

# [Mentoring]

## 2주차

db가 있다 없다 → datafile

instance를 기동으로 (서버 memory를 이용하여)

[기동]

tbboot mount → datafile로는 접근 못함(사용자 db접근 x)

normal모드로는 → db 접근 가능

control file → db명, characteraset, datafile 위치, logfile 위치 (db전체적인 틀)  
→ archivemode인지

tbboot mount → controlfile까지만 있다

tbboot resetlogs → (기존 redo log file 없을 때) redo log file 초기화하겠다

tbboot noresetlogs → 기존에 있는 redo log file 사용하겠다

- recycle\_bin (휴지통) : 특정 시점으로 다시 돌아갈 수 있음 / index, object는 복구 안됨 다시 생성해야함  
→ 주기적으로 비워주지 않으면 db size 커짐  
→ db\_segment : db가 실제 사용하는 크기  
→ View 는 실제 공간을 차지하지 않음 (view가 data를 가지고 있지않음)
- object(view,...) vs Segment(실제 데이터 가지고 있는 : table, index)

## 3주차

- create directory TEST\_DIR as ' '; 후에 해야됨
- tberr [오류번호]: 더 자세한 예러

1) session\_monitor.sh (sid, serial#(number), user, ip,sql\_id,sql\_child\_number,status) → 어떤 session할지는 직접 query문 작성  
v\$session

2) database\_info.sh (dbname,sid,archivemod,tip\_file)  
v\$database, v\$instance

3) tablespace\_info.sh (tbs\_name, dtf\_count, size(M), autoextend)  
dba\_tablespaces, dba\_data\_files

4) valid\_object.sh (owner, object\_type,status, count(\*)) → status 가 valid인 object만  
dba\_objects

5) top5\_segments (owner,segment\_type,segment\_name,bytes(M)) → segment size 제일 큰 5개만 보이도록 (owner not in 'SYS','SYSCAT','OUTLN')  
dba\_segments

[test.sh]

tbsql sys/tibero <<EOF

select \* from v\ \$database;

EOF

```
SQL> desc v$session
```

COLUMN_NAME	TYPE	CONSTRAINT
SID	NUMBER	
SERIAL#	NUMBER	
AUDSID	NUMBER	
USER#	NUMBER	
USERNAME	VARCHAR(128)	
IPADDR	VARCHAR(46)	
COMMAND	NUMBER	
STATUS	VARCHAR(32)	
SCHEMA#	NUMBER	
SCHEMANAME	VARCHAR(128)	
TYPE	VARCHAR(15)	
SQL_ID	VARCHAR(13)	
SQL_CHILD_NUMBER	NUMBER	
PREV_SQL_ID	VARCHAR(13)	
PREV_CHILD_NUMBER	NUMBER	
SQL_ET	NUMBER	
LOGON_TIME	DATE	
STATE	VARCHAR(25)	
WLOCK_WAIT	VARCHAR(18)	
WAIT_EVENT	NUMBER	
WAIT_TIME	NUMBER	
PGA_USED_MEM	NUMBER	
SQL_TRACE	VARCHAR(8)	
PROG_NAME	VARCHAR(30)	
CLIENT_PID	NUMBER	
PID	NUMBER	
WTHR_ID	NUMBER	
OS_THR_ID	NUMBER	
OSUSER	VARCHAR(128)	

```
SQL> desc v$database
```

COLUMN_NAME	TYPE	CONSTRAINT
DBID	NUMBER	
NAME	VARCHAR(40)	
CREATE_DATE	DATE	
CURRENT_TSN	NUMBER	
OPEN_MODE	VARCHAR(10)	
RESETLOG_TSN	NUMBER	
RESETLOG_DATE	DATE	
PREV_RESETLOG_TSN	NUMBER	
PREV_RESETLOG_DATE	DATE	
LOG_MODE	VARCHAR(12)	
CKPT_TSN	NUMBER	
CKPT_DATE	DATE	
CPU_NAME	VARCHAR(32)	
PLATFORM_NAME	VARCHAR(32)	
CPU_MODEL	VARCHAR(256)	
OS_UPTIME	VARCHAR(256)	

```
SQL> desc v$instance
```

COLUMN_NAME	TYPE	CONSTRAINT
INSTANCE_NUMBER	NUMBER	
INSTANCE_NAME	VARCHAR(40)	
DB_NAME	VARCHAR(40)	
HOST_NAME	VARCHAR(63)	
PARALLEL	VARCHAR(3)	
THREAD#	NUMBER	
VERSION	VARCHAR(8)	
STARTUP_TIME	DATE	
STATUS	VARCHAR(16)	
SHUTDOWN_PENDING	VARCHAR(4)	
TIP_FILE	VARCHAR(256)	

```
SQL> desc dba_tablespaces
```

COLUMN_NAME	TYPE	CONSTRAINT
TABLESPACE_NAME	VARCHAR(128)	
TS_ID	NUMBER	
DATAFILE_COUNT	NUMBER	
BLOCK_SIZE	NUMBER	
NEXT_EXTENT	NUMBER	
STATUS	VARCHAR(9)	
CONTENTS	VARCHAR(9)	
LOGGING	VARCHAR(9)	
FORCE_LOGGING	VARCHAR(3)	
ALLOCATION_TYPE	VARCHAR(7)	
ENCRYPTED	VARCHAR(3)	
INITIAL_EXTENT	NUMBER	
MIN_EXTENTS	NUMBER	
MAX_EXTENTS	NUMBER	
EXTENT_MANAGEMENT	VARCHAR(5)	
SEGMENT_SPACE_MANAGEMENT	VARCHAR(4)	

```
SQL> desc dba_data_files
```

COLUMN_NAME	TYPE	CONSTRAINT
FILE_NAME	VARCHAR(256)	
FILE_ID	NUMBER	
TABLESPACE_NAME	VARCHAR(128)	
BYTES	NUMBER	
BLOCKS	NUMBER	
STATUS	CHAR(9)	
RELATIVE_FNO	NUMBER	
AUTOEXTENSIBLE	VARCHAR(3)	
MAXBYTES	NUMBER	
MAXBLOCKS	NUMBER	
INCREMENT_BY	NUMBER	

```
SQL> desc dba_objects

COLUMN_NAME                                TYPE                                CONSTRAINT
-----
OWNER                                       VARCHAR(128)
OBJECT_NAME                               VARCHAR(128)
SUBOBJECT_NAME                           VARCHAR(128)
OBJECT_ID                                  NUMBER
OBJECT_TYPE                               VARCHAR(23)
OBJECT_TYPE_NO                            NUMBER
CREATED                                   DATE
LAST_DDL_TIME                             DATE
TIMESTAMP                                 VARCHAR(19)
STATUS                                    VARCHAR(7)
TEMPORARY                                 VARCHAR(1)
```

```
SQL> desc dba_segments

COLUMN_NAME                                TYPE                                CONSTRAINT
-----
OWNER                                       VARCHAR(128)
SEGMENT_NAME                               VARCHAR(128)
PARTITION_NAME                           VARCHAR(128)
SEGMENT_TYPE                               VARCHAR(18)
TABLESPACE_NAME                           VARCHAR(128)
HEADER_FILE                                NUMBER
HEADER_BLOCK                               NUMBER
BLOCKS                                     NUMBER
EXTENTS                                    NUMBER
RELATIVE_FNO                               NUMBER
BUFFER_POOL                               VARCHAR(7)
MAX_EXTENTS                                NUMBER
BYTES                                      NUMBER
INITIAL_EXTENT                             NUMBER
NEXT_EXTENT                               NUMBER(0)
```

Answer

```
[1_session_monitor.sh]
tbsql -s sys/tibero <<EOF
set linesize 120
set feed off
col sid for a20
col username for a10
col ipaddr for a15
col sql_id for a20
col status for a10
select sid||'|'||serial# sid, username, ipaddr, sql_id||'|'||sql_child_number sql_id, status
from v$sqlsession;
EOF

[2_database_info.sh]
tbsql -s sys/tibero <<EOF
set linesi 120
set feed off
col dbname for a15
col sid for a10
col archivemod for a20
```

```
col tip for a50
select
a.name dbname , b.instance_name sid, a.log_mode, b.tip_file tip
from v\${database} a, v\${instance} b;
EOF
```

```
[4_valid_object.sh]
tbsql -s sys/tibero <<EOF
set linesi 120
set pagesi 200
set feed off
col owner for a10
col object_type for a25
col status for a10
col count for 9999
select owner, object_type, status, count (*) count
from dba_objects
where status='VALID'
group by owner, object_type, status
order by 1,2,4;
--select문에서 앞에서부터 1,2,3,4 순서
EOF
```

```
[5_top5_segments.sh]
tbsql -s sys/tibero<<EOF
set linesi 120
col owner for a10
col segment_type for a20
col segment_name for a30
set feed off
select *
from (
select owner, segment_type, segment_name,round(sum(bytes)/1024/1024,2) "size M"
from dba_segments
where owner not in ('SYS','SYSCAT','OUTLN')
group by owner, segment_type, segment_name
order by 4 desc)
where rownum < 6;
EOF
```

## 4주차

- UNION → 한번 SORTING(TEMP TABLESPACE 사용) 하고 난 다음 중복을 제거하기 때문에 SORTING된채로 출력

### [ oracle ]

- ps -ef | grep tbsvr → memory에 떠있는 server가 db server : .dtf, .redo, .ctl,
- memory에 접속하는 모든 것들 client
- client ex) tbsql, tibero studio
- Tibero to Oracle : DB link (source : tibero, target : oracle)
- sqlplus라는 client를 이용하여 oracle에 접속할 수 있어야 db link 생성 가능

- 기본적인 객체들 → 소문자로 만들어도 대문자로 생성됨
- 대소문자 구분되는 이유 : 아스키 문자 다름

- unique index 이려면 무조건 값이 1개 : 중복된 값이 없어야됨
- primary key : +null이 있으면 안됨.
- instance vs database : instance는 memory
- 멀티 instance : id에서 tiberoh하나더 설치하는 느낌

#### TIBERO TO TIBERO DB LINK

dbname=jeongmin

port=9629

/tiberoh/jeongmin

export TB\_SID=jeongmin 하면 변경됨

- 뷰

→ v\$ : memory에서 읽어오는 (동적 뷰) : 현재에 따라 달라지는 뷰

→ dba\_user\_all : 언제 조회하는 같은 결과 보여줌 (정적 뷰)

CREATE DATABASE LINK [DBLINK명]

CONNECT TO [TARGET DB의 USER명] IDENTIFIED BY [TARGET DB의 USER PASSWORD] USING 'ALLIAS'명';

CREATE DATABASE LINK MORI CONNECT TO SYS IDENTIFIED BY TIBERO USING 'tiberoh';

→ db link를 만든(가지고 있는) 사용자안에서 쓸 수 있음.

CREATE DATABASE LINK TLINK CONNECT TO TEST IDENTIFIED BY TEST USING 'jeongmin';

## 5주차

- 실체화 뷰

```
SQL> DESC DBMS_MVIEW
```

PROCEDURE 'REFRESH'		
ARGUMENT_NAME	TYPE	IN/OUT
QUALIFIED_OBJ	VARCHAR	IN
REFRESH_METHOD	VARCHAR	IN

  

PROCEDURE 'EXPLAIN_REWRITE'		
ARGUMENT_NAME	TYPE	IN/OUT
QUERY	VARCHAR	IN
STATEMENT_ID	VARCHAR	IN

- 실체화 뷰 : 10분 단위로 데이터의 변화를 보기 위해서 ↔ 일반 뷰와의 차이점 : 실제 데이터의 부활을 주어야함
- 실체화 뷰 : 실제 데이터를 가지고 있음.
- SQL 언어 : DDL, DML, DCL
- TRUNCATE : DDL

- INSERT 후 COMMIT 꼭 하기 → DML을 하고 난 후에는 COMMIT이나 ROLLBACK하여야함
- COMMIT, ROLLBACK을 찍을지 말지를 결정하는 것이 중요 → INSERT, DELETE보다는

## 6주차

- 테이블을 압축하는 이유 ? - 과거 테이블은 조회할 일이 얼마 없기 때문에 STORAGE TABLE 비용을 줄이기 위해 COMPRESS해준다.
- LOCAL INDEX vs GLOBAL INDEX
  - 파티션 테이블에서 쓰임
  - 각 파티션에서만 동작하는것이 LOCAL INDEX
  - 전체에서 동작하는것이 GLOBAL INDEX
- PARTITION 작업이 월별이 많은데 GLOBAL INDEX를 쓰면 REBUILD를 전체로 해야되기 때문에 시간이 오래 걸림 → GLOBAL INDEX 지양함
- 실제로는 SUBPARTITION까지 쓰지 않음
- HASH 파티션 HASH(파티션 키 칼럼) → 파티션 키 칼럼이 복합이다 = 파티션이 하나가 아니다.
- LOCK TYPE의 칼럼은 자동적으로 INDEX가 생성이 됨.