

What is Redolog?

In addition to datafiles for storing database data, every Oracle database has a set of (two or more) redo log files. The redo log records all changes made to data, including both uncommitted and committed changes. Changes are saved to redo (by LGWR process) before being permanently written to the datafiles.

Online redo logs should not be backed up.

What is Archivelog?

Archived log files are redo logs that Oracle has filled with redo entries, rendered inactive, and copied to one or more log archive destinations. Oracle can be run in either of two modes:

- ARCHIVELOG - Oracle archives the filled online redo log files before reusing them in the cycle.
- NOARCHIVELOG - Oracle does not archive the filled online redo log files before reusing them in the cycle.

Running the database in ARCHIVELOG mode has the following benefits:

- The database can be completely recovered from both instance and media failure.
- The user can perform online backups, i.e., back up tablespaces while the database is open and available for use.
- Archived redo logs can be transmitted and applied to the standby database
- Oracle supports multiplexed archive logs to avoid any possible single point of failure on the archive logs.
- The user has more recovery options, such as the ability to perform tablespace-point-in-time recovery (TSPITR)

Running the database in NOARCHIVELOG mode has the following consequences:

- The user can only back up the database while it is completely closed after a clean shutdown.
- Typically, the only media recovery option is to restore the whole database, which causes the loss of all transactions issued since the last backup.

Archived redo logs should be backed up and deleted regularly.

Dynamic views to support online and offline (archivelogs) redologs

Complete Listing can be found from dba_objects. The query below will you some idea:

```
set linesize 160
set pagesize 120
col object_name format a20
Select
      object_name,
      object_type
from
      dba_objects
where
      (lower(object_name) like 'v%log%' or lower(object_name) like 'gv%log%')
and
```

```
object_type in ('TABLE','VIEW') order by 1;
```

- gv\$log_history
- gv\$log_hist
- v\$log
- v\$logfile
- [v\\$thread](#)
- gv\$ARCHIVED_LOG
- gv\$ARCHIVE_DEST_STATUS

SQLs

1. How to find the size of the redo log in the current instance?

```
select
  n.name, t.value
from
  v$mystat t join
  v$statname n
on
  t.statistic# = n.statistic#
where
  n.name = 'redo size';
```

2. How to find the Redo Log location (additional information included)?

```
set linesize 200
set pagesize 200
column member format a70
Column group# format 99999
Column status format a8
column Size_MB format 999,999
column "Archived?" format a9
Column SEQ# format 999999
break on group# skip 1
select
  a.group#,
  a.member,
  b.bytes/1024/1024 as Size_MB,
  b.status,
  b.archived as "Archived?",
  b.Sequence# as SEQ#,
  to_char(b.first_time, 'Mon-DD-YY HH24:Mi:SS') as Log_Switch_Time
from
  v$logfile a,
  v$log b
where a.group#=b.group#
order by 6,2;
```

	GROUP#	MEMBER	SIZE_IN_MB	STATUS	Is it Archived?	SEQUENCE#	LOG_SWITCH_TIME
▶	4	+DG1/ractest/onlineolog/group_4.263.686843033	100	INACTIVE	YES	168	Nov-23-09 11:15:23
	4	+DG2/ractest/onlineolog/group_4.260.686843035	100	INACTIVE	YES	168	Nov-23-09 11:15:23
	3	+DG1/ractest/onlineolog/group_3.267.686843029	100	CURRENT	NO	169	Nov-24-09 03:10:26
	3	+DG2/ractest/onlineolog/group_3.259.686843031	100	CURRENT	NO	169	Nov-24-09 03:10:26
	2	+DG2/ractest/onlineolog/group_2.258.686842919	100	INACTIVE	YES	235	Nov-23-09 11:14:40
	2	+DG1/ractest/onlineolog/group_2.262.686842915	100	INACTIVE	YES	235	Nov-23-09 11:14:40
	1	+DG1/ractest/onlineolog/group_1.261.686842911	100	CURRENT	NO	236	Nov-24-09 03:10:17
	1	+DG2/ractest/onlineolog/group_1.257.686842913	100	CURRENT	NO	236	Nov-24-09 03:10:17

3. How to Check the Log Archive Times?

```
set pagesize 200
set numformat 9999999999
select
  thread#,
  Sequence#,
```

```

First_Change#,
next_change#,
to_char(first_time, 'Mon-DD-YY HH24:Mi:SS') as Log_Switch_Time
from
v$log_history
order by thread#, sequence#;

```

	THREAD#	SEQUENCE#	FIRST_CHANGE#	NEXT_CHANGE#	LOG_SWITCH_TIME
▶	1	233	27328305	27493615	Nov-22-09 06:59:54
	1	234	27493615	27494813	Nov-23-09 11:13:28
	1	235	27494813	27628602	Nov-23-09 11:14:40
	2	54	11811828	12366462	Jun-29-09 23:15:03
	2	55	12366462	12373800	Jul-03-09 06:58:32
	2	56	12373800	12467784	Jul-03-09 06:59:00

4. How to check all the log switches during a time interval?

For example, if we want to check the all the log switch times between 5 and 6 PM on Nov 21' 2009, the following SQL can be executed:

```

select
thread#,
Sequence#,
First_Change#,
next_change#,
to_char(first_time, 'Mon-DD-YY HH24:Mi:SS') as Log_Switch_Time
from
v$log_history
where
first_time between to_date('NOV 21 2009 17:00:00','Mon DD YYYY HH24:MI:SS')
and
to_date('NOV 21 2009 18:00:00','Mon DD YYYY HH24:MI:SS')
order by 2;

```

	THREAD#	SEQUENCE#	FIRST_CHANGE#	NEXT_CHANGE#	LOG_SWITCH_TIME
	1	40488	20757705257	20757933949	Nov-21-09 17:00:02
	1	40491	20758150904	20758261071	Nov-21-09 17:23:51
	1	40489	20757933949	20757978211	Nov-21-09 17:12:41
	2	39843	20758091162	20758150541	Nov-21-09 17:20:44
	2	39844	20758150541	20758210510	Nov-21-09 17:23:49
	2	39845	20758210510	20758261082	Nov-21-09 17:26:57

5. **How to check the current sequence number?**

SELECT THREAD#, SEQUENCE#, ARCHIVED, STATUS FROM V\$LOG where STATUS='CURRENT';

6. **How to switch the logfile?**

ALTER SYSTEM SWITCH LOGFILE; (to switch the logfile in a standalone database)

ALTER SYSTEM ARCHIVE LOG CURRENT; (to switch the logfile in all threads of the cluster)

7. **Verify new archived redo log files were applied:**

SELECT SEQUENCE#, FIRST_TIME, NEXT_TIME FROM V\$ARCHIVED_LOG ORDER BY SEQUENCE#;

SELECT SEQUENCE#, FIRST_TIME, NEXT_TIME FROM V\$ARCHIVED_LOG order by first_time;

8. **How to determine the status of redo log files?**

SELECT THREAD#, SEQUENCE#, ARCHIVED, STATUS FROM V\$LOG;

9. **How to determine the most recent archived redo log file?**

SELECT MAX(SEQUENCE#), THREAD# FROM V\$ARCHIVED_LOG GROUP BY THREAD#;

10. **How to determine the most recent archived redo log file at each destination?**

```
col destination format a30
set linesize 200
SELECT
  DESTINATION,
  STATUS,
  ARCHIVED_THREAD#,
  ARCHIVED_SEQ#
FROM
  gv$ARCHIVE_DEST_STATUS
WHERE
  STATUS <> 'DEFERRED'
AND
  STATUS <> 'INACTIVE';
```

	DESTINATION	STATUS	ARCHIVED_THREAD#	ARCHIVED_SEQ#
▶	+DG2/ractest/	VALID	2	168
	/backup/ractest/Archive	VALID	2	168
	+DG2/ractest/	VALID	1	235
	/backup/ractest/Archive	VALID	1	235

11. **How to check the active archivelogs destination that are in use and verify the status is valid?**

col dest_name format a30

col destination format a60

set linesize 200

SELECT inst_id, dest_name, destination, status, schedule FROM gv\$ARCHIVE_DEST where schedule='ACTIVE';

	INST_ID	DEST_NAME	DESTINATION	STATUS	SCHEDULE
▶	2	LOG_ARCHIVE_DEST_1	+DG2/ractest/	VALID	ACTIVE
	2	LOG_ARCHIVE_DEST_2	/backup/ractest/Archive	VALID	ACTIVE
	1	LOG_ARCHIVE_DEST_1	+DG2/ractest/	VALID	ACTIVE
	1	LOG_ARCHIVE_DEST_2	/backup/ractest/Archive	VALID	ACTIVE

12. **How to check how many switches occur per day in last 100 days?**

set pagesize 200

Select

to_char(trunc(first_time), 'Month') Month,

to_char(trunc(first_time), 'Mon-DD-YYYY :DY') Date_n_Day,

count(*) Total_logswitches

from

v\$log_history

where

trunc(first_time) > last_day(sysdate-100) +1

group by

trunc(first_time)

order by 2 desc;

	MONTH	DATE_N_DAY	TOTAL_LOGSWITCHES
▶	September	Sep-30-2009 :WED	2
	September	Sep-29-2009 :TUE	1
	September	Sep-28-2009 :MON	3
	September	Sep-26-2009 :SAT	1
	September	Sep-25-2009 :FRI	1
	September	Sep-24-2009 :THU	2
	September	Sep-23-2009 :WED	1
	September	Sep-22-2009 :TUE	1

13. How to check each log switch duration?

```

select a.sequence#,
       to_char(b.first_time, 'Mon-DD-YYYY :DY') start_time,
       to_char(a.first_time, 'Mon-DD-YYYY :DY') end_time,
       ('Log switch occurred after '|| (round(((a.first_time - b.first_time)*25)*60,0 )
       || ' minutes or after ')
       || (round(((a.first_time - b.first_time)*25),2 )
       || ' hours')) as Each_Log_switch_duration
from
     v$log_history a,
     v$log_history b
where
     a.recid=b.recid+1
order by
     a.first_time desc;

```

	SEQUENCE#	START_TIME	END_TIME	EACH_LOG_SWITCH_DURATION
▶	169	Nov-23-2009 :MON	Nov-24-2009 :TUE	Log switch occurred after 996 minutes or after 16.59 hours
	168	Nov-23-2009 :MON	Nov-23-2009 :MON	Log switch occurred after 0 minutes or after .01 hours
	167	Nov-23-2009 :MON	Nov-23-2009 :MON	Log switch occurred after 0 minutes or after 0 hours
	166	Nov-23-2009 :MON	Nov-23-2009 :MON	Log switch occurred after 1 minutes or after .02 hours
	235	Nov-23-2009 :MON	Nov-23-2009 :MON	Log switch occurred after -1 minutes or after -.01 hours

14. How to check how much redo is generating per day?

```

SELECT A.*,
       Round(A.Count#*B.AVG#) Daily_archived_Mb, round(A.count#*C.AVG#/1024/1024) daily_redo_space

```

```

FROM
  (SELECT
    To_Char(First_Time,'YYYY-MM-DD  :Dy') DAY,
    Count(1) Count#,
    Min(RECID) Min#,
    Max(RECID) Max#
  FROM
    v$log_history
  GROUP BY
    To_Char(First_Time,'YYYY-MM-DD  :Dy')
  ORDER BY 1 DESC) A,
  (SELECT
    avg(round(block_size * blocks/1024/1024,2)) avg#
  FROM
    v$archived_log) B,
  (SELECT
    Avg(BYTES) AVG#
  from
    v$log) C;

```

	DAY	COUNT#	MIN#	MAX#	DAILY_ARCHIVED_MB	DAILY_REDO_SPACE
▶	2009-11-24 :Tue	1	404	404	81	100
	2009-11-23 :Mon	9	394	403	729	900
	2009-11-22 :Sun	2	393	395	162	200
	2009-11-20 :Fri	6	387	392	486	600
	2009-11-19 :Thu	1	386	386	81	100
	2009-11-18 :Wed	1	384	384	81	100
	2009-11-17 :Tue	37	347	385	2998	3700
	2009-11-16 :Mon	1	349	349	81	100

15. How much Redo is generated in last 10 days?

```

select 'Total Redo Generated in last 10 days: '|| round(sum(daily_archived_mb)/1024,2)||' GB' as Total
FROM
  ( SELECT
    A.*,
    Round(A.Count#*B.AVG#) Daily_archived_Mb,
    round(A.count#*C.AVG#/1024/1024) daily_redo_space
  FROM
    ( SELECT
      To_Char(First_Time,'YYYY-MM-DD  :Dy') DAY_of_the_month,

```

```

        Count(1) Count#,
        Min(RECID) Min#,
        Max(RECID) Max#
    FROM
        v$log_history
    GROUP BY
        To_Char(First_Time,'YYYY-MM-DD :Dy')
    ORDER BY 1 DESC
) A,
( SELECT
    avg(round(block_size * blocks/1024/1024,2)) avg#
    FROM
        v$sarchived_log
) B,
( SELECT
    Avg(BYTES) AVG# from
    v$log
) C
where rownum <= 10
);

```

	TOTAL
	Total Redo Generated in last 10 days: 4.83 GB

16. Count the log switches occurring every hour of the day? [Good when running in TOAD]

A) From TOAD

```

select thread# as Instance#, to_char(FIRST_TIME,'DY, DD-MON-YYYY') day,
decode(sum(decode(substr(to_char(FIRST_TIME,'HH24'),1,2),'00',1,0)),0,'-',sum(decode(substr(to_char(FIRST_TIME,'HH24'),1,2),'00',1,0)))
H0,
decode(sum(decode(substr(to_char(FIRST_TIME,'HH24'),1,2),'01',1,0)),0,'-',sum(decode(substr(to_char(FIRST_TIME,'HH24'),1,2),'01',1,0)))
H1,
decode(sum(decode(substr(to_char(FIRST_TIME,'HH24'),1,2),'02',1,0)),0,'-',sum(decode(substr(to_char(FIRST_TIME,'HH24'),1,2),'02',1,0)))
H2,
decode(sum(decode(substr(to_char(FIRST_TIME,'HH24'),1,2),'03',1,0)),0,'-',sum(decode(substr(to_char(FIRST_TIME,'HH24'),1,2),'03',1,0)))
H3,
decode(sum(decode(substr(to_char(FIRST_TIME,'HH24'),1,2),'04',1,0)),0,'-',sum(decode(substr(to_char(FIRST_TIME,'HH24'),1,2),'04',1,0)))
H4,
decode(sum(decode(substr(to_char(FIRST_TIME,'HH24'),1,2),'05',1,0)),0,'-',sum(decode(substr(to_char(FIRST_TIME,'HH24'),1,2),'05',1,0)))
H5,
decode(sum(decode(substr(to_char(FIRST_TIME,'HH24'),1,2),'06',1,0)),0,'-',sum(decode(substr(to_char(FIRST_TIME,'HH24'),1,2),'06',1,0)))
H6,

```

```

decode(sum(decode(substr(to_char(FIRST_TIME,'HH24'),1,2),'07',1,0)),0,'-',sum(decode(substr(to_char(FIRST_TIME,'HH24'),1,2),'07',1,0)))
H7,
decode(sum(decode(substr(to_char(FIRST_TIME,'HH24'),1,2),'08',1,0)),0,'-',sum(decode(substr(to_char(FIRST_TIME,'HH24'),1,2),'08',1,0)))
H8,
decode(sum(decode(substr(to_char(FIRST_TIME,'HH24'),1,2),'09',1,0)),0,'-',sum(decode(substr(to_char(FIRST_TIME,'HH24'),1,2),'09',1,0)))
H9,
decode(sum(decode(substr(to_char(FIRST_TIME,'HH24'),1,2),'10',1,0)),0,'-',sum(decode(substr(to_char(FIRST_TIME,'HH24'),1,2),'10',1,0)))
H10,
decode(sum(decode(substr(to_char(FIRST_TIME,'HH24'),1,2),'11',1,0)),0,'-',sum(decode(substr(to_char(FIRST_TIME,'HH24'),1,2),'11',1,0)))
H11,
decode(sum(decode(substr(to_char(FIRST_TIME,'HH24'),1,2),'12',1,0)),0,'-',sum(decode(substr(to_char(FIRST_TIME,'HH24'),1,2),'12',1,0)))
H12,
decode(sum(decode(substr(to_char(FIRST_TIME,'HH24'),1,2),'13',1,0)),0,'-',sum(decode(substr(to_char(FIRST_TIME,'HH24'),1,2),'13',1,0)))
H13,
decode(sum(decode(substr(to_char(FIRST_TIME,'HH24'),1,2),'14',1,0)),0,'-',sum(decode(substr(to_char(FIRST_TIME,'HH24'),1,2),'14',1,0)))
H14,
decode(sum(decode(substr(to_char(FIRST_TIME,'HH24'),1,2),'15',1,0)),0,'-',sum(decode(substr(to_char(FIRST_TIME,'HH24'),1,2),'15',1,0)))
H15,
decode(sum(decode(substr(to_char(FIRST_TIME,'HH24'),1,2),'16',1,0)),0,'-',sum(decode(substr(to_char(FIRST_TIME,'HH24'),1,2),'16',1,0)))
H16,
decode(sum(decode(substr(to_char(FIRST_TIME,'HH24'),1,2),'17',1,0)),0,'-',sum(decode(substr(to_char(FIRST_TIME,'HH24'),1,2),'17',1,0)))
H17,
decode(sum(decode(substr(to_char(FIRST_TIME,'HH24'),1,2),'18',1,0)),0,'-',sum(decode(substr(to_char(FIRST_TIME,'HH24'),1,2),'18',1,0)))
H18,
decode(sum(decode(substr(to_char(FIRST_TIME,'HH24'),1,2),'19',1,0)),0,'-',sum(decode(substr(to_char(FIRST_TIME,'HH24'),1,2),'19',1,0)))
H19,
decode(sum(decode(substr(to_char(FIRST_TIME,'HH24'),1,2),'20',1,0)),0,'-',sum(decode(substr(to_char(FIRST_TIME,'HH24'),1,2),'20',1,0)))
H20,
decode(sum(decode(substr(to_char(FIRST_TIME,'HH24'),1,2),'21',1,0)),0,'-',sum(decode(substr(to_char(FIRST_TIME,'HH24'),1,2),'21',1,0)))
H21,
decode(sum(decode(substr(to_char(FIRST_TIME,'HH24'),1,2),'22',1,0)),0,'-',sum(decode(substr(to_char(FIRST_TIME,'HH24'),1,2),'22',1,0)))
H22,
decode(sum(decode(substr(to_char(FIRST_TIME,'HH24'),1,2),'23',1,0)),0,'-',sum(decode(substr(to_char(FIRST_TIME,'HH24'),1,2),'23',1,0)))
H23,
count(trunc(FIRST_TIME)) Total
from v$log_history
group by thread#,to_char(FIRST_TIME,'DY, DD-MON-YYYY')
order by thread#,to_date(substr(to_char(FIRST_TIME,'DY, DD-MON-YYYY'),5,15) );

```

17.

INST...	DAY	H0	H1	H2	H3	H4	H5	H6	H7	H8	H9	H10	H11	H12	H13	H14	H15	H16	H17	H18	H19	H20	H21	H22	H23	TOTAL
1	THU, 19-NOV-2009	4	8	4	7	4	4	4	4	4	4	4	4	4	4	4	4	4	11	4	4	4	4	4	4	110
1	FRI, 20-NOV-2009	5	6	4	5	4	4	4	4	4	4	4	4	4	4	4	4	4	6	4	4	4	4	4	4	102
1	SAT, 21-NOV-2009	5	5	-	3	7	5	4	6	4	4	4	4	4	4	4	4	4	6	4	4	4	4	4	4	101
1	SUN, 22-NOV-2009	4	7	4	6	4	6	4	4	4	4	4	4	4	4	4	4	4	12	4	4	4	4	4	4	111
1	MON, 23-NOV-2009	4	7	4	4	4	5	4	4	4	4	4	4	4	4	4	4	4	6	4	4	4	4	4	4	102
1	TUE, 24-NOV-2009	4	8	4	6	4	4	4	4	4	4	4	4	4	4	4	4	-	-	-	-	-	-	-	-	70
2	FRI, 13-NOV-2009	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	3	4	11	4	4	4	4	4	4	42
2	SAT, 14-NOV-2009	4	9	4	7	4	4	4	4	4	4	4	4	4	4	4	4	4	11	4	4	4	4	4	4	111
2	SUN, 15-NOV-2009	4	7	4	7	4	4	4	4	4	4	4	4	4	4	4	4	4	11	4	4	4	4	4	4	109
2	MON, 16-NOV-2009	4	5	4	5	4	4	4	4	4	4	4	4	4	4	4	4	4	6	4	4	4	4	4	4	100
2	TUE, 17-NOV-2009	5	7	4	7	4	4	4	4	4	4	4	4	4	4	4	4	4	6	4	4	4	4	4	4	105

B) From Putty

```

col instance# heading Ins col 999
col day format a18
set linesize 300
set pagesize 24
col total format 9,999
col h0 format a3
col h1 format a3
col h2 format a3
col h3 format a3
col h4 format a3
col h5 format a3
col h6 format a3
col h7 format a3
col h8 format a3
col h9 format a3
col h10 format a3
col h11 format a3
col h12 format a3
col h13 format a3
col h14 format a3
col h15 format a3
col h16 format a3
col h17 format a3
col h18 format a3
col h19 format a3
col h20 format a3
col h21 format a3
col h22 format a3
col h23 format a3
select thread# as Instance#, to_char(FIRST_TIME,'DY, DD-MON-YYYY') day,
decode(sum(decode(substr(to_char(FIRST_TIME,'HH24'),1,2),'00',1,0)),0,'-',sum(decode(substr(to_char(FIRST_TIME,'HH24'),1,2),'00',1,0))) H0,
decode(sum(decode(substr(to_char(FIRST_TIME,'HH24'),1,2),'01',1,0)),0,'-',sum(decode(substr(to_char(FIRST_TIME,'HH24'),1,2),'01',1,0))) H1,
decode(sum(decode(substr(to_char(FIRST_TIME,'HH24'),1,2),'02',1,0)),0,'-',sum(decode(substr(to_char(FIRST_TIME,'HH24'),1,2),'02',1,0))) H2,

```

```

decode(sum(decode(substr(to_char(FIRST_TIME,'HH24'),1,2),'03',1,0)),0,'-',sum(decode(substr(to_char(FIRST_TIME,'HH24'),1,2),'03',1,0))) H3,
decode(sum(decode(substr(to_char(FIRST_TIME,'HH24'),1,2),'04',1,0)),0,'-',sum(decode(substr(to_char(FIRST_TIME,'HH24'),1,2),'04',1,0))) H4,
decode(sum(decode(substr(to_char(FIRST_TIME,'HH24'),1,2),'05',1,0)),0,'-',sum(decode(substr(to_char(FIRST_TIME,'HH24'),1,2),'05',1,0))) H5,
decode(sum(decode(substr(to_char(FIRST_TIME,'HH24'),1,2),'06',1,0)),0,'-',sum(decode(substr(to_char(FIRST_TIME,'HH24'),1,2),'06',1,0))) H6,
decode(sum(decode(substr(to_char(FIRST_TIME,'HH24'),1,2),'07',1,0)),0,'-',sum(decode(substr(to_char(FIRST_TIME,'HH24'),1,2),'07',1,0))) H7,
decode(sum(decode(substr(to_char(FIRST_TIME,'HH24'),1,2),'08',1,0)),0,'-',sum(decode(substr(to_char(FIRST_TIME,'HH24'),1,2),'08',1,0))) H8,
decode(sum(decode(substr(to_char(FIRST_TIME,'HH24'),1,2),'09',1,0)),0,'-',sum(decode(substr(to_char(FIRST_TIME,'HH24'),1,2),'09',1,0))) H9,
decode(sum(decode(substr(to_char(FIRST_TIME,'HH24'),1,2),'10',1,0)),0,'-',sum(decode(substr(to_char(FIRST_TIME,'HH24'),1,2),'10',1,0))) H10,
decode(sum(decode(substr(to_char(FIRST_TIME,'HH24'),1,2),'11',1,0)),0,'-',sum(decode(substr(to_char(FIRST_TIME,'HH24'),1,2),'11',1,0))) H11,
decode(sum(decode(substr(to_char(FIRST_TIME,'HH24'),1,2),'12',1,0)),0,'-',sum(decode(substr(to_char(FIRST_TIME,'HH24'),1,2),'12',1,0))) H12,
decode(sum(decode(substr(to_char(FIRST_TIME,'HH24'),1,2),'13',1,0)),0,'-',sum(decode(substr(to_char(FIRST_TIME,'HH24'),1,2),'13',1,0))) H13,
decode(sum(decode(substr(to_char(FIRST_TIME,'HH24'),1,2),'14',1,0)),0,'-',sum(decode(substr(to_char(FIRST_TIME,'HH24'),1,2),'14',1,0))) H14,
decode(sum(decode(substr(to_char(FIRST_TIME,'HH24'),1,2),'15',1,0)),0,'-',sum(decode(substr(to_char(FIRST_TIME,'HH24'),1,2),'15',1,0))) H15,
decode(sum(decode(substr(to_char(FIRST_TIME,'HH24'),1,2),'16',1,0)),0,'-',sum(decode(substr(to_char(FIRST_TIME,'HH24'),1,2),'16',1,0))) H16,
decode(sum(decode(substr(to_char(FIRST_TIME,'HH24'),1,2),'17',1,0)),0,'-',sum(decode(substr(to_char(FIRST_TIME,'HH24'),1,2),'17',1,0))) H17,
decode(sum(decode(substr(to_char(FIRST_TIME,'HH24'),1,2),'18',1,0)),0,'-',sum(decode(substr(to_char(FIRST_TIME,'HH24'),1,2),'18',1,0))) H18,
decode(sum(decode(substr(to_char(FIRST_TIME,'HH24'),1,2),'19',1,0)),0,'-',sum(decode(substr(to_char(FIRST_TIME,'HH24'),1,2),'19',1,0))) H19,
decode(sum(decode(substr(to_char(FIRST_TIME,'HH24'),1,2),'20',1,0)),0,'-',sum(decode(substr(to_char(FIRST_TIME,'HH24'),1,2),'20',1,0))) H20,
decode(sum(decode(substr(to_char(FIRST_TIME,'HH24'),1,2),'21',1,0)),0,'-',sum(decode(substr(to_char(FIRST_TIME,'HH24'),1,2),'21',1,0))) H21,
decode(sum(decode(substr(to_char(FIRST_TIME,'HH24'),1,2),'22',1,0)),0,'-',sum(decode(substr(to_char(FIRST_TIME,'HH24'),1,2),'22',1,0))) H22,
decode(sum(decode(substr(to_char(FIRST_TIME,'HH24'),1,2),'23',1,0)),0,'-',sum(decode(substr(to_char(FIRST_TIME,'HH24'),1,2),'23',1,0))) H23,
count(trunc(FIRST_TIME)) Total
from v$log_history
group by thread#,to_char(FIRST_TIME,'DY, DD-MON-YYYY')
order by thread#,to_date(substr(to_char(FIRST_TIME,'DY, DD-MON-YYYY'),5,15) );

```

18. **How to clear Redo Log File?**The following statement clears the log files in redo log group number 3:

```
ALTER DATABASE CLEAR LOGFILE GROUP 3;
```

If the corrupt redo log file has not been archived, use the UNARCHIVED keyword in the statement.

```
ALTER DATABASE CLEAR UNARCHIVED LOGFILE GROUP 3;
```

http://download.oracle.com/docs/cd/B19306_01/server.102/b14231/onlineredo.htm#i100639

19. **How to apply log switch (automatically) after every 15 minutes?**

```
ALTER SYSTEM SET ARCHIVE_LAG_TARGET = 900 SCOPE=BOTH;
```

20. **How to ADD, create or replace Redo Logs.**

<http://www.mydbanotes.com/2011/03/how-to-replace-current-oracle-redo-log.html>

20. **How to disable Oracle redo log thread?**

<http://www.mydbanotes.com/2013/12/how-to-disable-oracle-redo-log-thread.html>

21. **How to drop a redo log group?**

Task is to drop group 4. Then follow the steps below.

- Verify the group you want to delete is INACTIVE (or unused) first.
 - select group#,thread#,sequence#,members,status,first_change# from v\$log;
- Then issue the following command to drop the group.
 - **Alter database drop logfile group 4;**

22. How to add standby Redo Log Group?

STANDALONE

```
alter database add standby logfile group 4 ('+DATA(ONLINELOG)', '+FRA(ONLINELOG)') SIZE 1000M;
```

RAC:

```
alter database add standby logfile THREAD 1 group 4 ('+DATA', '+FRA') SIZE 1000M;
```

STANDALONE with multiple members:

```
alter database add standby logfile group 7  
('/database/dbredo01/oradata/newark/stbyredo07a.dbf', '/database/dbredo02/oradata/newark/stbyredo07b.dbf') size 256M;
```

23. How to drop standby Redo Log Group?

```
ALTER DATABASE DROP STANDBY LOGFILE GROUP 4;
```

24. Check the size of online Redo Logs and standby Redo logs

```
set linesize 200 pagesize 200  
ol member format a50  
select  
    lg.group#,  
    lg.type,  
    lg.member,  
    l.bytes/1024/1024 as SIZE_MB  
from  
    v$logfile lg,  
    v$log l  
where  
    lg.group#=l.group#  
union all  
select  
    lg.group#,  
    lg.type,  
    lg.member,  
    st.bytes/1024/1024 as SIZE_MB  
from
```

```

v$logfile lg,
v$standby_log st
where
lg.group#=st.group#;

```

GROUP#	TYPE	MEMBER	SIZE_MB
1	ONLINE	/u15/oradata/FAIZANPR/onlineolog/redo01a.rdo	100
1		/u16/oradata/FAIZANDG/onlineolog/redo01b.rdo	100
2		/u15/oradata/FAIZANPR/onlineolog/redo02a.rdo	100
2		/u16/oradata/FAIZANDG/onlineolog/redo02b.rdo	100
3		/u15/oradata/FAIZANPR/onlineolog/redo03a.rdo	100
3		/u16/oradata/FAIZANDG/onlineolog/redo03b.rdo	100
3		/u17/oradata/FAIZANDG/onlineolog/redo03c.rdo	100
1		/u17/oradata/FAIZANDG/onlineolog/redo01c.rdo	100
2		/u17/oradata/FAIZANDG/onlineolog/redo02c.rdo	100
4	STANDBY	/u15/oradata/FAIZANDG/onlineolog/redo04a.rdo	100
4		/u16/oradata/FAIZANDG/onlineolog/redo04b.rdo	100
4		/u17/oradata/FAIZANDG/onlineolog/redo04c.rdo	100
5		/u15/oradata/FAIZANDG/onlineolog/redo05a.rdo	100
5		/u16/oradata/FAIZANDG/onlineolog/redo05b.rdo	100
5		/u17/oradata/FAIZANDG/onlineolog/redo05c.rdo	100
6		/u15/oradata/FAIZANDG/onlineolog/redo06a.rdo	100
6		/u16/oradata/FAIZANDG/onlineolog/redo06b.rdo	100
6		/u17/oradata/FAIZANDG/onlineolog/redo06c.rdo	100
7		/u15/oradata/FAIZANDG/onlineolog/redo07a.rdo	100
7		/u16/oradata/FAIZANDG/onlineolog/redo07b.rdo	100
7		/u17/oradata/FAIZANDG/onlineolog/redo07c.rdo	100

- 25. test
- 26. test
- 27. test

Other Notes:

Please note that some of the above queries might have came from the sources. My intent is to compile all the resources under one umbrella for any DBA looking to get the redo or archive logs info from oracle DB. Hope you'll find this handy when you need them. Any suggestions or new queries are welcome.

Others:

Deassembling the redo log ==> <http://www.orafaq.com/papers/redolog.pdf>
http://www.oracle.com/technology/deploy/availability/htdocs/BR_Overview.htm#arl
<http://www.databasedesign-resource.com/oracle-log-files.html>