

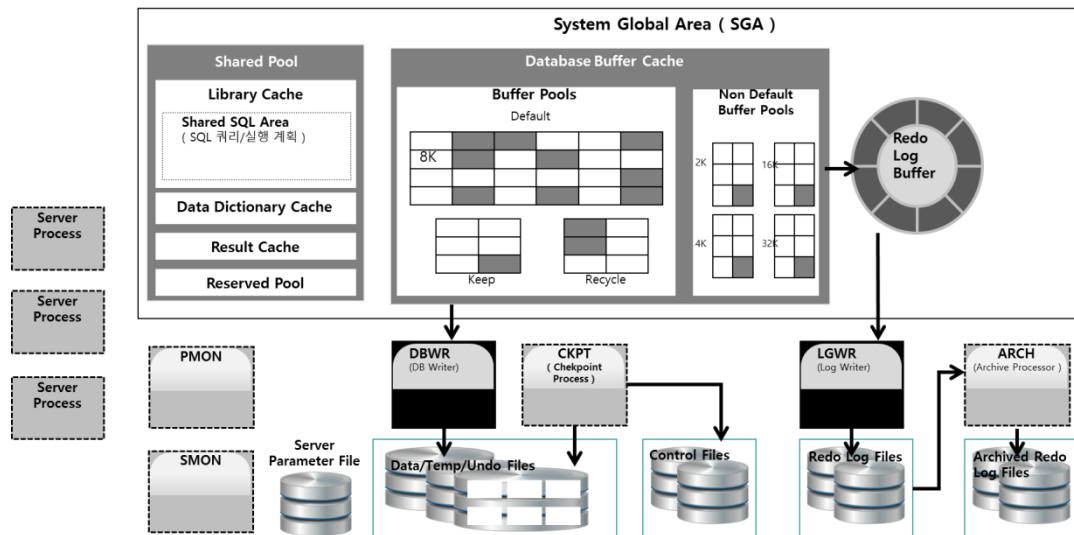
데이터베이스 실습 2

과제 내용

- 오라클의 구조 관련 실습
- 데이터베이스 구동과 순서

데이터베이스 아키텍처 확인하기

데이터베이스의 아키텍처는 메모리와 파일을 함께 사용하는 복잡한 구조로 되어 있습니다.



본 실습에서는 위 그림과 같은 아키텍처의 일부 구성요소들을 확인해 볼 것입니다.

SGA 영역 확인하기

과제 1 의 VM 에 접속하여 전체 SGA 의 크기를 알아보겠습니다. 현재 실습용 데이터베이스는 자동으로 SGA 영역을 구성하도록 되어 있습니다.

```
root [~] $ sqlplus sys/oracle_4U as sysdba
```

SQL*Plus: Release 11.2.0.2.0 Production on Mon Jun 10 11:53:05 2019

Copyright (c) 1982, 2011, Oracle. All rights reserved.

Connected to:

Oracle Database 11g Express Edition Release 11.2.0.2.0 - 64bit Production

SQL> **show sga**

Total System Global Area 1023004672 bytes

Fixed Size 2232896 bytes

Variable Size 620760512 bytes

Database Buffers 394264576 bytes

Redo Buffers 5746688 bytes

바이트로 되어 있는 단위를 MB 로 바꾸어서 크기를 가늠해 보도록 합니다. 1MB 의 크기는 아래와 같습니다. 여기서는 DB Buffer cache 와 Log Buffer 크기를 알 수 있습니다.

메가바이트 (MB) → 바이트 (B)

1 MB = 1,048,576 B

8388608 비트(bit)	1048576 바이트(B)	1024 킬로바이트(KB)
1 메가바이트(MB)	0.000977 기가바이트(GB)	9.5367e-7 테라바이트(TB)
9.3132e-10 페타바이트(PB)	9.0949e-13 엑사바이트(EB)	

SQL> **select name, value/1048576 as "size(MB)" from v\$sga;**

NAME	size(MB)
Fixed Size	2.12945557
Variable Size	600.003357
Database Buffers	368
Redo Buffers	5.48046875

여기에서 대략적으로 전체 DB Buffer 는 약 368MB 이고, Redo Buffer 는 약 5MB 임을 확인할 수 있습니다.

Shared Pool 사이즈는 아래와 같이 확인이 가능합니다.

SQL> **select COMPONENT, CURRENT_SIZE/1048576 as "size(MB)" from v\$sga_dynamic_components;**

COMPONENT	size(MB)
shared pool	196
large pool	4
java pool	4
streams pool	4
DEFAULT buffer cache	272
KEEP buffer cache	0
RECYCLE buffer cache	0
DEFAULT 2K buffer cache	0
DEFAULT 4K buffer cache	0
DEFAULT 8K buffer cache	0
DEFAULT 16K buffer cache	0
COMPONENT	size(MB)
DEFAULT 32K buffer cache	0
Shared IO Pool	96
ASM Buffer Cache	0
14 rows selected.	

이를 통해 Shared Pool 의 크기는 약 196MB 임을 확인할 수 있습니다.

파라미터 파일 확인하기

이번에는 데이터베이스 관련 파일들을 확인해보겠습니다. 먼저 파라미터 파일을 알아보겠습니다. 파라미터 파일은 파라미터 파일을 생성하도록 한 후 확인하면 됩니다.

먼저 아래와 같이 파라미터 파일을 생성한 후 데이터베이스에서 나가보겠습니다.

```
SQL> create pfile from spfile;

File created.

SQL> exit
Disconnected from Oracle Database 11g Express Edition Release 11.2.0.2.0 - 64bit
Production
```

이러면 PFILE 의 위치는 아래와 같은 경로에 위치하게 됩니다.

```
root [~] $ cd /u01/app/oracle/product/11.2.0/xe/dbs/
root [dbs] $ ls
hc_XE.dat init.ora initXE.ora lkXE orapwXE spfileXE.ora
```

해당 파일을 확인해 보겠습니다. (cat 명령어를 이용하면 파일의 내용을 읽을 수 있습니다.)

```

root [dbs] $ cat initXE.ora
XE._db_cache_size=293601280
XE._java_pool_size=4194304
XE._large_pool_size=4194304
XE._oracle_base='/u01/app/oracle'#ORACLE_BASE set from environment
XE._pga_aggregate_target=411041792
XE._sga_target=616562688
XE._shared_io_pool_size=100663296
XE._shared_pool_size=197132288
XE._streams_pool_size=4194304
*.aq_tm_processes=2
*.audit_file_dest='/u01/app/oracle/admin/XE/adump'
*.compatible='11.2.0.0.0'
*.control_files='/u01/app/oracle/oradata/XE/control.dbf'
*.db_name='XE'
*.db_recovery_file_dest='/opt/fra'
*.db_recovery_file_dest_size=10737418240
*.diagnostic_dest='/u01/app/oracle'
*.dispatchers=(PROTOCOL=TCP) (SERVICE=XEXDB)'
*.job_queue_processes=4
*.memory_target=1027604480
*.open_cursors=300
*.remote_login_passwordfile='EXCLUSIVE'
*.sessions=250
*.shared_servers=4
*.undo_management='AUTO'
*.undo_tablespace='UNDOTBS1'

```

위에서 몇 가지 내용을 확인할 수 있습니다. CONTROL FILE 의 위치, 데이터베이스의 이름, SGA 전체의 크기 등등이 명시되어 있습니다.

컨트롤 파일 확인하기

위에서 확인한 컨트롤 파일 경로로 가서 컨트롤 파일을 확인해보겠습니다. 컨트롤 파일은 `*.control_files='/u01/app/oracle/oradata/XE/control.dbf'` 에서 확인이 가능합니다.

```

root [dbs] $ cd /u01/app/oracle/oradata/XE/
root [XE] $ ls
control.dbf log1b.LOG log3b.LOG sysaux.dbf temp.dbf  univ.dbf
example.dbf log2b.LOG music.dbf system.dbf undotbs1.dbf users.dbf

```

컨트롤 파일은 이진(바이너리) 파일로, vi 혹은 cat 같은 명령어로 열어보려고 하면 식별이 불가능한 값들로 구성되어 있습니다. 파일을 읽어보려고 하여 이를 확인해보겠습니다.

```

root [XE] $ cat control.dbf
4@R}{z
XE R@Q$/z8$PuTTYu

```

```
I L& ;91z 8 c 8Q$ PuTTY PuTTY
(생략)
CTRL + C 를 눌러 강제로 빠져 나옵니다.
root [XE] $
```

데이터 파일 및 리두 로그 파일 위치 확인하기

데이터 파일 역시 현재 경로에 함께 있는 것이 확인이 가능합니다.

```
root [XE] $ ls
company.dbf control.dbf example.dbf log1b.LOG log2b.LOG log3b.LOG music.dbf
sysaux.dbf system.dbf temp.dbf undotbs1.dbf univ.dbf users.dbf
```

이들 모두 역시 이진(바이너리) 파일로 읽기가 불가능 하므로 데이터베이스에서 이들의 위치를 확인해보도록 하겠습니다. 데이터파일은 v\$datafile 에서 조회가 가능합니다.

```
root [XE] $ sqlplus sys/oracle_4U as sysdba

SQL*Plus: Release 11.2.0.2.0 Production on Thu Nov 16 13:28:20 2017

Copyright (c) 1982, 2011, Oracle. All rights reserved.

Connected to:
Oracle Database 11g Express Edition Release 11.2.0.2.0 - 64bit Production

SQL> desc v$datafile;
Name                               Null?  Type
-----
FILE#                               NUMBER
CREATION_CHANGE#                   NUMBER
CREATION_TIME                       DATE
TS#                                 NUMBER
RFILE#                             NUMBER
STATUS                             VARCHAR2(7)
ENABLED                             VARCHAR2(10)
CHECKPOINT_CHANGE#                 NUMBER
CHECKPOINT_TIME                     DATE
UNRECOVERABLE_CHANGE#              NUMBER
UNRECOVERABLE_TIME                 DATE
LAST_CHANGE#                       NUMBER
LAST_TIME                           DATE
OFFLINE_CHANGE#                    NUMBER
```

ONLINE_CHANGE#	NUMBER
ONLINE_TIME	DATE
BYTES	NUMBER
BLOCKS	NUMBER
CREATE_BYTES	NUMBER
BLOCK_SIZE	NUMBER
NAME	VARCHAR2(513)
PLUGGED_IN	NUMBER
BLOCK1_OFFSET	NUMBER
AUX_NAME	VARCHAR2(513)
FIRST_NONLOGGED_SCN	NUMBER
FIRST_NONLOGGED_TIME	DATE
FOREIGN_DBID	NUMBER
FOREIGN_CREATION_CHANGE#	NUMBER
FOREIGN_CREATION_TIME	DATE
PLUGGED_READONLY	VARCHAR2(3)
PLUGIN_CHANGE#	NUMBER
PLUGIN_RESETLOGS_CHANGE#	NUMBER
PLUGIN_RESETLOGS_TIME	DATE

```
SQL> set linesize 120
SQL> col name for a40
SQL> select file#, name, status from v$datafile;
```

FILE#	NAME	STATUS
1	/u01/app/oracle/oradata/XE/system.dbf	SYSTEM
2	/u01/app/oracle/oradata/XE/sysaux.dbf	ONLINE
3	/u01/app/oracle/oradata/XE/undotbs1.dbf	ONLINE
4	/u01/app/oracle/oradata/XE/users.dbf	ONLINE
5	/u01/app/oracle/oradata/XE/example.dbf	ONLINE
6	/u01/app/oracle/oradata/XE/univ.dbf	ONLINE
7	/u01/app/oracle/oradata/XE/music.dbf	ONLINE
8	/u01/app/oracle/oradata/XE/company.dbf	ONLINE

8 rows selected.

이번에는 리두로그 파일의 위치를 확인해보겠습니다.

```
SQL> desc v$logfile;
```

Name	Null?	Type
GROUP#		NUMBER
STATUS		VARCHAR2(7)
TYPE		VARCHAR2(7)
MEMBER		VARCHAR2(513)
IS_RECOVERY_DEST_FILE		VARCHAR2(3)

```
SQL> col member for a50
SQL> select GROUP#, TYPE, MEMBER, STATUS from v$logfile;
```

GROUP#	TYPE	MEMBER	STATUS
1	ONLINE	/opt/fra/XE/onlinelog/o1_mf_1_dsrgcm7o_.log	
1	ONLINE	/u01/app/oracle/oradata/XE/log1b.LOG	
2	ONLINE	/u01/app/oracle/oradata/XE/log2b.LOG	
2	ONLINE	/opt/fra/XE/onlinelog/o1_mf_2_dsrgclx8_.log	
3	ONLINE	/opt/fra/XE/onlinelog/o1_mf_3_dsrgsv7s_.log	
3	ONLINE	/u01/app/oracle/oradata/XE/log3b.LOG	

6 rows selected.

위와 같이 리두로그 파일은 3 * 2 개의 복사로 이루어져 운영되고 있습니다.

데이터베이스 백그라운드 프로세스

백그라운드 프로세스 확인

데이터베이스 백그라운드 프로세스에는 굉장히 많은 종류가 있습니다. 여기에서는 기본적으로 구동되는 백그라운드 프로세스를 확인하는 작업을 수행해 보겠습니다.

먼저 백그라운드 프로세스는 아래 v\$bgprocess 에서 확인이 가능합니다. 그리고 현재 구동되는 백그라운드 프로세스만 확인해보도록 하겠습니다.

```
SQL> desc v$bgprocess
```

Name	Null?	Type
PADDR		RAW(8)
PSERIAL#		NUMBER
NAME		VARCHAR2(5)
DESCRIPTION		VARCHAR2(64)
ERROR		

```
SQL> set linesize 120
```

```
SQL> col name for a10
```

```
SQL> col description for a60
```

```
SQL> select paddr, name, description from v$bgprocess where paddr != '00';
```

PADDR	NAME	DESCRIPTION
000000009C8793E0	PMON	process cleanup
000000009C87B4E0	VKTM	Virtual Keeper of TiMe process

000000009C87C560	GEN0	generic0
000000009C87D5E0	DIAG	diagnosibility process
000000009C87E660	DBRM	DataBase Resource Manager
000000009C88EE60	VKRM	Virtual sKeduler for Resource Manager
000000009C87A460	PSP0	process spawner 0
000000009C87F6E0	DIA0	diagnosibility process 0
000000009C880760	MMAN	Memory Manager
000000009C8817E0	DBW0	db writer process 0
000000009C882860	LGWR	Redo etc.

PADDR	NAME	DESCRIPTION
000000009C8838E0	CKPT	checkpoint
000000009C884960	SMON	System Monitor Process
000000009C895160	SMCO	Space Manager Process
000000009C8859E0	RECO	distributed recovery
000000009C8940E0	CJQ0	Job Queue Coordinator
000000009C88FEE0	QMNC	AQ Coordinator
000000009C886A60	MMON	Manageability Monitor Process
000000009C887AE0	MMNL	Manageability Monitor Process 2

19 rows selected.

수업 때 다룬 주요 프로세스 외에도 다양한 프로세스들이 구동되어 데이터베이스를 운영하고 있다는 것을 확인할 수 있습니다.

데이터베이스를 나가서 OS 레벨에서 프로세스를 조회해 보겠습니다.

```
SQL> exit
Disconnected from Oracle Database 11g Express Edition Release 11.2.0.2.0 - 64bit
Production
[root@db11gxe XE]# ps -ef | grep XE
oracle 4392 1 0 Nov12 ? 00:00:00 xe_pmon_XE
oracle 4394 1 0 Nov12 ? 00:00:00 xe_psp0_XE
oracle 4396 1 0 Nov12 ? 00:00:00 xe_vktm_XE
oracle 4400 1 0 Nov12 ? 00:00:00 xe_gen0_XE
oracle 4402 1 0 Nov12 ? 00:00:00 xe_diag_XE
oracle 4404 1 0 Nov12 ? 00:00:00 xe_dbrm_XE
oracle 4406 1 0 Nov12 ? 00:00:06 xe_dia0_XE
oracle 4408 1 0 Nov12 ? 00:00:00 xe_mman_XE
oracle 4410 1 0 Nov12 ? 00:00:00 xe_dbw0_XE
oracle 4412 1 0 Nov12 ? 00:00:00 xe_lgwr_XE
oracle 4414 1 0 Nov12 ? 00:00:00 xe_ckpt_XE
oracle 4416 1 0 Nov12 ? 00:00:03 xe_smon_XE
oracle 4418 1 0 Nov12 ? 00:00:00 xe_reco_XE
oracle 4420 1 0 Nov12 ? 00:00:20 xe_mmon_XE
oracle 4422 1 0 Nov12 ? 00:00:01 xe_mmnl_XE
oracle 4424 1 0 Nov12 ? 00:00:00 xe_d000_XE
oracle 4426 1 0 Nov12 ? 00:00:00 xe_s000_XE
```



```

oracle 4428 1 0 Nov12 ? 00:00:00 xe_s001_XE
oracle 4430 1 0 Nov12 ? 00:00:00 xe_s002_XE
oracle 4432 1 0 Nov12 ? 00:00:00 xe_s003_XE
oracle 4508 1 0 Nov12 ? 00:00:01 xe_vkrm_XE
oracle 4510 1 0 Nov12 ? 00:00:00 xe_qmnc_XE
oracle 4526 1 0 Nov12 ? 00:00:03 xe_cjq0_XE
oracle 4529 1 0 Nov12 ? 00:00:00 xe_q000_XE
oracle 4531 1 0 Nov12 ? 00:00:00 xe_q001_XE
oracle 4548 1 0 Nov12 ? 00:00:00 xe_smco_XE
oracle 5852 1 0 Nov13 ? 00:00:00 xe_q002_XE
oracle 6093 6081 0 Nov13 ? 00:00:00 oracleXE
(DESCRIPTION=(LOCAL=YES)(ADDRESS=(PROTOCOL=beq)))
oracle 28697 1 0 14:50 ? 00:00:00 xe_w000_XE
root 28707 16383 0 14:53 pts/1 00:00:00 grep XE

```

위와 같이 XE 데이터베이스에 관련되어 운영중인 프로세스들 확인이 가능합니다.

xe_XXXX_XE 로 되어 있는 것들이 이들에 해당합니다.

PMON 은 프로세스 중 가장 중요한 프로세스로, PMON 이 죽으면 데이터베이스가 내려간다고 설명 했습니다. PMON 을 강제로 Kill 하여 데이터베이스가 꺼지는지 확인해 보겠습니다. 프로세스에서 PMON 의 PID 를 찾습니다.

<주의: 실습 환경마다 PID 는 다를 수 있으니 체크하길 바랍니다. - 문제 2 번과 연결>

```

root [XE] $ ps -ef | grep pmon
oracle 4392 1 0 Nov12 ? 00:00:00 xe_pmon_XE
root 28951 16383 0 16:02 pts/1 00:00:00 grep pmon
[root@db11gxe XE]# kill -9 <위의 확인값 입력, 여기서는 4392>
[root@db11gxe XE]#

```

<본 화면을 캡처하여 문제 2 에 첨부하십시오.>

```

[root@db11gxe XE]# ps -ef | grep pmon
oracle 4392 1 0 Nov12 ? 00:00:00 xe_pmon_XE
root 28951 16383 0 16:02 pts/1 00:00:00 grep pmon
[root@db11gxe XE]# kill -9 4392

```

```

root [XE] $ ps -ef | grep XE
root 28994 16383 0 16:25 pts/1 00:00:00 grep XE
root [XE] $ sqlplus sys/oracle_4U as sysdba

```

SQL*Plus: Release 11.2.0.2.0 Production on Thu Nov 16 16:25:29 2017

Copyright (c) 1982, 2011, Oracle. All rights reserved.

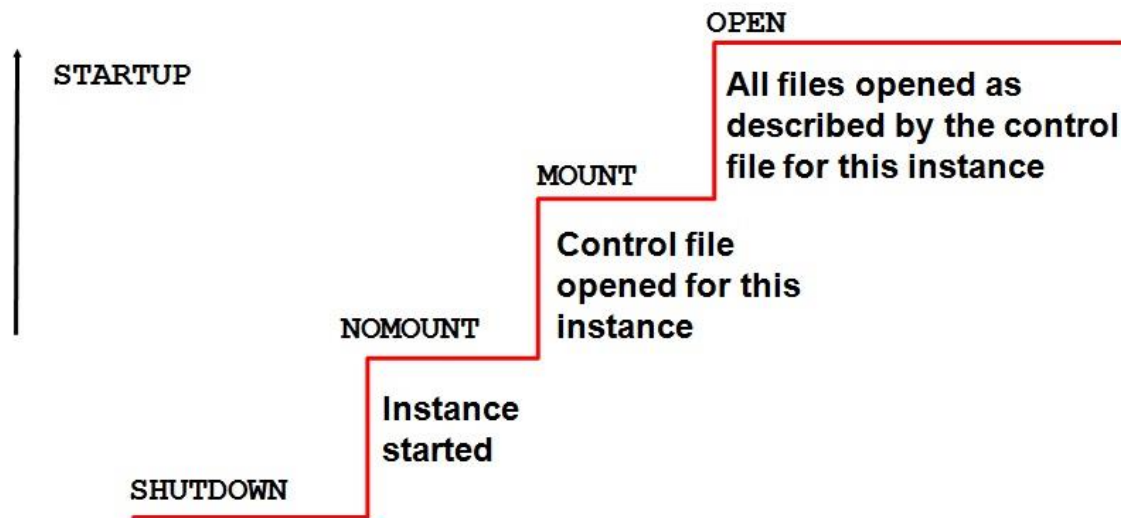
Connected to an idle instance.

SQL>

이제는 백그라운드 프로세스를 확인해도 모든 프로세스가 꺼진 상태여서 아무것도 출력되지 않음을 알 수 있으며, 아래 노란색 idle instance 는 데이터베이스가 꺼진 상태를 의미합니다.

데이터베이스 구동하기

데이터베이스가 현재는 꺼진 상태이므로 이제 구동을 해보겠습니다. 데이터베이스 구동은 3 단계를 거쳐 OPEN 이 됩니다.



데이터베이스 NOMOUNT

아래와 같이 데이터베이스를 NOMOUNT 까지만 구동합니다. NOMOUNT 로 갈 때에는 파라미터 파일을 읽어들이어 SGA 의 공간을 메모리에서 확보합니다.

```
SQL> startup nomount;  
ORACLE instance started.
```

```
Total System Global Area 1023004672 bytes  
Fixed Size                2232896 bytes  
Variable Size             637537728 bytes
```

Database Buffers	377487360 bytes
Redo Buffers	5746688 bytes

한 번 백그라운드 프로세스의 상태는 어떤지 확인해보겠습니다. !를 쓰면, SQL 창에서도 OS 명령어 수행이 가능합니다.

```
SQL> !ps -ef | grep xe_
oracle 29096 1 0 16:35 ? 00:00:00 xe_pmon_XE
oracle 29098 1 0 16:35 ? 00:00:00 xe_psp0_XE
oracle 29100 1 0 16:35 ? 00:00:00 xe_vktm_XE
oracle 29104 1 0 16:35 ? 00:00:00 xe_gen0_XE
oracle 29106 1 0 16:35 ? 00:00:00 xe_diag_XE
oracle 29108 1 0 16:35 ? 00:00:00 xe_dbrm_XE
oracle 29110 1 0 16:35 ? 00:00:00 xe_dia0_XE
oracle 29112 1 0 16:35 ? 00:00:00 xe_mman_XE
oracle 29114 1 0 16:35 ? 00:00:00 xe_dbw0_XE
oracle 29116 1 0 16:35 ? 00:00:00 xe_lgwr_XE
oracle 29118 1 0 16:35 ? 00:00:00 xe_ckpt_XE
oracle 29120 1 0 16:35 ? 00:00:00 xe_smon_XE
oracle 29122 1 0 16:35 ? 00:00:00 xe_reco_XE
oracle 29124 1 0 16:35 ? 00:00:00 xe_mmon_XE
oracle 29126 1 0 16:35 ? 00:00:00 xe_mmln_XE
oracle 29128 1 0 16:35 ? 00:00:00 xe_d000_XE
oracle 29130 1 0 16:35 ? 00:00:00 xe_s000_XE
oracle 29132 1 0 16:35 ? 00:00:00 xe_s001_XE
oracle 29134 1 0 16:35 ? 00:00:00 xe_s002_XE
oracle 29136 1 0 16:35 ? 00:00:00 xe_s003_XE
root 29229 28995 0 16:47 pts/1 00:00:00 /bin/bash -c ps -ef | grep xe_
```

위와 같이 백그라운드 프로세스가 구동되어 있는 것을 확인하실 수 있습니다.

데이터베이스의 정보를 보기 위해 쿼리를 수행하면, 데이터베이스가 마운트 상태가 아니어서 할 수 없다는 내용을 볼 수 있습니다. 즉 데이터베이스 시스템에 관한 조회는 마운트 단계에서 부터 가능합니다.

```
SQL> select NAME, OPEN_MODE from v$database;
select NAME, OPEN_MODE from v$database
*
ERROR at line 1:
ORA-01507: database not mounted
```

데이터베이스 MOUNT

이제 데이터베이스 MOUNT 를 수행합니다. MOUNT 단계에서는 Control File 을 읽으며, 만에하나 파라미터 파일에 지정된 위치에 Control File 이 없으면 데이터베이스는 구동되지 않습니다.

```
SQL> alter database mount;
```

```
Database altered.
```

구동한 후 확인하려던 쿼리를 수행합니다.

```
SQL> select NAME, OPEN_MODE from v$database;
```

NAME	OPEN_MODE
XE	MOUNTED

이제는 데이터베이스가 MOUNT 상태임이 확인이 가능합니다.

데이터베이스 OPEN

이제 데이터베이스를 OPEN 하겠습니다. 그리고 다시 한 번 쿼리를 수행합니다.

```
SQL> alter database open;
```

```
Database altered.
```

```
SQL> select NAME, OPEN_MODE from v$database;
```

NAME	OPEN_MODE
XE	READ WRITE

이제 MOUNT 에서 데이터베이스가 읽기/쓰기가 가능한 OPEN 모드로 바뀐 것을 확인할 수 있습니다.

데이터베이스를 나간 후 구동까지의 실습을 마치겠습니다.

```
SQL> exit
```

```
Disconnected from Oracle Database 11g Express Edition Release 11.2.0.2.0 - 64bit  
Production
```

연습 문제

실습을 수행한 내용을 바탕으로 아래 문제에 답하시오.

1. 실습을 수행한 환경에서 파라미터 파일의 SGA 크기는 몇으로 되어 있는가?
2. pmon 프로세스를 외에 다른 프로세스를 kill 하면 어떤 현상이 발생하는가?
3. 데이터베이스가 NOMOUNT 단계로 가면서 발생한 두 가지 현상은 무엇인가?