Role Transitions

An Oracle Data Guard configuration consists of one database that functions in the primary role and one or more databases that function in the standby role.

To see the current role of the databases, query the DATABASE_ROLE column in the V$DATABASE view.

The number, location, and type of standby databases in an Oracle Data Guard configuration and the way in which redo data from the primary database is propagated to each standby database determine the role-management options available to you in response to a primary database outage.

Information about using the Oracle Data Guard broker to:

- Simplify switchovers and failovers by allowing you to invoke them using either a single key click in Oracle Enterprise Manager Cloud Control or a single command in the DGMGRL command-line interface.

- Enable **fast-start failover** to fail over *automatically* when the primary database becomes unavailable. When fast-start failover is enabled, the Oracle Data Guard broker determines if a failover is necessary and initiates the failover to the specified target standby database automatically, with no need for DBA intervention.

Introduction to Role Transitions

A database operates in one of the following mutually exclusive roles: **primary** or **standby**.

Oracle Data Guard enables you to change these roles dynamically by using SQL statements, or by using either of the Oracle Data Guard broker's interfaces. Oracle Data Guard supports the following role transitions:

- **Switchover**
  Allows the primary database to switch roles with one of its standby databases. There is no data loss during a switchover. After a switchover, each database continues to participate in the Oracle Data Guard configuration with its new role.
- **Failover**
  Changes a standby database to the primary role in response to a primary database failure. If the primary database was not operating in either maximum protection mode or maximum availability mode before the failure, some data loss may occur. If Flashback Database is enabled on the primary database, it can be reinstated as a standby for the new primary database once the reason for the failure is corrected.

Starting with Oracle Database Release 21c, you can perform switchover or failover for a pluggable database (PDB) within a multitenant container database (CDB). This functionality is supported only with Oracle Data Guard broker.

Preparing for a Role Transition

Before starting any role transitions, you must verify that each database is properly configured and that there are no redo transport errors or redo gaps at the standby database.

- Verify that each database is properly configured for the role that it is about to assume..

  Note:

  You must define the LOG_ARCHIVE_DEST_*n* and LOG_ARCHIVE_DEST_STATE_*n* parameters on each standby database so that when a switchover or failover occurs, all standby sites continue to receive redo data from the new primary database.

- Verify that there are no redo transport errors or redo gaps at the standby database by querying the V$ARCHIVE_DEST_STATUS view on the primary database.
  For example, the following query would be used to check the status of the standby database associated with LOG_ARCHIVE_DEST_2:

```
SQL> SELECT STATUS, GAP_STATUS FROM V$ARCHIVE_DEST_STATUS WHERE DEST_ID
= 2;

STATUS GAP_STATUS
--------- ------------------------

VALID NO GAP
```

- Do not proceed until the value of the STATUS column is VALID and the value of the GAP_STATUS column is NOGAP, for the row that corresponds to the standby database.

- Ensure temporary files exist on the standby database that match the temporary files on the primary database.
- Remove any delay in applying redo that may be in effect on the standby database that is set to become the new primary database. Not removing the delay results in a longer switchover time, and may cause the switchover to be disallowed.
- Before performing a switchover to a physical standby database that is in real-time query mode, consider bringing all instances of that standby database to the mounted but not open state to achieve the fastest possible role transition and to cleanly terminate any user sessions connected to the physical standby database prior to the role transition.
- When you perform a switchover from an Oracle RAC primary database to a physical standby database, it is not necessary to shut down all but one primary database instance.

Choosing a Target Standby Database for a Role Transition

For an Oracle Data Guard configuration with multiple standby databases, there are a number of factors to consider when choosing the target standby database for a role transition.

These include the following:

- Locality of the standby database.
- The capability of the standby database (hardware specifications—such as the number of CPUs, I/O bandwidth available, and so on).
- The time it takes to perform the role transition. This is affected by how far behind the standby database is in applying redo data, and how much flexibility you have in terms of trading off application availability with data loss.
- Standby database type.

Oracle Data Guard provides the V$DATAGUARD_STATS view, which you can use to evaluate each standby database in terms of the currency of the data in the standby database, and the time needed to perform a role transition if all available redo data is applied to the standby database.

For example:

```
SQL> COLUMN NAME FORMAT A24
SQL> COLUMN VALUE FORMAT A16
SQL> COLUMN DATUM_TIME FORMAT A24
SQL> SELECT NAME, VALUE, DATUM_TIME FROM V$DATAGUARD_STATS;


NAME                     VALUE            DATUM_TIME
------------------------ ---------------- ------------------------
transport lag            +00 00:00:00     06/18/2009 12:22:06
apply lag                +00 00:00:00     06/18/2009 12:22:06
apply finish time        +00 00:00:00.000

estimated startup time   9
```

This query output shows that the standby database has received and applied all redo generated by the primary database. These statistics were computed using data received from the primary database as of 12:22.06 on 06/18/09.

The apply lag and transport lag metrics are computed based on data received from the primary database. These metrics become stale if communications between the primary and standby database are disrupted. An unchanging value in the DATUM_TIME column for the apply lag and transport lag metrics indicates that these metrics are not being updated and have become stale, possibly due to a communications fault between the primary and standby databases.

Switchovers

A switchover is typically used to reduce primary database downtime during planned outages.

Planned outages are events such as operating system or hardware upgrades, or rolling upgrades of the Oracle database software and patch sets.

A switchover takes place in two phases. In the first phase, the existing primary database undergoes a transition to a standby role. In the second phase, a standby database undergoes a transition to the primary role.

Figure 9-1 Oracle Data Guard Configuration Before Switchover

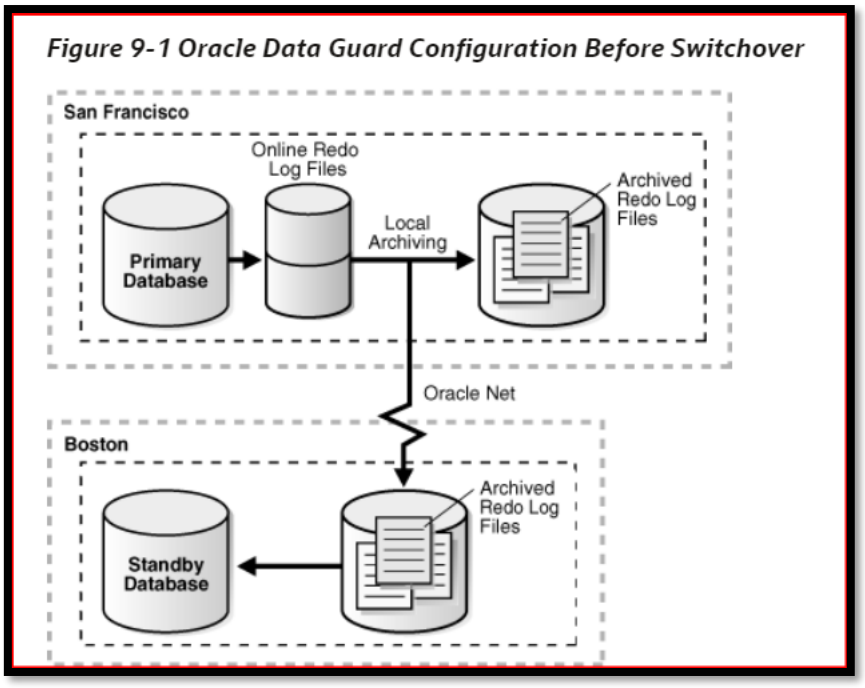
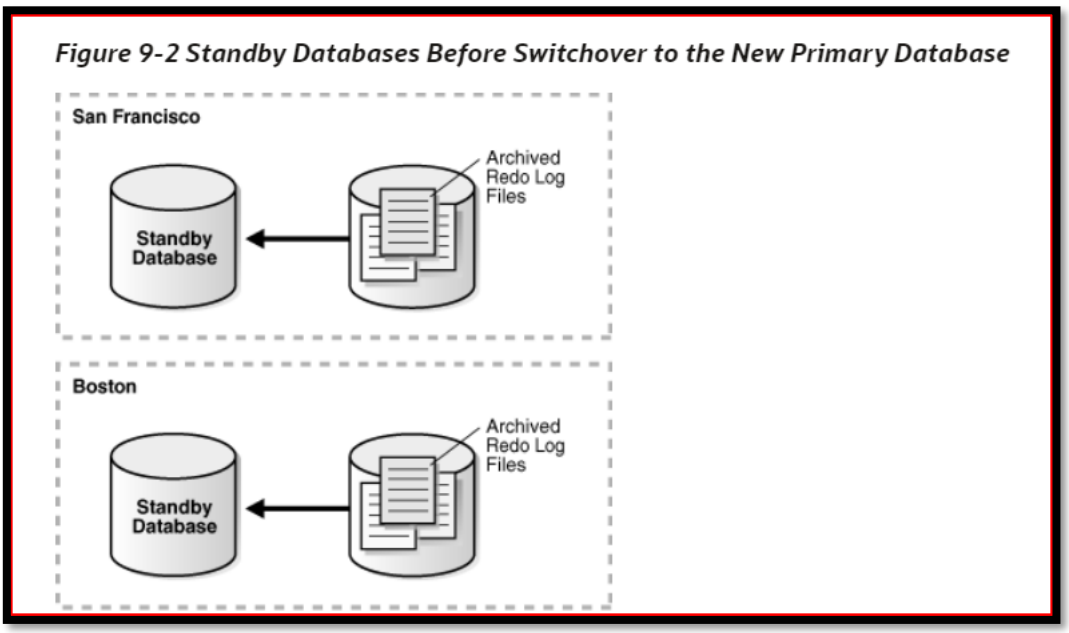
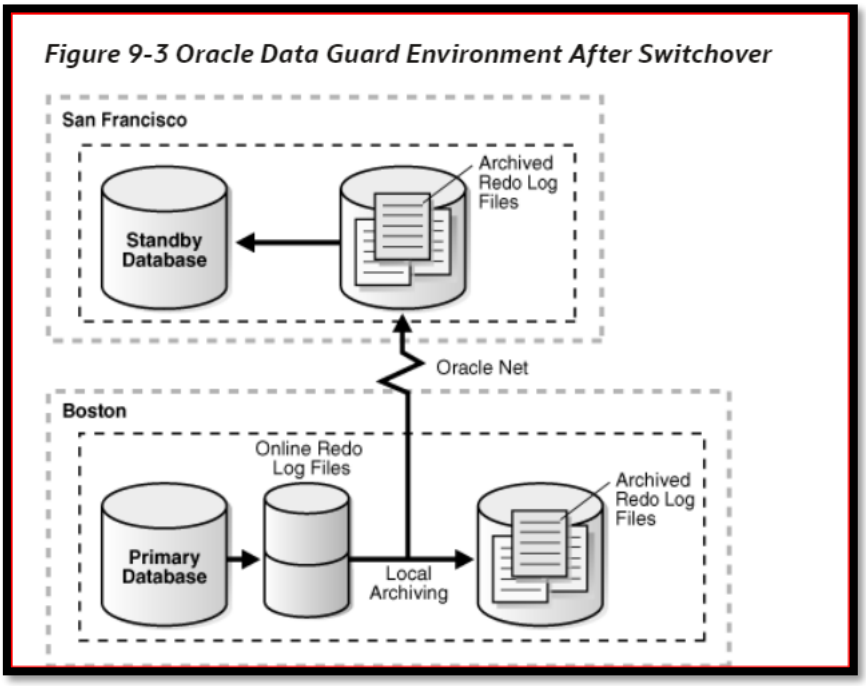


Figure 9-2 Standby Databases Before Switchover to the New Primary Database





Preparing for a Switchover

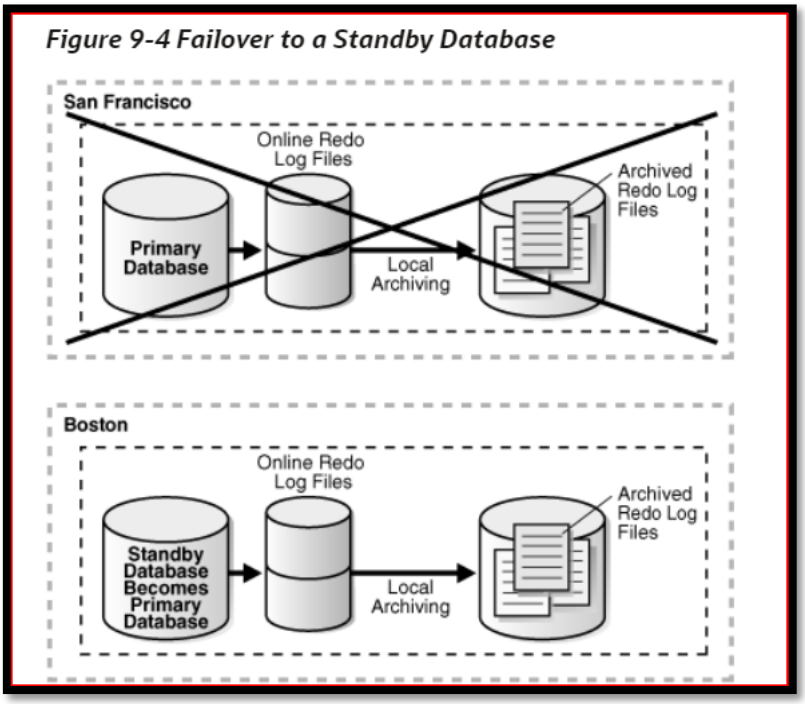
In addition, the following prerequisites must be met for a switchover:

- For switchovers involving a physical standby database, verify that the primary database is open and that Redo Apply is active on the standby database.
- For switchovers involving a logical standby database, verify that both the primary and standby database instances are open and that SQL Apply is active.

Failovers

A failover is typically used only when the primary database becomes unavailable, and there is no possibility of restoring it to service within a reasonable period of time.

The specific actions performed during a failover vary based on whether a logical or a physical standby database is involved in the failover, the state of the Oracle Data Guard configuration at the time of the failover, and on the specific SQL statements used to initiate the failover.



If possible, before performing a failover, transfer as much of the available and unapplied primary database redo data as possible to the standby database.

In addition, the following prerequisites must be met for a failover:

- If a standby database currently running in maximum protection mode is involved in the failover, then first place it in maximum performance mode by issuing the following statement on the standby database:

```
SQL> ALTER DATABASE SET STANDBY DATABASE TO MAXIMIZE PERFORMANCE;
```

Keeping Physical Standby Sessions Connected During Role Transition

As of Oracle Database 12c Release 2 (12.2.0.1), when a physical standby database is converted into a primary you have the option to keep any sessions connected to the physical standby connected, without disruption, during the switchover/failover.
To enable this feature, set the STANDBY_DB_PRESERVE_STATES initialization parameter in your init.ora file before the standby instance is started.

This parameter applies to physical standby databases only. The allowed values are:

- NONE — No user sessions or current buffers on the standby are retained during a switchover/failover. This is the default value.
- ALL — Both user sessions and current buffers are retained during switchover/failover.
- SESSION — User sessions are retained during switchover/failover.
- BUFFER — Current buffers are retained during switchover/failover.

Performing a Switchover to a Physical Standby Database

These steps describe how to perform a switchover to a physical standby database.

Verify that the target standby database is ready for switchover.

The switchover statement has a VERIFY option that results in checks being performed of many conditions required for switchover. Some of the items checked are: whether Redo Apply is running on the switchover target; whether the release version of the switchover target is 12.1 or later; whether the switchover target is synchronized; and whether it has MRP running.

Suppose the primary database has a DB_UNIQUE_NAME of BOSTON and the switchover target standby database has a DB_UNIQUE_NAME of CHICAGO. On the primary database BOSTON, issue the following SQL statement to verify that the switchover target, CHICAGO, is ready for switchover:

```
SQL> ALTER DATABASE SWITCHOVER TO CHICAGO VERIFY;
ERROR at line 1:
ORA-16470: Redo Apply is not running on switchover target
```

If this operation had been successful, a Database Altered message would have been returned but in this example an ORA-16470 error was returned. This error means that the switchover target CHICAGO is not ready for switchover. Redo Apply must be started before the switchover operation.

After Redo Apply is started, issue the following statement again:

```
SQL> ALTER DATABASE SWITCHOVER TO CHICAGO VERIFY;
ERROR at line 1:

ORA-16475: succeeded with warnings, check alert log for more details
```

The switchover target, CHICAGO, is ready for switchover. However, the warnings indicated by the ORA-16475 error may affect switchover performance. The alert log contains messages similar to the following:

SWITCHOVER VERIFY WARNING: switchover target has dirty online redo logfiles that require clearing. It takes time to clear online redo logfiles. This may slow down switchover process.

You can fix the problems or if switchover performance is not important, those warnings can be ignored. After making any fixes you determine are necessary, issue the following SQL statement again:

```
SQL> ALTER DATABASE SWITCHOVER TO CHICAGO VERIFY;

Database altered.
```

1. The switchover target, CHICAGO, is now ready for switchover.
2. Initiate the switchover on the primary database, BOSTON, by issuing the following SQL statement:

```
SQL> ALTER DATABASE SWITCHOVER TO CHICAGO;

Database altered.
```

Value of DATABASE_ROLE column in V$DATABASE	Cause and Remedial Action
BOSTON database is primary, CHICAGO database is standby	Cause: The BOSTON database failed to convert to a standby database role. Action: See the alert log for details on the error that prevented BOSTON from switching to a standby role, take the necessary actions to fix the error, reopen one of the nodes of BOSTON if necessary, and repeat the switchover process from Step 1.
BOSTON database is standby, CHICAGO database is standby	Cause: The CHICAGO database failed to convert to a primary database role. Action: Issue the following SQL statement to convert either BOSTON or CHICAGO to a primary database: SQL> ALTER DATABASE SWITCHOVER TO *target_db_name* FORCE; For example: • On the CHICAGO database, issue the following SQL statement to convert it to a primary database: • ALTER DATABASE SWITCHOVER TO CHICAGO FORCE; • On the BOSTON database, issue the following SQL statement to convert it to a primary database: ALTER DATABASE SWITCHOVER TO BOSTON FORCE; If the SQL statement fails with an ORA-16473 error, then you must start Redo Apply before reissuing the command. Restart Redo Apply as follows: SQL> ALTER DATABASE RECOVER MANAGED STANDBY DATABASE DISCONNECT; Reissue the switchover command as follows: SQL> ALTER DATABASE SWTICHOVER TO BOSTON FORCE;

Value of DATABASE_ROLE column in V$DATABASE	Cause and Remedial Action
	Database altered.
BOSTON database is standby, CHICAGO database is primary	Cause: The BOSTON and CHICAGO databases have successfully switched to their new roles, but there was an error communicating the final success status back to BOSTON. Action: Continue to Step 3 to finish the switchover operation.

2. Issue the following SQL statement on the new primary database, CHICAGO, to open it.

   ```
   SQL> ALTER DATABASE OPEN;
   ```

3. Issue the following SQL statement to mount the new physical standby database, BOSTON:

   ```
   SQL> STARTUP MOUNT;
   ```

   Or, if BOSTON is an Oracle Active Data Guard physical standby database, then issue the following SQL statement to open it read only:

   ```
   SQL> STARTUP;
   ```

4. Start Redo Apply on the new physical standby database. For example:

   ```
   SQL> ALTER DATABASE RECOVER MANAGED STANDBY DATABASE DISCONNECT FROM SESSION;
   ```

Performing a Failover to a Physical Standby Database

These steps describe how to perform a failover to a physical standby database.

1. If the primary database can be mounted, then flush any unsent archived and current redo from the primary database to the standby database. If this operation is successful, a zero data loss failover is possible even if the primary database is not in a zero data loss data protection mode.

   First, ensure that Redo Apply is active at the target standby database. Then mount, but do not open the primary database. If the primary database cannot be mounted, go to Step 2.

If not already done, then set up the remote LOG_ARCHIVE_DEST_*n* configured at the primary to point to the target destination. (You may not have any remote LOG_ARCHIVE_DEST_*n* configured if the target destination was serviced by a far sync instance, or was a terminal standby in a cascaded configuration.) Also, ensure that the primary can connect to the target destination by verifying that the NET_ALIAS_TARGET_DB_NAME is valid and properly established.

```
SQL> ALTER SYSTEM SET
LOG_ARCHIVE_DEST_6='SERVICE=NET_ALIAS_TARGET_DB_NAME ASYNC
VALID_FOR=(online_logfile, primary_role)
DB_UNIQUE_NAME="target_db_unique_name"' SCOPE=memory;


SQL> ALTER SYSTEM SET LOG_ARCHIVE_DEST_STATE_6=ENABLE;
```

It is also assumed that the LOG_ARCHIVE_CONFIG specification includes the DB_UNIQUE_NAME of the target destination at the primary (and LOG_ARCHIVE_CONFIG at the target destination includes the DB_UNIQUE_NAME of the primary). If not, then add that information to the LOG_ARCHIVE_CONFIG at the primary and target destination as required.

Issue the following SQL statement at the primary database:

```
SQL> ALTER SYSTEM FLUSH REDO TO target_db_name;
```

1. For target_db_name, specify the DB_UNIQUE_NAME of the standby database that is to receive the redo flushed from the primary database.

   This statement flushes any unsent redo from the primary database to the standby database, and waits for that redo to be applied to the standby database.

   If this statement completes without any errors, go to Step 5.If the statement completes with any error, or if it must be stopped because you cannot wait any longer for the statement to complete, continue with Step 2.

2. Query the V$ARCHIVED_LOG view on the target standby database to obtain the highest log sequence number for each redo thread.

   For example:

```
SQL> SELECT UNIQUE THREAD# AS THREAD, MAX(SEQUENCE#) OVER (PARTITION
BY thread#) AS LAST from V$ARCHIVED_LOG;

    THREAD       LAST
---------- ----------
   1            100
```

If possible, copy the most recently archived redo log file for each primary database redo thread to the standby database if it does not exist there, and register it. This must be done for each redo thread.

For example:

```
SQL> ALTER DATABASE REGISTER PHYSICAL LOGFILE 'filespec1';
```

Query the V$ARCHIVE_GAP view on the target standby database to determine if there are any redo gaps on the target standby database.

For example:

```
SQL> SELECT THREAD#, LOW_SEQUENCE#, HIGH_SEQUENCE# FROM V$ARCHIVE_GAP;

THREAD#    LOW_SEQUENCE# HIGH_SEQUENCE#
---------- ------------- --------------
         1            90             92
```

In this example, the gap comprises archived redo log files with sequence numbers 90, 91, and 92 for thread 1.

If possible, copy any missing archived redo log files to the target standby database from the primary database and register them at the target standby database. This must be done for each redo thread.

For example:

```
SQL> ALTER DATABASE REGISTER PHYSICAL LOGFILE 'filespec1';
```

The query executed in Step 3 displays information for the highest gap only. After resolving a gap, you must repeat the query until no more rows are returned.

If, after performing Step 2 through Step 4, you are not able to resolve all gaps in the archived redo log files (for example, because you do not have access to the system that hosted the failed primary database), then you can expect some data loss during the failover.

Issue the following SQL statement on the target standby database:

```
SQL> ALTER DATABASE RECOVER MANAGED STANDBY DATABASE CANCEL;
```

Issue the following SQL statement on the target standby database:

```
SQL> ALTER DATABASE FAILOVER TO target_db_name;
```

For example, suppose the target standby database is named CHICAGO:

```
SQL> ALTER DATABASE FAILOVER TO CHICAGO;
```

If this statement completes without any errors, proceed to Step 10.

If there are errors, go to Step 7.

If an error occurs, try to resolve the cause of the error and then reissue the statement.

- If successful, go to Step 10.
- If the error still occurs and it involves a far sync instance, go to Step 8.

- If the error still occurs and there is no far sync instance involved, go to Step 9.

This step is for far sync instance error cases only. If the error involves a far sync instance (for example, it is unavailable) and you have tried resolving the issue and reissuing the statement without success, then you can use the FORCE option. For example:

```
SQL> ALTER DATABASE FAILVOVER TO CHICAGO FORCE;
```

1. The FORCE option instructs the failover to ignore any failures encountered when interacting with the far sync instance and proceed with the failover, if at all possible. (The FORCE option has meaning only when the failover target is serviced by a far sync instance.)
   If the FORCE option is successful, go to Step 10.
   If the FORCE option is unsuccessful, go to Step 9.

2. Perform a data loss failover.

   If an error condition cannot be resolved, a failover can still be performed (with some data loss) by issuing the following SQL statement on the target standby database:

   ```
   SQL> ALTER DATABASE ACTIVATE PHYSICAL STANDBY DATABASE;
   ```

In the following example, the failover operation fails with an ORA-16472 error. That error means the database is configured in MaxAvailability or MaxProtection mode but data loss is detected during failover.

```
SQL> ALTER DATABASE FAILOVER TO CHICAGO;
ERROR at line 1:
ORA-16472: failover failed due to data loss
```

You can complete the data loss failover by issuing the following SQL statement:

```
SQL> ALTER DATABASE ACTIVATE PHYSICAL STANDBY DATABASE;

Database altered.
```

Open the new primary database:

```
SQL> ALTER DATABASE OPEN;
```

Oracle recommends that you perform a full backup of the new primary database. If Redo Apply has stopped at any of the other physical standby databases in your Data Guard configuration, then restart it. For example:

```
SQL> ALTER DATABASE RECOVER MANAGED STANDBY DATABASE DISCONNECT;
```