


# Part 1 .Administration

1. Oracle Architecture
2. Oracle SGA (System Global Area)
3. Oracle Background Processes
4. Oracle Startup & Shutdown
5. Control files
- 6. Redo log files** 
7. Tablespace & Datafiles
8. User Managemens

## 6. Redo log files



- Redo Log
  - 오라클 서버는 데이터가 변경될 경우 장애를 대비해 변경되기 전의 내용과 변경된 후의 내용을 기록해둔다
  - 기록되는 장소 중 메모리는 Redo Log Buffer
  - 기록되는 장소 중 파일은 Redo Log File

## 6. Redo log files

### 1. Redo Log 생성 원리

- Write Log Ahead
  - 데이터를 변경하기 전에 Redo Log에 먼저 기록 후 데이터를 변경( LGWR 작동 후 DBWR 작동)
- Log Force at commit
  - 사용자로부터 Commit요청이 들어오면 관련된 모든 Redo Recode들을 redo Log file에 저장한 후 Commit을 완료
  - 대량의 데이터 변경 후 Commit이 한꺼번에 수행 시 성능이슈

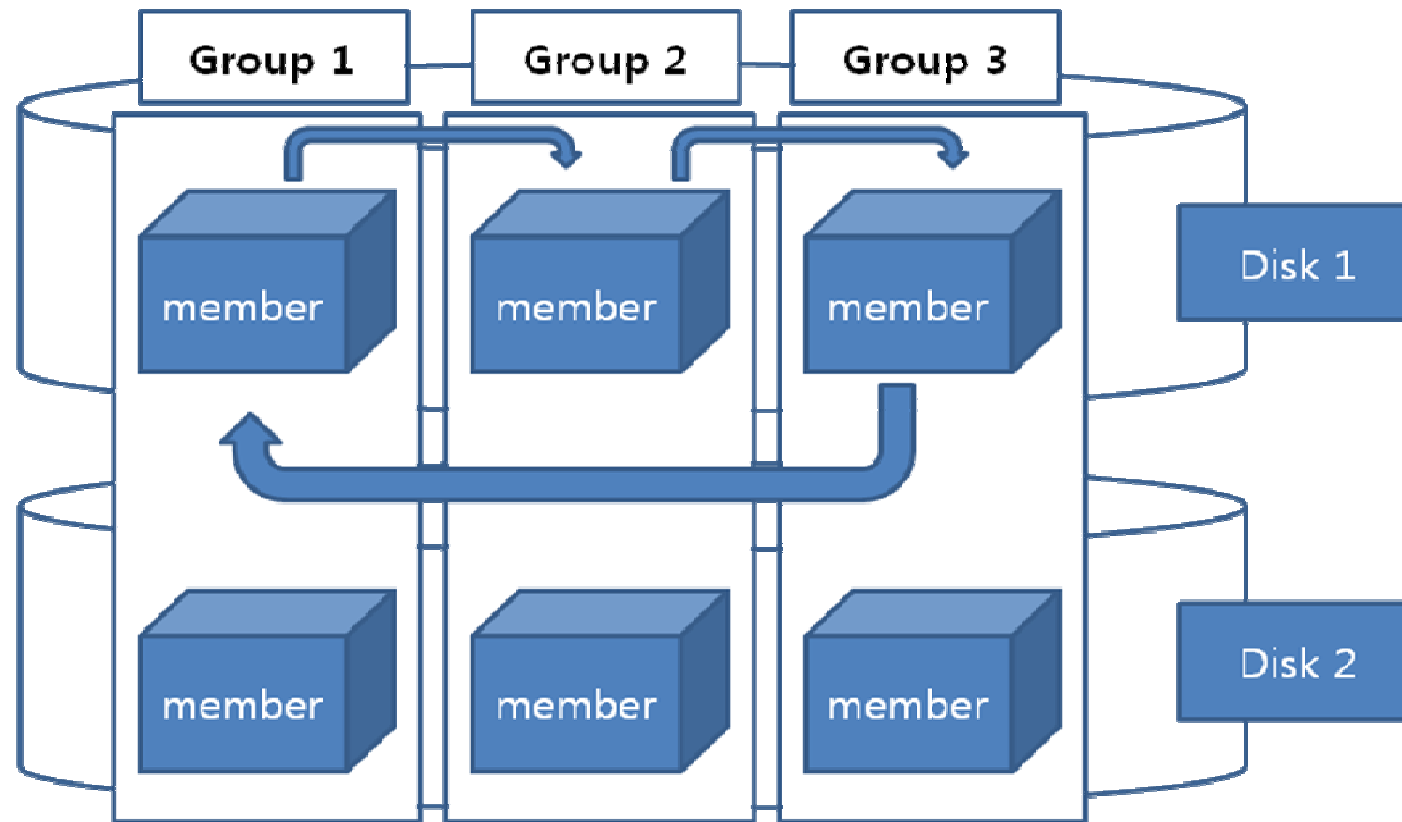
## 6. Redo log files

### 2. Redo Log File 구성

- Redo Log File은 그룹과 멤버라는 개념으로 관리
- 최소 그룹의 개수는 2개이며, 그룹별로 필요한 최소 Member 수는 1개
- 같은 그룹의 Member는 같은 내용 저장
- Member를 많이 추가하면 안정적일 수는 있지만, 기록시간이 커서 부하를 줄 수 있음
- 같은 그룹의 멤버는 서로 다른 디스크에 저장되는 것을 권장
- 그룹에 멤버가 여러 개일 경우 병렬로 동시에 같은 내용을 기록하는데 멤버가 같은 디스크에 존재한다면 직렬로 기록하게 된다

## 6. Redo log files

Redo log file



## 6. Redo log files

### 2. Redo Log File 구성

- LGWR이 Redo Log Buffer의 내용을 Redo Log File에 내려 쓰다가 해당 파일이 가득 차게 되면 Log Switch가 발생하여 자동으로 다음 그룹으로 넘어감
- Log Switch가 발생하게 되면 Checkpoint 신호가 발생
- Log Switch가 일어나는 그룹의 순서는 Oracle이 라운드 로빈 방식으로 결정
- Redo Log File크기가 너무 작을 경우 LOG SWITCH가 자주 발생하여 성능저하가 될 수 있고, 너무 크면 데이터의 손상 가능성이 커지므로 적절하게 설정 필요

## 6. Redo log files



### ❖ 빈번한 Log Switch와 관련된 에러

- checkpoint not completed
- Log Switch가 너무 빈번하게 일어날 경우 DBWR이 이전에 발생한 Check point 내용을 Data file에 다 기록을 못한 상태에서 다시 Log Switch가 발생하여 checkpoint 신호가 들어올 경우 발생

## 6. Redo log files

### 3. Redo Log File 관리

- 그룹 추가

```
SQL> alter database add logfile group 4  
2  '/home/oracle/oradata/testdb/redo04_a.log' size 5M ;
```

- 멤버 추가

```
SQL> alter database add logfile member  
2  '/home/oracle/oradata/testdb/redo04_b.log' to group 4;
```

- 멤버 삭제

```
SQL> alter database drop logfile member  
2  '/home/oracle/oradata/testdb/redo04_b.log' ;
```

- 그룹 삭제

```
SQL> alter database drop logfile group 4;
```



## 6. Redo log files



- Redo Log File 상태
  - CURRENT : LGWR이 내용을 기록하고 있는 상태
  - ACTIVE : Redo log file의 내용이 아직 DB Buffer Cache에서 Data file로 저장 안 되어서 지워지면 안 되는 상태(checkpoint 발생 후 inactive로 변경됨)
  - INACTIVE : Redo log file의 내용이 Data file로 저장되어 삭제되어도 된다는 의미

## 6. Redo log files

### 4. SCN

- 사용자가 Commit을 수행하게 되면, Oracle은 내부적으로 SCN이라는 고유번호를 생성해서 트랜잭션을 관리하게 된다.
- 이러한 SCN 번호를 통해 트랜잭션을 관리하면서, 장애 발생 시 복구한다
- System Commit Number(SCN)
  - Recover 시 Oracle은 SCN 정보를 사용하여 데이터베이스에 문제가 있는지 판별하고, 문제가 있다고 판단되는 경우 복구를 수행
  - SCN번호는 DML문장 단위가 아니라 트랜잭션 단위로 할당

## 6. Redo log files



- SCN 기록되는 곳
  - 1) Control File
    - Check point 발생했을 때
    - Reset Log 발생했을 때
    - Incomplete Recovery 수행 때
  - 2) Data blocks
    - Block Cleanout(row level lock 해제)시  
마지막 SCN을 각 Block에 기록

## 6. Redo log files



### 3) Data File Headers

- 모든 파일의 헤더에 아래의 경우 SCN을 기록한다.
  - \* 마지막 Check Point 발생 시
  - \* Begin Backup 수행 때
  - \* 복구가 되었다면, 사용된 마지막 SCN 을 기록한다.

### 4) Redo Records, Log Buffer

- Commit이 수행되면 Commit Record에 SCN을 포함하여 저장함

### 5) Undo Segement와 Tablespace Headers에도 기록이 된다.

## 6. Redo log files

### 5. CHECKPOINT

- Commit된 데이터를 어디까지 저장했는지 확인하기 위해서 만들어 놓은 개념
- ex) SCN이 100번까지 Commit되었고, CheckPoint 정보가 90번 이라면 SCN 90번 트랜잭션까지 데이터 파일에 저장 되었다고 확인하는 것.
- Datafile의 복구를 결정하는 기초적인 정보로써 Control File과 Data File의 Check Point 정보를 비교하고 서로 정보가 다르면 틀린 부분을 Online Redo 나 Archived Redo Log를 참조해서 복구

## 6. Redo log files

### 1) Database / Global Checkpoint

- DB Buffer Cache내에 있는 모든 저장 안된 Dirty Buffer들의 내용을 전부 데이터파일로 저장
- 저장된 SCN 중 가장 큰 SCN번호를 Control File과 Data File Header부분에 기록

### 2) Thread Checkpoint / Logical Checkpoint

- 해당 Thread 내의 저장되지 않은 모든 Dirty Buffer들을 Datafile로 내려 쓴다
- Log Switch가 발생 시 생김
- RAC환경일 경우 각 노드 별로 다르게 발생

## 6. Redo log files



### 3) Datafile Checkpoint

- 특정 데이터파일에만 발생하는 Check Point
- 해당 테이블스페이스를 offline, begin backup 시 발생
- 이 체크포인트 발생하면 해당 정보를 Control File과 데이터 파일 헤더 부분에 기록

### 4) Mini Checkpoint

- Drop Table과 같이 특정한 DDL 발생 시 특정 블록에만 발생하는 Checkpoint

## 6. Redo log files

---

### 5) Recovery Checkpoint

- 데이터 파일에 장애가 발생 했을 때 백업된 데이터파일 복원 시 Recovery된 블록을 데이터 파일에 저장 해야하는데 이때 발생하는 CheckPoint를 Recovery Check Point라고 함



## 6. Redo log files

### \* 실무 실습 6. Redo log file 관리 하기

```
SQL> set line 200
```

```
SQL> col a.group# for 999
```

```
SQL> col member for a50
```

```
SQL> select a.group# , a.member, b.bytes/1024/1024 MB , b.archived , b.status  
2 from v$logfile a , v$log b  
3 where a.group#=b.group#  
4 order by 1,2 ;
```

변경 전 상태

GROUP#	MEMBER	MB	ARCHIV	STATUS
1	/home/oracle/oradata/testdb/redo01.log	50	NO	CURRENT
2	/home/oracle/oradata/testdb/redo02.log	50	NO	INACTIVE
3	/home/oracle/oradata/testdb/redo03.log	50	NO	INACTIVE

## 6. Redo log files

### \* 연습 문제 2: Redo log file 변경

모든 redo log file 을 /home/oracle/disk1/ 로 이동하세요~  
방법은 기존 로그를 전부 지우고 위 경로에 새로 만드시면 됩니다.  
그룹번호는 1,2,3 으로 3개의 그룹을 만드시고 각 멤버는 1개씩,  
크기는 각 5 M 입니다.

## 6. Redo log files

---

### \* 연습 문제 3: Redo log file 변경

`/home/oracle/disk1/redo01_a.log , redo02_a.log , redo03_a.log`  
`/home/oracle/disk2/redo01_b.log , redo02_b.log , redo03_b.log`  
가 되도록 변경해 보세요.

## 6. Redo log files

### \* 연습 문제 4: Control file / Redo log file 변경

/home/oracle/disk3/control01.ctl , redo01\_a.log ,

redo02\_a.log , redo03\_a.log

/home/oracle/disk4/control02.ctl , redo01\_b.log ,

redo02\_b.log , redo03\_b.log

/home/oracle/disk5/control03.ctl , redo01\_c.log ,

redo02\_c.log , redo03\_c.log

## 6. Redo log files

### Redo log file dump

```
[oracle@localhost ~]$ sqlplus / as sysdba;
```

```
SQL> alter session set events 'immediate trace name  
      REDOHDR level 10';
```

```
SQL> oradebug setmypid;
```

```
SQL> oradebug dump redohdr 10;
```

여기 까지 수행 한 후 Udump 경로에서 trace 파일 내용 확인