4️⃣

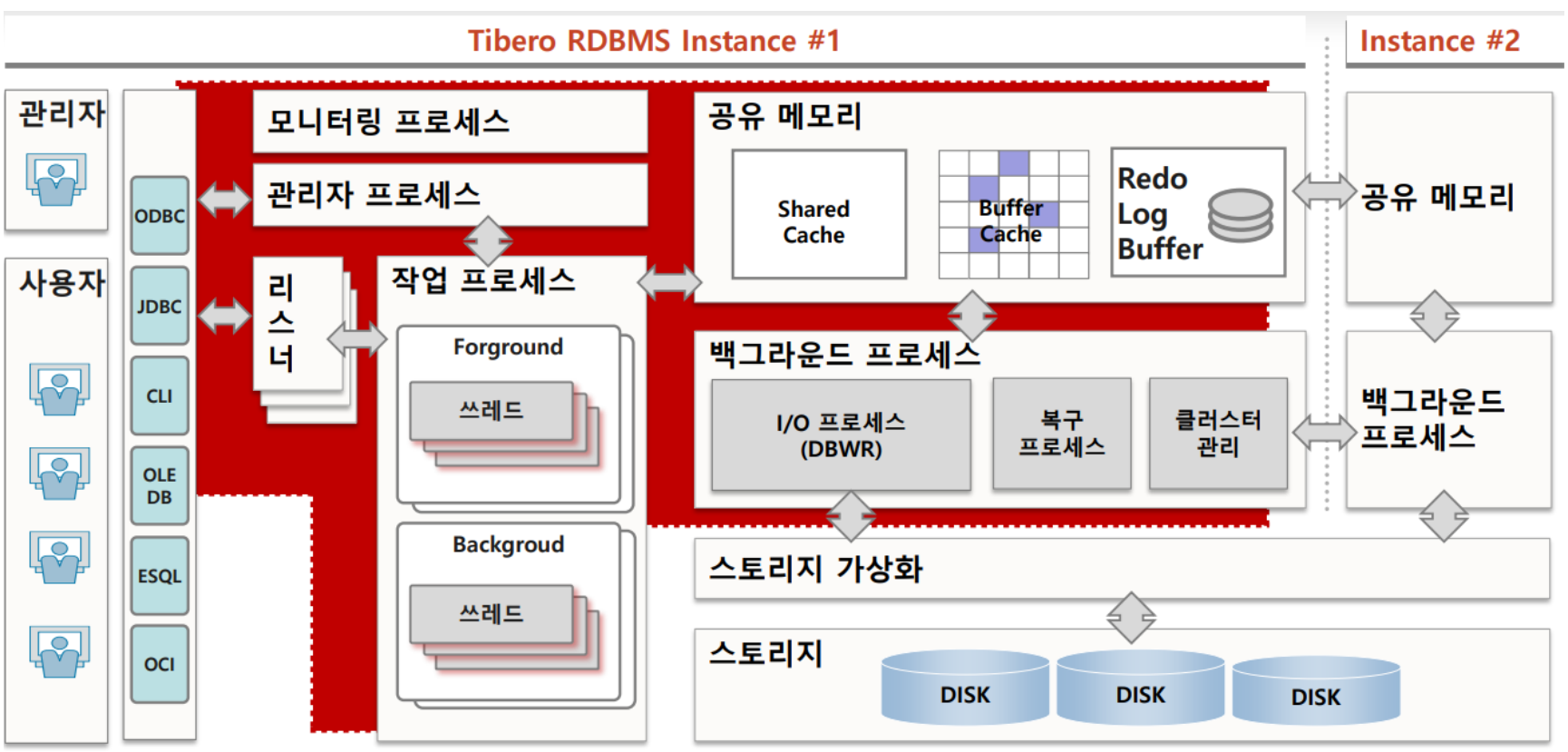
티베로 DBMS

[Summary]

▼ DAY 1 - Tibero Architecture

▼ 1. 인스턴스 구조

▼ Tibero 전체 구조



- 티베로가 설치가 된 곳엔 티베로 인스턴스, 티베로 데이터베이스가 설치됨
- 티베로 binary에서 티베로 인스턴스(여러 process, memory) 실행됨
- 실행파일(tibero binary)가 존재해야 instance 실행함
- tibero instance : db 관리 (하는 역할, 목적)
  - 파일 (binary, db file - 두가지 종류) →tibero instance, db 에 초점을 맞춤
- tibero server → tibero binary → tibero instance(tibero process로 이루어짐)
- tibero instance - tibero database = 1:n 관계
  - 확장하면 db는 무조건 1개 instance는 여러 개 될 수 있음 → TAC
  - cluster(여러개의 instance 모아서 씀)하는 이유
    1. 가용성 : 최대한 다양한 상태를 유지하고자함.

       → 하나의 문제가 생겨도 다른 것으로 대체 가능
    2. 확장성 : 사용자의 입장에서 사용자가 동시에 사용하려고 보니 instance 하나로는 부족하여 사용자가 100명정도 많다면 instance를 늘려 사용

    → instance 가 한개일때만 다룸
- tibero instance : 사용자의 요청을 받아 데이터 처리해주는 역할을 한다.
  - instance가 연결이 되어있어야함. intrerface라는 연결체 interface driver라는 library file이 있어야 사용자가 전달 tibero binary만들 때 같이 만들어짐 → 접속
  - 여러 단계가 있음.

- 정해진 언어를 통해서 요청할 수 있음. (instance가 이해할 수 있는 언어 : sql) → sql실행
- 사용자 요청(sql 내용)대로 db 파일 찾아 열고 처리를 하여 사용자에게 응답을 함

- whats server(=application server) : 점유율 높은 software server
- listener(process) : instance 접속할 때 신호 알려주는 것 → 접속이 들어오면 듣고 있음(접속 중개하는 역할) - 1개만 필요
  - listener한 port 사용자가 알고 있어야함
  - 어떤 IP를 사용하는지 알고 있어야함
- 작업 process(thread) : 여러 개 있음 (작업 process 1개당 작업 process 10개 들어있음)
  - 사용자가 요청을 하려면 작업 thread와 연결해야함 → 사용자는 여러명이 있음
  - 일반적인 dbms 동작방식 : 1번의 데이터 요청만 가능 → 또다른 사용자가 요청을 동시다발적으로 할 수도 있음.
  - 작업 thread는 서로 영향을 받지 않고 사용자의 요청을 받고 동시에 작업

    ex) ktx예매, 수강신청
- 공유 memory(성능 빨라짐) : 내가 사용하고 싶은 작업 memory에 있다가 사용할 수 있을 때 사용
  - 공유 : 사용자 한명이 데이터를 메모리에서 사용했을 때 사용이 끝나고 난 메모리에 남겨진 데이터를 버리지 않고 다른 사용자가 남겨놓은 메모리를 재사용하는 것
  - buffer cache : 데이터 모음
  - redo log buffer : 데이터 수정 (입력,삭제,수정) → db 변경

    → 변경이력 데이터 : 또 다른 데이터 (a→a' 로 수정한다면 별도로 저장)

    → 여러 redo log 변경

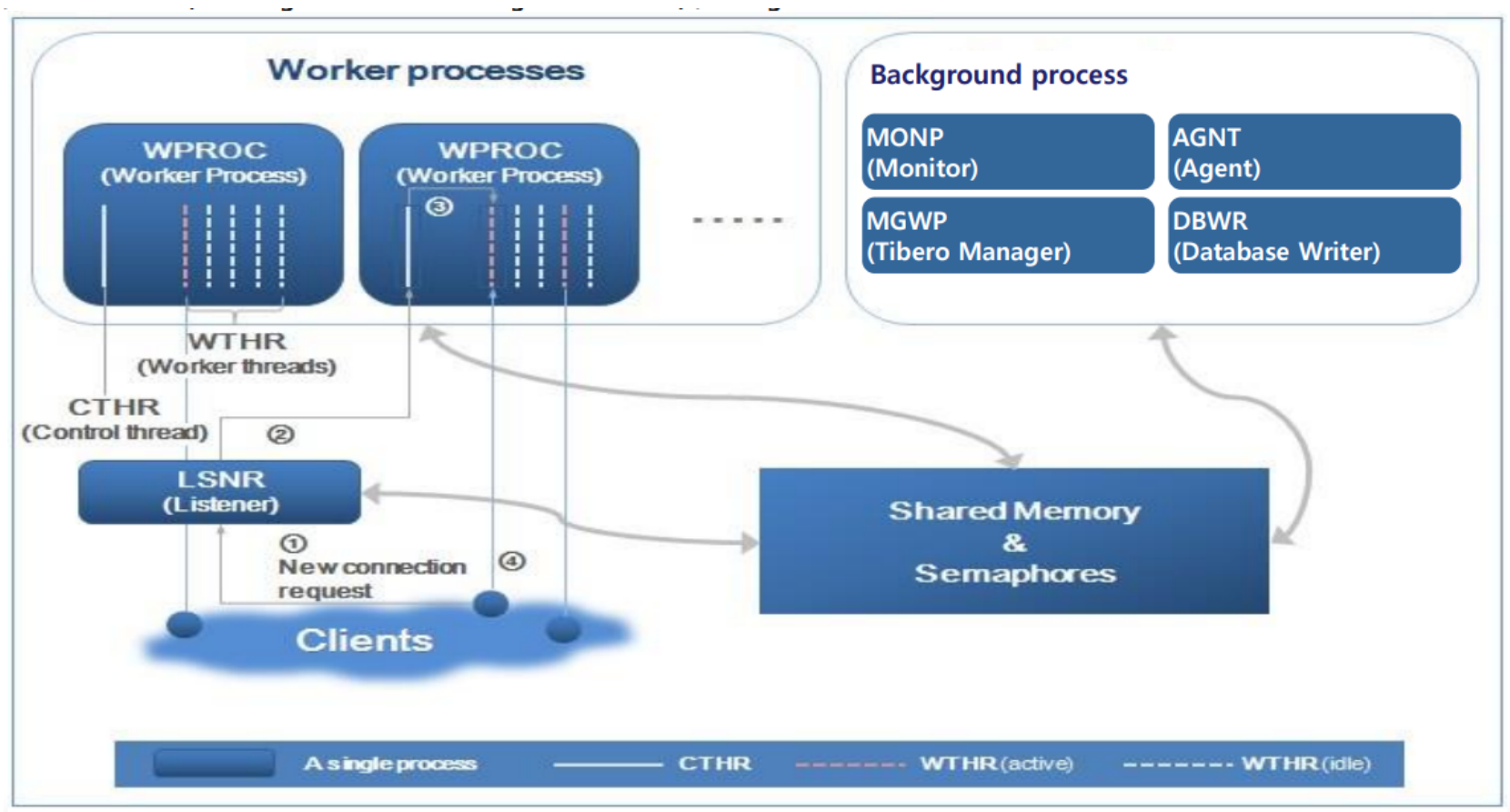
    → redo log가 발생할때마다 즉시 파일에 쓰여짐 → 성능 나쁨

    따라서, redo log buffer가 한꺼번에 처리해줌
  - shared cache : sql 저장됨 (사용자와 연결되는 언어)
    - 실행계획 - thread가 개별적으로 sql을 처리하며 만들어짐
    - sql을 이용하여 실행 계획을 세움.
    - CPU를 이용하여 실행 계획을 세우기 시간이 오래걸림.

      → 지난번 썼던 sql이 필요한다면 가져다쓰면됨.
    - 저장을 하고있다가 실행계획을 다시 만드는 것을 최대한 줄여줌.

▼ Tibero Process

- Tibero 프로세스 구조

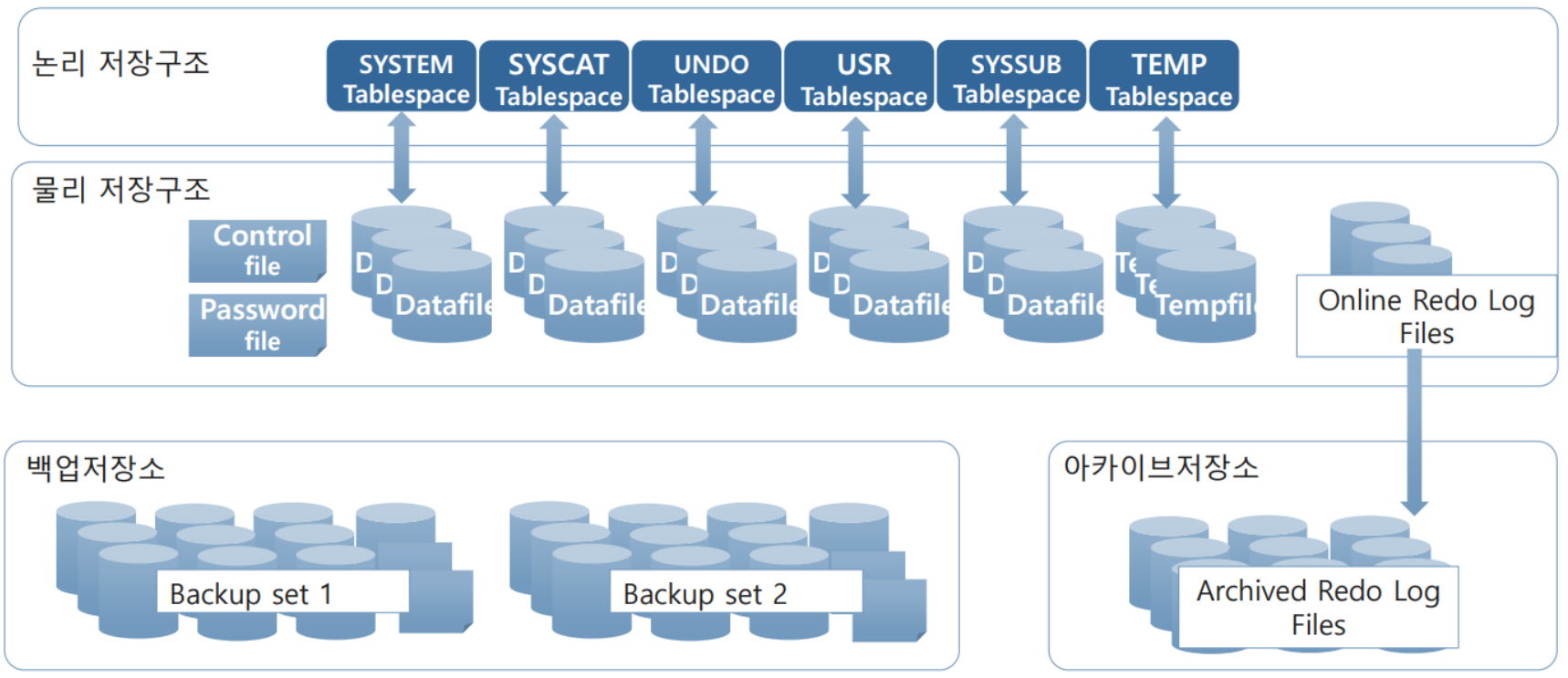


    - client가 접속을 하는 5가지 접속 정보 → 없이는 query 실행 불가능
        1. IP 정보 : 서버들은 interface card 여러개가 있음 각각의 network card에는 IP가 있음 - IP를 알아야 접속을 할 수 있음 → 접속 정보 보냄
        2. port 정보 : listening 하는 port - server 전체에서 unique한 port → listener 접속 process 할당
        3. db name 정보 : 데이터를 담고 있는 file에 붙여진 이름 - 설치과정에서 file이 만들어짐 - tibero에 접속하는 사용자가 알아야하는 정보 → thread를 할당(1개의 control thread가)
        4. user password 정보 - 데이터를 사용하는 사용자들의 이름 → thread와 client 간의 연결 (user명 확인 과정 - table에 들어있는 정보 - thread는 db 접근 가능)-thread가 password이 맞는지 db에 접근하여 table에서 찾음

- 워커프로세스
    - 사용자와 대면 : forground worker process
    - 사용자와 직접 대면하지 않음 : background worker process : 작업 보호 - 안전하게 항상 실행이 되는데 문제가 없도록 할 수 있음 - 외부말고 내부에서 작업을 보호함
    - forground : 사용자가 접속했을 때 접속대상이 되는 process
    - background : 사용자가 접속 요청을 했을 때 listener가 경계하지 않아 접속대상이 되지 않은 process

▼ 2. 데이터베이스 구조

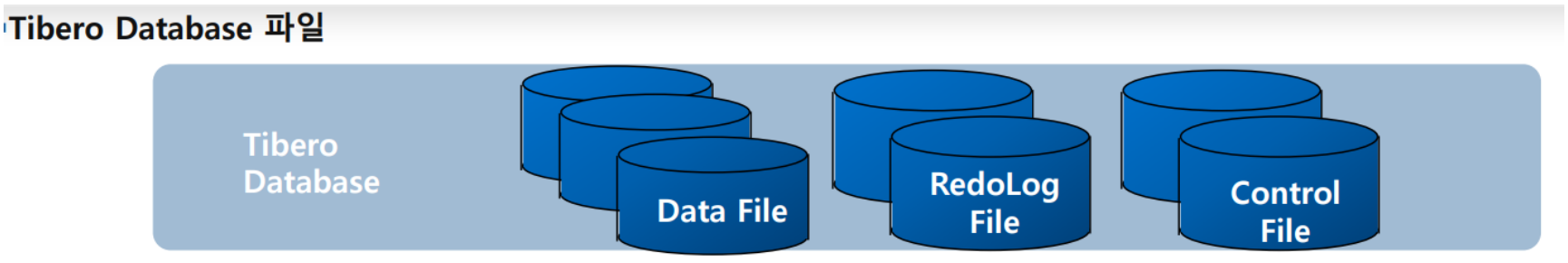
▼ Tibero database 저장소 구조



- backup : db 파일들을 복사하는 것 (묶음 : set)
- backup 저장소 : backup 장비 구매 - db server 장비
- db server 설치하고 운영하려면 file들을 운영하는 장소와 나열할 곳 필요

- 논리 저장구조(=table space)
    - worker process가 있어서 instance에 있는 sql을 던져 어떤 table에 있는지 알려주는 것 (논리 저장구조- 데이터 파일 = 1:N(N은 1이상) 관계) - table space가 있으면 적어도 한개의 data file이 존재해야함
- 물리 저장구조
    - sql 실행하면 물리 저장 구조 만들어짐
    - sql을 통해 처음 db 자체 만들어짐 (tibero 처음 설치할 때 sql 명령어 보내면 만들어짐)

▼ Tibero database 파일

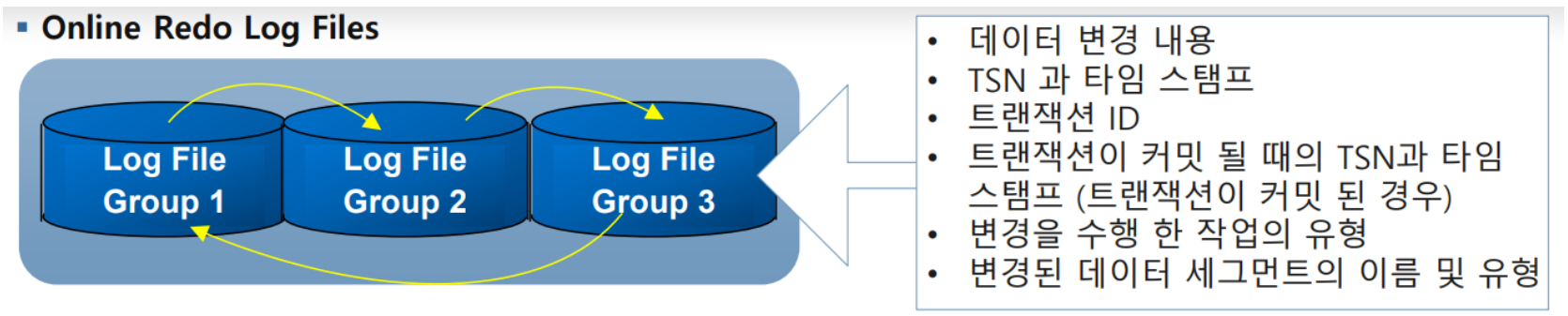


- 설치하는 과정에서 위에 파일 만들어짐
- 3가지 종류의 파일의 차이점
  - data file - 데이터 저장
    - worker thread(데이터 접근하기 위해) data file에 접근
    - 데이터에 접근하려면 정보가 어디에 있는지 알아야함 - control file부터 접근되어야함.
  - redolog file : redo log 저장
  - control file : 물리적인 저장구조(파일들의 목록, 정보)와 물리적인 상태구조 담고 있음
- 접근 순서

  → control file(기준/data file list 보고) → data file(목록에 쓰여있는 file만 접근/control file에 없는 파일이라고 있어도 없는 파일 취급)

▼ Tibero Database 구조 (물리 저장 구조)

- 데이터 파일 헤더
  - check point : data file에 변경(db 변경)을 일으키는 작업 -TSN
  - 사용 공간 : 가장 작은 공간(데이터 불러오는 공간)
  - 데이터를 누군가 사용하고자 할 때 데이터파일의 어떤 부분을 떼어내어 사용하라고 요청
- 공간 활용 방법
  - 데이터 파일 추가 - 파일 하나 resize - 기존의 사용하는 공간 정리



▼ DAY 2 - Tibero installation_1

- instance 설치할 때 JDK 무조건 필요한건 아님
- database(datafile, redologfile,controlfile)
- 설치 준비사항
  - Tibero 환경변수
  - Tibero 파라미터
  - Tibero 데이터베이스 생성항목 (데이터 문장)

<파일 보기>

```
ls -al /tibero/s/*
```

<파일 조회>

```
ls -l /tibero
```

<파일 tibero에 복사하기>

```
cp /tibero/s/*  /tibero
```

<설치 전 3가지 환경변수 설정>

######## TIBERO ENV ########
export TB_HOME=/tibero/tibero6
#binary정보
export TB_SID=tibero
#instance 이름
export TB_PROF_DIR=$TB_HOME/bin/prof
export PATH=.:$TB_HOME/bin:$TB_HOME/client/bin:~/tbinary/monitor:$PATH
export LD_LIBRARY_PATH=$TB_HOME/lib:$TB_HOME/client/lib:$LD_LIBRARY_PATH
export SHLIB_PATH=$LD_LIBRACDRY_PATH:$SHLIB_PATH
export LIBPATH=$LD_LIBRARY_PATH:$LIBPATH

######## TIBERO alias ########
alias tbhome='cd $TB_HOME'
alias tbbin='cd $TB_HOME/bin'
alias tblog='cd $TB_HOME/instance/$TB_SID/log'
alias tbcfg='cd $TB_HOME/config'
alias tbcfgv='vi $TB_HOME/config/$TB_SID.tip'
alias tbcli='cd ${TB_HOME}/client/config'
alias tbcliv='vi ${TB_HOME}/client/config/tbdsn.tbr'
alias tbcliv='vi ${TB_HOME}/client/config/tbnet_alias.tbr'
alias tbdata='cd $TB_HOME/tbdata'
alias tbi='cd ~/tbinary'
alias clean='tbdown clean'
alias dba='tbsql sys/tibero'
alias tm='cd ~/tbinary/monitor;monitor;cd -'

<설치 : 사용자 환경변수 설정>

- 파일 : /home/tibero/.bash_profile
- 준비한 내용을 위의 파일에 추가함.

[20220927_노트 .txt]

<IP관련 설정, 수동으로 변경하기>
제어판\모든 제어판 항목\네트워크 및 공유 센터
-> 어댑터설정변경 -> 이더넷-> 속성-> 이더넷속성 ->
-> 인터넷 프로토콜 버전4(TCP/IPv4)-> 속성
에서 아래(ipconfig)에서 확인한 값을 입력하여 설정함.

cmd ->
ipconfig ->
이더넷 어댑터 이더넷
무선 LAN 어댑터 WIFI

ipconfig 다시 실행하여 출력내용확인
웹브라우저에서 인터넷 접근 잘되는지 확인

< virtualbox 설치>

< virtualbox 에서 가상시스템 가져오기 >

- 머신기본폴더 ==> C:_TIBERO_USB\VM
- MAC주소정책 ==> 모든 네트워크 어댑터 MAC 주소 포함
- 가져오기 이후
  1. 설정-> 네트워크 -> 어댑터2 -> 확인
  2. 설정-> 공유폴더-> C:_TIBERO_USB\s 설정하기
  3. 설정->네트워크 -> 어댑터1->고급 -> 포트 포워딩
     -> 위에서ipconfig 로 확인한 ip를 "호스트ip" 에 넣기
  4. T1 -> 스냅샷 -> 현재상태 -> 마우스오른쪽 버튼-> 찍기
     -> 가상머신의 스냅샷 찍기 -> 확인
- 시작 : T1 머신 선택하고, 시작 버튼 클릭

< Putty 설치 및 리눅스(T1머신) 접속>

IP : 192.168.56.241 사용하여 LOCAL_T1 접속정보 등록

리눅스 로그인시 tibero 유저 사용(패스워드: tibero)

< FileZilla 설치 및 실행 >

- 설치하고, 공유폴더접속(빠른 연결 버튼)을 함

  호스트 : 172.23.14.72
  사용자명: edu
  비밀번호:edu
  포트: 21

< 티베로 설치를 위한 파일 리눅스에 넣기 >

- 윈도우에서 파일을 넣기 C:_TIBERO_USB\s
- 리눅스에서 파일을 가져오기 /tibero/s

< 티베로 설치를 위해 설치 경로로 파일 가져오기 >

- 티베로 설치 위치 : /tibero
- 파일 복사하기
  cp /tibero/s/* /tibero
- 파일 조회
  ls -l /tibero

< 티베로 설치전에 3가지 내용 준비 >

```
######## TIBERO ENV ########
export TB_HOME=/tibero/tibero6
export TB_SID=tibero
```

```
export PATH=.:$TB_HOME/bin:$TB_HOME/client/bin:~/tbinary/monitor:$PATH
export LD_LIBRARY_PATH=$TB_HOME/lib:$TB_HOME/client/lib:$LD_LIBRARY_PATH
```

```
DB_NAME=tibero
LISTENER_PORT=8629
CONTROL_FILES="/tibero/tbdata/tibero/c1.ctl","/tibero/tbdata/tibero/c2.ctl"
DB_CREATE_FILE_DEST=/tibero/tbdata/tibero
LOG_ARCHIVE_DEST=/tibero/tbdata/tibero/arch
MAX_SESSION_COUNT=20
TOTAL_SHM_SIZE=600M
MEMORY_TARGET=1G
```

```
SET ECHO ON
CREATE DATABASE
USER sys IDENTIFIED BY tibero
MAXDATAFILES 256
CHARACTER SET UTF8
NATIONAL CHARACTER SET UTF8
LOGFILE
    GROUP 0 (
    '/tibero/tbdata/tibero/redo1/log01.log',
    '/tibero/tbdata/tibero/redo2/log02.log') SIZE 25M,
    GROUP 1 (
    '/tibero/tbdata/tibero/redo1/log11.log',
    '/tibero/tbdata/tibero/redo2/log12.log') SIZE 25M,
    GROUP 2 (
    '/tibero/tbdata/tibero/redo1/log21.log',
    '/tibero/tbdata/tibero/redo2/log22.log') SIZE 25M,
    GROUP 3 (
    '/tibero/tbdata/tibero/redo1/log31.log',
    '/tibero/tbdata/tibero/redo2/log32.log') SIZE 25M,
    GROUP 4 (
    '/tibero/tbdata/tibero/redo1/log41.log',
    '/tibero/tbdata/tibero/redo2/log42.log') SIZE 25M
MAXLOGFILES 100
MAXLOGMEMBERS 2
ARCHIVELOG
    DATAFILE '/tibero/tbdata/tibero/system001.dtf' SIZE 380M
    AUTOEXTEND ON NEXT 64M MAXSIZE 3G
DEFAULT TEMPORARY TABLESPACE TEMP
    TEMPFILE '/tibero/tbdata/tibero/temp001.dtf' SIZE 380M
    AUTOEXTEND ON NEXT 64M MAXSIZE 3G
    EXTENT MANAGEMENT LOCAL AUTOALLOCATE
UNDO TABLESPACE UNDO
    DATAFILE '/tibero/tbdata/tibero/undo001.dtf' SIZE 380M
    AUTOEXTEND ON NEXT 64M MAXSIZE 3G
    EXTENT MANAGEMENT LOCAL UNIFORM SIZE 128k
DEFAULT TABLESPACE USR
    DATAFILE '/tibero/tbdata/tibero/usr001.dtf' SIZE 100m
    AUTOEXTEND ON NEXT 64m MAXSIZE 3G
SYSSUB
    DATAFILE '/tibero/tbdata/tibero/syssub001.dtf' SIZE 380m
    AUTOEXTEND ON NEXT 64M MAXSIZE 3G ;
```

< 설치 : 사용자 환경변수 설정 >

- 파일 : /home/tibero/.bash_profile

- 준비한 내용을 위의 파일에 추가함

```
vi /home/tibero/.bash_profile
```

```
cat /home/tibero/.bash_profile
```

< 환경변수 적용 >

```
source ~/.bash_profile
```

<환경변수 확인>

```
echo $TB_SID
echo $TB_HOME
```

< 바이너리 설치(압축 해제)>

```
cd /tibero
tar -xvzf  tibero6-bin-*
```

```
ls -al   tibero6
```

< 라이센스 파일 넣기 >

```
  cp /tibero/license.xml    /tibero/tibero6/license
```

```
ls -al  /tibero/tibero6/license/license.xml
```

<초기 환경파일 생성>

```
cd $TB_HOME/config
```

```
./gen_tip.sh
```

<Tibero 파라미터 파일 수정>

```
vi tibero.tip

cat tibero.tip
```

<Network 설정 파일 (tbdsn.tbr) 수정>

```
 vi $TB_HOME/client/config/tbdsn.tbr

 cat $TB_HOME/client/config/tbdsn.tbr
```

< 티베로 인스턴스 NOMOUNT 모드로 기동 >

```
tbboot nomount
```

< tbsql 이용하여 인스턴스 접속>

```
tbsql sys/tibero
```

< Database 생성 쿼리 실행 >

```
   - 직접 실행해도 되지만, 파일에 담아서 실행하기~

  vi  /tibero/credb.sql
  cat /tibero/credb.sql

- 파일에 담기 쿼리 실행

  tbsql sys/tibero

  @/tibero/credb.sql
```

< 티베로 인스턴스 normal 모드로 시작 >

```
    tbboot
```

< Data Dictionary 및 system 패키지 생성 >

```
cd $TB_HOME/scripts

sh system.sh -p1 tibero -p2 syscat -a1 y -a2 y -a3 y -a4 y
```

< 인스턴스 시작 종료 >

```
   tbboot

   tbdown immediate
```

▼ DAY 3 - Tibero installation_2

<설치 조건 작성하고, 조건에 맞게 설치하기>

1. 티베로 바이너리 위치 ($TB_HOME 환경변수 값)

```
echo $TB_SID
echo $TB_HOME

DESC V$DATABASE
```

→ /tibero/0928/tibero6

2. 티베로 인스턴스 이름 ($TB_SID 환경변수 값)

```
DESC V$INSTANCE
SELECT INSTANCE_NAME FROM V$INSTANCE;
```

→ t0928

3. 데이터베이스 이름 (parameter file 에서 정의)

```
SELECT DB_NAME FROM V$INSTANCE;
SELECT NAME FROM V$DATABASE;
```

→ t0928db

4. 리스너 포트 (서비스 포트) - (parameter file에서 정의)

```
SELECT VALUE FROM V$PARAMETERS WHERE NAME=’LISTENER_PORT’
```

- 사용자들이 접속할 때 필요한 port
- 사용자가 접근할 때 데이터베이스를 위한 서비스

  → 9629

5. 기본 캐릭터셋 (DBMS CHARACTER SET) - (create database에서 정의)

```
SELECT VALUE FROM DATABASE_PROPERTIES WHERE NAME='NLS_CHARATERSET';
```

- 데이터베이스에 default 값으로 사용할 문자 집합

→ UTF8

6. Redo log SIZE - (개당 각각의 크기) - (create database에서 정의)

```
SELECT AVG(BYTES)/1024/1024 FROM V$LOG;
```

→ 25MB

7. Redo log group 개수 - (최소 두개)

```
SELECT GROUP#, BYTES/1024/1024 FROM V$LOG;
SELECT COUNT(*) FROM V$LOG;
```

→ 5

8. Redo log 그룹당 멤버갯수 - (그룹옆에 또다른 멤버 나열 가능)

   → 2

```
SELECT AVG(MEMBERS) FROM V$LOG;
```

9. Redo log 멤버(파일) 경로

   → 멤버1 : /tibero/0928/dbs/redo1

   → 멤버2 : /tibero/0928/dbs/redo2

```
DESC V$LOGFILE
####member column에서 redolog file 설정됨####
SELECT GROUP#, MEMBER FROM V$LOGFILE;
!ls -l /tibero/0928/dbs/redo1/log01.log,

SELECT GROUP#, MEMBER FROM V$LOGFILE;
```

10. 테이블 스페이스 SYSTEM 크기

```
SELECT SUM(BYTES)/1024/1024 FROM DBA_DATA_FILES WHERE TABLESPACE_NAME='SYSTEM';
```

```
DESC DBA_DATA_FILES
SELECT BYTES/1024/1024 FROM DBA_DATA_FILES WHERE TABLESPACE_NAME='SYSTEM'
```

→ 380MB

11. 테이블 스페이스 UNDO 크기

    → 380MB

```
SELECT BYTES/1024/1024 FROM DBA_DATA_FILES WHERE TABLESPACE_NAME='UNDO'

SELECT SUM(BYTES)/1024/1024 FROM DBA_DATA_FILES WHERE TABLESPACE_NAME='UNDO';
```

12. 테이블 스페이스 TEMP 크기

    → 380MB

```
SELECT BYTES/1024/1024 FROM DBA_TEMP_FILES WHERE TABLESPACE_NAME='TEMP'

SELECT SUM(BYTES)/1024/1024 FROM DBA_TEMP_FILES WHERE TABLESPACE_NAME='TEMP';
```

13. 테이블 스페이스 USR 크기

    → 100MB

```
SELECT BYTES/1024/1024 FROM DBA_DATA_FILES WHERE TABLESPACE_NAME='USR'

SELECT SUM(BYTES)/1024/1024 FROM DBA_DATA_FILES WHERE TABLESPACE_NAME='USR';
```

14. 테이블 스페이스 SYSSUB 크기

    → 380MB

```
SELECT BYTES/1024/1024 FROM DBA_DATA_FILES WHERE TABLESPACE_NAME='SYSSUB'

SELECT SUM(BYTES)/1024/1024 FROM DBA_DATA_FILES WHERE TABLESPACE_NAME='SYSSUB';
```

15. CONTROLFILE 경로와 이름 -(paramter file에서 정의)

    → /tibero/0928/dbs/control_1/c1.ctl

    → /tibero/0928/dbs/control_2/c2.ctl

```
DESC V$CONTROLFILE
##column name 조회
SELECT NAME FROM V$CONTROLFILE;SELECT VALUE FROM V$PARAMETERS WHERE NAME='DB_CREATE_FILE_DEST
SELECT VALUE FROM V$PARAMETERS WHERE NAME='LOG_ARCHIVE_DEST';
```

16. 패스워드 파일 경로

    → /tibero/0928/dbs/password

```
SELECT VALUE FROM DATABASE_PROPERTIES WHERE NAME='DB_CREATE_FILE_DEST';
!ls -al /tibero/0928/dbs/password/
```

17. 티베로 인스턴스가 사용하는 전체 메모리 크기

    → 2300MB

```
SELECT VALUE/1024/1024 FROM V$PARAMETERS WHERE NAME=’MEMORY_TARGET’;
```

18. 티베로 인스턴스가 사용하는 공유 메모리 크기

    → 1300GB

```
SELECT VALUE/1024/1024 FROM V$PARAMETERS WHERE NAME=’TOTAL_SIZE’;
```

19. DB SESSION 최대 갯수 (=worker thread)

    → 30

```
SELECT VALUE/1024/1024 FROM V$PARAMETERS WHERE NAME='MAX_SESSION_COUNT';
```

20. DBMS LOG MODE

    → ARCHIVELOG

```
SELECT LOG_MODE FROM V$DATABASE;
```

21. 아카이브로그 경로

    → /tibero/0928/dbs/arch

```
SELECT VALUE FROM V$PARAMETERS WHERE NAME='LOG_ACHIVE_DEST';
```

22. (위에서 정의한 기본테이블스페이스에서 사용하는)

    데이터파일들의 경로

    → /tibero/0928/dbs/dbs

```
DESC DBA_DATA_FILES
COL TABLE_SIZE_NAME FOR A20
SELECT TABLESPACE_NAME, FILE_NAME FROM DBA_DATA_FILES;
#temp file
SELECT TABLESPACE_NAME, FILE_NAME FROM DBA_TEMP_FILES;
```

<.bash_profile>

```
######## TIBERO ENV ########
export TB_HOME=/tibero/0928/tibero6
export TB_SID=t0928
export PATH=.:$TB_HOME/bin:$TB_HOME/client/bin:~/tbinary/monitor:$PATH
export LD_LIBRARY_PATH=$TB_HOME/lib:$TB_HOME/client/lib:$LD_LIBRARY_PATH
```

<t0928.tip>

```
DB_NAME=t0928db
LISTENER_PORT=9629
CONTROL_FILES="/tibero/0928/dbs/control_1/c1.ctl","/tibero/0928/dbs/control_2/c2.ctl"
DB_CREATE_FILE_DEST=/tibero/0928/dbs/password
LOG_ARCHIVE_DEST=/tibero/0928/dbs/arch
MAX_SESSION_COUNT=30
TOTAL_SHM_SIZE=1300M
MEMORY_TARGET=2300M
```

<credb2.tip>

```
SET ECHO ON
CREATE DATABASE
USER sys IDENTIFIED BY tibero
MAXDATAFILES 256
CHARACTER SET UTF8
NATIONAL CHARACTER SET UTF8
LOGFILE
    GROUP0 ('/tibero/0928/dbs/redo1/log01.log','/tibero/0928/dbs/redo2/log02.log') SIZE 25M,
    GROUP1 ('/tibero/0928/dbs/redo1/log11.log','/tibero/0928/dbs/redo2/log12.log') SIZE 25M,
    GROUP2 ('/tibero/0928/dbs/redo1/log21.log','/tibero/0928/dbs/redo2/log22.log') SIZE 25M,
    GROUP3 ('/tibero/0928/dbs/redo1/log31.log','/tibero/0928/dbs/redo2/log32.log') SIZE 25M,
    GROUP4 ('/tibero/0928/dbs/redo1/log41.log','/tibero/0928/dbs/redo2/log42.log') SIZE 25M
```

```
MAXLOGFILES 100
MAXLOGMEMBERS 2
ARCHIVELOG
    DATAFILE '/tibero/0928/dbs/system001.dtf' SIZE 380M
    AUTOEXTEND ON NEXT 64M MAXSIZE 3G
DEFAULT TEMPORARY TABLESPACE TEMP
    TEMPFILE '/tibero/0928/dbs/temp001.dtf' SIZE 380M
    AUTOEXTEND ON NEXT 64M MAXSIZE 3G
    EXTENT MANAGEMENT LOCAL AUTOALLOCATE
UNDO TABLESPACE UNDO
    DATAFILE '/tibero/0928/dbs/undo001.dtf' SIZE 380M
    AUTOEXTEND ON NEXT 64M MAXSIZE 3G
    EXTENT MANAGEMENT LOCAL UNIFORM SIZE 128k
DEFAULT TABLESPACE USR
    DATAFILE '/tibero/0928/dbs/usr001.dtf' SIZE 100m
    AUTOEXTEND ON NEXT 64m MAXSIZE 3G
SYSSUB
    DATAFILE '/tibero/0928/dbs/syssub001.dtf' SIZE 380m
    AUTOEXTEND ON NEXT 64M MAXSIZE 3G ;
```

Q. DB 캐릭터셋을 조건에 맞지 않게 정의하여 설치하였다. 고치는 방법은?

→ DB를 새로 생성하면서 제대로 설정하여야 한다.

1. 인스턴스 종료
2. DATABASE 삭제(controlfile, datafile, redolog file, passwd
   → 만약 archivelogfile 있으면 그 파일 포함)
3. tbboot nomount
4. tbsql sys/tibero
5. CREATE DATABASE ~ 문장 실행
   (** 캐릭터셋 항목을 제대로 포함시켜야 함)
6. tbboot
7. system.sh 실행

▼ Day 4 - Tibero Administration

- class 속성

1.2. Class Attributes

This section explains the class attributes necessary for users to understand the information configured in the initialization parameter file.

Group	Class	Description
Static/Dynamic	Static	A parameter that cannot be modified during system operation.
	Dynamic	A parameter that can be modified during system operation.
Persistent/Adjustable	Persistent	A parameter that cannot be modified after database creation.
	Adjustable	A parameter that can be modified after database creation.
Mandatory/Optional	Mandatory	A parameter required to start up a system. If these parameter are not defined in the initialization parameter file, the system cannot be booted.
	Optional	A parameter that uses a default value if not defined in the initialization parameter.
System/Session	System	A parameter that the entire system shares.
	Session	A parameter that can be configured separately for each session.

- DB_NAME

- Parameter Format

Property	Description
Parameter Type	String
Default Value	""
Class	Mandatory, Persistent, Static, System
Range	Limited to 40 characters

→ persistent (수정하지 않아도 됨) ↔ adjustable (수정 가능)

- USER 조회

```
DESC DBA_USERS;
```

- 사용자 관리 (user 6개)

계정	설명
SYS	데이터베이스 관리를 위한 계정으로서 시스템 패키지, 동의어, 사용자, 역할, 가상 테이블, 시퀀스, 동적 뷰 등을 생성하고 관리한다.
SYSCAT	데이터베이스 관리를 위한 정적 카탈로그 뷰를 생성하고 관리하는 계정이다.
OUTLN	동일한 SQL을 수행할 때 항상 같은 질의 플랜(plan)으로 수행될 수 있게 관련 힌트(hint)를 저장하는 등의 일을 하는 계정이다.
SYSGIS	GIS(Geographic Information System)와 관련된 테이블 생성 및 관리를 하는 계정이다.
TIBERO	CONNECT, RESOURCE, DBA 역할이 부여된 샘플 사용자 계정이다.
TIBERO1	CONNECT, RESOURCE, DBA 역할이 부여된 샘플 사용자 계정이다.

→ user는 schema 소유

→ user 존재(schema 있음)

→ schema는 객체 관리하여 user에게 권한이 있는 table(Dictionary object) 만들기 가능

- 사용자 계정

→ Login 필요

→ db내의 table 안에 password 있음

시험 관련 내용

1. 설치
   설치에 대한 요건 - 설치 실습
2. 점검
   설치를 한것에 대해서 조회(점검) - 설치가 제대로 됐는지 확인 - 어떤 식으로 점검을 했었는지(SELECT query) - query 작성 어떻게 해야하는지

```
SET ECHO ON

SELECT INSTANCE_NAME FROM V$INSTANCE;

QUIT
```

- 점검 query (인스턴스 이름 확인)

```
tbsql sys/tibero @sample.sql
```

- 접속 후 실행하고자 하는 sql query를 @뒤에 넣기
- 언급한 내용(자료)들 추가됨
- 변형이 되어 나올 수도 있음 (새로운 내용 추가)

CREATE TABLE TBL0930 (TBL0930_ID NUMBER PRIMARY KEY, name VARCHAR(20) UNIQUE)

SELECT * FROM ALL_TAB_COMMENTS WHERE TABLE_NAME = 'TBL0930';

[tibero@T1:/tibero/0930/tibero6/scripts]

SELECT TABLESPACE_NAME, FILE_NAME, BYTES FROM dba_data_files;

▼ DAY5 - Database Link_1

-Tibero6 Online Manual (관리자 참조) & 구성가이드pdf

Database Link

- Remote Database
  - 데이터 처리를 위한 데이터베이스 따로 만듦 (내부 업무용 데이터 들어있음)
  - 비용절감을 위해 mysql 사용/ oracle 사용 등 여러개 사용
  - 공통분모와 관련된 데이터 뽑아 같이 연결하여 봐야함
- 사용자가 db server 와 interface 하려면 → interface driver (ex) library file)
  - 스키마 객체 중 하나 (CREATE DATABASE LINK ~~~~;)
  - 특정 user 가 소유하고 있게 되는 것
  - create database link query 실행하여 만드는 것
    - local database에서 실행 : database link가 local database에 query를 통해 만들어짐
- gateway (모든 db에 적용됨) - query를 remote database에 전달해줌
  - tibero가 아닌 다른 db에 query를 던지려면 접속을 할 interface driver 필요
  - gateway에는 interface driver가 들어가서 접속할 수 있도록 함
  - 직접 interface driver를 핸들링할 순 없으므로 gateway를 사용하여 실행
  - gateway, interface driver 모두 file → 모두 binary안에 들어있음
  - db 객체 어디에 있는지 → gateway는 tibero(local db) binary안에 client directory안에 있음
  - 같은 db로 전달할때는 별도로 gateway를 설치할 필요 없음 → 바로 query 실행 가능
    - local db에 이미 tibero gateway와 관련된 정보 있기 때문
    - db에 정보 던지려면 접속 정보에 대한 설정 필요

```
create public database link public_tibero using 'remote_2';
```

→ remote_2(접속정보) 를 이용하여 public_tibero라는 db만든다

→ remote_2는 연결할 db 를 가리키는 이름으로 tbdsn.tbr 파일에 해당 db 연결 정보가 저장되어야 함. (ip, post, db name)

- Public DB Link
  - 누구나 다 사용(접근) 가능 → 다른 user들도 db 이용 가능
- Private DB Link
  - db link객체는 테이블처럼 특정 스키마에 속함

- user create했으므로 그 user의 스키마에 속함. → 그 user만 사용

DBMS 벤더명	게이트웨이 바이너리명	프로그래밍 언어	DBMS 버전
Oracle	gw4orcl	C	Oracle 9i, 10g, 11g
DB2	gw4db2	C	DB2 V8, DB2 9, DB2 9.5
MS-SQL SERVER	tbgateway.jar	Java	MS-SQL SERVER 2000, 2005, 2008
Adaptive Server Enterprise(Sybase)	tbgateway.jar	Java	Sybase SQL Server 10.0.2 or later
GREENPLUM	tbgateway.jar	Java	GREENPLUM or PostgreSQL

- 멀티 스레드 서버 방식(listener 방식)
  - <tbdsn.tbr> : gateway에 접속하도록함

```
ora_link_remote=(
                  (GATEWAY=(LISTENER=(HOST=12.34.56.78)
                                     (PORT=9999))
                           (TARGET=orcl)
                           (TX_MODE=GLOBAL))
)
```

  - ora_link_remote : 뭐라고 지정하던 상관 x → db link 객체 이름(using 절에 들어가는)

```
create database link remote_tibero connect to user1
identified by 'password' using 'remote_1';
```

  - 실행 및 종료

원격에 있는 게이트웨이를 사용하기 위해서는 먼저 원격에 있는 게이트웨이를 실행시켜야 한다.

다음은 원격에서 Oracle 서버의 게이트웨이를 실행시키는 예이다.

```
$ gw4orcl
```

다음은 원격에서 MS-SQL 서버의 게이트웨이를 실행시키는 예이다.

```
$ tbgw
```

  - 환경변수(bash_profile에 추가)
    - 게이트웨이는 기본적으로 TBGW_HOME 환경변수를 통해 설정 파일을 읽고 로그 파일을 기록한다.
    - TBGW_HOME 환경변수가 설정되어 있지 않은 경우에는 기본값은 '${TB_HOME}/client/gateway'이다. Windows 환경에서는 기본값이 '%TB_HOME%\client\gateway'로 설정
    - directory 구조 (사용자가 직접 지정해주어야함)

```
$TBGW_HOME
    |--- DBMS 벤더명
            |--- config
            |      |--- tbgw.cfg
            |--- log
                   |--- 게이트웨이의 로그 파일
```

→ 게이트웨이의 로그파일 빼고는 사용자가 설치과정에서 모두 만들어야함

<tbgw.cfg>

```
LOG_DIR=${TBGW_HOME}/{DBMS 벤더명}/log
LOG_LVL=2
LISTENER_PORT=9999
```

```
MAX_LOG_SIZE=20k
FETCH_SIZE=32k
```

- V$DBLINK

```
SQL> select * from employee@remote_tibero;

 ID     NAME
--- ---------
  1      KIM
  2      LEE
  3     HONG
```

- table이 emplyee만(@없이) local db :sys 사용자
- employee@ 뒤에 table이름은 remote db안에 있는 employee 테이블

```
SQL> select * from V$DBLINK;

DB_LINK         OWNER_ID   OPEN_CURSORS   IN_TRANSACTION  HETEROGENEOUS
--------------  ---------  -------------  --------------  --------------
REMOTE_TIBERO         15              0    YES             NO

COMMIT_POINT_STRENGTH
---------------------
                    1

1 row selected.
```

현재 remote_tibero의 소유자의 ID는 15번이고, 현재 열려있는 커서는 없으며 트랜잭션을 수행하고 있다. 동일한 Tibero 서버에 대한 데이터베이스 링크이며, 원격 데이터베이스의 commit point strength는 1이다.

- 데이터베이스 링크 종료하려면

→ commit 문을 통해 해당 트랜잭션을 종료 (commit;) 후 링크 종료

```
alter session close database link remote_tibero;
```

```
select * from V$DBLINK;
```

DB_Link 설치과정

- 오라클 클라이언드 위치
  → /tibero에서 압축해제하여 /tibero/instantclient_11_2 디렉토리가 만들어짐
- 게이트웨이 위치
  → /tibero에서 mkdir tbgateway 실행하여 /tibero/tbgateway 디렉토리가 만들어지고 이를 $TBGW_HOME으로 사용한다.

▼ DAY6 - Database Link_2, Tools

DATABASE LINK

###DBLINK (T to T)

Database link (t to t) 객체를 이용하여, A 데이터베이스에서 테이블을 가져오려고 함.

** A 데이터베이스 접속정보 (본인의 현재 TIBERO)

IP :

PORT :

SID :

USER : SYS

PASSWORD : TIBERO

문제1) 수강생 DB에 새로운 스키마(이름: EDU_DBLINK, 암호 : EDU_DBLINK)를 생성하고, 이곳에 DATABASE LINK 객체(이름 : EDU_TLINK)를 만드세요.

문제2) DATABASE LINK를 이용하여 A 데이터베이스의 ALL_USERS를 수강생 DB의 EDU_DBLINK 스키마로 가져오세요 (이름은 MY_USERS, 데이터 포함)

원격지 DB에 접근하여 (TLINK DB링크 이용) 거기에 있는 players 테이블의 선수데이터중에서 젊은 (25세이하 나이) 선수들의 데이터만 가져와서 테이블을 생성하시오.

```
## 테이블을 가져올때 사용하는 CREATE TABLE ~ AS SELECT ~ 쿼리 사용예)
CREATE TABLE young_players ( player_id, name, alais, age, team )
AS
SELECT player_id, name, nickname, age, team
FROM players
WHERE age <= 25;
```

```
## CREATE TABLE ~ AS SELECT ~ 사용하여 DB링크 건너편의 원격지 DB의 테이블, 데이터를 가져오는 방법의 예)
CREATE TABLE young_players ( player_id, name, alais, age, team )
AS
SELECT player_id, name, nickname, age, team
FROM players@TLINLK
WHERE age <= 25;
```

→ 데이터만 가져오고 싶으면 CREATE → INSERT 로 변경

```
### 데이터만 가져오는(DB링크 사용) 경우의 예)
INSERT INTO young_players (player_id,name,alais,age,team)
AS
SELECT player_id, name, nickname, age, team
FROM players@TLINK
WHERE age ≤ 25;
```

* 위의 조건대로 작업완료후 아래와 같이 실행시, 생성된 테이블로 부터 조회가 되어야 함.

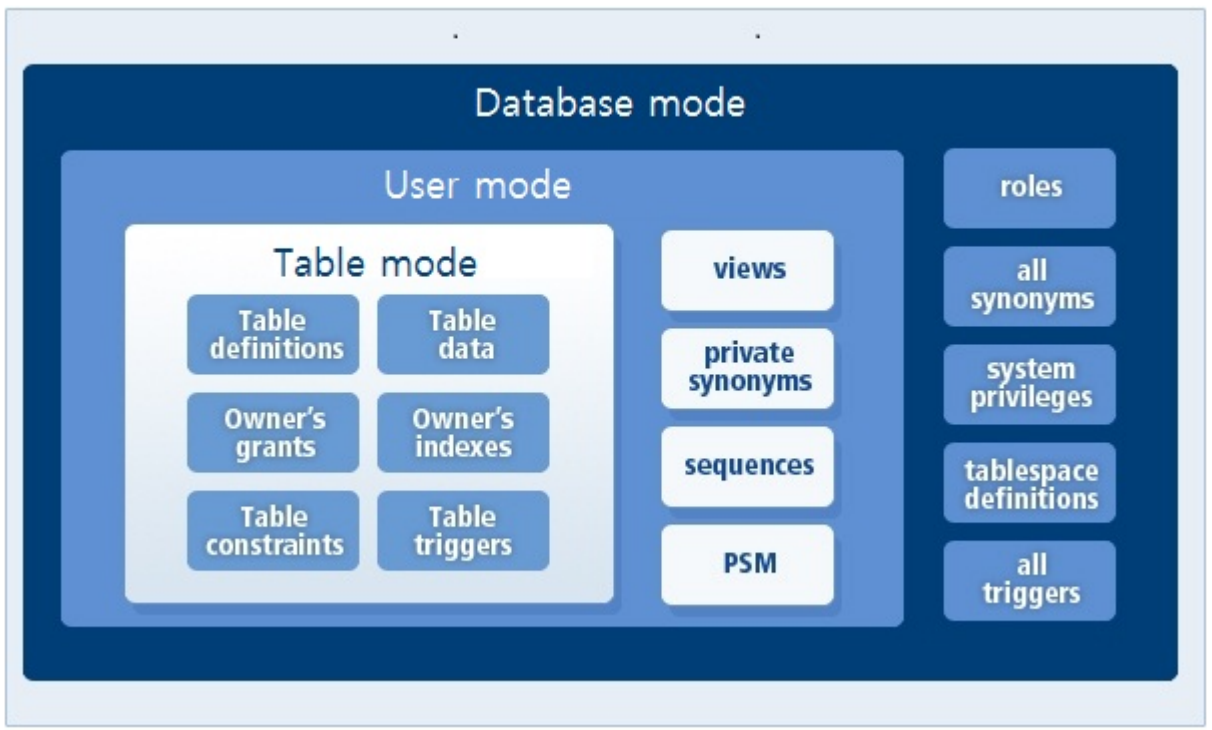
```
tbsql EDU_DBLINK/EDU_DBLINK
SELECT * FROM MY_USERS;
```

Tibero Tools

tibero_tools.pdf & Online manual - 유틸리티 안내서

- tbexport

[그림 3.1] Export 모드



- 고정된 레코드 형태

```
example.ctl:
     LOAD DATA
     INFILE  'example.dat'
     LOGFILE 'example.log'
     BADFILE 'example.bad'
     APPEND
     INTO TABLE EMP
     LINES FIX 12
     (
         empno position (01:04),
         ename position (06:12)
     )

     example.dat:
     7922 MILLER 7777 KKS    7839 KING   7934 MILLER 7566 JONES
```

→ 6번째 자리부터 12번째 자리까지 두번째 필드 ename

```
ename position (06:12)
```

- 분리된 레코드 형태

```
example.ctl:
   LOAD DATA
   INFILE  'example.dat'
   LOGFILE 'example.log'
   BADFILE 'example.bad'
   APPEND
   INTO TABLE emp
   FIELDS TERMINATED BY ','
          OPTIONALLY ENCLOSED BY '"'
          ESCAPED BY '\\'
   LINES TERMINATED BY '\n'
   (
       empno,
       ename,
       job,
       mgr,
       hiredate,
       sal,
       comm,
       deptno
   )
```

```
example.dat:
   7654, "Martin", "Sales",7698,1981/10/28,1312.50,3,10
   7782, "\,Clark","Manager" ,7839, 1981/01/11 ,2572.50,10,20
   7839, "King",President,,1981/11/17,5500.00,,10
   7934,"Miller","Clerk",7782 ,1977/10/12,920.00,,10
   7566, "Jones",Manager" ,7839, 1981/04/02,3123.75,,20
   7658, "Chan", Ana lyst, 7566,1982/05/03,3450,,20
```

- 오류파일 (로딩이 안된 실패한 레코드파일)
  - 로딩안된 원인 찾아 수정 후 다시 로드

- tbloader

https://technet.tmaxsoft.com/upload/download/online/tibero/pver-20150504-000001/tibero_util/chapter_tbloader.html#d5e9273

1. 테이블을 생성한다(모든 예제에서 공통 사항이다).

```
CREATE TABLE MEMBER
(
    ID        NUMBER(4) NOT NULL,
    NAME      VARCHAR2(10),
    JOB       VARCHAR2(9),
    BIRTHDATE DATE,
    CITY      VARCHAR2(10),
    PICTURE   BLOB,
    AGE       NUMBER(3),
    RESUME    CLOB
);

CREATE TABLE CLUB
(
    ID       NUMBER(6) NOT NULL,
    NAME     VARCHAR2(10),
    MASTERID NUMBER(4)
);
```

2. 컨트롤 파일을 작성한다. (control.ctl)

→ loader directory에 vi 작성

```
LOAD DATA
INFILE './data.dat'
APPEND
INTO TABLE club
FIELDS TERMINATED BY ','
       OPTIONALLY ENCLOSED BY '"'
LINES TERMINATED BY '|\n'
IGNORE 1 LINES
(
 id integer external,
 name,
 masterid integer external
)
```

3. 데이터 파일을 작성한다. (data.dat)

```
id     name      masterid|
111111,FC-SNIFER,2345|
dkkkkkkkkkk|
111112,"DOCTOR CLUBE ZZANG",2222|
111113,"ARTLOVE",3333|
111114,FINANCE,1235|
111115,"DANCE MANIA",2456|
111116,"MUHANZILZU",2378|
111117,"INT'L",5555
```

4. tbLoader 유틸리티를 실행한다.

```
tbloader userid=tibero/tmax@tibero control=./control.ctl
```

5. 로그 파일과 오류 파일을 확인한다.

```
cat control.log
cat data.bad
```

▼ DAY7 - Backup_Recovery

Offline Backup

##백업하기 위한 정상상태인지 확인하는 법

```
tbdown clean
tbboot
tbsql sys/tibero

SQL> select name from v$database;

NAME
----------------------------------------
tibero

1 row selected.

SQL> select username from dba_users;

USERNAME
--------------------------------------------------------------------------------
SYS
SYSCAT
SYSGIS
OUTLN
TIBERO
TIBERO1
EDU_DBLINK

7 rows selected.
```

##datafile 조회

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

```
NAME
--------------------------------------------------------------------------------
/tibero/tbdata/tibero/system001.dtf
/tibero/tbdata/tibero/undo001.dtf
/tibero/tbdata/tibero/usr001.dtf
/tibero/tbdata/tibero/syssub001.dtf

4 rows selected.

SQL> select file_name from dba_data_files;

FILE_NAME
--------------------------------------------------------------------------------
/tibero/tbdata/tibero/system001.dtf
/tibero/tbdata/tibero/undo001.dtf
/tibero/tbdata/tibero/usr001.dtf
/tibero/tbdata/tibero/syssub001.dtf

4 rows selected.
```

##logfile 조회

```
SQL> SELECT MEMBER FROM V$LOGFILE;

MEMBER
--------------------------------------------------------------------------------
/tibero/tbdata/tibero/log01.log
/tibero/tbdata/tibero/log11.log
/tibero/tbdata/tibero/log21.log
```

##log_mode 조회

```
SQL> SELECT LOG_MODE FROM V$DATABASE;

LOG_MODE
------------
NOARCHIVELOG

1 row selected.
```

##NOARCHIVELOG → ARCHIVELOG 모드로 변경

```
tbdown immediate
tbboot mount  ##mount 모드에서 변경해야됨

tbsql sys/tibero
SQL> ALTER DATABASE ARCHIVELOG;

Database altered.

SQL> QUIT
Disconnected.

tbdown
tbboot
```

```
tbsql sys/tibero
SQL> SELECT LOG_MODE FROM V$DATABASE;

LOG_MODE
------------
ARCHIVELOG

1 row selected.

SQL> ALTER SYSTEM SWITCH LOGFILE;

System altered.

SQL> SELECT NAME FROM V$ARCHIVE_DEST_FILES;

NAME
-------------------------------------------------------------------------------
/tibero/tbdata/tibero/arch/log-t0-r0-s3.arc

1 row selected.

SQL> SELECT NAME FROM V$CONTROLFILE;

NAME
-------------------------------------------------------------------------------
/tibero/tbdata/tibero/c1.ctl
/tibero/tbdata/tibero/c2.ctl

2 rows selected.

SQL> tbdown immediate
DB instance was terminated.
SQL>
SQL> quit
```

##백업 디렉토리 생성 후 원래 파일 복사 (tbdown immediate 상태에서)

```
[tibero@T1:/]$ mkdir /tibero/s/backup_off_1014
[tibero@T1:/]$
[tibero@T1:/]$
[tibero@T1:/]$ cp /tibero/tbdata/tibero/*.dtf  /tibero/s/backup_off_1014
[tibero@T1:/]$
[tibero@T1:/]$ cp /tibero/tbdata/tibero/log*.log   /tibero/s/backup_off_1014
[tibero@T1:/]$
[tibero@T1:/]$ /tibero/tbdata/tibero/arch/log-*.arc    /tibero/s/backup_off_1014
-bash: /tibero/tbdata/tibero/arch/log-t0-r0-s3.arc: Permission denied
[tibero@T1:/]$
[tibero@T1:/]$ cp /tibero/tbdata/tibero/arch/log-*.arc    /tibero/s/backup_off_1014
[tibero@T1:/]$ cp /tibero/tbdata/tibero/*.ctl   /tibero/s/backup_off_1014
[tibero@T1:/]$
[tibero@T1:/]$
[tibero@T1:/]$ ls -al /tibero/s/backup_off_1014
total 710632
drwxrwx--- 1 root vboxsf      4096 Oct  6 10:20 .
drwxrwx--- 1 root vboxsf         0 Oct  6 10:14 ..
```

```
-rwxrwx--- 1 root vboxsf  22822912 Oct  6 10:20 c1.ctl
-rwxrwx--- 1 root vboxsf  22822912 Oct  6 10:20 c2.ctl
-rwxrwx--- 1 root vboxsf  52428800 Oct  6 10:18 log01.log
-rwxrwx--- 1 root vboxsf  52428800 Oct  6 10:18 log11.log
-rwxrwx--- 1 root vboxsf  52428800 Oct  6 10:18 log21.log
-rwxrwx--- 1 root vboxsf    461824 Oct  6 10:19 log-t0-r0-s3.arc
-rwxrwx--- 1 root vboxsf  52428800 Oct  6 10:16 syssub001.dtf
-rwxrwx--- 1 root vboxsf 104857600 Oct  6 10:16 system001.dtf
-rwxrwx--- 1 root vboxsf 104857600 Oct  6 10:16 temp001.dtf
-rwxrwx--- 1 root vboxsf 209715200 Oct  6 10:16 undo001.dtf
-rwxrwx--- 1 root vboxsf  52428800 Oct  6 10:16 usr001.dtf
[tibero@T1:/]$
[tibero@T1:/]$
[tibero@T1:/]$ ls -al $TB_HOME/config/$TB_SID.tip
-rw-r--r-- 1 tibero dba 255 Sep 27 16:21 /tibero/tibero6/config/tibero.tip
[tibero@T1:/]$
[tibero@T1:/]$
[tibero@T1:/]$ cp /tibero/tibero6/config/tibero.tip   /tibero/s/backup_off_1014
[tibero@T1:/]$
[tibero@T1:/]$ cat /tibero/tibero6/config/tibero.tip | grep DB_CREATE_FILES_DEST
[tibero@T1:/]$ cat /tibero/tibero6/config/tibero.tip | grep DB_CREATE_FILE_DEST
DB_CREATE_FILE_DEST=/tibero/tbdata/tibero
[tibero@T1:/]$
[tibero@T1:/]$ ls -al /tibero/tbdata/tibero/.passwd
-r-------- 1 tibero dba 24 Sep 27 16:55 /tibero/tbdata/tibero/.passwd
[tibero@T1:/]$
[tibero@T1:/]$ cp /tibero/tbdata/tibero/.passwd   /tibero/s/backup_off_1014
[tibero@T1:/]$
[tibero@T1:/]$ ls -al /tibero/s/backup_off_1014
total 710633
drwxrwx--- 1 root vboxsf      4096 Oct  6 11:03 .
drwxrwx--- 1 root vboxsf         0 Oct  6 10:14 ..
-rwxrwx--- 1 root vboxsf  22822912 Oct  6 10:20 c1.ctl
-rwxrwx--- 1 root vboxsf  22822912 Oct  6 10:20 c2.ctl
-rwxrwx--- 1 root vboxsf  52428800 Oct  6 10:18 log01.log
-rwxrwx--- 1 root vboxsf  52428800 Oct  6 10:18 log11.log
-rwxrwx--- 1 root vboxsf  52428800 Oct  6 10:18 log21.log
-rwxrwx--- 1 root vboxsf    461824 Oct  6 10:19 log-t0-r0-s3.arc
-rwxrwx--- 1 root vboxsf        24 Oct  6 11:03 .passwd
-rwxrwx--- 1 root vboxsf  52428800 Oct  6 10:16 syssub001.dtf
-rwxrwx--- 1 root vboxsf 104857600 Oct  6 10:16 system001.dtf
-rwxrwx--- 1 root vboxsf 104857600 Oct  6 10:16 temp001.dtf
-rwxrwx--- 1 root vboxsf       255 Oct  6 11:01 tibero.tip
-rwxrwx--- 1 root vboxsf 209715200 Oct  6 10:16 undo001.dtf
-rwxrwx--- 1 root vboxsf  52428800 Oct  6 10:16 usr001.dtf
[tibero@T1:/]$
[tibero@T1:/]$
[tibero@T1:/]$
[tibero@T1:/]$
[tibero@T1:/]$ tbboot
Listener port = 8629

Tibero 6

TmaxData Corporation Copyright (c) 2008-. All rights reserved.
Tibero instance started up (NORMAL mode).
```

tbdown abnormal (tibero)

```
[tibero@T1:/]$ tbdown abnormal
Tibero instance terminated (ABNORMAL mode).

[tibero@T1:/]$
[tibero@T1:/]$
[tibero@T1:/]$
[tibero@T1:/]$
[tibero@T1:/]$
[tibero@T1:/]$ ls -al /tibero/tbdata/tibero/*.dtf
-rw------- 1 tibero dba  52428800 Oct  6 11:04 /tibero/tbdata/tibero/syssub001.dtf
-rw------- 1 tibero dba 104857600 Oct  6 11:06 /tibero/tbdata/tibero/system001.dtf
-rw------- 1 tibero dba 104857600 Sep 27 16:55 /tibero/tbdata/tibero/temp001.dtf
-rw------- 1 tibero dba 209715200 Oct  6 11:06 /tibero/tbdata/tibero/undo001.dtf
-rw------- 1 tibero dba  52428800 Oct  6 11:04 /tibero/tbdata/tibero/usr001.dtf
[tibero@T1:/]$
[tibero@T1:/]$
[tibero@T1:/]$ rm /tibero/tbdata/tibero/*.dtf
[tibero@T1:/]$
[tibero@T1:/]$ ls -al /tibero/tbdata/tibero/log*.log
-rw------- 1 tibero dba 52428800 Oct  6 11:07 /tibero/tbdata/tibero/log01.log
-rw------- 1 tibero dba 52428800 Oct  6 11:04 /tibero/tbdata/tibero/log11.log
-rw------- 1 tibero dba 52428800 Oct  6 11:04 /tibero/tbdata/tibero/log21.log
[tibero@T1:/]$
[tibero@T1:/]$
[tibero@T1:/]$ rm /tibero/tbdata/tibero/log*.log
[tibero@T1:/]$
[tibero@T1:/]$ ls -al /tibero/tbdata/tibero/arch/log-*.arc
-rw------- 1 tibero dba 461824 Oct  6 10:09 /tibero/tbdata/tibero/arch/log-t0-r0-s3   .arc
[tibero@T1:/]$
[tibero@T1:/]$ rm /tibero/tbdata/tibero/arch/log-*.arc
[tibero@T1:/]$
[tibero@T1:/]$ ls -al /tibero/tbdata/tibero/*.ctl
-rw------- 1 tibero dba 22822912 Oct  6 11:07 /tibero/tbdata/tibero/c1.ctl
-rw------- 1 tibero dba 22822912 Oct  6 11:07 /tibero/tbdata/tibero/c2.ctl
[tibero@T1:/]$
[tibero@T1:/]$
[tibero@T1:/]$ rm /tibero/tbdata/tibero/*.ctl
[tibero@T1:/]$
[tibero@T1:/]$ ls -al /tibero/tbdata/tibero/.passwd
-r-------- 1 tibero dba 24 Sep 27 16:55 /tibero/tbdata/tibero/.passwd
[tibero@T1:/]$
[tibero@T1:/]$ rm /tibero/tbdata/tibero/.passwd
rm: remove write-protected regular file ‘/tibero/tbdata/tibero/.passwd’?
[tibero@T1:/]$
[tibero@T1:/]$ rm -rf /tibero/tbdata/tibero/.passwd
[tibero@T1:/]$
[tibero@T1:/]$ ls -al /tibero/tibero6/config/tibero.tip
-rw-r--r-- 1 tibero dba 255 Sep 27 16:21 /tibero/tibero6/config/tibero.tip
[tibero@T1:/]$
[tibero@T1:/]$ rm /tibero/tibero6/config/tibero.tip
[tibero@T1:/]$
[tibero@T1:/]$
[tibero@T1:/]$
[tibero@T1:/]$
[tibero@T1:/]$
[tibero@T1:/]$
```

```
[tibero@T1:/]$ cp /tibero/s/backup_off_1014/*.dtf    /tibero/tbdata/tibero
[tibero@T1:/]$ cp /tibero/s/backup_off_1014/log*.log   /tibero/tbdata/tibero
[tibero@T1:/]$ cp /tibero/s/backup_off_1014/log-*.arc  /tibero/tbdata/tibero/arch
[tibero@T1:/]$
[tibero@T1:/]$ cp /tibero/s/backup_off_1014/*.ctl    /tibero/tbdata/tibero
[tibero@T1:/]$ cp /tibero/s/backup_off_1014/.passwd    /tibero/tbdata/tibero
[tibero@T1:/]$ cp /tibero/s/backup_off_1014/*.tip    /tibero/tibero6/config
[tibero@T1:/]$
[tibero@T1:/]$
[tibero@T1:/]$
[tibero@T1:/]$
[tibero@T1:/]$ tbboot
Listener port = 8629
```

##controlfile만 백업 (online상태)

```
[tibero@T1:/]$ mkdir /tibero/s/backup_on_1124
[tibero@T1:/]$
[tibero@T1:/]$ tbsql sys/tibero

tbSQL 6

TmaxData Corporation Copyright (c) 2008-. All rights reserved.

Connected to Tibero.

SQL> ALTER DATABASE BACKUP CONTROLFILE TO TRACE AS '/tibero/s/backup_on_1124/crectl.sql' REUS

Database altered.

SQL>
SQL> !ls -al /tibero/s/backup_on_1124/crectl.sql
-rwxrwx--- 1 root vboxsf 835 Oct  6 11:26 /tibero/s/backup_on_1124/crectl.sql
SQL>
SQL> !cat /tibero/s/backup_on_1124/crectl.sql
CREATE CONTROLFILE REUSE DATABASE "tibero"
LOGFILE
 GROUP 0 '/tibero/tbdata/tibero/log01.log' SIZE 50M,
 GROUP 1 '/tibero/tbdata/tibero/log11.log' SIZE 50M,
 GROUP 2 '/tibero/tbdata/tibero/log21.log' SIZE 50M
NORESETLOGS
DATAFILE
        '/tibero/tbdata/tibero/system001.dtf',
        '/tibero/tbdata/tibero/undo001.dtf',
        '/tibero/tbdata/tibero/usr001.dtf',
        '/tibero/tbdata/tibero/syssub001.dtf'
ARCHIVELOG
MAXLOGFILES 100
MAXLOGMEMBERS 8
MAXDATAFILES 256
MAXARCHIVELOG 500
MAXBACKUPSET 500
MAXLOGHISTORY 500
CHARACTER SET MSWIN949
NATIONAL CHARACTER SET UTF16
;
```

```
---- Recovery is required in MOUNT mode.
--ALTER DATABASE RECOVER AUTOMATIC;

--ALTER DATABASE OPEN ;


---- Adding Tempfiles is required in OPEN mode.
-- ALTER TABLESPACE TEMP ADD TEMPFILE '/tibero/tbdata/tibero/temp001.dtf'
--    SIZE 100M REUSE AUTOEXTEND ON NEXT 64M MAXSIZE 3G;
```

- controlfile이 삭제되었을때 tbdown abnormal 후 tbboot시도하려면?

→ nomount 모드가 됨

```
DESC DBA DATA_FILES;
```

→ dba_data_files의 칼럼이름 출력

- 데이터파일의 경로와 이름 확인

```
select tablespace_name, file_name from dba_data_file;
```

- 온라인 백업시작

```
alter tablespace system begin backup;
```

- 해당 테이블스페이스의 데이터 파일 복사

```
!cp /tbdata/tibero/user001.dtf /tibero/s/backup_0254
```

- 온라인 백업 종료

```
alter tablespace system end backup;
```

- 0번 file이 어떤 tablespace와 연결되어있는지

```
desc v$tablespace;
select ts# from v$datafiles where file#=0;
```

0번 tablespace의 이름

```
select name from v$tablespace where ts#=0;
```

[시험안내]

1교시

실습하지 않은 내용들도 포함

tibero database link

tibero tool-여러 object 뽑아서

2교시

[migration]

export/import/loader → 추출

백업 복구 용도 - 손상된 table

a db에서 b bd로 옮기는 작업

고정된 자리에 data있는 경우

필드가 구분되는 경우

세개의 데이터가 실패되는 경우

3교시

백업

[접속]

```
[tibero@T1:/]$ vi /tibero/tibero6/client/config tbdsn.tbr
2 files to edit
[tibero@T1:/]$ vi /tibero/tibero6/client/config/tbdsn.tbr
[tibero@T1:/]$ tbsql edu/edu@edu

tbSQL 6

TmaxData Corporation Copyright (c) 2008-. All rights reserved.

Connected to Tibero using edu.

SQL> select * from a;

        C1         C2
---------- ----------
         1          1

1 row selected.

SQL>
```

[Test]

▼ Database Link

Tibero Online Manual 7.4 데이터베이스링크

- Public DB Link

→ using 절 이후의 ‘remote_2’는 연결할 데이터베이스 가리키는 이름 tbdsn.tbr에 해당 데이터베이스 정보가 저장되어 있어야한다. (link 명 public_tibero)

```
create public database link public_tibero using 'remote_2';
```

→ public db link 제거

```
drop public database link public_tibero;
```

- Private DB Link

→ 데이터베이스 링크를 생성한 사용자만 데이터베이스링크 사용

```
create database link remote_tibero using 'remote_1';
drop database link remote_tibero;
```

- 원격 데이터베이스에 연결

→ 지정한 계정(user_1)을 이용하는 데이터베이스 링크의 생성

```
create database link remote_tibero connect to user1
identified by 'password' using 'remote_1';
```

→ 현재 연결된 계정을 이용하는 데이터베이스 링크의 생성

```
create database link remote_tibero using 'remote_1';
```

- 게이트웨이

1. 멀티 스레드 서버 방식 (listener 방식)

→ Oracle

```
ora_link_remote=(
                (GATEWAY=(LISTENER=(HOST=12.34.56.78)
                                   (PORT=9999))
                         (TARGET=orcl)
                         (TX_MODE=GLOBAL))
)
```

→ ora_link_remote (=create database link에서의 using절에 사용되는 이름)

- 실행 및 종료

```
gw4orcl
tbgw
```

Tibero 와 Oracle 연결

→ Tibero Gateway DBLINK 구성

1. 게이트웨이 파일 복사

- 파일 경로 : $TB_HOME/client/bin

2. Tibero 설치된 계쩡에서 .profile에 oracle client library 사용과 관련된 내용을 밑에 추가
3. Tibero server에서 oracle server에 접근하기 위해 아래와 같이 tnsnames.ora 만든다.

▼ Tibero Tools

tbexport/tbimport/tbloader 순서

- table 생성

tbsql tibero/tmax

create table edu(c1 number, c2 varchar(10));

insert into edu values(1, ‘TIBERO’);

insert into edu values(2, ‘TMAX’);

commit;

select * from edu;

q

- 디렉토리 생성 후 tbexport

mkdir /tibero/expimp

tbexport username=sys password=tibero ip=192.168.56.241 port=8629 file=exp_data.dat log exp_data.log full=y script=y sid=tibero

- table drop후 tbimport

tbsql tibero/tmax

drop table edu;

select * from edu;

q

tbimport username=sys password=tibero ip=192.168.56.241 port=8629 file=exp_data.dat log exp_data.log full=y script=y sid=tibero table=tibero.edu

- table 생성되었는지 확인

tbsql tibero/tmax

select * from edu;

→ 테이블 생성됨

- table은 있고 안에 data만 지운후 tbimport

delete from edu;

commit;

#오류 무시

tbimport username=sys password=tibero ip=192.168.56.241 port=8629 file=exp_data.dat log exp_data.log full=y script=y sid=tibero table=tibero.edu ignore=y

- 테이블 생성되었는지 확인

tbsql tibero/tmax

select * from edu;

- tibero1에 테이블 tbimport

tbimport username=sys password=tibero ip=192.168.56.241 port=8629 file=exp_data.dat log exp_data.log full=y script=y sid=tibero table=edu ignore=y fromuser=tibero touser=tibero1

tbsql tibero1/tmax

select * from edu;

- tbloader
- loader할 디렉토리 생성

mkdir loader

cd loader

- table 생성

```
CREATE TABLE MEMBER
(
    ID        NUMBER(4) NOT NULL,
    NAME      VARCHAR2(10),
    JOB       VARCHAR2(9),
    BIRTHDATE DATE,
    CITY      VARCHAR2(10),
    PICTURE   BLOB,
    AGE       NUMBER(3),
    RESUME    CLOB
);

CREATE TABLE CLUB
(
    ID       NUMBER(6) NOT NULL,
    NAME     VARCHAR2(10),
    MASTERID NUMBER(4)
);
```

<control.ctl 파일 생성>

vi control.ctl

```
LOAD DATA
INFILE './data.dat'
APPEND
INTO TABLE club
FIELDS TERMINATED BY ','
       OPTIONALLY ENCLOSED BY '"'
LINES TERMINATED BY '|\n'
IGNORE 1 LINES
(
 id integer external,
 name,
 masterid integer external
)
```

<data.dat 파일 생성>

```
id     name      masterid|
111111,FC-SNIFER,2345|
dkkkkkkkkk|
111112,"DOCTOR CLUBE ZZANG",2222|
111113,"ARTLOVE",3333|
111114,FINANCE,1235|
111115,"DANCE MANIA",2456|
```

```
111116,"MUHANZILZU",2378|
111117,"INT'L",5555
```

- tbloader 유틸리티 실행 (로그파일 & 오류파일)

tbloader userid=tibero/tmax@tibero control=./control.ctl

▼ Tibero_Backup_Recovery

[이론]

- redo 저장시 일어나는 일들 (순서)

→ redolog file안에 redolog 존재

→ redolog는 어떤 data가 언제, 어떻게 (모든 변경내역 있는 파일)

→ data < datablock <datafile → datablock을 memory에 불러와 쓰거나 수정한다.

→ redolog file ↔ datafile은 다름

→ archive 작업은 redolog file을 achive file에 복사하는 것

- database(controlfile, redolog file, datafile) 동기화 방식

1. TSN (file 언제 발생, 언제 수정? 을 구분해줌)

   : 처음은 0 → transaction +1증가 → db내의 모든 작업들에 TSN 모두 부여

2. Checkpoint 발생 상황

   → 모든 로그 스위치 발생시

   → 인스턴스가 normal, post_tx, immediate 옵션으로 종료

   → 사용자가 수동으로 alter system checkpoint (sql)

- 백업종류

1. 논리적인 백업

→ 데이터베이스의 논리적인 단위 백업 (export tool 사용)

2. 물리적인 백업

→ 데이터베이스를 구성하는 파일을 운영체제 레벨에서 copy 명령으로 백업

(datafile, controlfile, archive logfile)

1. noarchivelog mode에서의 백업 (offline/cold)

→ 데이터베이스를 구성하는 전체 파일에 대해 운영을 멈춘 상태에서 백업

→ 데이터베이스를 백업받은 시점으로의 복구만 가능

2. archivelog mode에서의 백업 (online/hot)

→ 운영중에도 백업 가능

→ controlfile 생성문, datafile, archive logfile 백업

→ 백업된 archive logfile의 시점에 따라 datafile 백업 시점 이전도 복구 가능

[실습]

- controlfile(v$controlfile) 백업
  - 다중화

  → tbdown → controlfile 다른 위치로 copy → $TB_SID.tip에 parameter추가 → tbboot

- offline backup - copy 명령어를 통해 별도의 위치 copy

→ tibero 정상 종료한 후 OS의 copy 명령어를 통해 datafile, logfile, controlfile, tip file 백업

→ mount, open 모드에서 백업할 파일 정보 조회

```
tbdown immediate
tbboot
tbsql sys/tibero
```

tibero가 정상상태인지 확인

```
SELECT NAME FROM V$DATABASE;
SELECT USERNAME FROM DBA_USERS;
```

datafile 조회

```
SELECT NAME FROM V$DATAFILE;
SELECT FILE_NAME FROM DBA_DATA_FILES;
```

##logfile 조회

```
SELECT MEMBER FROM V$LOGFILE;
```

##현재 archivelog file 상태 확인

```
SELECT LOG_MODE FROM V$DATABASE;
```

##noarchive→archive로 변경

```
SELECT NAME FROM V$ARCHIVE_DEST_FILES;
tbdown immediate
tbboot mount  #mount 모드에서 변경해야됨

tbsql sys/tibero
SQL> ALTER DATABASE ARCHIVELOG;

Database altered.

SQL> QUIT
Disconnected.

tbdown
tbboot

tbsql sys/tibero
SELECT LOG_MODE FROM V$DATABASE;

ALTER SYSTEM SWITCH LOGFILE;
```

```
alter database backup controlfile to trace as 백업경로 reuser noresetlogs;
```

##archivelogfile 조회

```
SELECT NAME FROM V$ARCHIVE_DEST_FILES;
```

##controlfile 조회 (백업대상 control 조회)

```
SELECT NAME FROM V$CONTROLFILE;
```

tibero 정상 종료 !!!!!! 후 copy 해야됨

```
tbdown (NORMAL MODE)

ps -ef | grep tbsvr
```

##백업할 디렉토리 생성후 파일 copy

```
mkdir /tibero/s/backup_off_1014

#data->log->archive->control 순으로 copy
cp /tibero/tbdata/tibero/*.dtf  /tibero/s/backup_off_1014
cp /tibero/tbdata/tibero/log*.log   /tibero/s/backup_off_1014
cp /tibero/tbdata/tibero/arch/log-*.arc    /tibero/s/backup_off_1014
cp /tibero/tbdata/tibero/*.ctl   /tibero/s/backup_off_1014
ls -al /tibero/s/backup_off_1014

#$TB_SID.tip 파일 copy
ls -al $TB_HOME/config/$TB_SID.tip
cp /tibero/tibero6/config/tibero.tip   /tibero/s/backup_off_1014

#tibero.tip 파일 위치 출력
cat /tibero/tibero6/config/tibero.tip | grep DB_CREATE_FILE_DEST

#password 파일 copy
ls -al /tibero/tbdata/tibero/.passwd
cp /tibero/tbdata/tibero/.passwd   /tibero/s/backup_off_1014
ls -al /tibero/s/backup_off_1014
```

```
#tibero 재시동후 비정상 down
tbboot
tbdown abnormal

#원래 file들 삭제
ls -al /tibero/tbdata/tibero/*.dtf
rm /tibero/tbdata/tibero/*.dtf
rm /tibero/tbdata/tibero/log*.log
rm /tibero/tbdata/tibero/arch/log-*.arc

ls -al /tibero/tbdata/tibero/*.ctl
rm /tibero/tbdata/tibero/*.ctl

ls -al /tibero/tbdata/tibero/.passwd
rm -rf /tibero/tbdata/tibero/.passwd

rm /tibero/tibero6/config/tibero.tip

#backup_off_1014에 backup해놓은 파일들 copy
cp /tibero/s/backup_off_1014/*.dtf    /tibero/tbdata/tibero
cp /tibero/s/backup_off_1014/log*.log   /tibero/tbdata/tibero
cp /tibero/s/backup_off_1014/log-*.arc  /tibero/tbdata/tibero/arch
cp /tibero/s/backup_off_1014/*.ctl    /tibero/tbdata/tibero
cp /tibero/s/backup_off_1014/.passwd    /tibero/tbdata/tibero
cp /tibero/s/backup_off_1014/*.tip    /tibero/tibero6/config
```

$TB_SID 파일의 control_files 항목 추가

- online backup

→ controlfile 구문 생성됨

##백업할 디렉토리 생성 후 controlfile 백업

```
tbboot

#s폴더 tibero가 관여x
mkdir /tibero/s/backup_on_1124

#백업
tbsql sys/tibero
ALTER DATABASE BACKUP CONTROLFILE TO TRACE AS '/tibero/s/backup_on/crectl.sql' REUSE NORES

!ls -al /tibero/s/backup_on/crectl.sql
!cat /tibero/s/backup_on/crectl.sql

#원래 데이터 삭제
SELECT NAME FROM V$CONTROLFILE;
!rm /tibero/tbdata/tibero/c1.ctl
!rm /tibero/tbdata/tibero/c2.ctl
!ls -al /tibero/tbdata/tibero/*.ctl
q

#비정상 종료
tbdown abnormal
#인스턴스 종료 확인
ps -ef | grep tbsvr
tbboot nomount
#NOMOUNT 모드

#controlfile 없는 것 확인
ls -al /tibero/tbdata/tibero/c1.ctl
ls -al /tibero/tbdata/tibero/c2.ctl

#controlfile 재생성
tbsql sys/tibero
@/tibero/s/backup_on/crectl.sql
!ls -al /tibero/tbdata/tibero/c1.ctl
!ls -al /tibero/tbdata/tibero/c2.ctl
q

#mount모드로 재기동
tbdown
tbboot mount

#media recovery
tbsql sys/tibero
alter database recover automatic database;
q

#normal mode로 tibero 재기동
tbdown
tbboot (NORMAL MODE)
```

```
##temp datafile 추가
tbsql sys/tibero
select file_id, tablespace_name from dba_temp_files;
#0row

##내용 controlfile 맨 밑 2줄
ALTER TABLESPACE TEMP ADD TEMPFILE '/tibero/tbdata/tibero/temp001.dtf'
 SIZE 100M REUSE AUTOEXTEND ON NEXT 64M MAXSIZE 3G;

select file_id, tablespace_name from dba_temp_files;
#1row
q
```

##datafile 백업

```
mkdir /tibero/s/backup_on_0254
tbdown
tbboot (NORMAL MODE)

tbsql sys/tibero
DESC DBA_DATA_FILES;

SELECT TABLESPACE_NAME, FILE_NAME FROM DBA_DATA_FILES;

TABLESPACE_NAME
--------------------------------------------------------------------------------
FILE_NAME
--------------------------------------------------------------------------------
SYSTEM
/tibero/tbdata/tibero/system001.dtf

UNDO
/tibero/tbdata/tibero/undo001.dtf

USR
/tibero/tbdata/tibero/usr001.dtf

SYSSUB
/tibero/tbdata/tibero/syssub001.dtf

SELECT NAME FROM V$TABLESPACE;

#sysem tablespace
#tablespace backup 모드로 변경
ALTER TABLESPACE SYSTEM BEGIN BACKUP;
#백업 경로로 copy
!cp /tibero/tbdata/tibero/system001.dtf   /tibero/s/backup_on
#backup모드 종료
ALTER TABLESPACE SYSTEM END BACKUP;

#undo
ALTER TABLESPACE UNDO BEGIN BACKUP;
!cp /tibero/tbdata/tibero/undo001.dtf   /tibero/s/backup_on_0254
ALTER TABLESPACE UNDO END BACKUP;

#usr
ALTER TABLESPACE USR BEGIN BACKUP;
!cp /tibero/tbdata/tibero/usr001.dtf   /tibero/s/backup_on_0254
```

```
ALTER TABLESPACE USR END BACKUP;

#syssub
ALTER TABLESPACE SYSSUB BEGIN BACKUP;
!cp /tibero/tbdata/tibero/syssub001.dtf   /tibero/s/backup_on_0254
ALTER TABLESPACE SYSSUB END BACKUP;

#logswitch 수행
ALTER SYSTEM SWITCH LOGFILE;

#controlfile 백업
ALTER DATABASE BACKUP CONTROLFILE TO TRACE AS '/tibero/s/backup_on/crectl.sql' REUSE NORES

#logfile 백업
#logswitch 수행
ALTER SYSTEM SWITCH LOGFILE;
!cp /tibero/tbdata/tibero/arch/*.arc     /tibero/s/backup_on_0254

#tibero.tip 백업
!cp /tibero/tibero6/config/tibero.tip    /tibero/s/backup_on_0254

#password 백업
!cp /tibero/tbdata/tibero/.passwd        /tibero/s/backup_on_0254
!ls -al /tibero/s/backup_on_0254
```

```
DESC DBA_DATA_FILES;
DESC V$TABLESPACE;
DESC V$DATAFILE;

SELECT TS# FROM V$DATAFILE WHERE FILE#=0;
SELECT NAME FROM V$TABLESPACE WHERE TS#=0;
q

tbdown
tbboot
```

##일부 datafile 손상

```
SELECT NAME FROM V$DATAFILE;
!rm /tibero/tbdata/tibero/usr001.dtf
q

tbdown abnormal
tbboot (warning: mount mode)

#MOUNT MODE
tbsql sys/tibero
COL TIME FOR A10
COL ERROR FOR A30

#복구 필요한 file (기준: controlfile)
SELECT * FROM V$RECOVER_FILE;

#복구 필요한 파일 이름,경로
SELECT NAME FROM V$DATAFILE WHERE FILE# = 2;
!ls /tibero/tbdata/tibero/usr001.dtf
```

```
#백업
!cp /tibero/s/backup_on/system001.dtf      /tibero/tbdata/tibero/system001.dtf
!ls /tibero/tbdata/tibero/usr001.dtf

SELECT * FROM V$RECOVER_FILE;
#CHANGE#은 backup했을 때 checkpoint
#장애가 난 현시점이랑은 다름

#복구명령(CHANGE# 갱신)
ALTER DATABASE RECOVER AUTOMATIC DATABASE;

COL TIME FOR A10
COL ERROR FOR A30

SELECT * FROM V$RECOVER_FILE;
#0row (복구 필요x)

tbdown
tbboot
```

- redo logfile(vlog, vlogfile) 백업
  - 다중화

→ log group 추가

```
ALTER DATABASE ADD LOGFILE
GROUP 4 (
(‘‘/home/ tbdata /redo41.log ’’, ‘‘/home/ tbdata /redo42.log ’’) SIZE
```

→ log member 추가

```
ALTER DATABASE ADD LOGFILE MEMBER
‘‘/home/ tbdata /redo43.log TO GROUP 4;
```

- archive log 정보

→ v$archive_dest_files : log_archive_dest 내의 archive log files의 정보 표시 (현재)

→ v$archive_log : archive log 정보 표시(log switch할 때 archive log file 만들어질 때 입력됨)