Oracle Database 19c: New Features

# Table of Contents

[Course Practice Environment: Security Credentials 5](#_TOC_250039)

[Course Practice Environment: Security Credentials 6](#_TOC_250038)

[Practices for Lesson 1: Using General Database Overall Enhancements 7](#_TOC_250037)

[Practices for Lesson 1: Overview 8](#_TOC_250036)

[Practice 1-1: Discovering the Practices Environment 9](#_TOC_250035)

Practice 1-2: Installing Oracle Database 19c with Automatic root.sh Execution 10

[Practice 1-3: Cloning a PDB by Using DBCA in Silent Mode 12](#_TOC_250034)

[Practice 1-4: Relocating a PDB by Using DBCA in Silent Mode 15](#_TOC_250033)

[Practice 1-5: Duplicating a CDB by Using DBCA in Silent Mode 18](#_TOC_250032)

[Practice 1-6: Decreasing TTS Import Time 26](#_TOC_250031)

[Practice 1-7: Decreasing TTS Export Time 31](#_TOC_250030)

[Practice 1-8: Omitting the Column Encryption Attribute During Import 36](#_TOC_250029)

[Practice 1-9: Avoiding Errors Due to Values Generated by LISTAGG 41](#_TOC_250028)

[Practices for Lesson 2: Using Security Enhancements 51](#_TOC_250027)

[Practices for Lesson 2: Overview 52](#_TOC_250026)

[Practice 2-1: Exploring Oracle-supplied Schemas Only Accounts 53](#_TOC_250025)

[Practice 2-2: Protecting Application Data by Using Database Vault Operations Control 56](#_TOC_250024)

Practice 2-3: Constraining AUDIT POLICY and NOAUDIT POLICY SQL Commands with

Oracle Database Vault Command Rules 69

[Practice 2-4: Auditing Direct User Activities 74](#_TOC_250023)

[Practice 2-5: Handling Operations on Oracle-managed and User-managed Tablespaces Encrypted in TDE 78](#_TOC_250022)

[Practices for Lesson 3: Using Availability Enhancements 85](#_TOC_250021)

[Practices for Lesson 3: Overview 86](#_TOC_250020)

[Practice 3-1: Using RMAN to Connect to a PDB and Use the Recovery Catalog 87](#_TOC_250019)

[Practice 3-2: Exploring Automatic Deletion of Flashback Logs 95](#_TOC_250018)

[Practices for Lesson 4: Using Performance Enhancements 103](#_TOC_250017)

[Practices for Lesson 4: Overview 104](#_TOC_250016)

[Practice 4-1: Using Memoptimized Rowstore - Fast Ingest 105](#_TOC_250015)

[Practice 4-2: Completing an ADDM Analysis Inside a PDB 114](#_TOC_250014)

[Practice 4-3: Using Real-time SQL Monitoring as a SQL Developer 120](#_TOC_250013)

[Practice 4-4: Waiting In-memory Objects to be Populated 124](#_TOC_250012)

[Practice 4-5: Configuring and Using Database Replay at PDB Level 135](#_TOC_250011)

[Practices for Lesson 5: Using Big Data and Data Warehousing Enhancements 139](#_TOC_250010)

[Practices for Lesson 5: Overview 140](#_TOC_250009)

[Practice 5-1: Managing Hybrid Partitioned Tables 141](#_TOC_250008)

[Practice 5-2: Creating and Populating In-memory Hybrid Partitioned Tables 155](#_TOC_250007)

[Practice 5-3: Creating and Populating In-memory External Tables 160](#_TOC_250006)

[Practices for Lesson 6: Using Diagnosability Enhancements 165](#_TOC_250005)

[Practices for Lesson 6: Overview 166](#_TOC_250004)

[Practice 6-1: Fixing SQL Statements by Using Automatic SQL Diagnosis and Repair 167](#_TOC_250003)

[Practice 6-2: Creating SRDCs for Specific Events 184](#_TOC_250002)

[Practices for Lesson 7: Using Sharding Enhancements 203](#_TOC_250001)

[Practices for Lesson 7 204](#_TOC_250000)

# Course Practice Environment: Security Credentials

## Course Practice Environment: Security Credentials

### Overview

For OS usernames and passwords, ask your instructor for OS credential information.

For product-specific credentials used in this course, see the following table:

|  |  |  |
| --- | --- | --- |
| **Product-Specific Credentials** | | |
| **Product/Application** | **Username** | **Password** |
| VM1 |  |  |
| ORCL and PDB1 | Any user | Welcome\_1 |
| Other PDBs created in ORCL | Any user | Welcome\_1 |
| Other CDBs created and PDBs | Any user | Welcome\_1 |

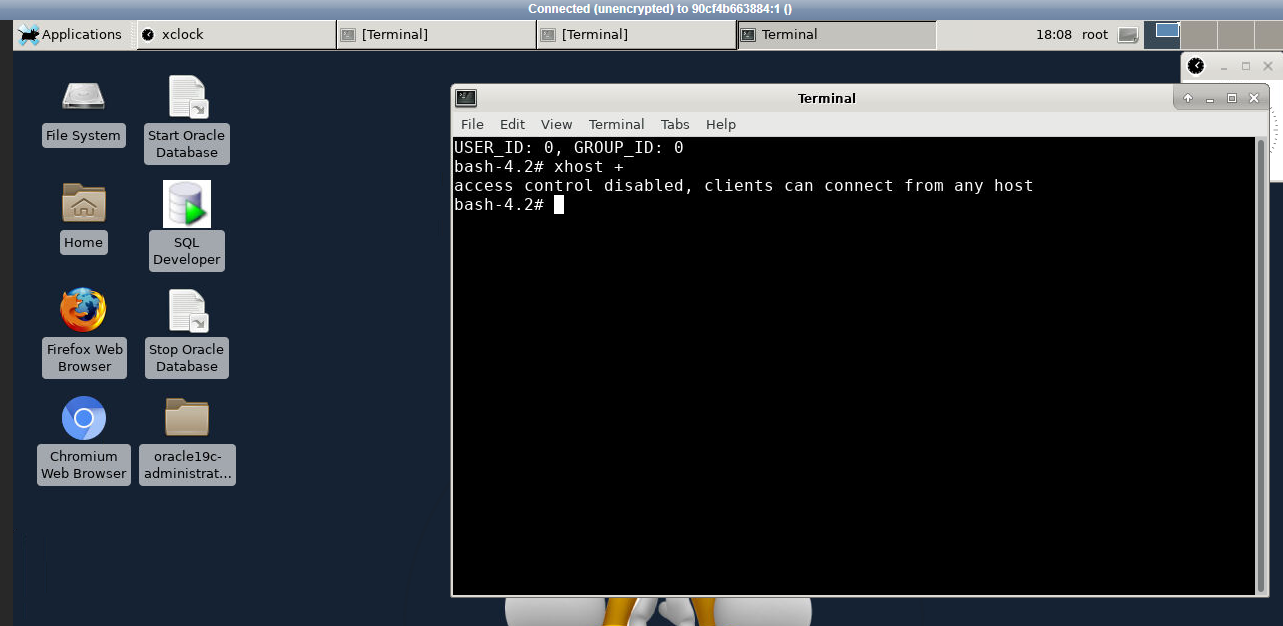
**Switch to oracle user from terminal**

### Overview

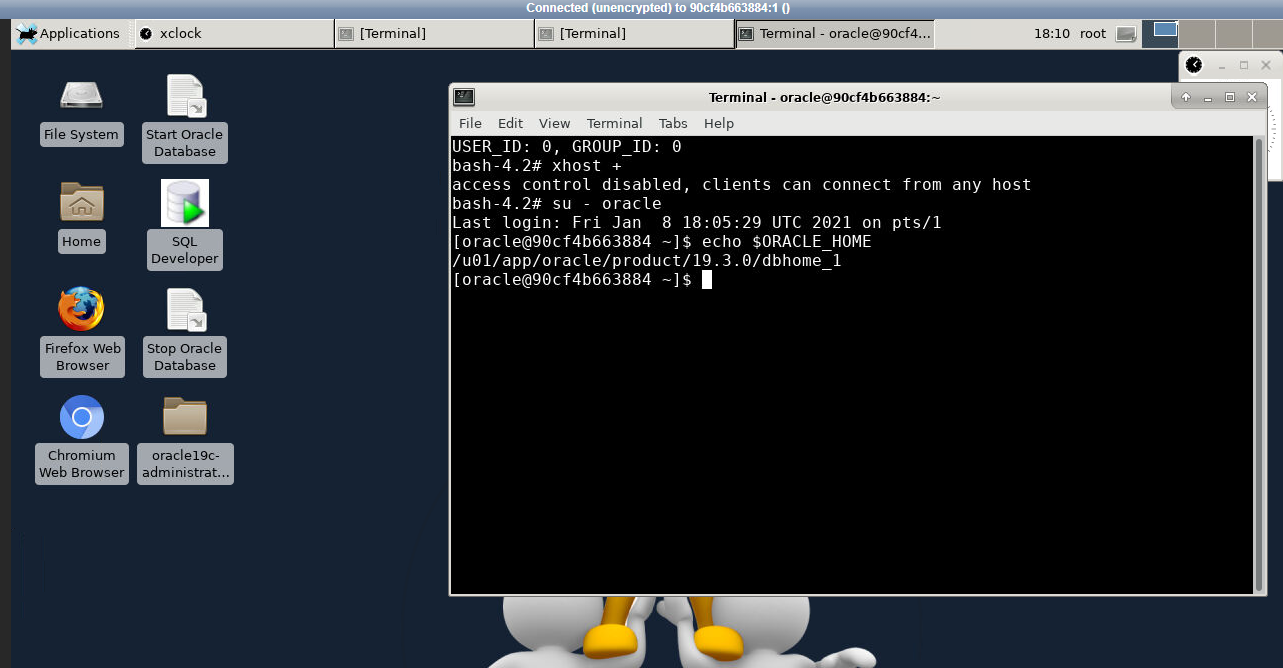
In this practice, you will switch to oracle user from terminal

### Tasks

1. Open terminal and run “xhost +” command as root user:



1. Run and run “su - oracle” command in the terminal to switch to **oracle** user:



Note: Open Install\_Database\_PDB.pdf and install database/PDB as per instructions before proceeding.

# Practices for Lesson 1: Using General Database Overall Enhancements

## Practices for Lesson 1: Overview

### Practices Overview

#### All practices are independent from one lesson to another.

Your system currently has one VM dedicated to RDBMS.

* Oracle Database 19*.*3.0.0 is already installed in

/u01/app/oracle/product/19.3.0/dbhome\_1

* The ORCL CDB with one PDB, PDB1, is precreated.
* Net service names for any of the future CDBs and PDBs that you will create through the practices are already logged in the

$ORACLE\_HOME/network/admin/tnsnames.ora file.

**To clean up your PDBs at the beginning of the practices of each lesson**, you can execute the /home/oracle/labs/admin/cleanup\_PDBs.sh shell script. The shell script drops all PDBs that may have been created by any of the practices, and finally recreates PDB1.

$ **$HOME/labs/admin/cleanup\_PDBs.sh**

…

$

In case you need to **re-create the ORCL CDB and its PDB1 PDB**, use the

/home/oracle/labs/admin/recreate\_ORCL.sh shell script.

$ **$HOME/labs/admin/recreate\_ORCL.sh**

…

$

## Practice 1-1: Discovering the Practices Environment

### Overview

In this practice, you discover the CDB and PDB that exist on your server.

### Tasks

1. Log in to the VM as the oracle UNIX user. Check the connections to the ORCL CDB and verify that the status of PDB PDB1 is open.

$ **env | grep ORA**

ORACLE\_SID=**ORCL**

ORACLE\_BASE=/u01/app/oracle ORACLE\_HOME=**/u01/app/oracle/product/19.3.0/dbhome\_1**

$

If you are working in a Database Cloud environment, the output would be the following one.

$ **env | grep ORA** ORACLE\_UNQNAME=ORCL ORACLE\_SID=**ORCL**

ORACLE\_BASE=/u01/app/oracle ORACLE\_HOSTNAME=*<your\_vm>* ORACLE\_HOME=**/u01/app/oracle/product/19.3.0/dbhome\_1**

$

$ **sqlplus / AS SYSDBA**

SQL> **SELECT name, cdb FROM V$DATABASE;**

NAME

CDB

ORCL

YES

SQL> **SHOW PDBS**

CON\_ID CON\_NAME

OPEN MODE RESTRICTED

1. PDB$SEED
2. PDB1 SQL> **EXIT**

$

READ ONLY NO

READ WRITE NO

## Practice 1-2: Installing Oracle Database 19c with Automatic root.sh

**Execution**

### Overview

In this practice, you will install the Oracle Database 19c and use the capability to have the

root.sh script automatically executed.

You can also use the Oracle By Example “Installing Oracle Database 19c with Automatic Root Scripts Execution” to see how to proceed. Launch a browser and click the bookmark from the Bookmarks toolbar of the browser. The URL is file:////home/oracle/labs/OBEs/F11836\_01/html/index.html.

### Tasks

1. On your VM, run the installer to install the Oracle Database 19c. Because the Oracle Database 19c is already installed for other lessons in

/u01/app/oracle/product/19.3.0/dbhome\_1, you will use another Oracle home directory to install.

Open TigerVNC as big as possible in order to see options at bottom of installer window.

$ **cd /u01/app/oracle/product/19.3.0/dbhome\_2**

$ **./runInstaller**

1. On the Select Configuration Option page, select Set Up Software Only.
2. On the Select Database Installation Option page, click Next.
3. On the Select Database Edition page, click Next.
4. On the Specify Installation Location page, click Next.
5. On the Privileged Operating System Groups page, click Next.
6. On the Root script execution configuration page, check “Automatically run configuration scripts”, and provide the password for the root user. Click Next.
7. In Perform Prerequisite Checks, check “Ignore All” and click Next. If there is a warning message, choose Yes.
8. On the Summary page, click the Save Response File… because you only need to observe the root scripts recorded in the response file and do not need to install.
9. Select the /tmp directory as the Location to save the db.rsp response file. Click Save and then Cancel. A message asks you if you want to exit, meaning that you do not want to install right now. Click Yes.
10. Read the /tmp/db.rsp response file.

#### $ cat /tmp/db.rsp

… ################################################################ #

#### # Root script execution configuration

# ################################################################

# # Specify the root script execution mode.

#

# - **true** : To execute the root script automatically by using the appropriate configuration methods.

# - **false** : To execute the root script manually. #

# If this option is selected, password should be specified on the console.

# oracle.install.db.**rootconfig.executeRootScript=true**

#

# Specify the configuration method to be used for automatic root script execution.

#

# Following are the possible choices:

# - ROOT

# - SUDO

# oracle.install.db.**rootconfig.configMethod=ROOT**

# # Specify the absolute path of the sudo program.

#

# Applicable only when SUDO configuration method was chosen.

# oracle.install.db.rootconfig.sudoPath=

# # Specify the name of the user who is in the sudoers list.

# Applicable only when SUDO configuration method was chosen.

# Note:For Single Instance database installations,the sudo user name must be the username of the user installing the database.

# oracle.install.db.rootconfig.sudoUserName=

…

$

## Practice 1-3: Cloning a PDB by Using DBCA in Silent Mode

### Overview

In this practice, you will clone PDB1 from ORCL as PDB19 in CDB19 by using DBCA in silent mode.

You can also use the Oracle By Example “Cloning PDBs Using DBCA in Silent Mode” to see how to proceed. Launch a browser and click the bookmark from the Bookmarks toolbar of the browser. The URL is file:///home/oracle/labs/OBEs/F11839\_01/html/index.html.

### Tasks

1. Create CDB19 with no PDBs. Execute the

/home/oracle/labs/admin/create\_CDB19.sh shell script to create CDB19.

$ **. oraenv**

ORACLE\_SID = [oracle] ? **ORCL**

The Oracle base has been set to /u01/app/oracle

$ **/home/oracle/labs/admin/create\_CDB19.sh**

…

$

1. Before proceeding to the next steps, execute the $HOME/labs/DB/glogin.sh shell script. It sets formatting for all columns selected in queries.

$ **$HOME/labs/DB/glogin.sh**

**…**

$

1. Before cloning PDB1 from ORCL as PDB19 in CDB19, verify that PDB1 contains the

HR.EMPLOYEES table.

$ **sqlplus system@PDB1**

Enter password: password

SQL> **SELECT count(\*) FROM hr.employees;**

COUNT(\*)

107

SQL> **EXIT**

$

1. Launch DBCA to clone PDB1 from ORCL as PDB19 in CDB19. Do not omit to set the right passwords in the following command.

$ **dbca -silent -createPluggableDatabase -createFromRemotePDB - remotePDBName PDB1 -remoteDBConnString ORCL - remoteDBSYSDBAUserName SYS -remoteDBSYSDBAUserPassword *password***

**-sysDBAUserName sys -sysDBAPassword *password* -dbLinkUsername c##remote\_user -dbLinkUserPassword *password* -sourceDB CDB19 - pdbName PDB19**

[FATAL] [DBT-19404] Specified database link user (C##REMOTE\_USER) does not exist in the database(ORCL).

ACTION: Specify an existing database link user.

$

*Q1/ What does the error message mean?*

##### A1/ You must create the common user in ORCL that is required for the database link that will be automatically created in CDB19.

$ **sqlplus system@ORCL**

Enter password: *password*

SQL> **CREATE USER c##remote\_user IDENTIFIED BY *password***

**CONTAINER=ALL;**

User created.

SQL> **GRANT create session TO c##remote\_user CONTAINER=ALL;**

Grant succeeded.

SQL> **EXIT**

$

$ **dbca -silent -createPluggableDatabase -createFromRemotePDB**

**-remotePDBName PDB1 -remoteDBConnString ORCL**

**-remoteDBSYSDBAUserName SYS -remoteDBSYSDBAUserPassword *password***

**-sysDBAUserName sys -sysDBAPassword *password* -dbLinkUsername c##remote\_user -dbLinkUserPassword *password* -sourceDB CDB19 - pdbName PDB19**

[FATAL] [DBT-19403] (C##REMOTE\_USER) user does not have ("CREATE

PLUGGABLE DATABASE") privilege in the remote container database.

ACTION: Specify user with ("CREATE PLUGGABLE DATABASE")

privilege in the remote container database to perform the operation.

$

*Q2/ Did you grant enough privileges to the common user used in the database link to connect to CDB19?*

##### A2/ No. You must grant the common user the CREATE PLUGGABLE DATABASE

***system privilege in ORCL.***

$ **sqlplus system@ORCL**

Enter password: *password*

SQL> **GRANT create pluggable database TO c##remote\_user CONTAINER=ALL;**

2

Grant succeeded.

SQL> **EXIT**

$

$ **dbca -silent -createPluggableDatabase -createFromRemotePDB**

**-remotePDBName PDB1 -remoteDBConnString ORCL**

**-remoteDBSYSDBAUserName SYS -remoteDBSYSDBAUserPassword *password***

**-sysDBAUserName sys -sysDBAPassword *password* -dbLinkUsername c##remote\_user -dbLinkUserPassword *password* -sourceDB CDB19 - pdbName PDB19**

Prepare for db operation 50% complete

Create pluggable database using remote clone operation 100% complete

Pluggable database "PDB19" plugged successfully.

Look at the log file "/u01/app/oracle/cfgtoollogs/dbca/CDB19/PDB19/CDB19.log" for further details.

$

1. Launch SQL\*Plus to connect to PDB19. Verify that the cloned PDB contains the

HR.EMPLOYEES table as in PDB1.

$ **sqlplus system@PDB19**

Enter password: *password*

SQL> **SELECT count(\*) FROM hr.employees;**

COUNT(\*)

107

SQL> **EXIT**

$

## Practice 1-4: Relocating a PDB by Using DBCA in Silent Mode

### Overview

In this practice, you will relocate PDB19 from CDB19 as PDB19\_IN\_ORCL in ORCL by using DBCA in silent mode.

You can also use the Oracle By Example “Relocating PDBs Using DBCA in Silent Mode” to see how to proceed. Launch a browser and click the bookmark from the Bookmarks toolbar of the browser. The URL is file:///home/oracle/labs/OBEs/F11840\_01/html/index.html.

### Tasks

1. Launch DBCA to relocate PDB19 from CDB19 as PDB19\_IN\_ORCL in ORCL. Do not omit to set the right passwords in the following command.

$ **. oraenv**

ORACLE\_SID = [oracle] ? **CDB19**

The Oracle base has been set to /u01/app/oracle

$ **dbca -silent -relocatePDB -remotePDBName PDB19 - remoteDBConnString CDB19 -sysDBAUserName SYSTEM -sysDBAPassword *password* -remoteDBSYSDBAUserName SYS -remoteDBSYSDBAUserPassword *password* -dbLinkUsername c##remote\_user -dbLinkUserPassword *password* -sourceDB ORCL -pdbName PDB19\_IN\_ORCL**

[FATAL] [DBT-19404] Specified database link user (C##REMOTE\_USER) does not exist in the database(CDB19).

ACTION: Specify an existing database link user.

$

*Q1/ What does the error message mean?*

##### A1/ You must create the common user in CDB19 that is required for the database link that will be automatically created in ORCL.

$ **sqlplus system@CDB19**

Enter password: *password*

SQL> **CREATE USER c##remote\_user IDENTIFIED BY *password***

**CONTAINER=ALL;**

2

User created.

SQL> **GRANT create session, create pluggable database TO c##remote\_user CONTAINER=ALL;**

2

Grant succeeded.

SQL> **EXIT**

$

$ **dbca -silent -relocatePDB -remotePDBName PDB19 - remoteDBConnString CDB19 -sysDBAUserName SYSTEM -sysDBAPassword *password* -remoteDBSYSDBAUserName SYS -remoteDBSYSDBAUserPassword *password* -dbLinkUsername c##remote\_user -dbLinkUserPassword *password* -sourceDB ORCL -pdbName PDB19\_IN\_ORCL**

[FATAL] [DBT-19403] (C##REMOTE\_USER) user does not have ("SYSOPER") privilege in the remote container database.

ACTION: Specify user with ("SYSOPER") privilege in the remote container database to perform the operation.

$

*Q2/ Did you grant enough privileges to the common user used in the database link to connect to CDB19?*

##### A2/ No. You must grant the common user the SYSOPER system privilege in CDB19

***for the relocation PDB operation.***

$ **sqlplus sys@CDB19 AS SYSDBA**

Enter password: *password*

SQL> **GRANT sysoper TO c##remote\_user CONTAINER=ALL;**

Grant succeeded.

SQL> **EXIT**

$

$ **dbca -silent -relocatePDB -remotePDBName PDB19 - remoteDBConnString CDB19 -sysDBAUserName SYSTEM -sysDBAPassword *password* -remoteDBSYSDBAUserName SYS -remoteDBSYSDBAUserPassword *password* -dbLinkUsername c##remote\_user -dbLinkUserPassword *password* -sourceDB ORCL -pdbName PDB19\_IN\_ORCL**

Prepare for db operation 50% complete

Create pluggable database using relocate PDB operation 100% complete

Pluggable database "PDB19\_IN\_ORCL" plugged successfully.

Look at the log file "/u01/app/oracle/cfgtoollogs/dbca/ORCL/PDB19\_IN\_ORCL/ORCL.log" for further details.

$

1. Launch SQL\*Plus to connect to PDB19\_IN\_ORCL. Verify that PDB19 is relocated as

PDB19\_IN\_ORCL in ORCL.

$ **sqlplus sys@CDB19 AS SYSDBA**

Enter password: *password*

SQL> **SHOW PDBS**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| CON\_ID CON\_NAME |  | OPEN | MODE |  | RESTRICTED |
| 2 PDB$SEED |  | READ | ONLY |  | NO |
| SQL> **EXIT**  $ | | | | | |

$ **sqlplus sys@ORCL AS SYSDBA** Enter password: *password* SQL> **SHOW PDBS**

CON\_ID CON\_NAME

OPEN MODE RESTRICTED

1. PDB$SEED
2. PDB1
3. **PDB19\_IN\_ORCL**

SQL> **EXIT**

$

READ ONLY NO

READ WRITE NO

READ WRITE NO

## Practice 1-5: Duplicating a CDB by Using DBCA in Silent Mode

### Overview

In this practice, you will duplicate ORCL as DUPORCL by using DBCA in silent mode.

You can also use the Oracle By Example “Duplicating CDBs Using DBCA in Silent Mode” to see how to proceed. Launch a browser and click the bookmark from the Bookmarks toolbar of the browser. The URL is file:///home/oracle/labs/OBEs/F11841\_01/html/index.html.

### Tasks

1. Launch DBCA to duplicate ORCL as DUPORCL by using DBCA in silent mode. Do not omit to set the right password in the following command and replace the *hostname* by your host name.

#### $ . oraenv

ORACLE\_SID = [oracle] ? **ORCL**

The Oracle base has been set to /u01/app/oracle

#### $ dbca -silent -createDuplicateDB -gdbName DUPORCL - primaryDBConnectionString *hostname*:1521/ORCL -sid DUPORCL - databaseConfigType SI -initParams db\_unique\_name=DUPORCL - sysPassword *password*

[FATAL] [DBT-06607] The data file (/u02/app/oracle/oradata/ORCL/users01.dbf) already exists on the file system.

CAUSE: This configuration is going to create the data file location (/u02/app/oracle/oradata/ORCL/users01.dbf). But it is detected that there is an existing database using this data file location or the data files from a previous configuration may be left behind.

ACTION: Clean up the existing data files, or provide a different db\_unique\_name.

[FATAL] [DBT-06607] The data file (/u02/app/oracle/oradata/ORCL/sysaux01.dbf) already exists on the file system.

CAUSE: This configuration is going to create the data file location (/u02/app/oracle/oradata/ORCL/sysaux01.dbf). But it is detected that there is an existing database using this data file location or the data files from a previous configuration may be left behind.

ACTION: Clean up the existing data files, or provide a different db\_unique\_name.

[FATAL] [DBT-06607] The data file (/u02/app/oracle/oradata/ORCL/system01.dbf) already exists on the file system.

CAUSE: This configuration is going to create the data file location (/u02/app/oracle/oradata/ORCL/system01.dbf). But it is detected that there is an existing database using this data file location or the data files from a previous configuration may be left behind.

ACTION: Clean up the existing data files, or provide a different db\_unique\_name.

[FATAL] [DBT-06607] The data file (/u02/app/oracle/oradata/ORCL/undotbs01.dbf) already exists on the file system.

CAUSE: This configuration is going to create the data file location (/u02/app/oracle/oradata/ORCL/undotbs01.dbf). But it is detected that there is an existing database using this data file location or the data files from a previous configuration may be left behind.

ACTION: Clean up the existing data files, or provide a different db\_unique\_name.

[FATAL] [DBT-06607] The data file (/u02/app/oracle/oradata/ORCL/pdbseed/sysaux01.dbf) already exists on the file system.

CAUSE: This configuration is going to create the data file location (/u02/app/oracle/oradata/ORCL/pdbseed/sysaux01.dbf).

But it is detected that there is an existing database using this data file location or the data files from a previous configuration may be left behind.

ACTION: Clean up the existing data files, or provide a different db\_unique\_name.

[FATAL] [DBT-06607] The data file (/u02/app/oracle/oradata/ORCL/pdbseed/system01.dbf) already exists on the file system.

CAUSE: This configuration is going to create the data file location (/u02/app/oracle/oradata/ORCL/pdbseed/system01.dbf).

But it is detected that there is an existing database using this data file location or the data files from a previous configuration may be left behind.

ACTION: Clean up the existing data files, or provide a different db\_unique\_name.

[FATAL] [DBT-06607] The data file (/u02/app/oracle/oradata/ORCL/pdbseed/undotbs01.dbf) already exists on the file system.

CAUSE: This configuration is going to create the data file location (/u02/app/oracle/oradata/ORCL/pdbseed/undotbs01.dbf).

But it is detected that there is an existing database using this data file location or the data files from a previous configuration may be left behind.

ACTION: Clean up the existing data files, or provide a different db\_unique\_name.

[FATAL] [DBT-06607] The data file (/u02/app/oracle/oradata/ORCL/PDB1/sysaux01.dbf) already exists on the file system.

CAUSE: This configuration is going to create the data file location (/u02/app/oracle/oradata/ORCL/PDB1/sysaux01.dbf). But it is detected that there is an existing database using this data file location or the data files from a previous configuration may be left behind.

ACTION: Clean up the existing data files, or provide a different db\_unique\_name.

[FATAL] [DBT-06607] The data file (/u02/app/oracle/oradata/ORCL/PDB1/undotbs01.dbf) already exists on the file system.

CAUSE: This configuration is going to create the data file location (/u02/app/oracle/oradata/ORCL/PDB1/undotbs01.dbf). But it is detected that there is an existing database using this data file location or the data files from a previous configuration may be left behind.

ACTION: Clean up the existing data files, or provide a different db\_unique\_name.

[FATAL] [DBT-06607] The data file (/u02/app/oracle/oradata/ORCL/PDB1/users01.dbf) already exists on the file system.

CAUSE: This configuration is going to create the data file location (/u02/app/oracle/oradata/ORCL/PDB1/users01.dbf). But it is detected that there is an existing database using this data file location or the data files from a previous configuration may be left behind.

ACTION: Clean up the existing data files, or provide a different db\_unique\_name.

[FATAL] [DBT-06607] The data file (/u02/app/oracle/oradata/ORCL/PDB1/system01.dbf) already exists on the file system.

CAUSE: This configuration is going to create the data file location (/u02/app/oracle/oradata/ORCL/PDB1/system01.dbf). But it is detected that there is an existing database using this data file location or the data files from a previous configuration may be left behind.

ACTION: Clean up the existing data files, or provide a different db\_unique\_name.

[FATAL] [DBT-06607] The data file (/u02/app/oracle/oradata/ORCL/PDB19\_IN\_ORCL/system01.dbf) already exists on the file system.

CAUSE: This configuration is going to create the data file location (/u02/app/oracle/oradata/ORCL/PDB19\_IN\_ORCL/system01.dbf). But it is detected that there is an existing database using this data file location or the data files from a previous configuration may be left behind.

ACTION: Clean up the existing data files, or provide a different db\_unique\_name.

[FATAL] [DBT-06607] The data file (/u02/app/oracle/oradata/ORCL/PDB19\_IN\_ORCL/users01.dbf) already exists on the file system.

CAUSE: This configuration is going to create the data file location (/u02/app/oracle/oradata/ORCL/PDB19\_IN\_ORCL/users01.dbf). But it is detected that there is an existing database using this data file location or the data files from a previous configuration may be left behind.

ACTION: Clean up the existing data files, or provide a different db\_unique\_name.

[FATAL] [DBT-06607] The data file (/u02/app/oracle/oradata/ORCL/PDB19\_IN\_ORCL/sysaux01.dbf) already exists on the file system.

CAUSE: This configuration is going to create the data file location (/u02/app/oracle/oradata/ORCL/PDB19\_IN\_ORCL/sysaux01.dbf). But it is detected that there is an existing database using this data file location or the data files from a previous configuration may be left behind.

ACTION: Clean up the existing data files, or provide a different db\_unique\_name.

[FATAL] [DBT-06607] The data file (/u02/app/oracle/oradata/ORCL/PDB19\_IN\_ORCL/undotbs01.dbf) already exists on the file system.

CAUSE: This configuration is going to create the data file location (/u02/app/oracle/oradata/ORCL/PDB19\_IN\_ORCL/undotbs01.dbf). But it is detected that there is an existing database using this data file location or the data files from a previous configuration may be left behind.

ACTION: Clean up the existing data files, or provide a different db\_unique\_name.

[FATAL] [DBT-06607] The data file (/u02/app/oracle/oradata/ORCL/temp01.dbf) already exists on the file system.

CAUSE: This configuration is going to create the data file location (/u02/app/oracle/oradata/ORCL/temp01.dbf). But it is detected that there is an existing database using this data file location or the data files from a previous configuration may be left behind.

ACTION: Clean up the existing data files, or provide a different db\_unique\_name.

[FATAL] [DBT-06607] The data file (/u02/app/oracle/oradata/ORCL/pdbseed/temp012019-05-07\_11-39-50- 925-AM.dbf) already exists on the file system.

CAUSE: This configuration is going to create the data file location (/u02/app/oracle/oradata/ORCL/pdbseed/temp012019-05- 07\_11-39-50-925-AM.dbf). But it is detected that there is an existing database using this data file location or the data files from a previous configuration may be left behind.

ACTION: Clean up the existing data files, or provide a different db\_unique\_name.

[FATAL] [DBT-06607] The data file (/u02/app/oracle/oradata/ORCL/PDB1/temp01.dbf) already exists on the file system.

CAUSE: This configuration is going to create the data file location (/u02/app/oracle/oradata/ORCL/PDB1/temp01.dbf). But it is detected that there is an existing database using this data file location or the data files from a previous configuration may be left behind.

ACTION: Clean up the existing data files, or provide a different db\_unique\_name.

[FATAL] [DBT-06607] The data file (/u02/app/oracle/oradata/ORCL/PDB19\_IN\_ORCL/temp01.dbf) already exists on the file system.

CAUSE: This configuration is going to create the data file location (/u02/app/oracle/oradata/ORCL/PDB19\_IN\_ORCL/temp01.dbf). But it is detected that there is an existing database using this data file location or the data files from a previous configuration may be left behind.

ACTION: Clean up the existing data files, or provide a different db\_unique\_name.

[FATAL] [DBT-06607] The data file (/u02/app/oracle/oradata/ORCL/control01.ctl) already exists on the file system.

CAUSE: This configuration is going to create the data file location (/u02/app/oracle/oradata/ORCL/control01.ctl). But it is detected that there is an existing database using this data file location or the data files from a previous configuration may be left behind.

ACTION: Clean up the existing data files, or provide a different db\_unique\_name.

[FATAL] [DBT-06607] The data file (/u02/app/oracle/oradata/ORCL/redo01.log) already exists on the file system.

CAUSE: This configuration is going to create the data file location (/u02/app/oracle/oradata/ORCL/redo01.log). But it is detected that there is an existing database using this data file location or the data files from a previous configuration may be left behind.

ACTION: Clean up the existing data files, or provide a different db\_unique\_name.

[FATAL] [DBT-06607] The data file (/u02/app/oracle/oradata/ORCL/redo02.log) already exists on the file system.

CAUSE: This configuration is going to create the data file location (/u02/app/oracle/oradata/ORCL/redo02.log). But it is detected that there is an existing database using this data file location or the data files from a previous configuration may be left behind.

ACTION: Clean up the existing data files, or provide a different db\_unique\_name.

[FATAL] [DBT-06607] The data file (/u02/app/oracle/oradata/ORCL/redo03.log) already exists on the file system.

CAUSE: This configuration is going to create the data file location (/u02/app/oracle/oradata/ORCL/redo03.log). But it is detected that there is an existing database using this data file location or the data files from a previous configuration may be left behind.

ACTION: Clean up the existing data files, or provide a different db\_unique\_name.

$

*Q1/ What does the error message exactly mean?*

##### A1/ The message means that the command considers that the datafile and controlfile destination for DUPORCL is the destination of ORCL. The operation cannot be completed under these conditions because the duplication would overwrite the source CDB. Add the appropriate clauses in the command.

$ **dbca -silent -createDuplicateDB -gdbName DUPORCL -sid DUPORCL**

**-primaryDBConnectionString *hostname*:1521/ORCL - databaseConfigType SI -initParams db\_unique\_name=DUPORCL - sysPassword *password* -datafileDestination**

**/u02/app/oracle/oradata** Prepare for db operation 22% complete

Listener config step 44% complete

Auxiliary instance creation 67% complete

RMAN duplicate 89% complete

Post duplicate database operations 100% complete

Look at the log file "/u01/app/oracle/cfgtoollogs/dbca/DUPORCL/DUPORCL.log" for further details.

$

1. Launch SQL\*Plus to connect to the CDB root of DUPORCL.

#### $ . oraenv

ORACLE\_SID = [oracle] ? **DUPORCL**

The Oracle base has been set to /u01/app/oracle

#### $ sqlplus / AS SYSDBA

SQL> **SELECT name FROM v$database;**

NAME

DUPORCL

SQL> **SHOW PDBS**

CON\_ID CON\_NAME OPEN MODE RESTRICTED

* 1. PDB$SEED READ ONLY NO
  2. PDB1 READ WRITE NO
  3. PDB19\_IN\_ORCL READ WRITE NO SQL> **SELECT name FROM v$datafile;**

NAME

/u02/app/oracle/oradata/DUPORCL/system01.dbf

/u02/app/oracle/oradata/DUPORCL/sysaux01.dbf

/u02/app/oracle/oradata/DUPORCL/undotbs01.dbf

/u02/app/oracle/oradata/DUPORCL/pdbseed/system01.dbf

/u02/app/oracle/oradata/DUPORCL/pdbseed/sysaux01.dbf

/u02/app/oracle/oradata/DUPORCL/users01.dbf

/u02/app/oracle/oradata/DUPORCL/pdbseed/undotbs01.dbf

/u02/app/oracle/oradata/DUPORCL/PDB1/system01.dbf

/u02/app/oracle/oradata/DUPORCL/PDB1/sysaux01.dbf

/u02/app/oracle/oradata/DUPORCL/PDB1/undotbs01.dbf

/u02/app/oracle/oradata/DUPORCL/PDB1/users01.dbf

/u02/app/oracle/oradata/DUPORCL/PDB19\_IN\_ORCL/system01.dbf

/u02/app/oracle/oradata/DUPORCL/PDB19\_IN\_ORCL/sysaux01.dbf

/u02/app/oracle/oradata/DUPORCL/PDB19\_IN\_ORCL/undotbs01.dbf

/u02/app/oracle/oradata/DUPORCL/PDB19\_IN\_ORCL/users01.dbf

15 rows selected.

SQL>

*Q1/ Which is the default value used with the –useOMF parameter?*

##### A1/ The default value used is FALSE. The command used in step 1 is equivalent to the following command.

dbca -silent -createDuplicateDB -gdbName DUPORCL -sid DUPORCL - primaryDBConnectionString *hostname*:1521/ORCL -databaseConfigType SI -initParams db\_unique\_name=DUPORCL -sysPassword *password* - datafileDestination /u02/app/oracle/oradata –useOMF false

1. Connect to PDB19\_IN\_ORCL. Verify that the PDB contains the HR.EMPLOYEES table as in

PDB1 of ORCL.

SQL> **CONNECT system@PDB19\_IN\_ORCL**

Enter password: *password*

Connected.

SQL> **SELECT name FROM v$database;**

NAME DUPORCL

SQL> **SELECT count(\*) FROM hr.employees;**

COUNT(\*)

107

SQL> **EXIT**

$

1. Drop the DUPORCL CDB. Execute the /home/oracle/labs/DB/drop\_DUPORCL.sh

shell script to drop DUPORCL.

$ **/home/oracle/labs/DB/drop\_DUPORCL.sh**

…

$

## Practice 1-6: Decreasing TTS Import Time

### Overview

In this practice, you will use new Data Pump parameters to decrease the transportable tablespace import time.

### Tasks

1. Execute the /home/oracle/labs/DB/create\_drop\_TBS.sh shell script. The script creates the TEST tablespace and the directory for Data Pump in PDB1 in ORCL, and drops the TEST tablespace in PDB19\_IN\_ORCL.

$ **/home/oracle/labs/DB/create\_drop\_TBS.sh**

…

$

1. You plan to transport the TEST tablespace from PDB1 into PDB19\_IN\_ORCL in ORCL and keep the transported tablespace in read-only mode after import.
   1. First export the TEST tablespace from PDB1 from ORCL with the transportable tablespace mode.
      1. Set the TEST user-defined tablespace that stores the HR.EMPLOYEES table to read- only before exporting.

$ **sqlplus system@PDB1**

Enter password: *password*

SQL> **SELECT \* FROM hr.tabtest;**

LABEL

DATA FROM system.tabtest ON TABLESPACE test SQL> **ALTER TABLESPACE test READ ONLY;**

Tablespace altered.

SQL> **EXIT**

$

* + 1. Export the TEST tablespace from PDB1 with the transportable tablespace mode.

$ **expdp \"sys@PDB1 as sysdba\" DIRECTORY=dp\_pdb1 dumpfile=PDB1.dmp TRANSPORT\_TABLESPACES=test TRANSPORT\_FULL\_CHECK=YES LOGFILE=tts.log REUSE\_DUMPFILES=YES**

**…**

Password: *password*

Starting "SYS"."SYS\_EXPORT\_TRANSPORTABLE\_01":

"sys/\*\*\*\*\*\*\*\*@PDB1 AS SYSDBA" DIRECTORY=dp\_pdb1 dumpfile=PDB1.dmp TRANSPORT\_TABLESPACES=test TRANSPORT\_FULL\_CHECK=YES LOGFILE=tts.log REUSE\_DUMPFILES=YES

Processing object type TRANSPORTABLE\_EXPORT/STATISTICS/TABLE\_STATISTICS

Processing object type TRANSPORTABLE\_EXPORT/STATISTICS/MARKER

Processing object type TRANSPORTABLE\_EXPORT/PLUGTS\_BLK

Processing object type TRANSPORTABLE\_EXPORT/POST\_INSTANCE/PLUGTS\_BLK

Processing object type TRANSPORTABLE\_EXPORT/TABLE

Master table "SYS"."SYS\_EXPORT\_TRANSPORTABLE\_01" successfully loaded/unloaded

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* Dump file set for SYS.SYS\_EXPORT\_TRANSPORTABLE\_01 is:

**/tmp/PDB1.dmp**

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* Datafiles required for transportable tablespace TEST:

**/u02/app/oracle/oradata/ORCL/pdb1/test01.dbf**

Job "SYS"."SYS\_EXPORT\_TRANSPORTABLE\_01" successfully completed at Thu Sep 6 16:51:49 2018 elapsed 0 00:00:43

$

1. Create the directory for Data Pump in PDB19\_IN\_ORCL in ORCL.

$ **sqlplus system@PDB19\_IN\_ORCL**

Enter password: *password*

SQL> **CREATE DIRECTORY dp\_pdb19\_in\_orcl AS '/tmp';**

Directory created. SQL>

1. Verify that the tablespace TEST does not exist in PDB19\_IN\_ORCL.

SQL> **SELECT tablespace\_name FROM dba\_tablespaces;**

TABLESPACE\_NAME SYSTEM

SYSAUX UNDOTBS1 TEMP USERS

SQL>

If the tablespace already exists, drop it.

SQL> **DROP TABLESPACE test INCLUDING CONTENTS AND DATAFILES;**

DROP TABLESPACE test INCLUDING CONTENTS AND DATAFILES

\*

ERROR at line 1:

ORA-00959: tablespace 'TEST' does not exist SQL> **EXIT**

$

1. Copy the datafiles of the TEST tablespace of PDB1 of ORCL to the target directory of

PDB19\_IN\_ORCL.

$ **cp /u02/app/oracle/oradata/ORCL/pdb1/test01.dbf**

**/u02/app/oracle/oradata/ORCL/PDB19\_IN\_ORCL**

$

1. Import the PDB1 TEST tablespace in PDB19\_IN\_ORCL in ORCL and keep the imported tablespace in read-only mode.

$ **impdp \'sys@PDB19\_IN\_ORCL as sysdba\' DIRECTORY=dp\_pdb19\_in\_orcl DUMPFILE=PDB1.dmp TRANSPORT\_DATAFILES='/u02/app/oracle/oradata/ORCL/PDB19\_IN\_ORCL/ test01.dbf' TRANSPORTABLE=KEEP\_READ\_ONLY**

**…**

Password: *password*

Master table "SYS"."SYS\_IMPORT\_TRANSPORTABLE\_01" successfully loaded/unloaded

Starting "SYS"."SYS\_IMPORT\_TRANSPORTABLE\_01":

"sys/\*\*\*\*\*\*\*\*@PDB19\_IN\_ORCL AS SYSDBA" DIRECTORY=dp\_pdb19\_in\_orcl DUMPFILE=PDB1.dmp TRANSPORT\_DATAFILES=/u02/app/oracle/oradata/ORCL/PDB19\_IN\_ORCL/t est01.dbf TRANSPORTABLE=KEEP\_READ\_ONLY

Processing object type TRANSPORTABLE\_EXPORT/PLUGTS\_BLK

Processing object type TRANSPORTABLE\_EXPORT/TABLE

Processing object type TRANSPORTABLE\_EXPORT/STATISTICS/TABLE\_STATISTICS

Processing object type TRANSPORTABLE\_EXPORT/STATISTICS/MARKER

Processing object type TRANSPORTABLE\_EXPORT/POST\_INSTANCE/PLUGTS\_BLK

Job "SYS"."SYS\_IMPORT\_TRANSPORTABLE\_01" successfully completed at Thu Sep 6 17:06:15 2018 elapsed 0 00:00:27

$

1. Verify that PDB19\_IN\_ORCL is still in read-only mode after import. For a huge tablespace import, the KEEP\_READ\_ONLY parameter can decrease the time spent.

$ **sqlplus system@PDB19\_IN\_ORCL**

Enter password: *password*

SQL> **SELECT status FROM dba\_tablespaces WHERE tablespace\_name='TEST';**

2

STATUS

READ ONLY SQL> **EXIT**

$

1. Execute the /home/oracle/labs/DB/drop\_TBS.sh shell script to drop the tablespace imported in PDB19\_IN\_ORCL.

$ **/home/oracle/labs/DB/drop\_TBS.sh**

…

$

1. You now plan to transport the TEST tablespace from PDB1 into PDB19\_IN\_ORCL. After import, the bitmaps in the datafile are not rebuilt but the tablespace can be set to read/write. This type of operation decreases the time at import time. As you already exported the tablespace in the previous steps, you can reuse the /tmp/PDB1.dmp dump file.
   1. Copy the datafiles of the TEST tablespace of PDB1 of ORCL to the target directory of

PDB19\_IN\_ORCL.

$ **cp /u02/app/oracle/oradata/ORCL/pdb1/test01.dbf**

**/u02/app/oracle/oradata/ORCL/PDB19\_IN\_ORCL**

$

* 1. Import the TEST tablespace from PDB1 into PDB19\_IN\_ORCL with no bitmap rebuild.

$ **impdp \'sys@PDB19\_IN\_ORCL as sysdba\' DIRECTORY=dp\_pdb19\_in\_orcl DUMPFILE=PDB1.dmp TRANSPORT\_DATAFILES='/u02/app/oracle/oradata/ORCL/PDB19\_IN\_ORCL/ test01.dbf' TRANSPORTABLE=NO\_BITMAP\_REBUILD**

**…**

Password: *password*

Master table "SYS"."SYS\_IMPORT\_TRANSPORTABLE\_01" successfully loaded/unloaded

Starting "SYS"."SYS\_IMPORT\_TRANSPORTABLE\_01":

"sys/\*\*\*\*\*\*\*\*@PDB19\_IN\_ORCL AS SYSDBA" DIRECTORY=dp\_pdb19\_in\_orcl DUMPFILE=PDB1.dmp TRANSPORT\_DATAFILES=/u02/app/oracle/oradata/ORCL/PDB19\_IN\_ORCL/t est01.dbf TRANSPORTABLE=NO\_BITMAP\_REBUILD

Processing object type TRANSPORTABLE\_EXPORT/PLUGTS\_BLK

Processing object type TRANSPORTABLE\_EXPORT/TABLE

Processing object type TRANSPORTABLE\_EXPORT/STATISTICS/TABLE\_STATISTICS

Processing object type TRANSPORTABLE\_EXPORT/STATISTICS/MARKER

Processing object type TRANSPORTABLE\_EXPORT/POST\_INSTANCE/PLUGTS\_BLK

Job "SYS"."SYS\_IMPORT\_TRANSPORTABLE\_01" successfully completed at Thu Sep 6 17:09:54 2018 elapsed 0 00:00:25

$

1. Verify that PDB19\_IN\_ORCL is in read-only mode after import. For a huge tablespace import, the NO\_BITMAP\_REBUILD parameter can decrease the time spent.

$ **sqlplus sys@PDB19\_IN\_ORCL AS SYSDBA**

Enter password: *password*

SQL> **SELECT status FROM dba\_tablespaces WHERE tablespace\_name='TEST';**

2

STATUS READ ONLY

SQL>

*Q1/ Can you set the tablespace to read write although the bitmaps are not rebuilt?*

SQL> **ALTER TABLESPACE test READ WRITE;**

Tablespace altered. SQL>

***A1/ Yes, the tablespace can be set to read/write. The bitmaps can be rebuilt later with the DBMS\_SPACE\_ADMIN.TABLESPACE\_REBUILD\_BITMAPS procedure.***

SQL> **exec DBMS\_SPACE\_ADMIN.TABLESPACE\_REBUILD\_BITMAPS('TEST')**

PL/SQL procedure successfully completed.

SQL> **EXIT**

$

## Practice 1-7: Decreasing TTS Export Time

### Overview

In this practice, you will use new Data Pump parameters to decrease the transportable tablespace export time.

### Tasks

1. Execute the /home/oracle/labs/DB/create\_drop\_TBS.sh shell script. The script creates the TEST tablespace and the directory for Data Pump in PDB1 in ORCL, and drops the TEST tablespace in PDB19\_IN\_ORCL.

$ **/home/oracle/labs/DB/create\_drop\_TBS.sh**

…

$

1. You plan to transport the TEST tablespace from PDB1 into PDB19\_IN\_ORCL. Because the time it takes to conduct the closure check can be long, the closure check can be unnecessary when the DBA knows that the transportable set is self-contained. Skipping the closure check allows the tablespaces to remain read/write. First, determine the length of time that tablespace files are required to be read-only during transportable operations. Running the data pump transportable operation with the TTS\_CLOSURE\_CHECK parameter in TEST\_MODE mode provides timing estimation of the TTS export operation.

$ **expdp \"sys@PDB1 as sysdba\" DIRECTORY= dp\_pdb1 dumpfile=PDB1.dmp TRANSPORT\_TABLESPACES=test TRANSPORT\_FULL\_CHECK=YES TTS\_CLOSURE\_CHECK=TEST\_MODE LOGFILE=tts.log REUSE\_DUMPFILES=YES**

**…**

Password: *password*

Starting "SYS"."SYS\_EXPORT\_TRANSPORTABLE\_01":

"sys/\*\*\*\*\*\*\*\*@PDB1 AS SYSDBA" DIRECTORY=dp\_pdb1 dumpfile=PDB1.dmp TRANSPORT\_TABLESPACES=test TRANSPORT\_FULL\_CHECK=YES TTS\_CLOSURE\_CHECK=TEST\_MODE LOGFILE=tts.log REUSE\_DUMPFILES=YES

Processing object type TRANSPORTABLE\_EXPORT/STATISTICS/TABLE\_STATISTICS

Processing object type TRANSPORTABLE\_EXPORT/STATISTICS/MARKER Processing object type TRANSPORTABLE\_EXPORT/PLUGTS\_BLK

Processing object type TRANSPORTABLE\_EXPORT/POST\_INSTANCE/PLUGTS\_BLK

Processing object type TRANSPORTABLE\_EXPORT/TABLE

Master table "SYS"."SYS\_EXPORT\_TRANSPORTABLE\_01" successfully loaded/unloaded

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* Dump file set for SYS.SYS\_EXPORT\_TRANSPORTABLE\_01 is:

**/tmp/PDB1.dmp**

**Dump file set is unusable. TEST\_MODE requested.**

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* Datafiles required for transportable tablespace TEST:

**/u02/app/oracle/oradata/ORCL/pdb1/test01.dbf**

Job "SYS"."SYS\_EXPORT\_TRANSPORTABLE\_01" successfully completed at Mon Sep 10 09:34:05 2018 elapsed 0 00:00:37

$

*Q1/ Can you use the dump file to import the TEST tablespace into PDB19\_IN\_ORCL?*

$ **impdp \"sys@PDB19\_IN\_ORCL as sysdba\" DIRECTORY=dp\_pdb19\_in\_orcl dumpfile=PDB1.dmp TRANSPORT\_DATAFILES='/u02/app/oracle/oradata/ORCL/PDB19\_IN\_ORCL/ test01.dbf' LOGFILE=tts.log**

**…**

Password: *password*

ORA-39001: invalid argument value

ORA-39000: bad dump file specification

ORA-39398: **Cannot load data. Data Pump dump file "/tmp/PDB1.dmp" was created in TEST\_MODE.**

$

##### A1/ The resulting export dump file is not available for use by Data Pump import.

1. The timing estimation leads you to complete the data pump export transportable operation with the possibility to decrease the time required for Data Pump TTS to complete with the TTS\_CLOSURE\_CHECK parameter set to OFF. Of course, you are sure that the transportable tablespace set is contained.

$ **expdp \"sys@PDB1 as sysdba\" DIRECTORY= dp\_pdb1 dumpfile=PDB1.dmp TRANSPORT\_TABLESPACES=test TTS\_CLOSURE\_CHECK=OFF LOGFILE=tts.log REUSE\_DUMPFILES=YES**

**…**

Password: *password*

Starting "SYS"."SYS\_EXPORT\_TRANSPORTABLE\_01":

"sys/\*\*\*\*\*\*\*\*@PDB1 AS SYSDBA" DIRECTORY=dp\_pdb1 dumpfile=PDB1.dmp TRANSPORT\_TABLESPACES=test TTS\_CLOSURE\_CHECK=OFF LOGFILE=tts.log REUSE\_DUMPFILES=YES

Processing object type TRANSPORTABLE\_EXPORT/STATISTICS/TABLE\_STATISTICS

Processing object type TRANSPORTABLE\_EXPORT/STATISTICS/MARKER ORA-39123: Data Pump transportable tablespace job aborted

ORA-39185: The transportable tablespace failure list is

ORA-29335: **tablespace 'TEST' is not read only**

Job "SYS"."SYS\_EXPORT\_TRANSPORTABLE\_01" stopped due to fatal error at Mon Sep 10 10:14:22 2018 elapsed 0 00:00:03

$

*Q1/ Does the TTS\_CLOSURE\_CHECK parameter set to another value than the*

*TEST\_MODE allow you to export the tablespace in read/write mode?*

##### A1/ No. Only the TEST\_MODE value allows you to test the export operation timing. If you use other values such as ON, OFF and FULL, the tablespace needs to be set to read-only.

* 1. Set the TEST user-defined tablespace that stores the HR.EMPLOYEES table to read- only before exporting.

$ **sqlplus system@PDB1**

Enter password: *password*

SQL> **SELECT \* FROM hr.tabtest;**

LABEL

DATA FROM system.tabtest ON TABLESPACE test SQL> **ALTER TABLESPACE test READ ONLY;**

Tablespace altered.

SQL> **EXIT**

$

* 1. Export the tablespace.

$ **expdp \"sys@PDB1 as sysdba\" DIRECTORY= dp\_pdb1 dumpfile=PDB1.dmp TRANSPORT\_TABLESPACES=test TTS\_CLOSURE\_CHECK=OFF LOGFILE=tts.log REUSE\_DUMPFILES=YES**

…

Password: *password*

Starting "SYS"."SYS\_EXPORT\_TRANSPORTABLE\_01":

"sys/\*\*\*\*\*\*\*\*@PDB1 AS SYSDBA" DIRECTORY=dp\_pdb1 dumpfile=PDB1.dmp TRANSPORT\_TABLESPACES=test TTS\_CLOSURE\_CHECK=OFF LOGFILE=tts.log REUSE\_DUMPFILES=YES

Processing object type TRANSPORTABLE\_EXPORT/STATISTICS/TABLE\_STATISTICS

Processing object type TRANSPORTABLE\_EXPORT/STATISTICS/MARKER Processing object type TRANSPORTABLE\_EXPORT/PLUGTS\_BLK

Processing object type TRANSPORTABLE\_EXPORT/POST\_INSTANCE/PLUGTS\_BLK

Processing object type TRANSPORTABLE\_EXPORT/TABLE

Master table "SYS"."SYS\_EXPORT\_TRANSPORTABLE\_01" successfully loaded/unloaded

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* Dump file set for SYS.SYS\_EXPORT\_TRANSPORTABLE\_01 is:

**/tmp/PDB1.dmp**

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* Datafiles required for transportable tablespace TEST:

**/u02/app/oracle/oradata/ORCL/pdb1/test01.dbf**

Job "SYS"."SYS\_EXPORT\_TRANSPORTABLE\_01" successfully completed at Mon Sep 10 10:26:47 2018 elapsed 0 00:00:11

$

1. Copy the datafiles of the TEST tablespace of PDB1 of ORCL to the target directory of

PDB19\_IN\_ORCL.

$ **cp /u02/app/oracle/oradata/ORCL/pdb1/test01.dbf**

**/u02/app/oracle/oradata/ORCL/PDB19\_IN\_ORCL**

$

1. Import the PDB1 TEST tablespace in PDB19\_IN\_ORCL.

$ **impdp \'sys@PDB19\_IN\_ORCL as sysdba\' DIRECTORY=dp\_pdb19\_in\_orcl DUMPFILE=PDB1.dmp TRANSPORT\_DATAFILES='/u02/app/oracle/oradata/ORCL/PDB19\_IN\_ORCL/ test01.dbf'**

**…**

Password: *password*

Master table "SYS"."SYS\_IMPORT\_TRANSPORTABLE\_01" successfully loaded/unloaded

Starting "SYS"."SYS\_IMPORT\_TRANSPORTABLE\_01":

"sys/\*\*\*\*\*\*\*\*@PDB19\_IN\_ORCL AS SYSDBA" DIRECTORY=dp\_pdb19\_in\_orcl DUMPFILE=PDB1.dmp TRANSPORT\_DATAFILES=/u02/app/oracle/oradata/ORCL/PDB19\_IN\_ORCL/t est01.dbf

Processing object type TRANSPORTABLE\_EXPORT/PLUGTS\_BLK Processing object type TRANSPORTABLE\_EXPORT/TABLE

Processing object type TRANSPORTABLE\_EXPORT/STATISTICS/TABLE\_STATISTICS

Processing object type TRANSPORTABLE\_EXPORT/STATISTICS/MARKER

Processing object type TRANSPORTABLE\_EXPORT/POST\_INSTANCE/PLUGTS\_BLK

Job "SYS"."SYS\_IMPORT\_TRANSPORTABLE\_01" successfully completed at Mon Sep 10 10:34:49 2018 elapsed 0 00:00:24

$

1. Verify that PDB19\_IN\_ORCL is in read-only mode after import. For a huge tablespace to export, the TTS\_CLOSURE\_CHECK parameter can decrease the time spent during the export operation.

$ **sqlplus system@PDB19\_IN\_ORCL**

Enter password: *password*

SQL> **SELECT status FROM dba\_tablespaces WHERE tablespace\_name='TEST';**

2

STATUS

READ ONLY SQL> **EXIT**

$

## Practice 1-8: Omitting the Column Encryption Attribute During Import

### Overview

In this practice, you will export tables with encrypted columns to import them in a database that does not support the encrypted column feature as this is the case in the Oracle Public Cloud environment. In the Oracle Public Cloud environment, data is encrypted by default using TDE and the encrypted tablespace feature, and not the encrypted column feature.

### Tasks

1. Create the table HR.TABENC with an encrypted column in PDB1 of ORCL by executing the

/home/oracle/labs/DB/create\_TABENC.sh shell script. Ignore any error at the end of the script like ORA-46665: master keys not activated for all PDBs during REKEY.

$ **/home/oracle/labs/DB/create\_TABENC.sh**

…

$

1. Verify that the HR.TABENC table in PDB1 PDB in ORCL has an encrypted column.

$ **. oraenv**

ORACLE\_SID = [ORCL] ? **ORCL**

The Oracle base remains unchanged with value /u01/app/oracle

$ **sqlplus system@PDB1** Enter password: *password* SQL> **DESC hr.tabenc**

Name Null? Type

C1

LABEL

NUMBER

VARCHAR2(50) **ENCRYPT**

SQL> **SELECT \* FROM hr.tabenc;**

C1 LABEL

1 DATA encrypted with column TDE SQL> **EXIT**

$

1. Export the HR.TABENC table from PDB1 in ORCL.

$ **expdp system@PDB1 DIRECTORY=dp\_pdb1 dumpfile=PDB1\_TAB.dmp TABLES=hr.tabenc LOGFILE=tts.log REUSE\_DUMPFILES=YES**

…

Password: *password*

Starting "SYSTEM"."SYS\_EXPORT\_TABLE\_01": system/\*\*\*\*\*\*\*\*@PDB1 DIRECTORY=dp\_pdb1 dumpfile=PDB1\_TAB.dmp TABLES=hr.tabenc LOGFILE=tts.log REUSE\_DUMPFILES=YES

Processing object type TABLE\_EXPORT/TABLE/TABLE\_DATA

Processing object type TABLE\_EXPORT/TABLE/STATISTICS/TABLE\_STATISTICS

Processing object type TABLE\_EXPORT/TABLE/STATISTICS/MARKER Processing object type TABLE\_EXPORT/TABLE/TABLE

. . exported "HR"."TABENC" 5.5 KB 1 rows

**ORA-39173: Encrypted data has been stored unencrypted in dump file set.**

Master table "SYSTEM"."SYS\_EXPORT\_TABLE\_01" successfully loaded/unloaded

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* Dump file set for SYSTEM.SYS\_EXPORT\_TABLE\_01 is:

/tmp/PDB1\_TAB.dmp

Job "SYSTEM"."SYS\_EXPORT\_TABLE\_01" successfully completed at Thu Oct 18 08:55:42 2018 elapsed 0 00:00:27

$

1. Execute the /home/oracle/labs/DB/setup\_CDB19\_TDE\_TBS.sh shell script. The script creates the PDB19 PDB in CDB19, the HR user, the dp\_pdb19 directory, and the TEST encrypted tablespace.

If the script encounters the “ORA-28374: typed master key not found in wallet” error during the TEST tablespace creation, first execute the

/home/oracle/labs/admin/create\_CDB19.sh shell script, then the

/home/oracle/labs/DB/setup\_CDB19\_TDE.sh shell script, and finally retry the

/home/oracle/labs/DB/setup\_CDB19\_TDE\_TBS.sh shell script.

$ **/home/oracle/labs/DB/setup\_CDB19\_TDE\_TBS.sh**

…

$

1. Verify that the TEST tablespace in PDB19 PDB in CDB19 is encrypted.

$ **sqlplus system@PDB19**

Enter password: *password*

SQL> **SELECT encrypted, tablespace\_name FROM dba\_tablespaces;**

ENC TABLESPACE\_NAME NO SYSTEM

NO SYSAUX NO UNDOTBS1 NO TEMP

**YES TEST**

SQL> **EXIT**

$

1. Use the Data Pump parameter to suppress the encryption clause associated with the

HR.TABENC table creation during the import operation into PDB19.

#### $ impdp system@PDB19 DIRECTORY=dp\_pdb19 dumpfile=PDB1\_TAB.dmp LOGFILE=tts.log TRANSFORM=OMIT\_ENCRYPTION\_CLAUSE:Y TABLE\_EXISTS\_ACTION=REPLACE

…

Password: *password*

Master table "SYSTEM"."SYS\_IMPORT\_FULL\_01" successfully loaded/unloaded

Starting "SYSTEM"."SYS\_IMPORT\_FULL\_01": system/\*\*\*\*\*\*\*\*@PDB19 DIRECTORY=dp\_pdb19 dumpfile=PDB1\_TAB.dmp LOGFILE=tts.log TRANSFORM=OMIT\_ENCRYPTION\_CLAUSE:Y TABLE\_EXISTS\_ACTION=REPLACE

Processing object type TABLE\_EXPORT/TABLE/TABLE

ORA-39083: Object type TABLE:"HR"."TABENC" failed to create with error:

ORA-01950: **no privileges on tablespace 'TEST'**

Failing sql is:

#### CREATE TABLE "HR"."TABENC" ("C1" NUMBER, "LABEL" VARCHAR2(50 BYTE)) SEGMENT CREATION IMMEDIATE PCTFREE 10 PCTUSED 40 INITRANS

**1 MAXTRANS 255 NOCOMPRESS LOGGING STORAGE(INITIAL 65536 NEXT 1048576 MINEXTENTS 1 MAXEXTENTS 2147483645 PCTINCREASE 0 FREELISTS 1 FREELIST GROUPS 1 BUFFER\_POOL DEFAULT FLASH\_CACHE DEFAULT CELL\_FLASH\_CACHE DEFAULT) TABLESPACE "TEST"**

Processing object type TABLE\_EXPORT/TABLE/TABLE\_DATA

Processing object type TABLE\_EXPORT/TABLE/STATISTICS/TABLE\_STATISTICS

Processing object type TABLE\_EXPORT/TABLE/STATISTICS/MARKER

Job "SYSTEM"."SYS\_IMPORT\_FULL\_01" completed with 1 error(s) at Thu Oct 18 09:05:01 2018 elapsed 0 00:00:06

$

The missing privilege has not been granted on purpose in order to display the CREATE TABLE statement that fails during the import operation. Observe that the ENCRYPT clause is omitted in the column description for LABEL.

1. Grant the UNLIMITED TABLESPACE privilege to HR in PDB19.

$ **sqlplus system@PDB19**

Enter password: *password*

SQL> **GRANT unlimited TABLESPACE TO hr;**

Grant succeeded.

SQL> **EXIT**

$

1. Import the table.

$ **impdp system@PDB19 DIRECTORY=dp\_pdb19 dumpfile=PDB1\_TAB.dmp LOGFILE=tts.log TRANSFORM=OMIT\_ENCRYPTION\_CLAUSE:Y TABLE\_EXISTS\_ACTION=REPLACE**

…

Password: *password*

Master table "SYSTEM"."SYS\_IMPORT\_FULL\_01" successfully loaded/unloaded

Starting "SYSTEM"."SYS\_IMPORT\_FULL\_01": system/\*\*\*\*\*\*\*\*@PDB19 DIRECTORY=dp\_pdb19 dumpfile=PDB1\_TAB.dmp LOGFILE=tts.log TRANSFORM=OMIT\_ENCRYPTION\_CLAUSE:Y TABLE\_EXISTS\_ACTION=REPLACE

Processing object type TABLE\_EXPORT/TABLE/TABLE Processing object type TABLE\_EXPORT/TABLE/TABLE\_DATA

. . imported "HR"."TABENC" 5.5 KB 1 rows

Processing object type TABLE\_EXPORT/TABLE/STATISTICS/TABLE\_STATISTICS

Processing object type TABLE\_EXPORT/TABLE/STATISTICS/MARKER

Job "SYSTEM"."SYS\_IMPORT\_FULL\_01" successfully completed at Thu Oct 18 09:08:26 2018 elapsed 0 00:00:28

$

1. Verify that the HR.TABENC table does not hold the column encryption attribute.

$ **sqlplus system@PDB19** Enter password: *password* SQL> **DESC hr.tabenc**

Name

Null?

Type

C1 LABEL

SQL> **SELECT \* FROM hr.tabenc;**

NUMBER

VARCHAR2(50)

C1 LABEL

1 DATA encrypted with column TDE SQL> **EXIT**

$

*Q1/ What happens if on a Cloud environment, you do not use the Data Pump parameter that suppresses the encryption clause associated with the table creation during the import operation?*

*A****1/ The import operation fails because the Cloud environment does not support the encrypted column feature.***

## Practice 1-9: Avoiding Errors Due to Values Generated by LISTAGG

### Overview

In this practice, you will use new clauses of the LISTAGG function in order to avoid ORA- 01489: result of string concatenation is too long when the combined length of values generated exceeds the maximum length supported by VARCHAR2.

### Tasks

1. Before starting the practice, execute the

/home/oracle/labs/admin/cleanup\_PDBs.sh shell script. The shell script drops all PDBs that may have been created by any of the practices, removes TDE usage and finally recreates PDB1.

$ **$HOME/labs/admin/cleanup\_PDBs.sh**

…

$

1. Execute the /home/oracle/labs/DB/create\_CITIES.sh shell script to create the

HR.CITIES table.

$ **/home/oracle/labs/DB/create\_CITIES.sh**

…

$

1. Query the HR.CITIES table to order the rows by city code and use the LISTAGG function to concatenate the resulting city names for each city code into a single string.

$ **sqlplus system@PDB1** Enter password: *password* SQL> **SELECT code,**

**LISTAGG(name, ',') WITHIN GROUP (ORDER BY code)**

**FROM hr.cities GROUP BY code ORDER BY code;**

2 3 4 5 ERROR:

ORA-01489: result of string concatenation is too long no rows selected

SQL>

*Q1/ Which is the new clause for the LISTAGG function that truncates the string to fit within the limit of the VARCHAR2 object?*

SQL> **SELECT code,**

#### LISTAGG (name, ',' ON OVERFLOW TRUNCATE '...'

**WITHOUT COUNT) WITHIN GROUP (ORDER BY code) AS CITIES**

#### FROM hr.cities GROUP BY code ORDER BY code;

2 3 4 5 6 7

CODE

CITIES

0

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| City1  10 | ,City1 | ,Mexico | ,Mexico |  |
| Aix | ,Aix | ,City10 | ,City10 | ,City |
| 11 | ,City11 | ,City12 | ,City12 | ,City13 |

,City13 ,City14 ,City14 ,City15

,City15 ,City16 ,City16 ,City17 ,City17

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| ,City18 | | | ,City18 | ,City19 | ,City19 |
| ,City2 | | ,City2 | ,City20 | ,City20 | ,City |
| 21 | ,City21 | | ,City22 | ,City22 | ,City23 |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| ity25 | ,City23 | ,City24  ,City26 | ,City24  ,City26 | ,City25  ,City3 | ,C  ,City3 |
| ity6 ijon | ,City4 ,City4  ,City6  ,City8 ,City9  ,Lyon | | ,City5  ,City7  ,City9  ,Lyon | ,City5  ,City7  ,Dijon  ,Marseille | ,C  ,City8  ,D  ,Marseil |
| le | ,Paris ,Paris | |  |  |  |

20

City27 ,City28 ,City29 ,City30 ,City

31 ,City32 ,City33 ,City34 ,City35

,Dallas ,Houston ,LA ,NY

,Nashville ,Philadelphia ,SF ,Toronto ,Was hington

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 30 |  | | | |
| Bogota | ,Bogota10 | ,Bogota11 | ,Bogota12 | ,Bogo |
| ta13 | ,Bogota14 | ,Bogota15 | ,Bogota16 | ,Bogota17 |

,Bogota18 ,Bogota19 ,Bogota20 ,Bogota21

,Bogota22 ,Bogota23 ,Bogota24 ,Bogota25 ,Bo

gota26 ,Bogota27 ,Bogota28 ,Bogota29 ,Bogota

30 ,Bogota31 ,Bogota32 ,Bogota33 ,Bogota34

,Bogota35 ,Bogota36 ,Bogota37 ,Bogota38 ,B ogota39 ,Bogota39 ,Bogota40 ,Bogota41 ,Bogota42

,Bogota43 ,Bogota44 ,Bogota45 ,Bogota46

,Bogota47 ,Bogota48 ,Bogota49 ,Bogota50 ,Bogot a51 ,Bogota52 ,Bogota8 ,Bogota9 ,Brasilia

,Brasilia10 ,Brasilia11 ,Brasilia12 ,Brasilia13 , Brasilia14 ,Brasilia15 ,Brasilia16 ,Brasilia17 ,Bras ilia18 ,Brasilia19 ,Brasilia20 ,Brasilia21 ,Brasilia2

2 ,Brasilia23 ,Brasilia24 ,Brasilia25 ,Brasilia26

,Brasilia27 ,Brasilia28 ,Brasilia29 ,Brasilia30 ,Br asilia31 ,Brasilia32 ,Brasilia33 ,Brasilia34 ,Brasil ia35 ,Brasilia36 ,Brasilia37 ,Brasilia8 ,Brasilia9

,Buenos Aires ,Buenos Aires10,Buenos Aires11,Buenos Aires12,Buenos Aires13,Buenos Aires14,Buenos Aires15,Buenos Aires16,Buenos Aires17,Buenos Aires18,Buenos Aires19,Buenos Aires20,Buenos Aires21,Buenos Aires22,Buenos Aires23,Buenos Aires24,Buenos Aires25,Buenos Aires26,Buenos Aires27,Buenos Aires28,Buenos Aires29,Buenos Aires30,Buenos Aires31,Buenos Aires32,Buenos Aires33,Buenos Aires34,Buenos Aires35,Buenos Aires36,Buenos Aires37,Buenos Aires38,Buenos Aires39,Buenos Aires39,Buenos Aires40,Buenos Aires41,Buenos Aires42,Buenos Aires43,Buenos Aires44,Buenos Aires45,Buenos Aires46,Buenos Aires47,Buenos Aires48,Buenos Aires49,Buenos Aires50,Buenos Aires51,Buenos Aires52,Buenos Aires8 ,Buenos

Aires9 ,City36 ,City37 ,City38 ,City39

,City40 ,City41 ,City42 ,City43 ,Cit y44 ,City45 ,City46 ,City47 ,City48

,City49 ,City50 ,City51 ,City52

,City53 ,City54 ,City55 ,City56 ,Ci ty57 ,City58 ,City59 ,City60 ,City61

,City62 ,City63 ,City64 ,City65

,City66 ,City67 ,City68 ,City69 ,City70

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| ,City71 | ,City72 | ,City73 | ,City74 | |
| ,City75  ,City79 | ,City76  ,City80 | ,City77  ,City81 | ,City78  ,City82 | , |

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| City83  87 ,City88 Paz10 | ,City84  ,City89  ,La Paz11 | | ,La | ,City85  ,La Paz Paz12 | ,La | ,City86  ,La Paz13 | ,City  ,La |
| Paz14 | ,La Paz15 | | ,La | Paz16 | ,La | Paz17 | ,La |
| Paz18 | ,La Paz19 | | ,La | Paz20 | ,La | Paz21 | ,La |
| Paz22 | ,La Paz23 | | ,La | Paz24 | ,La | Paz25 | ,La |
| Paz26 | ,La Paz27 | | ,La | Paz28 | ,La | Paz29 | ,La |
| Paz30 | ,La Paz31 | | ,La | Paz32 | ,La | Paz33 | ,La |
| Paz34 | ,La Paz35 | | ,La | Paz36 | ,La | Paz37 | ,La |
| Paz38 | ,La Paz39 | | ,La | Paz39 | ,La | Paz40 | ,La |
| Paz41 | ,La Paz42 | | ,La | Paz43 | ,La | Paz44 | ,La |
| Paz45 | ,La Paz46 | | ,La | Paz47 | ,La | Paz48 | ,La |
| Paz49 Paz8 | ,La Paz50  ,La | | ,La | Paz51 | ,La | Paz52 | ,La |
| Paz9 | ,Lima | ,Lima10 | | ,Lima11 | | ,Lima12 | |

,Lima13 ma17

,Lima22 Lima26

,Lima31

,Lima35

,Lima14

,Lima18 ,Lima19

,Lima23

,Lima27 ,Lima28

,Lima15

,Lima20

,Lima24

,Lima29

,Lima16

,Lima21

,Lima25

,Lima30

,Li

,

,Lima32

,Lima36

,Lima33

,Lima34

,Lima37

,Lima38

,Lima39

,Lima40 ,Lima40 ,Lima41 ,Lima42

,Lima43 ,Lima44 ,Lima45 ,Lima46 ,Lima 47 **,...**

SQL>

##### A1/ The LISTAGG function can be used with the ON OVERFLOW TRUNCATE clause.

***The value defined with the ON OVERFLOW TRUNCATE clause is the value that is displayed to replace the missing values, which is ‘…’.***

*Q2/ How do you know the number of values missing?*

SQL> **SELECT code,**

**LISTAGG (name, ',' ON OVERFLOW TRUNCATE '...' WITH COUNT) WITHIN GROUP (ORDER BY code) AS CITIES**

**FROM hr.cities**

**GROUP BY code ORDER BY code;**

2

0

City1

3

4

5

6

,City1

,Mexico

,Mexico

ity25 ity6

ijon le

,City26 ,City26 ,City3 ,City3

,City4 ,City4 ,City5 ,City5 ,C

,City6 ,City7 ,City7 ,City8

,City8 ,City9 ,City9 ,Dijon ,D

,Lyon ,Lyon ,Marseille ,Marseil

,Paris

,Paris

20

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 10 |  | | | | | | | | |
| Aix | ,Aix | | | ,City10 | | | ,City10 | | ,City |
| 11 ,City11 | | | | ,City12 | | ,City12 | | ,City13 | |
| ,City13 ,City14  ,City15 ,City16 ,City16  ,City18 ,City18 | | | | | | ,City14 ,City15  ,City17 ,City17  ,City19 ,City19 | | | |
| 21 | ,City2 | ,City2  ,City21 | | | ,City20  ,City22 | | ,City20  ,City22 | | ,City  ,City23 |
| ,City23 | | | ,City24 | | ,City24 | | ,City25 | | ,C |

City27 ,City28 ,City29 ,City30 ,City

31 ,City32 ,City33 ,City34 ,City35

,Dallas ,Houston ,LA ,NY

,Nashville ,Philadelphia ,SF ,Toronto ,Was hington

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 30 |  | | | |
| Bogota | ,Bogota10 | ,Bogota11 | ,Bogota12 | ,Bogo |
| ta13 | ,Bogota14 | ,Bogota15 | ,Bogota16 | ,Bogota17 |

,Bogota18 ,Bogota19 ,Bogota20 ,Bogota21

,Bogota22 ,Bogota23 ,Bogota24 ,Bogota25 ,Bo gota26 ,Bogota27 ,Bogota28 ,Bogota29 ,Bogota

30 ,Bogota31 ,Bogota32 ,Bogota33 ,Bogota34

,Bogota35 ,Bogota36 ,Bogota37 ,Bogota38 ,B ogota39 ,Bogota39 ,Bogota40 ,Bogota41 ,Bogota42

,Bogota43 ,Bogota44 ,Bogota45 ,Bogota46

,Bogota47 ,Bogota48 ,Bogota49 ,Bogota50 ,Bogot a51 ,Bogota52 ,Bogota8 ,Bogota9 ,Brasilia

,Brasilia10 ,Brasilia11 ,Brasilia12 ,Brasilia13 , Brasilia14 ,Brasilia15 ,Brasilia16 ,Brasilia17 ,Bras ilia18 ,Brasilia19 ,Brasilia20 ,Brasilia21 ,Brasilia2

2 ,Brasilia23 ,Brasilia24 ,Brasilia25 ,Brasilia26

,Brasilia27 ,Brasilia28 ,Brasilia29 ,Brasilia30 ,Br asilia31 ,Brasilia32 ,Brasilia33 ,Brasilia34 ,Brasil ia35 ,Brasilia36 ,Brasilia37 ,Brasilia8 ,Brasilia9

,Buenos Aires ,Buenos Aires10,Buenos Aires11,Buenos Aires12,Buenos Aires13,Buenos Aires14,Buenos Aires15,Buenos Aires16,Buenos Aires17,Buenos Aires18,Buenos Aires19,Buenos Aires20,Buenos Aires21,Buenos Aires22,Buenos Aires23,Buenos Aires24,Buenos Aires25,Buenos Aires26,Buenos Aires27,Buenos Aires28,Buenos Aires29,Buenos Aires30,Buenos Aires31,Buenos Aires32,Buenos Aires33,Buenos Aires34,Buenos Aires35,Buenos Aires36,Buenos Aires37,Buenos Aires38,Buenos Aires39,Buenos Aires39,Buenos Aires40,Buenos Aires41,Buenos Aires42,Buenos Aires43,Buenos Aires44,Buenos Aires45,Buenos Aires46,Buenos Aires47,Buenos Aires48,Buenos Aires49,Buenos Aires50,Buenos Aires51,Buenos Aires52,Buenos Aires8 ,Buenos

Aires9 ,City36 ,City37 ,City38 ,City39

,City40 ,City41 ,City42 ,City43 ,Cit y44 ,City45 ,City46 ,City47 ,City48

,City49 ,City50 ,City51 ,City52

,City53 ,City54 ,City55 ,City56 ,Ci ty57 ,City58 ,City59 ,City60 ,City61

,City62 ,City63 ,City64 ,City65

,City66 ,City67 ,City68 ,City69 ,City70

,City71 ,City72 ,City73 ,City74

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| ,City75  ,City79 | ,City76  ,City80 | ,City77  ,City81 | ,City78  ,City82 | , |
| City83  87 ,City88 | ,City84  ,City89 | ,City85  ,La Paz | ,City86  ,La | ,City |
| Paz10 | ,La Paz11 | ,La Paz12 | ,La Paz13 ,La | |
| Paz14 | ,La Paz15 | ,La Paz16 | ,La Paz17 ,La | |

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Paz18 | ,La | Paz19 | ,La | Paz20 | ,La | Paz21 | ,La |
| Paz22 | ,La | Paz23 | ,La | Paz24 | ,La | Paz25 | ,La |
| Paz26 | ,La | Paz27 | ,La | Paz28 | ,La | Paz29 | ,La |
| Paz30 | ,La | Paz31 | ,La | Paz32 | ,La | Paz33 | ,La |
| Paz34 | ,La | Paz35 | ,La | Paz36 | ,La | Paz37 | ,La |
| Paz38 | ,La | Paz39 | ,La | Paz39 | ,La | Paz40 | ,La |
| Paz41 | ,La | Paz42 | ,La | Paz43 | ,La | Paz44 | ,La |
| Paz45 | ,La | Paz46 | ,La | Paz47 | ,La | Paz48 | ,La |
| Paz49 | ,La | Paz50 | ,La | Paz51 | ,La | Paz52 | ,La |
| Paz8 | ,La |  |  |  |  |  |  |

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| ,Lima35 |  | ,Lima36 | ,Lima37 | | ,Lima38 | | ,Lima39 |
|  | ,Lima40 |  | ,Lima40 | | ,Lima41 | | ,Lima42 |
| ,Lima43 | | ,Lima44 | | ,Lima45 | | **,... (147)** | |

##### A2/ The way to display the number of missing values is to use the WITH COUNT clause in conjunction with the ON OVERFLOW TRUNCATE clause. This is the default in the ON OVERFLOW TRUNCATE clause. Observe that the information of the number of missing values takes characters in place of possible values (Lima46 and Lima47).

Paz9

,Lima13 ma17

,Lima22 Lima26

,Lima31

,Lima

,Lima10

,Lima14

,Lima18

,Lima19

,Lima23

,Lima11

,Lima15

,Lima20

,Lima24

,Lima29

,Lima33

,Lima12

,Lima16

,Lima21

,Lima25

,Lima30

,Lima34

,Li

,

,Lima27 ,Lima28

,Lima32

SQL>

*Q3/ Is there a way to eliminate the duplicate values for all codes from the specified expression before concatenating the values into a single string?*

SQL> **SELECT code,**

**LISTAGG (DISTINCT name, ','**

**ON OVERFLOW TRUNCATE 'other values') WITHIN GROUP (ORDER BY code) AS CITIES**

**FROM hr.cities GROUP BY code**

**ORDER BY code;**

2

CODE

3

4

5

6

7

CITIES

CODE

CITIES

0

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| City1  10 | ,Mexico |  | | |
| Aix | ,City10 | ,City11 | ,City12 | ,City |
| 13 | ,City14 | ,City15 | ,City16 | ,City17 |

,City18 ,City19 ,City2 ,City20

,City21 ,City22 ,City23 ,City24 ,City25

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| ,City26 | | | ,City3 | ,City4 | ,City5 |
| n | ,City6 | ,City7  ,Lyon | ,City8  ,Marseille | ,City9  ,Paris | ,Dijo |

20

City27 ,City28 ,City29 ,City30 ,City

31 ,City32 ,City33 ,City34 ,City35

,Dallas ,Houston ,LA ,NY

,Nashville ,Philadelphia ,SF ,Toronto ,Was hington

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 30 |  | | | |
| Bogota | ,Bogota10 | ,Bogota11 | ,Bogota12 | ,Bogo |
| ta13 | ,Bogota14 | ,Bogota15 | ,Bogota16 | ,Bogota17 |

,Bogota18 ,Bogota19 ,Bogota20 ,Bogota21

,Bogota22 ,Bogota23 ,Bogota24 ,Bogota25 ,Bo gota26 ,Bogota27 ,Bogota28 ,Bogota29 ,Bogota

30 ,Bogota31 ,Bogota32 ,Bogota33 ,Bogota34

,Bogota35 ,Bogota36 ,Bogota37 ,Bogota38 ,B ogota39 ,Bogota40 ,Bogota41 ,Bogota42 ,Bogota43

,Bogota44 ,Bogota45 ,Bogota46 ,Bogota47

,Bogota48 ,Bogota49 ,Bogota50 ,Bogota51 ,Bogot a52 ,Bogota8 ,Bogota9 ,Brasilia ,Brasilia1

0 ,Brasilia11 ,Brasilia12 ,Brasilia13 ,Brasilia14

,Brasilia15 ,Brasilia16 ,Brasilia17 ,Brasilia18 ,Bra silia19 ,Brasilia20 ,Brasilia21 ,Brasilia22 ,Brasilia

23 ,Brasilia24 ,Brasilia25 ,Brasilia26 ,Brasilia27

,Brasilia28 ,Brasilia29 ,Brasilia30 ,Brasilia31 ,B rasilia32 ,Brasilia33 ,Brasilia34 ,Brasilia35 ,Brasi lia36 ,Brasilia37 ,Brasilia8 ,Brasilia9 ,Buenos Aires ,Buenos Aires10,Buenos Aires11,Buenos Aires12,Buenos Ai

res13,Buenos Aires14,Buenos Aires15,Buenos Aires16,Buenos Aires17,Buenos Aires18,Buenos Aires19,Buenos Aires20,Buenos Aires21,Buenos Aires22,Buenos Aires23,Buenos Aires24,Buenos Aires25,Buenos Aires26,Buenos Aires27,Buenos Aires28,Buenos Aires29,Buenos Aires30,Buenos Aires31,Buenos Aires32,Buenos Aires33,Buenos Aires34,Buenos Aires35,Buenos Aires36,Buenos

Aires37,Buenos Aires38,Buenos Aires39,Buenos Aires40,Buenos Aires41,Buenos Aires42,Buenos Aires43,Buenos Aires44,Bueno

s Aires45,Buenos Aires46,Buenos Aires47,Buenos Aires48,Buenos Aires49,Buenos Aires50,Buenos Aires51,Buenos Aires52,Buenos Aires8 ,Buenos

Aires9 ,City36 ,City37 ,City38 ,City39

,City40 ,City41 ,City42 ,City43 ,Cit

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| y44 | ,City45 | ,City46 | ,City47 | ,City48 |
|  | ,City49 | ,City50 | ,City51 | ,City52 |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| y70 | ,City71 | ,City72 | ,City73 | ,City74 |  |
| ,City75 |  | ,City76 | ,City77 | ,City78 | , |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| ,City53  y57 | ,City54  ,City58 | | ,City55  ,City59 | ,City56  ,City60 | ,Cit  ,City61 |
| ,City62 | | ,City63 | ,City64 | ,City65 | |
| ,City66 | ,City67 | | ,City68 | ,City69 | ,Cit |

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| City79  ,City84  ,City88 | | | ,City80 ,City81  ,City85  ,City89 ,La Paz | | | ,City82  ,City86  ,La Paz10 | | ,City83  ,City87  ,La | |
| Paz11 | ,La | | Paz12 | ,La Paz13 | | ,La Paz14 | | ,La | |
| Paz15 | ,La | | Paz16 | ,La Paz17 | | ,La Paz18 | | ,La | |
| Paz19 | ,La | | Paz20 | ,La Paz21 | | ,La Paz22 | | ,La | |
| Paz23 | ,La | | Paz24 | ,La Paz25 | | ,La Paz26 | | ,La | |
| Paz27 | ,La | | Paz28 | ,La Paz29 | | ,La Paz30 | | ,La Paz3 | |
| 1  Paz35 | ,La | Paz32  ,La Paz36 | | ,La | Paz33  ,La Paz37 | ,La | Paz34  ,La Paz38 | ,La | ,La |
| Paz39 Paz43 Paz47 Paz51  Paz9 |  | ,La Paz40  ,La Paz44  ,La Paz48  ,La Paz52  ,Lima | |  | ,La Paz41  ,La Paz45  ,La Paz49  ,La Paz8  ,Lima10 |  | ,La Paz42  ,La Paz46  ,La Paz50  ,La  ,Lima11 |  | ,La  ,La  ,La  ,Lima12 |

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| ,Lima13  ,Lima17  a21 ,Lima22  ,Lima26 ima30 ,Lima31  ,Lima35  Lima39 ,Lima40 | | ,Lima14  ,Lima18  ,Lima23  ,Lima27  ,Lima32  ,Lima36  ,Lima41 | | ,Lima15  ,Lima19  ,Lima24  ,Lima28  ,Lima33  ,Lima37  ,Lima42 | ,Lima16  ,Lima20  ,Lima25  ,Lima29  ,Lima34  ,Lima38  ,Lima43 | | ,Lim  ,L  , |
| ,Lima44  Lima48 | ,Lima45  ,Lima49 | | ,Lima46  ,other values(**140**) | | | ,Lima47 | , |

SQL>

SQL> **EXIT**

$

##### A3/ Two identical strings such as ‘Paris’ and ‘Paris ‘ with trailing spaces are treated as duplicates. The DISTINCT and ON OVERFLOW TRUNCATE clauses used together allow the elimination of duplicates before the concatenation of the values into a single string. This reduces the chance to miss values as this is the case in the example where Lima48 and Lima49 are now displayed.

***Note: Oracle uses blank-padded comparison semantics only when both values in the comparison are either expressions of data type CHAR, NCHAR, text literals, or values returned by the USER function.***

# Practices for Lesson 2: Using Security Enhancements

## Practices for Lesson 2: Overview

### Overview

In these practices, you will discover and practice new Oracle Database 19c enhancements related to security such as Oracle-supplied schemas defined as schema only accounts, enable Database Vault operations control to protect application data in PDBs against common users, constrain AUDIT POLICY and NOAUDIT POLICY SQL commands with Oracle Database Vault command rules, audit direct user activities, and finally manage operations on Oracle-managed and user-managed tablespaces encrypted in TDE.

## Practice 2-1: Exploring Oracle-supplied Schemas Only Accounts

### Overview

In this practice, you will find which Oracle-supplied schemas are schema only accounts.

### Tasks

1. If DUPORCL CDB was not dropped during Practice 1-4 step 4, execute the

/home/oracle/labs/DB/drop\_DUPORCL.sh shell script to drop DUPORCL.

$ **/home/oracle/labs/DB/drop\_DUPORCL.sh**

…

$

1. Execute the /home/oracle/labs/admin/cleanup\_PDBs.sh shell script. The shell script drops all PDBs that may have been created by any of the practices in ORCL, and finally recreates PDB1.

$ **$HOME/labs/admin/cleanup\_PDBs.sh**

…

$

1. Before starting the practice, execute the $HOME/labs/SEC/glogin.sh shell script. It sets formatting for all columns selected in queries.

$ **$HOME/labs/SEC/glogin.sh**

**…**

$

1. In ORCL, find the Oracle-supplied schemas that are now schemas only accounts.

$ **. oraenv**

ORACLE\_SID = [oracle] ? **ORCL**

The Oracle base has been set to /u01/app/oracle

$ **sqlplus / AS SYSDBA**

SQL> **SELECT username FROM dba\_users WHERE authentication\_type = 'NONE' ORDER BY 1;**

2 3

USERNAME

APPQOSSYS AUDSYS DBSFWUSER DIP

DVF

DVSYS

GGSYS GSMADMIN\_INTERNAL GSMCATUSER GSMROOTUSER GSMUSER

LBACSYS MDDATA MDSYS OJVMSYS OLAPSYS ORACLE\_OCM ORDDATA ORDPLUGINS ORDSYS OUTLN

REMOTE\_SCHEDULER\_AGENT SI\_INFORMTN\_SCHEMA SYS$UMF

SYSBACKUP SYSDG SYSKM SYSRAC WMSYS

XDB

30 rows selected.

SQL>

*Q1/ Can there be any administrative privileged users that are schema only accounts?*

SQL> **CREATE USER c##user NO AUTHENTICATION CONTAINER=ALL;**

User created.

SQL> **GRANT sysoper TO c##user CONTAINER=ALL;**

Grant succeeded.

SQL> **SELECT username, authentication\_type FROM dba\_users WHERE username = 'C##USER';**

2

USERNAME

AUTHENTI

C##USER

NONE

SQL> **SELECT username, authentication\_type FROM v$pwfile\_users;**

USERNAME

AUTHENTI

SYS

C##USER

PASSWORD

**NONE**

SQL> **EXIT**

$

***A1/ Yes, in Oracle Database 19c, administrative privileged users can be schema only accounts.***

## Practice 2-2: Protecting Application Data by Using Database Vault Operations Control

### Overview

In this practice you will enable Database Vault operations control and observe how application data in PDBs is protected against common users.

### Tasks

1. Create the HR.EMPLOYEES table and its application data in PDB1 by executing the

/home/oracle/labs/SEC/appdata.sh.

$ **/home/oracle/labs/SEC/appdata.sh**

…

$

*Q1/ Can SYS read the HR.EMPLOYEES table data in PDB?*

$ **sqlplus sys@PDB1 AS SYSDBA**

Enter password: *password*

SQL> **SELECT count(\*) FROM hr.employees;**

COUNT(\*)

107

SQL>

##### A2/ Neither Oracle Database Vault nor Database Vault operations control is configured in PDB1.

SQL> **CONNECT / AS SYSDBA**

Connected.

SQL> **SELECT \* FROM dba\_dv\_status;**

NAME

STATUS

DV\_APP\_PROTECTION

DV\_CONFIGURE\_STATUS DV\_ENABLE\_STATUS

NOT CONFIGURED FALSE

FALSE

SQL>

1. Execute the /home/oracle/labs/SEC/drop\_create\_user\_sec\_admin.sql SQL script. The script creates the C##SEC\_ADMIN and C##ACCTS\_ADMIN accounts in ORCL.

SQL> **@/home/oracle/labs/SEC/drop\_create\_user\_sec\_admin.sql**

… SQL>

1. Configure Database Vault operations control at the CDB root level in ORCL ensuring that the DV\_OWNER role is granted locally in the CDB root to the common Oracle Database Vault owner.

SQL> **exec DVSYS.CONFIGURE\_DV( -**

**dvowner\_uname =>'c##sec\_admin',- dvacctmgr\_uname =>'c##accts\_admin', - force\_local\_dvowner => TRUE)**

> > >

PL/SQL procedure successfully completed.

SQL>

1. Observe the Oracle Database Vault status and application protection status in the CDB root and in PDB1.

SQL> **SELECT \* FROM dba\_dv\_status;**

NAME

STATUS

DV\_APP\_PROTECTION

DV\_CONFIGURE\_STATUS DV\_ENABLE\_STATUS

NOT CONFIGURED TRUE

FALSE

SQL>

* 1. Connect to PDB1.

SQL> **CONNECT sys@PDB1 AS SYSDBA**

Enter password: *password*

Connected.

SQL> **SELECT \* FROM dba\_dv\_status;**

NAME

STATUS

DV\_APP\_PROTECTION

DV\_CONFIGURE\_STATUS DV\_ENABLE\_STATUS

NOT CONFIGURED FALSE

FALSE

SQL>

Q1/ *Can SYS read the HR.EMPLOYEES table data in PDB?*

SQL> **SELECT count(\*) FROM hr.employees;**

COUNT(\*)

107

SQL>

##### A2/ Yes. SYS can still read the application data in PDB1. Oracle Database Vault is configured in the CDB root but Oracle Database Vault is not configured in PDB1 and Database Vault operations control is not enabled in the CDB root and PDB1.

1. Enable Database Vault operations control in the CDB root.
   1. Create a common user granted the CREATE SESSION and SELECT ANY TABLE

privileges.

SQL> **CONNECT c##accts\_admin** Enter password: *password* Connected.

SQL> **CREATE USER c##common\_user IDENTIFIED BY *password***

**CONTAINER=ALL;**

2

User created.

SQL> **CONNECT / AS SYSDBA**

Connected.

SQL> **GRANT create session, select any table TO c##common\_user CONTAINER=ALL;**

2

Grant succeeded.

SQL>

* 1. Connect as the common user and check if the common user can access the application data in PDB1.

SQL> **CONNECT c##common\_user@PDB1** Enter password: *password* Connected.

SQL> **SELECT count(\*) FROM hr.employees;**

COUNT(\*)

107

SQL>

* 1. Enable the application protection via the Database Vault operations control in the CDB root.

SQL> **CONNECT / AS SYSDBA**

Connected.

SQL> **EXEC dvsys.dbms\_macadm.enable\_app\_protection (NULL)**

BEGIN dvsys.dbms\_macadm.enable\_app\_protection (NULL); END;

\*

ERROR at line 1:

ORA-47503: Database Vault is not enabled in CDB$ROOT or application root.

ORA-06512: at "DVSYS.DBMS\_MACADM", line 2811

ORA-06512: at line 1

SQL>

* 1. Enable Oracle Database Vault in the CDB root.

SQL> **CONNECT c##sec\_admin** Enter password: password Connected.

SQL> **EXEC dvsys.dbms\_macadm.enable\_dv**

PL/SQL procedure successfully completed. SQL> **CONNECT / AS SYSDBA**

Connected.

SQL> **SELECT \* FROM dba\_dv\_status;**

NAME

STATUS

DV\_APP\_PROTECTION

DV\_CONFIGURE\_STATUS DV\_ENABLE\_STATUS

NOT CONFIGURED TRUE

FALSE

SQL>

* 1. Restart the database instance to enforce DV configuration and enablement.

SQL> **SHUTDOWN IMMEDIATE**

Database closed.

Database dismounted. ORACLE instance shut down. SQL> **STARTUP**

ORACLE instance started.

Total System Global Area 1426062768 bytes Fixed Size 9129392 bytes Variable Size 486539264 bytes

Database Buffers 922746880 bytes Redo Buffers 7647232 bytes Database mounted.

Database opened.

SQL> **ALTER PLUGGABLE DATABASE pdb1 OPEN;**

Pluggable database altered.

SQL> **SELECT \* FROM dba\_dv\_status;**

NAME

STATUS

DV\_APP\_PROTECTION DV\_CONFIGURE\_STATUS

DV\_ENABLE\_STATUS

NOT CONFIGURED

**TRUE**

**TRUE**

SQL>

*Q1/ Which user can enable the Database Vault operations control?*

SQL> **EXEC dvsys.dbms\_macadm.enable\_app\_protection**

BEGIN dvsys.dbms\_macadm.enable\_app\_protection; END;

\*

ERROR at line 1: ORA-06550: line 1, column 13: PLS-00904: insufficient privilege to access object DVSYS.DBMS\_MACADM

ORA-06550: line 1, column 7: PL/SQL: Statement ignored

SQL>

##### A1/ Only common accounts with DV\_OWNER role can enable Database Vault operations control, even if the DV\_OWNER role is granted locally.

SQL> **CONNECT c##sec\_admin** Enter password: password Connected.

SQL> **EXEC dvsys.dbms\_macadm.enable\_app\_protection**

PL/SQL procedure successfully completed. SQL>

1. Display the status of Database Vault operations control in the CDB root and in PDB1.

SQL> **CONNECT / AS SYSDBA**

Connected.

SQL> **SELECT \* FROM dba\_dv\_status;**

NAME

STATUS

**DV\_APP\_PROTECTION**

DV\_CONFIGURE\_STATUS DV\_ENABLE\_STATUS

**ENABLED**

TRUE TRUE

SQL> **CONNECT sys@PDB1 AS SYSDBA**

Enter password: *password*

Connected.

SQL> **SELECT \* FROM dba\_dv\_status;**

NAME

STATUS

**DV\_APP\_PROTECTION**

DV\_CONFIGURE\_STATUS DV\_ENABLE\_STATUS

**ENABLED**

FALSE FALSE

SQL>

*Q1/ Is the common user able to view the application data in PDB1?*

SQL> **CONNECT c##common\_user@PDB1** Enter password: *password* Connected.

SQL> **SELECT COUNT(\*) FROM hr.employees;**

SELECT COUNT(\*) FROM hr.employees

\*

ERROR at line 1:

ORA-01031: insufficient privileges

SQL>

##### A1/ No. Database Vault operations control being enabled prevents any common user to view application data in PDBs.

*Q2/ Does Database Vault operations control prevent common users from backing up the PDB in which they are granted the SYSDBA privilege locally?*

SQL> **CONNECT sys@PDB1 AS SYSDBA**

Enter password: *password*

Connected.

SQL> **GRANT sysdba TO c##common\_user;**

Grant succeeded.

SQL> **EXIT**

$

$ **rman target c##common\_user@PDB1**

target database Password:

connected to target database: ORCL:PDB1 (DBID=505765219) RMAN> **BACKUP DATABASE;**

Starting backup at 07-MAY-19

using target database control file instead of recovery catalog allocated channel: ORA\_DISK\_1

channel ORA\_DISK\_1: SID=37 device type=DISK

channel ORA\_DISK\_1: starting full datafile backup set channel ORA\_DISK\_1: specifying datafile(s) in backup set

input datafile file number=00031 name=/u02/app/oracle/oradata/ORCL/pdb1/ORCL/884E0B72E5E159F9E053 E311ED0A1FC9/datafile/o1\_mf\_sysaux\_gf35l9s5\_.dbf

input datafile file number=00030 name=/u02/app/oracle/oradata/ORCL/pdb1/ORCL/884E0B72E5E159F9E053 E311ED0A1FC9/datafile/o1\_mf\_system\_gf35l9ry\_.dbf

input datafile file number=00032 name=/u02/app/oracle/oradata/ORCL/pdb1/ORCL/884E0B72E5E159F9E053 E311ED0A1FC9/datafile/o1\_mf\_undotbs1\_gf35l9s9\_.dbf

channel ORA\_DISK\_1: starting piece 1 at 07-MAY-19 channel ORA\_DISK\_1: finished piece 1 at 07-MAY-19

piece handle=/u03/app/oracle/fast\_recovery\_area/ORCL/884E0B72E5E159F9E 053E311ED0A1FC9/backupset/2019\_05\_07/o1\_mf\_nnndf\_TAG20190507T143 541\_gf35xy3p\_.bkp tag=TAG20190507T143541 comment=NONE

channel ORA\_DISK\_1: backup set complete, elapsed time: 00:00:08 Finished backup at 07-MAY-19

RMAN> **EXIT**

$

##### A2/ If common users cannot view application data in PDBs, they can still complete administrative tasks for which they are granted privileges.

1. Disable the Database Vault operations control.

$ **sqlplus c##sec\_admin**

Enter password: password

SQL> **EXEC dvsys.dbms\_macadm.disable\_app\_protection**

PL/SQL procedure successfully completed. SQL>

1. Display the status of Database Vault operations control in the CDB root and in PDB1.

SQL> **CONNECT / AS SYSDBA**

Connected.

SQL> **SELECT \* FROM dba\_dv\_status;**

NAME

STATUS

**DV\_APP\_PROTECTION**

DV\_CONFIGURE\_STATUS DV\_ENABLE\_STATUS

**DISABLED**

TRUE TRUE

SQL> **CONNECT sys@PDB1 AS SYSDBA**

Enter password: *password*

Connected.

SQL> **SELECT \* FROM dba\_dv\_status;**

*Q1/ What do you observe about the status value?*

NAME

STATUS

**DV\_APP\_PROTECTION**

DV\_CONFIGURE\_STATUS DV\_ENABLE\_STATUS

**DISABLED**

FALSE FALSE

SQL>

##### A1/ The status shows DISABLED and not NOT CONFIGURED. This means that Database Vault operations control has been enabled previously.

*Q2/ Is the common user able to view the application data in PDB1?*

SQL> **CONNECT c##common\_user@PDB1** Enter password: *password* Connected.

SQL> **SELECT COUNT(\*) FROM hr.employees;**

COUNT(\*)

107

SQL>

***A2/*** *Yes.*

*Q3/ Can the Database Vault operations control be enabled in a PDB and disabled in another PDB?*

1. Use the /home/oracle/labs/SEC/create\_PDB2.sql to create another PDB in ORCL.

SQL> **CONNECT / AS SYSDBA**

Connected.

SQL> **@/home/oracle/labs/SEC/create\_PDB2.sql**

… SQL>

1. Enable the Database Vault operations control in PDB2.

SQL> **CONNECT c##sec\_admin@PDB2**

Enter password: password

Connected.

SQL> **EXEC dvsys.dbms\_macadm.enable\_app\_protection**

BEGIN dvsys.dbms\_macadm.enable\_app\_protection; END;

\*

ERROR at line 1:

ORA-65040: operation not allowed from within a pluggable database

ORA-06512: at "DVSYS.DBMS\_MACADM", line 2811

ORA-06512: at line 1

SQL>

##### A3/ No. Database Vault operations control be enabled in the CDB root only.

1. Reenable the Database Vault operations control.

SQL> **CONNECT c##sec\_admin**

Enter password: password

SQL> **EXEC dvsys.dbms\_macadm.enable\_app\_protection**

PL/SQL procedure successfully completed. SQL>

1. The situation is that HR application data in PDB1 is very sensitive and should be protected against common users in the CDB. Nevertheless the C##REPORT common user should be able to access some of the HR application information in PDB1 to generate statistics for Human Resources.
   1. Create the common user C##REPORT granted the CREATE SESSION and SELECT ANY TABLE privileges.

SQL> **CONNECT c##accts\_admin** Enter password: *password* Connected.

SQL> **CREATE USER c##report IDENTIFIED BY *password***

**CONTAINER=ALL;**

2

User created.

SQL> **CONNECT / AS SYSDBA**

Connected.

SQL> **GRANT create session, select any table TO c##report**

**CONTAINER=ALL;**

2

Grant succeeded.

SQL>

* 1. Connect as the common user in PDB1 and check if the common user can access the application data in PDB1.

SQL> **CONNECT c##report@PDB1** Enter password: *password* Connected.

SQL> **SELECT count(\*) FROM hr.employees;**

SELECT count(\*) FROM hr.employees

\*

ERROR at line 1:

ORA-01031: insufficient privileges

SQL>

*Q1/ Is it the expected behavior?*

##### A1/ Yes. Database Vault operations control is enabled.

*Q2/ Is it possible to let C##REPORT access HR data in PDB1?*

##### A2/ Yes. The common user can be added in the exception list of users and packages allowed to access local data in PDBs.

SQL> **CONNECT c##sec\_admin@PDB1** Enter password: password Connected.

SQL> **EXEC dvsys.dbms\_macadm.add\_app\_exception( -**

**owner => 'C##REPORT', -**

**package\_name => '')**

BEGIN dvsys.dbms\_macadm.add\_app\_exception(owner => 'C##REPORT', package\_name => ''); END;

\*

ERROR at line 1:

ORA-65040: operation not allowed from within a pluggable database

ORA-06512: at "DVSYS.DBMS\_MACADM", line 1403 ORA-06512: at "DVSYS.DBMS\_MACADM", line 1741

ORA-06512: at line 1

SQL>

Observe that any operation related to Database Vault operations control must be completed from the CDB root.

SQL> **CONNECT c##sec\_admin** Enter password: password Connected.

SQL> **EXEC dvsys.dbms\_macadm.add\_app\_exception( - owner => 'C##REPORT', -**

**package\_name => '')**

> >

PL/SQL procedure successfully completed.

SQL>

1. Display the exception list.
2. Connect as the common user in PDB1 and check if the common user can access the application data in PDB1.

SQL> **CONNECT c##report@PDB1** Enter password: *password* Connected.

SQL> **SELECT count(\*) FROM hr.employees;**

COUNT(\*)

107

SQL>

SQL> **SELECT \* FROM DVSYS.DBA\_DV\_APP\_EXCEPTION;**

OWNER

PACKAGE

C##REPORT

%

SQL>

*Q1/ Is it the expected behavior?*

##### A1/ Yes. Database Vault operations control handles C##REPORT user as an exception who can access local data.

SQL> **CONNECT c##common\_user@PDB1**

Enter password: *password*

Connected.

SQL> **SELECT count(\*) FROM hr.employees;**

SELECT count(\*) FROM hr.employees

\*

ERROR at line 1:

ORA-01031: insufficient privileges

SQL> **EXIT**

$

1. Disable Database Vault operations control by executing the

/home/oracle/labs/SEC/disable\_DVOps.sh.

$ **/home/oracle/labs/SEC/disable\_DVOps.sh**

…

$

## Practice 2-3: Constraining AUDIT POLICY and NOAUDIT POLICY SQL Commands with Oracle Database Vault Command Rules

### Overview

In this practice you will forfid any user other than SYS and SYSTEM from using the AUDIT POLICY and NOAUDIT POLICY commands.

### Tasks

1. Configure and enable Oracle Database Vault in the CDB root and in PDB1 by executing the

/home/oracle/labs/SEC/setup\_DV\_ORCL.sh shell script.

$ **/home/oracle/labs/SEC/setup\_DV\_ORCL.sh**

…

$

1. Connect as SYSTEM to PDB1 and let SYSTEM create the required audit policies.

$ **sqlplus system@PDB1**

Enter password: *password*

SQL> **CREATE AUDIT POLICY pol1 ACTIONS SELECT ON hr.employees;**

Audit policy created. SQL>

1. Enable the audit policy.

SQL> **AUDIT POLICY pol1;**

Audit succeeded.

SQL>

1. Connect in PDB1 as the security officer, C##SEC\_ADMIN.

SQL> **CONNECT c##sec\_admin@PDB1** Enter password: *password* Connected.

SQL>

1. Create a command rule that forbids users that are not SYS not SYSTEM from using the

AUDIT POLICY and NOAUDIT POLICY commands in any circumstance and in PDB1 only.

* 1. First create the rule set to which you will associate the **Is Database Administrator** rule that checks whether the user executing a NOAUDIT POLICY command is granted the DBA role.

SQL> **EXEC dvsys.DBMS\_MACADM.CREATE\_RULE\_SET( -**

**rule\_set\_name => 'Check\_user', -**

**description**

**enabled eval\_options**

**=> 'Check user', -**

**=> DBMS\_MACUTL.G\_YES, -**

**=> DBMS\_MACUTL.G\_RULESET\_EVAL\_ANY,-**

**audit\_options => DBMS\_MACUTL.G\_RULESET\_AUDIT\_FAIL,- fail\_options => DBMS\_MACUTL.G\_RULESET\_FAIL\_SILENT,- fail\_message => '',-**

**fail\_code => '',-**

**handler\_options => DBMS\_MACUTL.G\_RULESET\_HANDLER\_OFF,- handler => '',-**

**is\_static => TRUE,-**

**scope => DBMS\_MACUTL.G\_SCOPE\_LOCAL)**

> > > > > > > > > > > > > > > >

PL/SQL procedure successfully completed.

SQL>

* 1. Then associate the predefined Is SYS or SYSTEM User rule to the rule set.

SQL> **EXEC dvsys.DBMS\_MACADM.ADD\_RULE\_TO\_RULE\_SET( -**

**rule\_set\_name => 'Check\_user',-**

**rule\_name => 'Is SYS or SYSTEM User')**

> >

PL/SQL procedure successfully completed.

SQL>

* 1. Then create the command rule.

SQL> **EXEC dvsys.DBMS\_MACADM.CREATE\_COMMAND\_RULE( - command => 'AUDIT POLICY', -**

**rule\_set\_name => 'Check\_user',- object\_owner => '%', - object\_name => 'POL1',-**

**enabled => DBMS\_MACUTL.G\_YES, - scope => DBMS\_MACUTL.G\_SCOPE\_LOCAL)**

> > > > > >

PL/SQL procedure successfully completed.

SQL>

1. Reconnect as SYSTEM to PDB1, disable the audit policy and drop the policy.

SQL> **CONNECT system@PDB1** Enter password: *password* Connected.

SQL> **NOAUDIT POLICY pol1;**

Noaudit succeeded.

SQL> **DROP AUDIT POLICY pol1;**

Audit Policy dropped. SQL>

*Q1/ How is it possible that SYSTEM can disable the POL1 audit policy and drop it although the policy is protected by a command rule?*

##### A1/ The command rule is associated with the pre-defined rule “Is SYS or SYSTEM User”. The rule verifies that the user executing AUDIT POLICY or NOAUDIT POLICY command is either SYS or SYSTEM. If this is not the case, the command violates the command rule and is rejected.

1. Create a DBA junior and grant the user the DBA role in PDB1.

SQL> **CONNECT c##accts\_admin@PDB1** Enter password: *password* Connected.

SQL> **CREATE USER dba\_junior IDENTIFIED BY *password*;**

User created.

SQL> **CONNECT sys@PDB1 AS SYSDBA**

Enter password: *password*

Connected.

SQL> **GRANT dba TO dba\_junior;**

Grant succeeded.

SQL>

1. Connect as the DBA junior and create an audit policy.

SQL> **CONNECT dba\_junior@PDB1** Enter password: password Connected.

SQL> **CREATE AUDIT POLICY pol1 ACTIONS SELECT ON hr.employees;**

Audit policy created. SQL>

1. Enable the audit policy.

SQL> **AUDIT POLICY pol1;**

AUDIT POLICY pol1

\*

ERROR at line 1:

ORA-47400: Command Rule violation for AUDIT POLICY on POL1 SQL>

SQL>

*Q1/ Who can now disable the audit policy if necessary?*

SQL> **CONNECT c##sec\_admin@PDB1** Enter password: *password* Connected.

SQL> **NOAUDIT POLICY pol1;**

NOAUDIT POLICY pol1

\*

ERROR at line 1:

ORA-47400: Command Rule violation for AUDIT POLICY on POL1

SQL>

##### A1/ The command rule works for AUDIT POLICY and NOAUDIT POLICY commands even if the command defined is AUDIT POLICY. The Oracle Database Vault owner cannot disable the policy because C##SEC\_ADMIN is neither SYS nor SYSTEM, the only user names checked by the “Is SYS or SYSTEM User” rule.

*Q2/ Can SYSTEM disable the audit policy even if SYSTEM did not create it?*

SQL> **CONNECT system@PDB1** Enter password: *password* Connected.

SQL> **NOAUDIT POLICY pol1;**

Noaudit succeeded.

SQL> **DROP AUDIT POLICY pol1;**

Audit Policy dropped.

SQL> **EXIT**

$

##### A2/ SYS or SYSTEM can disable the audit policy even if they did not create it because an audit policy is a non-schema object, and therefore not owned by anyone.

1. Remove the Oracle Database Vault command rule and disable Oracle Database Vault in the CDB root and PDB1.

$ **/home/oracle/labs/SEC/disable\_DV\_ORCL.sh**

…

$

## Practice 2-4: Auditing Direct User Activities

### Overview

In this practice you will audit top-level user activities in the database without collecting indirect user activity.

### Tasks

1. Execute the /home/oracle/labs/SEC/create\_proc.sh shell script. The shell script creates a procedure that allows the HR user to raise employees’ salary in PDB1.

$ **/home/oracle/labs/SEC/create\_proc.sh**

…

$

1. In *Session1*, create and enable an audit policy that audits any salary increase.

$ **sqlplus system@PDB1**

Enter password: *password*

SQL> **CREATE AUDIT POLICY pol\_sal\_increase ACTIONS UPDATE ON hr.employees;**

2

Audit policy created.

SQL> **AUDIT POLICY pol\_sal\_increase WHENEVER SUCCESSFUL;**

Audit succeeded.

SQL>

1. In another session, *Session2*, connect as HR and increase the salary for employee ID 106 through the RAISE\_SALARY procedure and then with the UPDATE command directly on the row.

$ **. oraenv**

ORACLE\_SID = [oracle] ? **ORCL**

The Oracle base has been set to /u01/app/oracle

$ **sqlplus hr@PDB1**

Enter password: *password*

SQL> **EXEC emp\_admin.raise\_salary(106,10)**

PL/SQL procedure successfully completed.

SQL> **UPDATE hr.employees SET salary=salary\*0.1 WHERE employee\_id = 106;**

2

1 row updated.

SQL> **COMMIT;**

Commit complete.

SQL> **EXIT**

$

*Q1/ In Session1, are the update actions executed through the PL/SQL procedure and directly by the UPDATE command audited?*

SQL> **SELECT action\_name, object\_name, sql\_text FROM unified\_audit\_trail**

**WHERE unified\_audit\_policies = 'POL\_SAL\_INCREASE';**

2 3

ACTION\_NAME OBJECT\_NAME

SQL\_TEXT

UPDATE

EMPLOYEES

UPDATE EMPLOYEES SET SALARY = SALARY + :B2 WHERE EMPLOYEE\_ID

= :B1

UPDATE

EMPLOYEES

UPDATE hr.employees SET salary=salary\*0.1 WHERE employee\_id = 106

SQL>

##### A1/ Yes, they are.

1. Drop the policy.

SQL> **NOAUDIT POLICY pol\_sal\_increase;**

Noaudit succeeded.

SQL> **DROP AUDIT POLICY pol\_sal\_increase;**

Audit Policy dropped. SQL>

1. In *Session1*, create and enable another audit policy that audits any salary increase executed directly with an UPDATE command.

SQL> **CREATE AUDIT POLICY pol\_sal\_direct\_increase**

**ACTIONS UPDATE ON hr.employees ONLY TOPLEVEL;**

2

Audit policy created.

SQL> **AUDIT POLICY pol\_sal\_direct\_increase WHENEVER SUCCESSFUL;**

Audit succeeded.

SQL>

1. In *Session2*, repeat step 3 for employee ID 107.

$ **sqlplus hr@PDB1**

Enter password: *password*

SQL> **EXEC emp\_admin.raise\_salary(107,30)**

PL/SQL procedure successfully completed.

SQL> **UPDATE hr.employees SET salary=salary\*0.1 WHERE employee\_id = 107;**

2

1 row updated.

SQL> **COMMIT;**

Commit complete.

SQL> **EXIT**

$

*Q1/ In Session1, are the update actions executed through the PL/SQL procedure and directly by the UPDATE command audited?*

SQL> **SELECT action\_name, object\_name FROM unified\_audit\_trail**

**WHERE unified\_audit\_policies = 'POL\_SAL\_DIRECT\_INCREASE';**

2 3

ACTION\_NAME OBJECT\_NAME

UPDATE

EMPLOYEES

SQL> **EXIT**

$

##### A1/ Only the direct UPDATE statement is audited as this is the purpose of the ONLY TOPLEVEL clause of the CREATE AUDIT POLICY command.

1. In *Session1*, drop the audit policies by executing the

/home/oracle/labs/SEC/drop\_audit.sh shell script.

$ **/home/oracle/labs/SEC/drop\_audit.sh**

…

$

## Practice 2-5: Handling Operations on Oracle-managed and User- managed Tablespaces Encrypted in TDE

### Overview

In this practice you will manage operations on data of Oracle-managed and user-managed tablespaces unaffected by closing the TDE keystore.

You use also use the Oracle By Example “Handling Operations on Oracle-Managed and User- Managed Tablespaces Encrypted” to see how to proceed. Launch a browser and click the bookmark from the Bookmarks toolbar of the browser. The URL is file:///home/oracle/labs/OBEs/F11940\_01/html/index.html.

### Tasks

1. Execute the /home/oracle/labs/SEC/create\_TBS.sh shell script to create a user- managed tablespace in PDB1. The shell script first creates the CDB root keystore, opens the keystore and sets the master encryption key at the CDB root level and in PDB1.

$ **/home/oracle/labs/SEC/create\_TBS.sh**

…

$

1. Check which tablespaces in the CDB root are encrypted.

$ **. oraenv**

ORACLE\_SID = [ORCL] ? **ORCL**

The Oracle base remains unchanged with value /u01/app/oracle

$ **sqlplus / AS SYSDBA**

SQL> **SELECT tablespace\_name, encrypted FROM dba\_tablespaces;**

TABLESPACE\_NAME

ENC

SYSTEM SYSAUX UNDOTBS1 TEMP

USERS

NO NO NO NO

NO

SQL>

1. Switch one of the Oracle-managed tablespaces and one of the user-managed tablespaces to encryption.

SQL> **ALTER TABLESPACE system ENCRYPTION USING 'AES192' ENCRYPT;**

ALTER TABLESPACE system ENCRYPTION USING 'AES192' ENCRYPT

\*

ERROR at line 1:

ORA-28365: wallet is not open

SQL> **ALTER TABLESPACE users ENCRYPTION USING 'AES192' ENCRYPT;**

ALTER TABLESPACE users ENCRYPTION USING 'AES192' ENCRYPT

\*

ERROR at line 1:

ORA-28365: wallet is not open

SQL>

1. Open the CDB root keystore and set the key.

SQL> **ADMINISTER KEY MANAGEMENT SET KEYSTORE OPEN**

**IDENTIFIED BY *password* CONTAINER = ALL;**

2

keystore altered.

SQL>

SQL> **ALTER TABLESPACE users ENCRYPTION USING 'AES192' ENCRYPT;**

Tablespace altered.

SQL> **ALTER TABLESPACE system ENCRYPTION USING 'AES192' ENCRYPT;**

Tablespace altered. SQL>

1. Verify that the tablespaces SYSTEM and USERS in the CDB root are encrypted.

SQL> **SELECT tablespace\_name, encrypted FROM dba\_tablespaces;**

TABLESPACE\_NAME

ENC

SYSTEM SYSAUX UNDOTBS1 TEMP

USERS

YES NO NO NO

YES

SQL>

1. Close the CDB root keystore.

SQL> **ADMINISTER KEY MANAGEMENT SET KEYSTORE CLOSE**

**IDENTIFIED BY *password* CONTAINER = ALL;**

2

keystore altered.

SQL>

*Q1/ Can you change the encryption algorithm for the tablespace SYSTEM?*

SQL> **ALTER TABLESPACE system ENCRYPTION USING 'AES128' ENCRYPT;**

ALTER TABLESPACE system ENCRYPTION USING 'AES128' ENCRYPT

\*

ERROR at line 1:

ORA-28365: wallet is not open

SQL>

##### A1/ No, it is not possible because the operation affects the encryption metadata of the Oracle-managed tablespace. The metadata updates are prevented with an ORA-28365 "wallet is not open" error, because the TDE master encryption key is not available when the TDE keystore is closed.

*Q2/ Can you create tables, insert data in the tablespace SYSTEM?*

SQL> **CREATE TABLE system.test**

**(c NUMBER, C2 CHAR(4)) TABLESPACE system;**

2

Table created.

SQL> **INSERT INTO system.test VALUES (1,'Test');**

1 row created.

SQL> **COMMIT;**

Commit complete.

SQL>

##### A2/ Yes, it is possible because the operation affects only the data of the Oracle- managed tablespace and because the tablespace is an Oracle-managed tablespace.

*Q3/ Can you change the encryption algorithm for the tablespace USERS and create tables, insert data in the tablespace USERS?*

SQL> **ALTER TABLESPACE users ENCRYPTION DECRYPT;**

ALTER TABLESPACE users ENCRYPTION DECRYPT

\*

ERROR at line 1:

ORA-28365: wallet is not open

SQL> **CREATE TABLE system.test2**

**(c NUMBER, C2 CHAR(4)) TABLESPACE users;**

CREATE TABLE system.test2(c NUMBER, C2 CHAR(4)) TABLESPACE users

\*

ERROR at line 1:

ORA-28365: wallet is not open

SQL>

##### A3/ No, it not possible in either case because operations on user-managed tablespaces still raise the ORA-28365 "wallet is not open" error when the CDB root keystore is closed.

*Q4/ Is the behavior still the same in PDBs?*

SQL> **CONNECT sys@PDB1 AS SYSDBA**

Enter password: *password*

Connected.

SQL> **SELECT tablespace\_name, encrypted FROM dba\_tablespaces;**

TABLESPACE\_NAME

ENC

SYSTEM SYSAUX UNDOTBS1 TEMP

USERS

NO NO NO NO

NO

OMTS\_TBS

YES

6 rows selected.

SQL>

* 1. Reopen the CDB root keystore.

SQL> **CONNECT / AS SYSDBA**

Connected.

SQL> **ADMINISTER KEY MANAGEMENT SET KEYSTORE OPEN IDENTIFIED BY *password* CONTAINER = ALL;**

2

keystore altered.

SQL>

* 1. Encrypt the SYSTEM tablespace.

SQL> **CONNECT sys@PDB1 AS SYSDBA**

Connected.

SQL> **ALTER TABLESPACE system ENCRYPTION USING 'AES192' ENCRYPT;**

Tablespace altered.

SQL> **SELECT tablespace\_name, encrypted FROM dba\_tablespaces;**

TABLESPACE\_NAME

ENC

SYSTEM SYSAUX UNDOTBS1 TEMP USERS

OMTS\_TBS

YES NO NO NO NO

YES

SQL>

* 1. Close the CDB root keystore and therefore the PDBs keystores.

SQL> **CONNECT / AS SYSDBA**

Connected.

SQL> **ADMINISTER KEY MANAGEMENT SET KEYSTORE CLOSE**

**IDENTIFIED BY *password* CONTAINER = ALL;**

2

keystore altered.

SQL>

* 1. Complete metadata and data operations on the Oracle-managed tablespace SYSTEM.

SQL> **CONNECT sys@PDB1 AS SYSDBA**

Enter password: *password*

Connected.

SQL> **ALTER TABLESPACE system ENCRYPTION USING 'AES128' ENCRYPT;**

ALTER TABLESPACE system ENCRYPTION USING 'AES128' ENCRYPT

\*

ERROR at line 1:

ORA-28365: wallet is not open

SQL> **CREATE TABLE system.tab1 (c NUMBER) TABLESPACE system;**

Table created.

SQL> **INSERT INTO system.tab1 VALUES (1);**

1 row created.

SQL> **COMMIT;**

Commit complete.

SQL>

* 1. Complete metadata and data operations on the user-managed tablespace OMTS\_TBS.

SQL> **ALTER TABLESPACE omts\_tbs ENCRYPTION USING 'AES128'**

**ENCRYPT;**

ALTER TABLESPACE omts\_tbs ENCRYPTION USING 'AES128' ENCRYPT

\*

ERROR at line 1:

ORA-28365: wallet is not open

SQL> **CREATE TABLE system.tab2 (c NUMBER) TABLESPACE omts\_tbs;**

CREATE TABLE system.tab2 (c NUMBER) TABLESPACE omts\_tbs

\*

ERROR at line 1:

ORA-28365: wallet is not open

SQL> **EXIT**

$

##### A4/ Yes, the behavior still the same in PDBs.

1. Disable encryption in ORCL. Execute the

/home/oracle/labs/SEC/remove\_TDE\_in\_ORCL.sh shell script.

$ **/home/oracle/labs/SEC/remove\_TDE\_in\_ORCL.sh**

…

$

1. Recreate the database so as not to use TDE. Use the

/home/oracle/labs/admin/recreate\_ORCL.sh shell script.

$ **$HOME/labs/admin/recreate\_ORCL.sh**

…

$

# Practices for Lesson 3: Using Availability Enhancements

## Practices for Lesson 3: Overview

### Overview

In these practices, you will use RMAN to connect to PDBs to use the recovery catalog to back up and restore PDBs. You will also observe the automatic deletion of flashback logs.

## Practice 3-1: Using RMAN to Connect to a PDB and Use the Recovery Catalog

In this practice, you will use RMAN to connect to PDB1 and use the recovery catalog to back up

PDB1.

### Tasks

1. Execute the /home/oracle/labs/admin/cleanup\_PDBs.sh shell script. The shell script drops all PDBs that may have been created by any of the practices in ORCL, and finally re-creates PDB1.

$ **$HOME/labs/admin/cleanup\_PDBs.sh**

…

$

1. Execute the /home/oracle/labs/HA/create\_PDB2.sh shell script. The shell script creates PDB2 in ORCL.

$ **$HOME/labs/HA/create\_PDB2.sh**

…

$

1. Before starting the practice, execute the $HOME/labs/HA/glogin.sh shell script. It sets formatting for all columns selected in queries.

$ **$HOME/labs/HA/glogin.sh**

**…**

$

1. Execute the **/home/oracle/labs/HA/create\_CDB19.sh** to re-create CDB19 and

PDB19. CDB19 will be used as the recovery catalog CDB.

$ **$HOME/labs/HA/create\_CDB19.sh**

…

$

1. To be able to connect to the recovery catalog and to PDB1 as the target database, create a virtual private RMAN catalog (VPC) in PDB19 for groups of databases and users of ORCL, PDB1 and PDB2.
   1. Create the catalog owner in PDB19.

$ **. oraenv**

ORACLE\_SID = [oracle] ? **CDB19**

The Oracle base has been set to /u01/app/oracle

$ **sqlplus system@PDB19**

Enter password : *password*

SQL> **CREATE USER catowner IDENTIFIED BY *password*;**

User created.

SQL> **GRANT create session, recovery\_catalog\_owner, unlimited tablespace TO catowner;**

2

Grant succeeded.

SQL> **EXIT**

$

* 1. Create the RMAN base catalog in PDB19.

$ **rman**

RMAN> **CONNECT CATALOG catowner@PDB19**

recovery catalog database Password: *password*

connected to recovery catalog database RMAN> **CREATE CATALOG;**

recovery catalog created

RMAN> **EXIT**

$

* 1. Register ORCL in the catalog.

$ **. oraenv**

ORACLE\_SID = [CDB19] ? **ORCL**

The Oracle base remains unchanged with value /u01/app/oracle

$ **rman target / catalog catowner@PDB19**

connected to target database: ORCL (DBID=1515130002) recovery catalog database Password: *password* connected to recovery catalog database

RMAN> **REGISTER DATABASE;**

database registered in recovery catalog starting full resync of recovery catalog full resync complete

RMAN> **EXIT**

$

* 1. Execute the $ORACLE\_HOME/rdbms/admin/dbmsrmanvpc.sql script after connecting to the catalog as SYS to grant VPD-required privileges to the base catalog owner.

$ **sqlplus sys@PDB19 AS SYSDBA**

Enter password: *password*

SQL> **@$ORACLE\_HOME/rdbms/admin/dbmsrmanvpc.sql -vpd catowner**

Checking the operating user... Passed

Granting VPD privileges to the owner of the base catalog schema CATOWNER

======================================== VPD SETUP STATUS:

VPD privileges granted successfully!

Connect to RMAN base catalog and perform UPGRADE CATALOG.

$

* 1. Reconnect to RMAN base catalog and perform UPGRADE CATALOG.

$ **rman**

RMAN> **CONNECT CATALOG catowner@PDB19**

recovery catalog database Password: *password*

connected to recovery catalog database

RMAN> **UPGRADE CATALOG;**

recovery catalog owner is CATOWNER

enter UPGRADE CATALOG command again to confirm catalog upgrade RMAN> **UPGRADE CATALOG;**

recovery catalog upgraded to version 19.03.00.00.00 DBMS\_RCVMAN package upgraded to version 19.03.00.00 DBMS\_RCVCAT package upgraded to version 19.03.00.00.

RMAN> **EXIT**

$

1. Create the VPC users, VPC\_PDB1 and VPC\_PDB2 in the catalog who will be given access for the metadata of PDB1 and PDB2, respectively.

$ **sqlplus system@PDB19**

Enter password : *password*

SQL> **CREATE USER vpc\_pdb1 IDENTIFIED BY *password*;**

User created.

SQL> **CREATE USER vpc\_pdb2 IDENTIFIED BY *password*;**

User created.

SQL> **GRANT create session TO vpc\_pdb1, vpc\_pdb2;**

Grant succeeded.

SQL> **EXIT**

$

1. As the base catalog owner, give the VPC users the access for the metadata of PDB1 and

PDB2, respectively.

$ **rman**

RMAN> **CONNECT CATALOG catowner@PDB19**

recovery catalog database Password: *password*

connected to recovery catalog database

RMAN> **GRANT CATALOG FOR PLUGGABLE DATABASE pdb1 TO vpc\_pdb1;**

Grant succeeded.

RMAN> **GRANT CATALOG FOR PLUGGABLE DATABASE pdb2 TO vpc\_pdb2;**

Grant succeeded.

RMAN> **EXIT**

$

1. Connect to the PDB1 target PDB and to the recovery catalog as VPC\_PDB1 user to back up and restore the PDB1 target PDB.

#### $ rman TARGET sys@PDB1 CATALOG vpc\_pdb1@PDB19

target database Password: *password*

connected to target database: ORCL:PDB1 (DBID=4095280305) recovery catalog database Password: *password*

connected to recovery catalog database

RMAN> **BACKUP DATABASE;**

Starting backup at 08-MAY-19 allocated channel: ORA\_DISK\_1

channel ORA\_DISK\_1: SID=276 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=00014

name=/u02/app/oracle/oradata/ORCL/pdb1/ORCL/884FEEE353031076E053 E311ED0A028E/datafile/o1\_mf\_sysaux\_gf3fhrkw\_.dbf

input datafile file number=00013 name=/u02/app/oracle/oradata/ORCL/pdb1/ORCL/884FEEE353031076E053 E311ED0A028E/datafile/o1\_mf\_system\_gf3fhrkq\_.dbf

input datafile file number=00015 name=/u02/app/oracle/oradata/ORCL/pdb1/ORCL/884FEEE353031076E053 E311ED0A028E/datafile/o1\_mf\_undotbs1\_gf3fhrky\_.dbf

channel ORA\_DISK\_1: starting piece 1 at 08-MAY-19 channel ORA\_DISK\_1: finished piece 1 at 08-MAY-19

piece handle=/u03/app/oracle/fast\_recovery\_area/ORCL/884FEEE353031076E 053E311ED0A028E/backupset/2019\_05\_08/o1\_mf\_nnndf\_TAG20190508T112 738\_gf5h9byf\_.bkp tag=TAG20190508T112738 comment=NONE

channel ORA\_DISK\_1: backup set complete, elapsed time: 00:00:07 Finished backup at 08-MAY-19

RMAN> **EXIT**

$

1. Retrieve the tag value from the backup completed above and display the backup set for

PDB1 via the recovery catalog view.

$ **sqlplus catowner@PDB19**

Enter password: *password*

SQL> **SELECT handle FROM rc\_backup\_piece WHERE tag = 'TAG20190508T112738';**

2

HANDLE

/u03/app/oracle/fast\_recovery\_area/ORCL/884FEEE353031076E053E311 ED0A028E/backupset/2019\_05\_08/o1\_mf\_nnndf\_TAG20190508T112738\_gf5 h9byf\_.bkp

SQL> **EXIT**

$

*Q1/ Can the VPC user VPC\_PDB2 backup other PDBs than the PDB for which the VPC user VPC\_PDB2 was granted access to?*

#### $ rman TARGET sys@PDB1 CATALOG vpc\_pdb2@PDB19

target database Password: *password*

connected to target database: ORCL:PDB1 (DBID=4095280305) recovery catalog database Password: *password*

connected to recovery catalog database

RMAN> **BACKUP DATABASE;**

Starting backup at 08-MAY-19

RMAN-00571: =================================================== RMAN-00569: ========== ERROR MESSAGE STACK FOLLOWS ============ RMAN-00571: ===================================================

RMAN-03002: failure of backup command at 05/08/2019 11:29:28 RMAN-03014: implicit resync of recovery catalog failed

RMAN-06004: Oracle error from recovery catalog database: RMAN- 20001: target database not found in recovery catalog

#### RMAN> BACKUP PLUGGABLE DATABASE pdb1;

Starting backup at 08-MAY-19

RMAN-00571: ==================================================== RMAN-00569: =========== ERROR MESSAGE STACK FOLLOWS ============ RMAN-00571: ====================================================

RMAN-03002: failure of backup command at 05/08/2019 11:30:26 RMAN-03014: implicit resync of recovery catalog failed

RMAN-06004: Oracle error from recovery catalog database: RMAN- 20001: target database not found in recovery catalog

RMAN> **EXIT**

$

##### A1/ The VPC user can perform operations only on the target PDB that he has been granted access to.

1. Connect as the catalog owner and revoke the CATALOG FOR PLUGGABLE DATABASE

privilege on PDB1 and PDB2 from the VPC users.

$ **rman CATALOG catowner@PDB19**

recovery catalog database Password: *password*

connected to recovery catalog database

RMAN> **REVOKE CATALOG FOR PLUGGABLE DATABASE pdb1 FROM vpc\_pdb1;**

Revoke succeeded.

RMAN> **REVOKE CATALOG FOR PLUGGABLE DATABASE pdb2 FROM vpc\_pdb2;**

Revoke succeeded. RMAN> **EXIT**

$

1. Verify that the VPC\_PDB1 user cannot back up the PDB1 target PDB via the recovery catalog.

$ **rman TARGET sys@PDB1 CATALOG vpc\_pdb1@PDB19**

target database Password: *password*

connected to target database: ORCL:PDB1 (DBID=4095280305) recovery catalog database Password: *password*

connected to recovery catalog database RMAN> **BACKUP DATABASE;**

Starting backup at 08-MAY-19

RMAN-00571: ==================================================== RMAN-00569: ============ ERROR MESSAGE STACK FOLLOWS =========== RMAN-00571: ====================================================

RMAN-03002: failure of backup command at 05/08/2019 11:31:23 RMAN-03014: implicit resync of recovery catalog failed

RMAN-06428: recovery catalog is not installed

RMAN> **EXIT**

$

1. Drop the recovery catalog in PDB19.

RMAN> **DROP CATALOG;**

recovery catalog owner is CATOWNER

enter DROP CATALOG command again to confirm catalog removal RMAN> **DROP CATALOG;**

recovery catalog dropped

RMAN> **EXIT**

$

$ **rman CATALOG catowner@PDB19**

recovery catalog database Password: *password*

connected to recovery catalog database

## Practice 3-2: Exploring Automatic Deletion of Flashback Logs

### Overview

In this practice, you will observe the automatic deletion of flashback log files. The minimum flashback retention is one hour (60 minutes). Even if a user sets flashback retention to less than an hour, Oracle still considers flashback retention to be one hour. This is partly because Oracle logs flashback metadata in flashback logs approximately every thirty minutes.

### Tasks

1. Verify that ORCL is in FLASHBACK mode and display the flashback retention period.

$ **. oraenv**

ORACLE\_SID = [oracle] ? **ORCL**

The Oracle base has been set to /u01/app/oracle

$ **sqlplus / AS SYSDBA**

SQL> **SELECT flashback\_on FROM v$database;**

FLASHBACK\_ON YES

SQL>

If the database is not in FLASHBACK mode, use the following command to configure it in

FLASHBACK mode.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| SQL> **ALTER DATABASE FLASHBACK ON;**  Database altered.  SQL> **SHOW PARAMETER DB\_FLASHBACK\_RETENTION\_TARGET** | | | | |
| NAME |  | TYPE |  | VALUE |
| db\_flashback\_retention\_target  SQL> |  | integer |  | 70 |

The flashback retention period is set to 70 minutes.

1. Increase the FRA size to ensure that there will be no space pressure that would automatically delete flashback logs.

SQL> **ALTER SYSTEM SET db\_recovery\_file\_dest\_size=100G;**

System altered.

SQL>

1. Check the flashback logs in the FRA.

|  |  |
| --- | --- |
| SQL> **HOST**  $ **cd /u03/app/oracle/fast\_recovery\_area/ORCL/flashback**  $ **ls -l** | |
| total 614424 |  |
| -rw-r----- 1 oracle oinstall 209723392 May | 9 05:28 |
| o1\_mf\_gf601ksm\_.flb |  |
| -rw-r----- 1 oracle oinstall 209723392 May | 9 07:33 |
| o1\_mf\_gf601mrc\_.flb |  |
| -rw-r----- 1 oracle oinstall 209723392 May | 9 05:28 |
| o1\_mf\_gf7gnf6g\_.flb |  |
| $ |  |

1. In another terminal window, execute the /home/oracle/labs/HA/workload.sh to generate flashback logs. The script takes a long time to complete. Continue to step 5 then step 6 to complete job after starting task 4.

$ **. oraenv**

ORACLE\_SID = [oracle] ? **ORCL**

The Oracle base has been set to /u01/app/oracle

$ **/home/oracle/labs/HA/workload.sh**

…

SQL> UPDATE hr.tabflash SET c1=c1+12; 17895424 rows updated.

SQL> COMMIT;

Commit complete.

SQL> EXIT

…

$

1. Back in the initial terminal session, check again the flashback logs in the FRA. As we know that Oracle logs flashback metadata in flashback logs approximately every thirty minutes, it is not pertinent to check the list before. Nevertheless due to limit resource issue, go directly to task 6.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| $ **ls -l** |  | | | | |
| total 3481748 |
| -rw-r----- 1 oracle o1\_mf\_gf601ksm\_.flb | oinstall | 209723392 | May | 9 | 07:46 |
| -rw-r----- 1 oracle o1\_mf\_gf601mrc\_.flb | oinstall | 209723392 | May | 9 | 07:45 |
| -rw-r----- 1 oracle o1\_mf\_gf7gnf6g\_.flb | oinstall | 209723392 | May | 9 | 07:46 |
| -rw-r----- 1 oracle o1\_mf\_gf7pp3rb\_.flb | oinstall | 209723392 | May | 9 | 07:47 |
| -rw-r----- 1 oracle o1\_mf\_gf7ppxms\_.flb | oinstall | 209723392 | May | 9 | 07:47 |
| -rw-r----- 1 oracle o1\_mf\_gf7pqtcl\_.flb | oinstall | 209723392 | May | 9 | 07:48 |
| -rw-r----- 1 oracle o1\_mf\_gf7prq9x\_.flb | oinstall | 209723392 | May | 9 | 07:48 |
| -rw-r----- 1 oracle o1\_mf\_gf7psl1o\_.flb | oinstall | 209723392 | May | 9 | 07:48 |
| -rw-r----- 1 oracle o1\_mf\_gf7ptdsb\_.flb | oinstall | 209723392 | May | 9 | 07:49 |
| -rw-r----- 1 oracle o1\_mf\_gf7pv7of\_.flb | oinstall | 209723392 | May | 9 | 07:49 |
| -rw-r----- 1 oracle o1\_mf\_gf7pw1ph\_.flb | oinstall | 209723392 | May | 9 | 07:50 |
| -rw-r----- 1 oracle o1\_mf\_gf7pww55\_.flb | oinstall | 209723392 | May | 9 | 07:51 |
| -rw-r----- 1 oracle o1\_mf\_gf7pxstt\_.flb | oinstall | 209723392 | May | 9 | 07:52 |
| -rw-r----- 1 oracle o1\_mf\_gf7q0ytp\_.flb | oinstall | 209723392 | May | 9 | 07:53 |
| -rw-r----- 1 oracle o1\_mf\_gf7q1ypl\_.flb | oinstall | 209723392 | May | 9 | 07:53 |
| -rw-r----- 1 oracle o1\_mf\_gf7q2wkj\_.flb | oinstall | 209723392 | May | 9 | 07:53 |
| -rw-r----- 1 oracle o1\_mf\_gf7q3wm2\_.flb | oinstall | 209723392 | May | 9 | 07:53 |
| $ **ls -ltr | wc -l** |  |  |  |  |  |
| 18 |  |  |  |  |  |
| $ **exit** |  |  |  |  |  |
| SQL> |  |  |  |  |  |

1. Decrease the flashback retention period to 60 minutes. From the time you decrease the flashback retention period, you will observe that the list of flashback logs will remain stable and not increase any more.

SQL> **ALTER SYSTEM SET**

**DB\_FLASHBACK\_RETENTION\_TARGET=60 SCOPE=BOTH;**

2

System altered.

SQL>

1. Check regularly the flashback logs in the FRA. Meanwhile, let the instructor teach the lesson about Performance enhancements.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| SQL> **EXIT**  $ **cd /u03/app/oracle/fast\_recovery\_area/ORCL/flashback**  $ **ls -ltr** | | | | | |
| total 3891364 |  |  |  |  |  |
| -rw-r----- 1 oracle o1\_mf\_gf601mrc\_.flb | oinstall | 209723392 | May | 9 | 07:45 |
| -rw-r----- 1 oracle o1\_mf\_gf7gnf6g\_.flb | oinstall | 209723392 | May | 9 | 07:46 |
| -rw-r----- 1 oracle o1\_mf\_gf601ksm\_.flb | oinstall | 209723392 | May | 9 | 07:46 |
| -rw-r----- 1 oracle o1\_mf\_gf7pp3rb\_.flb | oinstall | 209723392 | May | 9 | 07:47 |
| -rw-r----- 1 oracle o1\_mf\_gf7ppxms\_.flb | oinstall | 209723392 | May | 9 | 07:47 |
| -rw-r----- 1 oracle o1\_mf\_gf7pqtcl\_.flb | oinstall | 209723392 | May | 9 | 07:48 |
| -rw-r----- 1 oracle o1\_mf\_gf7prq9x\_.flb | oinstall | 209723392 | May | 9 | 07:48 |
| -rw-r----- 1 oracle o1\_mf\_gf7psl1o\_.flb | oinstall | 209723392 | May | 9 | 07:48 |
| -rw-r----- 1 oracle o1\_mf\_gf7ptdsb\_.flb | oinstall | 209723392 | May | 9 | 07:49 |
| -rw-r----- 1 oracle o1\_mf\_gf7pv7of\_.flb | oinstall | 209723392 | May | 9 | 07:49 |
| -rw-r----- 1 oracle o1\_mf\_gf7pw1ph\_.flb | oinstall | 209723392 | May | 9 | 07:50 |
| -rw-r----- 1 oracle o1\_mf\_gf7pww55\_.flb | oinstall | 209723392 | May | 9 | 07:51 |
| -rw-r----- 1 oracle o1\_mf\_gf7pxstt\_.flb | oinstall | 209723392 | May | 9 | 07:52 |
| -rw-r----- 1 oracle o1\_mf\_gf7q0ytp\_.flb | oinstall | 209723392 | May | 9 | 07:53 |
| -rw-r----- 1 oracle o1\_mf\_gf7q1ypl\_.flb | oinstall | 209723392 | May | 9 | 07:53 |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| -rw-r----- 1 oracle o1\_mf\_gf7q2wkj\_.flb | oinstall | 209723392 | May | 9 | 07:54 |
| -rw-r----- 1 oracle o1\_mf\_gf7q3wm2\_.flb | oinstall | 209723392 | May | 9 | 07:54 |
| -rw-r----- 1 oracle o1\_mf\_gf7q5mnb\_.flb | oinstall | 209723392 | May | 9 | 07:54 |
| -rw-r----- 1 oracle o1\_mf\_gf7q4sgt\_.flb | oinstall | 209723392 | May | 9 | 07:54 |
| $ **ls -ltr | wc -l** |  |  |  |  |  |
| **20** |  |  |  |  |  |
| $ |  |  |  |  |  |
|  |  |  |  |  |  |
| $ **ls -ltr** |  |  |  |  |  |
| total 6553888 |  |  |  |  |  |
| -rw-r----- 1 oracle o1\_mf\_gf601mrc\_.flb | oinstall | 209723392 | May | 9 | 07:45 |
| -rw-r----- 1 oracle o1\_mf\_gf7gnf6g\_.flb | oinstall | 209723392 | May | 9 | 07:46 |
| -rw-r----- 1 oracle o1\_mf\_gf601ksm\_.flb | oinstall | 209723392 | May | 9 | 07:46 |
| -rw-r----- 1 oracle o1\_mf\_gf7pp3rb\_.flb | oinstall | 209723392 | May | 9 | 07:47 |
| -rw-r----- 1 oracle o1\_mf\_gf7ppxms\_.flb | oinstall | 209723392 | May | 9 | 07:47 |
| -rw-r----- 1 oracle o1\_mf\_gf7pqtcl\_.flb | oinstall | 209723392 | May | 9 | 07:48 |
| -rw-r----- 1 oracle o1\_mf\_gf7prq9x\_.flb | oinstall | 209723392 | May | 9 | 07:48 |
| -rw-r----- 1 oracle o1\_mf\_gf7psl1o\_.flb | oinstall | 209723392 | May | 9 | 07:48 |
| -rw-r----- 1 oracle o1\_mf\_gf7ptdsb\_.flb | oinstall | 209723392 | May | 9 | 07:49 |
| -rw-r----- 1 oracle o1\_mf\_gf7pv7of\_.flb | oinstall | 209723392 | May | 9 | 07:49 |
| -rw-r----- 1 oracle o1\_mf\_gf7pw1ph\_.flb | oinstall | 209723392 | May | 9 | 07:50 |
| -rw-r----- 1 oracle o1\_mf\_gf7pww55\_.flb | oinstall | 209723392 | May | 9 | 07:51 |
| -rw-r----- 1 oracle o1\_mf\_gf7pxstt\_.flb | oinstall | 209723392 | May | 9 | 07:52 |
| -rw-r----- 1 oracle o1\_mf\_gf7q0ytp\_.flb | oinstall | 209723392 | May | 9 | 07:53 |
| -rw-r----- 1 oracle o1\_mf\_gf7q1ypl\_.flb | oinstall | 209723392 | May | 9 | 07:53 |
| -rw-r----- 1 oracle o1\_mf\_gf7q2wkj\_.flb | oinstall | 209723392 | May | 9 | 07:54 |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| -rw-r----- 1 oracle | oinstall | 209723392 | May | 9 | 07:54 |
| o1\_mf\_gf7q3wm2\_.flb |  |  |  |  |  |
| -rw-r----- 1 oracle o1\_mf\_gf7q4sgt\_.flb | oinstall | 209723392 | May | 9 | 07:56 |
| -rw-r----- 1 oracle o1\_mf\_gf7q5mnb\_.flb | oinstall | 209723392 | May | 9 | 07:56 |
| -rw-r----- 1 oracle o1\_mf\_gf7q8vnb\_.flb | oinstall | 209723392 | May | 9 | 07:57 |
| -rw-r----- 1 oracle o1\_mf\_gf7q9ohv\_.flb | oinstall | 209723392 | May | 9 | 08:01 |
| -rw-r----- 1 oracle o1\_mf\_gf7qblcq\_.flb | oinstall | 209723392 | May | 9 | 08:01 |
| -rw-r----- 1 oracle o1\_mf\_gf7qlwv2\_.flb | oinstall | 209723392 | May | 9 | 08:02 |
| -rw-r----- 1 oracle o1\_mf\_gf7qmhcd\_.flb | oinstall | 209723392 | May | 9 | 08:02 |
| -rw-r----- 1 oracle o1\_mf\_gf7qn992\_.flb | oinstall | 209723392 | May | 9 | 08:03 |
| -rw-r----- 1 oracle o1\_mf\_gf7qnx9s\_.flb | oinstall | 209723392 | May | 9 | 08:04 |
| -rw-r----- 1 oracle o1\_mf\_gf7qpd92\_.flb | oinstall | 209723392 | May | 9 | 08:04 |
| -rw-r----- 1 oracle o1\_mf\_gf7qqv6b\_.flb | oinstall | 209723392 | May | 9 | 08:05 |
| -rw-r----- 1 oracle o1\_mf\_gf7qs75b\_.flb | oinstall | 209723392 | May | 9 | 08:06 |
| -rw-r----- 1 oracle o1\_mf\_gf7qtp80\_.flb | oinstall | 209723392 | May | 9 | 08:07 |
| -rw-r----- 1 oracle o1\_mf\_gf7qxggn\_.flb | oinstall | 209723392 | May | 9 | 08:07 |
| -rw-r----- 1 oracle o1\_mf\_gf7qw2fl\_.flb | oinstall | 209723392 | May | 9 | 08:07 |
| $ **ls -ltr | wc -l** |  |  |  |  |  |
| **35** |  |  |  |  |  |
| $ |  |  |  |  |  |
|  |  |  |  |  |  |
| $ **ls -ltr** |  |  |  |  |  |
| total 6789436 |  |  |  |  |  |
| -rw-r----- 1 oracle o1\_mf\_gf601mrc\_.flb | oinstall | 209723392 | May | 9 | 07:45 |
| -rw-r----- 1 oracle o1\_mf\_gf7gnf6g\_.flb | oinstall | 209723392 | May | 9 | 07:46 |
| -rw-r----- 1 oracle o1\_mf\_gf601ksm\_.flb | oinstall | 209723392 | May | 9 | 07:46 |
| -rw-r----- 1 oracle o1\_mf\_gf7pp3rb\_.flb | oinstall | 209723392 | May | 9 | 07:47 |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| -rw-r----- 1 oracle o1\_mf\_gf7ppxms\_.flb | oinstall | 209723392 | May | 9 | 07:47 |
| -rw-r----- 1 oracle o1\_mf\_gf7pqtcl\_.flb | oinstall | 209723392 | May | 9 | 07:48 |
| -rw-r----- 1 oracle o1\_mf\_gf7prq9x\_.flb | oinstall | 209723392 | May | 9 | 07:48 |
| -rw-r----- 1 oracle o1\_mf\_gf7psl1o\_.flb | oinstall | 209723392 | May | 9 | 07:48 |
| -rw-r----- 1 oracle o1\_mf\_gf7ptdsb\_.flb | oinstall | 209723392 | May | 9 | 07:49 |
| -rw-r----- 1 oracle o1\_mf\_gf7pv7of\_.flb | oinstall | 209723392 | May | 9 | 07:49 |
| -rw-r----- 1 oracle o1\_mf\_gf7pw1ph\_.flb | oinstall | 209723392 | May | 9 | 07:50 |
| -rw-r----- 1 oracle o1\_mf\_gf7pww55\_.flb | oinstall | 209723392 | May | 9 | 07:51 |
| -rw-r----- 1 oracle o1\_mf\_gf7pxstt\_.flb | oinstall | 209723392 | May | 9 | 07:52 |
| -rw-r----- 1 oracle o1\_mf\_gf7q0ytp\_.flb | oinstall | 209723392 | May | 9 | 07:53 |
| -rw-r----- 1 oracle o1\_mf\_gf7q1ypl\_.flb | oinstall | 209723392 | May | 9 | 07:53 |
| -rw-r----- 1 oracle o1\_mf\_gf7q2wkj\_.flb | oinstall | 209723392 | May | 9 | 07:54 |
| -rw-r----- 1 oracle o1\_mf\_gf7q3wm2\_.flb | oinstall | 209723392 | May | 9 | 07:54 |
| -rw-r----- 1 oracle o1\_mf\_gf7q4sgt\_.flb | oinstall | 209723392 | May | 9 | 07:56 |
| -rw-r----- 1 oracle o1\_mf\_gf7q5mnb\_.flb | oinstall | 209723392 | May | 9 | 07:56 |
| -rw-r----- 1 oracle o1\_mf\_gf7q8vnb\_.flb | oinstall | 209723392 | May | 9 | 07:57 |
| -rw-r----- 1 oracle o1\_mf\_gf7q9ohv\_.flb | oinstall | 209723392 | May | 9 | 08:01 |
| -rw-r----- 1 oracle o1\_mf\_gf7qblcq\_.flb | oinstall | 209723392 | May | 9 | 08:01 |
| -rw-r----- 1 oracle o1\_mf\_gf7qlwv2\_.flb | oinstall | 209723392 | May | 9 | 08:02 |
| -rw-r----- 1 oracle o1\_mf\_gf7qmhcd\_.flb | oinstall | 209723392 | May | 9 | 08:02 |
| -rw-r----- 1 oracle o1\_mf\_gf7qn992\_.flb | oinstall | 209723392 | May | 9 | 08:03 |
| -rw-r----- 1 oracle o1\_mf\_gf7qnx9s\_.flb | oinstall | 209723392 | May | 9 | 08:04 |
| -rw-r----- 1 oracle o1\_mf\_gf7qpd92\_.flb | oinstall | 209723392 | May | 9 | 08:04 |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| -rw-r----- 1 oracle o1\_mf\_gf7qqv6b\_.flb | oinstall | 209723392 | May | 9 | 08:05 |
| -rw-r----- 1 oracle o1\_mf\_gf7qs75b\_.flb | oinstall | 209723392 | May | 9 | 08:06 |
| -rw-r----- 1 oracle o1\_mf\_gf7qtp80\_.flb | oinstall | 209723392 | May | 9 | 08:07 |
| -rw-r----- 1 oracle o1\_mf\_gf7qw2fl\_.flb | oinstall | 209723392 | May | 9 | 08:07 |
| -rw-r----- 1 oracle o1\_mf\_gf7qxggn\_.flb | oinstall | 209723392 | May | 9 | 08:08 |
| -rw-r----- 1 oracle o1\_mf\_gf7qz0w1\_.flb | oinstall | 31473664 | May | 9 | 08:08 |
| -rw-r----- 1 oracle o1\_mf\_gf7qyhc9\_.flb | oinstall | 209723392 | May | 9 | 08:13 |
| $ **ls -ltr | wc -l** |  |  |  |  |  |
| 35 |  |  |  |  |  |
| $ |  |  |  |  |  |

*Q1/ Are flashback logs deleted even if there is still enough space in FRA?*

##### A1/ Yes. Flashback logs beyond retention period are proactively deleted without degrading the flashback performance and before there is space pressure. The number of flashback logs remains the same after decreasing the flashback retention.

1. In the other terminal window, if the /home/oracle/labs/HA/workload.sh did not complete, you can interrupt it by killing the shell script.

$ **pgrep -lf workload**

1366 workload.sh

$ **kill -9 1366**

$

# Practices for Lesson 4: Using Performance Enhancements

## Practices for Lesson 4: Overview

### Overview

In these practices, you will use Memoptimized Rowstore - Fast Ingest, complete a PDB-level ADDM analysis, use Real-Time SQL Monitor as a SQL developer, populate in-memory objects and wait until they are populated, and finally configure and use PDB- level Database Replay.

## Practice 4-1: Using Memoptimized Rowstore - Fast Ingest

### Overview

In this practice, you will see how Memoptimized Rowstore - Fast Ingest deferred inserts are handled in the SGA and on disk through the Space Management Coordinator (SMCO) and W*xxx* slave background processes, how deferred inserted rows are different from the conventional inserts.

### Tasks

If you are working on an on-premise database, you will observe that the feature is not available on on-premise environment.

If you are working with a Cloud database, you would be able to proceed with the following steps.

1. Execute the /home/oracle/labs/admin/cleanup\_PDBs.sh shell script. The shell script drops all PDBs that may have been created by any of the practices in ORCL, and finally re-creates PDB1. The session you are logged into is called *Session1*.

$ **$HOME/labs/admin/cleanup\_PDBs.sh**

…

$

1. Before starting the practice, execute the $HOME/labs/PERF/glogin.sh shell script. It sets formatting for all columns selected in queries.

$ **$HOME/labs/PERF/glogin.sh**

**…**

$

1. Create the HR.MEMOPTWRITES table in PDB1 to have rows inserted as deferred inserts. Ensure that the table data is written to the space allocated for fast ingest writes in the large pool in the shared pool area before being written to disk.
   1. Create the table with the appropriate attribute.

$ **sqlplus system@PDB1**

Enter password : *password*

SQL> **CREATE TABLE hr.memoptwrites**

**(c1 NUMBER, c2 VARCHAR2(12)) MEMOPTIMIZE FOR WRITE;**

2 CREATE TABLE hr.memoptwrites

\*

ERROR at line 1:

ORA-62145: MEMOPTIMIZE FOR WRITE feature not allowed on segment with deferred storage.

SQL>

*Q1/ Which initialization parameter does the error refer to?*

##### A1/ By default, an object created like a table does not have a segment created until a first row is inserted. MEMOPTIMIZE FOR WRITE tables require a segment created before the first row is inserted.

SQL> **SHOW PARAMETER deferred\_segment\_creation**

NAME

TYPE

VALUE

deferred\_segment\_creation

Boolean

TRUE

SQL> **ALTER SYSTEM SET deferred\_segment\_creation = FALSE SCOPE=BOTH;**

2

System altered.

SQL> **CREATE TABLE hr.memoptwrites**

**(c1 NUMBER, c2 VARCHAR2(12)) MEMOPTIMIZE FOR WRITE;**

2

Table created.

SQL>

* 1. Verify that the attribute is set.

SQL> **SELECT memoptimize\_read Mem\_read,**

**memoptimize\_write Mem\_write FROM dba\_tables**

**WHERE table\_name = 'MEMOPTWRITES';**

2 3 4

MEM\_READ MEM\_WRIT

DISABLED ENABLED SQL>

*Q2/ Is the space allocated for fast ingest writes in the large pool initialized?*

SQL> **SELECT \* FROM V$MEMOPTIMIZE\_WRITE\_AREA;**

TOTAL\_SIZE USED\_SPACE FREE\_SPACE NUM\_WRITES NUM\_WRITERS CON\_ID

0

0

0

0

0

3

SQL>

##### A2/ No, it is not yet initialized. It waits for the first inserted row.

* 1. Insert a row into the table so that the row goes to the space allocated for fast ingest writes in the large pool.

SQL> **COMMIT;**

Commit complete. SQL>

SQL> **INSERT /\*+ MEMOPTIMIZE\_WRITE \*/ INTO hr.memoptwrites VALUES (1, 'Memoptwrites');**

2

1 row created.

*Q3/ How many bytes are consumed in the space allocated for fast ingest writes in the large pool for the inserted row?*

##### A3/ 371195904 bytes initialize the space allocated for fast ingest writes in the large pool. 1076816 bytes are consumed in the space allocated for fast ingest writes in the large pool for the inserted row by one writer.

SQL> **SELECT \* FROM V$MEMOPTIMIZE\_WRITE\_AREA;**

TOTAL\_SIZE USED\_SPACE FREE\_SPACE NUM\_WRITES NUM\_WRITERS CON\_ID

371195904

1076816 370119088

0

1

3

SQL>

*Q4/ Why is only 1G allocated when 2G is the default value of allocation?*

##### A4/ 2G is allocated from the large pool when there is enough space. If there is not enough space in the large pool, an ORA-4031 is internally discovered and automatically cleared. The allocation is retried with half the requested size. In our case, the allocation succeeds at 1G. If there was not enough space in the large pool, the allocation could have been tried with 512M and 256M after which the feature would be disabled until the instance was restarted.

***After the space allocated for fast ingest writes in the large pool is initialized, the size remains static. It cannot grow or shrink.***

* 1. In another session, called *Session2*, insert and commit rows into the table by executing the /home/oracle/labs/PERF/insert.sql SQL script.

$ **sqlplus system@PDB1**

Enter password : *password*

SQL> **@/home/oracle/labs/PERF/insert.sql**

… SQL>

*Q5/ How many bytes are consumed in the space allocated for fast ingest writes in the large pool for the inserted rows? What do you observe (in Session1)?*

##### A5/ 1076816 bytes are consumed for the inserted rows by two writers. The same amount of space is used despite the number of rows inserted.

SQL> **SELECT \* FROM V$MEMOPTIMIZE\_WRITE\_AREA;**

TOTAL\_SIZE USED\_SPACE FREE\_SPACE NUM\_WRITES NUM\_WRITERS CON\_ID

371195904

1076816 370119088

0

2

3

SQL>

* 1. In each session, list the statistics about memoptimized writes.
     1. In *Session2*:

SQL> **SELECT display\_name, value FROM v$mystat m, v$statname n WHERE m.statistic# = n.statistic#**

**AND display\_name IN ( 'memopt w rows written',**

**'memopt w rows flushed');**

2

DISPLAY\_NAME VALUE

memopt w rows written

memopt w rows flushed

33

0

SQL>

* + 1. In *Session1*:

SQL> S**ELECT display\_name, value FROM v$mystat m, v$statname n**

**WHERE m.statistic# = n.statistic#**

*Q6/ How many writes were written via the space allocated for fast ingest writes in the large pool in Session1? And how many in Session2?*

**AND**

**display\_name IN ( 'memopt w rows written',**

**'memopt w rows flushed');**

2

DISPLAY\_NAME VALUE

memopt w rows written

memopt w rows flushed

1

0

SQL>

##### A1/ One row in Session1 and 33 in Session2.

*Q7/ Is the space used in the space allocated for fast ingest writes in the large pool released after a certain time? You can check in either session.*

##### A7/ Yes. Re-execute the query after a few minutes. The background processes, w000 - w999 slaves, which have SMCO as the coordinator process, flush the data from the space allocated for fast ingest writes in the large pool to data files after 1MB worth of writes (per session per object) or after 60 seconds.

SQL> **SELECT \* FROM V$MEMOPTIMIZE\_WRITE\_AREA;**

TOTAL\_SIZE USED\_SPACE FREE\_SPACE NUM\_WRITES NUM\_WRITERS CON\_ID

371195904

28320 371167584

0

2

3

SQL>

1. In *Session2*, read the content of the table.

SQL> **SELECT distinct c1 FROM hr.memoptwrites;**

C1 6

1

2

4

5

3

6 rows selected.

SQL>

1. In *Session1*, execute the /home/oracle/labs/PERF/insert\_before\_flush.sql

SQL script to insert and commit more rows into the HR.MEMOPTWRITES table.

SQL> **@/home/oracle/labs/PERF/insert\_before\_flush.sql**

… SQL>

*Q8/ In Session2, does HR see all the rows inserted?*

SQL> **SELECT distinct c1 FROM hr.memoptwrites;**

C1 6

1

2

4

5

3

6 rows selected.

SQL>

##### A8/ No. Any buffered data in the space allocated for fast ingest writes in the large pool cannot be read by any session, including the writer, until the background process sweep is complete, even if the data was committed.

* 1. In *Session1*, either wait for the background process to flush the space allocated for fast ingest writes in the large pool data or manually flush the data from space allocated for fast ingest writes in the large pool to disk.

SQL> **EXEC DBMS\_MEMOPTIMIZE\_ADMIN.WRITES\_FLUSH**

PL/SQL procedure successfully completed. SQL>

*Q9/ Can HR display the rows inserted now in Session2?*

SQL> **SELECT distinct c1 FROM** hr.**memoptwrites;** C1

6

1

7

2

8

11

4

5

10

3

9

11 rows selected.

SQL>

##### A9/ Yes, because the Wxxx background process flushed the data from the space allocated for fast ingest writes in the large pool to disk.

1. In *Session2*, observe how constraints are evaluated on tables that have rows inserted as deferred inserts.
   1. Create the HR.MEMOPTW table in PDB1 to have rows inserted as deferred inserts and a check constraint on C1. The values must be within the range of 1 to 10.
      1. Create the table.

SQL> **CREATE TABLE hr.memoptw**

**( c1 NUMBER(3), c2 VARCHAR2(12),**

**CONSTRAINT CC\_CHECK CHECK (c1 BETWEEN 1 AND 10)) MEMOPTIMIZE FOR WRITE;**

2 3 4

Table created.

SQL>

* + 1. Insert rows into the table by executing the following command:

SQL> **INSERT /\*+ MEMOPTIMIZE\_WRITE \*/ INTO hr.memoptw**

**VALUES (0,'Memoptw');**

INSERT /\*+ MEMOPTIMIZE\_WRITE \*/ INTO hr.memoptw VALUES

(0,'Memoptw')

\*

ERROR at line 1:

ORA-02290: check constraint (HR.CC\_CHECK) violated

SQL>

*Q10/ Does the INSERT statement successfully complete?*

##### A10/ No. The constraint is evaluated without looking at the existing data on disk and, therefore, is still honored in the foreground process.

* 1. Create the HR.MEMOPTW2 table in PDB1 to have rows inserted as deferred inserts and a UNIQUE constraint on C2.
     1. Create the table.

SQL> **CREATE TABLE hr.memoptw2**

**(c1 NUMBER(3), c2 VARCHAR2(12) CONSTRAINT un\_c2 UNIQUE) MEMOPTIMIZE FOR WRITE;**

2 3

Table created.

SQL>

* + 1. Insert rows with the same value for C2 into the table by executing the following command:

SQL> **INSERT /\*+ MEMOPTIMIZE\_WRITE \*/ INTO hr.memoptw2 VALUES (0,'Memoptw');**

2

1 row created.

SQL> **INSERT /\*+ MEMOPTIMIZE\_WRITE \*/ INTO hr.memoptw2 VALUES (1,'Memoptw');**

2

1 row created.

SQL>

*Q11/ Do the INSERT statements successfully complete?*

##### A11/ Yes. The UNIQUE constraint is evaluated when the insert is written to disk and, therefore, is deferred to the time of drainage to disk.

* + 1. In *Session1*, flush the data from the space allocated for fast ingest writes in the large pool to disk.

SQL> **EXEC DBMS\_MEMOPTIMIZE\_ADMIN.WRITES\_FLUSH**

PL/SQL procedure successfully completed. SQL> **EXIT**

$

* + 1. Commit the insert in *Session2*.

SQL> **COMMIT;**

Commit complete.

SQL>

*Q12/ Is the data successfully written to disk?*

SQL> **SELECT \* FROM hr.memoptw2;**

C1 C2

0 Memoptw SQL> **EXIT**

$

***A12/ No. The UNIQUE constraint is evaluated when the insert is written to disk. As such, the second row is not inserted when the data is written to disk from the space allocated for fast ingest writes in the large pool.***

## Practice 4-2: Completing an ADDM Analysis Inside a PDB

### Overview

In this practice you will perform an ADDM analysis and recommendations for PDB1. It is better to execute PDB-specific ADDM analysis and recommendations than ADDM analysis on the CDB root.

### Tasks

1. Before starting the practice, execute the $HOME/labs/PERF/glogin.sh shell script. It sets formatting for all columns selected in queries.

$ **$HOME/labs/PERF/glogin.sh**

**…**

$

1. Log in to PDB1. You are connected in *Session1*.

$ **. oraenv**

ORACLE\_SID = [CDB19] ? **ORCL**

The Oracle base has been set to /u01/app/oracle

$ **sqlplus sys@PDB1 AS SYSDBA**

Enter password : *password*

SQL>

1. In another terminal session called *Session2*, launch a workload on PDB1 by executing the

/home/oracle/labs/PERF/start\_workload.sh shell script. The workload continues until you remove the /home/oracle/labs/PERF/runload file.

$ **. oraenv**

ORACLE\_SID = [ORCL] ? **ORCL**

The Oracle base has been set to /u01/app/oracle

$ **cd /home/oracle/labs/PERF**

$ **./start\_workload.sh 1 PDB1**

… SQL>

1. While the workload executes in PDB1 in *Session2*, after about 2 minutes, create a snapshot in *Session1*.

SQL> **EXEC dbms\_workload\_repository.create\_snapshot()**

PL/SQL procedure successfully completed. SQL>

1. Execute the ADDM task manually in *Session1*.

SQL> **VAR tname VARCHAR2(60)**

SQL> **BEGIN**

**:tname := 'PDB1\_analysis\_mode\_task'; DBMS\_ADDM.ANALYZE\_DB( :tname, 1, 2); END;**

**/**

2 3 4 5

PL/SQL procedure successfully completed.

SQL>

1. View the PDB report in *Session1*. You can then schedule the task repetitively.

SQL> **SET PAGESIZE 50000**

SQL> **SELECT dbms\_addm.get\_report(:tname) FROM DUAL;**

DBMS\_ADDM.GET\_REPORT(:TNAME)

ADDM Report for Task 'PDB1\_analysis\_mode\_task'

Analysis Period

AWR snapshot range from 1 to 2.

Time period starts at 15-MAY-19 08.56.46 AM Time period ends at 15-MAY-19 08.58.55 AM

Analysis Target

Database 'ORCL' with DB ID **4095280305**. Database version 19.0.0.0.0.

Analysis was requested for all instances, but ADDM analyzed instance ORCL, numbered 1 and hosted at edvmr1p0.

See the "Additional Information" section for more information on the requested instances.

**ADDM detected that the system is a PDB**.

Activity During the Analysis Period Total database time was 12 seconds.

The average number of active sessions was .1. ADDM analyzed 1 of the requested 1 instances.

~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~ There are no findings to report.

~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~ Additional Information

Miscellaneous Information

There was no significant database activity to run the ADDM.

SQL>

*Q1/ Are the recommendations related to the PDB level only?*

##### A1/ Yes, they are. There are no recommendations at the CDB root level. The DB ID 4095280305 is the database ID of PDB1.

*Q2/ Which is the type of CDB ADDM thinks it was for analysis?*

##### A2/ The report displays: ADDM detected that the CDB type is PDB.

1. In *Session2*, stop the workload by removing the /home/oracle/labs/PERF/runload

SQL> **SELECT task\_name, cdb\_type\_detected FROM dba\_addm\_tasks WHERE how\_created = 'CMD';**

2

TASK\_NAME CDB\_TYPE\_DETECTED

PDB1\_analysis\_mode\_task

PDB

SQL>

file.

$ **rm /home/oracle/labs/PERF/runload**

$

1. In *Session2*, launch a workload on PDB1 again by executing the

/home/oracle/labs/PERF/start\_workload.sh shell script.

$ **/home/oracle/labs/PERF/start\_workload.sh 1 PDB1**

…

1. In *Session1*, connect to the CDB root.

SQL> **CONNECT / AS SYSDBA**

Connected.

SQL>

1. While the workload executes in PDB1 in *Session2*, after about 2 minutes, create a snapshot in *Session1*.

SQL> **EXEC dbms\_workload\_repository.create\_snapshot()**

PL/SQL procedure successfully completed. SQL>

1. Execute ADDM task manually in *Session1*. Retrieve recommendations for the CDB root only. First retrieve the snapshots values to analyze.

SQL> **SELECT min(snap\_id),max(snap\_id) FROM awr\_cdb\_snapshot;**

MIN(SNAP\_ID) MAX(SNAP\_ID)

1

121

SQL> **VAR tname VARCHAR2(60)**

SQL> **BEGIN**

**:tname := 'CDB analysis\_mode\_task'; DBMS\_ADDM.ANALYZE\_DB( :tname, 1, 2); END;**

**/**

> > > >

PL/SQL procedure successfully completed.

SQL>

1. View the PDB report in *Session1*. You can then schedule the task repetitively.

SQL> **SELECT DBMS\_ADDM.GET\_REPORT(:tname) FROM DUAL;**

DBMS\_ADDM.GET\_REPORT(:TNAME)

ADDM Report for Task 'CDB analysis\_mode\_task'

Analysis Period

AWR snapshot range from 1 to 2.

Time period starts at 15-MAY-19 07.00.57 AM

Time period ends at 15-MAY-19 08.00.10 AM Analysis Target

Database 'ORCL' with DB ID 1517631634. Database version 19.3.0.0.0.

Analysis was requested for all instances, but ADDM analyzed instance ORCL,numbered 1 and hosted at edvmr1p0.

See the "Additional Information" section for more information on the requested instances.

**ADDM detected that the database type is MULTITENANT DB.**

Activity During the Analysis Period Total database time was 1 seconds.

The average number of active sessions was .0. ADDM analyzed 1 of the requested 1 instances.

~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~ There are no findings to report.

~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~ Additional Information

Miscellaneous Information

There was no significant database activity to run the ADDM. SQL>

*Q1/ Are the recommendations related to the CDB level and PDB levels too?*

##### A1/ There is no recommendation at a specific PDB level.

SQL> **SELECT task\_name, cdb\_type\_detected FROM dba\_addm\_tasks WHERE how\_created = 'CMD';**

2

TASK\_NAME CDB\_TYPE\_DETECTED

CDB analysis\_mode\_task

MULTITENANT DB

SQL> **EXIT**

$

1. In *Session2*, stop the workload by removing the /home/oracle/labs/PERF/runload

file.

$ **rm /home/oracle/labs/PERF/runload**

$

## Practice 4-3: Using Real-time SQL Monitoring as a SQL Developer

$ **sqlplus sys@PDB1 AS SYSDBA**

Enter password : *password*

SQL> **SELECT \* FROM dba\_sys\_privs WHERE grantee = 'SQLDEV';**

GRANTEE PRIVILEGE

ADM COM INH

SQLDEV CREATE SESSION NO NO NO

SQL> **SELECT owner, table\_name, privilege**

**FROM dba\_tab\_privs WHERE grantee = 'SQLDEV';**

2

OWNER TABLE\_NAME PRIVILEGE

MONI

MONI\_TEST SELECT

### Overview

In this practice you will act in PDB1 as a SQL developer without any super-user privileges nor roles. The SQL developer will use real-time SQL monitoring to analyze the performance of his/her SQL statements.

### Tasks

1. Before starting the practice, execute the /home/oracle/labs/PERF/glogin.sh shell script in *Session1*. It sets formatting for all columns selected in queries.

$ **/home/oracle/labs/PERF/glogin.sh**

**…**

**$**

1. Execute the /home/oracle/labs/PERF/RTMonitor.sql SQL script in *Session1*. The script completes the following operations:
   * Creates the MONI user and MONI\_TEST table and loads the table with thousands of rows.
   * Creates a developer user.
   * Grants the developer user the CREATE SESSION and SELECT ON the

MONI.MONI\_TEST table.

$ **/home/oracle/labs/PERF/RTMonitor.sh**

…

$

1. Check the privileges and roles granted to the SQLDEV user.

SQL> **SELECT \* FROM dba\_role\_privs WHERE grantee = 'SQLDEV';**

no rows selected SQL>

*Q1/ Is SQLDEV granted super-user privileges and roles such as those required in Oracle Database 18c to be able to use real-time SQL monitor?*

##### A1/ No. He is not granted the SELECT CATALOG ROLE role.

1. In *Session2*, connect as the SQLDEV developer in PDB1 and execute a long-running query.

$ **sqlplus sqldev@PDB1**

Enter password: *password*

SQL> **SELECT count(\*) FROM moni.moni\_test t1, moni.moni\_test t2 WHERE t1.c = t2.c AND t1.c = 1;**

…

1. In *Session1*, connect as the SQLDEV developer to PDB1.

SQL> **CONNECT sqldev@PDB1** Enter password: *password* Connected.

SQL>

1. Get an overview of the long running queries.

SQL> **SELECT sql\_id, status, sql\_text FROM v$sql\_monitor;**

SELECT sql\_id, status, sql\_text FROM v$sql\_monitor

\*

ERROR at line 1:

ORA-00942: table or view does not exist

SQL>

*Q1/ Traditionally, real-time SQL monitor is mainly used by DBAs because they are responsible for monitoring and tuning database performance. Real-time SQL monitor tracks and collects SQL and execution plan statistics in fixed views which are only accessible by users who have been granted the SELECT CATALOG ROLE role. A regular user, such as an application developer or a low-privileged user without the SELECT CATALOG ROLE role and SELECT privilege on the real-time SQL monitor fixed views, can write a SQL statement, execute it, see the SQL result set and its SQL plan using the explain plan command, but not its execution plan because it is stored in V$SQL\_PLAN.*

*Is the SQLDEV user granted the SELECT\_CATALOG\_ROLE role? Is the SQLDEV user granted the SELECT privilege on V$SQL\_PLAN? Can the SQLDEV user use real-time SQL monitor to view the execution plan for his SQL statement execution?*

##### A1/ No. The SQLDEV user is not granted the SELECT\_CATALOG\_ROLE role nor the

***SELECT privilege on the V$SQL\_MONITOR view.***

1. Generate the SQL monitor report from the command line, run the

REPORT\_SQL\_MONITOR function in the DBMS\_SQLTUNE package.

SQL> **VARIABLE my\_rept CLOB**

SQL> **BEGIN**

**:my\_rept :=DBMS\_SQLTUNE.REPORT\_SQL\_MONITOR();**

**END;**

**/** 2 3 4

PL/SQL procedure successfully completed. SQL> **SET LINESIZE 78**

SQL> **PRINT :my\_rept**

MY\_REPT

SQL Monitoring Report SQL Text

SELECT count(\*) FROM moni.moni\_test t1, moni.moni\_test t2 WHERE t1.c = t2.c AND t1.c = 1

Global Information

Status : **EXECUTING**

Instance ID : 1

Session : SQLDEV (30:25679)

SQL ID : 9fqxj0xpnt222 SQL Execution ID : 16777217

Execution Started : 05/15/2019 09:32:21 First Refresh Time : 05/15/2019 09:32:29 Last Refresh Time : 05/15/2019 09:33:05 Duration : 43s

Module/Action : SQL\*Plus/-

Service : pdb1

Program : sqlplus@edvmr1p0 (TNS V1-V3)

Global Stats

===================================================================

| Elapsed | Cpu | IO | Other | Buffer | Read | Read |

| Time(s) | Time(s) | Waits(s) | Waits(s) | Gets | Reqs | Bytes |

===================================================================

| 411 | 388 | 0.00 | 23 | 502 | 18 | 3MB |

===================================================================

**SQL Plan Monitoring Details** (Plan Hash Value=183808681)

================================================================================

| Id | Operation | Name | Rows | Cost | Time

| Start | Execs | Rows | Read | Read | Mem | Activity | Activity Detail

|

| | | | (Estim) | | Active(s

) | Active | | (Actual) | Reqs | Bytes | | (%) | (# samples)

|

================================================================================

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| |  4 |  | | 0 | SELECT STATEMENT  +9 | 1 | | 0 | | | | | | | | |  . | | |  | | | | | 35 |
| | | 1 | SORT AGGREGATE |  | | |  |  | | | 1 | | | | | 35 |
| 4 | | +9 | 1 | | 0 | |  | | | | | . | |  | | |  |  |
| | |  |  |  |  |  |  |  |  |  |  |
| | -> 2 | HASH JOIN | |  | | |  | | | | 39G | | | 193K | | 41 |
| 3 | +2 | 1 | | | 4G | |  | | | | 15MB | | | 100.00 | | | Cpu (413) |  |
| | |  |  |  |  |  |  |  |  |  |  |

| 3 | **INDEX FAST FULL SCAN** | MONI\_TEST\_C\_INDX | 198K | 114 |

1 | +9 | 1 | 200K | 18 | 3MB | . | |

|

| -> 4 | **INDEX FAST FULL SCAN** | MONI\_TEST\_C\_INDX | 198K | 114 | 40

4 | +9 | 1 | 48640 | | | . | |

|

================================================================================

SQL> **EXIT**

$

1. In *Session2*, interrupt the long-running query.

**CTRL C**

SQL> SELECT count(\*) FROM moni.moni\_test t1, moni.moni\_test t2

\*

ERROR at line 1:

ORA-01013: user requested cancel of current operation

SQL> **EXIT**

$

## Practice 4-4: Waiting In-memory Objects to be Populated

### Overview

In this practice you will populate in-memory tables into the in-memory column store (IMCS) and wait until they are completely populated before your application can access them. Your application will take better advantage of the complete population of the tables into the IMCS.

### Tasks

1. Before starting the practice, execute the /home/oracle/labs/PERF/glogin.sh shell script in *Session1*. It sets formatting for all columns selected in queries.

$ **/home/oracle/labs/PERF/glogin.sh**

**…**

$

1. Configure the IMCS size to 800M, create and load in-memory tables OE.PART, OE.SUPPLIER, OE.DATE\_DIM, OE.CUSTOMER, and OE.LINEORDER. Execute the

/home/oracle/labs/PERF/IM\_tables.sh shell script to complete these tasks.

$ **/home/oracle/labs/PERF/IM\_tables.sh**

**…**

$

1. Verify that the in-memory column store size is set to 800M.

$ **. oraenv**

ORACLE\_SID = [oracle] ? **ORCL**

The Oracle base has been set to /u01/app/oracle

$ **sqlplus / AS SYSDBA**

SQL> **SHOW PARAMETER inmemory\_size**

NAME

TYPE

VALUE

inmemory\_size

SQL>

big integer 800M

1. Check whether in-memory tables are populated into the IMCS.

SQL> **SELECT segment\_name, bytes, inmemory\_size, bytes\_not\_populated, populate\_status**

**FROM v$im\_segments;**

2 3

no rows selected

SQL>

*Q/ Why are the in-memory tables not populated into the IMCS?*

##### A/ The in-memory tables are not populated into the IMCS because the on- demand population has not been requested.

1. In another session, *Session2*, log on as SYSTEM in PDB1. Execute the function to get information about the status of in-memory tables population at the percentage of 100.

$ **sqlplus system@PDB1**

Enter password: *password*

SQL> **SELECT DBMS\_INMEMORY\_ADMIN.POPULATE\_WAIT(PRIORITY=>'NONE', PERCENTAGE => 100, TIMEOUT => 180)**

**FROM dual;**

2

DBMS\_INMEMORY\_ADMIN.POPULATE\_WAIT(PRIORITY=>'NONE',PERCENTAGE => 100,T

**1**

SQL> **EXIT**

$

*Q1/ Are all in-memory tables populated into the IMCS?*

##### A1/ No, they are not. The population could not complete successfully. The code returned from the function is 1 which means that the in-memory objects are not fully populated into the IMCS because of the lack of space in the in-memory column store.

* 1. Verify this assumption in *Session1*.

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 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\_STAT |  |  |  |  |  |  |  |
| CUSTOMER |  |  | 30932992 |  | 7602176 |  | 0 |
| COMPLETED |  |  |  |  |  |  |  |
| LINEORDER  **OUT OF MEMORY** |  |  | 1592164352 |  | 575406080 |  | 236232704 |

DATE\_DIM

COMPLETED

352256

1310720

0

SQL> **EXIT**

$

*Q1/ Would you allow your application query the tables if you know that the tables queried are not fully populated into the IMCS?*

##### A1/ No, not necessarily because the performance might not be as good.

*Q2****/*** *What can you do to prevent your application from accessing the tables in this case?*

##### A2/ Writing a wrapper package that invokes the function at instance startup, the instance database being opened in restricted mode. The package first queries the in-memory tables, invokes the function and based on the returned code, either increases the INMEMORY\_SIZE parameter value, restarts the database instance, and rechecks the returned code until the function returns the value 0. Then the package can open the database instance in normal mode.

1. Increase the IMCS space, in *Session1*.

SQL> **ALTER SYSTEM SET inmemory\_size=1G SCOPE=SPFILE;**

System altered.

SQL> **ALTER SYSTEM SET sga\_target=1500M SCOPE=SPFILE;**

System altered.

SQL> **SHUTDOWN IMMEDIATE**

Database closed. Database dismounted.

ORACLE instance shut down. SQL> **STARTUP**

ORACLE instance started.

Total System Global Area 1417674048 bytes Fixed Size 9141568 bytes Variable Size 377487360 bytes

Database Buffers 184549376 bytes

Redo Buffers 7634944 bytes

In-Memory Area

Database mounted. Database opened.

838860800 bytes

SQL> **ALTER PLUGGABLE DATABASE pdb1 OPEN;**

Pluggable database altered. SQL>

1. In *Session2*, execute the function that waits until in-memory tables are populated into the IMCS to the specified percentage of 100.

$ **sqlplus system@PDB1**

Enter password: *password*

SQL> **SELECT DBMS\_INMEMORY\_ADMIN.POPULATE\_WAIT('NONE',100,180)**

**FROM dual;**

The query does not give any result until the population of the segments is 100% complete. Observe the population progress in *Session1*.

SQL> **SELECT segment\_name, bytes, inmemory\_size, bytes\_not\_populated, populate\_status**

**FROM v$im\_segments;**

2 3

SQL> **SELECT segment\_name, bytes, inmemory\_size, bytes\_not\_populated, populate\_status**

**FROM v$im\_segments;**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| SEGMENT\_NAME |  |  | BYTES |  | INMEMORY\_SIZE |  | BYTES\_NOT\_POPULATED |
| POPULATE\_STAT |  |  |  |  |  |  |  |
| CUSTOMER COMPLETED |  |  | 30932992 |  | 7602176 |  | 0 |
| LINEORDER  **STARTED** |  |  | 1592172544 |  | 302710784 |  | **879632384** |
| DATE\_DIM COMPLETED |  |  | 352256 |  | 1310720 |  | 0 |

2

3

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| SEGMENT\_NAME |  |  | BYTES |  | INMEMORY\_SIZE |  | BYTES\_NOT\_POPULATED |  |
| POPULATE\_STAT |  |  |  |  |  |  |  |
| CUSTOMER COMPLETED |  |  | 30932992 |  | 7602176 |  | 0 |
| LINEORDER  **STARTED** |  |  | 1592172544 |  | 354484224 |  | **762650624** |
| DATE\_DIM COMPLETED |  |  | 352256 |  | 1310720 |  | 0 |
| 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\_STAT |  |  |  |  |  |  |  | |
| CUSTOMER  **COMPLETED** |  |  | 30932992 |  | 7602176 |  | **0** | |
| LINEORDER  **COMPLETED** |  |  | 1592172544 |  | 674562048 |  | **0** | |
| SUPPLIER  **COMPLETED** |  |  | 2015232 |  | 1310720 |  | **0** | |
| DATE\_DIM  **COMPLETED** |  |  | 352256 |  | 1310720 |  | **0** | |
| PART  **COMPLETED** |  |  | 167591936 |  | 37355520 |  | **0** | |
| SQL> |  |  |  |  |  |  |  | |

When the population completes, the query in *Session2* completes.

2

DBMS\_INMEMORY\_ADMIN.POPULATE\_WAIT('NONE',100,180)

0

SQL>

*Q1/ Are all in-memory tables populated into the IMCS?*

##### A1/ Yes, they are. The population could complete successfully. The code returned from the function is 0 which means that the in-memory objects are fully populated into the IMCS.

*Q2/ Would you allow your application query the tables if you know that the tables queried are fully populated into the IMCS?*

##### A2/ Yes. This is why the wrapper package invoking the function at instance startup would be beneficial.

1. In *Session2*, execute the function that waits until in-memory tables are populated into the IMCS to the specified percentage of 100 before the applications can access the tables.
   1. First execute a SQL script in *Session2* that modifies the in-memory attribute of the OE

tables.

SQL> **@/home/oracle/labs/PERF/alter\_OE.sql**

…

SQL> **EXIT**

$

* 1. In *Session1*, restart PDB1 so that the in-memory tables are no more populated into the IMCS.

SQL> **ALTER PLUGGABLE DATABASE pdb1 close;**

Pluggable database altered.

SQL> **ALTER PLUGGABLE DATABASE pdb1 OPEN;**

Pluggable database altered. SQL>

* 1. In *Session2*, connect and execute the function that waits until in-memory tables are populated into the IMCS to the specified percentage of 100.

$ **sqlplus system@PDB1** Enter password: *password* Connected.

SQL> **SELECT DBMS\_INMEMORY\_ADMIN.POPULATE\_WAIT(PRIORITY=>'NONE', PERCENTAGE=>100, TIMEOUT => 180)**

**FROM dual;**

2

DBMS\_INMEMORY\_ADMIN.POPULATE\_WAIT(PRIORITY=>'NONE',PERCENTAGE=>1 00,T

**2**

SQL>

*Q1/ Are all in-memory tables populated into the IMCS? Check in Session1.*

SQL> **SELECT segment\_name, bytes, inmemory\_size, bytes\_not\_populated, populate\_status**

**FROM v$im\_segments;**

2 3

no rows selected

SQL>

##### A1/ No, they are not. The code returned from the function is 2 which means that there are no in-memory objects populated yet. This is the normal behavior as the in-memory tables have been set to no in-memory tables with the SQL script.

*Q2/ If you don’t know the different return code meaning, how can you retrieve a more meaningful message?*

SQL> **SET SERVEROUTPUT ON**

SQL> **DECLARE**

**v\_force boolean := TRUE; v\_return integer;**

**BEGIN**

**v\_return := dbms\_inmemory\_admin.populate\_wait( priority => 'NONE',**

**percentage => 100,**

**timeout => 180, force => v\_force);**

**END;**

**/**

2 3 4 5 6 7

8

DBMS\_INMEMORY.POPULATE\_WAIT

timeout: 180

now: 15-MAY-19 09.48.13.196624000 AM +00:00 timeout time: 15-MAY-19 09.51.13.134256 AM

percentage: 100 priority: NONE instances: 1 DBA\_SEGMENTS:

**NO INMEMORY OBJECTS TO POPULATE**

PL/SQL procedure successfully completed. SQL>

##### A2/ The message NO INMEMORY OBJECTS TO POPULATE is more meaningful.

* 1. In *Session2*, update the priority NONE to HIGH for the two tables called in the

/home/oracle/labs/PERF/IM\_query.sql SQL script.

SQL> **ALTER TABLE oe.lineorder INMEMORY PRIORITY high;**

Table altered.

SQL> **ALTER TABLE oe.date\_dim INMEMORY PRIORITY high;**

Table altered.

SQL> **EXIT**

$

* 1. In *Session1*, restart PDB1.

SQL> **ALTER PLUGGABLE DATABASE pdb1 close;**

Pluggable database altered.

SQL> **ALTER PLUGGABLE DATABASE pdb1 OPEN;**

Pluggable database altered.

SQL> **EXIT**

$

* 1. Use the function in *Session2* to verify that the in-memory tables with HIGH priority are fully populated.

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| $ **sqlplus system@PDB1** Enter password: *password* Connected.  SQL> **SELECT DBMS\_INMEMORY\_ADMIN.POPULATE\_WAIT(PRIORITY=>'HIGH',**  **PERCENTAGE=>100, TIMEOUT => 60)**  **FROM dual;**  2 3  DBMS\_INMEMORY\_ADMIN.POPULATE\_WAIT(PRIORITY=>'HIGH',PERCENTAGE=>1 00,T  **0**  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\_STAT |  |  |  |  |  |  |  |
| **LINEORDER** |  |  | 1592172544 |  | 675610624 |  | 0 |
| COMPLETED |  |  |  |  |  |  |  |
| DATE\_DIM COMPLETED |  |  | 352256 |  | 1310720 |  | 0 |
| SQL> **EXIT** |  |  |  |  |  |  |  |
| $ |  |  |  |  |  |  |  |

1. In *Session1*, complete a last test by executing the /home/oracle/labs/PERF/test.sh

shell script.

$ **/home/oracle/labs/PERF/test.sh**

…

$

1. In *Session1*, verify the return status of the function once the cleanup completed.

$ **sqlplus / AS SYSDBA**

Connected.

SQL> **SELECT DBMS\_INMEMORY\_ADMIN.POPULATE\_WAIT(PRIORITY=>'HIGH', PERCENTAGE=>100, TIMEOUT => 60)**

**FROM dual;**

2 3

DBMS\_INMEMORY\_ADMIN.POPULATE\_WAIT(PRIORITY=>'HIGH',PERCENTAGE=>1 00,T

**3**

SQL>

*Q1/ How do you retrieve a more meaningful message?*

##### A1/ Using the function in a PL/SQL block returns a more meaningful message. In the current situation, the function informs you that the INMEMORY\_SIZE is set to 0 and cannot therefore populate in-memory tables.

SQL> **SET SERVEROUTPUT ON**

SQL> **DECLARE**

**v\_force boolean := TRUE; v\_return integer;**

**BEGIN**

**v\_return := dbms\_inmemory\_admin.populate\_wait( priority => 'HIGH',**

**percentage => 100,**

**timeout => 60, force => v\_force);**

**END;**

**/**

2 3 4 5 6 7 8 9 10 11

**POPULATE ERROR, INMEMORY\_SIZE=0**

PL/SQL procedure successfully completed. SQL> **EXIT**

$

1. Execute the /home/oracle/labs/PERF/cleanup\_IM\_tables.sh shell script to drop the in-memory tables in PDB1 and disable the IMCS.

$ **/home/oracle/labs/PERF/cleanup\_IM\_tables.sh**

**…**

$

## Practice 4-5: Configuring and Using Database Replay at PDB Level

### Overview

In this practice you will capture a workload from PDB1 and replay the workload at the PDB level into PDB19. The Database Replay operations can be performed at the PDB level.

### Tasks

1. Before starting the practice, execute the /home/oracle/labs/PERF/DBReplay.sh shell script in *Session1*. The script re-creates PDB1 and PDB19, removes any existing database replay files.

$ **/home/oracle/labs/PERF/DBReplay.sh**

…

$

In *Session1,* log in to PDB1 and capture the workload data by using Database Replay.

$ **. oraenv**

ORACLE\_SID = [oracle] ? **ORCL**

The Oracle base has been set to /u01/app/oracle

$ **sqlplus system@PDB1**

Enter password: *password*

SQL>

* 1. The Database Replay capture creates files in a directory. Create the logical directory for the capture files.

SQL> **HOST mkdir -p /home/oracle/PDB1/replay**

SQL> **CREATE OR REPLACE DIRECTORY oltp**

**AS '/home/oracle/PDB1/replay';**

2

Directory created.

SQL>

* 1. Start capturing data with the Database Replay procedure.

SQL> **EXEC DBMS\_WORKLOAD\_CAPTURE.START\_CAPTURE ( -**

**name => 'OLTP\_peak', - dir => 'OLTP')**

> >

PL/SQL> procedure successfully completed.

SQL>

* 1. During the capture, in *Session2*, execute the workload on PDB1 by executing the

/home/oracle/labs/PERF/workload.sh shell script.

$ **/home/oracle/labs/PERF/workload.sh**

…

$

* 1. When you consider that the workload is sufficient for replay testing, stop the capture in

*Session1*.

SQL> **EXEC DBMS\_WORKLOAD\_CAPTURE.FINISH\_CAPTURE ()**

PL/SQL> procedure successfully completed. SQL> **EXIT**

$

Process the capture files and replay the workload in *Session1* in PDB19.

1. As in the normal whole process of Database Replay, after capturing the workload into files, you process the capture files. You will replay the capture files in PDB19.

$ **sqlplus system@PDB19**

Enter password: *password*

SQL>

* 1. Create the logical directory in PDB19 for processing and initializing the capture files stored in /home/oracle/PDB1/replay to be replayed.

SQL> **CREATE OR REPLACE DIRECTORY oltp**

**AS '/home/oracle/PDB1/replay';**

2

Directory created.

SQL>

* 1. Process the capture files

SQL> **EXEC DBMS\_WORKLOAD\_REPLAY.PROCESS\_CAPTURE ( -**

**capture\_dir => 'OLTP')**

>

PL/SQL> procedure successfully completed.

SQL>

* 1. Initialize the replay.

SQL> **EXEC DBMS\_WORKLOAD\_REPLAY.INITIALIZE\_REPLAY( -**

**replay\_name => 'R', replay\_dir => 'OLTP')**

>

PL/SQL> procedure successfully completed.

* 1. Prepare the replay.

SQL> **EXEC DBMS\_WORKLOAD\_REPLAY.PREPARE\_REPLAY ()**

PL/SQL> procedure successfully completed. SQL>

* 1. You are ready to start workload clients to replay the captured workload in PDB19 with

wrc clients. In *Session2*, if the workload is still not finished, interrupt the

/home/oracle/labs/PERF/workload.sh shell script, quit the SQL\*Plus session and start the wrc process into PDB19.

$ **wrc REPLAYDIR=/home/oracle/PDB1/replay USERID=system SERVER=PDB19**

…

Password: ***password***

Wait for the replay to start (11:40:35)

The password required is the SYSTEM user password.

* 1. The wrc client is waiting for Database Replay to start in the PDB. In *Session1*, execute the START\_REPLAY procedure.

SQL> **exec DBMS\_WORKLOAD\_REPLAY.START\_REPLAY ()**

PL/SQL> procedure successfully completed. SQL>

* 1. As soon as the Database Replay procedure is started in PDB19, the client starts replaying.

Replay client 1 started (11:42:05)

1. Meanwhile, in *Session1*, verify that the client is executing on PDB19.

SQL> **CONNECT system@PDB19** Enter password: *password* Connected.

SQL> **SELECT username, con\_id, module**

**FROM v$session**

**WHERE username <> 'SYS' AND con\_id <> 0;**

2 3

USERNAME CON\_ID MODULE

|  |  |  |
| --- | --- | --- |
| SYSTEM | 5 | **WRC$** |
| SYSTEM | 5 | SQL\*Plus |
| SYSTEM | 5 | WRC$ |
| SQL> **EXIT** |  |  |
| $ |  |  |

1. When the wrc client finally completes, execute the

/home/oracle/labs/PERF/cleanup\_DBReplay.sh shell script to drop the Database Replay capture files.

Replay client 1 finished (11:50:11)

$

$ **/home/oracle/labs/PERF/cleanup\_DBReplay.sh**

**…**

$

# Practices for Lesson 5: Using Big Data and Data Warehousing Enhancements

## Practices for Lesson 5: Overview

### Overview

In these practices, you will manipulate HyPTs.

## Practice 5-1: Managing Hybrid Partitioned Tables

### Overview

In this practice you create and maintain hybrid partitioned tables (HyPT).

### Tasks

1. Before starting the practice, execute the $HOME/labs/DW/glogin.sh shell script. It sets formatting for all columns selected in queries.

$ **$HOME/labs/DW/glogin.sh**

**…**

$

1. Create an HyPT in PDB1 with the following charateristics:
   * The table is partitioned by range on the TIME\_ID column.
   * The default tablespace for internal partitions is TS1.
   * The default tablespace for external partitions is CENT20.
   * The fields in the records of the external files are separated by **','**.
   * The table is partitioned into 5 partitions:
   * 3 external partitions: CENT18 is empty for the moment, CENT19 has the cent19.dat file stored in another directory than the default, CENT19, CENT20 has the cent20.dat file stored in the default directory.
     + 2 internal partitions: Y2000 is stored in tablespace TS2 and PMAX in the default tablespace.
2. Create the tablespaces for the internal partitions.

$ **. oraenv**

ORACLE\_SID = [oracle] ? **ORCL**

The Oracle base remains unchanged with value /u01/app/oracle

$ **sqlplus system@PDB1**

Enter password : *password*

SQL> **CREATE TABLESPACE ts1**

**DATAFILE '/u02/app/oracle/oradata/ORCL/pdb1/ts1.dbf' SIZE 100M;**

2 3

Tablespace created.

SQL> **CREATE TABLESPACE ts2**

**DATAFILE '/u02/app/oracle/oradata/ORCL/pdb1/ts2.dbf' SIZE 100M;**

2 3

Tablespace created.

SQL>

1. Create the directories for the external partitions.

SQL> **HOST mkdir -p /home/oracle/labs/DW/CENT18** SQL> **HOST mkdir -p /home/oracle/labs/DW/CENT19** SQL> **HOST mkdir -p /home/oracle/labs/DW/CENT20** SQL>

SQL> **CREATE DIRECTORY cent18 AS '/home/oracle/labs/DW/CENT18';**

Directory created.

SQL> **CREATE DIRECTORY cent19 AS '/home/oracle/labs/DW/CENT19';**

Directory created.

SQL> **CREATE DIRECTORY cent20 AS '/home/oracle/labs/DW/CENT20';**

Directory created. SQL>

1. Create the HYPT user.

SQL> **CREATE USER hypt IDENTIFIED BY *password***

**DEFAULT TABLESPACE ts1;**

2

User created.

SQL> **GRANT read, write ON DIRECTORY cent18 TO hypt;**

Grant succeeded.

SQL> **GRANT read, write ON DIRECTORY cent19 TO hypt;**

Grant succeeded.

SQL> **GRANT read, write ON DIRECTORY cent20 TO hypt;**

Grant succeeded.

SQL> **GRANT create session, unlimited tablespace, create table TO hypt;**

2

Grant succeeded.

SQL>

1. Create the HyPT table.

#### SQL> CREATE TABLE hypt.hypt\_tab (history\_event NUMBER , time\_id DATE)

**TABLESPACE ts1**

#### EXTERNAL PARTITION ATTRIBUTES (TYPE ORACLE\_LOADER

**DEFAULT DIRECTORY cent20 ACCESS PARAMETERS**

#### (FIELDS TERMINATED BY ','

**(history\_event , time\_id DATE 'dd-MON-yyyy')**

#### )

**REJECT LIMIT UNLIMITED**

#### )

**PARTITION BY RANGE (time\_id) (PARTITION cent18 VALUES LESS THAN**

#### (TO\_DATE('01-Jan-1800','dd-MON-yyyy')) EXTERNAL,

**PARTITION cent19 VALUES LESS THAN**

#### (TO\_DATE('01-Jan-1900','dd-MON-yyyy')) EXTERNAL

**DEFAULT DIRECTORY cent19**

#### LOCATION ('cent19.dat'),

**PARTITION cent20 VALUES LESS THAN**

#### (TO\_DATE('01-Jan-2000','dd-MON-yyyy')) EXTERNAL

**LOCATION('cent20.dat'),**

#### PARTITION y2000 VALUES LESS THAN

**(TO\_DATE('01-Jan-2001','dd-MON-yyyy')) TABLESPACE ts2, PARTITION pmax VALUES LESS THAN (MAXVALUE));**

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

Table created.

SQL>

1. Insert rows into the table.
2. Execute the /home/oracle/labs/DW/insert.sql SQL script to insert rows into the internal partitions of the HYPT\_TAB table.

SQL> **@/home/oracle/labs/DW/insert.sql**

… SQL>

1. Insert a row for the date of 12 August 1997.

SQL> **INSERT INTO hypt.hypt\_tab**

**VALUES (41, to\_date('12.08.1997', 'dd.mm.yyyy'));**

2 INSERT INTO hypt.hypt\_tab

\*

ERROR at line 1:

ORA-14466: Data in a read-only partition or subpartition cannot be modified.

SQL>

*Q1/ Why does it fail?*

##### A1/ The date can only be inserted into the external partition and therefore via the external file.

1. Insert the row for the date of “12 August 1997” into the appropriate external file.

SQL> **host echo "41,12-Aug-1997" >>**

**/home/oracle/labs/DW/CENT20/cent20.dat**

SQL> **SELECT history\_event, TO\_CHAR(time\_id, 'dd-MON-yyyy') FROM hypt.hypt\_tab PARTITION (cent20);**

2

HISTORY\_EVENT TO\_CHAR(TIME\_ID,'DD-

1 01-JAN-1976

2 01-JAN-1915

3 01-JAN-1928

4 01-JAN-1937

5 01-JAN-1949

6 01-FEB-1959

7 01-FEB-1996

8 01-FEB-1997

9 01-FEB-1998

10 01-FEB-1998

41 12-AUG-1997

11 rows selected.

SQL>

*Q1/ What happens if you append a row into the wrong external file?*

SQL> **host echo "42,12-Aug-1997" >>**

**/home/oracle/labs/DW/CENT19/cent19.dat**

SQL> **SELECT history\_event, TO\_CHAR(time\_id, 'dd-MON-yyyy') FROM hypt.hypt\_tab PARTITION (cent19);**

2

HISTORY\_EVENT TO\_CHAR(TIME\_ID,'DD-

11 01-JAN-1876

12 01-JAN-1815

13 01-JAN-1828

14 01-JAN-1837

15 01-JAN-1849

16 01-FEB-1859

17 01-FEB-1896

18 01-FEB-1897

19 01-FEB-1898

20 01-FEB-1898

42 12-AUG-1997

11 rows selected.

SQL>

##### A1/ There is no control on the TIME\_ID of the records inserted as rows into the external partitions, as it is the case for rows inserted into internal partitions.

1. Remove the line inserted into /home/oracle/labs/DW/CENT19/cent19.dat.

SQL> **host**

$ **vi /home/oracle/labs/DW/CENT19/cent19.dat**

**…**

**42,12-Aug-1997**

$ **exit**

SQL>

1. Query the rows in the external and internal partitions.

SQL> **SELECT history\_event, TO\_CHAR(time\_id, 'dd-MON-yyyy') FROM hypt.hypt\_tab PARTITION (cent18);**

2

no rows selected

#### SQL> SELECT history\_event, TO\_CHAR(time\_id, 'dd-MON-yyyy') FROM hypt.hypt\_tab PARTITION (cent19);

2

HISTORY\_EVENT TO\_CHAR(TIME\_ID,'DD-

11 01-JAN-1876

12 01-JAN-1815

13 01-JAN-1828

14 01-JAN-1837

15 01-JAN-1849

16 01-FEB-1859

17 01-FEB-1896

18 01-FEB-1897

19 01-FEB-1898

20 01-FEB-1898

10 rows selected.

#### SQL> SELECT history\_event, TO\_CHAR(time\_id, 'dd-MON-yyyy') FROM hypt.hypt\_tab PARTITION (y2000);

2

HISTORY\_EVENT TO\_CHAR(TIME\_ID,'DD-

21 31-DEC-2000

22 31-OCT-2000

23 01-FEB-2000

24 27-MAR-2000

25 31-MAR-2000

26 15-APR-2000

27 02-SEP-2000

28 12-AUG-2000

8 rows selected.

#### SQL> SELECT history\_event, TO\_CHAR(time\_id, 'dd-MON-yyyy') FROM hypt.hypt\_tab PARTITION (pmax);

2

HISTORY\_EVENT TO\_CHAR(TIME\_ID,'DD-

29 12-AUG-2018

30 15-SEP-2017

SQL>

*Q1/ How would you distinguish the partitioned tables from the hybrid partitioned tables?*

SQL> **SELECT partition\_name, high\_value FROM dba\_tab\_partitions WHERE table\_name = 'HYPT\_TAB';**

2

PARTITION\_NAME HIGH\_VALUE

CENT18

TO\_DATE(' 1800-01-01

00:00:00', 'SYYYY-M

M-DD HH24:MI:SS', 'N LS\_CALENDAR=GREGORIA

##### A1/ Verify the existence of the hybrid partitioned table in the following view and then the list of the partitions from DBA\_TAB\_PARTITIONS.

SQL> **SELECT \* FROM dba\_external\_tables WHERE owner = 'HYPT';**

OWNER TABLE\_NAME TYP TYPE\_NAME

DEF

--- DEFAULT\_DIRECTORY\_NAME

REJECT\_LIMIT

ACCESS\_

ACCESS\_PARAMETERS

PROPERTY INMEMORY INMEMORY\_COMPRESS

HYPT

**HYPT\_TAB** SYS

ORACLE\_LOADER SYS

CENT20 UNLIMITED

FIELDS TERMINATED BY ','

CLOB

(history\_event , time\_id DATE 'dd-MON-yyyy') ALL DISABLED

SQL>

CENT19

TO\_DATE(' 1900-01-01

00:00:00', 'SYYYY-M

M-DD HH24:MI:SS', 'N

LS\_CALENDAR=GREGORIA

CENT20

TO\_DATE(' 2000-01-01

00:00:00', 'SYYYY-M

M-DD HH24:MI:SS', 'N LS\_CALENDAR=GREGORIA

PMAX

Y2000

MAXVALUE

TO\_DATE(' 2001-01-01

00:00:00', 'SYYYY-M

M-DD HH24:MI:SS', 'N LS\_CALENDAR=GREGORIA

SQL>

1. Create another partitioned table with only internal partitions. Execute the

/home/oracle/labs/DW/internal\_part.sql SQL script.

SQL> **@/home/oracle/labs/DW/internal\_part.sql**

SQL> CREATE TABLE hypt.part\_tab

1. (history\_event NUMBER , time\_id DATE)
2. TABLESPACE ts1
3. PARTITION BY RANGE (time\_id)
4. (PARTITION cent18 VALUES LESS THAN (TO\_DATE('01-Jan- 1800','dd-MON-yyyy')) ,
5. PARTITION cent19 VALUES LESS THAN (TO\_DATE('01-Jan- 1900','dd-MON-yyyy')) ,
6. PARTITION cent20 VALUES LESS THAN (TO\_DATE('01-Jan- 2000','dd-MON-yyyy')) ,
7. PARTITION y2000 VALUES LESS THAN (TO\_DATE('01-Jan- 2001','dd-MON-yyyy')) TABLESPACE ts2,
8. PARTITION pmax VALUES LESS THAN (MAXVALUE)); Table created.

SQL>

1. Insert rows into the table. Execute the /home/oracle/labs/DW/insert2.sql SQL script to insert rows into the internal partitions of the PART\_TAB table.

SQL> **@/home/oracle/labs/DW/insert2.sql**

… SQL>

1. Display the rows in the table.

SQL> **SELECT history\_event, TO\_CHAR(time\_id, 'dd-MON-yyyy') FROM hypt.part\_tab PARTITION (y2000);**

2

HISTORY\_EVENT TO\_CHAR(TIME\_ID,'DD-

21 31-DEC-2000

22 31-OCT-2000

23 01-FEB-2000

24 27-MAR-2000

25 31-MAR-2000

26 15-APR-2000

27 02-SEP-2000

28 12-AUG-2000

8 rows selected.

SQL> **SELECT history\_event, TO\_CHAR(time\_id, 'dd-MON-yyyy') FROM hypt.part\_tab PARTITION (pmax);**

2

HISTORY\_EVENT TO\_CHAR(TIME\_ID,'DD-

29 12-AUG-2018

30 15-SEP-2017

SQL>

1. Add an external partition to the internal partitioned table HYPT.PART\_TAB for the 17th century. The external file cent17.dat storing historic events of the 17th century is stored in the directory /home/oracle/labs/DW/CENT17.
2. Create the logical directory.

SQL> **CREATE DIRECTORY cent17 AS '/home/oracle/labs/DW/CENT17';**

Directory created. SQL>

1. Grant the read and write privileges on the directory to HYPT.

SQL> **GRANT read, write ON DIRECTORY cent17 TO hypt;**

Grant succeeded.

SQL>

1. Add the external partition to the HYPT.PART\_TAB.
   1. First define the external parameters for all external partitions that might be added to the HYPT.PART\_TAB table.

SQL> **ALTER TABLE hypt.part\_tab**

**ADD EXTERNAL PARTITION ATTRIBUTES (TYPE ORACLE\_LOADER**

**DEFAULT DIRECTORY cent17 ACCESS PARAMETERS**

**(FIELDS TERMINATED BY ','**

**(history\_event , time\_id DATE 'dd-MON-yyyy')) REJECT LIMIT UNLIMITED**

**);**

2 3 4 5 6 7 8 9

Table altered.

SQL>

* 1. Verify the existence of the hybrid partitioned table in the following view.

SQL> **SELECT \* FROM dba\_external\_tables WHERE owner = 'HYPT';**

OWNER TABLE\_NAME TYP TYPE\_NAME

DEF

--- DEFAULT\_DIRECTORY\_NAME

REJECT\_LIMIT

ACCESS\_

ACCESS\_PARAMETERS

PROPERTY INMEMORY INMEMORY\_COMPRESS

HYPT

**HYPT\_TAB** SYS

ORACLE\_LOADER SYS

CENT20

UNLIMITED

FIELDS TERMINATED BY ','

CLOB

(history\_event , time\_id DATE 'dd-MON-yyyy')

ALL DISABLED

HYPT

**PART\_TAB** SYS

ORACLE\_LOADER SYS

**CENT17**

UNLIMITED

FIELDS TERMINATED BY ','

CLOB

(history\_event , time\_id DATE 'dd-MON-yyyy') ALL DISABLED

SQL>

* 1. Then add the external partition for all historic records of the 17th century.

SQL> **ALTER TABLE hypt.part\_tab**

**ADD PARTITION cent17 VALUES LESS THAN (TO\_DATE('01-Jan-1700','dd-MON-yyyy'))**

**EXTERNAL LOCATION('cent17.dat');**

2 3 4 ADD PARTITION cent17 VALUES LESS THAN

\*

ERROR at line 2:

ORA-14074: partition bound must collate higher than that of the last partition

SQL>

*Q1/ What can you do to add the partition below the first one?*

##### A2/ Split the first partition to a partition limit that will be the high limit of the partition added.

SQL> **ALTER TABLE hypt.part\_tab**

**SPLIT PARTITION cent18**

**AT (TO\_DATE('01-Jan-1700','dd-MON-yyyy'))**

**INTO (PARTITION cent17 EXTERNAL LOCATION('cent17.dat'), PARTITION cent18);**

2 3 4 5

Table altered.

SQL>

* 1. List the partitions of the table.

SQL> **SELECT partition\_name, high\_value FROM dba\_tab\_partitions WHERE table\_name='PART\_TAB';**

2

PARTITION\_NAME HIGH\_VALUE

CENT17

TO\_DATE(' 1700-01-01

00:00:00', 'SYYYY-M

M-DD HH24:MI:SS', 'N LS\_CALENDAR=GREGORIA

CENT18

TO\_DATE(' 1800-01-01

00:00:00', 'SYYYY-M

M-DD HH24:MI:SS', 'N LS\_CALENDAR=GREGORIA

CENT19

TO\_DATE(' 1900-01-01

00:00:00', 'SYYYY-M

M-DD HH24:MI:SS', 'N LS\_CALENDAR=GREGORIA

CENT20

TO\_DATE(' 2000-01-01

00:00:00', 'SYYYY-M

M-DD HH24:MI:SS', 'N LS\_CALENDAR=GREGORIA

PMAX

Y2000

MAXVALUE

TO\_DATE(' 2001-01-01

00:00:00', 'SYYYY-M

M-DD HH24:MI:SS', 'N

LS\_CALENDAR=GREGORIA

6 rows selected.

SQL>

SQL> **SELECT history\_event, TO\_CHAR(time\_id, 'dd-MON-yyyy') FROM hypt.part\_tab PARTITION (cent17);**

2

HISTORY\_EVENT TO\_CHAR(TIME\_ID,'DD-

101 01-JAN-1676

|  |  |
| --- | --- |
| 102 | 01-JAN-1615 |
| 103 | 01-JAN-1628 |
| 104 | 01-JAN-1637 |
| 105 | 01-JAN-1649 |
| 106 | 01-FEB-1659 |
| 107 | 01-FEB-1696 |
| 108 | 01-FEB-1697 |
| 109 | 01-FEB-1698 |
| 200 | 01-FEB-1698 |

1. The partition storing 17th century historic events is no longer required. Remove the external parameters for the hybrid partitioned that was added to the HYPT.PART\_TAB table.

SQL> **ALTER TABLE hypt.part\_tab**

**DROP EXTERNAL PARTITION ATTRIBUTES();**

ALTER TABLE hypt.part\_tab DROP EXTERNAL PARTITION ATTRIBUTES()

\*

ERROR at line 1:

ORA-14354: operation not supported for a hybrid-partitioned table

SQL>

10 rows selected.

SQL>

*Q1/ What does the error message mean?*

##### A1/ As long as there is one external partition left, the attributes for the external partitions cannot be removed from the hybrid partitioned table.

1. First drop the external partition from the HYPT.PART\_TAB table.

SQL> **ALTER TABLE hypt.part\_tab DROP PARTITION cent17;**

Table altered.

SQL>

1. Then remove the external parameters for the hybrid partitioned table.

SQL> **ALTER TABLE hypt.part\_tab**

**DROP EXTERNAL PARTITION ATTRIBUTES();**

2

Table altered.

SQL>

1. Drop the HyPTs.

SQL> **DROP TABLE hypt.hypt\_tab PURGE;**

Table dropped.

SQL> **DROP TABLE hypt.part\_tab PURGE;**

Table dropped.

SQL> **EXIT**

$

## Practice 5-2: Creating and Populating In-memory Hybrid Partitioned Tables

### Overview

In this practice you create and load partitions of an in-memory hybrid partitioned table.

### Tasks

1. If you completed the steps 1 and 2.a to 2.c of 5-1, then go on with the following steps, else complete the steps 1 and 2.a to 2.c of 5-1 first.
2. Set the in-memory column store size to 800M.

$ **sqlplus / AS SYSDBA**

SQL> **ALTER SYSTEM SET inmemory\_SIZE = 800M SCOPE=SPFILE;**

System altered.

SQL> **SHUTDOWN IMMEDIATE**

Database closed. Database dismounted.

ORACLE instance shut down. SQL> **STARTUP**

ORACLE instance started.

Total System Global Area 1417670704 bytes Fixed Size 9142320 bytes Variable Size 377487360 bytes

Database Buffers 184549376 bytes Redo Buffers 7630848 bytes In-Memory Area 838860800 bytes Database mounted.

Database opened.

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

Pluggable database altered. SQL>

1. Create the in-memory hybrid partitioned table HYPT.HYPT\_INMEM\_TAB in PDB1 with the

COMPRESSION FOR QUERY HIGH attribute.

SQL> **CONNECT system@PDB1**

Enter password: *password*

Connected.

#### SQL> CREATE TABLE hypt.hypt\_inmem\_tab (history\_event NUMBER , time\_id DATE)

**TABLESPACE ts1**

#### EXTERNAL PARTITION ATTRIBUTES (TYPE ORACLE\_LOADER

**DEFAULT DIRECTORY cent20 ACCESS PARAMETERS**

#### (FIELDS TERMINATED BY ','

**(history\_event , time\_id DATE 'dd-MON-yyyy')**

#### )

**REJECT LIMIT UNLIMITED**

#### )

**PARTITION BY RANGE (time\_id) (PARTITION cent18 VALUES LESS THAN**

#### (TO\_DATE('01-Jan-1800','dd-MON-yyyy')) EXTERNAL,

**PARTITION cent19 VALUES LESS THAN**

#### (TO\_DATE('01-Jan-1900','dd-MON-yyyy')) EXTERNAL

**DEFAULT DIRECTORY cent19**

#### LOCATION ('cent19.dat'),

**PARTITION cent20 VALUES LESS THAN**

#### (TO\_DATE('01-Jan-2000','dd-MON-yyyy')) EXTERNAL

**LOCATION('cent20.dat'),**

#### PARTITION y2000 VALUES LESS THAN

**(TO\_DATE('01-Jan-2001','dd-MON-yyyy')) TABLESPACE ts2, PARTITION pmax VALUES LESS THAN (MAXVALUE))**

#### INMEMORY MEMCOMPRESS FOR QUERY HIGH;

|  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 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 |  |  |  |  |

Table created.

SQL>

*Q1/ Which partitions are defined as in-memory segments?*

SQL> **SELECT partition\_name, inmemory, inmemory\_compression FROM dba\_tab\_partitions**

**WHERE table\_name = 'HYPT\_INMEM\_TAB';**

2 3

PARTITION\_NAME INMEMORY INMEMORY\_COMPRESS

CENT18

CENT19

DISABLED

DISABLED

CENT20

PMAX Y2000

DISABLED

ENABLED FOR QUERY HIGH ENABLED FOR QUERY HIGH

SQL>

##### A1/ Only the internal partitions are defined as in-memory segments.

1. Insert rows into the different partitions of the table by executing the

/home/oracle/labs/DW/insert\_select.sql SQL script and query the table to populate the data into the in-memory column store.

SQL> **@/home/oracle/labs/DW/insert\_select.sql**

… SQL>

*Q1/ Which partitions are populated into the in-memory column store?*

##### A1/ Only the partitions defined as in-memory segments are populated into the in- memory column store.

SQL> **SELECT segment\_name, partition\_name, tablespace\_name, Populate\_status**

**FROM v$im\_segments;**

2 3

SEGMENT\_NAME PARTITION\_NAME TABLESPACE\_NAME POPULATE\_STAT

HYPT\_INMEM\_TAB PMAX

HYPT\_INMEM\_TAB Y2000

TS1

TS2

COMPLETED

COMPLETED

SQL>

*Q2/ Does the execution plan show the different types of access to partitions?*

SQL> **EXPLAIN PLAN FOR SELECT \* FROM hypt.hypt\_inmem\_tab;**

Explained.

SQL> **SELECT \* FROM TABLE(DBMS\_XPLAN.DISPLAY);**

PLAN\_TABLE\_OUTPUT

| Id | Operation | Name

| Rows | Bytes | Cost (%CPU)| Time | Pstart| Pstop |

| 0 | SELECT STATEMENT | 368K| 7917K| 778 (11)| 00:00:01 | | |

| 1 | **PARTITION RANGE ALL** | 368K| 7917K| 778 (11)| 00:00:01 | 1 | 5 |

| 2 | **TABLE ACCESS HYBRID PART INMEMORY FULL**| HYPT\_INMEM\_TAB

368K| 7917K| 778 (11)| 00:00:01 | 1 | 5 |

| 3 | TABLE ACCESS INMEMORY FULL | HYPT\_INMEM\_TAB

| | | | 1 | 5 |

10 rows selected.

SQL> **EXPLAIN PLAN FOR**

#### SELECT \* FROM hypt.hypt\_inmem\_tab PARTITION (PMAX);

2

Explained.

#### SQL> SELECT \* FROM TABLE(DBMS\_XPLAN.DISPLAY);

PLAN\_TABLE\_OUTPUT

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| | Id Bytes | |  | | Operation  Cost (%CPU)| Time | | | | Name Pstart| Pstop | | | | | | Rows | | |
| | 0 |  1765 K| | | SELECT STATEMENT  25 (56)| 00:00:01 | | | | | | | | | | | 82171 | | |
| | 1 | | | **PARTITION RANGE SINGLE** | | | |  | | | | 82171 | | |
| 1765K| | | 25 (56)| 00:00:01 | | | 5 | | 5 | |  | |  |  |

| 2 **| TABLE ACCESS INMEMORY FULL**| HYPT\_INMEM\_TAB | 82171 | 1765K| 25 (56)| 00:00:01 | 5 | 5 |

9 rows selected.

SQL> **EXPLAIN PLAN FOR**

#### SELECT \* FROM hypt.hypt\_inmem\_tab PARTITION (CENT19);

2

Explained.

#### SQL> SELECT \* FROM TABLE(DBMS\_XPLAN.DISPLAY);

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| | Id Bytes | |  | | Operation  Cost (%CPU)| Time | | | | Name Pstart| Pstop | | | | | | Rows | | |
| | 0  175K| | | | SELECT STATEMENT  31 (7)| 00:00:01 | | | |  | | | | | | | 8169 | | |
| | 1 | | | **PARTITION RANGE SINGLE** | | | |  | | | | 8169 | | |
| 175K| |  | 31 (7)| 00:00:01 | | | 2 | | 2 | | |  |  |  |
| | 2 | | | **EXTERNAL TABLE ACCESS** | | **FULL**| | HYPT\_INMEM\_TAB | | | | 8169 | | |
| 175K| |  | 31 (7)| 00:00:01 | | | 2 | | 2 | | |  |  |  |

***A2/ According to the partition accessed, the operation shows either EXTERNAL TABLE ACCESS FULL (not INMEMORY) or TABLE ACCESS INMEMORY FULL.***

PLAN\_TABLE\_OUTPUT

9 rows selected.

SQL> **DROP TABLE hypt.hypt\_inmem\_tab PURGE;**

Table dropped. SQL> **EXIT**

$

## Practice 5-3: Creating and Populating In-memory External Tables

### Overview

In this practice you create and load an in-memory external table.

### Tasks

If you are working on an on-premise database, you will observe that the feature is not available on on-premise environment.

If you are working with a Cloud database, you would be able to proceed with the following steps.

1. If you completed steps 2.b and 2.c of 5-1, and step 1.a of 5-2, go on with the following steps. If not, complete steps 2.a to 2.c of 5-1 first, and then step 1.a of 5-2.
2. Create the in-memory external table HYPT.INMEM\_EXT\_TAB in PDB1 with the following attributes:
   * The default directory for external files is CENT20.
   * The fields in the records of the external files are separated by **','**.
   * The in-memory compression is FOR CAPACITY HIGH.

$ **sqlplus system@PDB1**

Enter password: *password*

SQL> **CREATE TABLE hypt.inmem\_ext\_tab (history\_event NUMBER, time\_id DATE) ORGANIZATION EXTERNAL**

**(TYPE ORACLE\_LOADER DEFAULT DIRECTORY cent20**

**ACCESS PARAMETERS (FIELDS TERMINATED BY ',')**

**LOCATION ('cent20.dat'))**

**INMEMORY MEMCOMPRESS FOR CAPACITY HIGH;**

2 3 4 5 6 7 8

Table created.

SQL>

1. Query the table.

SQL> **ALTER SESSION SET query\_rewrite\_integrity=stale\_tolerated;**

Session altered.

SQL> **SELECT \* FROM hypt.inmem\_ext\_tab;**

HISTORY\_EVENT TIME\_ID

1 01-JAN-76

2 01-JAN-15

3 01-JAN-28

4 01-JAN-37

5 01-JAN-49

6 01-FEB-59

7 01-FEB-96

8 01-FEB-97

9 01-FEB-98

10 01-FEB-98

41 12-AUG-97

11 rows selected.

SQL>

*Q1/ Is the data populated into the in-memory column store?*

SQL> **SELECT segment\_name, tablespace\_name, populate\_status FROM v$im\_segments;**

2 3

SEGMENT\_NAME TABLESPACE\_NAME POPULATE\_STAT

INMEM\_EXT\_TAB SYSTEM

COMPLETED

SQL>

##### A1/ Querying the in-memory external table initiates the population into the in- memory column store in the same way that it does for an internal table.

***Executing the DBMS\_INMEMORY.POPULATE procedure is useless.***

1. Display the in-memory attributes of the external table.

SQL> **SELECT \* FROM dba\_external\_tables WHERE owner='HYPT';**

2

OWNER TABLE\_NAME TYP TYPE\_NAME

DEF

---

DEFAULT\_DIRECTORY\_NAME

REJECT\_LIMIT

ACCESS\_

ACCESS\_PARAMETERS

PROPERTY INMEMORY INMEMORY\_COMPRESS

HYPT

**INMEM\_EXT\_TAB** SYS

ORACLE\_LOADER SYS

CENT20

0 CLOB

FIELDS TERMINATED BY ','

ALL **ENABLED FOR CAPACITY HIGH**

SQL>

1. Query the in-memory external with a degree of parallelism of 2.

SQL> **EXPLAIN PLAN FOR**

**SELECT /\*+ PARALLEL(2) \*/ \* FROM hypt.inmem\_ext\_tab;**

2

Explained.

SQL> **SELECT \* FROM TABLE(DBMS\_XPLAN.DISPLAY);**

PLAN\_TABLE\_OUTPUT

102K| 2193K| 197 (5)| 00:00:01 | Q1,00 | P->S | QC (RAND)

| 3 | PX BLOCK ITERATOR

102K| 2193K| 197 (5)| 00:00:01 |

| Q1,00 | PCWC |

|

| 4 | **EXTERNAL TABLE ACCESS INMEMORY FULL**| INMEM\_EXT\_TAB | 102K| 2193K| 197 (5)| 00:00:01 | Q1,00 | PCWP |

Note

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| | Id Rows | |  | | | Operation  Bytes | Cost (%CPU)| Time | | | | TQ | | Name  |IN-OUT| | | PQ | | |
| Distrib | | | | | | | | | | | |
| | | 0 | | | SELECT STATEMENT | | | | | | | | | |
| 102K|  | 1 | | 2193K| 197 (5)| 00:00:01  | **PX COORDINATOR** | | | | | | | | | | | | | |
| |  | 2 | | | |  | PX SEND QC (RANDOM) | | | | | | | | | |  :TQ10000 | | | |

- Degree of Parallelism is 2 because of hint

15 rows selected.

SQL>

1. Drop the HYPT schema.

SQL> **DROP USER hypt CASCADE;**

User dropped.

SQL> **EXIT**

$

# Practices for Lesson 6: Using Diagnosability Enhancements

## Practices for Lesson 6: Overview

### Overview

In these practices, you will use the Automatic SQL Diagnosis and Repair function to diagnose, get and implement recommendations for a SQL statement that executes with a poor performance and for another SQL statement that fails with an ORA-00600 error. You will also collect data with Service Request Data Collection (SRDC) for specific events like ORA-00600 error and hanging situation.

## Practice 6-1: Fixing SQL Statements by Using Automatic SQL Diagnosis and Repair

### Overview

In this practice, you diagnose, get and implement recommendations for a SQL statement that performs with a poor performance and for another SQL statement that fails with an ORA-00600 error.

### Tasks

1. Execute the /home/oracle/labs/admin/cleanup\_PDBs.sh shell script. The shell script drops all PDBs that may have been created by any of the practices in ORCL, and finally recreates PDB1. You are in *Session1*.

$ **$HOME/labs/admin/cleanup\_PDBs.sh**

…

$

1. Before starting the practice, execute the $HOME/labs/DIAG/glogin.sh shell script. It sets formatting for all columns selected in queries.

$ **$HOME/labs/DIAG/glogin.sh**

**…**

$

1. Execute the /home/oracle/labs/DIAG/table.sh shell script. The shell script creates and loads the DIAG.TAB1 table, and creates an index on the table, in PDB1.

$ **$HOME/labs/DIAG/table.sh**

…

$

1. Log in to PDB1 as DIAG and execute the query. The SQL statement executes with a poor performance.

$ **sqlplus diag@PDB1**

Enter password: *password*

SQL> **SELECT /\*+ FULL(a) FULL (b) \*/**

**sum(a.num),sum(b.num),count(\*) FROM tab1 a,tab1 b**

**WHERE a.id = b.id AND a.id = 100;**

2 3 4 5

SUM(A.NUM) SUM(B.NUM) COUNT(\*)

100

100

1

SQL>

1. Call the function to diagnose and automatically implement the recommendations to improve the performance of the SQL statement.

SQL> **DESC dbms\_sqldiag**

…

FUNCTION **SQL\_DIAGNOSE\_AND\_REPAIR** RETURNS NUMBER

Argument Name Default?

Type

In/Out

SQL>

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| SQL\_ID |  | VARCHAR2 |  | IN |  | |
| PLAN\_HASH\_VALUE |  | NUMBER |  | IN |  | DEFAULT |
| SCOPE |  | VARCHAR2 |  | IN |  | DEFAULT |
| TIME\_LIMIT |  | NUMBER |  | IN |  | DEFAULT |
| **PROBLEM\_TYPE** |  | **NUMBER** |  | IN |  | DEFAULT |
| AUTO\_APPLY\_PATCH |  | VARCHAR2 |  | IN |  | DEFAULT |

*Q1/ How do you set the problem type for the SQL statement diagnostic, as the*

*PROBLEM\_TYPE is a NUMBER attribute?*

##### A1/ Read the $ORACLE\_HOME/rdbms/admin/dbmsdiag.sql script:

* ***PERFORMANCE - User suspects this is a performance problem***

##### WRONG\_RESULTS - User suspects the query is giving inconsistent results

* ***COMPILATION\_ERROR - User sees a crash in compilation***

##### EXECUTION\_ERROR - User sees a crash in execution

* ***ALT\_PLAN\_GEN - Just explore all alternative plans***

 ***--***

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| * ***PROBLEM\_TYPE\_PERFORMANCE*** | ***CONSTANT*** | ***NUMBER*** | ***:=*** | ***1;*** |
| * ***PROBLEM\_TYPE\_WRONG\_RESULTS*** | ***CONSTANT*** | ***NUMBER*** | ***:=*** | ***2;*** |
| * ***PROBLEM\_TYPE\_COMPILATION\_ERROR*** | ***CONSTANT*** | ***NUMBER*** | ***:=*** | ***3;*** |
| * ***PROBLEM\_TYPE\_EXECUTION\_ERROR*** | ***CONSTANT*** | ***NUMBER*** | ***:=*** | ***4;*** |
| * ***PROBLEM\_TYPE\_ALT\_PLAN\_GEN*** | ***CONSTANT*** | ***NUMBER*** | ***:=*** | ***5;*** |

##### The PROBLEM\_TYPE value is 1 for performance type problems.

SQL> **SET SERVEROUTPUT ON**

SQL> **VAR incident\_id NUMBER**

SQL> **DECLARE**

**recom\_count number(10); BEGIN**

**:incident\_id := dbms\_sqldiag.sql\_diagnose\_and\_repair(**

**sql\_text => 'SELECT /\*+ FULL(a) FULL (b) \*/ sum(a.num),sum(b.num),count(\*) FROM tab1 a,tab1 b WHERE a.id = b.id and a.id = 100',**

**problem\_type => DBMS\_SQLDIAG.PROBLEM\_TYPE\_PERFORMANCE,**

**time\_limit => 1000, scope=>DBMS\_SQLDIAG.SCOPE\_COMPREHENSIVE,**

**auto\_apply\_patch => 'YES'); select count(\*) into recom\_count from**

**dba\_advisor\_recommendations where task\_name = to\_char(:incident\_id); dbms\_output.put\_line ( recom\_count || ' recommendations**

**generated for incident '||:incident\_id); end;**

**/**

**1 recommendations** generated for incident **50408**

2 3 4 5 6 7 8 9 10 11 12 13

PL/SQL procedure successfully completed.

SQL>

1. Find the recommendations generated from the diagnosis.

SQL> **SELECT finding\_id, type FROM dba\_advisor\_recommendations WHERE task\_name = to\_char(:incident\_id);**

2

FINDING\_ID TYPE

1 SQL PROFILE

SQL>

1. Report the detail of the recommendations.

**v\_tname := '50408';**

**:b\_report := dbms\_sqldiag.report\_diagnosis\_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;**

SQL> **VAR b\_report CLOB**

SQL>

SQL> **DECLARE**

**v\_tname VARCHAR2(32767); BEGIN**

**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

GENERAL INFORMATION SECTION

Tuning Task Name : 50408 Tuning Task Owner : DIAG

Workload Type : Single SQL Statement Scope : COMPREHENSIVE

Time Limit(seconds): 1000 Completion Status : COMPLETED

Started at : 05/15/2019 10:23:55 Completed at : 05/15/2019 10:23:56

Schema Name : DIAG Container Name: PDB1

SQL ID : 30u8jcwt90fw9

SQL Text : SELECT /\*+ FULL(a) FULL (b) \*/

sum(a.num),sum(b.num),count(\*) FROM tab1 a,tab1 b WHERE a.id = b.id AND a.id = 100

FINDINGS SECTION (1 finding)

* 1. SQL Profile Finding (see explain plans section below)

A potentially better execution plan was found for this statement.

Recommendation (estimated benefit: 99.83%)

- A manually-created SQL profile is present on the system.

Name: SYS\_SQLPROF\_0166b07518cc0001

Status: ENABLED

Validation results

#### The SQL profile was tested by executing both its plan and the original plan and measuring their respective execution statistics. A plan may have been only partially executed if the other could be run to completion in less time.

Original Plan With SQL Profile % Improved

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Completion Status: | | | COMPLETE | |  | COMPLETE |
| Elapsed Time (s): | | | .007051 | |  | .00008 |
| 98.86 % | | |  | |  |  |
| CPU Time (s): | | | .006919 | |  | .00008 |
| 98.84 % | | |  | |  |  |
| User I/O Time (s): | | | 0 | |  | 0 |
| Buffer Gets: 99.83 |  | | 3710 |  | 6 | |
| % |  | |  |  |  | |
| Physical Read | Requests: | |  | 0 | 0 | |
| Physical Write Requests: | |  | 0 |  | | 0 |
| Physical | |  |  |  | |  |
| Read Bytes: | | 0 |  | 0 | |  |
| Physical Write | |  |  |  | |  |

Bytes: 0 0

Rows Processed:

1 1

Fetches: 1

1

Executions: 1 1

Notes

* + 1. Statistics for the original plan were averaged over 10 executions.
    2. Statistics for the SQL profile plan were averaged over 10 executions.

PL/SQL procedure successfully completed.

SQL>

1. Check the SQL profile automatically created by the diagnosis and repair function.

SQL> **SELECT sql\_text, status FROM dba\_sql\_profiles;**

SQL\_TEXT STATUS

select /\*+ FULL(a) FULL (b) \*/ sum(a.num),sum(b.num), ENABLED coun t(\*) from tab1 a, tab1 b where a.id = b.id

and a.id = 100

SQL>

SQL> **EXPLAIN PLAN FOR**

**SELECT /\*+ FULL(a) FULL (b) \*/**

**sum(a.num),sum(b.num),count(\*) FROM tab1 a,tab1 b**

**WHERE a.id = b.id AND a.id = 100;**

2 3 4 5

Explained.

SQL> **SELECT \* FROM TABLE(DBMS\_XPLAN.DISPLAY);**

PLAN\_TABLE\_OUTPUT

Plan hash value: 3082937155

1. Verify that the poor performing SQL statement is now using the SQL profile.

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| |  | | Id By | | Operation  tes | Cost (%CPU)| Time | | | | | | | | Name | | | Rows |
| |  | | 0 |  20 | | | SELECT STATEMENT  4 (0)| 00:00:01 | | | | | | | | | | 1 |
| |  | | 1 |  20 | | | SORT AGGREGATE  | | | | | | | | | | | 1 |
| |  | | 2 |  20 | | | MERGE JOIN CARTESIAN 4 (0)| 00:00:01 | | |  |  | | |  | | | 1 |
| |  | | 3 |  10 | | | TABLE ACCESS BY INDEX 2 (0)| 00:00:01 | | | ROWID | BATCHED | | | TAB1 | | | 1 |
| |\*  | | 4 |  | | | INDEX RANGE SCAN  1 (0)| 00:00:01 | | |  |  | | | TAB1\_I | | | 1 |
| |  | | 5 |  10 | | | BUFFER SORT  2 (0)| 00:00:01 | | |  |  | | |  | | | 1 |
| |  | | 10 | 6 |  | | TABLE ACCESS BY INDEX ROWID BATCHED| 2 (0)| 00:00:01 | | | | | | TAB1 | | | 1 |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| |\* | 7 | | INDEX RANGE SCAN | | TAB1\_I | | 1 |
| | | | | 1 (0)| 00:00:01 | |  |  |
| Predicate Information (identified by operation id):  4 - access("A"."ID"=100)  7 - access("B"."ID"=100)  Hint Report (identified by operation id / Query Block Name / Object Alias):  Total hints for statement: 2 (U - Unused (2))  3 - SEL$1 / A@SEL$1  U - FULL(a) / rejected by IGNORE\_OPTIM\_EMBEDDED\_HINTS  6 - SEL$1 / B@SEL$1  U - FULL (b) / rejected by IGNORE\_OPTIM\_EMBEDDED\_HINTS  Note  - **SQL profile "SYS\_SQLPROF\_0166b07518cc0001" used for this statement**  34 rows selected.  SQL> | | | | |

1. Call the function on the poor performing SQL statement.

*Q1/ What happens if you ask the diagnosis function to explore all alternative plans for the SQL query?*

SQL> **DECLARE BEGIN**

**:incident\_id := dbms\_sqldiag.sql\_diagnose\_and\_repair(**

**sql\_text => ' SELECT /\*+ FULL(a) FULL (b) \*/ sum(a.num),sum(b.num),count(\*) FROM tab1 a,tab1 b WHERE a.id = b.id AND a.id = 100',**

**problem\_type => DBMS\_SQLDIAG.PROBLEM\_TYPE\_ALT\_PLAN\_GEN, auto\_apply\_patch => 'YES');**

**END;**

**/**

2

3

4

5

6

7

8

DECLARE

\*

ERROR at line 1:

ORA-20001: '5' - invalid problem type

ORA-06512: at "SYS.DBMS\_SQLDIAG\_INTERNAL", line 587 ORA-06512: at "SYS.DBMS\_SQLDIAG", line 391

ORA-06512: at "SYS.DBMS\_SQLDIAG", line 2454

ORA-06512: at line 3

SQL>

##### A1/ The problem type PROBLEM\_TYPE\_ALT\_PLAN\_GEN refers to the constant 5 (refer task 5). As there are no other plans in DBA\_SQL\_PLAN\_BASELINES for the same query, the function cannot satisfy the condition to find alternate plans.

1. You will now test a failing SQL statement for which SQL Diagnose and Repair provides and implements a patch. Execute the /home/oracle/labs/DIAG/crash\_delete.sql SQL script. The SQL statement fails with an ORA-00600 error. Click enter after each pause.

SQL> **CONNECT system@PDB1** Enter password: *password* Connected.

SQL> **@/home/oracle/labs/DIAG/crash\_delete.sql**

SQL> set echo on SQL>

SQL> -- This example generates a workaround for a crash. This bug has already

SQL> -- been fixed but we toggle the bug fix using an underscore parameter

SQL> -- which uses the (internal) feature called bug fix control.

SQL> -- This script will pause periodically to allow you to read the comments

SQL> -- and see the output of the previous command on the screen. Just press

SQL> -- return to make the demo resume. SQL>

SQL> pause

SQL>

SQL> -- To begin the demo we will create the user diag and grant SQL> -- advisor privileges to him.

SQL>

SQL> pause

SQL>

SQL> grant connect, resource, dba, query rewrite, unlimited tablespace to diag identified by *password*;

Grant succeeded.

SQL>

SQL> alter user diag account unlock;

User altered.

SQL>

SQL> -- Next we need to create and populate the table used by the demo.

SQL> -- We will also create an index on the table; SQL>

SQL> pause

SQL> connect diag/*password*@PDB1; Connected.

SQL>

SQL> drop table simple\_table; drop table simple\_table

\* ERROR at line 1:

ORA-00942: table or view does not exist

SQL>

SQL> create table simple\_table(a varchar(40), b number, c varchar(240), d varchar(240));

Table created.

SQL>

SQL> create index tc on simple\_table(b, d, a);

Index created.

SQL>

SQL> insert into simple\_table values('a', 1, 'b', 'c');

1 row created.

SQL> insert into simple\_table values('a', 1, 'x', 'c');

1 row created.

SQL> insert into simple\_table values('e', 2, 'f', 'g');

1 row created.

SQL>

SQL> -- In order to crash the system we need to switch off the code line that

SQL> -- normal protects against this type of crash. We can switch off the code

SQL> -- using the (internal) feature called bug fix control. SQL>

SQL> pause

SQL>

SQL> -- switch the code

SQL> alter system set "\_fix\_control"="5868490:OFF";

System altered.

SQL> -- alter session set optimizer\_dynamic\_sampling = 0; SQL>

SQL> -- Now that the code line has been switched off lets get the execution

SQL> -- plan for a simple delete statement. SQL>

SQL> pause

SQL>

SQL> --- explain the plan SQL> explain plan for delete

2 /\*+

3 USE\_HASH\_AGGREGATION(@"SEL$80F8B8C6")

4 USE\_HASH(@"SEL$80F8B8C6" "T1"@"DEL$1")

5 LEADING(@"SEL$80F8B8C6" "T2"@"SEL$1" "T1"@"DEL$1")

6 FULL(@"SEL$80F8B8C6" "T1"@"DEL$1")

7 FULL(@"SEL$80F8B8C6" "T2"@"SEL$1")

|  |  |  |
| --- | --- | --- |
| 8 |  | OUTLINE(@"DEL$1") |
| 9 |  | OUTLINE(@"SEL$1") |
| 10 |  | OUTLINE(@"SEL$AD0B6B07") |
| 11 |  | OUTLINE(@"SEL$7D4DB4AA") |
| 12 |  | UNNEST(@"SEL$1") |
| 13 |  | OUTLINE(@"SEL$75B5BFA2") |
| 14 |  | MERGE(@"SEL$7D4DB4AA") |
| 15 |  | OUTLINE\_LEAF(@"SEL$80F8B8C6") |
| 16 |  | ALL\_ROWS |
| 17 |  | OPT\_PARAM('\_optimizer\_cost\_model' 'fixed') |
| 18 |  | DB\_VERSION('11.1.0.7') |
| 19 |  | OPTIMIZER\_FEATURES\_ENABLE('11.1.0.7') |
| 20 |  | NO\_INDEX(@"SEL$1" "T2"@"SEL$1") |
| 21 | \*/ |  |
| 22 from simple\_table t1 where t1.a = 'a' and rowid <> (select max(rowid) from simple\_table t2 where t1.a= t2.a and t1.b = t2.b and t1.d=t2.d);  Explained. SQL>  SQL> --- display the plan  SQL> select plan\_table\_output from table(dbms\_xplan.display('plan\_table',null));  PLAN\_TABLE\_OUTPUT  Plan hash value: 1481897562  | Id | Operation | Name | Rows | Bytes | Cost | | |

(%CPU)| Time |

|  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| | 0 |  6 | | | DELETE STATEMENT | | | | | | | | 1 | | | 338 | | |
| (34)| | | | 00:00:01 | | | |  |  | |  |  |  |  |
| | | 1 | | | DELETE |  | | | | SIMPLE\_TABLE | | | | | | | | |
|  |  | | |  | | | |  |  |  |  | |  | |
| |\* | 2 | | | FILTER |  | | | |  | | | | | | | | |
|  |  | | |  | | | |  |  |  |  | |  | |
| | 6 | 3 | | | HASH GROUP | | BY | | | | | | 1 | | | 338 | | |

(34)| 00:00:01 |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| |\* 4 | HASH JOIN | 5 | | | 1 | | | 338 | | |
| (20)| 00:00:01 | |  |  |  |  |  |
| |\* 5 | TABLE ACCESS FULL| 2 | SIMPLE\_TABLE | | 2 | | | 338 | | |
| (0)| 00:00:01 | |  |  |  |  |  |
| |\* 6 | TABLE ACCESS FULL| | SIMPLE\_TABLE | | 2 | | | 338 | | |
| 2  (0)| 00:00:01 |  Predicate Information (identified by operation id):  2 - filter(ROWID<>MAX(ROWID))  4 - access("T1"."A"="T2"."A" AND "T1"."B"="T2"."B" AND "T1"."D"="  T2"."D")  5 - filter("T2"."A"='a')  6 - filter("T1"."A"='a')  Note  - dynamic statistics used: dynamic sampling (level=2)  25 rows selected.  SQL>  SQL> -- The plan shows that we we will do a full table scan oun r.  SQL> -- If we execute this simple system it will crash the system.  SQL>  SQL> Pause  SQL> --- This statement caused the system to crash. SQL> delete /\*+ USE\_HASH\_AGGREGATION(@"SEL$80F8B8C6")  USE\_HASH(@"SEL$80F8B8C6" "T1"@"DEL$1") LEADING(@"SEL$80F8B8C6" "T2"@"SEL$1" "T1"@"DEL$1") FULL(@"SEL$80F8B8C6" "T1"@"DEL$1") FULL(@"SEL$80F8B8C6" "T2"@"SEL$1") OUTLINE(@"DEL$1") OUTLINE(@"SEL$1") OUTLINE(@"SEL$AD0B6B07") OUTLINE(@"SEL$7D4DB4AA") UNNEST(@"SEL$1") OUTLINE(@"SEL$75B5BFA2") MERGE(@"SEL$7D4DB4AA") OUTLINE\_LEAF(@"SEL$80F8B8C6") ALL\_ROWS  OPT\_PARAM('\_optimizer\_cost\_model' 'fixed') | | | | | |

DB\_VERSION('11.1.0.7') OPTIMIZER\_FEATURES\_ENABLE('11.1.0.7')

NO\_INDEX(@"SEL$1" "T2"@"SEL$1") \*/ from simple\_table t1 where t1.a = 'a' and rowid <> (select max(rowid) from simple\_table t2 where t1.a= t2.a and t1.b = t2.b and t1.d=t2.d);

delete /\*+ USE\_HASH\_AGGREGATION(@"SEL$80F8B8C6") USE\_HASH(@"SEL$80F8B8C6" "T1"@"DEL$1") LEADING(@"SEL$80F8B8C6" "T2"@"SEL$1" "T1"@"DEL$1") FULL(@"SEL$80F8B8C6" "T1"@"DEL$1") FULL(@"SEL$80F8B8C6" "T2"@"SEL$1") OUTLINE(@"DEL$1") OUTLINE(@"SEL$1") OUTLINE(@"SEL$AD0B6B07") OUTLINE(@"SEL$7D4DB4AA") UNNEST(@"SEL$1") OUTLINE(@"SEL$75B5BFA2") MERGE(@"SEL$7D4DB4AA") OUTLINE\_LEAF(@"SEL$80F8B8C6") ALL\_ROWS

OPT\_PARAM('\_optimizer\_cost\_model' 'fixed') DB\_VERSION('11.1.0.7') OPTIMIZER\_FEATURES\_ENABLE('11.1.0.7')

NO\_INDEX(@"SEL$1" "T2"@"SEL$1") \*/ from simple\_table t1 where t1.a = 'a' and rowid <> (select max(rowid) from simple\_table t2 where t1.a= t2.a and t1.b = t2.b and t1.d=t2.d)

\*

ERROR at line 1:

**ORA-00600: internal error code, arguments: [13011]**, [73039], [4230321], [0], [4230321], [17], [], [], [], [], [], []

SQL>

1. Call the function to diagnose and automatically implement the patch for the failing SQL statement.
2. Find the SQL\_ID for the failing statement.

SQL> **SELECT sql\_id FROM v$sql**

**WHERE sql\_text LIKE 'delete%USE\_HASH%';**

2

SQL\_ID

53390wmjjqgra

1 row selected.

SQL>

1. Call the function with the SQL\_ID as input and the appropriate problem type. (Refer task 5).

SQL> **SET SERVEROUTPUT ON**

SQL> **VAR incident\_id NUMBER**

SQL> **DECLARE**

**recom\_count number(10);**

**BEGIN**

**:incident\_id := dbms\_sqldiag.sql\_diagnose\_and\_repair(**

**sql\_id => '53390wmjjqgra', problem\_type => 4, auto\_apply\_patch => 'YES');**

**SELECT count(\*) into recom\_count FROM dba\_advisor\_recommendations**

**WHERE task\_name = to\_char(:incident\_id);**

**dbms\_output.put\_line ( recom\_count || ' recommendations generated for incident '||:incident\_id);**

**END;**

**/**

**1** recommendations generated for incident **96043**

2 3 4 5 6 7 8 9 10 11 12 13

PL/SQL procedure successfully completed.

SQL>

1. Find the recommendations generated from the diagnosis.

SQL> **SELECT finding\_id, type FROM dba\_advisor\_recommendations WHERE task\_name = to\_char(:incident\_id);**

2

FINDING\_ID TYPE

1 **SQL PATCH**

1 row selected.

SQL>

1. Find the SQL patch.

SQL> **SELECT name, task\_exec\_name, status FROM dba\_sql\_patches**

**WHERE AND**

2 3

NAME

**name = to\_char(:incident\_id)**

**sql\_text LIKE 'delete%';**

TASK\_EXEC\_NAME STATUS

96043

EXEC\_1

ENABLED

1 row selected.

SQL>

*Q1/ Is the SQL patch implemented?*

#### SQL> EXPLAIN PLAN FOR delete

**/\*+**

#### USE\_HASH\_AGGREGATION(@"SEL$80F8B8C6") USE\_HASH(@"SEL$80F8B8C6" "T1"@"DEL$1") LEADING(@"SEL$80F8B8C6" "T2"@"SEL$1" "T1"@"DEL$1") FULL(@"SEL$80F8B8C6" "T1"@"DEL$1") FULL(@"SEL$80F8B8C6" "T2"@"SEL$1") OUTLINE(@"DEL$1")

**OUTLINE(@"SEL$1") OUTLINE(@"SEL$AD0B6B07") OUTLINE(@"SEL$7D4DB4AA") UNNEST(@"SEL$1") OUTLINE(@"SEL$75B5BFA2") MERGE(@"SEL$7D4DB4AA") OUTLINE\_LEAF(@"SEL$80F8B8C6") ALL\_ROWS**

#### OPT\_PARAM('\_optimizer\_cost\_model' 'fixed') DB\_VERSION('11.1.0.7') OPTIMIZER\_FEATURES\_ENABLE('11.1.0.7') NO\_INDEX(@"SEL$1" "T2"@"SEL$1")

**\*/**

#### from simple\_table t1 where t1.a = 'a' and rowid <> (select max(rowid) from simple\_table t2 where t1.a= t2.a and t1.b = t2.b and t1.d=t2.d);

2 3 4 5 6 7 8 9 10 11 12 13 14

15 16 17 18 19 20 21 22

Explained.

#### SQL> SELECT plan\_table\_output

**FROM TABLE(dbms\_xplan.display('plan\_table',null));**

PLAN\_TABLE\_OUTPUT

Plan hash value: 3259336479

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| | Id Cost | | | Operation (%CPU)| Time | | | | | | Name | | | Rows | | | | Bytes | | |  |
| | 3 | 0 | (0)| | | DELETE STATEMENT 00:00:01 | | | | |  | | | 1 | | | | 169 | | | |
| | | 1 | | | DELETE | | | | SIMPLE\_TABLE | | |  | | | |  | | | |
| |\* | 2 | | | FILTER | | | |  | | |  | | | |  | | | |

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| |\* 1 | 3 | (0)| | INDEX FULL SCAN 00:00:01 | | | | TC | | | 2 | | | 338 | | |
| | | 4 | | SORT AGGREGATE | | |  | | | 1 | | | 169 | | |

Predicate Information (identified by operation id):

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| |\* | 5 | | TABLE ACCESS FULL| | SIMPLE\_TABLE | | | 1 | | | 169 | | |
| 2 | (0)| | 00:00:01 | |  |  |  |  |  |  |

2 - filter(ROWID<> (SELECT /\*+ UNNEST UNNEST NO\_INDEX ("T2") NO\_INDEX ("T2") \*/ MAX(ROWID) FROM "SIMPLE\_TABLE" "T2" WHERE "T

2"."A"=:B1 AND "T2"."B"=:B2 AND "T2"."D"=:B3))

3 - access("T1"."A"='a')

filter("T1"."A"='a')

5 - filter("T2"."A"=:B1 AND "T2"."B"=:B2 AND "T2"."D"=:B3)

Hint Report (identified by operation id / Query Block Name / Object Alias):

Total hints for statement: 7 (U - Unused (2), N - Unresolved (5))

0 - SEL$7D4DB4AA

N - MERGE(@"SEL$7D4DB4AA")

0 - SEL$80F8B8C6

N - FULL(@"SEL$80F8B8C6" "T1"@"DEL$1") N - FULL(@"SEL$80F8B8C6" "T2"@"SEL$1")

N - LEADING(@"SEL$80F8B8C6" "T2"@"SEL$1" "T1"@"DEL$1") N - USE\_HASH(@"SEL$80F8B8C6" "T1"@"DEL$1")

4 - SEL$1

U - UNNEST(@"SEL$1") / hint overridden by NO\_QUERY\_TRANSFORMATION

5 - SEL$1 / T2@SEL$1

U - NO\_INDEX(@"SEL$1" "T2"@"SEL$1")

Note

* dynamic statistics used: dynamic sampling (level=2)

#### SQL patch "96043" used for this statement

46 rows selected.

SQL>

##### A1/ The SQL patch is automatically implemented.

1. Re-execute the failing SQL statement with the implemented patch.

SQL> **delete /\*+ USE\_HASH\_AGGREGATION(@"SEL$80F8B8C6") USE\_HASH(@"SEL$80F8B8C6" "T1"@"DEL$1") LEADING(@"SEL$80F8B8C6" "T2"@"SEL$1" "T1"@"DEL$1") FULL(@"SEL$80F8B8C6" "T1"@"DEL$1") FULL(@"SEL$80F8B8C6" "T2"@"SEL$1") OUTLINE(@"DEL$1") OUTLINE(@"SEL$1") OUTLINE(@"SEL$AD0B6B07") OUTLINE(@"SEL$7D4DB4AA") UNNEST(@"SEL$1") OUTLINE(@"SEL$75B5BFA2") MERGE(@"SEL$7D4DB4AA") OUTLINE\_LEAF(@"SEL$80F8B8C6") ALL\_ROWS**

**OPT\_PARAM('\_optimizer\_cost\_model' 'fixed') DB\_VERSION('11.1.0.7') OPTIMIZER\_FEATURES\_ENABLE('11.1.0.7')**

**NO\_INDEX(@"SEL$1" "T2"@"SEL$1") \*/ from simple\_table t1 where t1.a = 'a' and rowid <> (select max(rowid) from simple\_table t2 where t1.a= t2.a and t1.b = t2.b and t1.d=t2.d);**

1 row deleted.

SQL> **ROLLBACK;**

Rollback complete. SQL> **EXIT**

$

Observe that the statement does not fail anymore.

1. Set the fix for the error back ON and clean up the DIAG schema.

$ **/home/oracle/labs/DIAG/cleanup\_crash.sh**

…

$

## Practice 6-2: Creating SRDCs for Specific Events

### Overview

In this practice, you will collect diagnostic files for an ORA-00600 error and also for a hanging instance startup, and upload files directly to your service request.

### Tasks

1. Log in as root to the server and install Trace File Analyzer in $ORACLE\_BASE/tfa.

# **cd /u01/app/oracle**

# **unzip /home/oracle/labs/DIAG/TFA-LINUX\_v19.2.1.zip**

Archive: /home/oracle/labs/DIAG/TFA-LINUX\_v18.3.3.zip inflating: README.txt

inflating: installTFA-LINUX

#

# **./installTFA-LINUX**

TFA Installation Log will be written to File :

/tmp/tfa\_install\_6105\_2018\_10\_15-11\_55\_44.log

Starting TFA installation

TFA Version: 192100 Build Date: 201810050542

Enter a location for installing TFA (/tfa will be appended if not supplied) [**/u01/app/oracle/tfa**]:

Running Auto Setup for TFA as user oracle...

Would you like to do a [L]ocal only or [C]lusterwide installation ? [L|l|C|c] [C] : **L**

Installing TFA now...

Discovering Nodes and Oracle resources Starting Discovery...

No Grid Infrastructure Discovered on this system . . . . .

*hostname*

Searching for running databases...

* 1. ORCL

Searching out ORACLE\_HOME for selected databases...

Getting Oracle Inventory...

ORACLE INVENTORY: /u01/app/oraInventory

Discovery Complete...

TFA Will be Installed on *hostname*...

TFA will scan the following Directories

++++++++++++++++++++++++++++++++++++++++++++

. .

| *hostname* |

+ +

| Trace Directory Resource | Resource |

+ +

| /u01/app/oraInventory/ContentsXML | INSTALL |

| /u01/app/oraInventory/logs | INSTALL |

| /u01/app/oracle/cfgtoollogs | CFGTOOLS |

| /u01/app/oracle/diag/rdbms/orcl/ORCL/cdump | RDBMS |

| /u01/app/oracle/diag/rdbms/orcl/ORCL/trace | RDBMS |

| /u01/app/oracle/diag/tnslsnr | TNS |

| /u01/app/oracle/diag/tnslsnr/your/listener/cdump| TNS |

| /u01/app/oracle/diag/tnslsnr/*your\_hostname*/listener/trace| TNS

|

| /u01/app/oracle/product/19.3.0/dbhome\_1/cfgtoollogs|CFGTOOLS |

| /u01/app/oracle/product/19.3.0/dbhome\_1/install | INSTALL |

| /u01/app/oracle/product/19.3.0/dbhome\_1/rdbms/log | RDBMS | ' .

Installing TFA on *hostname*:

HOST: *hostname* TFA\_HOME: /u01/app/oracle/tfa/*hostname*/tfa\_home

.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| | Host | | Status of TFA | PID | | Port | | Version | | Build |
| ID | | |  |  |  |

+ + + + + +

| *hostname* | RUNNING | 7955 | 42182 | 19.2.1.0.0 | 18330020181005054218 |

' + + + + +

Running Inventory in All Nodes...

Enabling Access for Non-root Users on *hostname*... ERROR:

/u01/app/oracle/tfa/*hostname*/tfa\_home/internal/cached\_kv.out does not exists

Adding default users to TFA Access list...

Summary of TFA Installation:

. .

| *hostname* |

+ + +

| Parameter | Value |

+ + +

| Install location | /u01/app/oracle/tfa/*hostname*/tfa\_home |

| Repository location | /u01/app/oracle/tfa/repository |

| Repository usage | 0 MB out of 10240 MB | ' + '

TFA is successfully installed...

Usage : /u01/app/oracle/tfa/bin/tfactl <command> [options]

commands:collect|collection|analyze|ips|run|start|stop|enable|di sable|status|print|access|purge|directory|host|receiver|set|tool status|uninstall|nosetfa|syncnodes|setupmos|upload|availability| rest|events|search|changes|isa

For detailed help on each command use:

/u01/app/oracle/tfa/bin/tfactl <command> -help

#

If TFA is already installed, you will get the following output.

#### # ./installTFA-LINUX

TFA Installation Log will be written to File :

/tmp/tfa\_install\_7099\_2019\_05\_15-15\_11\_41.log

Starting TFA installation

TFA Version: 192100 Build Date: 201904251105

TFA HOME : /u01/app/oracle/tfa/*hostname*/tfa\_home

Installed Build Version: 183300 Build Date: 201810050542

TFA is already installed. Upgrading TFA

TFA Upgrade Log : /u01/app/oracle/tfa/*hostname*/tfapatch.log

TFA-00002 Oracle Trace File Analyzer (TFA) is not running TFA-00002 Oracle Trace File Analyzer (TFA) is not running Unable to determine the status of TFA in other nodes.

TFA will be upgraded on Node *hostname*:

Do you want to continue with TFA Upgrade ? [Y|N] [Y]: Y

Upgrading TFA on *hostname* :

Stopping TFA Support Tools...

Shutting down TFA for Patching...

Shutting down TFA

Removed symlink /etc/systemd/system/multi- user.target.wants/oracle-tfa.service.

Removed symlink

/etc/systemd/system/graphical.target.wants/oracle-tfa.service.

. . . . .

. . .

Successfully shutdown TFA..

No Berkeley DB upgrade required

Copying TFA Certificates...

Starting TFA in *hostname*...

Starting TFA..

Created symlink from /etc/systemd/system/multi- user.target.wants/oracle-tfa.service to

/etc/systemd/system/oracle-tfa.service.

Created symlink from

/etc/systemd/system/graphical.target.wants/oracle-tfa.service to

/etc/systemd/system/oracle-tfa.service.

Waiting up to 100 seconds for TFA to be started..

. . . . .

Successfully started TFA Process..

. . . . .

TFA Started and listening for commands

Enabling Access for Non-root Users on *hostname*...

. .

| Host | TFA Version |TFA Build ID | Upgrade Status |

+ + + + +

| *hostname* | 19.2.1.0.0 |19210020190425110550| UPGRADED |

' + + + '

#

* 1. Switch as oracle and check that the TFA Collector is running.

# **exit**

$ **/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|setupmos|upload|events|search|changes|blac kout

For detailed help on each command use:

/u01/app/oracle/tfa/bin/tfactl <command> -help

$

|  |  |  |  |
| --- | --- | --- | --- |
| $ **cd /u01/app/oracle/tfa/bin**  $ **./tfactl status**  . | | | |
| | Host ID | | Status of TFA | PID | Port  | Inventory Status | | | Version | | Build |

* 1. Start a Service Request Data Collection (SRDC) for all ORA-00600 errors that happened in

+ + + + + +

| *hostname* | RUNNING | 28467 | 55757 | 19.2.1.0.0 | 19210020190425110550 | COMPLETE

'

$

+

|

+ + + +

ORCL.

*Q1/ How do you get the list of possible types of SRDC?*

#### $ ./tfactl collect -srdc -help

Service Request Data Collection (SRDC).

Usage : /u01/app/oracle/tfa/bin/tfactl collect -srdc

<srdc\_profile> [-tag <tagname>] [-z <filename>] [-last <n><h|d>|

-from <time> -to <time> | -for <time>] -database <database>

-tag <tagname> The files will be collected into tagname directory inside repository

-z <zipname> The collection zip file will be given this name within the TFA collection repository

-last <n><h|d> Files from last 'n' [d]ays or 'n' [h]ours

-since Same as -last. Kept for backward compatibility.

-from "Mon/dd/yyyy hh:mm:ss" From <time> or "yyyy-mm-dd hh:mm:ss"

or "yyyy-mm-ddThh:mm:ss" or "yyyy-mm-dd"

-to "Mon/dd/yyyy hh:mm:ss" To <time> or "yyyy-mm-dd hh:mm:ss"

or "yyyy-mm-ddThh:mm:ss" or "yyyy-mm-dd"

-for "Mon/dd/yyyy" For <date>. or "yyyy-mm-dd"

<srdc\_profile> can be any of the following,

**Listener\_Services** SRDC - Data Collection for TNS-12516 / TNS-12518 / TNS-12519 / TNS-12520.

Naming\_Services SRDC - Data Collection for ORA-12154 / ORA-12514 / ORA-12528.

**ORA-00020** SRDC for database ORA-00020 Maximum number of processes exceeded

**ORA-00060** SRDC for ORA-00060. Internal error code.

**ORA-00600** SRDC for ORA-00600. Internal error code.

**ORA-00700** SRDC for ORA-00700. Soft internal error.

**ORA-01031** SRDC - How to Collect Standard Information for ORA - 1031 /ORA -1017 during SYSDBA connections

**ORA-01555** SRDC for database ORA-01555 Snapshot too Old problems

**ORA-01578** SRDC - Required diagnostic Data Collection for NOLOGGING ORA-1578/ORA-26040 DBV-00201.

**ORA-01628** SRDC for database ORA-01628 Snapshot too Old problems

**ORA-04030** SRDC for ORA-04030. OS process private memory was exhausted.

**ORA-04031** SRDC for ORA-04031. More shared memory is needed in the shared/streams pool.

**ORA-07445** SRDC for ORA-07445. Exception encountered, core dump.

**ORA-08102** SRDC - Required diagnostic Data Collection for ORA-08102.

**ORA-08103** SRDC - Required diagnostic Data Collection for ORA-08103.

ORA-22924 SRDC - ORA-22924 or ORA-1555 on LOB

data: Checklist of Evidence to Supply (Doc ID 1682707.1)

**ORA-27300** SRDC for ORA-27300. OS system

dependent operation:open failed with status: (status).

**ORA-27301** SRDC for ORA-27301. OS failure message: (message).

**ORA-27302** SRDC for ORA-27302. failure occurred at: (module).

**ORA-29548** SRDC - Providing Supporting Information for Oracle JVM Issues (Doc ID 2175568.1)

**ORA-30036** SRDC for database ORA-30036 Unable to extend Undo Tablespace pproblems

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **TNS-12154** | SRDC | - Data | Collection | for | TNS-12154. |
| TNS-12514 | SRDC | - Data | Collection | for | TNS-12514. |
| TNS-12516 | SRDC | - Data | Collection | for | TNS-12516. |
| TNS-12518 | SRDC | - Data | Collection | for | TNS-12518. |
| TNS-12519 | SRDC | - Data | Collection | for | TNS-12519. |
| TNS-12520 | SRDC | - Data | Collection | for | TNS-12520. |
| TNS-12528 | SRDC | - Data | Collection | for | TNS-12528. |

**dbasm** SRDC AUTOMATION: ENHANCE ASM/DBFS/DNFS/ACFS COLLECTIONS

**dbaudit** SRDC - How to Collect Standard Information for Database Auditing

**dbawrspace** SRDC for database AWR space problems

**dbblockcorruption** SRDC - Required diagnostic Data Collection for Alert Log Message "Corrupt block relative dba".

**dbdataguard** SRDC to capture diagnostic data for Data Guard issues

**dbdatapatch** SRDC - Data Collection for Datapatch issues.

**dbexp** SRDC - How to Collect Information for Troubleshooting Export (EXP) Related Problems

**dbexpdp** SRDC - diagnostic Collection for DataPump Export Generic Issues

**dbexpdpapi** SRDC - diagnostic Collection for DataPump Export API Issues

**dbexpdpperf** SRDC - diagnostic Collection for DataPump Export Performance Issues

**dbexpdptts** SRDC - Data to supply for Transportable Tablespace Datapump and original EXPORT, IMPORT

**dbfs** SRDC for dbfs.

**dbggclassicmode** SRDC for DOC ID 1913426.1, 1913376.1 and 1912964.1

**dbggintegratedmode** SRDC for GoldenGate extract/replicat abends problems.

**dbimp** SRDC - diagnostic Collection for Traditional Import Issues

**dbimpdp** SRDC - diagnostic Collection for DataPump Import (IMPDP) Generic Issues

**dbimpdpperf** SRDC - diagnostic Collection for DataPump Import (IMPDP) Performance Issues

**dbinstall** SRDC for Oracle RDBMS install problems.

**dbinvalidcomp** SRDC - Invalid Components and Objects :

Checklist of Evidence to Supply

**dbinvalidobj** SRDC - Objects Getting Invalidated:

Checklist of Evidence to Supply

**dbpartition** SRDC - Data to Supply for Create/Maintain Partitioned/Subpartitioned Table/Index Issues

**dbpartitionperf** SRDC - Data to Supply for Slow Create/Alter/Drop Commands Against Partitioned Table/Index

**dbpatchconflict** SRDC for Oracle RDBMS patch conflict problems.

**dbpatchinstall** SRDC for Single Instance Database Shutdown problems

**dbperf** SRDC for database performance problems. **dbpreupgrade** SRDC for database preupgrade problems. **dbrac** SRDC AUTOMATION: ENHANCE

ASM/DBFS/DNFS/ACFS COLLECTIONS

**dbracinst** SRDC AUTOMATION: ENHANCE ASM/DBFS/DNFS/ACFS COLLECTIONS

**dbrman** SRDC - Required diagnostic data collection for RMAN related issues, such as backup, maintenance, restore and recover, RMAN-08137 or RMAN-08120

**dbrman600** SRDC - Required diagnostic data collection for RMAN-00600 error (Doc ID 2045195.1).

**dbrmanperf** SRDC - Required diagnostic data collection for RMAN Performance(1671509.1).

**dbscn** SRDC for database SCN problems.

**dbshutdown** SRDC for Single Instance Database Shutdown problems

dbspatialexportimport SRDC - Data Collection for Oracle Spatial Export/Import Issues.

dbspatialinstall SRDC - Data Collection for Oracle Spatial Installation Issues.

**dbsqlperf** SRDC - How to Collect Standard Information for a SQL Performance Problem Using TFA Collector.

**dbstartup** SRDC for Single Instance Database Startup problems

**dbtde** SRDC - How to Collect Standard Information for Transparent Data Encryption (TDE) (Doc ID 1905607.1)

dbtextinstall SRDC - Data Collection for Oracle Text Installation Issues - 12c.

dbtextupgrade SRDC - Data Collection for Oracle Text Upgrade Issues - 12c.

**dbundocorruption** SRDC - Required diagnostic Data Collection for UNDO Corruption.

**dbunixresources** SRDC to capture diagnostic data for DB issues related to O/S resources

**dbupgrade** SRDC for database upgrade problems.

**dbxdb** SRDC for database XDB Installation and Invalid Object problems

**dnfs** SRDC for DNFS.

**emagentperf** EM SRDC - Collect diagnostic Data for EM Agent Performance Issues.

**emcliadd** EM SRDC - Errors during the adding of a database/listener/ASM target via EMCLI.

**emclusdisc** EM SRDC - Cluster target, cluster (RAC) database or ASM target is not discovered.

**emdbsys** EM SRDC - Database system target is not discovered/detected/removed/renamed correctly.

**emdebugoff** SRDC for unsetting EM Debug.

**emdebugon** SRDC for setting EM Debug.

**emgendisc** EM SRDC - General error is received when discovering or removing a database/listener/ASM target.

**emmetricalert** SRDC for EM Metric Events not Raised and General Metric Alert Related Issues.

**emomscrash** SRDC - Collect diagnostic Data for all Enterprise Manager OMS Crash / Restart Performance Issues.

**emomsheap** SRDC - Collecting diagnostic Data for Enterprise Manager OMS Heap Usage Alert Performance Issues.

**emomshungcpu** SRDC - Collecting diagnostic Data for Enterprise Manager OMS hung or High CPU Usage Performance Issues.

**emprocdisc** EM SRDC - Database/listener/ASM target is not discovered/detected by the discovery process.

**emrestartoms** EM SRDC - Re-start OMS.

**emtbsmetric** SRDC for EM Tablespace Space Used Metric Issues.

**esexalogic** SRDC - Exalogic Full Exalogs Data Collection Information.

exsmartscan SRDC - Exadata: Smart Scan Not Working Issues.

**ggintegratedmodenodb** SRDC for GoldenGate extract/replicat abends problems.

gridinfra SRDC AUTOMATION: ENHANCE ASM/DBFS/DNFS/ACFS COLLECTIONS

gridinfrainst SRDC AUTOMATION: ENHANCE ASM/DBFS/DNFS/ACFS COLLECTIONS

**internalerror** SRDC for all other types of internal database errors.

zlgeneric SRDC - Zero Data Loss Recovery Appliance (ZDLRA) Data Collection.

$

##### A1/ The -help option is always helpful to get the possible values of options.

* + 1. Collect diagnostic data for your service request SR12345 (this is a fake SR) for the

ORA-00600 errors that happened in practice 6-1.

$ **./tfactl collect -srdc ORA-00600 -sr SR12345**

MOS setup is not done. It is needed to upload collection to SR Run: tfactl setupmos

$

It is possible that the command does not request the MOS setup. In this case, jump to step b.

$ **./tfactl setupmos**

Access Denied: Only TFA Admin can run this command

$

The wallet file is secured to be read write by the root user only. This is the reason why you have to log on as root.

$ **su -**

Password: *password*

Last login: Tue Oct 16 07:38:22 UTC 2018 on pts/0 # **cd /u01/app/oracle/tfa/bin**

# **./tfactl setupmos** Enter User Id: **Test** Enter Password: *password*

Wallet does not exist ... creating Wallet created successfully

USER details added/updated in the wallet PASSWORD details added/updated in the wallet

SUCCESS - CERTIMPORT - Successfully imported certificate

#

It is possible that the command raises errors. Nevertheless, continue with b.

* + 1. Switch back to oracle and collect the diagnostic data related to the 2nd occurrence of the ORA-00600 errors in ORCL for the SR12345.

# **exit**

#### $ ./tfactl collect -srdc ORA-00600 -sr SR12345

Enter the time of the ORA-00600 [YYYY-MM-DD HH24:MI:SS,<RETURN>=ALL] :

Enter the Database Name [<RETURN>=ALL] : **ORCL**

* + - 1. May/15/2019 10:30:49: [orcl] ORA-00600: internal error code, arguments: [13011], [73039], [4230321], [0], [4230321], [17], [], [], [], [], [], []

Please choose the event : 1-1 [1] **1**

Selected value is : 1 ( May/15/2019 10:30:49 )

Scripts to be run by this srdc: ipspack rdahcve1210 rdahcve1120 rdahcve1110

Components included in this srdc: OS DATABASE NOCHMOS Collecting data for local node(s)

Scanning files from May/15/2019 08:30:49 to May/15/2019 12:30:49

Collection Id : 20190515154751*hostname*

Detailed Logging at :

/u01/app/oracle/tfa/repository/srdc\_ora600\_collection\_Wed\_May\_15

\_15\_47\_52\_UTC\_2019\_node\_local/diagcollect\_20190515154751\_

*hostname*.log

2019/05/15 15:47:55 UTC : NOTE : Any file or directory name containing the string .com will be renamed to replace .com with dotcom

2019/05/15 15:47:55 UTC : Collection Name : tfa\_srdc\_ora600\_Wed\_May\_15\_15\_47\_52\_UTC\_2019.zip

2019/05/15 15:47:55 UTC : Getting list of files satisfying time range [05/15/2019 08:30:49 UTC, 05/15/2019 12:30:49 UTC]

2019/05/15 15:47:55 UTC : Collecting additional diagnostic information...

2019/05/15 15:48:10 UTC : Collecting ADR incident files...

2019/05/15 15:48:18 UTC : Completed collection of additional diagnostic information...

2019/05/15 15:48:20 UTC : Completed Local Collection 2019/05/15 15:48:20 UTC : Uploading collection to SR - SR12345 2019/05/15 15:48:21 UTC : Failed to upload collection to SR

. .

| Collection Summary |

+ + + + +

| Host | Status | Size | Time |

+ + + + +

| *hostname* | Completed | 2.9MB | 25s | ' + + + '

Logs are being collected to:

/u01/app/oracle/tfa/repository/srdc\_ora600\_collection\_Wed\_May\_15

\_16\_08\_23\_UTC\_2019\_node\_local

/u01/app/oracle/tfa/repository/srdc\_ora600\_collection\_Wed\_May\_15

\_16\_08\_23\_UTC\_2019\_node\_local/*hostname*.tfa\_srdc\_ora600\_Wed\_May\_1 5\_16\_08\_23\_UTC\_2019.zip

$

Do not pay attention to the “Failed to upload collection to SR”. The SR is a fake one.

*Q1/ Which options would you use to upload the initialization parameter file to SR in MOS?*

$ **./tfactl upload -sr SR12345 -user TEST**

**$ORACLE\_HOME/dbs/initORCL.ora**

SR12345 is not a valid SR number.

$

##### A1/ The upload allows you to upload other files to your SR in MOS. The command above fails because the SR used is a fake one.

* 1. Start a Service Request Data Collection (SRDC) for the hanging situation after startup that happens in ORCL.
     1. Execute the /home/oracle/labs/DIAG/startup\_issue.sh shell script. Once the shell script is launched, a hanging situation appears after database instance startup.

$ **/home/oracle/labs/DIAG/startup\_issue.sh**

SQL> shutdown immediate Database closed.

Database dismounted. ORACLE instance shut down. SQL> exit

Disconnected from Oracle Database 19c Enterprise Edition Release

19.0.0.0.0 – Production Version 19.3.0.0.0

SQL\*Plus: Release 19.0.0.0.0 - Production on Tue Oct 16 15:48:17 2018 Version 19.3.0.0.0

Connected to an idle instance.

SQL> STARTUP

ORACLE instance started.

Total System Global Area 2382363480 bytes Fixed Size 9132888 bytes Variable Size 503316480 bytes

Database Buffers 1023410176 bytes Redo Buffers 7643136 bytes In-Memory Area 838860800 bytes Database mounted.

Database opened. SQL> set echo off

* + 1. Open another terminal window and log in as oracle. You are in *Session2*. Launch TFA to collect diagnostic data for the startup issue.

$ **. oraenv**

ORACLE\_SID = [oracle] ? **ORCL**

The Oracle base has been set to /u01/app/oracle

$ **cd $ORACLE\_BASE/tfa/bin**

$ **./tfactl collect -srdc dbstartup**

Enter the Database Name [Required for this SRDC] : **ORCL**

Instance startup was completed in the very last startup

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| [ | 1 | ]Wed | May | 15 | 09:35:32 | 2019 |
| [ | 2 | ]Wed | May | 15 | 09:41:59 | 2019 |
| [ | 3 | ]Wed | May | 15 | 09:50:56 | 2019 |
| [ | 4 | ]Wed | May | 15 | 09:52:51 | 2019 |
| [ | 5 | ]Wed | May | 15 | 10:17:20 | 2019 |
| [ | 6 | ]Wed | May | 15 | 10:21:43 | 2019 |
| [ | 7 | ]Wed | May | 15 | 15:50:54 | 2019 |
| [ | 8 | ]Wed | May | 15 | 15:54:07 | 2019 |
| [ | 9 | ]Wed | May | 15 | 15:56:32 | 2019 |
| [ | 10 ]Wed May 15 16:09:41 2019 | | | | | |

Please choose the problematic startup [1..10] [10] : **10**

Can you simulate the issue now so that TFA can automatically collect the diagnostic data ? [Y|y|N|n] [Y] : **Y**

Please execute the following commands:

#### strace -ftT -o

**/u01/app/oracle/tfa/repository/suptools/srdc/user\_oracle/1539700 389\_srdc\_dbstartup/start\_trace.out sqlplus /nolog**

#### SQL>conn / as sysdba;

SQL>**startup**

Wait 5-10 minutes then kill process and do shutdown immediate if necessary

Press <Enter> when complete

**Note:** Instead of using STARTUP command in the list of suggested commands, use the

startup.sql SQL script that creates the issue.

* + 1. In another terminal window, called *Session3*, execute the commands suggested by TFA.

$ **. oraenv**

ORACLE\_SID = [oracle] ? **ORCL**

The Oracle base has been set to /u01/app/oracle

$ **cd $ORACLE\_BASE/tfa/bin**

$ **strace -ftT -o**

**/u01/app/oracle/tfa/repository/suptools/srdc/user\_oracle/1539700 389\_srdc\_dbstartup/start\_trace.out sqlplus /nolog**

SQL\*Plus: Release 19.0.0.0.0 - Production on Tue Oct 16 14:34:39 2018 Version 19.3.0.0.0

SQL> **conn / as sysdba;**

Connected to an idle instance.

SQL> **startup**

ORA-01081: cannot start already-running ORACLE - shut it down first

SQL> **shutdown immediate**

Database closed. Database dismounted.

ORACLE instance shut down.

SQL> **@/home/oracle/labs/DIAG/startup.sql**

SQL> STARTUP

ORACLE instance started.

Total System Global Area 2382363480 bytes Fixed Size 9132888 bytes Variable Size 503316480 bytes

Database Buffers 1023410176 bytes Redo Buffers 7643136 bytes In-Memory Area 838860800 bytes Database mounted.

Database opened. SQL> set echo off

* + 1. In *Session2*, press **Enter**.

Please execute the following script in a sqlplus session. startup nomount

alter session set events '10046 trace name context forever, level 12';

alter database mount; alter database open; Wait for 10 minutes.

Press <Enter> when complete

* + 1. In another session, called *Session4*, execute the commands suggested by TFA.

$ **. oraenv**

ORACLE\_SID = [oracle] ? **ORCL**

The Oracle base has been set to /u01/app/oracle

$ **cd $ORACLE\_BASE/tfa/bin**

$ **strace -ftT -o**

**/u01/app/oracle/tfa/repository/suptools/srdc/user\_oracle/1539700 389\_srdc\_dbstartup/start\_trace.out sqlplus /nolog**

SQL\*Plus: Release 19.0.0.0.0 - Production on Tue Oct 16 14:34:39 2018 Version 19.3.0.0.0

SQL> **conn / as sysdba;** Connected to an idle instance. SQL> **startup nomount**

ORA-01081: cannot start already-running ORACLE - shut it down first

SQL> **shutdown immediate**

Database closed. Database dismounted.

ORACLE instance shut down. SQL> **startup nomount** ORACLE instance started.

Total System Global Area 2382363480 bytes Fixed Size 9132888 bytes Variable Size 503316480 bytes

Database Buffers 1023410176 bytes Redo Buffers 7643136 bytes In-Memory Area 838860800 bytes

SQL> **alter session set events '10046 trace name context forever, level 12';**

Session altered.

SQL> **@/home/oracle/labs/DIAG/startup\_mount.sql**

SQL> ALTER DATABASE MOUNT;

Database altered.

SQL> ALTER DATABASE OPEN;

Database altered.

* + 1. In *Session2*, press **Enter**.

**Running hanganalyze....**

Hang analysis in

/u01/app/oracle/diag/rdbms/orcl/ORCL/trace/ORCL\_ora\_16413.trc Running transaction recovery SQL script.......

Scripts to be run by this srdc: rdahcve1120 rdahcve1210 rdahcve1110 cp\_parameter

Components included in this srdc: DATABASE OS

Collecting data for local node(s)

Collection Id : 20190515162855*hostname*

Detailed Logging at :

/u01/app/oracle/tfa/repository/srdc\_dbstartup\_collection\_Wed\_May

\_15\_16\_28\_56\_UTC\_2019\_node\_local/diagcollect\_20190515162855\_*host name*.log

2019/05/15 16:29:00 UTC : NOTE : Any file or directory name containing the string .com will be renamed to replace .com with dotcom

2019/05/15 16:29:00 UTC : Collection Name : tfa\_srdc\_dbstartup\_Wed\_May\_15\_16\_28\_56\_UTC\_2019.zip

2019/05/15 16:29:00 UTC : Scanning of files for Collection in progress...

2019/05/15 16:29:00 UTC : Collecting additional diagnostic information...

2019/05/15 16:29:05 UTC : Getting list of files satisfying time range [05/15/2019 04:29:00 UTC, 05/15/2019 16:29:05 UTC]

|  |  |  |
| --- | --- | --- |
| 2019/05/15  diagnostic | 16:29:11 UTC :  information... | Completed collection of additional |
| 2019/05/15 | 16:29:20 UTC : | Collecting ADR incident files... |
| 2019/05/15 | 16:29:23 UTC : | Completed Local Collection |

. .

| Collection Summary |

+ + + + +

| Host | Status | Size | Time |

+ + + + +

| *hostname* | Completed | 19MB | 23s | ' + + + '

Logs are being collected to:

/u01/app/oracle/tfa/repository/srdc\_dbstartup\_collection\_Wed\_May

\_15\_16\_28\_56\_UTC\_2019\_node\_local

/u01/app/oracle/tfa/repository/srdc\_dbstartup\_collection\_Wed\_May

\_15\_16\_28\_56\_UTC\_2019\_node\_local/*hostname*.tfa\_srdc\_dbstartup\_Wed

\_May\_15\_16\_28\_56\_UTC\_2019.zip

$

*Q1/ Does the trace file generated hold information about the cause of the hang?*

##### A1/ The trace and diagnostic files help the Oracle Support analysts find the root cause of the issue.

|  |  |  |
| --- | --- | --- |
| Chain | 1 | Signature Hash: 0x456ca47b |
| [b] Chain | 2 | Signature: 'resmgr:cpu quantum' |
| Chain | 2 | Signature Hash: 0x8114dfec |
| [c] Chain | 3 | Signature: 'CPU or Wait CPU' |
| Chain | 3 | Signature Hash: 0xabe82914 |

You can send and upload the zip file automatically to your SR by using the same -sr option used in the previous task 3 step b. The Oracle Support analysts use the trace and diagnostic files to find the root cause of the hang and provide a solution.

================================================================

Chains most likely to have caused the hang:

[a] Chain 1 Signature: 'CPU or Wait CPU'<='**library cache load lock**'

================================================================

* 1. Kill and quit all the sessions.

$ **pgrep -lf sqlplus**

|  |  |  |  |
| --- | --- | --- | --- |
| 13807 | | sqlplus | |
| 14853 | | strace | |
| 14855 | | sqlplus | |
| 15759 | | strace | |
| 15761 | | sqlplus | |
| $ | **kill** | **-9** | **13807** |
| $ | **kill** | **-9** | **14853** |
| $ | **kill** | **-9** | **14855** |
| $ | **kill** | **-9** | **15759** |
| $ | **kill** | **-9** | **15761** |
| $ |  |  |  |

# Practices for Lesson 7: Using Sharding Enhancements

## Practices for Lesson 7

There are no practices for this lesson.