Capgemini

Oracle 11g DBA Fundamentals Overview

Lesson 05: Managing the Redo Log

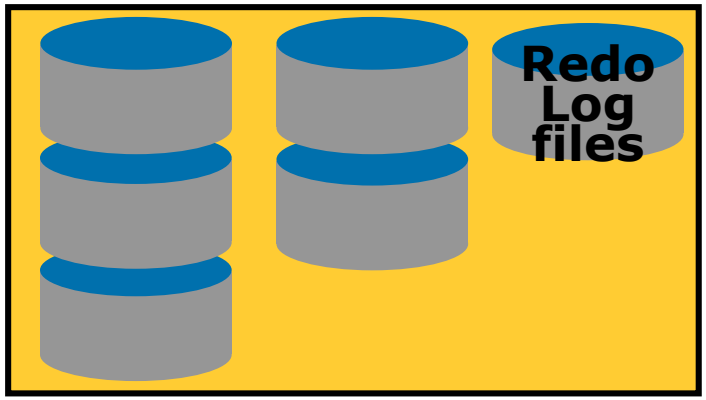
Lesson Objectives

After completing this lesson, you should be able to do the following:

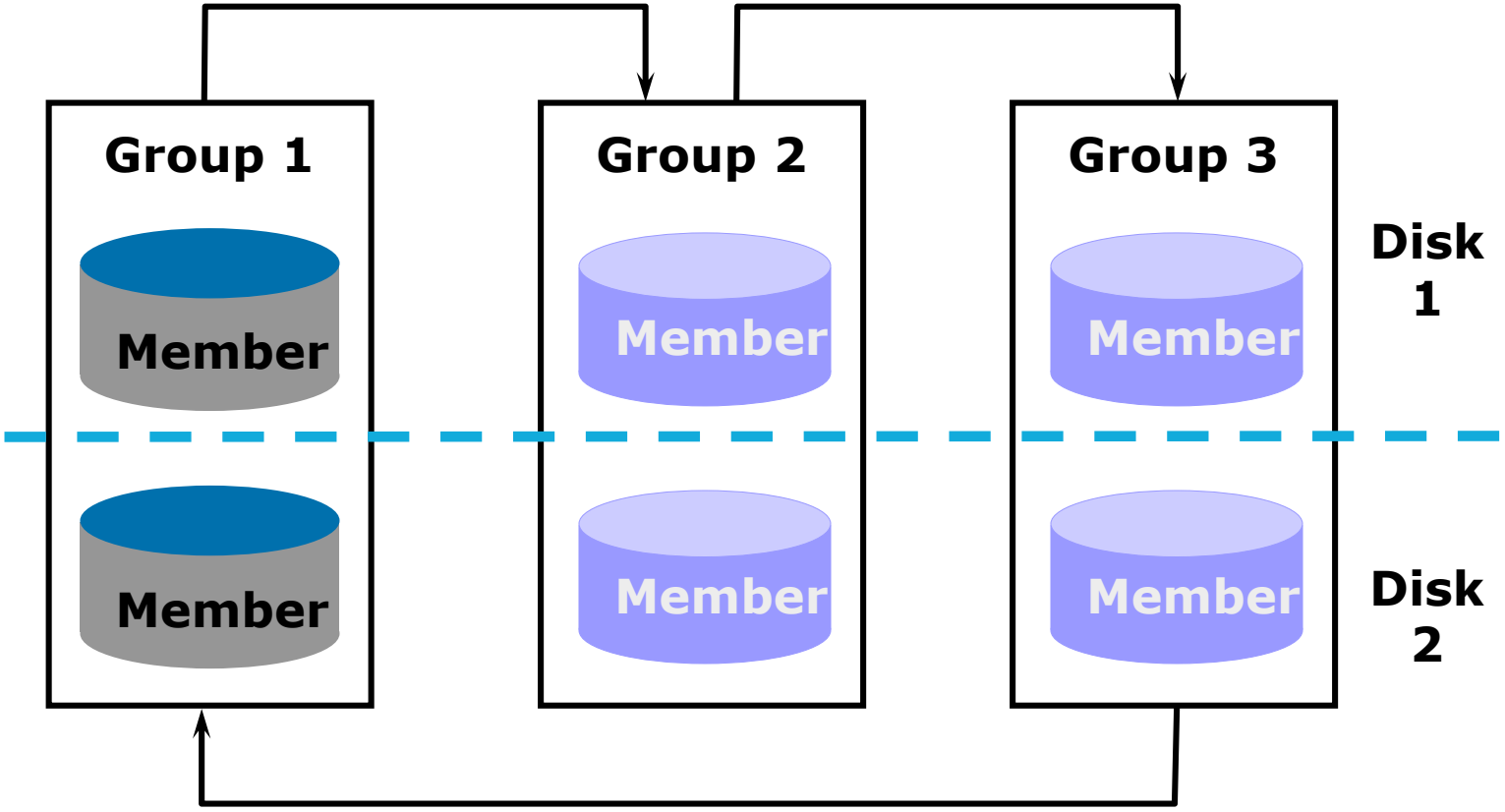
- Explain the purpose of online redo log files
- Outline the structure of online redo log files
- Control log switches and checkpoints
- Multiplex and maintain online redo log files
- Manage online redo logs files with OMF
- What Is the Archived Redo Log?
- Choosing Between NOARCHIVELOG and ARCHIVELOG Mode

Using Redo Log Files

- Redo log files have the following characteristics:
  - Record all changes made to data
  - Provide a recovery mechanism
  - Can be organized into groups
  - At least two groups required



Structure of Redo Log Files



How Redo Log Files Work

- Redo log files are used in a cyclic fashion.
- When a redo log file is full, LGWR will move to the next log group.
  - Called a log switch
  - Checkpoint operation also occurs
  - Information written to the control file

Forcing Log Switches and Checkpoints

- Forcing a log switch:

```
ALTER SYSTEM SWITCH LOGFILE;
```

- Checkpoints can be forced by using:
  - Setting FAST_START_MTTR_TARGET parameter

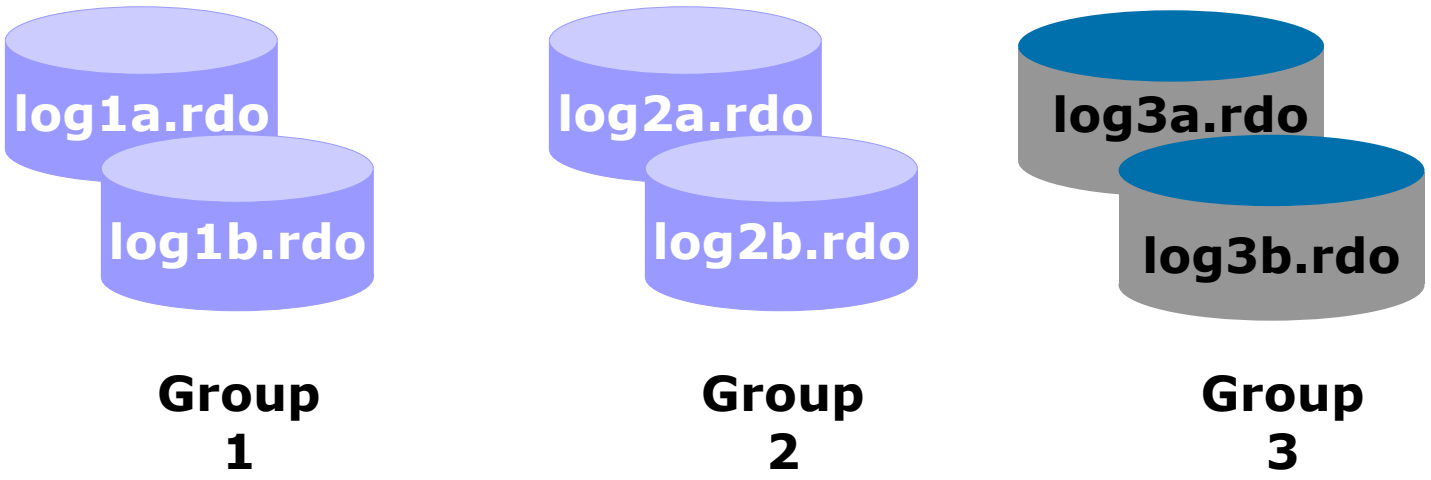
```
FAST_START_MTTR_TARGET = 600
```

  - ALTER SYSTEM CHECKPOINT command

```
ALTER SYSTEM CHECKPOINT;
```

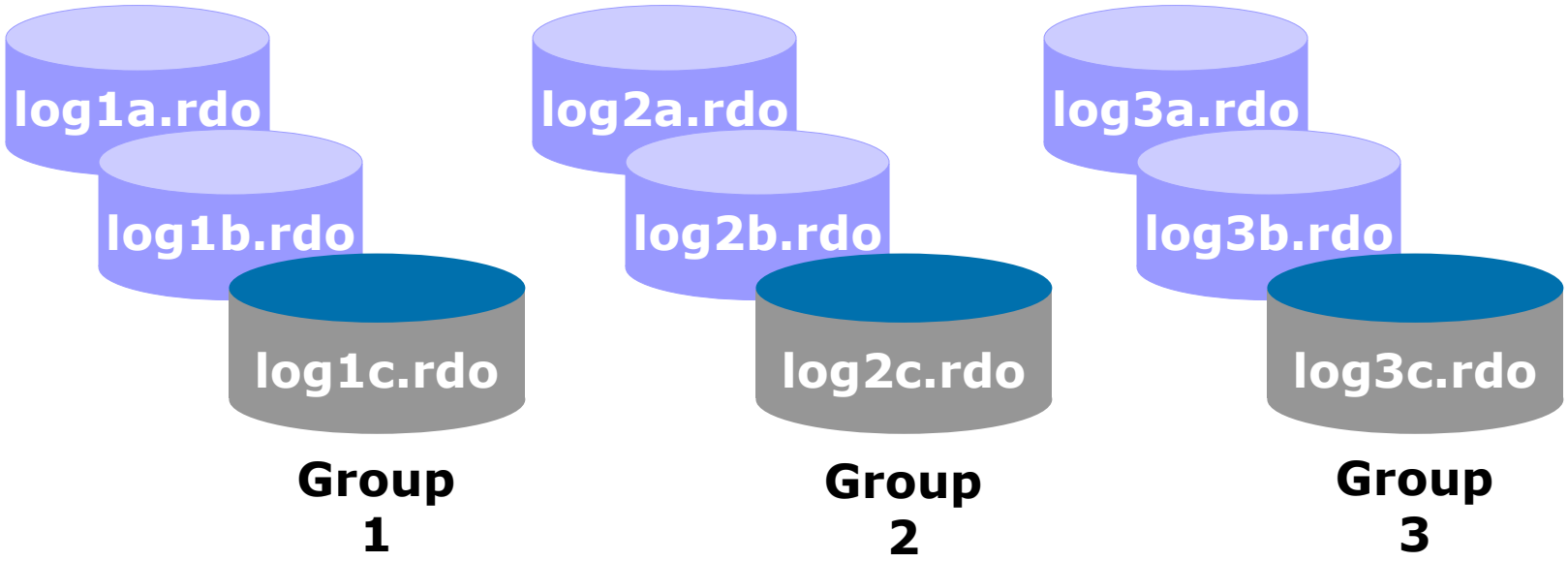
Adding Online Redo Log File Groups

```
ALTER DATABASE ADD LOGFILE GROUP 3
('$HOME/ORADATA/u01/log3a.rdo',
 '$HOME/ORADATA/u02/log3b.rdo')
SIZE 1M;
```



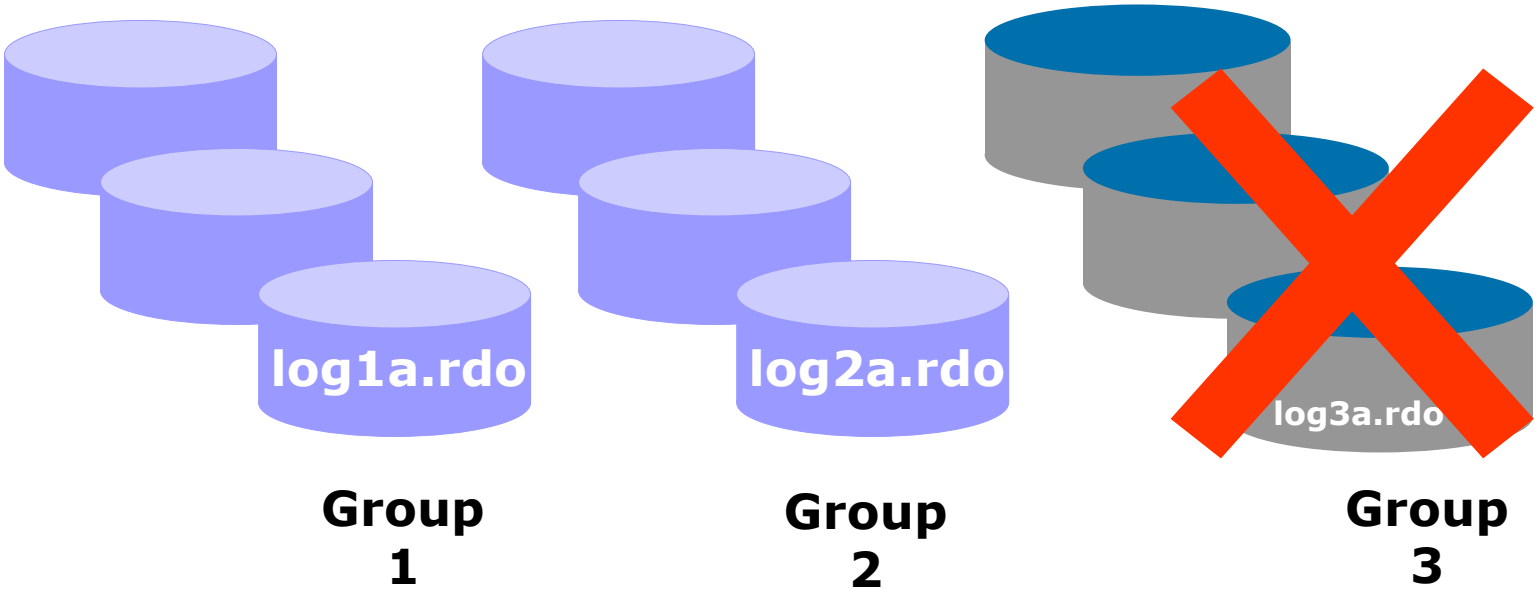
Adding Online Redo Log File Members

```
ALTER DATABASE ADD LOGFILE MEMBER
'$HOME/ORADATA/u04/log1c.rdo' TO GROUP 1,
'$HOME/ORADATA/u04/log2c.rdo' TO GROUP 2,
'$HOME/ORADATA/u04/log3c.rdo' TO GROUP 3;
```



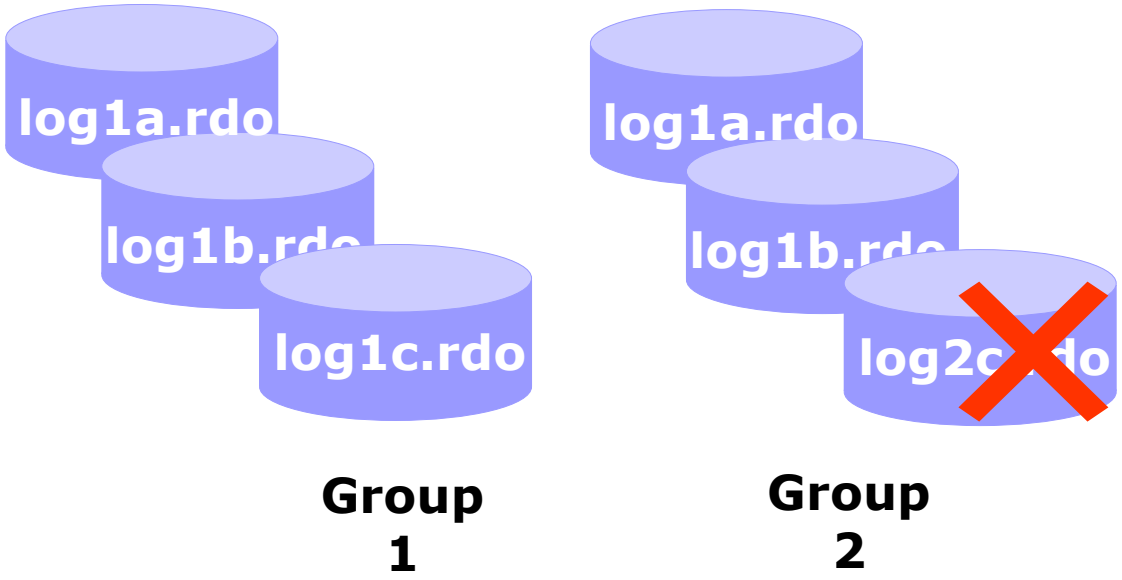
Dropping Online Redo Log File Groups

ALTER DATABASE DROP LOGFILE GROUP 3;



Dropping Online Redo Log File Members

```
ALTER DATABASE DROP LOGFILE MEMBER
'$HOME/ORADATA/u04/log3c.rdo';
```



Relocating or Renaming Online Redo Log Files

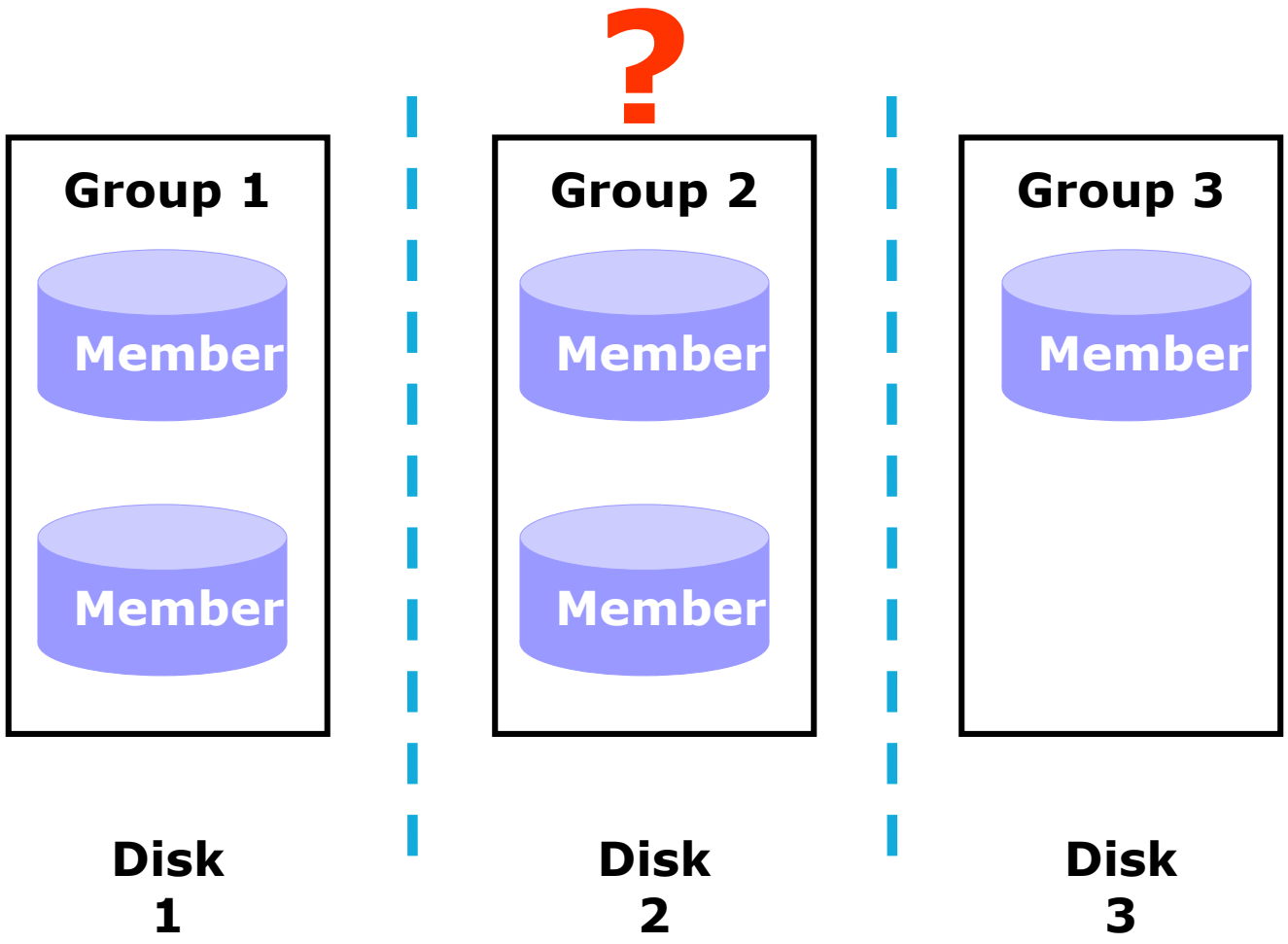
Relocate or rename online redo log files in one of the two following ways:

- ALTER DATABASE CLEAR LOGFILE command
  - Copy the online redo log files to the new location
  - Execute the command

```
ALTER DATABASE CLEAR LOGFILE
'$HOME/ORADATA/u01/log2a.rdo';
```

- Add new members and drop old members

Online Redo Log File Configuration



Managing Online Redo Log Files with OMF

- Define the DB_CREATE_ONLINE_LOG_DEST_n parameter:

```
DB_CREATE_ONLINE_LOG_DEST_1
DB_CREATE_ONLINE_LOG_DEST_2
```

- Group can be added with no file specification:

```
ALTER DATABASE ADD LOGFILE;
```

- Dropping a group:

```
ALTER DATABASE DROP LOGFILE GROUP 3;
```

Obtaining Group and Member Information

Information about a group and its members can be obtained by querying the following views:

- V$LOG
- V$LOGFILE

What Is the Archived Redo Log?

Oracle Database lets you save filled groups of redo log files to one or more offline destinations, known collectively as the archived redo log, or more simply the archive log.

The process of turning redo log files into archived redo log files is called archiving.

This process is only possible if the database is running in ARCHIVELOG mode.

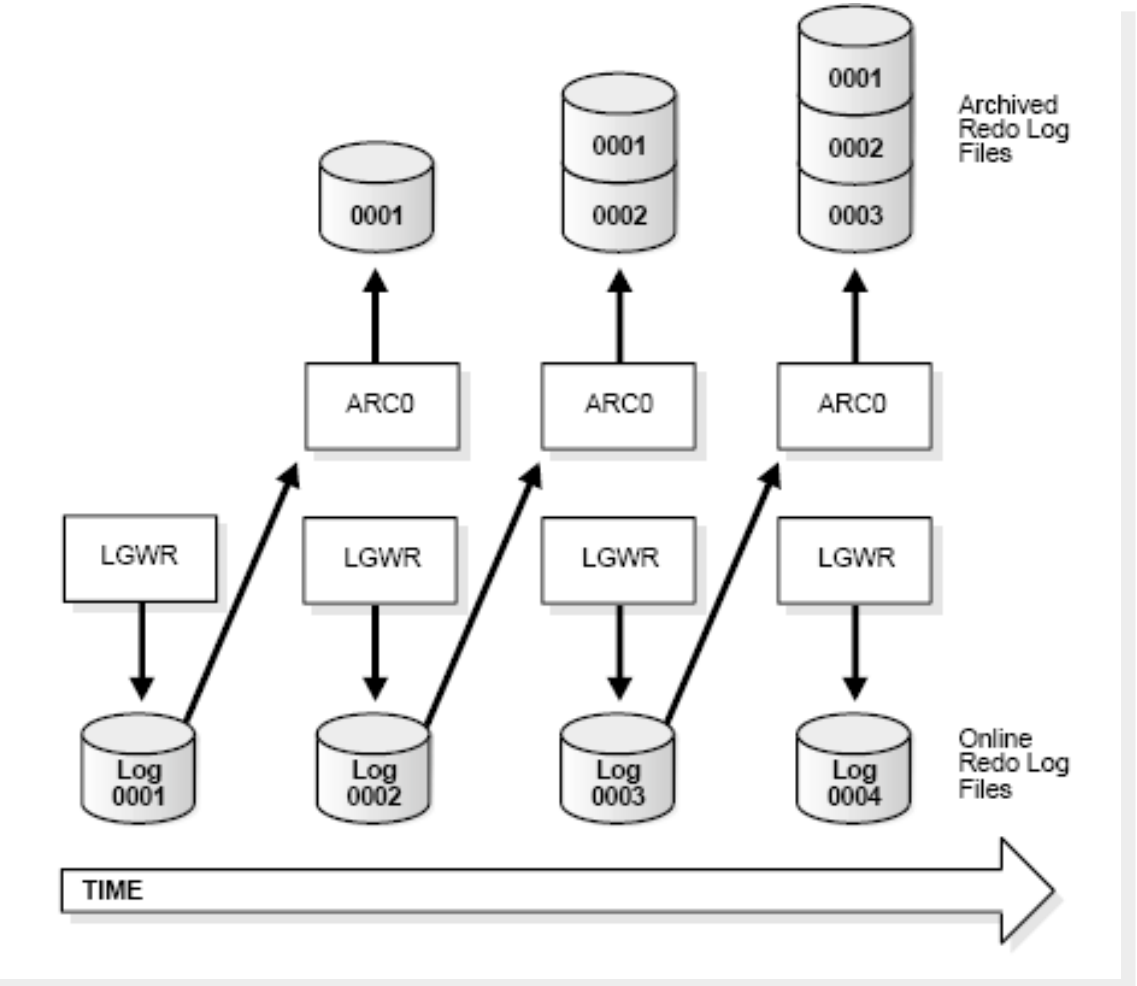
You can choose automatic or manual archiving.

Choosing Between NOARCHIVELOG and ARCHIVELOG Mode

Running a Database in NOARCHIVELOG Mode

- When you run your database in NOARCHIVELOG mode, you disable the archiving of the redo log.
- The database control file indicates that filled groups are not required to be archived.
- Therefore, when a filled group becomes inactive after a log switch, the group is available for reuse by LGWR.
- When you run a database in ARCHIVELOG mode, you enable the archiving of the redo log.
- The database control file indicates that a group of filled redo log files cannot be reused by LGWR until the group is archived.
- A filled group becomes available for archiving immediately after a redo log switch occurs.

Redo Log File Use in ARCHIVELOG Mode



Contd…Archived Redo Log Files

- Filled online redo log files can be archived.
- There are two advantages in running the database in ARCHIVELOG mode and archiving redo log files:
  - Recovery: A database backup together with online and archived redo log files can guarantee recovery of all committed transactions.
  - Backup: This can be performed while the database is open.
- By default, database is created in NOARCHIVELOG mode.

Contd…Archived Redo Log Files

- Accomplished automatically by ARCn
- Accomplished manually through SQL statements
- When successfully archived:
  - An entry in the control file is made
  - Records: archive log name, log sequence number, and high and low system change number (SCN)
  - Filled redo log file cannot be reused until:
  - A checkpoint has taken place
  - File has been archived by ARCn
- Can be multiplexed
- Maintained by the DBA

Summary

In this lesson, you should have learned how to:

- Explain the use of online redo log files
- Obtain redo log file information
- Control log switches and checkpoints
- Multiplex and maintain online redo log files
- Manage online redo log files with OMF
- What Is the Archived Redo Log?
- Choosing Between NOARCHIVELOG and ARCHIVELOG Mode

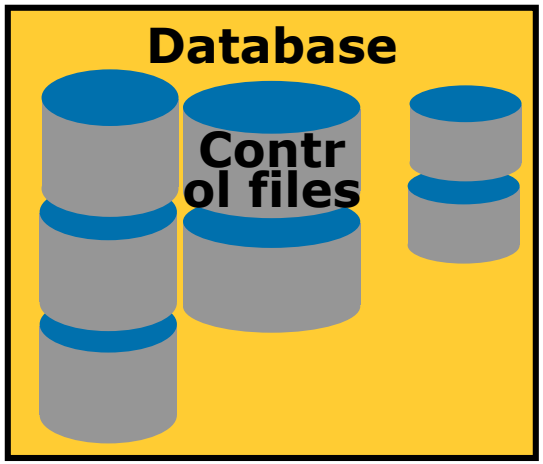
Lesson Objectives

After completing this lesson, you should be able to do the following:

- Explain the uses of the control file
- List the contents of the control file
- Multiplex and manage the control file
- Obtain control file information

Control File

- A small binary file
- Defines current state of physical database
- Maintains integrity of database
- Required:
  - At MOUNT state during database startup
  - To operate the database
- Linked to a single database
- Loss may require recovery
- Sized initially by CREATE DATABASE



Control File Contents

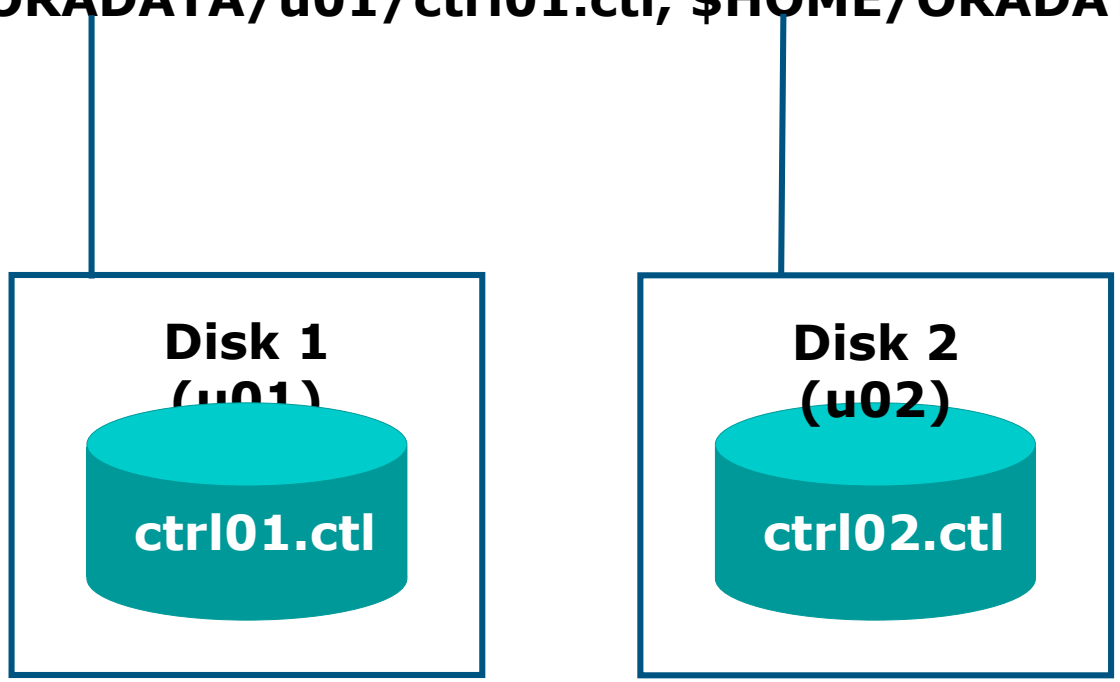
A control file contains the following entries:

- Database name and identifier
- Time stamp of database creation
- Tablespace names
- Names and locations of datafiles and redo log files
- Current redo log file sequence number
- Checkpoint information
- Begin and end of undo segments
- Redo log archive information
- Backup information

Multiplexing the Control File

CONTROL_FILES=

$HOME/ORADATA/u01/ctrl01.ctl, $HOME/ORADATA/u02/ctrl02.ct



Multiplexing the Control File When Using SPFILE

1. Alter the SPFILE:

```
ALTER SYSTEM SET control_files =
'$HOME/ORADATA/u01/ctrl01.ctl',
'$HOME/ORADATA/u02/ctrl02.ctl' SCOPE=SPFILE;
```

2. Shutdown the database:

```
shutdown immediate
```

3. Create additional control files:

```
cp $HOME/ORADATA/u01/ctrl01.ctl
$HOME/ORADATA/u02/ctrl02.ctl
```

4. Start the database:

```
startup
```

Multiplexing the Control File When Using PFILE

1. Shut down the database:

```
shutdown immediate
```

2. Create additional control files:

```
cp $HOME/ORADATA/u01/ctrl01.ctl
$HOME/ORADATA/u02/ctrl02.ctl
```

3. Add control file names to PFILE:

```
CONTROL_FILES = (/DISK1/control01.ctl,
                 /DISK3/control02.ctl)
```

4. Start the database:

```
startup
```

Obtaining Control File Information

Information about control file status and locations can be retrieved by querying the following views.

- V$CONTROLFILE: Lists the name and status of all control files associated with the instance
- V$PARAMETER: Lists status and location of all parameters
- V$CONTROLFILE_RECORD_SECTION: Provides information about the control file record sections
- SHOW PARAMETER CONTROL_FILES: Lists the name, status, and location of the control files

SUMMARY

- In this lesson, you should have learned how to:
  - Multiplex the control file when using an SPFILE
  - Multiplex the control file when using an init.ora