CONVERT PHYSCIAL STANDBY TO SNAPSHOT STANDBY

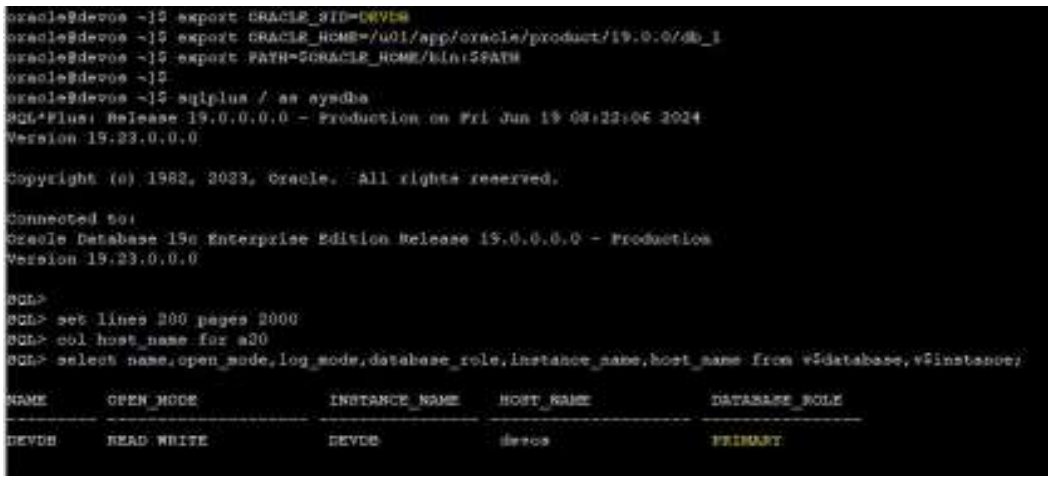
- Primary database changes will not applied to snapshot standby database why because there is no MRP process running on snapshot database.

- Whatever changes done on snapshot standby will be flushed out once convert back to physical standby database from snapshot standby

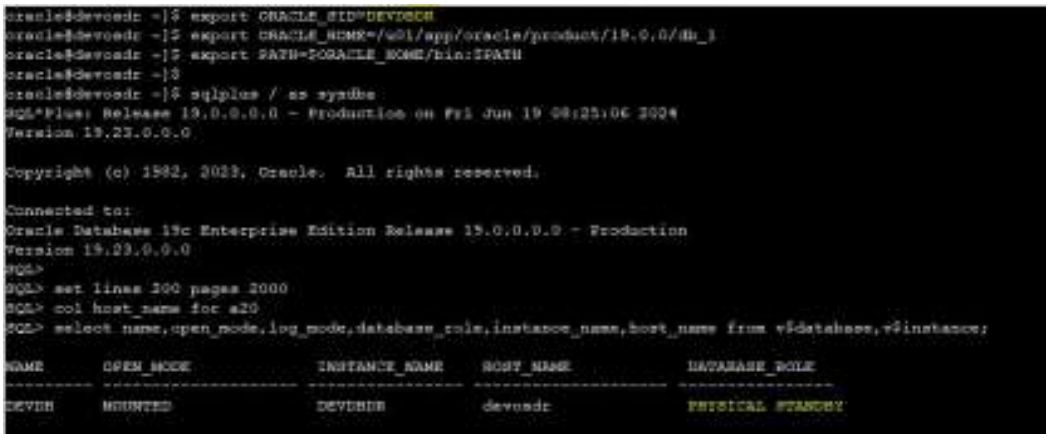
STEP:1 CHECK BOTH PRIMARY AND STANDBY ROLE.

```
set lines 200 pages 2000
col host_name for a20
select name,open_mode,log_mode,database_role,instance_name,host_name
fromv$database,v$instance;
```

PRIMARY:-



STANDBY:-



STEP 2:- CHECK THE ARCHIVED SEQUENCE IN PRIMARY AND STANDBY select

thread#,max(sequence#) from v$archived_log group by thread#; **PRIMARY ::**

```
SQL> select thread#,max(sequence#) from v$archived_log group by thread#;

   THREAD#  MAX(SEQUENCE#)
----------  --------------
         1              26
```

STANDBY ::

```
SQL> select thread#,max(sequence#) from v$archived_log group by thread#;

   THREAD#  MAX(SEQUENCE#)
----------  --------------
         1              26
```

STEP 3:- CHECK THE FLASHBACK STATUS AND DB_RECOVERY_FILE_SET LOCATION

select flashback_on from v$database;
show parameter db_recovery_file_dest;

```
SQL> select flashback_on from v$database;

FLASHBACK_ON
------------
YES

SQL> show parameter db_recovery_file_dest

NAME                                 TYPE        VALUE
------------------------------------ ----------- ------------------------------
db_recovery_file_dest                string      /u01/app/oracle/fra/
db_recovery_file_dest_size           big integer 8016M
```

IF FLASHBACK IS OFF THEN ENABLE IT BY BELOW COMMMAD :: alter database flashback on;

IF RECOVERY AREA NOT SET THEN SET BY USING BELOW ::

alter system set db_recovery_file_dest_size=10g;
alter system set db_recovery_file_dest='/u01/app/oracle/fra';

STEP 4: IN STANDBY SIDE,STOP THE MRP PROCESS.

alter database recover managed standby database cancel;

```
SQL> alter database recover managed standby database cancel;

Database altered.
```

STEP:5 BOUNCE THE DATABASE AND KEEP IN MOUNT STAGE

shut immediate
start mount

```
oracle@devosdr ~]$
oracle@devosdr ~]$ sqlplus / as sysdba
SQL*Plus: Release 19.0.0.0.0 - Production on Fri Jun 19 08:35:06 2024
Version 19.23.0.0.0

Copyright (c) 1982, 2023, Oracle.  All rights reserved.

Connected to:
Oracle Database 19c Enterprise Edition Release 19.0.0.0.0 - Production
Version 19.23.0.0.0
SQL> shut immediate
ORA-01109: database not open
Database dismounted.
ORACLE instance shut down.
SQL> exit
```

```
SQL> exit

Disconnected from Oracle Database 19c Enterprise Edition Release 19.0.0.0.0 - Production
Version 19.23.0.0.0
oracle@devosdr ~]$
oracle@devosdr ~]$ sqlplus / as sysdba

SQL*Plus: Release 19.0.0.0.0 - Production on Fri Jun 19 08:35:06 2024
Version 19.23.0.0.0

Copyright (c) 1982, 2023, Oracle.  All rights reserved.

Connected to:
Oracle Database 19c Enterprise Edition Release 19.0.0.0.0 - Production
Version 19.23.0.0.0

SQL> startup mount
ORACLE instance started.
Total System Global Area  838860800 bytes
Fixed Size                 8626240 bytes
Variable Size            645926848 bytes
Database Buffers         180355072 bytes
Redo Buffers               3952640 bytes
Database mounted.
```

set lines 200 pages 2000
col host_name for a20
select name,open_mode,log_mode,database_role,instance_name,host_name
fromv$database,v$instance;

```
SQL> set lines 200 pages 2000
SQL> col host_name for a20
SQL> select name,open_mode,log_mode,database_role,instance_name,host_name from v$database,v$instance;

NAME      OPEN_MODE            INSTANCE_NAME    HOST_NAME            DATABASE_ROLE
--------- -------------------- ---------------- -------------------- ----------------
DEVDB     MOUNTED              DEVDBDR          devosdr              PHYSICAL STANDBY
```

STEP:5 CONVERT TO SNAPSHOT STANDBY DATABASE alter database convert to snapshot standby;

```
SQL> alter database convert to snapshot standby;

Database altered.
```

STEP:6 OPEN THE STANDBY DATABASE IN READ/WRITE MODE alter database open;

```
SQL> alter database open;

Database altered.
```

STEP:7 CHECK THE DATABASE_ROLE AND OPEN_MODE

```
SQL> set lines 200 pages 2000
SQL> col host_name for a20
SQL> select name,open_mode,log_mode,database_role,instance_name,host_name from v$database,v$instance;

NAME      OPEN_MODE            INSTANCE_NAME    HOST_NAME            DATABASE_ROLE
--------- -------------------- ---------------- -------------------- ----------------
DEVDB     READ WRITE           DEVDBDB          devoadr              SNAPSHOT STANDBY
```

STEP:8 TESTING ON THE SNAPSHOT STANDBY DATABASE

```
SQL> create user hari identified by dev;

User created.

SQL> grant connect,resource to hari

grant alerted.

SQL> conn hari/dev
Connected.

SQL> show user

USER IS "HARI"

SQL> create table sample1(code number, name char(20));
Table created.

SQL> insert into sample values(1,'sam');
1 row created.
commit;

SQL> select * from sample1;
  CODE        NAME
--------- ------------
 1          sam
```

WE CAN SEE WE ARE ABLE TO DO WRITE OPERATION ON SNAPSHOT DATABASE ALSO.

STEP:9 BOTH SIDE VERIFY THE ARCHIVED SEQUENCE select thread#,max(sequence#)

from v$archived_log group by thread#;

PRIMARY ::

```
SQL> select thread#,max(sequence#) from v$archived_log group by thread#;

THREAD#         MAX(SEQUENCE#)
----------    --------------
 1                  28
```

STANDBY ::

```
SQL> select thread#,max(sequence#) from v$archived_log group by thread#;

THREAD#         MAX(SEQUENCE#)
----------    --------------
 1                  28
```

select process,status,sequence# from v$managed_standby;

CONVERT BACK TO PHYSICAL STANDBY IF WE DIRECTLY TRYING TO CONVERT THEN WE WILL FACE THE BELOW ERROR

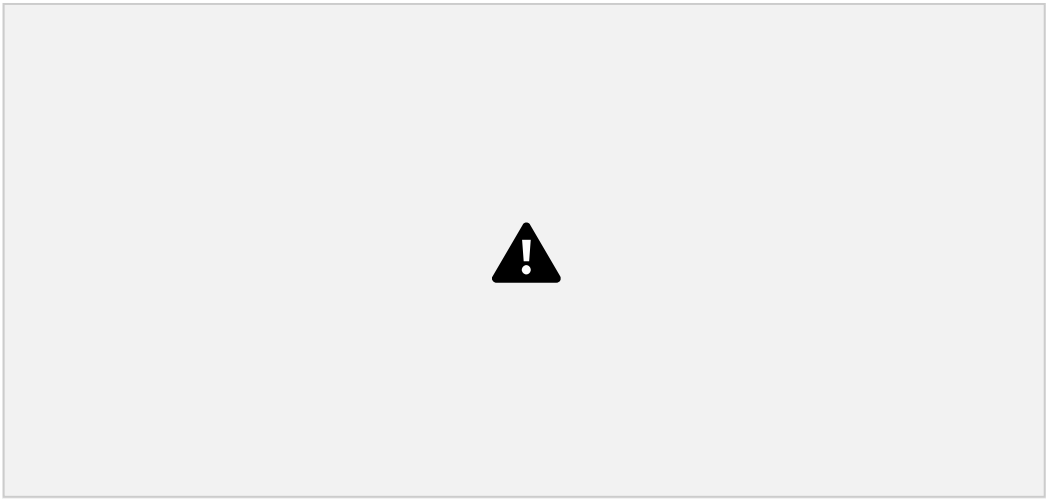
STEP 1 :: DOWN THE DATABASE AND OPEN IN MOUNT MODE

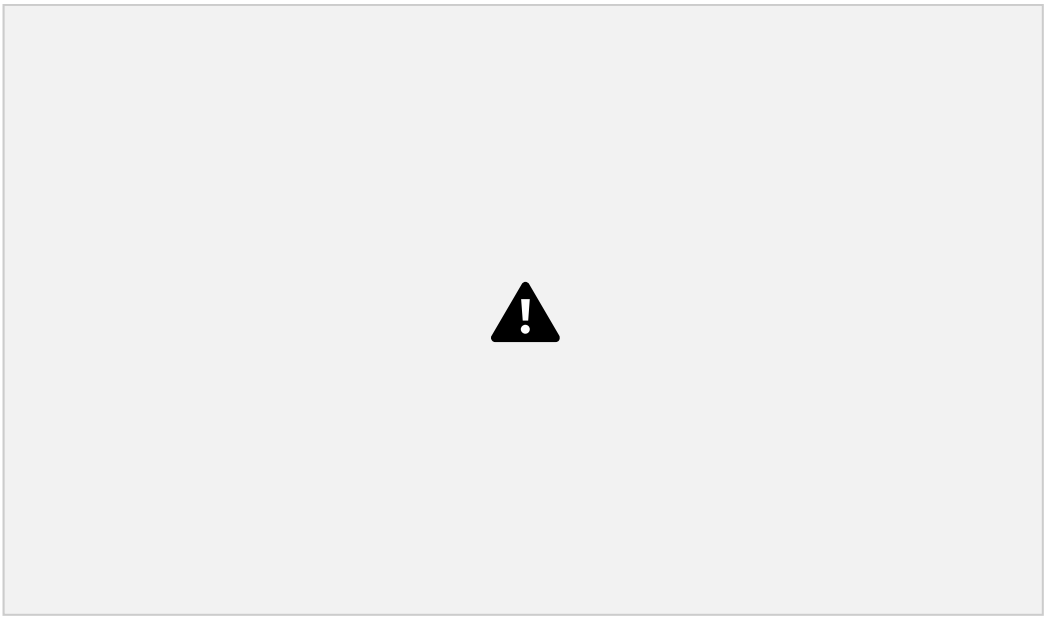
Shut immediate
Startup mount

STEP:: 2 CHECK THE STANDBY DATABASE_ROLE AND MODE. set lines 200 pages 2000
col host_name for a20
select name,open_mode,log_mode,database_role,instance_name,host_name fromv$database,v$instance;

STEP 3 :: CONVERT THE SNAPSHOT STANDBY DATABASE TO PHYSICAL STANDBY DATABASEalter database convert to physical standby;

ALERT LOG OUTPUT ::





STEP 4 :: TAKE BOUNCE OF DATABASE

shut immediate
startup mount

STEP 5 :: CHECK THE DATABASE ROLE

```
set lines 200 pages 2000
col host_name for a20
select name,open_mode,log_mode,database_role,instance_name,host_name
fromv$database,v$instance;
```

STEP 6 :: START THE RECOVERY

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


STEP 7 ::AFTER CONVERT TO PHYSICAL STANDBY DATABASE,CHECK THE TABLE STATUS FIRST CHECK THE DATABASE SYNC , IT SHOULD BE IN SYNC .

FROM ABOVE SNIP WE CAN SEE THAT , THE SAMPLE1 TABLE IS NOT PRESENT IN STANDBY DATABASE. AS WHILE CONVERTING BACK TO PHYSICAL MODE DATABASE ROLLEBACKED ALL THE TRANSACTIONS PERFORMED IN SNAPSHOT MODE.