**20 Standby**

**Standby database** is a backup database that mirrors the primary database. It is part of Oracle's **Data Guard** technology, which ensures high availability, disaster recovery, and **ZERO** data loss. Standby databases are crucial for maintaining business continuity and minimizing downtime during unexpected failures. Standby database is in either **MOUNT** or **READ ONLY**.

**There are two main types of standby databases:**

**Physical Standby**: This is an exact replica of the primary database, maintained through redo log application.

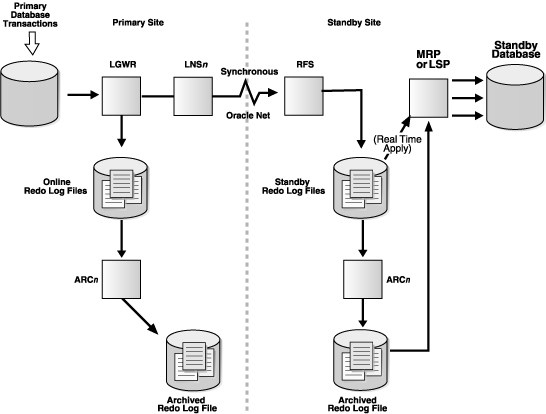
**Logical Standby**: This contains the same data as the primary database but allows for additional querying and reporting.

* From 11g 30 standby database can be created.
* For standby creation primary database should be enable with **ARCHIVELOG** mode.
* **PRIMARY** and **STANDBY** version should be same.
* Create **Password file** in primary and copy to standby server.
* **FORCE LOGGING** must be enables at database level.

**SQL>** **alter database force logging;**

**SQL>** **ALTER DATABASE NO FORCE LOGGING;**

* **STANDBY\_FILE\_MANAGEMENT=AUTO** in standby parameter file.



**LNS**: The **Log Network Service** process reads redo data from the redo log buffer or online redo logs on the primary database and is responsible for transmitting redo data from the primary database to the standby database

**RFS**: The **Remote File Server** process receives redo data sent from the primary database and writes it to the standby redo logs or archived redo logs on the standby database.

**MRP**: The **Managed Recovery Process** is responsible for applying redo data to the standby database to keep it synchronized with the primary database.

**STANDBY\_FILE\_MANAGEMENT**

* By-default this parameter is MANUAL in Standby.
* ALTER SYSTEM SET STANDBY\_FILE\_MANAGEMENT=AUTO;

**MANUAL**:

* File management is manual, meaning any changes to the primary database (like adding or deleting data files) must be manually replicated on the standby database.

**AUTO**:

* File management is automatic. When a data file is added or deleted on the primary database, the corresponding changes are automatically applied to the standby database.

**Enable Auto ARCHIVE or LOG shiping from primary to standby:**

**ARCHIVE Shipping:**

* Whenever archive is generated, only then data will sync with standby.

**SQL> alter system set Log\_archive\_dest\_1=‘LOCATION=/prod/ram/admin/arch’;**

**SQL>Log\_archive\_dest\_2=‘SERVICE=JMSSTBY ARCH’;**

**LOG** **Shipping:**

* This will not wait for archive to generate.
* Whenever the transaction is committed immediately it will sync with standby.

**Primary:**

* alter system set LOG\_ARCHIVE\_DEST\_2='service="JPSTBY" ASYNC valid\_for=(online\_logfile,all\_roles)';
* alter system set LOG\_ARCHIVE\_DEST\_2='service=JPSTBY LGWR’;

**Standby:**

* To enable **LOG** shipping STANDBY side we have to create standby redologfiles.
* If we have **N** number of redolog files in primary we have to create **N+1.(3+3+1)**
* Better to add standby redolog files in primary also, helps in future switchover.

SQL>alter database add standby logfile group 4 'D:\APP\ORADATA\REDO\_stby04.LOG' size 50m;

SQL> alter database add standby logfile group 5 'D:\APP\ORADATA\REDO\_stby05.LOG' size 50m;

SQL> alter database add standby logfile group 6 'D:\APP\ORADATA\REDO\_stby06.LOG' size 50m;

SQL> alter database add standby logfile group 7 'D:\APP\ORADATA\REDO\_stby07.LOG' size 50m;

**MRP Enable / Disable:**

* alter database recover managed standby database cancel; --Disable
* alter database recover managed standby database disconnect; --Enable
* alter database recover managed standby database using current **logfile** disconnect;--Enable

**PROTECTION MODES**

* In standby database configurations, **protection modes** often refer to the levels of data consistency and protection against data loss. These modes are critical in determining how the primary and secondary databases interact, especially in log shipping or other high-availability setups.
* By-default protection mode is set to **MAXIMUM PERFORMANCE.**
* **MAXIMUM PERFORMANCE –ARCH,LGWR**
* **MAXIMUM PROTECTION --LGWR**
* **MAXIMUM AVAILABILITY --LGWR**

**MAXIMUM PERFORMANCE:**

* If Standby is configured with **MAXIMUM PERFORMANCE**. If in case standby db is down or not responding. PRIMARY database will leave about standby and works as usual.
* This can be used for both **ARCH** shipping and **LOG** shipping.
* **Alter database set standby database to maximum performance;**

**MAXIMUM PROTECTION:**

* If Standby is configured with **MAXIMUM PROTECTION**. If in case standby db is down or not responding PRIMARY database will hang.
* In this we can achive ‘0’ data loss.
* When the transaction is commited first it writes the information in standby and give acknowledgment to primary then primary will show **COMMIT COMPLETE**.
* Better to configure this MODE with atleast 2 standby databases if one db goes down another will give acknowledgment.
* This will be used for **LOG** shipping.
* **Alter database set standby database to maximum protection;**

**MAXIMUM AVAILABILITY:**

* Initially this also works like **MAXIMUM PROTECTION**.
* It will wait for acknowledgment from standby for some time if not getting any acknowledgment it leaves standby. Works like **MAXIMUM PERFORMANCE**.
* There is a parameter for wait time interver **NET\_TIMEOUT** By-default it is 30 seconds.
* **Alter database set standby database to maximum availability;**

**To check MRP status:**

* select process,status from v$managed\_standby;

**To check archives applied or not:**

* Select sequence# ,applied from v$archived\_log;
* Select max(sequence#) from v$archivde\_log where applied=’YES’;

**To check STANDBY error:**

* Select error from v$archive\_dest where dest\_id=2;

**Active duplicate standby database creation without backup:**

* Same db name in primary and standby.
* We can have different SID.

**Primary:**

1. Enable force logging.
2. Add Standby TNS\_names
3. Create password file and copy to standby
4. Conn to rman

* rman target / auxiliary=sys/welcome@tns\_stby
* rman> duplicate target database for standby from active database nofilenamecheck;

**Standby**:

1. Create required directories & Create listener
2. Copy pfile from primary and edit accordingly
3. Copy password file from primary
4. If **PRIMARY** and **STANDBY** directory structure is differnet add this parameters.

* **db\_file\_name\_convert**='/data/jms/','/SSD/STBY/' '/data/oracle/','/SSD/STBY/ORADATA/'
* **log\_file\_name\_convert**='/data/oracle/app/oracle/oradata/jms/','/SSD/STBY/ORADATA/'

1. Startup nomount the database.

**Auxiliary duplicate standby database creation with backup:**

* Same db name in primary and standby.
* We can have different SID.

**Primary:**

1. Take Rman hot backup

* backup database;
* Backup archivelog all;
* Backup current controlfile for standby;

1. Enable force\_logging.

**Standby**:

1. Create required directories.
2. Copy pfile from primary and edit accordingly
3. Copy password file from primary
4. Create listener
5. If **PRIMARY** and **STANDBY** directory structure is differnet add this parameters.

* **db\_file\_name\_convert**='/data/jms/','/SSD/STBY/' '/data/oracle/','/SSD/STBY/ORADATA/'
* **log\_file\_name\_convert**='/data/oracle/app/oracle/oradata/jms/','/SSD/STBY/ORADATA/'

1. Startup nomount the database.

* rman auxiliary /
* **RMAN>** duplicate target database for standby backup ‘/SSD/BKP’ nofilenamecheck;

1. After restore enable data sync.

**Manual standby database creation using Rman hot backup:**

**Primary:**

1. Take Rman hot backup

* backup database;
* Backup archivelog all;
* Backup current controlfile for standby;

1. Enable force\_logging.
2. Transfer backups to standby.

**Standby:**

1. Create required directories.
2. Copy pfile and edit
3. Put db in nomount

**Rman> restore standby controlfile from ‘backup /location and/file name’;**

**Rman> alter database mount;**

1. register backups in standby controlfile

**RMAN>**catalog start with ‘/backup/location/‘;

**RMAN>**Restore database;

**RMAN>**Recover database;

**RMAN>**alter database open;

**If directory structure is different from primary prepare script**

* **db\_file\_name\_convert**='/data/oracle /jms/','/SSD/STBY/ORADATA/'
* **log\_file\_name\_convert**='/data/oracle/app/oracle/oradata/jms/','/SSD/STBY/ORADATA/'

1. **Check the datafiles location and file# in primary and create script for standby**

**RMAN>**run

{

Set newname for datafile 1 to ‘/prod/hydtst/oradata/system01.dbf’;

Set newname for datafile 2 to ‘/prod/hydtst/oradata/sysaux01.dbf’;

Set newname for datafile 3 to ‘/prod/hydtst/oradata/undotbs01.dbf’;

Set newname for datafile 4 to ‘/prod/hydtst/oradata/users01.dbf’;

restore database;

}

1. **Rman>**switch database to copy;
2. **Rman>**recover database;
3. **Rename LOG files:**

* alter database rename file ‘/prod/oradata/redo01.log’ to ‘/prod/oradata/redo01.log’;
* alter database rename file ‘/prod/oradata/redo02.log’ to ‘/prod/oradata/redo02.log’;
* alter database rename file ‘/prod/oradata/redo03.log’ to ‘/prod/oradata/redo03.log’;
* Add standby **N+1** redologfiles from primary.

1. **alter database open;**
2. Start MRP.

**Snapshot Standby:**

* This can be performed only on standby database.
* FRA should be enabled.
* Flashback should be enabled.
* Cancel MRP process.

**Convert Physical Standby to Snapshot standby**

* **Alter database convert to snapshot standby;**
* Bounce the database.

**Snapshot standby To Physical standby**

* Changes made during **SNAPSHOT** are not available when it is converted to **PHYSICAL** stansby again.
* Shutdown snapshot standby.
* Startup in mount.

**SQL>Alter database convert to physical standby;**

* Start **MRP** process.

**Switch Over Activity:**

* Before doing switchover activity both the primary and standby database should be in sync.
* Switch over first should be done on primary side.

**Primary**

* select name,open\_mode,database\_role,switchover\_status from v$database;
* Disconnect the active user sessions.
* Select USERNAME,SID FROM V$SESSION WHERE STATUS='ACTIVE';

**SQL>Alter database commit to switchover to standby;**

* Bounce the db and Check the db status.

**Standby**

* select name,open\_mode,database\_role,switchover\_status from v$database;

**SQL>Alter database commit to switchover to primary;**

* Bounce the db and Check the db status.
* enable the sync between primary and standby

**Switch Over Status:**

**NOT ALLOWED**:

* Either this is a standby database and the primary database has not been switched first, or this is primary database and no standby databases available.

**SESSIONS ACTIVE**:

* Indicates that there are active SQL sessions attached to the primary or standby database that need to be disconnect before the switchover.

**SWITCHOVER PENDING**:

* This is a standby database and the primary database switchover request has been received but not processed.

**SWITCHOVER LATENT**:

* The switchover was in pending mode, but did not complete and went back to the primary database.

**TO PRIMARY**:

* This is a standby database, with no active sessions, that is allowed to switch over to a primary database.

**TO STANDBY**:

* This is a primary database, with no active sessions, that is allowed to switch over to a standby database.

**RECOVERY NEEDED**:

* This is a standby database that has not received the switchover request.

**Failover Activity:**

* In case of worst situation primary database will not work or not available for production than we can activated standby database as a primary production database.
* If once the **MPR** is finished in standby again it will not be converted as standby.

1. **Check the database role and open\_mode.**

SQL>Select name,open\_mode,database\_role from v$database;

1. **Cancel the MRP process.**

SQL>alter database recover managed standby database cancel;

1. **The below commands will help to bring up standby as primary.**

SQL>alter database recover managed standby database finish;

SQL>alter database activate standby database;

1. Bounce your database and verify database name its open mode and its role.

SQL>shutdown immediate;

SQL>startup;

SQL>select name,open\_mode,database\_role from v$database;

**Primary & Standby Sync Failure Resolution:**

**FOR 11g:** Recover From Backup

1. Check the current\_scn from standby.

* Select current\_scn from v$database;

1. Take incremental\_scn backup on primary.

* RMAN>Backup incremental from scn 6542352 database format ‘/prod/bk/stby\_%u.bkp’;

1. Take controlfile backup for standby in primary.

* SQL>Alter database create standby controlfile as ‘/prod/bkp/standby\_controlfile.ctl’;

1. Copy the backup files to standby server.
2. Replace the backup controlfie with standby controlfile.
3. Shut down & mount the database.
4. Catalog the backup and Restore on standby.

* RMAN>catalog start with ‘/SSD/BKP/’;
* RMAN>recover database noredo;

1. Start the database & MPR process.
2. Check the scn of primary and standby.

FROM 12c: Recover From Service

1. Add primary TNS in standby server.
2. Check the current\_scn on both PRIMARY & STANDBY.

* Select current\_scn from v$database;

1. Stop MPR process.
2. Shut down database and start in mount state.
3. Connect to RMAN.

* rman target /
* RMAN>Recover database from service tns\_prim noredo using compressed backupset;

1. **Check the scn on both PRIMARY & STANDBY.**
2. **Shut down database and start in mount state.**
3. **Restore standby controlfile.**

* RMAN>restore standby controlfile from service tns\_prim;

1. **After restore check the database status and start the database.**
2. **Start MPR process.**