

## Oracle Database Administration Workshop I Test Scenario

### Practice 9 Database Maintenance

1. HR의 테이블들에 Optimizer 통계정보가 있는지 확인하고 새로운 통계를 생성합니다.

```
SQL> conn hr/hr
```

```
SQL> SELECT table_name, num_rows, blocks, empty_blocks, last_analyzed  
FROM user_tables; → 통계정보가 없음을 확인
```

```
SQL> analyze table employees compute statistics;
```

```
SQL> analyze table employees delete statistics;
```

```
SQL> SELECT table_name, num_rows, blocks, empty_blocks, last_analyzed  
FROM user_tables; → 단일 테이블 통계 생성
```

```
SQL> analyze table employees delete statistics;
```

```
SQL> exec dbms_stats.gather_schema_stats('HR');
```

```
SQL> SELECT table_name, num_rows, blocks, empty_blocks, last_analyzed  
FROM user_tables; → 스키마의 모든 테이블 통계 생성
```

```
SQL> exec dbms_stats.delete_schema_stats('HR');
```

2. EM을 통해 객체통계(Optimizer 통계) 수집의 자동화 설정을 확인합니다.

a) EM 실행 > Server Page로 이동 > Scheduler 머리글 아래에서 Automated Maintenance Tasks 선택

b) Task Name이 Optimizer Statistics Gathering인 항목을 찾아 일정을 확인

c) Configure를 누르면 Automated Maintenance Tasks Configuration 페이지로 이동하여 일정을 조정할 수 있습니다.

d) 변경하고자 하는 요일의 윈도우를 선택하여 일정을 조정합니다.

- 날짜 : 실습당일

- 시간 : 현재 시각 10분 후

- 기간(Duration) : 0시간 30분

e) 예정된 시간 이후 테이블 통계를 확인합니다.

3. 현재 Database 시스템 통계 수집 레벨을 확인합니다.

```
SQL> conn / as sysdba
```

```
SQL> show parameter statistics_level
```

4. 현재 Database 통계 수집 관련 설정을 확인합니다.

a) EM의 Server 탭 페이지에서 Statistics Management의 Automatic Workload Repository를 누릅니다.

b) Automatic Workload Repository 화면에서 다음 값을 확인합니다.

- 통계수집레벨
- 통계수집간격
- 통계정보(스냅샷) 보관기간

c) 리눅스 터미널에서 다음을 실행하고 자동수집관련 Background Process를 찾아봅니다.

```
$ ps -ef | grep orcl ➔MMON Process 확인
```

5. System 관련 Metric 설정 값을 확인한 후 Metric에 설정된 임계치가 초과되는 상황을 만들고 Server Generated Alert를 확인한 후 문제를 해결합니다.

a) 다음과 같이 EM을 통해 Metric 임계치를 확인합니다.

EM Database Home 페이지의 하단 Related Links 영역에서 "Metric and Policy Settings"를 누릅니다.

b) Tablespace Used(%) Metric의 임계치를 확인합니다. (Warning 85%, Critical 97%)

c) 사이즈가 작은 테이블스페이스를 만들고 데이터를 채워서 Tablespace Used 임계치의 초과상황을 만듭니다.

```
SQL> conn / as sysdba
```

```
SQL> CREATE TABLESPACE sales_data
```

```
DATAFILE '/u01/app/oracle/oradata/orcl/sales_data01.dbf' size 10m;
```

```
SQL> conn hr/hr
```

```
SQL> CREATE TABLE emp_sales
```

```
TABLESPACE sales_data
```

```
AS SELECT * FROM employees;
```

```
SQL> DECLARE
```

```
n NUMBER;
```

```
BEGIN
```

```
FOR n IN 1..13
```

```
LOOP
```

```
INSERT INTO emp_sales SELECT * FROM emp_sales;
```

```
END LOOP;
```

```
COMMIT;
```

```
END;
```

```
/
```

SQL> SELECT COUNT(\*) FROM emp\_sales; ⇒ 공간부족 오류가 발생 및 ROLLBACK 발생

d) 시간이 조금 흐른 후 EM Database Home 페이지의 Alert영역에서 경고를 확인합니다. 경고메시지의 링크를 누르면 상세 페이지로 이동합니다.

d) 문제를 해결한 후 경고로그가 지원진 것을 확인합니다.

SQL> conn / as sysdba

SQL> ALTER DATABASE DATAFILE '/u01/app/oracle/oradata/orcl/sales\_data.dbf'

RESIZE 100m;

SQL> conn hr/hr

SQL> TRUNCATE TABLE emp\_list;

SQL> INSERT INTO emp\_list

SELECT \* FROM employees;

SQL> DECLARE

n NUMBER;

BEGIN

FOR n IN 1..13

LOOP

INSERT INTO emp\_list SELECT \* FROM emp\_list;

END LOOP;

COMMIT;

END;

/

SQL> SELECT COUNT(\*) FROM emp\_list;

e) 다시 EM Database Home 페이지의 Alert 영역에 경고가 사라진 것을 확인합니다.

f) 실습내용을 정리한 후 데이터베이스에서 Clear합니다.(lab09\_04.sql)

SQL> DROP TABLESPACE sales\_data INCLUDING CONTENTS AND DATAFILES;

Q1. ANALYZE 명령과 DBMS\_STATS Package의 차이점을 쓰시오.

Q2 STATISTICS\_LEVEL 파라미터가 어떤 값으로 설정되어 있습니까?

Q3. Snapshot 보관 날 수 및 Snapshot 수집 간격, 현재까지 수집된 Snapshot의 개수를 각각 적으시오.

보관 날 수 :

수집 간격 :

스냅샷 개수 :

Q4. 시스템관련 통계정보를 자동으로 수집하는 Background Process의 이름은 무엇입니까?

## Practice 10 Performance Management

1. 메모리관련 Parameter 값을 확인하고 다음에 기록합니다.

```
SQL> conn / as sysdba
```

```
SQL> show parameter pool_size
```

```
SQL> show parameter db_cache
```

```
SQL> show parameter target
```

다음 메모리 설정 파라미터의 값을 기록합니다.

shared\_pool\_size

large\_pool\_size

java\_pool\_size

db\_cache\_size

sga\_target

memory\_target

pga\_aggregate\_target

2. SGA 전체사이즈를 조사한 후 SGA 메모리의 할당 상황을 점검합니다.

```
SQL> select sum(value)/1024/1024 from v$sga;
```

```
SQL> select pool, sum(bytes)/1024/1024 pool_size from v$sgastat
```

```
Where pool is not null
```

```
group by pool;
```

```
SQL> select current_size from v$buffer_pool;
```

<참고>V\$SGA\_Dynamic\_components 뷰를 조회하면 자동 튜닝 component들의 실제 사이즈를 확인할 수 있다.

```
SQL> col component format a30
```

```
SQL> SELECT component, current_size/1024/1024 AS current_size  
FROM v$sga_dynamic_components;
```

3. 오라클 메모리 관리방식을 AMM에서 ASMM으로 변경해 봅니다.

a) ASMM 방식 설정을 위해 MEMORY\_TARGET=0으로 설정하여 SGA\_TARGET과 PGA\_AGGREGATE\_TARGET이 별도로 설정되는 것을 확인합니다.

```
SQL> ALTER SYSTEM SET memory_target = 0 scope=memory;
```

```
SQL> show parameter target
```

b) 데이터베이스를 재시작 한 후 AMM 방식으로 재설정 된 것을 확인합니다.

```
SQL> startup force
```

```
SQL> show parameter target
```

4. 인덱스화 된 테이블에 MOVE 명령 실행 후 인덱스의 상태를 조사한 후 Index Rebuild를 통해 문제를 해결하고 확인합니다.

```
SQL> conn hr/hr
```

```
SQL> CREATE TABLE emp10
```

```
AS SELECT * FROM employees;
```

```
SQL> CREATE INDEX emp10_id_ix ON emp10(employee_id);
```

```
SQL> SELECT index_name, status FROM user_indexes
```

```
WHERE table_name = 'EMP10';
```

```
SQL> ALTER TABLE emp10 MOVE;
```

```
SQL> SELECT index_name, status FROM user_indexes
```

```
WHERE table_name = 'EMP10';
```

```
SQL> ALTER INDEX emp10_id_ix REBUILD;
```

```
SQL> SELECT index_name, status FROM user_indexes
```

```
WHERE table_name = 'EMP10';
```

5. 실습내용을 CLEAR 합니다.

```
SQL> DROP TABLE emp10 PURGE;
```

Q1 현재 데이터베이스에서 사용하는 메모리관리 설정은 AMM 인니까, ASMM 인니까?

Q2 메모리 관리방식 AMM과 ASMM의 차이점을 정리하세요.

Q3 테이블이동과 같은 작업을 수행한 후 테이블관련 인덱스의 상태가 어떻게 바뀌었습니까?

Q4 문제가 발생한 인덱스를 재구축하는 명령문은 무엇입니까?

6. 추가 실습 : SQL 추적기능을 사용해 봅니다.

a) PLUSTRACE 롤을 데이터베이스에 생성한 후 HR에게 권한을 부여합니다.

```
SQL> conn / as sysdba
```

```
SQL> @?/sqlplus/admin/plustrce.sql
```

```
SQL> GRANT plustrace TO hr;
```

b) 추적기능을 활성화하여 인덱스 사용 여부에 따른 SQL 실행계획 및 성능비교를 수행해 봅니다.

```
SQL> conn hr/hr
```

```
SQL> DROP INDEX bigemp_empid_ix;
```

```
SQL> SELECT index_name FROM user_indexes
```

```
WHERE table_name = 'BIGEMP';
```

```
SQL> set autotrace traceonly
```

```
SQL> SELECT * FROM bigemp
```

```
WHERE employee_id =123;
```

```
SQL> set autotrace off
```

```
SQL> CREATE INDEX bigemp_empid_ix ON bigemp(employee_id);
```

```
SQL> SELECT index_name FROM user_indexes
```

```
SQL> WHERE table_name = 'BIGEMP';
```

```
SQL> set autotrace traceonly
```

```
SQL> SELECT * FROM bigemp
```

```
WHERE employee_id =234;
```

```
SQL> set autotrace off
```

## **Practice 11 Backup and Recovery Concepts**

1. redundancy 보장을 위해 두 개의 Control file이 있는지 확인합니다.

```
SQL> select name from v$controlfile;
```

2. 파라미터 파일을 백업한 후 새 control file을 추가합니다.

```
SQL> CREATE PFILE FROM SPFILE;
```

```
SQL> ALTER SYSTEM SET control_files =
```

```
'/u01/app/oracle/oradata/orcl/control01.ctl' ,
```

```
'/u01/app/oracle/flash_recovery_area/orcl/control02.ctl' ,
```

```
'/u01/app/oracle/oradata/orcl/control03.ctl' SCOPE=SPFILE;
```

```
SQL> shutdown immediate
```

```
SQL> exit
```

```
$ cd /u01/app/oracle/oradata/orcl
```

```
$ cp control01.ctl control03.ctl
```

```
$ sqlplus / as sysdba
```

```
SQL> startup
```

2. 각 리두 로그 그룹과 멤버의 수를 확인하고 그룹과 멤버를 추가합니다.

a) 로그 그룹에 대한 정보를 수집합니다.

```
SQL> select group#, sequence#, members, status,  
  
        TO_CHAR(first_time,'yy/mm/dd hh24:mi:ss') AS first_time  
  
        from v$log;
```

b) 로그 멤버에 대한 정보를 수집합니다.

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

```
SQL> select group#, member, status from v$logfile;
```

c) 각 리두 로그 그룹에 멤버를 하나씩 추가합니다.

```
SQL> conn /as sysdba
```

```
SQL> alter database add logfile member
```

```
'/u01/app/oracle/oradata/orcl/redo01a.log' to group 1,
```

```
'/u01/app/oracle/oradata/orcl/redo02a.log' to group 2,
```

```
'/u01/app/oracle/oradata/orcl/redo03a.log' to group 3 ;
```

```
SQL> alter system switch logfile; 여러 번 수행
```

d) 4번 로그 그룹을 추가합니다.

```
SQL> ALTER DATABASE ADD LOGFILE GROUP 4
```

```
('/u01/app/oracle/oradata/orcl/redo04.log', '/u01/app/oracle/oradata/orcl/redo04a.log') SIZE 50M;
```

e) 로그그룹과 멤버의 추가를 확인합니다.

```
SQL> select group#, members, status, bytes/1024/1024 AS file_size from v$log
```

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

```
SQL> select group#, member, status from v$logfile;
```

3. 현재 데이터베이스의 운영모드를 확인한 후 리두 로그가 아카이브되도록 데이터베이스 ARCHIVELOG 모드로 변경합니다.

a) 데이터베이스의 운영모드를 확인합니다.

```
SQL> conn / as sysdba
```

```
SQL> archive log list
```

b) 다음을 절차적으로 실행하여 데이터베이스 운영모드를 변경한 후 확인합니다.

```
SQL> shutdown immediate
```

```
SQL> startup mount;
```

```
SQL> ALTER DATABASE ARCHIVELOG;
```

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

```
SQL> archive log list
```

4. Archive Log Destination 설정을 확인합니다.

a) EM에서 Availability 탭을 열고 Backup/Recovery > Setup > Recovery Settings를 선택합니다.

b) Media Recovery 영역에서 ARCHIVELOG Mode 체크박스를 확인합니다.

c) Log Archive Filename Format에 %t, %s, %r 이 포함 되어 있는지 확인합니다.

d) Flash Recovery Area에 아카이브 로그를 저장하도록 데이터베이스가 사전 구성되어 있음을 확인합니다.

5. BIGEMP 테이블 수정 및 강제로 로그스위치를 실행하고 archive log file이 생성되는 것을 확인합니다.

```
SQL> UPDATE hr.bigemp
```

```
SET salary = salary*1.05;
```

```
SQL> alter system switch logfile; ➔ 여러 번 수행
```

```
SQL> exit
```

\$ cd \$ORACLE\_BASE/flash\_recovery\_area/ORCL/archivelog ⇒ 해당 디렉토리로 가서 Archived Log File 생성을 확인합니다.

6. /home/oracle/backup/arch1으로 추가 Archive Log Destination을 구성합니다.

a) Archived Log File을 위하여 새 디렉토리를 생성합니다.(\$HOME/backup/arch1)

```
$ mkdir $HOME/backup
```

```
$ cd $HOME/backup
```

```
$ mkdir arch1
```

b) EM에서 Availability 탭을 열고 > Backup/Recovery > Setup > Recovery Settings > Archive Log Destination 에서 새로운 아카이브 경로를 입력하기 위해 Add Another Row 버튼을 클릭합니다. 생성된 number 2에 /home/oracle/backup/arch1/를 입력합니다. **디렉토리 경로는 반드시 “/(슬래시)”로 끝나야** 하며 Quota는 비워둡니다.



c) Show SQL을 눌러 명령문을 검토하고 Return을 누르고 Apply를 합니다.

7. BIGEMP 갱신 후 Archived Log File이 두 경로에 모두 생성되는 것을 확인합니다.

```
SQL> alter system switch logfile ⇒ 여러 번 수행
```

```
SQL> conn hr/hr
```

```
SQL> update bigemp
```

```
Set salary = salary * 1.05 ;
```

```
SQL> ! ls $HOME/backup/arch1
```

8. 다음을 실행하여 실습을 클리어 합니다.

a) EM에서 Availability 탭을 열고 > Backup/Recovery > Setup > Recovery Settings > Archive Log Destination의 number 2에 지정된 /home/oracle/backup/arch1/ 경로 정보를 지우고 Apply를 합니다.

b) 이후의 실습을 위해 redo log file과 control file의 다중화 설정은 계속 유지합니다. 선택적으로 본 단원의 실습을 다시 진행하기를 원한다면 lab11\_clear.sql을 실행합니다.

- Q1. orcl 데이터베이스에 control file이 몇 개 있습니까?
- Q2. 리두 로그 그룹과 멤버의 수를 기록하십시오.
- Q3. 현재 데이터베이스 운영모드가 무엇입니까?(데이터베이스 초기값)
- Q4. Archive Log의 기본 경로는 어디입니까?

## Practice 12 Performing Database Backups

1. 다음과 같이 Control File Backup을 실행합니다.

a) trace 된 control file은 user\_dump\_dest에 생성됩니다. 해당 경로를 정리한 후 trace를 실행합니다.

```
$ cd $ORACLE_BASE/diag/rdbms/orcl/orcl/trace
```

```
$ rm *
```

```
$ sqlplus / as sysdba
```

```
SQL> show parameter user_dump
```

```
SQL> alter database backup controlfile to trace;
```

```
SQL> select s.username, p.spid
```

```
from v$session s JOIN v$process p
```

```
ON (s.paddr = p.addr)
```

```
and s.username = 'SYS'; ⇒ ex)12345
```

```
SQL> exit
```

```
$ cd $ORACLE_BASE/diag/rdbms/orcl/orcl/trace
```

\$ ls

b) User\_dump\_dest에서 조회된 process id를 포함하는 trace 파일이 생성되었음을 확인하고 파일을 봅니다.

⇒ ex) orcl\_ora\_12345.trc

\$ vi orcl\_ora\_12345.trc

c) trace file을 다른 이름으로 backup 합니다.

\$ cp orcl\_ora\_12345.trc /home/oracle/backup/cre\_con.sql

d) control file image backup을 생성합니다.

\$ sqlplus / as sysdba

SQL> alter database backup controlfile

To '/home/oracle/backup/control.bk';

2. EM에서 Backup Setting을 확인합니다.

EM > Availability > Backup /Recovery > Setup > Settings 페이지의 Device tab에서 다음 항목을 조사합니다.

- Parallelism(Channel의 병렬도) :
- Disk Backup Location :
- Disk Backup Type

3. EM에서 전체 데이터베이스 Backup을 수행합니다.

a) EM Homepage > Availability > Schedule Backup(Manage 섹션)을 선택합니다.

b) 여기에서는 연습 목적으로 Customized Backup의 일정을 잡습니다.

c) 백업할 객체로 Whole Database를 선택합니다.

d) 서버의 Host Credentials Username 및 Password에 각각 oracle을 확인하거나 입력합니다.

e) Schedule Customized Backup을 누릅니다.

f) Schedule Customized Backup: Options 페이지에서 Backup Type에 Full Backup을

선택하고 "Use as the base of an incremental backup strategy" 체크 박스를 선택합니다.

g) Backup Mode로 Online Backup을 선택합니다.

h) Advanced 섹션에서 "Also back up all archived logs on disk" 및 "Delete all archived logs from disk after they are successfully backed up"을 선택한 다음 Next를 눌러 계속합니다.

i) Schedule Customized Backup: Settings 페이지에서 백업 위치에 Disk를 선택합니다. (Disk Backup Location을 그대로 두고 일회용 백업을 위해 현재 설정을 무효화할 수 있습니다. 그러나 이번에는 해당 버튼을 누르지 마십시오.)

j) Next를 누릅니다.

k) Schedule Customized Backup: Schedule 페이지에서 기본값을 모두 적용한 다음 Next를 눌러 계속 진행합니다.

주: Schedule Type은 One Time(Immediately)이어야 합니다.

l) Schedule Customized Backup: Review 페이지에서 RMAN 스크립트를 검토한 다음 Submit Job을 누릅니다.

m) View Job을 눌러 백업 작업의 상태를 모니터링합니다. 이 백업 시간은 하드웨어 및 시스템 리소스에 따라 다릅니다.

n) 작업이 완료될 때까지 브라우저의 Refresh 또는 Requery 버튼을 누릅니다.

4. 생성된 Backup을 확인합니다.

```
$ cd $ORACLE_BASE/flash_recovery_area/ORCL/
```

5. RMAN을 실행하고 데이터베이스 Backup을 수행해 봅니다.

```
$ rman target /
```

```
RMAN> backup database;
```

### Practice 13 Performing Database Recovery

1. 손실된 컨트롤파일을 복구합니다.

a) 다중화된 컨트롤파일 중 하나를 삭제합니다. 다음 명령으로 controlfile이 다중화되어 있는지 확인합니다.

```
SQL> select name from v$controlfile;
```

b) 다중화 된 control file 가운데 하나를 삭제합니다.

```
$ rm $ORACLE_BASE/flash_recovery_area/orcl/control02.ctl
```

c) SQL\*Plus를 사용하여 데이터베이스를 시작해보고 인스턴스의 현재 상태를 확인합니다.

```
SQL> conn /as sysdba
```

```
SQL> startup force
```

```
SQL> select status from v$instance;
```

```
SQL> alter database mount; ➔ ERROR!!
```

d) Instance가 컨트롤파일 중 하나를 찾지 못했으므로 마운트 단계로 이동할 수 없습니다. Alert log의 마지막 부분을 확인하고 어떤 컨트롤파일이 문제인지 알아봅니다.

```
SQL> host tail -10 $ORACLE_BASE/diag/rdbms/orcl/orcl/trace/alert*
```

e) Flash Recovery Area의 컨트롤 파일이 누락되었습니다. 기존 컨트롤 파일을 복사하여 누락된 컨트롤 파일을 복원한 다음 데이터베이스를 마운트하고 엽니다.

```
SQL> host
```

```
$ cd $ORACLE_BASE/oradata/orcl
```

```
$ ls -l
```

```
$ cp control01.ctl $ORACLE_BASE/flash_recovery_area/orcl/control02.ctl
```

\$ exit

SQL> alter database mount;

SQL> alter database open;

: ALTER DATABASE 명령은 명령 별로 하나의 상태 레벨만 변경할 수 있기 때문에 NOMOUNT에서 OPEN으로 이동할 때 두 개의 명령을 사용해야만 합니다.

2. 손실된 응용 프로그램 데이터파일을 복구합니다.

a) 현재 데이터베이스 상태를 확인한 후 RMAN backup을 수행합니다.

SQL> select avg(salary) from hr.employees;

SQL> conn / as sysdba

SQL> archive log list ⇒ CURRENT LOG SEQUENCE 번호 확인

SQL> exit

\$ rman target /

RMAN> backup database;

RMAN> exit;

b) 데이터베이스 운영을 시뮬레이션 합니다.

SQL> alter system switch logfile; ➔ 여러 번 수행

SQL> update hr.employees

Set salary = salary \*1.2;

SQL> commit;

SQL> select avg(salary) from hr.employees;

SQL> alter system switch logfile; ⇒ 여러 번 수행

SQL> archive log list ➔ CURRENT LOG SEQUENCE 번호 확인

SQL> exit

c) 장애 발생을 시뮬레이션 합니다.

\$ cd \$ORACLE\_BASE/oradata/orcl/

\$ rm users01.dbf

SQL> conn / as sysdba

SQL> SELECT \* FROM hr.locations; ➔ 에러발생

SQL> alter tablespace users offline immediate;

d) RMAN을 실행하고 데이터베이스 복구를 수행합니다.

```
$ rman target /
```

```
RMAN> restore tablespace users;
```

➔ 실행 후 별도의 터미널에서 \$ORACLE\_BASE/oradata/orcl에 users01.dbf file이 생겼음을 확인합니다.

```
RMAN> recover tablespace users;
```

```
RMAN> sql 'alter tablespace users online';
```

```
RMAN> exit;
```

e) 복구의 성공여부를 확인합니다.

```
SQL> conn / as sysdba
```

```
SQL> archive log list
```

```
SQL> SELECT * FROM hr.locations;
```

```
SQL> select avg(salary) from hr.employees;
```

3. 손실된 시스템파일을 복구합니다.

a) 데이터베이스 운영을 시뮬레이션 합니다.

```
SQL> alter system switch logfile; ⇒ 여러 번 수행
```

```
SQL> update hr.employees
```

```
Set salary = salary *1.2;
```

```
SQL> commit;
```

```
SQL> select avg(salary) from hr.employees;
```

```
SQL> alter system switch logfile; ⇒ 여러 번 수행
```

```
SQL> archive log list ⇒ CURRENT LOG SEQUENCE 번호 확인
```

```
SQL> exit
```

b) 장애 발생을 시뮬레이션 합니다.

```
$ cd $ORACLE_BASE/oradata/orcl/
```

```
$ rm system01.dbf
```

```
SQL> conn / as sysdba
```

```
SQL> SELECT * FROM hr.locations;
```

```
SQL> shutdown abort
```

c) RMAN을 실행하고 데이터베이스 복구를 수행합니다.

\$ rman target /

RMAN> startup mount

RMAN> restore datafile 1; ⇒ 실행 후 별도의 터미널에서 \$ORACLE\_BASE/oradata/orcl에 system01.dbf file이 생겼음을 확인합니다.

RMAN> recover datafile 1;

RMAN> alter database open;

RMAN> exit;

d) 복구의 성공여부를 확인합니다.

SQL> conn / as sysdba

SQL> archive log list

SQL> SELECT \* FROM hr.locations;

SQL> select avg(salary) from hr.employees;

4. 손실된 리두 로그 멤버를 복구합니다. 리두 로그 파일 중 하나를 삭제한 후 Alert log의 오류를 확인하고 복구합니다.

a) 다음을 실행하여 GROUP 2의 리두 로그 파일 중 하나를 삭제합니다.

SQL> conn / as sysdba

SQL> SELECT group#, members, status FROM v\$log;

➔ STATUS 열의 상태가 2번 group은 INACTIVE 3번 group이 CURRENT 이어야 합니다. 이 조건에 만족할 수 있도록 필요하다면 로그스위치를 강제로 수행한 후 질의를 다시 실행합니다.

SQL> alter system switch logfile;

SQL> select group#, member from v\$logfile;

SQL> alter system checkpoint;

SQL> exit

\$ rm \$ORACLE\_BASE/oradata/orcl/redo02.log ⇒ Redo Log File중 GROUP 2의 member file 하나를 삭제

b) 데이터베이스는 정상적으로 작동하지만 누락된 리두 로그 파일에 대해 확인할 수 있습니다.

SQL> col member for a50

SQL> select group#, member, status from v\$logfile;

c) 문제를 찾아 해결합니다. 운영 체제 명령을 사용하여 INVALID로 표시된 리두 로그에 문제가 있는지 확인합니다.

SQL> host ls - \$ORACLE\_BASE/oradata/orcl

: 여섯 개의 파일이 있어야 하는데 다섯 개의 파일만 있습니다. 로그 그룹 별로 유효한 멤버가 하나 이상 있으면 데이터베이스 작업이 계속됩니다.

SQL> shutdown abort

SQL> startup

d) 누락된 redo log file을 복구합니다.

SQL> alter database clear logfile group 2;

\$ ls \$ORACLE\_BASE/oradata/orcl

\$ sqlplus /nolog

SQL> conn /as sysdba

SQL> alter system switch logfile; ⇒ 여러 번 수행

SQL> select member, status from v\$logfile;

SQL> exit

### **Practice 14 Moving Data**

1. SQL\*Loader를 실습합니다.

a) 실습에 필요한 테이블을 다음과 같이 생성합니다.

SQL> conn hr/hr

SQL> DROP TABLE departments2 PURGE;

SQL> CREATE TABLE departments2

(DEPT\_ID NUMBER(4),

DEPT\_NAME VARCHAR2(30))

TABLESPACE users;

SQL> SELECT \* FROM departments2;

SQL> exit

b) SQL\*Loader를 실행하여 데이터를 DEPARTMENTS2 테이블로 로드합니다.

sqlldr hr/hr control=lcase1.ctl

SQL> conn hr/hr

SQL> SELECT \* FROM departments2;

c) DEPARTMENTS2 테이블을 TRUNCATE 한 후 lcase2.ctl 파일을 이용하여 로드합니다.

\$sqlldr hr/hr control=lcase2.ctl data=lcase2.dat

SQL> conn hr/hr

SQL> SELECT \* FROM departments2;

2. Export / Import를 수행해 봅니다.

a) Export Utility를 사용하여 HR schema의 table을 export 합니다.

```
SQL> conn hr/hr
```

```
SQL> CREATE TABLE c_emp
```

```
AS SELECT * FROM employees;
```

```
SQL> CREATE TABLE c_dept
```

```
AS SELECT * FROM departments;
```

```
$ exp hr/hr file=/home/oracle/backup/hrtab.dmp tables=c_emp, c_dept
```

b) HR User로 connect 하고 c\_emp, c\_dept table을 drop 합니다.

```
SQL> conn hr/hr
```

```
SQL> select count(*) from c_emp;
```

```
SQL> select count(*) from c_dept;
```

```
SQL> drop table c_emp;
```

```
SQL> drop table c_dept;
```

```
SQL> exit;
```

c) Import Utility를 사용하여 C\_EMP, C\_DEPT table을 Restore 하고 확인합니다.

```
$ imp hr/hr file=/home/oracle/backup/hrtab.dmp tables=c_emp, c_dept
```

```
SQL> conn hr/hr
```

```
SQL> select count(*) from c_emp;
```

```
SQL> select count(*) from c_dept;
```

3. Datapump를 사용하여 봅니다.

a) OS에서 data directory 생성합니다.

```
$ mkdir $HOME/data
```

```
SQL> conn /as sysdba
```

```
SQL> create directory data_dir as '/home/oracle/data';
```

```
SQL> grant read, write on directory data_dir to public;
```

```
SQL> exit
```

b) HR 유저의 EMPLOYEES , DEPARTMETNS 테이블 EXPDP 실행합니다.

```
$ expdp hr/hr tables=employees, departments directory=data_dir dumpfile=hrtab.dat
```



c) 새 사용자 DP 유저를 생성합니다.

```
SQL> conn /as sysdba
```

```
SQL> create user dp identified by dp
```

```
default tablespace users
```

```
temporary tablespace temp;
```

```
SQL> grant connect, resource, dba to dp;
```

```
SQL> CONN dp/dp
```

```
SQL> SELECT * from tab; → 소유한 테이블이 없음을 확인
```

```
SQL> exit
```

d) DP 유저에 실습2번의 DUMP file을 이용하여 EMPLOYEES, DEPARTMENTS 테이블을 생성합니다.

```
$ impdp system/oracle directory=data_dir dumpfile=hrtab.dat remap_schema=hr:dp
```

e) DP 유저에 테이블이 생성된 것을 확인합니다.

```
SQL> CONN DP/DP
```

```
SQL> select * from tab;
```

```
SQL> exit
```

#### 4. 실행중인 datapump 작업 모니터 및 관리하기

a) 테스트를 위하여 세 개의 터미널을 open 한다.

<session 1> 현재 실행중인 datapump 작업이 없음을 확인

```
SQL> conn system/oracle
```

```
SQL> col owner_name for a10
```

```
SQL> col job_name for a10
```

```
SQL> col job_mode for a10
```

```
SQL> col state for a15
```

```
SQL> col operation for a15
```

```
SQL> set line 100
```

```
SQL> select * from dba_datapump_jobs;
```

```
SQL> select table_name from user_tables
```

```
where table_name='DPTEST';
```

<session 2> expdp 작업을 시작

```
$ expdp system/oracle directory=data_dir job_name=dptest
```

```
logfile=dptest.log dumpfile=dptest%U.dmp full=y
```

b) 현재 작업 모니터링 및 attach

<session 1> 현재 작업 내용확인

```
SQL> select * from dba_datapump_jobs;
```

```
SQL> select table_name from user_tables
```

```
where table_name='DPTEST'; ➔ DPTTEST 테이블 생성
```

<session 3> attach로 현재 작업중인 exp에 연결하여 작업상태 확인하기

```
$ expdp system/oracle attach=dptest
```

c) 작업중인 datapump 작업 변경하고 확인

<session 3> 에서 병렬도 변경

```
EXPORT> parallel=4
```

```
EXPORT> status
```

```
EXPORT> continue_client
```

<session 1> 작업상태 확인

```
SQL> select * from dba_datapump_jobs; ➔ degree 수가 변경됨
```

c) expdp 작업 종료 후 <session 1>에서 job 확인하기

```
SQL> select * from dba_datapump_jobs;
```

```
SQL> select table_name from user_tables where table_name='DPTEST';
```