MAA / Data Guard 12c Setup Guide

Creating a RAC Physical Standby Database for a RAC Primary Database

By Nassyam Basha & YV RaviKumar

Introduction

Oracle Real application clusters (RAC) provides Business continuity, High availability, Scalability, Flexibility and Agility combined with ease management are the pillars of successful IT infrastructure and cloud deployments. Oracle RAC environments can also provide continuous service for both planned and unplanned outages as well as runtime and capacity-on-demand management, the Oracle RAC Stack ensures uninterrupted data center operations for applications of any kind. Customer applications benefits through Rolling upgrades for system and hardware changes, Rolling patch upgrades for some interim patches, Fast, automatic, and intelligent connection and service relocation and failover and Load balancing advisory and runtime connection load balancing.

Technical Overview

This technical paper focusses on creating a RAC 2-node physical standby database (stdrac) for a RAC 2-node primary database (cdbrac) with step by step procedure including involvement of Data Guard Broker and how to manage. This article entirely written on Oracle database 12c.
This technical paper assumes that there is an existing RAC primary database with two instances (cdbrac1 & cdbrac2) and you want to implement Data Guard by adding RAC physical standby database with two instances (stdrac1 & stdrac2).

Throughout this document, have used the below naming for database name, database unique name, Oracle net services, instances and the hostnames where they located in below table.

	DB_NAME	DB_UNIQUE_NAME	Oracle Net Service Name	Instances	Hostnames
Primary	CDBRAC	CDBRAC	CDBRAC	cdbrac1,cdbrac2	racnroll1, racnroll2
Standby	CDBRAC	STDRAC	STDRAC	stdrac1,stdrac2	racnroll3, racnroll4

The steps outlined in this technical document they assume using ASM, and that the software and ASM instance on the standby host have already been installed/created.

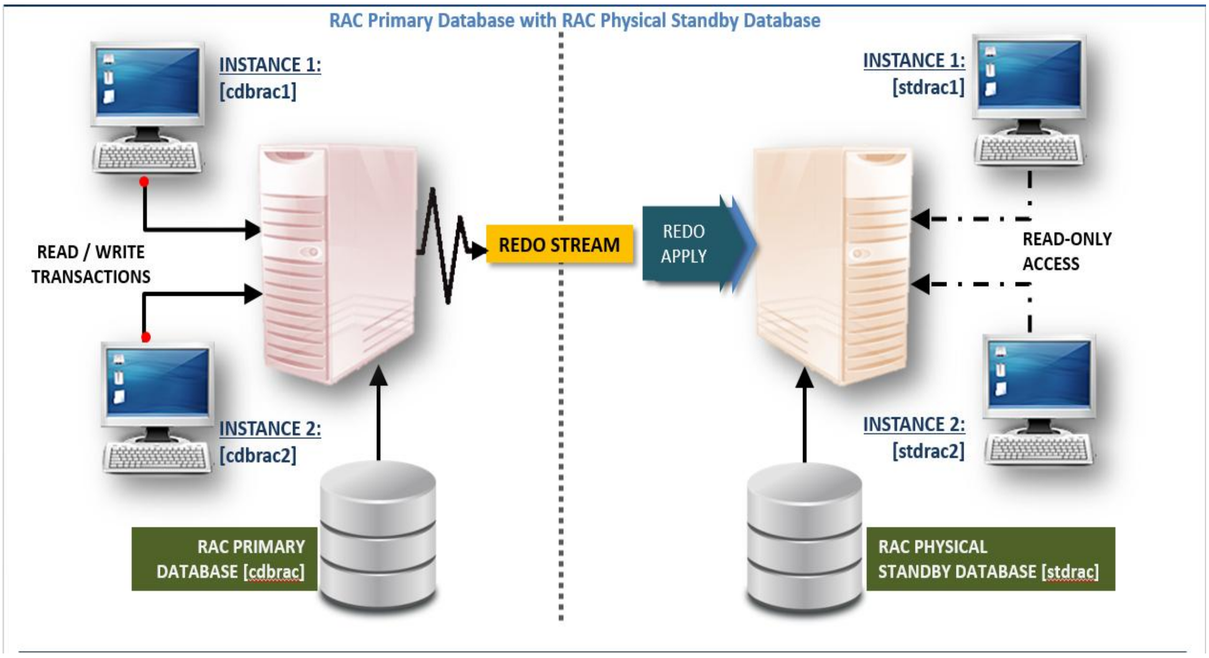

RAC Primary Database with RAC Physical Standby Database
INSTANCE 1: [cdbrac1]
READ / WRITE TRANSACTIONS
INSTANCE 2: [cdbrac2]
RAC PRIMARY DATABASE [cdbrac]
REDO STREAM
REDO APPLY
INSTANCE 1: [stdrac1]
READ-ONLY ACCESS
INSTANCE 2: [stdrac2]
RAC PHYSICAL STANDBY DATABASE [stdrac]

Figure-1: RAC Primary Database with RAC Physical Standby Database

This document includes the following steps:

- Changes of initialization parameters in primary and standby system
- Configure Oracle Net on the RAC primary database and RAC physical standby database
- Perform RMAN Duplicate method to restore database for standby use
- Copy the password file RAC primary database to RAC physical standby database
- Verify Data Guard configuration with creating object with rows.
- Performing switchover using Data Guard Broker.

This technical assumes the following pre-requisites are met for RAC primary database and RAC physical standby database

- Ensure the primary database is in Archivelog mode and Force logging enabled.

```
SQL> select log_mode,database_role,open_mode from v$database;

LOG_MODE       DATABASE_ROLE           OPEN_MODE
-----------------  ------------------------      --------------------
ARCHIVELOG     PRIMARY                 READ WRITE

SQL> select force_logging from v$database;

FORCE_LOGGING
-----------------------
YES
```

  Command to enable force logging if it's not configured: SQL> alter database force logging;

- The primary RAC database is located in ASM.
- The standby RAC database will be created on ASM.
- The primary and standby databases are using flash recovery area.

- The standby RAC nodes already installed with Grid Software and Oracle database software with the version 12.1.0.1.0.

1. Configure the RAC primary database initialization parameters to support primary role.

   ```
   SQL> alter system set log_archive_config='DG_CONFIG=(cdbrac,stdrac)' scope=both sid='*';

   SQL> alter system set log_archive_dest_1='LOCATION=USE_DB_RECOVERY_FILE_DEST
   VALID_FOR=(ALL_LOGFILES,ALL_ROLES)
   DB_UNIQUE_NAME=cdbrac' scope= both sid='*';

   SQL> alter system set log_archive_dest_2='SERVICE=stdrac
   VALID_FOR=(ONLINE_LOGFILES,PRIMARY_ROLE)
   DB_UNIQUE_NAME=stdrac' scope=both sid='*'

   SQL> alter system set LOG_ARCHIVE_FORMAT='%t_%s_%r.arc' scope=both sid='*'
   SQL> alter system set LOG_ARCHIVE_MAX_PROCESSES=8 scope=both sid='*'
   SQL> alter system set REMOTE_LOGIN_PASSWORDFILE=EXCLUSIVE scope=both sid='*'
   ```

   If you prefer to change the DB_UNIQUE_NAME then we must update values in spfile and then bounce of database required, because it's an static parameter.

   ```
   SQL> alter system set DB_UNIQUE_NAME=cdbrac scope=spfile sid='*'
   ```

   Note: Recommended to use SPFILE, if PFILE is in use the update/append the above parameters as required.

2. Configure the RAC primary database initialization parameters to support standby role.

   ```
   SQL> alter system set FAL_SERVER='stdrac1,stdrac2' scope=both sid='*'
   ```

   DB_FILE_NAME_CONVERT/LOG_FILE_NAME_CONVERT : If the primary and standby file system/Disk group are different then we must configure these two parameters.

   ```
   SQL> alter system set DB_FILE_NAME_CONVERT='+DATA','+DATADG' scope=spfile sid='*'
   SQL> alter system set LOG_FILE_NAME_CONVERT= '+FRA','+FRADG' scope=spfile sid='*'
   SQL> alter system set STANDBY_FILE_MANAGEMENT=AUTO scope=spfile sid='*'
   ```

3. On the RAC primary database, create staging directory. Create directory called 'backup' under /home/oracle and create initialization parameter file for RAC physical standby database.

   ```
   [oracle@racnroll1 ~]$ cd $HOME
   [oracle@racnroll1 ~]$ mkdir backup
   SQL> create pfile='/home/oracle/backup/initSTDRAC.ora' from spfile;
   ```

4. Using 'scp' command copy initSTDRAC.ora file from RAC primary database to one of the RAC physical standby database

   ```
   [oracle@racnroll1 backup]$ scp /home/oracle/backup/initSTDRAC.ora
   oracle@racnroll3:/u01/app/oracle/product/12.1.0.1/db_1/dbs/
   ```

5. We are performing Active duplicate of 11g feature, hence no need to take backup or copy to the standby server.

6. Create standby redo logs on the RAC primary database to support the standby role.
   The recommended number of standby redo logs is:
   (maximum # of logfiles +1) * maximum # of threads

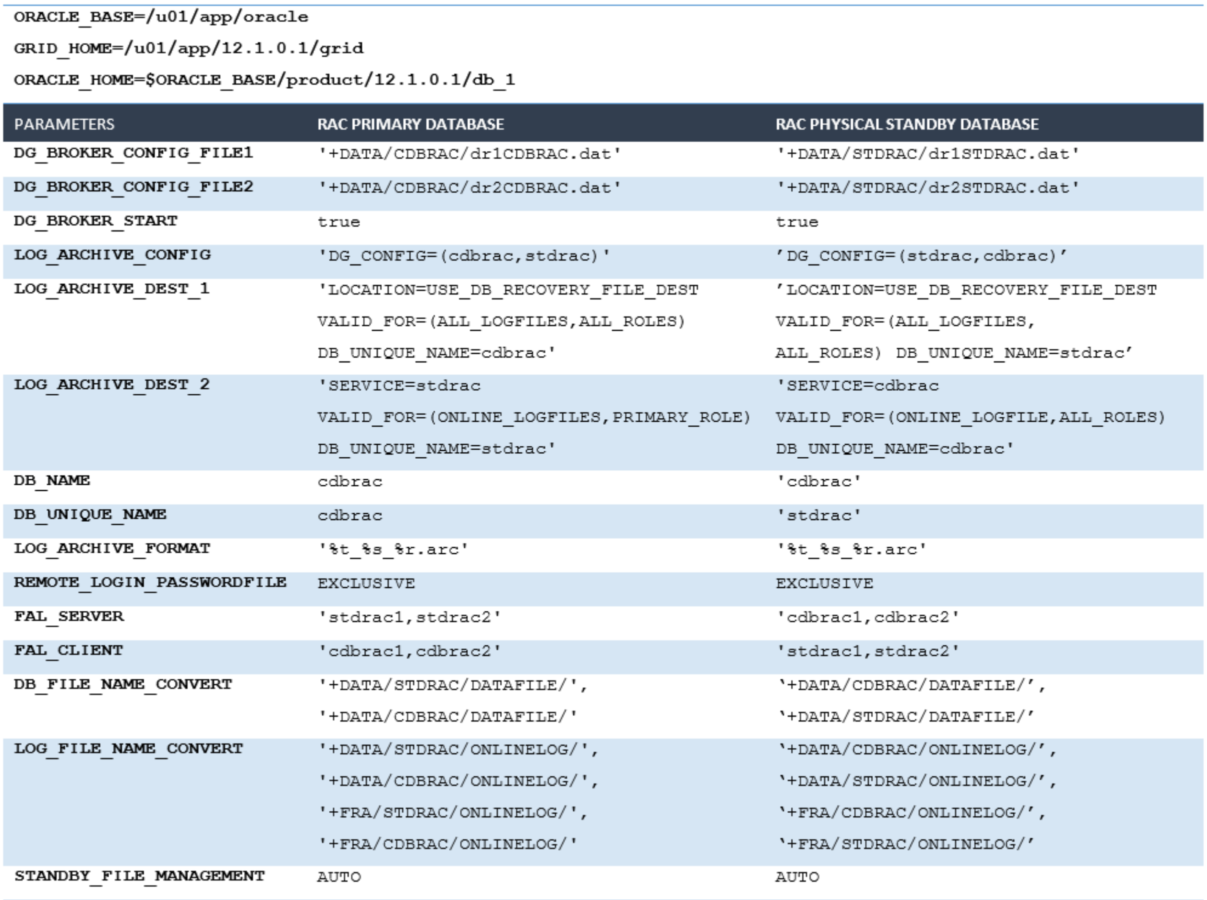
   The SRL files must be the same size as your online redo log (ORL) files, and you also need to have the same number of SRL files as you do ORL files, plus one. If you have a RAC primary, you need “plus one” per RAC instance. These files need to be created on your standby as well as on your primary in preparation for switchover.

   We must consider having single member in each group so that avoid waits with commit for each transaction in each member.

```
SQL> select group#,thread#,bytes from v$log;

   GROUP#       THREAD#        BYTES
   ------------ -------------  ----------
       1           1           52428800
       2           1           52428800
       3           2           52428800
       4           2           52428800

SQL> ALTER DATABASE ADD STANDBY LOGFILE THREAD 1 ('+FRA') SIZE 50M;
SQL> ALTER DATABASE ADD STANDBY LOGFILE THREAD 1 ('+FRA') SIZE 50M;
SQL> ALTER DATABASE ADD STANDBY LOGFILE THREAD 1 ('+FRA') SIZE 50M;
SQL> ALTER DATABASE ADD STANDBY LOGFILE THREAD 2 ('+FRA') SIZE 50M;
SQL> ALTER DATABASE ADD STANDBY LOGFILE THREAD 2 ('+FRA') SIZE 50M;
SQL> ALTER DATABASE ADD STANDBY LOGFILE THREAD 2 ('+FRA') SIZE 50M;

SQL> select group#,thread#,bytes from v$standby_log;

   GROUP#       THREAD#        BYTES
   ----------   -------------  ----------
       5           1           52428800
       6           1           52428800
       7           1           52428800
       8           2           52428800
       9           2           52428800
      10           2           52428800

6 rows selected.
```

   One more advantage is, We no need to create standby redo log files on standby and Oracle take cares of it during RMAN duplicate.

   1. Overview of Data Guard configuration and parameters.



```
ORACLE_BASE=/u01/app/oracle
GRID_HOME=/u01/app/12.1.0.1/grid
ORACLE_HOME=$ORACLE_BASE/product/12.1.0.1/db_1
```

PARAMETERS	RAC PRIMARY DATABASE	RAC PHYSICAL STANDBY DATABASE
DG_BROKER_CONFIG_FILE1	'+DATA/CDBRAC/dr1CDBRAC.dat'	'+DATA/STDRAC/dr1STDRAC.dat'
DG_BROKER_CONFIG_FILE2	'+DATA/CDBRAC/dr2CDBRAC.dat'	'+DATA/STDRAC/dr2STDRAC.dat'
DG_BROKER_START	true	true
LOG_ARCHIVE_CONFIG	'DG_CONFIG=(cdbrac,stdrac)'	’DG_CONFIG=(stdrac,cdbrac)’
LOG_ARCHIVE_DEST_1	'LOCATION=USE_DB_RECOVERY_FILE_DEST VALID_FOR=(ALL_LOGFILES,ALL_ROLES) DB_UNIQUE_NAME=cdbrac'	’LOCATION=USE_DB_RECOVERY_FILE_DEST VALID_FOR=(ALL_LOGFILES, ALL_ROLES) DB_UNIQUE_NAME=stdrac’
LOG_ARCHIVE_DEST_2	'SERVICE=stdrac VALID_FOR=(ONLINE_LOGFILES,PRIMARY_ROLE) DB_UNIQUE_NAME=stdrac'	'SERVICE=cdbrac VALID_FOR=(ONLINE_LOGFILE,ALL_ROLES) DB_UNIQUE_NAME=cdbrac'
DB_NAME	cdbrac	'cdbrac'
DB_UNIQUE_NAME	cdbrac	'stdrac'
LOG_ARCHIVE_FORMAT	'%t_%s_%r.arc'	'%t_%s_%r.arc'
REMOTE_LOGIN_PASSWORDFILE	EXCLUSIVE	EXCLUSIVE
FAL_SERVER	'stdrac1,stdrac2'	'cdbrac1,cdbrac2'
FAL_CLIENT	'cdbrac1,cdbrac2'	'stdrac1,stdrac2'
DB_FILE_NAME_CONVERT	'+DATA/STDRAC/DATAFILE/', '+DATA/CDBRAC/DATAFILE/'	‘+DATA/CDBRAC/DATAFILE/’, ‘+DATA/STDRAC/DATAFILE/’
LOG_FILE_NAME_CONVERT	'+DATA/STDRAC/ONLINELOG/', '+DATA/CDBRAC/ONLINELOG/', '+FRA/STDRAC/ONLINELOG/', '+FRA/CDBRAC/ONLINELOG/'	‘+DATA/CDBRAC/ONLINELOG/’, ‘+DATA/STDRAC/ONLINELOG/’, ‘+FRA/CDBRAC/ONLINELOG/’, ‘+FRA/STDRAC/ONLINELOG/’
STANDBY_FILE_MANAGEMENT	AUTO	AUTO

Figure-2: Noted parameters for RAC primary database and RAC physical standby database

Note: comment two initialization parameters in initSTDRAC.ora file (control_files & cluster_database parameters).

7. Copy the Password file from RAC primary database to RAC physical standby database using ‘pwcopy’ command from ASMCMD prompt.

```
[oracle@racnroll1 ~]$ . oraenv
ORACLE_SID = [cdbrac1] ? +ASM1
The Oracle base remains unchanged with value /u01/app/oracle

[oracle@racnroll1 ~]$ asmcmd
ASMCMD> pwget --dbuniquename cdbrac
+DATA/cdbrac/orapwcdbrac

ASMCMD> pwcopy --dbuniquename cdbrac '+DATA/cdbrac/orapwcdbrac'
'/home/oracle/backup'
copying +DATA/cdbrac/orapwcdbrac -> /home/oracle/backup/orapwcdbrac
ASMCMD> exit

[oracle@racnroll1 ~]$ cd /home/oracle/backup/
[oracle@racnroll1 backup]$ ls -lrth
[oracle@racnroll1 backup]$ scp orapwcdbrac oracle@racnroll3:/home/oracle/backup/
```

8. Login to one of the RAC physical standby database and copy the password file from local directory to ASM directory.

```
[oracle@racnroll3 ~]$ . oraenv
ORACLE_SID = [stdrac1] ? +ASM1
The Oracle base remains unchanged with value /u01/app/oracle
[oracle@racnroll3 ~]$ asmcmd

ASMCMD> pwcopy --dbuniquename stdrac '/home/oracle/backup/orapwcdbrac'
'+DATA/stdrac/orapwcdbrac'
copying /home/oracle/backup/orapwcdbrac -> +DATA/stdrac/orapwcdbrac
ASMCMD> exit
```

9. Configure Oracle net service/TNS names for standby system, so that in RMAN duplicate while connecting to the auxiliary instance we connect using the net service.

```
STDRAC1 =
(DESCRIPTION =
  (ADDRESS = (PROTOCOL = TCP)(HOST = racnroll3-vip)(PORT = 1521))
  (CONNECT_DATA =
    (SERVER = DEDICATED)
    (SERVICE_NAME = stdrac) (UR=A)
    (INSTANCE_NAME = stdrac1)
  )
)
```

We have mentioned (UR=A) with the SERVICE_NAME, The reason is during the RMAN duplicate internally it performs bounce of standby database and whenever instance is closed there will be no more services registered with the listener and hence by setting, we can escape from "ORA-12514: TNS:listener does not currently know of service requested in connect descriptor" and Oracle can perform bounce of standby system safely.

10. Login to one of the RAC physical standby database instance and start the instance in 'NOMOUNT' state after setting up the appropriate environment variables on RAC standby host, such as ORACLE_SID, ORACLE_BASE, and ORACLE_HOME, start the RAC standby database instance on the standby host in NOMOUNT status. Using RMAN we will perform Active Duplicate to restore the database.

```
[oracle@racnroll3 ~]$ export ORACLE_SID=stdrac1
[oracle@racnroll3 ~]$ export ORACLE_BASE=/u01/app/oracle
[oracle@racnroll3 ~]$ export ORACLE_HOME=/u01/app/oracle/product/12.1.0.1/db_1
[oracle@racnroll3 ~]$ sqlplus / as sysdba

SQL> startup nomount pfile='/u01/app/oracle/product/12.1.0.1/db_1/dbs/initSTDRAC.ora

[oracle@racnroll3 ~]$ rman target sys/oracle@cdbrac1 auxiliary sys/oracle@stdrac1

RMAN> duplicate target database for standby from active database nofilenamecheck;

Starting Duplicate Db at 27-JUL-15
using target database control file instead of recovery catalog
allocated channel: ORA_AUX_DISK_1
```

```
channel ORA_AUX_DISK_1: SID=28 device type=DISK
```

(Output is truncated……)

```
RMAN> alter system register;
using target database control file instead of recovery catalog
Statement processed
RMAN> exit
```

11. Shutdown the RAC physical standby database instance (stdrac1) and incorporate the changes for the initialization parameters control_files and cluster_database in initSTDRAC.ora file.

```
SQL> shutdown immediate;
[oracle@racnroll3 ~]$ vi /u01/app/oracle/product/12.1.0.1/db_1/dbs/initSTDRAC.ora
[oracle@racnroll3 ~]$ export ORACLE_SID=stdrac1
[oracle@racnroll3 ~]$ export ORACLE_BASE=/u01/app/oracle
[oracle@racnroll3 ~]$ export ORACLE_HOME=/u01/app/oracle/product/12.1.0.1/db_1
[oracle@racnroll3 ~]$ sqlplus / as sysdba
SQL> select open_mode, database_role from v$database;

OPEN_MODE            DATABASE_ROLE
--------------------         --------------------------
MOUNTED              PHYSICAL STANDBY

SQL> show parameter control_files
SQL> show parameter cluster_database
SQL> exit
```

12. On either node of the standby cluster, register the standby database and the database instances with the Oracle Cluster Registry (OCR) using the Server Control.

```
[oracle@racnroll3 ~]$ srvctl add database -d stdrac -n cdbrac -o
/u01/app/oracle/product/12.1.0.1/db_1 -p +DATA/STDRAC/parameterfile/spfileSTDRAC.ora
-r physical_standby -a DATA,FRA
[oracle@racnroll3 ~]$ srvctl add instance -d stdrac -i stdrac1 -n racnroll3
[oracle@racnroll3 ~]$ srvctl add instance -d stdrac -i stdrac2 -n racnroll4
[oracle@racnroll3 ~]$ srvctl start database -d stdrac -o mount
[oracle@racnroll3 ~]$ ps -ef | grep pmon
[oracle@racnroll3 ~]$ srvctl config database -d stdrac
```

The following are descriptions of the options in these SRVCTL commands:

- -d option specifies the database unique name (DB_UNIQUE_NAME) of the database.
- -i option specifies the database instance name.
- -n option specifies the node on which the instance is running.
- -o option specifies the Oracle home of the database.

13. Start Recovery on RAC Physical Standby Database
After successful RMAN duplicate and the configuration related to cluster, We can start the MRP (Media Recovery Process) on any one node of the RAC physical standby database.

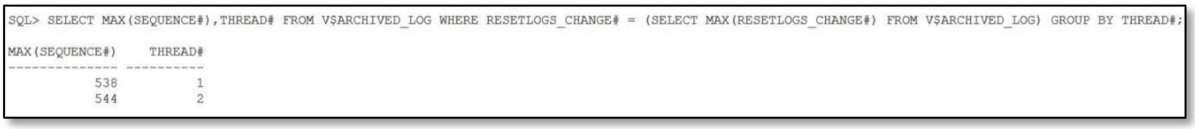
```
SQL> alter database recover managed standby database disconnect from session;
```

14. Monitoring REDO transport status
    The next part is to monitor the redo transport and checking apply lag or transport lag. After the configuration still there is scope to have issues where standby may not fetch redo. So we need to ensure the standby is catching up the redo data. For this we can use various views in order to estimate the lag.

    The very simple and easy commands to know the brief status we can use below commands
    Primary: SQL> select thread#,max(sequence#) from v$archived_log group by thread#;
    Standby: SQL> select thread#,max(sequence#) from v$archived_log where applied='YES' group by thread#;

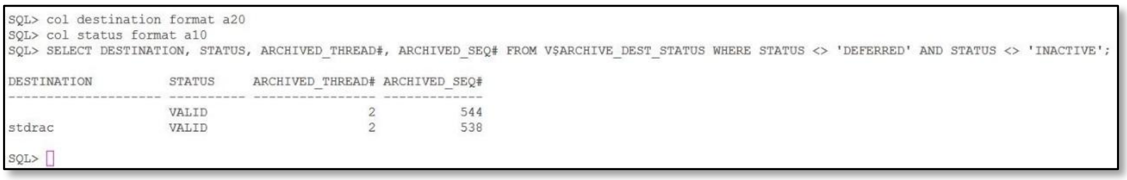
    As we said there are many views available, few are below.

```
SQL> SELECT MAX(SEQUENCE#), THREAD#
        FROM V$ARCHIVED_LOG
        WHERE RESETLOGS_CHANGE# = (SELECT MAX(RESETLOGS_CHANGE#)
                                                        FROM V$ARCHIVED_LOG)
                                                        GROUP BY THREAD#;
```



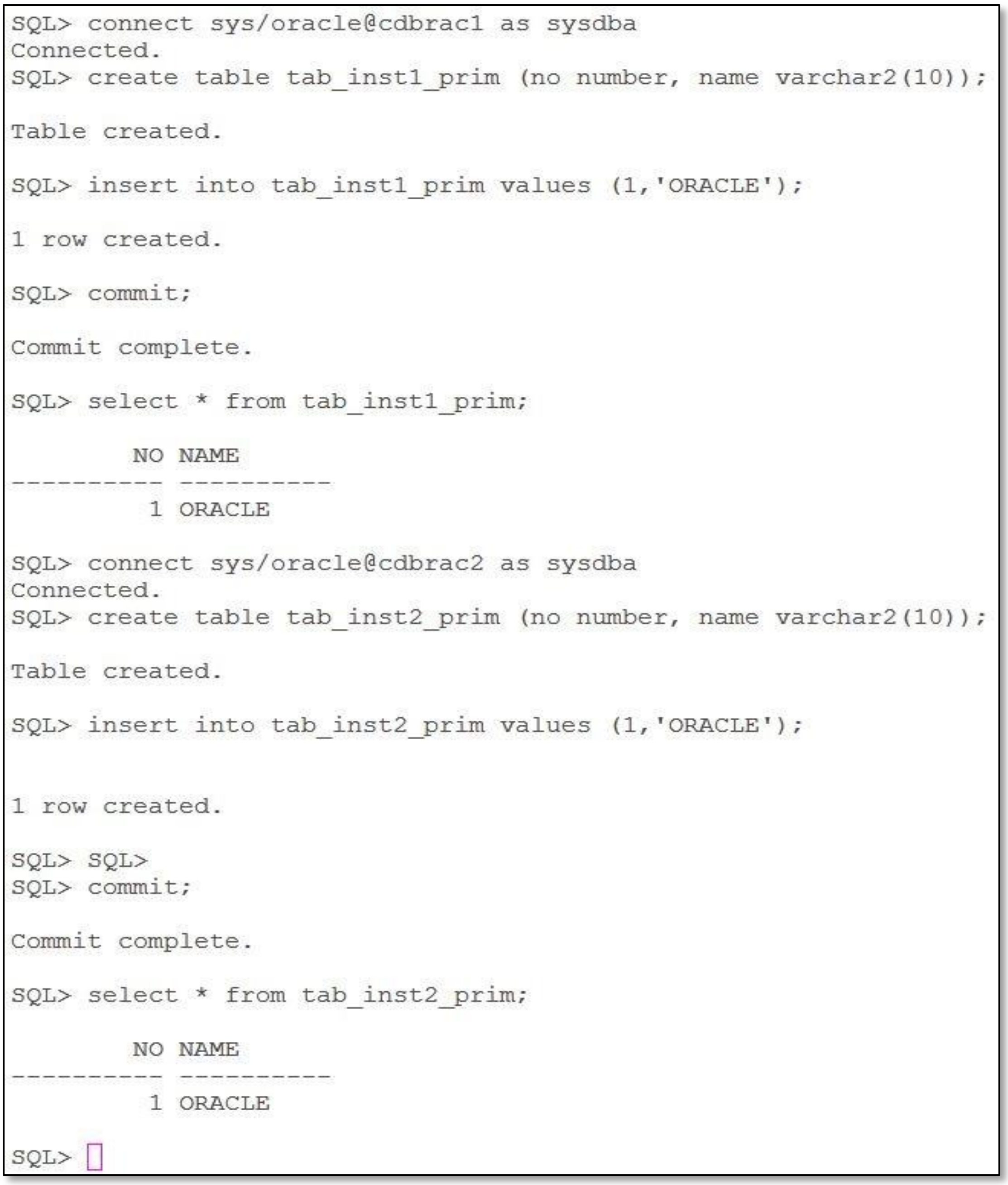
```
SQL> SELECT MAX(SEQUENCE#),THREAD# FROM V$ARCHIVED_LOG WHERE RESETLOGS_CHANGE# = (SELECT MAX(RESETLOGS_CHANGE#) FROM V$ARCHIVED_LOG) GROUP BY THREAD#;

MAX(SEQUENCE#)    THREAD#
-------------- ----------
           538          1
           544          2
```

```
SQL> col destination format a20
SQL> col status format a10
SQL> SELECT DESTINATION, STATUS, ARCHIVED_THREAD#, ARCHIVED_SEQ#
        FROM V$ARCHIVE_DEST_STATUS
        WHERE STATUS <> 'DEFERRED' AND STATUS <> 'INACTIVE';
```



```
SQL> col destination format a20
SQL> col status format a10
SQL> SELECT DESTINATION, STATUS, ARCHIVED_THREAD#, ARCHIVED_SEQ# FROM V$ARCHIVE_DEST_STATUS WHERE STATUS <> 'DEFERRED' AND STATUS <> 'INACTIVE';

DESTINATION          STATUS     ARCHIVED_THREAD# ARCHIVED_SEQ#
-------------------- ---------- ---------------- -------------
                     VALID                     2           544
stdrac               VALID                     2           538

SQL> 
```

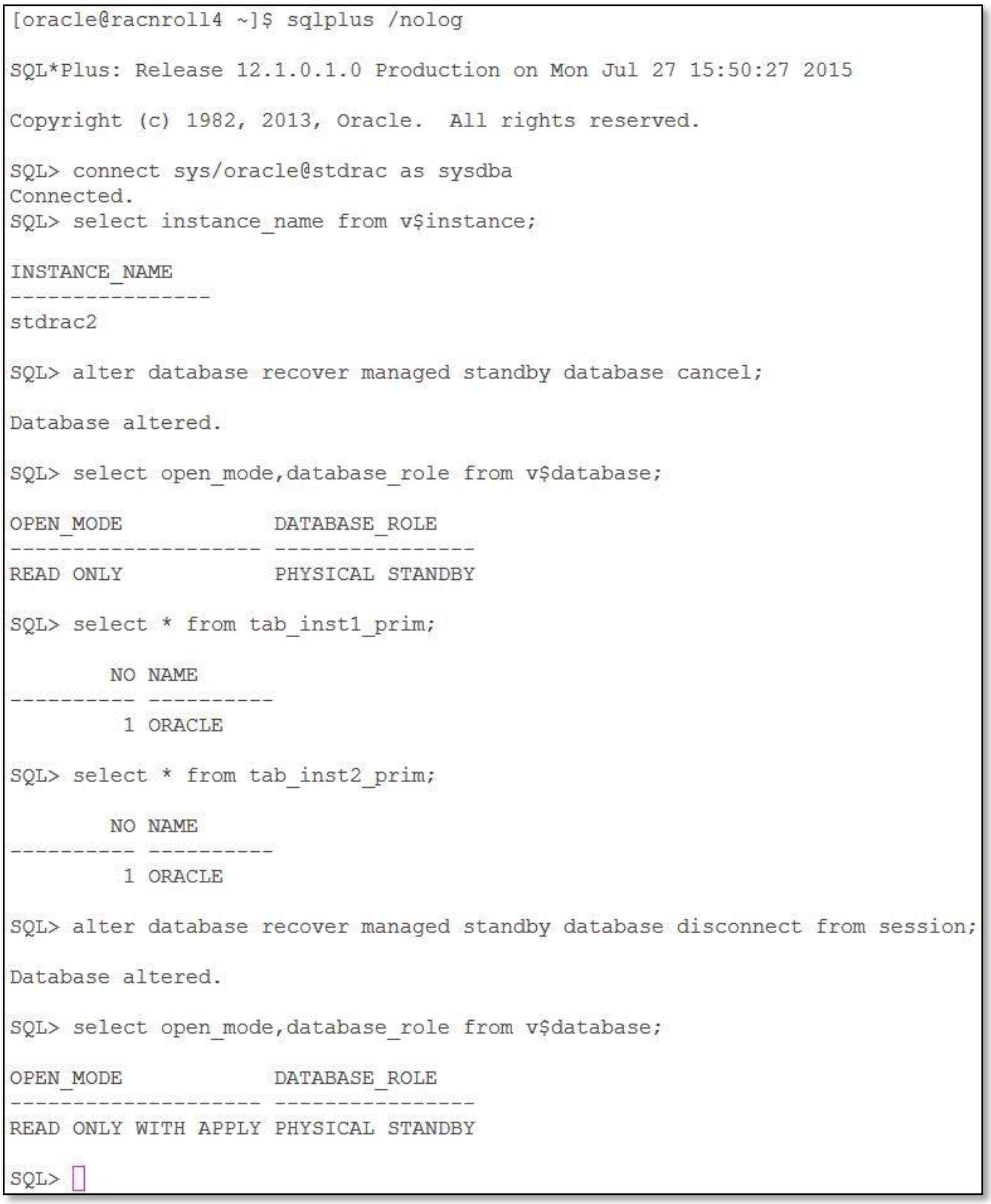
15. Testing the Data Guard 12c configuration between RAC primary database and RAC physical standby database. Create objects from RAC primary database instances and check those objects in RAC physical standby database.



```
SQL> connect sys/oracle@cdbrac1 as sysdba
Connected.
SQL> create table tab_inst1_prim (no number, name varchar2(10));

Table created.

SQL> insert into tab_inst1_prim values (1,'ORACLE');

1 row created.

SQL> commit;

Commit complete.

SQL> select * from tab_inst1_prim;

        NO NAME
---------- ----------
         1 ORACLE

SQL> connect sys/oracle@cdbrac2 as sysdba
Connected.
SQL> create table tab_inst2_prim (no number, name varchar2(10));

Table created.

SQL> insert into tab_inst2_prim values (1,'ORACLE');


1 row created.

SQL> SQL>
SQL> commit;

Commit complete.

SQL> select * from tab_inst2_prim;

        NO NAME
---------- ----------
         1 ORACLE

SQL> 
```

Figure-3: Inserted two tables in RAC primary database from stdrac1 instance through cdbrac1 and cdbrac2 instances.

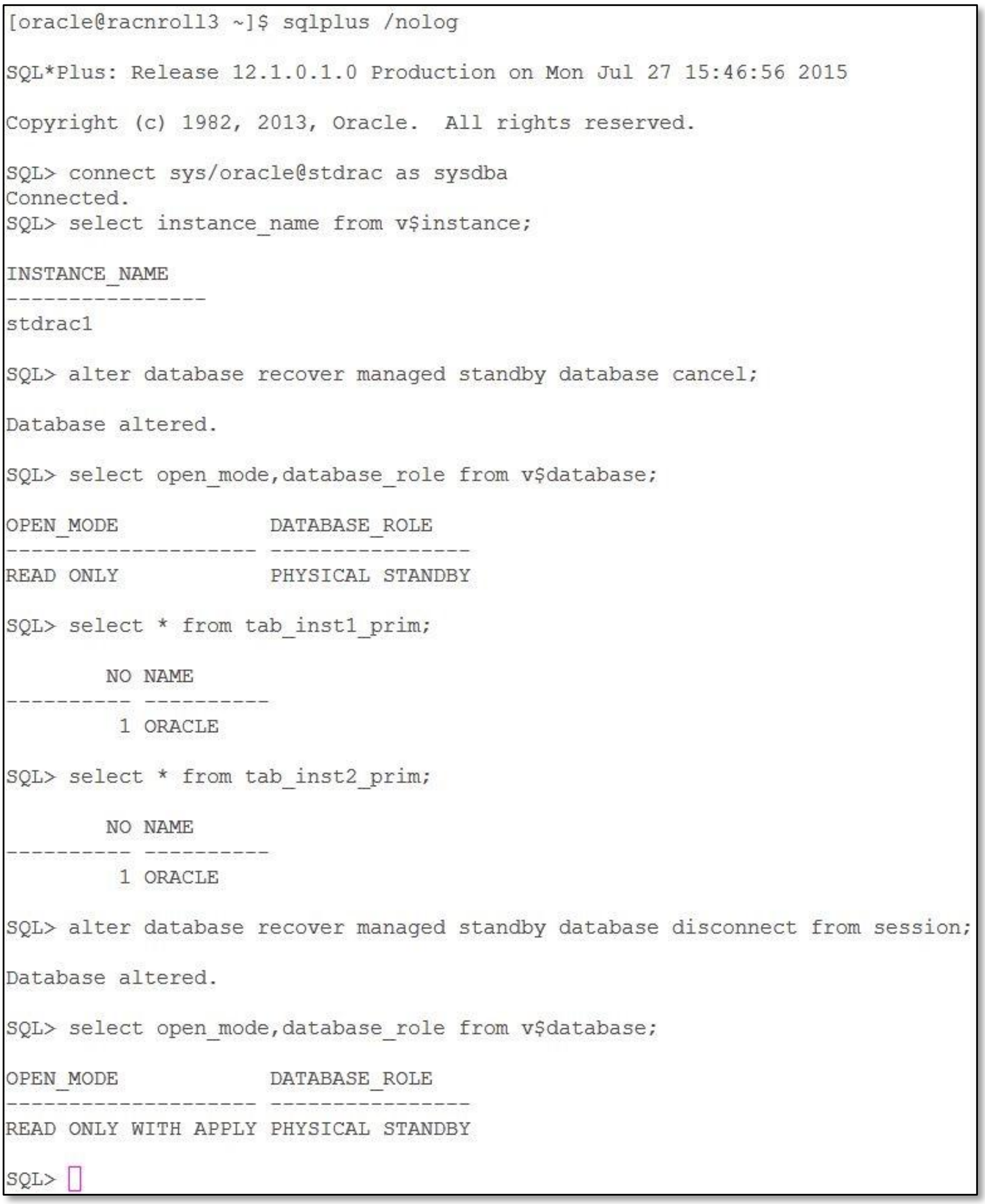
Checking the object with rows from RAC physical standby database’s first instance



```
[oracle@racnroll4 ~]$ sqlplus /nolog

SQL*Plus: Release 12.1.0.1.0 Production on Mon Jul 27 15:50:27 2015

Copyright (c) 1982, 2013, Oracle.  All rights reserved.

SQL> connect sys/oracle@stdrac as sysdba
Connected.
SQL> select instance_name from v$instance;

INSTANCE_NAME
----------------
stdrac2

SQL> alter database recover managed standby database cancel;

Database altered.

SQL> select open_mode,database_role from v$database;

OPEN_MODE            DATABASE_ROLE
-------------------- ----------------
READ ONLY            PHYSICAL STANDBY

SQL> select * from tab_inst1_prim;

        NO NAME
---------- ----------
         1 ORACLE

SQL> select * from tab_inst2_prim;

        NO NAME
---------- ----------
         1 ORACLE

SQL> alter database recover managed standby database disconnect from session;

Database altered.

SQL> select open_mode,database_role from v$database;

OPEN_MODE            DATABASE_ROLE
-------------------- ----------------
READ ONLY WITH APPLY PHYSICAL STANDBY

SQL> 
```

Figure-4: Checked tables in RAC physical standby database inserted from RAC primary database.

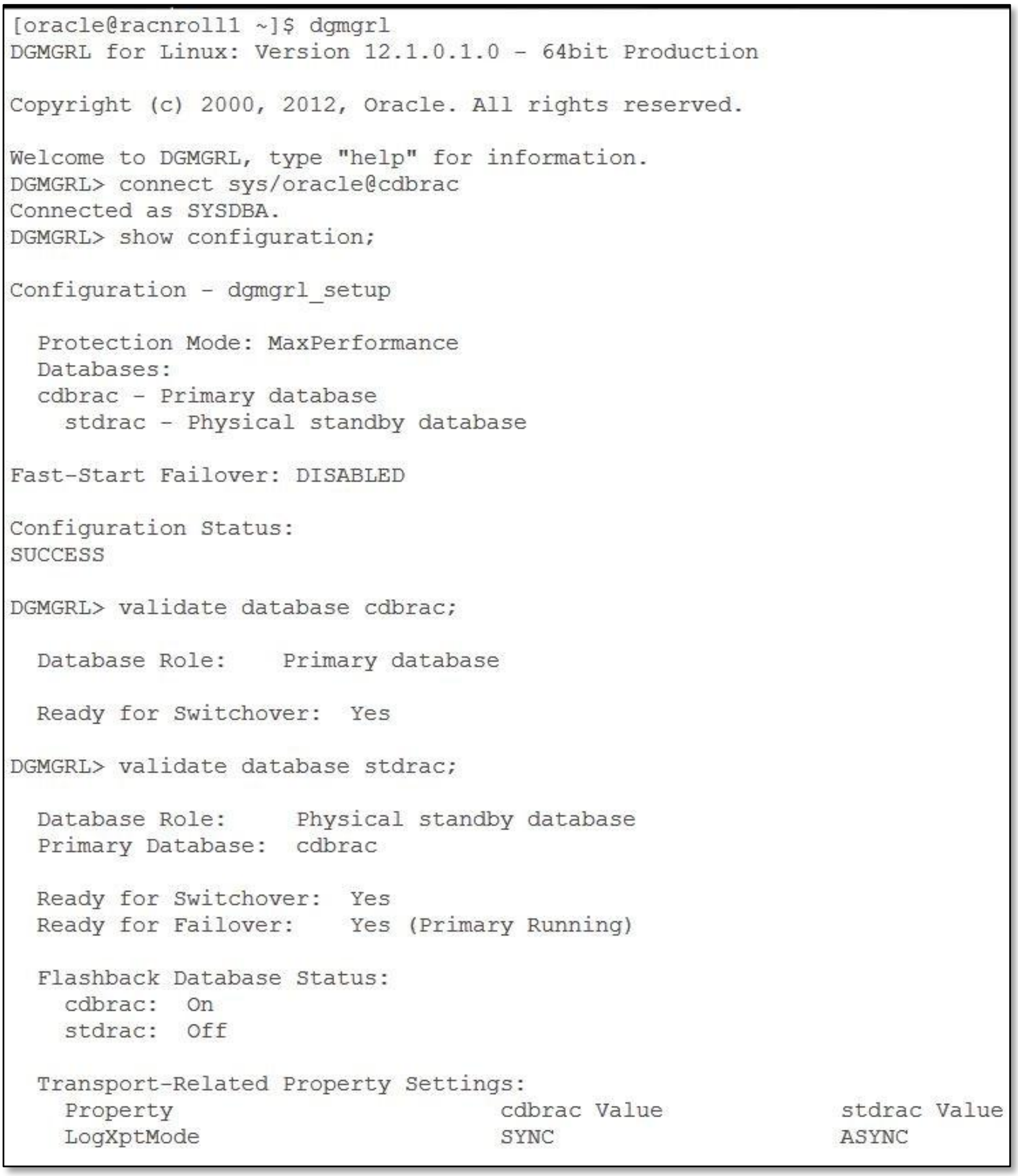
Checking the object with rows from RAC physical standby database's second instance



```
[oracle@racnroll3 ~]$ sqlplus /nolog

SQL*Plus: Release 12.1.0.1.0 Production on Mon Jul 27 15:46:56 2015

Copyright (c) 1982, 2013, Oracle.  All rights reserved.

SQL> connect sys/oracle@stdrac as sysdba
Connected.
SQL> select instance_name from v$instance;

INSTANCE_NAME
----------------
stdrac1

SQL> alter database recover managed standby database cancel;

Database altered.

SQL> select open_mode,database_role from v$database;

OPEN_MODE            DATABASE_ROLE
-------------------- ----------------
READ ONLY            PHYSICAL STANDBY

SQL> select * from tab_inst1_prim;

        NO NAME
---------- ----------
         1 ORACLE

SQL> select * from tab_inst2_prim;

        NO NAME
---------- ----------
         1 ORACLE

SQL> alter database recover managed standby database disconnect from session;

Database altered.

SQL> select open_mode,database_role from v$database;

OPEN_MODE            DATABASE_ROLE
-------------------- ----------------
READ ONLY WITH APPLY PHYSICAL STANDBY

SQL> 
```

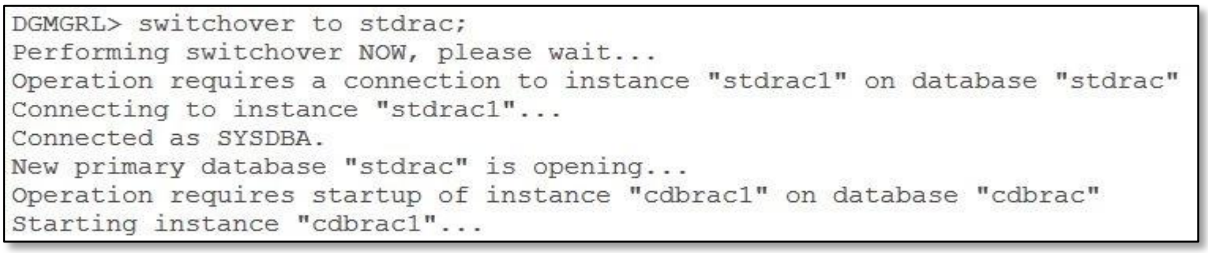
Figure-5: Checked tables in RAC physical standby database from stdrac2 instance inserted from RAC primary database.

16. Performing switchover activity from RAC primary database to RAC physical standby database using DGMGRL prompt. Login to RAC primary database and check the validity of the CDBRAC and STDRAC instances for switchover activity.



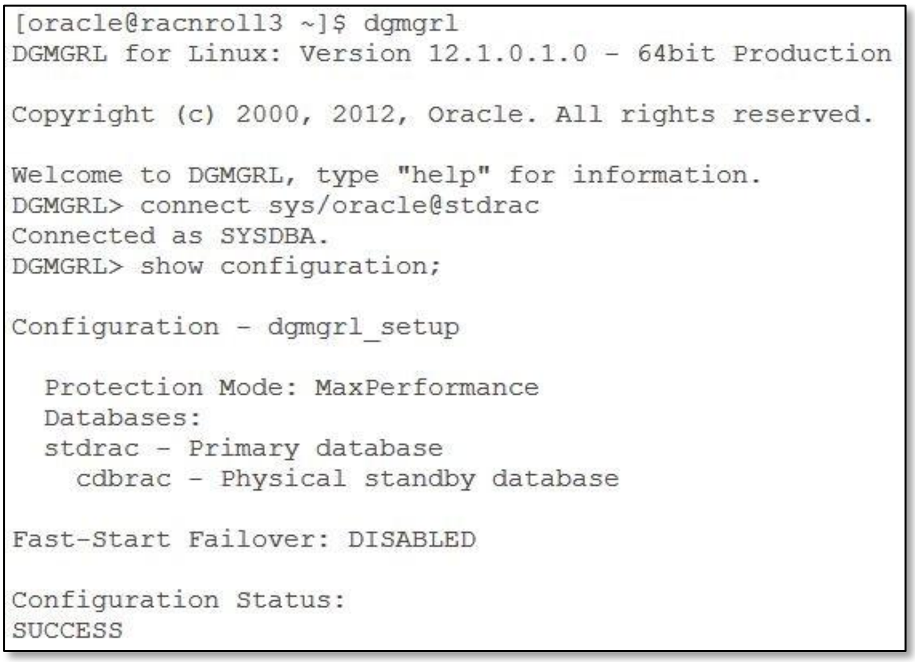
```
[oracle@racnroll1 ~]$ dgmgrl
DGMGRL for Linux: Version 12.1.0.1.0 - 64bit Production

Copyright (c) 2000, 2012, Oracle. All rights reserved.

Welcome to DGMGRL, type "help" for information.
DGMGRL> connect sys/oracle@cdbrac
Connected as SYSDBA.
DGMGRL> show configuration;

Configuration - dgmgrl_setup

  Protection Mode: MaxPerformance
  Databases:
  cdbrac - Primary database
    stdrac - Physical standby database

Fast-Start Failover: DISABLED

Configuration Status:
SUCCESS

DGMGRL> validate database cdbrac;

  Database Role:    Primary database

  Ready for Switchover:  Yes

DGMGRL> validate database stdrac;

  Database Role:     Physical standby database
  Primary Database:  cdbrac

  Ready for Switchover:  Yes
  Ready for Failover:    Yes (Primary Running)

  Flashback Database Status:
    cdbrac:  On
    stdrac:  Off

  Transport-Related Property Settings:
    Property                        cdbrac Value             stdrac Value
    LogXptMode                      SYNC                     ASYNC
```

17. Performing switchover from CDBRAC to STDRAC



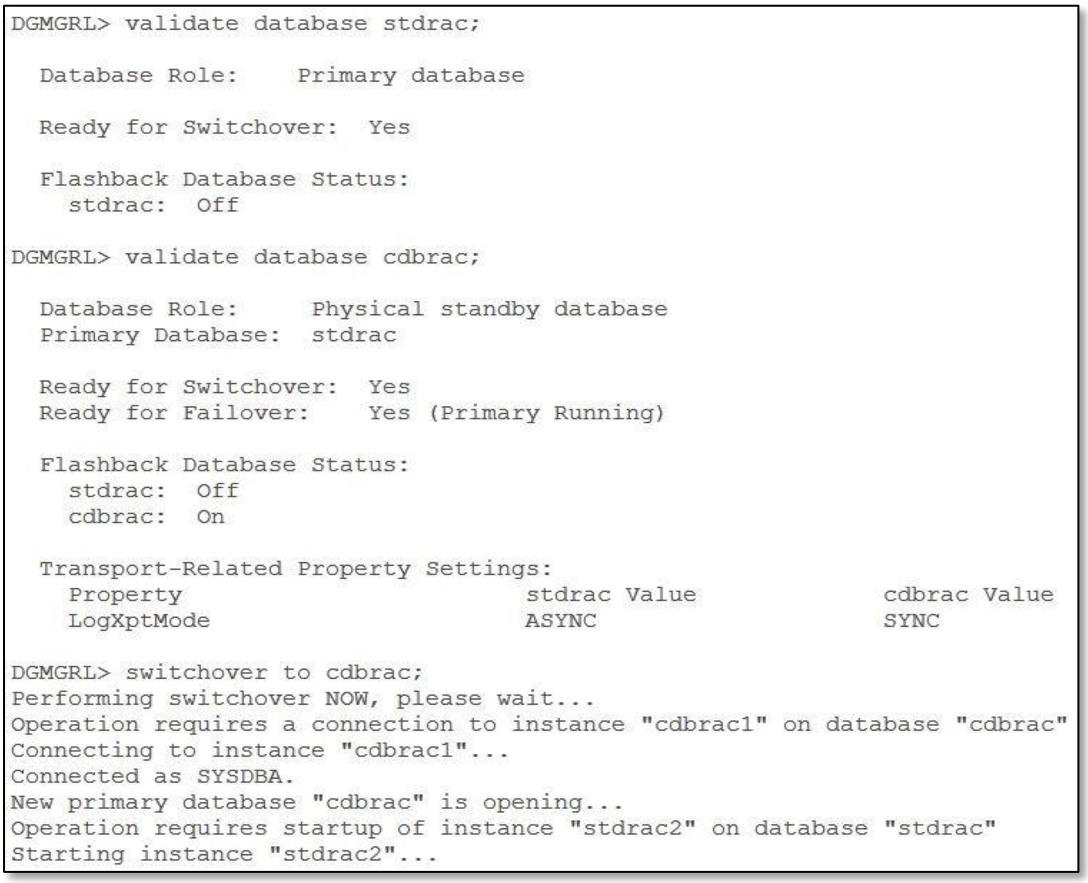
```
DGMGRL> switchover to stdrac;
Performing switchover NOW, please wait...
Operation requires a connection to instance "stdrac1" on database "stdrac"
Connecting to instance "stdrac1"...
Connected as SYSDBA.
New primary database "stdrac" is opening...
Operation requires startup of instance "cdbrac1" on database "cdbrac"
Starting instance "cdbrac1"...
```

18. Login to new RAC primary database (STDRAC) and check the configuration status from DGMGRL prompt.



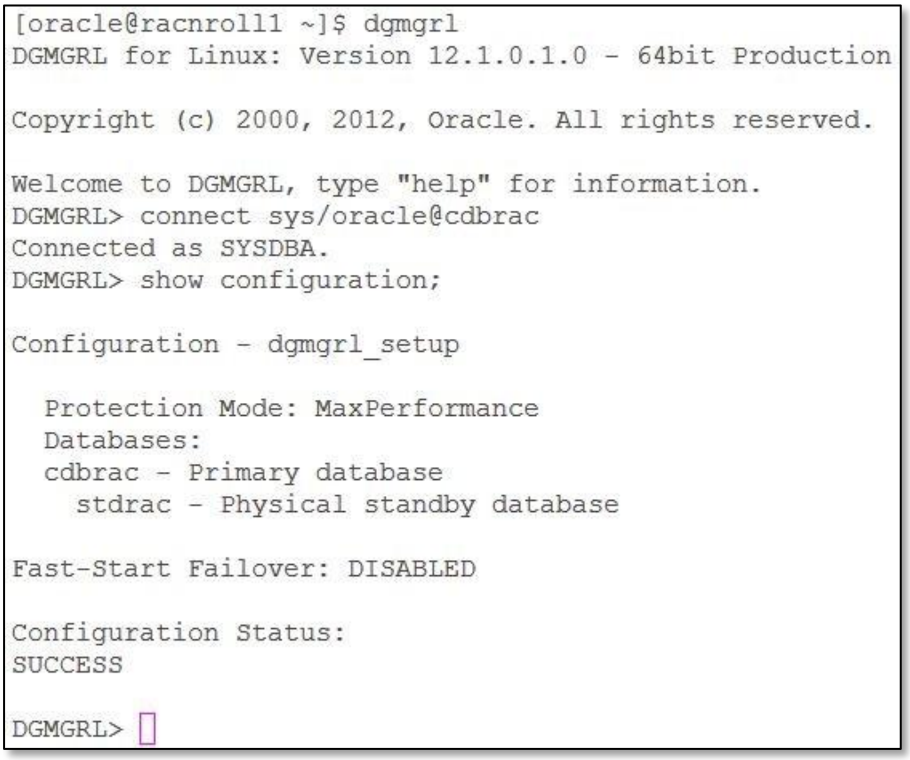
```
[oracle@racnroll3 ~]$ dgmgrl
DGMGRL for Linux: Version 12.1.0.1.0 - 64bit Production

Copyright (c) 2000, 2012, Oracle. All rights reserved.

Welcome to DGMGRL, type "help" for information.
DGMGRL> connect sys/oracle@stdrac
Connected as SYSDBA.
DGMGRL> show configuration;

Configuration - dgmgrl_setup

  Protection Mode: MaxPerformance
  Databases:
  stdrac - Primary database
    cdbrac - Physical standby database

Fast-Start Failover: DISABLED

Configuration Status:
SUCCESS
```

19. Performing switchover activity from new RAC primary database (STDRAC), before that check the validity of the STDRAC and CDBRAC instances for switchover activity.



```
DGMGRL> validate database stdrac;

  Database Role:    Primary database

  Ready for Switchover:  Yes

  Flashback Database Status:
    stdrac:  Off

DGMGRL> validate database cdbrac;

  Database Role:     Physical standby database
  Primary Database:  stdrac

  Ready for Switchover:  Yes
  Ready for Failover:    Yes (Primary Running)

  Flashback Database Status:
    stdrac:  Off
    cdbrac:  On

  Transport-Related Property Settings:
    Property                        stdrac Value             cdbrac Value
    LogXptMode                      ASYNC                    SYNC

DGMGRL> switchover to cdbrac;
Performing switchover NOW, please wait...
Operation requires a connection to instance "cdbrac1" on database "cdbrac"
Connecting to instance "cdbrac1"...
Connected as SYSDBA.
New primary database "cdbrac" is opening...
Operation requires startup of instance "stdrac2" on database "stdrac"
Starting instance "stdrac2"...
```

20. Login to RAC primary database (CDBRAC) and check the configuration status from DGMGRL prompt.



```
[oracle@racnroll1 ~]$ dgmgrl
DGMGRL for Linux: Version 12.1.0.1.0 - 64bit Production

Copyright (c) 2000, 2012, Oracle. All rights reserved.

Welcome to DGMGRL, type "help" for information.
DGMGRL> connect sys/oracle@cdbrac
Connected as SYSDBA.
DGMGRL> show configuration;

Configuration - dgmgrl_setup

  Protection Mode: MaxPerformance
  Databases:
  cdbrac - Primary database
    stdrac - Physical standby database

Fast-Start Failover: DISABLED

Configuration Status:
SUCCESS

DGMGRL> 
```

Summary

Implemented Maximum Availability Architecture (MAA) configuration for RAC primary database with RAC physical standby database using Oracle Data Guard and tested the configuration with objects. The goal of Maximum Availability Architecture (MAA) is to achieve optimal high availability for Oracle customers at the lowest cost and complexity. We have used in this document HA features like Real Application Clusters (RAC), Automatic Storage Management (ASM), Recovery Manager (RMAN) and Oracle Data Guard.

About the Authors

Nassyam Basha Oracle DBA, OCM 11g, Oracle ACE Director, Author of Data Guard 11gR2 Book, Blogger, OTN Super Hero, MOSC Guru, Writer with OTN, DELL and having around 9 years of hands on experience in High Availability technologies Like Oracle RAC, Data Guard, Exadata and much more. Currently working with The Pythian Group, In the past I have worked for AT&T(Bell Labs), dbaDirect, SLK software services.

YV RaviKumar is an Oracle ACE and Oracle Certified Master with 16 years of experience in Banking, Financial Services and Insurance and played roles like Senior Database Architect and Production DBA. Written 30+ OTN articles, 10+ articles in TOAD World, All things ORACLE from redgate & OTech Magazine.

Twitter: @yvrk1973

Blog: http://yvrk1973.blogspot.in