SQL Statements Relevant to Oracle Data Guard

There are many SQL and SQL*Plus statements that are useful for performing operations on standby databases in an Oracle Data Guard environment.

See the following topics:

- `ALTER DATABASE Statements`
- `ALTER SESSION Statements`
- `ALTER SYSTEM Statements`

ALTER DATABASE Statements

This table describes ALTER DATABASE statements that are relevant to Oracle Data Guard.

Table-1: ALTER DATABASE Statements Used in Data Guard Environments

ALTER DATABASE Statement	**Description**
`ACTIVATE [PHYSICAL|LOGICAL] STANDBY DATABASE FINISH APPLY]`	Performs a failover. The standby database must be mounted before it can be activated with this statement. **Note:** Do not use the ALTER DATABASE ACTIVATE STANDBY DATABASE statement to failover because it causes data loss. Instead, use the following best practices: • For physical standby databases, use the ALTER DATABASE RECOVER MANAGED STANDBY DATABASE statement with the FINISH keyword to perform the role transition as quickly as possible with little or no data loss and without rendering other standby databases unusable. • For logical standby databases, use the ALTER DATABASE PREPARE TO SWITCHOVER and ALTER DATABASE COMMIT TO SWITCHOVER statements.

`ADD [STANDBY] LOGFILE`
`[THREAD` *`integer`*`]`
`[GROUP` *`integer`*`]` *`filespec`*
	Adds one or more online redo log file groups or standby redo log file groups to the specified thread, making the log files available to the instance to which the thread is assigned.

`ADD`
`[STANDBY] LOGFILE MEMBER '`*`filename`*`' [ REUSE] TO` *`logfile-descriptor`*
	Adds new members to existing online redo log file groups or standby redo log file groups.

<table>
<tr><th>ALTER DATABASE Statement</th><th>Description</th></tr>
<tr><td>[ADD|DROP] SUPPLEMENTAL LOG DATA {PRIMARY KEY|UNIQUE INDEX} COLUMNS</td><td>This statement is for logical standby databases only.
Use it to enable full supplemental logging before you create a logical standby database. This is necessary because supplemental logging is the source of change to a logical standby database. To implement full supplemental logging, you must specify either the PRIMARY KEY COLUMNS or the UNIQUE INDEX COLUMNS keyword on this statement.</td></tr>
<tr><td>COMMIT TO SWITCHOVER</td><td>Performs a switchover to:
• Change the current primary database to the standby database role
• Change one standby database to the primary database role.
When switching over to a physical standby database, as of Oracle Database 12<i>c</i> Release 1 (12.1), the COMMIT TO SWITCHOVER statement has been replaced with the SWITCHOVER TO statement. The COMMIT TO SWITCHOVER statement is still supported, but Oracle recommends that you use the new SWITCHOVER TO statement.
<b>Note:</b> On logical standby databases, you issue the ALTER DATABASE PREPARE TO SWITCHOVER statement to prepare the database for the switchover before you issue the ALTER DATABASE COMMIT TO SWITCHOVER statement.</td></tr>
<tr><td>CONVERT TO [[PHYSICAL|SNAPSHOT] STANDBY] DATABASE</td><td>Converts a physical standby database into a snapshot standby database and vice versa.</td></tr>
<tr><td>CREATE [PHYSICAL|LOGICAL] STANDBY CONTROLFILE AS '<i>filename</i>' [REUSE]</td><td>Creates a control file to be used to maintain a physical or a logical standby database. Issue this statement on the primary database.</td></tr>
<tr><td>DROP [STANDBY] LOGFILE <i>logfile_descriptor</i></td><td>Drops all members of an online redo log file group or standby redo log file group.</td></tr>
<tr><td>DROP [STANDBY] LOGFILE MEMBER '<i>filename</i>'</td><td>Drops one or more online redo log file members or standby redo log file members.</td></tr>
</table>

ALTER DATABASE Statement	Description
`FAILOVER TO` *`target_db_name`*	This statement is for physical standby databases only. It initiates a failover to the specified host database.
`[NO]FORCE LOGGING`	Controls whether or not the Oracle database logs all changes in the database except for changes to temporary tablespaces and temporary segments. The [NO]FORCE LOGGING clause is required to prevent inconsistent standby databases. The primary database must at least be mounted (and it can also be open) when you issue this statement.
`GUARD`	Controls user access to tables in a logical standby database. Possible values are ALL, STANDBY, and NONE.
`MOUNT [STANDBY DATABASE]`	Mounts a standby database, allowing the standby instance to receive redo data from the primary instance.
`OPEN`	Opens a previously started and mounted database: • Physical standby databases are opened in read-only mode, restricting users to read-only transactions and preventing the generating of redo data. • Logical standby database are opened in read/write mode.
`PREPARE TO SWITCHOVER`	This statement is for logical standby databases only. It prepares the primary database and the logical standby database for a switchover by building the LogMiner dictionary *before* the switchover takes place. After the dictionary build has completed, issue the ALTER DATABASE COMMIT TO SWITCHOVER statement to switch the roles of the primary and logical standby databases.
`RECOVER MANAGED STANDBY DATABASE [ { DISCONNECT [FROM SESSION] | PARALLEL`	This statement starts and controls Redo Apply on physical standby databases. You can use

ALTER DATABASE Statement	Description
n| NODELAY | UNTIL CHANGE integer }...]	the RECOVER MANAGED STANDBY DATABASE clause on a physical standby database that is mounted, open, or closed.
RECOVER MANAGED STANDBY DATABASE CANCEL	The CANCEL clause cancels Redo Apply on a physical standby database after applying the current archived redo log file. **Note:** Several clauses and keywords were deprecated and are supported for backward compatibility only.
RECOVER MANAGED STANDBY DATABASE FINISH	The FINISH clause initiates failover on the target physical standby database and recovers the current standby redo log files. Use the FINISH clause only in the event of the failure of the primary database. This clause overrides any delay intervals specified. **Note:** Several clauses and keywords were deprecated and are supported for backward compatibility only.
REGISTER [OR REPLACE] [PHYSICAL|LOGICAL] LOGFILE *filespec*	Allows the registration of manually copied archived redo log files. **Note**: Issue this command only after manually copying the corresponding archived redo log file to the standby database. Issuing this command while the log file is in the process of being copied or when the log file does not exist may result in errors on the standby database at a later time.
RECOVER TO LOGICAL STANDBY *new_database_name*	Instructs apply services to continue applying changes to the *physical* standby database until you issue the command to convert the database to a *logical* standby database.
RESET DATABASE TO INCARNATION *integer*	Resets the target recovery incarnation for the database from the current incarnation to a different incarnation.
SET STANDBY DATABASE TO MAXIMIZE {PROTECTION|AVAILABILITY|PERFORMANCE}	Use this clause to specify the level of protection for the data in your Oracle Data Guard configuration. You specify this clause from the primary database.
START LOGICAL STANDBY APPLY INITIAL [*scn-value*]] [NEW PRIMARY *dblink*]	This statement is for logical standby databases only.

ALTER DATABASE Statement	Description
	It starts SQL Apply on a logical standby database.
`{STOP|ABORT} LOGICAL STANDBY APPLY`	This statement is for logical standby databases only. Use the STOP clause to stop SQL Apply on a logical standby database in an orderly fashion. Use the ABORT clause to stop SQL Apply abruptly.
`SWITCHOVER TO` *`target_db_name`*	This statement is for physical standby databases only. It initiates a switchover on the primary database to the specified physical standby database.

ALTER SESSION Statements

This table describes the ALTER SESSION statements that are relevant to Oracle Data Guard.

Table-2: ALTER SESSION Statements Used in Oracle Data Guard Environments

ALTER SESSION Statement	Description
`ALTER SESSION [ENABLE|DISABLE] GUARD`	This statement is for logical standby databases only. This statement allows privileged users to turn the database guard on and off for the current session.
`ALTER SESSION SYNC WITH PRIMARY`	This statement is for physical standby databases only. This statement synchronizes a physical standby database with the primary database, by blocking until all redo data received by the physical standby at the time of statement invocation has been applied.

ALTER SYSTEM Statements

This table describes the ALTER SYSTEM statements that are relevant to Oracle Data Guard.

Table-3: ALTER SYSTEM Statements Used in Oracle Data Guard Environments

ALTER SYSTEM Statement	Description
`ALTER SYSTEM FLUSH REDO TO target_db_name [[NO] CONFIRM APPLY]`	This statement flushes redo data from a primary database to a standby database and optionally waits for the flushed redo data to be applied to a physical or logical standby database. This statement must be issued on a mounted, but not open, primary database.