# Practices for Lesson 2: Backup and Recovery Configuration

Practices for Lesson 2: Overview

Overview

In these practices, you learn how to configure your database to enable recovery from various losses. You verify the control file configuration, the fast recovery area (FRA), redo log groups, ARCHIVELOG mode, and redundant archive log destinations.

How to configure your database for recovery:

Ensure redundancy of control files. If a control file is damaged or lost, recovery is easier if you have another copy.

Review the fast recovery area configuration.

Ensure that there are at least two redo log members in each group. If a redo log member is damaged or lost, recovery is easier when you have an additional member in the group.

Place your database in ARCHIVELOG mode. In all cases, you will be able to recover the database either completely or incompletely depending on which database files have been damaged or lost.

Configure redundant archive log destinations. In cases where you lost archive log files and you need them to recover the database, you will be able to perform an incomplete recovery, unless you have a duplicate version of the archive log in another destination.

Practice 2-1: Verifying that the Control File is Multiplexed

Overview

In this practice, you verify that the control file is multiplexed.

A control file is a small binary file that describes the structure of the database. It must be available for writing by the Oracle server whenever the database is mounted or opened. Without this file, the database cannot be mounted, and recovery or re-creation of the control file is required. Your database should have a minimum of two control files on different storage devices to minimize the impact of a loss of one control file. The loss of a single control file causes the instance to fail because all control files must be available at all times. However, recovery can be a simple matter of copying one of the other control files. The loss of all control files is slightly more difficult to recover from, but is not usually catastrophic.

Assumptions

You are logged in as the oracle user.

Time Estimate

It should take approximately 15 minutes to complete this practice.

Tasks

Open a terminal window and use the oraenv script to set the environment variables for the

orclcdb container database.

Start SQL\*Plus and connect to the CDB root as the SYS user with the SYSDBA privilege.

If the database instance is not started, start the instance and open the database.

Find out how many control files exist in the database. The query returns the names of two control files (control01.ctl and control02.ctl), which verifies that the control files are multiplexed.

When the CDB was created, DBCA created two control files. When you use the CREATE DATABASE command in SQL\*Plus to create a database, you configure the CONTROL\_FILES parameter to generate two control files and set their names.

View the CONTROL\_FILES parameter. Notice that the paths to the control files are stored in this parameter. The results below are formatted for easier viewing.

Create a parameter file (PFILE) from the server parameter file (SPFILE).

Shut down the database instance in IMMEDIATE mode.

Exit SQL\*Plus.

Create a directory for the new control file.

Before you edit your PFILE, make a backup of it.

Copy one of the control files to the directory you created in a previous step (/u01/app/oracle/controlfiles\_dir/ORCLCDB) and name the file control03.ctl.

Open the PFILE (initorclcdb.ora) in the editor of your choice (vi or gedit) and add the name of the new control file to the end of the list of control files. Include the path. Be certain not to enter spaces between the single quotes and commas in the control\_files= line. Be certain that this line is one continuous line, without line breaks. Save and close the file.

Start SQL\*Plus and connect to the root container as the SYS user with the SYSDBA privilege. You are connected to an idle instance.

Start the database instance.

View the CONTROL\_FILES parameter again.

**Question:** Why does the CONTROL\_FILES parameter still show only two control files? **Answer:** By default, the database instance starts up with the SPFILE. If an SPFILE does not exist, then the instance starts up with a PFILE. In this case, both an SPFILE and PFILE

are present, so the SPFILE takes precedence. You configured the PFILE, not the SPFILE.

The SPFILE still contains only two references.

Re-create the third control file because the current version is no longer an exact copy of the others.

Shut down the database instance with the IMMEDIATE option.

Exit SQL\*Plus.

Use the cp command to re-create control03.ctl.

Re-create the SPFILE from the updated PFILE.

Start SQL\*Plus and connect to the CDB root as the SYS user with the SYSDBA privilege. You are connected to an idle instance.

Create the SPFILE.

Start the database instance.

View the CONTROL\_FILES parameter again. The third control file is now included in the list, which indicates that the SPFILE is configured properly. The results below are formatted for easier viewing.

Query the V$CONTROLFILE view to confirm the number of control files. The result indicates that three control files are defined.

Practice 2-2: Configuring the Size of the Fast Recovery Area

Overview

In this practice, you review the fast recovery area (FRA) configuration and change its size to 12 GB.

Assumptions

You are logged in to SQL\*Plus from the previous practice.

Time Estimate

It should take approximately 5 minutes to complete this practice.

Tasks

Evaluate the space needed for the FRA. The amount of disk space to allocate for the FRA depends on the size and activity levels of your database. As a general rule, the larger the FRA, the more useful it is. Ideally, the FRA should be large enough for copies of your data and control files, as well as for flashback, online redo, and archived logs needed to recover the database with the backups kept based on the retention policy (covered in one of the next practices). In short, the FRA should be at least twice the size of the database so that it can hold one backup and several archived logs.

View the values of the DB\_RECOVERY\_FILE\_DEST and DB\_RECOVERY\_FILE\_ DEST\_SIZE initialization parameters.

**Question:** Is the fast recovery area enabled?

**Answer:** Yes. The DB\_RECOVERY\_FILE\_DEST and DB\_RECOVERY\_FILE\_DEST\_SIZE

parameters values are not null, indicating that the fast recovery area is enabled.

**Question:** What changes can you make to the fast recovery area?

**Answer:** You can change the location and size of the fast recovery area.

**Question:** Does changing the size of the fast recovery area require the database to be restarted?

**Answer:** No, a restart is not required for this change because the

DB\_RECOVERY\_FILE\_DEST\_SIZE parameter is dynamic.

Change the size of the fast recovery area to 12GB and set the scope to BOTH.

**Note:** If the archived redo log file destination fills up or cannot be written to, the database will halt. You would then need to remove archived redo log files from the archived redo log file destination so that the database could resume operations. This activity is covered in one of the next practices.

View the DB\_RECOVERY\_FILE\_DEST\_SIZE initialization parameter again. The result verifies that the size has been set to 12GB.

Practice 2-3: Verifying that the Redo Log File Is Multiplexed

Overview

Ensure that there are at least two redo log members in each group. If you are using file system storage, then each member should be distributed on separate disks or controllers so that no single equipment failure impacts an entire log group. The loss of an entire current log group is one of the most serious media failures because it can result in data loss. The loss of a single member of a multi-member log group does not affect database operation (other than causing an alert to be published in the alert log). One set of members should be stored in the FRA.

Assumptions

You are logged in to SQL\*Plus from the previous practice.

Time Estimate

It should take approximately 20 minutes to complete this practice.

Tasks

Query V$LOGFILE to determine the configuration (number of members) for each redo log group. The result shows that there are currently three log groups (1, 2, and 3) and only one member in each group.

**Question:** Why is it recommended to have three groups when two would be sufficient?

**Answer:** The Oracle Database server treats the online redo log groups as a circular buffer in which to store transaction information, filling one group and then moving on to the next. After all groups have been written to, the Oracle Database server begins overwriting information in the first log group. If the database is configured in ARCHIVELOG mode, the LGWR cannot overwrite data in the first log group if it has not been archived.

**Question:** Can multiplexing redo logs impact database performance?

**Answer:** Multiplexing redo logs may heavily influence database performance because a commit cannot complete until the transaction information has been written to the logs by LGWR. You must place your redo log files on your fastest disks served by your fastest controllers. If possible, do not place any other database files on the same disks as your redo log files. Because only one group is written to at a given time, there is no performance impact in having members from several groups on the same disk.

Add another member to each redo log group. Name each member redo*nn*b.log, where

*nn* represents the group number.

Verify that the redo log files are now multiplexed. The query result shows that each group has two members, and therefore, the redo log files are multiplexed. Observe the INVALID status of the newly added redo log members. This status is expected because the new members have not yet been written to by LGWR. When a log switch occurs and the group containing the new member becomes CURRENT, the new member's status will change to null.

Switch the log files and observe the changes.

Find out which log group is the current log group. In this example, the query result shows that group 1 is the current group. Your current group may be different.

Switch the log files three times.

Query the V$LOGFILE view again. Notice that as a result of the log file switch, the new members' statuses have changed to null.

Query the V$LOG view again to learn which log group is now the current group. In this example, the results show that the LGWR is writing to group 1. Your group may be different. Your statuses may be different too. An INACTIVE status means the log group is no longer needed for database instance recovery.

Switch the log file.

Query the V$LOG view again. The current group has changed to group 2, and the former current group's status is now ACTIVE. Your current group may be different. An ACTIVE status means that the log group is active, but it’s not the current log group. It is needed for crash recovery. It may be in use for block recovery.

Switch the log file again.

Query the V$LOG view again. The current group has changed again to group 3, and the status of both the other groups is now ACTIVE. Your current group may be different.

**Question:** Can the LGWR background process write to only one member of the CURRENT

group in case the other members are missing or damaged?

**Answer:** Yes, it can. As long as there is one member left in the CURRENT group, LGWR can work.

To save space in your course practice environment, drop the redo log file members you created in step 4.

Determine which redo log group is current. You cannot drop a member of the current group.

Drop the member in the previous group and then perform a log switch. In this example, group 3 is current, so the command in this example drops a member in group 2.

Drop the member in the next group and then perform a log switch.

Drop the member in the final group and then perform a log switch.

Verify that each group now has only one member.

Exit from SQL\*Plus.

Remove the physical files from the operating system.

Verify that the redo log files have been removed.

Practice 2-4: Configuring ARCHIVELOG Mode

Overview

In this practice, you configure your database for ARCHIVELOG mode so that redo logs are archived.

Assumptions

You are logged in as the oracle user.

Time Estimate

It should take approximately 10 minutes to complete this practice.

Tasks

Log in to SQL\*Plus as the SYS user with the SYSDBA privilege.

Issue the ARCHIVE LOG LIST command to determine whether the database is in

ARCHIVELOG mode.

You must put the database in MOUNT mode to enable archiving. Shut down the database instance, start it in MOUNT mode, and then enable archiving.

Shut down the database instance.

Start the database instance and mount the database.

Enable archiving.

Verify that the database is now in ARCHIVELOG mode.

Open the database.

Exit from SQL\*Plus.