt/oracle/bkup_api/bkup_api
bkup_status
DBaaS Backup API V1.5 @2015 Multi-Oracle home
-> Action : bkup_status

```

```

-> logfile: /var/opt/oracle/bkup_api/log/bkup_api.log
Warning: unable to get current configuration of: catalog
* Current backup settings:
* Last registered Bkup: 09-12 14:20 API::21230:: Starting dbaas
backup process
* Bkup state: running
*****
* API History: API steps
  API:: NEW PROCESS 21230
  API:: Starting dbaas backup process
*****
* Backup steps
  -> API:: Oracle database state is up and running
  -> API:: DB instance: MYORCL
  -> API:: Determining if the filesystem is not full
  -> API:: ..... OK
  -> API:: Performing backup to local storage (primary backup)
  -> API:: Executing rman instructions
*
* RETURN CODE:0

```

**Q/ What does the "backup to local storage (primary backup)" mean?**

**A/ The tool proceeds with two backups: a first one on the compute node and a second one on the Cloud Storage container defined during the database deployment.**

```

[root@MYDBCS MYORCL]# /var/opt/oracle/bkup_api/bkup_api
bkup_status
DBaaS Backup API V1.5 @2015 Multi-Oracle home
-> Action : bkup_status
-> logfile: /var/opt/oracle/bkup_api/log/bkup_api.log
Warning: unable to get current configuration of: catalog
* Current backup settings:
* Last registered Bkup: 09-12 14:20 API::21230:: Starting dbaas
backup process
* Bkup state: running
*****
* API History: API steps
  API:: NEW PROCESS 21230
  API:: Starting dbaas backup process
*****
* Backup steps
  -> API:: Oracle database state is up and running

```



```

-> API:: DB instance: MYORCL
-> API:: Determining if the filesystem is not full
-> API:: ..... OK
-> API:: Performing backup to local storage (primary backup)
-> API:: Executing rman instructions
-> API:: ..... OK
-> API:: Backup to local storage is completed
-> API:: Clean MOTD.
-> API:: Performing backup to cloud storage (secondary backup)
-> API:: Executing rman instructions
*
* RETURN CODE:0
#####
[root@DJDBCS MYORCL]#

```

#### 14) Check regularly.

```

[root@MYDBCS opc]# /var/opt/oracle/bkup_api/bkup_api bkup_status
DBaaS Backup API V1.5 @2015 Multi-Oracle home
-> Action : bkup_status
-> logfile: /var/opt/oracle/bkup_api/log/bkup_api.log
Warning: unable to get current configuration of: catalog
* Current backup settings:
* Last registered Bkup: 09-12 14:20 API::21230:: Starting dbaas
backup process
* Bkup state: finished
*****
* API History: API steps
  API:: NEW PROCESS 21230
  API:: Starting dbaas backup process
  API:: Your new dbaas backup tag is TAG20160912T142315
  API:: BKUP COMPLETE YOUR BKUP TAG TAG20160912T142315
*****
* Backup steps
-> API:: Oracle database state is up and running
-> API:: DB instance: MYORCL
-> API:: Determining if the filesystem is not full
-> API:: ..... OK
-> API:: Performing backup to local storage (primary backup)
-> API:: Executing rman instructions
-> API:: ..... OK
-> API:: Backup to local storage is completed
-> API:: Clean MOTD.
-> API:: Performing backup to cloud storage (secondary backup)
-> API:: Executing rman instructions
-> API:: .....OK

```

Copyright © 2016, Oracle and/or its affiliates. All rights reserved.

```

-> API:: Backup to cloud storage is completed
-> API:: Clean MOTD.
-> API:: Validating the backup repository .....
-> API::      All backup pieces are ok
-> API:: Starting backup of config files
-> API:: Executing rman instructions
-> API:: at time: 2016-09-12:14:23:15
-> API:: Determining the oracle database id
-> API::  DBID: 342954598
-> API:: Creating directories to store config files
-> API:: Determining the oracle database id
-> API::  DBID: 342954598
-> API:: Compressing config files into tar files
-> API:: .... OK
-> API:: Uploading config files to cloud storage
-> API:: Completed at time: 2016-09-12:14:23:43
-> API:: at time: 2016-09-12:14:23:43
-> API:: Config files backup ended successfully
-> API:: Clean MOTD.
-> API:: All requested tasks are completed
*
* RETURN CODE:0
#####
[root@MYDBCS opc]# exit
[opc@MYDBCS ~]$

```

4. From the `oracle` session, check that the database files have been backed up using RMAN.

```

[opc@MYDBCS ~]$ exit
logout
Connection to your_compute_node_IP_Address closed.
$ ssh -i mykey oracle@your_compute_node_IP_Address
[oracle@MYDBCS ~]$ rman target /

```

List of Backup Sets

=====

BS Key	Size	Device Type	Elapsed Time	Completion Time
2	2.25M	SBT_TAPE	00:00:04	12-SEP-16
BP Key: 2    Status: AVAILABLE    Compressed: YES    Tag: TAG20160912T140013				
Handle: 03rfkghd_1_1    Media: em2.storage.oraclecloud.com/v1/Storage-dbtestcs4/DJcontainer				

## List of Archived Logs in backup set 2

Thrd	Seq	Low SCN	Low Time	Next SCN	Next Time
1	2	2001175	12-SEP-16	2006332	12-SEP-16

BS Key	Size	Device Type	Elapsed Time	Completion Time
3	14.25M	SBT_TAPE	00:00:07	12-SEP-16

BP Key: 3 Status: AVAILABLE Compressed: YES Tag: TAG20160912T140013

Handle: 02rfkghd\_1\_1 Media:  
em2.storage.oraclecloud.com/v1/Storage-dbtestcs4/DJcontainer

## List of Archived Logs in backup set 3

Thrd	Seq	Low SCN	Low Time	Next SCN	Next Time
1	1	1977427	12-SEP-16	2001175	12-SEP-16

BS Key	Type	LV Size	Device Type	Elapsed Time	Completion Time
4	Full	17.25M	SBT_TAPE	00:00:06	12-SEP-16

BP Key: 4 Status: AVAILABLE Compressed: NO Tag: TAG20160912T140028

Handle: c-342954598-20160912-01 Media:  
em2.storage.oraclecloud.com/v1/Storage-dbtestcs4/DJcontainer

SPFILE Included: Modification time: 12-SEP-16

SPFILE db\_unique\_name: MYORCL

Control File Included: Ckp SCN: 2016018 Ckp time: 12-SEP-16

BS Key	Size
5	100.57M

## List of Archived Logs in backup set 5

Thrd	Seq	Low SCN	Low Time	Next SCN	Next Time
1	1	1977427	12-SEP-16	2001175	12-SEP-16
1	2	2001175	12-SEP-16	2006332	12-SEP-16
1	3	2006332	12-SEP-16	2018436	12-SEP-16

## Backup Set Copy #1 of backup set 5

Device Type	Elapsed Time	Completion Time	Compressed	Tag
DISK	00:00:02	12-SEP-16	NO	

DBAAS\_INCR\_BACKUP

## List of Backup Pieces for backup set 5 Copy #1

BP Key	Pc#	Status	Piece Name
5	1	AVAILABLE	/u03/app/oracle/fast_recovery_area/MYORCL/backupset/2016_09_12/o1_mf_a

-----

5 1 AVAILABLE

/u03/app/oracle/fast\_recovery\_area/MYORCL/backupset/2016\_09\_12/o1\_mf\_a  
nnnn\_DBAAS\_INCR\_BACKUP\_cxfgg5ho\_.bkp

## Backup Set Copy #2 of backup set 5

Device	Type	Elapsed Time	Completion Time	Compressed	Tag
SBT_TAPE		00:00:02	12-SEP-16	NO	

-----

SBT\_TAPE 00:00:02 12-SEP-16 NO  
DBAAS\_INCR\_BACKUP

## List of Backup Pieces for backup set 5 Copy #2

BP Key	Pc#	Status	Media	Piece Name
10	1	AVAILABLE	em2.storage.oraclecloud.com/v1/Storage-	

-----

10 1 AVAILABLE em2.storage.oraclecloud.com/v1/Storage-  
dbtestcs4/DJcontainer 05rfkhi5\_1\_2

BS Key	Size
6	7.00K

-----

6 7.00K

## List of Archived Logs in backup set 6

Thrd	Seq	Low SCN	Low Time	Next SCN	Next Time
1	4	2018436	12-SEP-16	2018492	12-SEP-16

-----

1 4 2018436 12-SEP-16 2018492 12-SEP-16

## Backup Set Copy #1 of backup set 6

Device	Type	Elapsed Time	Completion Time	Compressed	Tag
DISK		00:00:00	12-SEP-16	NO	

-----

DISK 00:00:00 12-SEP-16 NO  
DBAAS\_INCR\_BACKUP

## List of Backup Pieces for backup set 6 Copy #1

BP Key	Pc#	Status	Piece Name
6	1	AVAILABLE	/u03/app/oracle/fast_recovery_area/MYORCL/backupset/2016_09_12/o1_mf_a

-----

6 1 AVAILABLE  
/u03/app/oracle/fast\_recovery\_area/MYORCL/backupset/2016\_09\_12/o1\_mf\_a  
nnnn\_DBAAS\_INCR\_BACKUP\_cxfghw7t\_.bkp

## Backup Set Copy #2 of backup set 6

Device	Type	Elapsed Time	Completion Time	Compressed	Tag
SBT_TAPE		00:00:00	12-SEP-16	NO	

-----

SBT\_TAPE 00:00:00 12-SEP-16 NO  
DBAAS\_INCR\_BACKUP

```

List of Backup Pieces for backup set 6 Copy #2
BP Key  Pc# Status      Media                                Piece Name
-----
  9      1 AVAILABLE    em2.storage.oraclecloud.com/v1/Storage-
dbtestcs4/DJcontainer 0frfkhjs_1_2

BS Key  Type LV Size
-----
  7      Full   17.20M
SPFILE Included: Modification time: 12-SEP-16
SPFILE db_unique_name: MYORCL
Control File Included: Ckp SCN: 2018501      Ckp time: 12-SEP-16

Backup Set Copy #1 of backup set 7
Device Type Elapsed Time Completion Time Compressed Tag
-----
DISK          00:00:02      12-SEP-16          NO
TAG20160912T142149

List of Backup Pieces for backup set 7 Copy #1
BP Key  Pc# Status      Piece Name
-----
  7      1 AVAILABLE
/u03/app/oracle/fast_recovery_area/MYORCL/autobackup/2016_09_12/o1_mf_
s_922371709_cxfghycp_.bkp

Backup Set Copy #2 of backup set 7
Device Type Elapsed Time Completion Time Compressed Tag
-----
SBT_TAPE      00:00:02      12-SEP-16          NO
TAG20160912T142149

List of Backup Pieces for backup set 7 Copy #2
BP Key  Pc# Status      Media                                Piece Name
-----
  8      1 AVAILABLE    em2.storage.oraclecloud.com/v1/Storage-
dbtestcs4/DJcontainer c-342954598-20160912-02

BS Key  Type LV Size      Device Type Elapsed Time Completion Time
-----
  8      Full   17.25M      SBT_TAPE    00:00:02      12-SEP-16
BP Key: 11      Status: AVAILABLE Compressed: NO Tag:
TAG20160912T142222
Handle: c-342954598-20160912-03      Media:
em2.storage.oraclecloud.com/v1/Storage-dbtestcs4/DJcontainer
SPFILE Included: Modification time: 12-SEP-16
SPFILE db_unique_name: MYORCL

```

Control File Included: Ckp SCN: 2018590 Ckp time: 12-SEP-16

BS Key	Type	LV Size	Device Type	Elapsed Time	Completion Time
9	Full	465.00M	SBT_TAPE	00:00:15	12-SEP-16
BP Key: 12 Status: AVAILABLE Compressed: NO Tag:					
TAG20160912T142231					
Handle: 0krfkh17_1_1 Media:					
em2.storage.oraclecloud.com/v1/Storage-dbttestcs4/DJcontainer					
List of Datafiles in backup set 9					
Container ID: 3, PDB Name: MYPDB1					
File	LV	Type	Ckp SCN	Ckp Time	Name
9	Full	2018460	12-SEP-16	/u02/app/oracle/oradata/MYORCL/MYPDB1/sysaux01.dbf	

BS Key	Type	LV Size	Device Type	Elapsed Time	Completion Time
10	Full	460.00M	SBT_TAPE	00:00:16	12-SEP-16
BP Key: 13 Status: AVAILABLE Compressed: NO Tag:					
TAG20160912T142231					
Handle: 0mrfrkh17_1_1 Media:					
em2.storage.oraclecloud.com/v1/Storage-dbttestcs4/DJcontainer					
List of Datafiles in backup set 10					
Container ID: 2, PDB Name: PDB\$SEED					
File	LV	Type	Ckp SCN	Ckp Time	Name
7	Full	2015997	12-SEP-16	/u02/app/oracle/oradata/MYORCL/pdbseed/sysaux01.dbf	

BS Key	Type	LV Size	Device Type	Elapsed Time	Completion Time
11	Full	507.50M	SBT_TAPE	00:00:17	12-SEP-16
BP Key: 14 Status: AVAILABLE Compressed: NO Tag:					
TAG20160912T142231					
Handle: 0jrfrkh17_1_1 Media:					
em2.storage.oraclecloud.com/v1/Storage-dbttestcs4/DJcontainer					
List of Datafiles in backup set 11					
File	LV	Type	Ckp SCN	Ckp Time	Name
3	Full	2018455	12-SEP-16	/u02/app/oracle/oradata/MYORCL/sysaux01.dbf	

BS Key	Type	LV Size	Device Type	Elapsed Time	Completion Time
12	Full	548.25M	SBT_TAPE	00:00:20	12-SEP-16
BP Key: 15 Status: AVAILABLE Compressed: NO Tag:					
TAG20160912T142231					

```

      Handle: 0lrfkhl7_1_1   Media:
em2.storage.oraclecloud.com/v1/Storage-dbtestcs4/DJcontainer
  List of Datafiles in backup set 12
  File LV Type Ckp SCN      Ckp Time  Name
  -----
    4          Full 2018466    12-SEP-16
/u02/app/oracle/oradata/MYORCL/undotbs01.dbf

BS Key   Type LV Size          Device Type Elapsed Time Completion Time
-----
13        Full    680.25M      SBT_TAPE    00:00:22      12-SEP-16
      BP Key: 16   Status: AVAILABLE Compressed: NO   Tag:
TAG20160912T142231

      Handle: 0irfkh17_1_1   Media:
em2.storage.oraclecloud.com/v1/Storage-dbtestcs4/DJcontainer
  List of Datafiles in backup set 13
  File LV Type Ckp SCN      Ckp Time  Name
  -----
    1          Full 2018449    12-SEP-16
/u02/app/oracle/oradata/MYORCL/system01.dbf

BS Key   Type LV Size          Device Type Elapsed Time Completion Time
-----
14        Full    1.50M       SBT_TAPE    00:00:02      12-SEP-16
      BP Key: 17   Status: AVAILABLE Compressed: NO   Tag:
TAG20160912T142231

      Handle: 0prfkhl1_1_1   Media:
em2.storage.oraclecloud.com/v1/Storage-dbtestcs4/DJcontainer
  List of Datafiles in backup set 14
  File LV Type Ckp SCN      Ckp Time  Name
  -----
    6          Full 2018481    12-SEP-16
/u02/app/oracle/oradata/MYORCL/users01.dbf

BS Key   Type LV Size          Device Type Elapsed Time Completion Time
-----
15        Full    1.25M       SBT_TAPE    00:00:02      12-SEP-16
      BP Key: 18   Status: AVAILABLE Compressed: NO   Tag:
TAG20160912T142231

      Handle: 0qrfkhl4_1_1   Media:
em2.storage.oraclecloud.com/v1/Storage-dbtestcs4/DJcontainer
  List of Datafiles in backup set 15
  Container ID: 3, PDB Name: MYPDB1
  File LV Type Ckp SCN      Ckp Time  Name
  -----
    10         Full 2018485    12-SEP-16
/u02/app/oracle/oradata/MYORCL/MYPDB1/MYPDB1_users01.dbf

```

```

BS Key   Type LV Size          Device Type Elapsed Time Completion Time
-----
16       Full    214.50M    SBT_TAPE      00:00:07      12-SEP-16
        BP Key: 19   Status: AVAILABLE Compressed: NO   Tag:
TAG20160912T142231
        Handle: 0nrfkxm0_1_1   Media:
em2.storage.oraclecloud.com/v1/Storage-dbtestcs4/DJcontainer
        List of Datafiles in backup set 16
        Container ID: 2, PDB Name: PDB$SEED
        File LV Type Ckp SCN      Ckp Time  Name
        -----
        5          Full 2015997    12-SEP-16
/u02/app/oracle/oradata/MYORCL/pdbseed/system01.dbf

BS Key   Type LV Size          Device Type Elapsed Time Completion Time
-----
17       Full    214.75M    SBT_TAPE      00:00:06      12-SEP-16
        BP Key: 20   Status: AVAILABLE Compressed: NO   Tag:
TAG20160912T142231
        Handle: 0orfkxm1_1_1   Media:
em2.storage.oraclecloud.com/v1/Storage-dbtestcs4/DJcontainer
        List of Datafiles in backup set 17
        Container ID: 3, PDB Name: MYPDB1
        File LV Type Ckp SCN      Ckp Time  Name
        -----
        8          Full 2018477    12-SEP-16
/u02/app/oracle/oradata/MYORCL/MYPDB1/system01.dbf

BS Key   Type LV Size          Device Type Elapsed Time Completion Time
-----
18       Full    17.25M    SBT_TAPE      00:00:03      12-SEP-16
        BP Key: 21   Status: AVAILABLE Compressed: NO   Tag:
TAG20160912T142315
        Handle: c-342954598-20160912-04   Media:
em2.storage.oraclecloud.com/v1/Storage-dbtestcs4/DJcontainer
        SPFILE Included: Modification time: 12-SEP-16
        SPFILE db_unique_name: MYORCL
        Control File Included: Ckp SCN: 2018634      Ckp time: 12-SEP-16

RMAN> EXIT

Recovery Manager complete.
[oracle@MYDBCS ~]$

```

**Q/ Have the database files listed in the configuration file  
/home/oracle/bkup/<DBNAME>/dbcfg.spec been backed up? (The dbcfgfiles  
zip file is created in /u03/app/oracle/fast\_recovery\_area/<DBNAME>.)**



```

[oracle@MYDBCS ~]$ cd
/u03/app/oracle/fast_recovery_area/MYORCL/ohcfigfiles/201*
[oracle@MYDBCS ~]$ ls *.gz
ohcfigfiles_20160912_1423.tar.gz
[oracle@MYDBCS ~]$
[oracle@MYDBCS ~]$ tar tzvf ohcfigfiles_20160912_1423.tar.gz
-rwx----- oracle/dba          937 2016-09-12 13:58 dbcfg.spec
-rw-rw-rw- oracle/oinstall    10 2016-09-12 14:23 dbid
drwxr-xr-x oracle/oinstall     0 2016-09-12 13:45
u01/app/oracle/product/12.1.0/dbhome_1/admin/MYORCL/xdw_wallet/
-rw-r--r-- oracle/oinstall 3878 2016-09-12 13:45
u01/app/oracle/product/12.1.0/dbhome_1/admin/MYORCL/xdw_wallet/c
wallet.sso
-rw-r--r-- oracle/oinstall 3833 2016-09-12 13:45
u01/app/oracle/product/12.1.0/dbhome_1/admin/MYORCL/xdw_wallet/e
wallet.p12
drwxr-xr-x oracle/oinstall     0 2016-09-12 13:47
u01/app/oracle/admin/MYORCL/db_wallet/
-rw-rw-rw- oracle/oinstall     0 2016-09-12 13:47
u01/app/oracle/admin/MYORCL/db_wallet/cwallet.sso.lck
-rw----- oracle/oinstall   581 2016-09-12 13:47
u01/app/oracle/admin/MYORCL/db_wallet/cwallet.sso
drwxr-xr-x oracle/oinstall     0 2016-09-12 13:57
u01/app/oracle/admin/MYORCL/opc_wallet/
-rw-r--r-- oracle/oinstall 11156 2016-09-12 13:58
u01/app/oracle/admin/MYORCL/opc_wallet/cwallet.sso
drwxr-xr-x oracle/oinstall     0 2016-09-12 13:57
u01/app/oracle/admin/MYORCL/tde_wallet/
-rw----- oracle/oinstall   2408 2016-09-12 13:57
u01/app/oracle/admin/MYORCL/tde_wallet/ewallet_2016091213572657.
p12
-rw----- oracle/oinstall   5373 2016-09-12 13:57
u01/app/oracle/admin/MYORCL/tde_wallet/cwallet.sso
-rw----- oracle/oinstall   5328 2016-09-12 13:57
u01/app/oracle/admin/MYORCL/tde_wallet/ewallet.p12
-rw-r----- oracle/oinstall   7680 2016-09-12 13:47
u01/app/oracle/product/12.1.0/dbhome_1/dbs/orapwMYORCL
-rw-r--r-- oracle/oinstall   437 2016-09-12 13:53
u01/app/oracle/product/12.1.0/dbhome_1/network/admin/listener.or
a
-rw-r--r-- oracle/oinstall   530 2016-09-12 13:57
u01/app/oracle/product/12.1.0/dbhome_1/network/admin/sqlnet.ora
-rw-r--r-- oracle/dba          508 2016-09-12 13:51
u01/app/oracle/product/12.1.0/dbhome_1/network/admin/tnsnames.or
a
-rwxr-xr-x oracle/oinstall 163158 2016-06-28 14:09
u01/app/oracle/product/12.1.0/dbhome_1/rdbms/lib/env_rdbms.mk

```

```
-rwxr-xr-x oracle/oinstall 47329 2016-06-28 14:09
u01/app/oracle/product/12.1.0/dbhome_1/rdbms/lib/ins_rdbms.mk
-rw-rw---- oracle/oinstall 3050 2016-09-12 14:01
var/opt/oracle/creg/MYORCL.ini
[oracle@MYDBCS 2016_09_12]$
```

**A/ Yes.**

**Q/ Was the backup encrypted? Read the backup log.**

```
[oracle@MYDBCS 2016_09_12]$ cd /home/oracle/bkup/MYORCL/log
[oracle@MYDBCS log]$
[oracle@DJDBCS log]$ ls
obkup2016-09-12_14:00:02.log  obkup2016-09-12_14:20:37.log
obkup.log
[oracle@DJDBCS log]$ more obkup.log
OBKUP:: Oracle database state is up and running
catalog mode no
##### PERIODICAL BACKUP STARTING
#####
#####
logfile: /home/oracle/bkup/MYORCL/log/obkup2016-09-
12_14:20:37.log
OBKUP:: DB instance: MYORCL
OBKUP:: Determining if the filesystem is not full
Case OK : Your partition /u03 has 6% used space
Case OK : Your partition /u01 has 14% used space
Case OK : Your partition /u02 has 16% used space
OBKUP:: ..... OK
OBKUP:: Performing backup to local storage (primary backup)
OBKUP:: Executing rman instructions
Connection string:
/u01/app/oracle/product/12.1.0/dbhome_1/bin/rman msgno target
/
-> set encryption on;
-> RECOVER COPY OF DATABASE WITH TAG 'dbaas_incr_backup' UNTIL
TIME 'SYSDATE -
8';
-> BACKUP INCREMENTAL LEVEL 1 FOR RECOVER OF COPY WITH TAG
'dbaas_incr_backup'
DATABASE plus archivelog;
-> DELETE FORCE NOPROMPT ARCHIVELOG ALL BACKED UP 1 TIMES TO
DISK COMPLETED BEF
ORE 'SYSDATE - 1';
-> DELETE FORCE NOPROMPT OBSOLETE;
-> RMAN OUTPUT:
```

```

Recovery Manager: Release 12.1.0.2.0 - Production on Mon
Sep 12 14:20:42 2
016

Copyright (c) 1982, 2014, Oracle and/or its affiliates.
All rights reserved.

RMAN-06005: connected to target database: MYORCL
(DBID=342954598)
...
[oracle@DJDBCS log]$

```

**A/ Yes.**

5. Reenable the scheduled backup configuration (/etc/crontab). Edit the crontab file to reset the obkup command.

```

35 0 * * * root /var/opt/oracle/bkup_api/bkup_api bkup_start --
dbname=MYORCL
0,30 * * * * root /home/oracle/bkup/MYORCL/obkup -dbname=MYORCL
-archivelog

```

**Q/ What does the scheduled command do? Execute the command.**

```

[root@MYDBCS opc]# /home/oracle/bkup/MYORCL/obkup -dbname=MYORCL
-archivelog
Dataguard is not enabled
OBKUP:: Oracle database state is up and running
catalog mode no
OBKUP:: ARCHIVELOGS MANAGEMENT
OBKUP:: BACKING UP TO OSS
OBKUP:: Executing rman instructions
Connection string:
/u01/app/oracle/product/12.1.0/dbhome_1/bin/rman msgno target /
-> SET ENCRYPTION ON;
-> BACKUP DEVICE TYPE SBT AS COMPRESSED BACKUPSET ARCHIVELOG
ALL;
-> RMAN OUTPUT:

Recovery Manager: Release 12.1.0.2.0 - Production on Mon
Sep 12 15:27:00 2016

Copyright (c) 1982, 2014, Oracle and/or its affiliates.
All rights reserved.

```

```
        RMAN-06005: connected to target database: MYORCL
(DBID=342954598)
```

```
    RMAN>
```

```
    RMAN-03029: echo set on
```

```
    RMAN> SET ENCRYPTION ON;
```

```
    RMAN-03023: executing command: SET encryption
```

```
    RMAN-06009: using target database control file instead of
recovery catalog
```

```
    RMAN> BACKUP DEVICE TYPE SBT AS COMPRESSED BACKUPSET
ARCHIVELOG ALL;
```

```
    RMAN-03090: Starting backup at 12-SEP-2016 15:27:03
```

```
    RMAN-03033: current log archived
```

```
    RMAN-08030: allocated channel: ORA_SBT_TAPE_1
```

```
    RMAN-08500: channel ORA_SBT_TAPE_1: SID=260 device
type=SBT_TAPE
```

```
    RMAN-08526: channel ORA_SBT_TAPE_1: Oracle Database Backup
Service Library VER=3.16.7.11
```

```
    RMAN-08030: allocated channel: ORA_SBT_TAPE_2
```

```
    RMAN-08500: channel ORA_SBT_TAPE_2: SID=36 device
type=SBT_TAPE
```

```
    RMAN-08526: channel ORA_SBT_TAPE_2: Oracle Database Backup
Service Library VER=3.16.7.11
```

```
    RMAN-08030: allocated channel: ORA_SBT_TAPE_3
```

```
    RMAN-08500: channel ORA_SBT_TAPE_3: SID=274 device
type=SBT_TAPE
```

```
    RMAN-08526: channel ORA_SBT_TAPE_3: Oracle Database Backup
Service Library VER=3.16.7.11
```

```
    RMAN-08030: allocated channel: ORA_SBT_TAPE_4
```

```
    RMAN-08500: channel ORA_SBT_TAPE_4: SID=26 device
type=SBT_TAPE
```

```
    RMAN-08526: channel ORA_SBT_TAPE_4: Oracle Database Backup
Service Library VER=3.16.7.11
```

```
    RMAN-08030: allocated channel: ORA_SBT_TAPE_5
```

```
    RMAN-08500: channel ORA_SBT_TAPE_5: SID=258 device
type=SBT_TAPE
```

```
    RMAN-08526: channel ORA_SBT_TAPE_5: Oracle Database Backup
Service Library VER=3.16.7.11
```

```
    RMAN-06502: skipping archived logs of thread 1 from
sequence 1 to 4; already backed up
```

```

        RMAN-08049: channel ORA_SBT_TAPE_1: starting compressed
        archived log backup set
        RMAN-08014: channel ORA_SBT_TAPE_1: specifying archived
        log(s) in backup set
        RMAN-08504: input archived log thread=1 sequence=5 RECID=5
        STAMP=922375624
        RMAN-08038: channel ORA_SBT_TAPE_1: starting piece 1 at 12-
        SEP-2016 15:27:10
        RMAN-08044: channel ORA_SBT_TAPE_1: finished piece 1 at 12-
        SEP-2016 15:27:17
        RMAN-08530: piece handle=0srfklee_1_1
        tag=TAG20160912T152710 comment=API Version 2.0,MMS Version
        3.16.7.11
        RMAN-08540: channel ORA_SBT_TAPE_1: backup set complete,
        elapsed time: 00:00:07
        RMAN-03091: Finished backup at 12-SEP-2016 15:27:17

        RMAN-03090: Starting Control File and SPFILE Autobackup at
        12-SEP-2016 15:27:18
        RMAN-08503: piece handle=c-342954598-20160912-05
        comment=API Version 2.0,MMS Version 3.16.7.11
        RMAN-03091: Finished Control File and SPFILE Autobackup at
        12-SEP-2016 15:27:33

        RMAN> set echo off;
        RMAN-03030: echo set off

        RMAN>

        Recovery Manager complete.
        OBKUP:: Clean MOTD.
        OBKUP:: ARCHIVELOG BACKUP COMPLETED
        [root@MYDBCS opc]#

```

***A/ The command cleans up the archivelogs older than one day from the FRA. This command is scheduled every half hour.***

## Practice 22-4: Monitoring the Database Deployment

### Overview

In this practice, you will monitor your database deployment by using the provided Cloud tool DBaaS Monitor.

### Tasks

1. DBaaS Monitor can access the database deployment via HTTPS port (443). There is a predefined security rule allowing anyone (from PUBLIC-INTERNET source) to access your database deployment which is by default disabled.
  - a. To enable the security rule, open the Oracle Compute Cloud Service console.
  - b. Click the Network tab.
  - c. From the left navigator, select Security Rules.
  - d. Locate the appropriate rule to enable. Select the one for your own database deployment.

**Update Security Rule**

Update your security rule as required. You can enable or disable this rule or modify the description. [Learn more.](#)

**Name** MYDBCS/db\_1/ora\_p2\_https

**Status** Enabled

**Security Application** MYDBCS/db\_1/ora\_https

**Source**

- ☐ Security List  
default
- ☒ Security IP List  
public-internet

**Destination**

- ☒ Security List  
MYDBCS/db\_1/ora\_db
- ☐ Security IP List  
instance

**Description**

**Update** **Cancel**

- e. From the menu for the located rule, select Enabled for Status and click Update. The given port on the compute node is opened to the public internet.

2. From a browser, launch DBaaS Monitor to access your database deployment.
  - a. Retrieve the Public IP address of the compute node of your database deployment. Click the Instances tab. Note the address.
  - b. In the browser, use the following URL:  
`https://public_IP_Address_your_VM/dbaas_monitor` to connect to the DBaaS Monitor console.
  - c. A warning message appears: Your connection is not secure. Click Advanced and then Add Exception, then Confirm Security Exception.
  - d. When prompted for a user name and password, enter `dbaas_monitor` as the user name and the password specified when the database deployment was created.
3. Explore the home page.
4. Close your PDB. Read *Administer Pluggable Databases*.  
<http://www.oracle.com/pls/topic/lookup?ctx=cloud&id=CSDBI-GUID-A0A71EFB-FB8F-4466-AD1B-19C8812ACA87>)
5. Reopen your PDB.
6. Find out the Connection Details for `MYPDB1`.
7. Back in the Home page, explore the Online Database Storage.
8. Before finding the list of segments stored in the `USERS` tablespace in your PDB, create a table in the `USERS` tablespace in your PDB with SQL\*Plus.

```
[oracle@MYDBCS ~]$ sqlplus
system/<your_password>@MYDBCS:1521/mypdb1.<your_domain>.oraclecl
oud.internal

SQL*Plus: Release 12.1.0.2.0 Production on Tue Jul 26 15:06:56
2016

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 Oracle Label Security option

SQL> CREATE TABLE system.test (C NUMBER, c2 CHAR(10)) TABLESPACE
users;

Table created.

SQL>
```

9. Back in the Home page, explore the Listener status.
10. Check if your disks are not yet nearly full. (Overall OS Storage)





## **Practices for Lesson 23: SQL\*Plus and SQLcl**

### **Chapter 23**

## Practices for Lesson 23: Overview

---

### Practices Overview

In these practices, you will manipulate the new `HISTORY` command of SQL\*Plus and use the new SQLcl utility. You will also use the new `VALIDATE_CONVERSION` SQL function.

## Practice 23-1: Using SQL\*Plus and SQLcl Commands

### Overview

In this practice, you will manipulate the new commands of SQL\*Plus and use SQLcl, the new take on SQL\*Plus.

### Tasks

1. Before starting the practice, execute the `$HOME/labs/admin/glogin_23.sh` script. The shell script sets formatting for all columns selected in queries.

```
$ $HOME/labs/admin/glogin_23.sh
```

2. List the command history list in the `system` session to `pdb_orcl`.

```
$ . oraenv
ORACLE_SID = [ORCL] ? ORCL
The Oracle base has been set to /u01/app/oracle
$ sqlplus system@pdb_orcl
Enter password: *****
Connected.
SQL> HISTORY
SP2-1650: History is off, use "SET HIST[ORY] ON" to enable
History.
SQL>
```

3. After activating the history, load commands by executing the `$HOME/labs/SQL/setup_hist.sql` SQL script in your session. It executes many commands in `pdb_orcl`.

```
SQL> SET HISTORY ON
SQL> HISTORY
SP2-1651: History list is empty.
SQL>
SQL> @$HOME/labs/SQL/setup_hist.sql
...
SQL>
```

4. List the command history list and run one of them.

```
SQL> HISTORY
 1  @$HOME/labs/SQL/setup_hist.sql
SQL>
```

*Q/ What do you observe?*

***A/ The commands executed through the SQL script are not visible in the command history list.***

5. Copy and paste two of the commands executed by the SQL script. Revisit the command history list.

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

NAME
```

```

-----
/u02/app/oracle/oradata/ORCL/control01.ctl
/u03/app/oracle/fast_recovery_area/ORCL/control02.ctl

SQL> SELECT member from v$logfile;

MEMBER
-----
/u04/app/oracle/redo/ORCL/redo03.log
/u04/app/oracle/redo/ORCL/redo02.log
/u04/app/oracle/redo/ORCL/redo01.log

SQL>
SQL> HISTORY
  1  @$HOME/labs/SQL/setup.sql
  2  select name from v$controlfile;
  3  SELECT member from v$logfile;
SQL>

```

6. Reexecute the second one.

```

SQL> HIST 2 run

NAME
-----
/u02/app/oracle/oradata/ORCL/control01.ctl
/u03/app/oracle/fast_recovery_area/ORCL/control02.ctl

SQL>

```

7. Delete the command 1 that ran the SQL script from the command history list.

```

SQL> HIST 1 delete
SQL> HISTORY
  1  select name from v$controlfile;
  2  SELECT member from v$logfile;
SQL>

```

8. Execute a new manual command and reexecute command 2 from the command history list.

```

SQL> SELECT sysdate FROM dual;

SYSDATE
-----
02-MAY-16

SQL> HIST 2 run

```

```
MEMBER
```

```
-----  
/u04/app/oracle/redo/ORCL/redo03.log  
/u04/app/oracle/redo/ORCL/redo02.log  
/u04/app/oracle/redo/ORCL/redo01.log
```

```
SQL> HIST
```

```
1 select name from v$controlfile;  
* 2 SELECT member from v$logfile;  
3 SELECT sysdate FROM dual;
```

```
SQL>
```

***An asterisk (\*) indicates the last used command in the command history list.***

9. Reuse command 2 from the command history list and add an ORDER BY clause.

```
SQL> HIST 2 edit
```

```
SELECT member from v$logfile; order by 1  
~  
~
```

```
SQL>
```

```
SQL> HIST
```

```
1 select name from v$controlfile;  
2 SELECT member from v$logfile;  
3 SELECT sysdate FROM dual;  
4 SELECT member from v$logfile; ORDER BY 1
```

```
SQL> HIST 4 edit
```

```
...
```

```
SQL> HIST
```

```
1 select name from v$controlfile;  
2 SELECT member from v$logfile;  
3 SELECT sysdate FROM dual;  
4 SELECT member from v$logfile; ORDER BY 1  
5 SELECT member from v$logfile ORDER BY 1;
```

```
SQL> HIST 5 run
```

```
MEMBER
```

```
-----  
/u04/app/oracle/redo/ORCL/redo01.log  
/u04/app/oracle/redo/ORCL/redo02.log  
/u04/app/oracle/redo/ORCL/redo03.log
```

```
SQL> EXIT
$
```

*Q2/ Can you do the same type of operation under SQLcl?*

**A2/ Regarding the history and command recall, SQLcl is able to complete the same operations as SQL\*Plus, even more.**

*Q3/ What happens to the command history list if you reconnect to a system session to pdb\_orcl?*

```
$ sqlplus system@pdb_orcl
Enter password: *****
Connected.
SQL> HISTORY
SP2-1650: History is off, use "SET HIST[ORY] ON" to enable
History.
SQL> SET HISTORY ON
SQL> HISTORY
SP2-1651: History list is empty.
SQL> EXIT
$
```

**A3/ The command history list is not maintained from one session to the next.**

10. Connect to SQLcl as system to pdb\_orcl.

- a. Install SQLcl by executing the \$HOME/labs/SQL/install\_SQLcl.sh shell script. The shell script unzips the downloaded sqlcl-4.2.0.16.112.0616-no-jre.zip file.

```
$ $HOME/labs/SQL/install_SQLcl.sh
...
$
```

- b. Connect to SQLcl.

```
$ export PATH=$ORACLE_HOME/jdk/bin:$PATH
$ cd $ORACLE_HOME/sqldeveloper/sqlcl/bin
$ ./sql system@pdb_orcl
SQLcl: Release 4.2.0.16.131.1023 RC on Mon May 23 03:36:22 2016

Copyright (c) 1982, 2016, Oracle. All rights reserved.

Password? (*****?) *****
Last Successful login time: Mon May 23 2016 03:36:25 +00:00

Connected to:
Oracle Database 12c Enterprise Edition Release 12.2.0.1.0 -
64bit Production

SP2-0158: unknown SET option "history"
```

```
SQL>
```

*Q1/ Is the command history list maintained from one session to another?*

```
SQL> history
```

```
History: show no failures
```

```
SQL> SELECT * FROM dual;
```

```
D
```

```
-
```

```
X
```

```
SQL> EXIT
```

```
Disconnected from Oracle Database 12c Enterprise Edition Release  
12.2.0.1.0 - 64bit Production
```

```
$ ./sql system@pdb_orcl
```

```
SQLcl: Release 4.2.0.16.131.1023 RC on Mon May 23 03:37:23 2016
```

```
Copyright (c) 1982, 2016, Oracle. All rights reserved.
```

```
Password? (*****?) *****
```

```
Last Successful login time: Mon May 23 2016 03:37:27 +00:00
```

```
Connected to:
```

```
Oracle Database 12c Enterprise Edition Release 12.2.0.1.0 -  
64bit Production
```

```
SQL> hist
```

```
History: show no failures
```

```
1 SELECT * FROM dual
```

```
SQL>
```

***A1/ Unlike SQL\*Plus, SQLcl maintains the command history list from one session to another.***

*Q2/ What does SQLcl display that SQL\*Plus does not display and requires you to look in configuration files?*

```
SQL> show all
```

```
_prelim OFF
```

```
appinfo is OFF and set to "java@edcdr19p1 (TNS V1-V3)"
```

```
arraysize 15
```

```
autocommit OFF
```

```
autoprint OFF
```

```
autotrace OFF
```

Copyright © 2016, Oracle and/or its affiliates. All rights reserved.

```
BTitle OFF
SQL*Plus mode: OFF
Clear Screen mode: top
No CLOUDCONFIG location defined

colsep " "
concat "." (hex 2e)
copycommit 0
define "&"
echo OFF
editfile "afiedt.buf"

embedded OFF
escape "\" (hex 5c)
exitcommit ON
FEEDBACK ON for 6 or more rows
heading ON
headsep "|" (hex 7c)
linesize 68
long 80
longchunksize 80
newpage 1
NOHISTORY: default
null /*notset*/
numformat ""
numwidth 10
pagesize 100
PAUSE is OFF
release 1202000100
scan ON
serveroutput OFF
showmode OFF
space " "
spool OFF
sqlcode 0
sqlprompt "SQL> "
suffix "sql"
termout ON
time OFF
timing OFF
TNS Lookup locations
-----
```



1. USER Home dir  
/home/oracle
2. ORACLE\_HOME  
/u01/app/oracle/product/12.2.0/dbhome\_1/network/admin

Location used:

-----

/u01/app/oracle/product/12.2.0/dbhome\_1/network/admin

#### **Available TNS Entries**

-----

ACCOUNTING  
CDB1  
CDB2  
CDBEM  
CONV\_PDB2  
DOLLS  
DOODLES  
HR\_ROOT  
NEWPDB  
OPERATIONS  
ORCL  
PDB\_ENCRYPT  
PDB\_IM  
PDB\_ORCL  
PDB\_SOURCE\_FOR\_HOTCLONE  
PDB1\_1  
PDB1\_2  
PDB1  
PDB2\_IN\_ORCL  
PDB2\_NODE1  
PDB2\_NODE2  
PDB2  
PDBEM  
PX\_TOYS\_RR  
RESEARCH  
ROBOTS  
SALES  
SAMPLE\_PDB  
TOYS\_ROOT\$SEED  
TOYS\_ROOT  
TOYS\_RR

```

trimout OFF
trimspool OFF
tttitle OFF and is the first few characters of the next SELECT
statementUSER is "SYSTEM"
verify ON
No Wallet location defined

wrap : lines will be wrapped
SQL>

```

**A2/ The content of the *tnsnames.ora* file is automatically displayed through the *SHOW ALL* command.**

- c. Use the top arrow to move to a command in the history that contains a typo.

```

SQL> select USER_ANME
      2  from DBA_USERS;

Error starting at line : 1 in command -
select USER_ANME
from DBA_USERS
Error at Command Line : 1 Column : 8
Error report -
SQL Error: ORA-00904: "USER_ANME": invalid identifier
00904. 00000 - "%s: invalid identifier"
*Cause:
*Action:
SQL>

```

- 1) Move to the misspelled word in the last command. Use the ↑ arrow to reach the command, then `ctrl W` to reach the top of the buffer.
- 2) Use the → and ← arrows to reach the misspelled word `USER_ANME` and correct the characters to get `USERNAME`.
- 3) Use `ctrl R` to run the command.

```

SQL> select USERNAME
      2  from DBA_USERS
USERNAME
-----
SYS
SYSTEM
...
OE

38 rows selected.

SQL>

```

- d. Use input parameters in commands.

- 1) Execute the following command with `CHAR` as the data type:

```
SQL> select owner,table_name, column_name
      from dba_tab_cols
      where data_type = '&DATA_TYPE'
      and owner not in ('GSMADMIN_INTERNAL','LBACSYS', 'DVSYS',
'CTXSYS', 'SYS', 'SYSTEM', 'SYSAUX','SYSMAN', 'MDSYS','WMSYS')
      order by 1,2,3
      fetch first 25 rows only;
```

Enter value for DATA\_TYPE: **CHAR**

```
old:select owner,table_name, column_name
from dba_tab_cols
where data_type = '&DATA_TYPE'
and owner not in ('GSMADMIN_INTERNAL','LBACSYS', 'DVSYS',
'CTXSYS', 'SYS', 'SYSTEM', 'SYSAUX','SYSMAN', 'MDSYS','WMSYS')
order by 1,2,3
fetch first 25 rows only
new:select owner,table_name, column_name
from dba_tab_cols
where data_type = 'CHAR'
and owner not in ('GSMADMIN_INTERNAL','LBACSYS', 'DVSYS',
'CTXSYS', 'SYS', 'SYSTEM', 'SYSAUX','SYSMAN', 'MDSYS','WMSYS')
order by 1,2,3
fetch first 25 rows only
```

OWNER	TABLE_NAME	COLUMN_NAME
APPQOSSYS	WLM_CLASSIFIER_PLAN	ACTIVE
DBSNMP	MGMT_BSLN_DATASOURCES	SOURCE_TYPE
DBSNMP	MGMT_BSLN_METRICS	TAIL_ESTIMATOR
DBSNMP	MGMT_BSLN_METRICS	THRESHOLD_METHOD_DEFAULT
XDB	XDB\$NONCEKEY	NONCEKEY

SQL>

- 2) Use the **↑** arrow to reach the same command and reexecute it with `DATE` as the data type.

```
SQL> select owner,table_name, column_name
      from dba_tab_cols
      where data_type = '&DATA_TYPE'
      and owner not in ('GSMADMIN_INTERNAL','LBACSYS', 'DVSYS',
'CTXSYS', 'SYS', 'SYSTEM', 'SYSAUX','SYSMAN', 'MDSYS','WMSYS')
      order by 1,2,3
```

```

    fetch first 25 rows only;
Enter value for DATA_TYPE: DATE
old:select owner,table_name, column_name
from dba_tab_cols
where data_type = '&DATA_TYPE'
and owner not in ('GSMADMIN_INTERNAL','LBACSYS', 'DVSYS',
'CTXSYS', 'SYS', 'SYSTEM', 'SYSAUX','SYSMAN', 'MDSYS','WMSYS')
order by 1,2,3
fetch first 25 rows only
new:select owner,table_name, column_name
from dba_tab_cols
where data_type = 'CHAR'
and owner not in ('GSMADMIN_INTERNAL','LBACSYS', 'DVSYS',
'CTXSYS', 'SYS', 'SYSTEM', 'SYSAUX','SYSMAN', 'MDSYS','WMSYS')
order by 1,2,3
fetch first 25 rows only

```

OWNER	TABLE_NAME	COLUMN_NAME
-----	-----	-----
APPQOSSYS	WLM_CLASSIFIER_PLAN	TIMESTAMP
APPQOSSYS	WLM_METRICS_STREAM	TIMESTAMP
APPQOSSYS	WLM_VIOLATION_STREAM	TIMESTAMP
DBSNMP	BSLN_BASELINES	LAST_COMPUTE_DATE
DBSNMP	BSLN_STATISTICS	COMPUTE_DATE
DBSNMP	BSLN_THRESHOLD_PARAMS	LAST_SET_DATE
DBSNMP	MGMT_BASELINE	CAPTURE_TIME
DBSNMP	MGMT_BASELINE	PREV_CAPTURE_TIME
DBSNMP	MGMT_BSLN_STATISTICS	COMPUTE_DATE
DBSNMP	MGMT_CAPTURE	CAPTURE_TIME
DBSNMP	MGMT_HISTORY	CAPTURE_TIME
DBSNMP	MGMT_HISTORY	PREV_CAPTURE_TIME
DBSNMP	MGMT_LATEST	CAPTURE_TIME
DBSNMP	MGMT_LATEST	PREV_CAPTURE_TIME
DBSNMP	MGMT_RESPONSE_CONFIG	STARTUP_TIME
DBSNMP	MGMT_SNAPSHOT	CAPTURE_TIME
OJVMSYS	OJDS\$INODE\$	CREATION_TS
OJVMSYS	OJDS\$INODE\$	LAST_MODIFIED
OJVMSYS	OJDS\$SHARED\$OBJ\$	TIMESTAMP
OLAPSYS	XML_LOAD_LOG	XML_DATE
ORDDATA	ORDDCM_DOCS	CREATE_DATE
ORDDATA	ORDDCM_DOCS_TMP	CREATE_DATE
ORDDATA	ORDDCM_DOCS_USR	CREATE_DATE
ORDDATA	ORDDCM_DOCS_WRK	CREATE_DATE

```

ORDDATA          ORDDCM_DOCUMENTS          CREATE_DATE

25 rows selected.

SQL>

```

- e. After counting the number of times the command was executed, you decide to create an alias for this long text command.

```

SQL> history usage
1  (1) SELECT * FROM dual
2  (1) show all
3  (1) select USERNAME from DBA_USERS
4  (2) select owner,table_name, column_name      from
dba_tab_cols      where
SQL>

```

- 1) Create the datatype alias for this long text command.

```

SQL> alias datatype=select owner,table_name, column_name
      from dba_tab_cols
      where data_type = '&DATA_TYPE'
      and owner not in ('GSMADMIN_INTERNAL','LBACSYS', 'DVSYS',
'CTXSYS', 'SYS', 'SYSTEM', 'SYS_AUX','SYSMAN', 'MDSYS','WMSYS')
      order by 1,2,3
      fetch first 25 rows only;
Enter value for DATA_TYPE: CHAR
SQL>

```

- f. Use the datatype alias to reexecute the command.

```

SQL> datatype

OWNER          TABLE_NAME          COLUMN_NAME
-----
APPQOSSYS      WLM_CLASSIFIER_PLAN  ACTIVE
DBSNMP         MGMT_BSLN_DATASOURCES SOURCE_TYPE
DBSNMP         MGMT_BSLN_METRICS    TAIL_ESTIMATOR
DBSNMP         MGMT_BSLN_METRICS    THRESHOLD_METHOD_DEFAULT
...
XDB            XDB$NONCEKEY         NONCEKEY

SQL>

```

*Q1/ How can you recreate the alias so as to be able to enter a new data type at each execution?*

```

SQL> alias datatype=select owner,table_name, column_name
      from dba_tab_cols
      where data_type = :DATA_TYPE

```

```

and owner not in ('GSMADMIN_INTERNAL','LBACSYS', 'DVSYS',
'CTXSYS', 'SYS', 'SYSTEM', 'SYS_AUX','SYSMAN', 'MDSYS','WMSYS')
order by 1,2,3
fetch first 25 rows only;

```

SQL>

SQL> **datatype DATE**

OWNER	TABLE_NAME	COLUMN_NAME
APPQOSSYS	WLM_CLASSIFIER_PLAN	TIMESTAMP
APPQOSSYS	WLM_METRICS_STREAM	TIMESTAMP
APPQOSSYS	WLM_VIOLATION_STREAM	TIMESTAMP
DBSNMP	BSLN_BASELINES	LAST_COMPUTE_DATE
DBSNMP	BSLN_STATISTICS	COMPUTE_DATE
DBSNMP	BSLN_THRESHOLD_PARAMS	LAST_SET_DATE
DBSNMP	MGMT_BASELINE	CAPTURE_TIME
DBSNMP	MGMT_BASELINE	PREV_CAPTURE_TIME
DBSNMP	MGMT_BSLN_STATISTICS	COMPUTE_DATE
DBSNMP	MGMT_CAPTURE	CAPTURE_TIME
DBSNMP	MGMT_HISTORY	CAPTURE_TIME
DBSNMP	MGMT_HISTORY	PREV_CAPTURE_TIME
DBSNMP	MGMT_LATEST	CAPTURE_TIME
DBSNMP	MGMT_LATEST	PREV_CAPTURE_TIME
DBSNMP	MGMT_RESPONSE_CONFIG	STARTUP_TIME
DBSNMP	MGMT_SNAPSHOT	CAPTURE_TIME
OJVMSYS	OJDS\$INODE\$	CREATION_TS
OJVMSYS	OJDS\$INODE\$	LAST_MODIFIED
OJVMSYS	OJDS\$SHARED\$OBJ\$	TIMESTAMP
OLAPSYS	XML_LOAD_LOG	XML_DATE
ORDDATA	ORDDCM_DOCS	CREATE_DATE
ORDDATA	ORDDCM_DOCS_TMP	CREATE_DATE
ORDDATA	ORDDCM_DOCS_USR	CREATE_DATE
ORDDATA	ORDDCM_DOCS_WRK	CREATE_DATE
ORDDATA	ORDDCM_DOCUMENTS	CREATE_DATE

25 rows selected.

SQL>

SQL> **set sqlformat ansiconsole**

SQL> **datatype CHAR**

OWNER	TABLE_NAME	COLUMN_NAME
-------	------------	-------------

APPQOSSYS	WLM_CLASSIFIER_PLAN	ACTIVE
DBSNMP	MGMT_BSLN_DATASOURCES	SOURCE_TYPE
DBSNMP	MGMT_BSLN_METRICS	TAIL_ESTIMATOR
DBSNMP	MGMT_BSLN_METRICS	THRESHOLD_METHOD_DEFAULT
...		
XDB	XDB\$NONCEKEY	NONCEKEY

SQL>

**A1/ Create the alias using the : before the input parameter.**

**Q2/ How do you get the equivalent command of an alias?**

```
SQL> alias
datatype
locks
sessions
tables
tables2
SQL> alias list datatype
datatype
-----

select owner,table_name, column_name
      from dba_tab_cols
      where data_type = 'DATE'
      and owner not in ('GSMADMIN_INTERNAL','LBACSYS', 'DVSYS',
'CTXSYS', 'SYS', 'SYSTEM', 'SYSAUX','SYSMAN', 'MDSYS','WMSYS')
      order by 1,2,3
      fetch first 25 rows only

SQL>
```

**A2/ Use the alias command to get the full list of aliases and the alias list command to list the contents of the alias.**

- g. Use the **info** command to get the definition details of the `oe.lineorder` table (or any other existing table). Add a primary key on this table.

```
SQL> CREATE INDEX i_sqlcl ON oe.lineorder (lo_orderkey);

Index i_sqlcl created.

SQL> info oe.lineorder
TABLE: LINEORDER
      LAST ANALYZED:2016-10-13 02:11:28.0
      ROWS           :8009720
      SAMPLE SIZE    :8009720
```

INMEMORY	:ENABLED			
COMMENTS	:			
Columns				
NAME	DATA TYPE	NULL	DEFAULT	COMMENTS
LO_ORDERKEY	NUMBER	Yes		
LO_LINENUMBER	NUMBER	Yes		
LO_CUSTKEY	NUMBER	Yes		
LO_PARTKEY	NUMBER	Yes		
LO_SUPPKEY	NUMBER	Yes		
LO_ORDERDATE	NUMBER	Yes		
LO_ORDERPRIORITY	CHAR(15 BYTE)	Yes		
LO_SHIPPRIORITY	CHAR(1 BYTE)	Yes		
LO_QUANTITY	NUMBER	Yes		
LO_EXTENDEDPRICE	NUMBER	Yes		
LO_ORDTOTALPRICE	NUMBER	Yes		
LO_DISCOUNT	NUMBER	Yes		
LO_REVENUE	NUMBER	Yes		
LO_SUPPLYCOST	NUMBER	Yes		
LO_TAX	NUMBER	Yes		
LO_COMMITDATE	NUMBER	Yes		
LO_SHIPMODE	CHAR(10 BYTE)	Yes		
Indexes				
INDEX_NAME	UNIQUENESS	STATUS	FUNCIDX_STATUS	COLUMNS
COLUMN_EXPRESSION				
SYSTEM.I_SQLCL	NONUNIQUE	VALID		LO_ORDERKEY
SQL>				

**Q/ What is the difference between the *info* command and the *info+* command?**

SQL> <b>info+ oe.lineorder</b>				
TABLE: LINEORDER				
LAST ANALYZED:2016-10-13 02:11:28.0				
ROWS :8009720				
SAMPLE SIZE :8009720				
INMEMORY :ENABLED				
COMMENTS :				
Columns				
NAME	DATA TYPE	NULL	DEFAULT	LOW_VALUE
HIGH_VALUE	NUM_DISTINCT	HISTOGRAM		
LO_ORDERKEY	NUMBER	Yes		4998819
48000004	1966592	NONE		



LO_LINENUMBER	NUMBER	Yes	1
7	7	NONE	
LO_CUSTKEY	NUMBER	Yes	1
270018	90848	NONE	
LO_PARTKEY	NUMBER	Yes	1
740070	637440	NONE	
LO_SUPPKEY	NUMBER	Yes	1
130085	6063	NONE	
LO_ORDERDATE	NUMBER	Yes	19920101
19981014	3990	NONE	
LO_ORDERPRIORITY	CHAR(15 BYTE)	Yes	
5	NONE		
LO_SHIPPRIORITY	CHAR(1 BYTE)	Yes	
1	NONE		
LO_QUANTITY	NUMBER	Yes	1
50	50	NONE	
LO_EXTENDEDPRICE	NUMBER	Yes	90300
10469950	610880	NONE	
LO_ORDTOTALPRICE	NUMBER	Yes	87552
49975301	250128	NONE	
LO_DISCOUNT	NUMBER	Yes	0
10	11	NONE	
LO_REVENUE	NUMBER	Yes	81720
3767382000	6559232	NONE	
LO_SUPPLYCOST	NUMBER	Yes	54060
2518780000	77944	NONE	
LO_TAX	NUMBER	Yes	0
464	72	NONE	
LO_COMMITDATE	NUMBER	Yes	19920131
19981633	4015	NONE	
LO_SHIPMODE	CHAR(10 BYTE)	Yes	
7	NONE		
Indexes			
INDEX_NAME	UNIQUENESS	STATUS	FUNCIDX_STATUS
COLUMN_EXPRESSION			COLUMNS
SYSTEM.I_SQLCL	NONUNIQUE	VALID	LO_ORDERKEY
SQL>			

**A/ You retrieve the histogram values collected during statistics collection.**

- h. Execute the /home/oracle/labs/SQL/script\_test.sql SQL script. Edit the script if you wish to change the table request.

```
SQL> cd /home/oracle/labs/SQL
SQL> pwd
```

```

/home/oracle/labs/SQL
SQL> start script_test
LO_LINENUMBER  SUM_QTY
1              51130032
6              14636312
2              43770208
4              29200912
5              21901016
3              36526048
7              7346800

```

7 rows selected.

```
SQL>
```

**Q1/ Does the script execution belong to history?**

```

SQL> hist
History: show no failures
 1  SELECT * FROM dual
 2  show all
 3  select USERNAME from DBA_USERS
 4  select owner,table_name, column_name      from dba_tab_cols
where data
 5  alias datatype=select owner,table_name, column_name
from dba_tab_cols
 6  datatype
 7  alias datatype=select owner,table_name, column_name
from dba_tab_cols
 8  datatype DATE
 9  set sqlformat ansiconsole
10  datatype CHAR
11  alias
12  alias list datatype
13  info sh.sales
14  info+ sh.sales
15  cd /home/oracle/labs/SQL
16  pwd
17  start script_test
SQL>

```

**A1/ Yes.**

**Q2/ Can SQL scripts from the history be reexecuted?**

**A2/ Yes. Move up the ↑ arrow to reach the START script from the history and enter to reexecute it.**

```
SQL> start script_test
```

```
...
```

- i. Clear up the whole history.

```
SQL> history clear
```

```
History Cleared
```

```
SQL> EXIT
```

```
$ ./sql system@pdb_orcl
```

```
SQLcl: Release 4.2.0.16.131.1023 RC on Mon May 23 03:54:50 2016
```

```
Copyright (c) 1982, 2016, Oracle. All rights reserved.
```

```
Password? (*****?) *****
```

```
Last Successful login time: Mon May 23 2016 03:54:53 +00:00
```

```
Connected to:
```

```
Oracle Database 12c Enterprise Edition Release 12.2.0.1.0 -  
64bit Production
```

```
SQL> hist
```

```
History: show no failures
```

```
SQL> EXIT
```

```
$
```

## Practice 23-2: Identifying Nonconvertible Columns

### Overview

In this practice, you will use the new `VALIDATE_CONVERSION` SQL function to identify rows in a table that will fail data type conversion.

### Tasks

1. Before starting the practice, execute the `$HOME/labs/SQL/setup_val.sh` script that creates the `system.tab_val` table in `pdb_orcl`.

```
$ $HOME/labs/SQL/setup_val.sh
...
Name                                     Null?      Type
-----
C_TO_NBR                                VARCHA2(10)
C_TO_CHAR                                NUMBER
C_TO_DATE                                CHAR(10)
$
```

2. Identify the rows in the `system.tab_val` table where the conversion of the `c_to_nbr` column values to `NUMBER` data type would fail.

```
$ sqlplus system@pdb_orcl
Enter password: *****
Connected.
SQL> SELECT * FROM tab_val
      WHERE VALIDATE_CONVERSION(tab_val.c_to_nbr AS NUMBER) = 0;
2
C_TO_NBR    C_TO_CHAR C_TO_DATE
-----
A              200 May 10, 2016, 14:18

SQL>
```

*Q/ What would happen if you converted the column values to insert them in another table?*

```
SQL> CREATE TABLE test
      AS SELECT TO_NUMBER(c_to_nbr) nbr FROM tab_val;
2
      AS SELECT TO_NUMBER(c_to_nbr) nbr FROM tab_val
      *
ERROR at line 2:
ORA-01722: invalid number

SQL>
```

***A/ The conversion detects the row where the column value is not convertible to the `NUMBER` data type.***

3. Create the table with rows compatible with the conversion for `c_to_nbr` column.

```
SQL> CREATE TABLE test
      AS SELECT TO_NUMBER(c_to_nbr) nbr FROM tab_val
      WHERE VALIDATE_CONVERSION(tab_val.c_to_nbr AS NUMBER) = 1;
2      3
Table created.

SQL> SELECT * FROM test ORDER BY 1 DESC;

          NBR
-----
          1
          0

SQL>
```

4. Identify the rows in the `system.tab_val` table where the conversion of the `c_to_char` column values to `CHAR` data type would fail.

```
SQL> SELECT * FROM tab_val
      WHERE VALIDATE_CONVERSION(tab_val.c_to_char AS CHAR) = 0;
2      WHERE VALIDATE_CONVERSION(tab_val.c_to_char AS CHAR) =
0
                                     *

ERROR at line 2:
ORA-43908: invalid output data type

SQL>
```

*Q/ Why does it fail?*

***A/ The output data type is not a valid type of conversion.***

***oerr ora 43908***

***43908, 0000, "invalid output data type"***

***// \*Document: YES***

***// \*Cause: An invalid output data type was provided.***

***// The valid output data types for VALIDATE\_CONVERSION operator and  
// the operators supporting DEFAULT ON CONVERSION ERROR syntax  
are:***

***// NUMBER, BINARY\_FLOAT, BINARY\_DOUBLE, DATE, TIMESTAMP,  
// TIMESTAMP WITH TIME\_ZONE, INTERVAL DAY TO SECOND,  
// INTERVAL YEAR TO MONTH.***

***// \*Action: Provide a valid output data type.***

**Converting a number to a character (CHAR, VARCHAR2) data type is always possible as digits are a subset of the string characters. Thus, it is useless to provide this type of validation.**

```
SQL> CREATE TABLE test2
      AS SELECT TO_CHAR(c_to_char) chr FROM tab_val;
2
Table created.

SQL> SELECT * FROM test2 ORDER BY 1 DESC;

CHR
-----
400
300
200
100

SQL>
```

5. Identify the rows in the `system.tab_val` table where the conversion of the `c_to_date` column values to `DATE` data type would fail, considering that the stored values are of the 'Month dd, YYYY, HH24:MI' date format.

```
SQL> SELECT * FROM tab_val
      WHERE VALIDATE_CONVERSION(tab_val.c_to_date AS DATE,
                                'Month dd, YYYY, HH24:MI',
                                'NLS_DATE_LANGUAGE = American') = 0;
2      3      4

C_TO_NBR      C_TO_CHAR C_TO_DATE
-----
0              300 Janvier 31, 2015, 10:20
              400 February 30, 2016, 4:18

SQL>
```

**Q/ What prevents the two values from being converted to the `DATE` data type?**

**A/ In the first row, the string value contains a French month name whereas the `NLS_DATE_LANGUAGE` expects months in American. In the second row, the day is 30 for February whereas February can only have 28 or 29 days.**

6. Create the table with rows compatible with the conversion for `c_to_date` column.

```
SQL> CREATE TABLE test3
      AS SELECT TO_DATE(c_to_date, 'Month dd, YYYY, HH24:MI') dte
      FROM tab_val
      WHERE VALIDATE_CONVERSION(tab_val.c_to_date AS DATE,
```

```
'Month dd, YYYY, HH24:MI',  
'NLS_DATE_LANGUAGE = American')=1;
```

```
2      3      4      5      6  
Table created.
```

```
SQL> SELECT * FROM test3 ORDER BY 1 DESC;
```

```
DTE  
-----  
10-MAY-16  
20-JUL-69
```

```
SQL> EXIT  
$
```

