**===== 190129 (화) Day 12 4교시 ==============================================**

**oracle admin 한글교재** – passwdls

: **@oraclewikis!**

**----------------------------------------------------------------------**

**오라클 Admin**

**----------------------------------------------------------------------**

**!-----< 오라클 설치용 Linux VMWare 이미지 설치 >------------------------**

! 리눅스위에 오라클설치하여 admin 연습

! **DBA기본과정**: 오라클, 리눅스, 네트워크

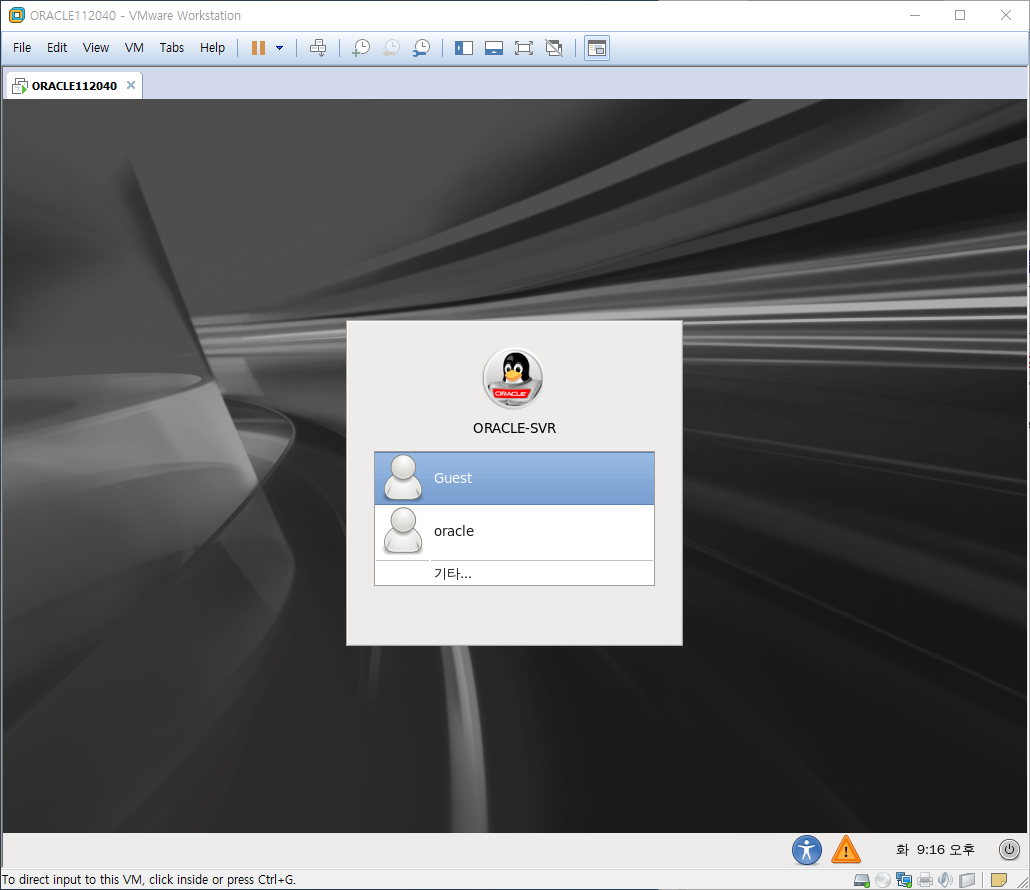
**🡪 SB의 oracle-linux6.3-64bit\_설치.egg**를 **d:\oracle폴더**로 복사*(노트북: e:\oracle)*

🡪 **oracle-linux6.3-64bit\_설치.egg** 압출풀기

**🡪 d:\oracle\ORACLE-Linux6.3-64bit\_설치\ORACLE112040.vmx** 클릭

🡪 I copy it

🡪 play 버튼



**!-----------네이버 클라우드 내려받기-------------------**

1. Admin🡪Linux🡪**01\_1 Linux기본명령.pdf**

2. Admin🡪Linux🡪**02\_Linux\_vi편집기.pdf**

2. Admin🡪Linux\_11gR2\_database🡪**p13390677\_112040\_Linux-x86-64\_1of7 및 2of7**

**!--------Linux용 오라클 설치화일 내려받기 ------------**

! 오라클 홈피에서 설치파일 다운로드

- 오라클 홈피🡪다운로드 🡪 Database 🡪 Database 11g Enterprise/Standard Editions

**➀ p13390677\_112040\_Linux-x86-64\_1of7.zip**

**➁ p13390677\_112040\_Linux-x86-64\_2of7.zip**

**!-------------------------------------------------------------**

**! 리눅스 로그인 :** 초기화면🡪**기타**

- 사용자이름 **root**

- 비밀번호 **oracle**

! **바탕화면 🡪 오른쪽 마우스 🡪 open in terminal**



**----< 리눅스 기본명령어 >----------------------------**

# **su** – **oracle** *--* ***su****:switch user ,* ***-*** *: bash\_profile실행*

$

! 윈도우에서 이동 cd c:\

! unix에서 최상위경로는 🡪 **/** *-- root경로 = 윈도우의 c:\*

# **pwd** *-- 현재 디렉토리*

*/root/바탕화면*

# **cd /** *-- 최상위 경로로 (절대경로 지정)*

# **cd ../../** *-- 최상위 경로로 (상대경로 지정)*

# **cd ~** *-- 홈디렉토리로 이동*

# **cd $home** *-- 홈디렉토리로 이동*

# **ls**  *-- 하위 파일, 디렉토리 조회*

# **cd /home** *-- /home으로 이동*

# **ls**

# **cd oracle** *--/home 하위폴더 oracle로 이동*

**===== 190130 (수) Day 13 ===================================================**

[root@oracle-svr oracle]# **ifconfig –a**  *-- network config확인*

**! VMware설치 vs 오라클**

|  |  |  |
| --- | --- | --- |
| **Host Computer** | | **🡪 인터넷망** |
| **hardware** | **500G** |  |
| **memory** | **16G** |  |
| **CPU** |  |  |
| **LAN** | **192.168.0.100** |  |

**! VM가상머신 설치**

🡪 새로만들기

- hw 얼마사용? 200G

- 메모리 얼마나? 4G

- CPU갯수?

- Network설정 🡪 **bridge방식, host\_only방식, NAT방식**

🡪가상OS

**!------< 리눅스에 오라클 설치 순서 >-------------------**

**1 리눅스 설치**

- 리눅스 ISO CD삽입 (여기서는 🡪oracle-linux6.3-64bit\_설치.egg)

**2. VMWARE tools설치** for shared 기능

- shared기능: host computer특정폴더와 linux특정디렉토리 공유기능

*@리눅스 /mnt/hgfs/myfolder 공유*

*@윈도우 d:\myfolder*

**3. Network설정**

**4. 오라클 설치화일을 FTP로 전송 (window🡪 linux)**

- **xmanager** 프로그램 사용

**5. 압출풀기 & 설정**

**6. 오라클 설치 on linux**

**-------< 리눅스에 오라클 설치 순서 >----------끝---------!**

**!** 스위칭역할- **VMNET**

- **VMNET 0번** -bridge방식

- **VMNET 1번** –host only

- **VMNET 8번** -NAT방식

**<Virtual Machine에서 네트워킹이 되게하려면>**

1. 인터넷 🡪 PC LAN 🡪 VMNET0 or 1 or 8 (**bridge방식**: 많이 사용) 또는,

2. 인터넷 🡪 내컴퓨터와 별도로, **독립된 Guest Linux 가상머신**을

동일 IP로 별도 연결 (**NAT방식**)

- 192.18.0.100 🡪 gate way 🡪 IP별도 할당 @ vmnet 8: 192.168.0.555 또는,

3.가상머신끼리만 연결할 때 (이중화방식) 🡪(**Host only 방식**:잘 안씀)

-- 오라클 이중화시

**!------- LINUX 네트워크 설정 ----------------------------------**

[root@oracle-svr oracle]# **ifconfig –a**

@LINUX Window> **시스템🡪 기본설정 🡪 네트워크연결**

! 기존설정 2개 지우기

**! 추가**

연결이름[N]: eth0

모든사용자 사용가능: check

Ipv4설정 탭 🡪 방식[M] **자동**(DHCP)

적용

# **service network restart** *(@root계정)*

# **ifconfig –a**

- inet addr: 192.168.10.132 Bcast: 192.168.10.255 Mask:255.255.255.0

*- (노트북) inet addr: 192.168.137.129 Bcast: 192.168.137.255 Mask: 255.255.255.0*

@LINUX Window> **시스템🡪 기본설정 🡪 네트워크연결**

! **eth0 변경하기**

IP4설정 탭🡪방식[M]: **수동**

주소🡪추가: 192.168.10.132 *(집 192.168.32.129)* *(노트북 192.168.137.129) +Tab키*

🡪넷마스크: 24 (유지)

DNS서버: 168.126.63.1 (집에서도 동일?)255

# **service network restart** *(@root계정)* -- **OK** 사인 3개 확인

*- eth0 인터페이스 활성화됨, 활성화된 연결경로 /org/freedeskto~~~~*

**!-----< X-manager 설치 및 오라클설치화일 전송>------------**

! X-manager 용도: window 설치화일을 🡪 리눅스로 전송

! @MS Windows에서, 복사한 xmanager 파일폴더에서,

🡪 **xme40.exe** 더블클릭

🡪 **key.txt** 에서 키값 찾아서 입력

! @바탕화면에서, **Xmanager Enterprise 4** 아이콘 클릭

🡪 **xshell** 클릭

! 파일>**새로만들기**

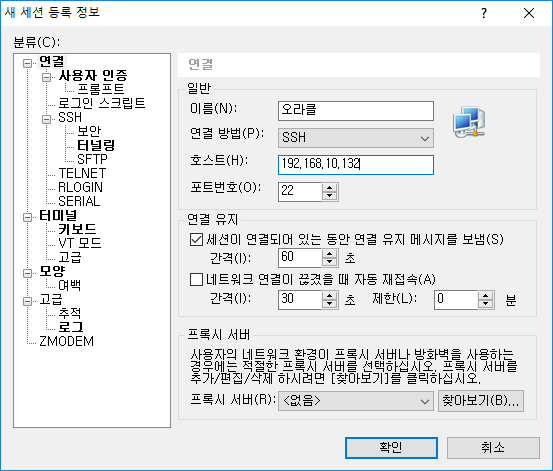
세션이름: **oracle (임의로)**

호스트: **192.168.10.132** (자기 PC IP주소)

🡪 확인

로그인이름: **root**

비밀번호: **oracle**



**----- X-manager 설정 ----------!**

@Xshell4 초기화면 상단 아이콘중 **새화일전송ICON** (Ctrl+Alt+F) click

🡪 *192.168.137.129* -**Xftp4**창 열림

- ***왼쪽****이 WIndow화일,* ***오른쪽창****이 리눅스화일*

**!** **windows**에 저장된 리눅스용 오라클 설치화일 2개를 리눅스로 이동

- **p1339~ 1of7.zip**과 **p1339~ 2of7.zip**파일 🡪 **리눅스** **/u01**로 끌어서 이동

**!------< 리눅스에 오라클 설치하기 > -------------------------------**

**🡪 0130-4 4.Oracle-Linux 6.3 64-bit 에 ORACLE 11.2.0.4.0 설치하기.pdf**

<목차>

1. root 로 로그인

2. GUI 환경 설정

**3. Hardware (memory, disk) check**

**4. Software (pakage) check**

**5. 필요한 운영 체제 그룹 및 사용자 만들기**

**6. Kernel 매개 변수 및 제한 설정**

**7. 설치 디렉토리 생성 및 권한 설정**

**8. 제품 디스크 탑재**

**9. oracle 11.2.0.4 software 설치**

!-----< **설치화일 압축풀기** >--------------------------------

#**cd /u01**

#**ls** -- 파일 2개 확인

- p13390677\_1120410\_Linux-x86-64\_1of7.zip

- p13390677\_112040\_Linux-x86-64\_2of7.zip

#**unzip** p1*탭키*\_1*탭키* -- 1번화일 압축풀기

#**unzip** p1*탭키*\_2*탭키* -- 2번화일 압축풀기

**!** 창하나 더열기: **Ctr+Shift+T**

**!--- 3. Hardwarer Check -------------------------**

**! 메모리 확인**

#grep MemTotal /proc/meminfo

- 오라클 운영시 1GB는 최소한도, only for test

- 실제운영시 128GB정도

**! 스왑공간 확인**

#grep SwapTotal /pro/meminfo

- 메모리 대체공간, 실제로는 안쓰는것이 좋은상태

**! 프리공간 확인**

#**free** *-- 남는공간 확인*

**! 디스크공간 확인**

#**df –h /tmp** *-- h: human 이 보기쉬운 상태로*

#**df /tmp**

**!** **Linux x86-64**에서 설치 유형 별 소프트웨어 및 데이터 파일 **디스크공간 요구사항**

- Enterprise Edition : 소프트웨어 4.7GB, 데이터 파일 1.7GB

- standard Edition : 소프트웨어 4.6GB, 데이터 파일 1.5GB

# **df –h** /tmp *-- /tmp 내 available 공간 확인*

**!--- 4. software(package) check ------------------**

# rpm –q binutils *-- rpm: redhat package manager , -q 질문모드*

**!--- 5. 필요한 운영 체제 그룹 및 사용자 만들기 ---**

**! 설치여부 확인 (생략가능)**

# cat /etc/group | grep dba

# cat /etc/group | grep oinstall

# cat /etc/passwd | grep oracle

**! Oracle User 환경 설정**

#**su** – **oracle**  *-- oracle user로 login*

*- /home/oracle – 오라클 유저의 홈디렉토리*

*- 유저의 환경변수(.bash\_profile) 🡪 유저의 홈디렉토리로 저장*

$ **pwd**

- /home/oracle

$ **ls –al** -- -a: all, l: 모두

*- .****bash\_profile*** *확인 -- 오라클유저의 환경변수화일*

*- oracle로 로그인할떄 .bash\_profile을 읽어서 환경 설정 (#su '-' 의역할)*

**$vi .bash\_profile** *-- oracle 홈디렉토리(/home/oracle)에 있는 .bash\_profile 수정*

vi)shfit-G *-- 맨아래줄로 이동*

vi)o *-- 아래줄에* *끼워넣기*

vi) 자료 4page oracle user 환경설정 box내용 맨 끝에 추가 입력

umask 022

export ORACLE\_BASE=/u01/app/oracle

export ORACLE\_HOME=$ORACLE\_BASE/product/11.2.0 *(****neca****설치시 오류나면 /11.2 로 수정)*

export ORACLE\_SID=ORCL

PATH=$ORACLE\_HOME/bin:$PATH

export LANG=en

export EDITOR=vi

vi)esc *-- 명령모드 전환*

vi)shift + : *-- 저장모드 전환*

vi)wq

$cat .bash\_profile -- cat: 조회하기

- **ORACLE\_BASE** : 로그화일등 저장 기본경로

- **ORACLE\_HOME**: 엔진경로

- **ORACLE\_SID** : 오라클 시스템명?

$echo $ORACLE\_BASE

$**.** .**bash\_profile** *-- 앞의”* ***.*** *“: 실행*

$**echo** $ORACLE\_BASE

*/u01/app/oracle*

**! --- 6. kernel 변수 및 제한 설정 ----------**

! 오라클 11g버전 이전에서는 일일이 설정해야 했으나, 11g부터 아무때나 script로 편하게 설정

*-- ORACLE 설치 단계에서 Kernel parameter가 적절하지 않을 경우*

*이를 수정하는 스크립트를 제공해준다.*

*-- 이 때 제공되는 스크립트만 실행하면 적절한 파라미터 값으로 자동 변경된다.*

🡪 so let’s SKIP!!!

**! --- 7. 설치디렉토리 생성 및 권한 설정 ---**

$**exit**

#**cd /**

#**ls -al**

- /u01 확인 *– 맨앞에 d이면 디렉토리*

- 소유그룹=root, 소유자=root ?

# **chown –R oracle:dba /u01**

- **chown**: change owner and group *-- 파일의 소유주와 그룹을 바끔*

- **user**: oracle

- **소유그룹**: dba

- **-R**: recursive 경로와 그 하위파일을 모두 바끔

# **ls -al**

- u01 폴더의 user와 group이 oracle dba로 변경된거 확인

**!--- 8 제품 탐재 ------------------------------**

**!--- 9. ORACLE 11.2.0.4.0 Software 설치 ---**

! 먼저 유저에게 GUI환경설정

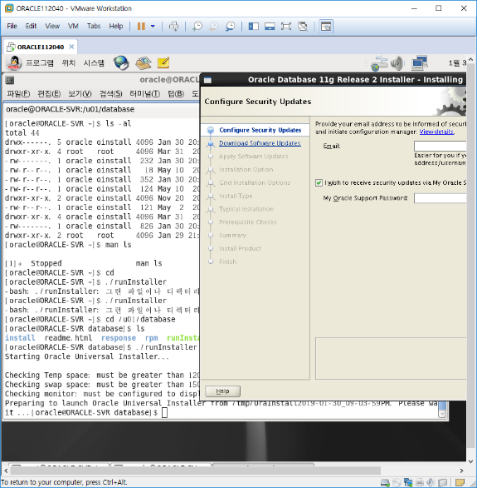
**# xhost +**

# **su – oracle**

$**xclock**

🡪 시계화면이 떠야함!! 🡪 Ctrl+C

$cd **/u01/database**

$ **./runInstaller** *-- runInstaller 파일 실행*

(이후 상세과정은 🡪 **Oracle Linux6.3 64bit에 ORACLE11.2.0.4.0설치하기.pdf** 참조)

🡪 I wish to receive security updates~~ checkbox해제

🡪 Skip software update : check후 next

🡪 Install database software only : check 후 next

🡪 Single instance database installation : check 후 next

🡪 Korean 추가후 next

🡪 Enterprise Endition 선택후 next

🡪 Inventory directory 확인후 next

🡪 Database Administrator group은 dba로 선택한후 next

@ prerequite checks

Fix & Check Again탭

- runfixup.sh 실행

- swap 메모리 부족은 무시하고 진행해도 됨

------------------------------------------------------------------ ---

**🡪 0130-5 Oracle NLS.pdf**

Oracle NLS 지원특성

Oracle 제품올바르게설치하기

올바른캐릭터셋선택

- 대부분 KO16MSWIN99 선택 (주로 한글을 사용하는 서버)

올바른NLS 환경변수값설정하기

------------------------------------------------------------------ ---

**===== 190131 (목) Day 14 ===================================================**

**!--- 10. ORACLE network설정(Listener생성, netca실행)**

**! 네트워크 설정 = 리스너설치과정**

! 사전작업: IP 확인, Host name 확인

**#ifconfig –a**

- inet 192.168.10.132 (@학원pc) *192.168.137.129(노트북)*

**# vi /etc/hosts** *-- ip주소 수정*

- 첫줄 127.0.0.1 : 자기 roop back ip주소

- **ip주소 변경** to **192.168.10.132 --** *192.168.137.129(노트북)*

- Shift+G: 명령모드에서 커서를 맨아래로

- Shift+R: 명령모드에서 replace

- 192.168.10.132 입력

vi) esc

vi) :wq

**#xhost +**

**#su – oracle**

**$netca -- netca 실행**

*! 리스너 프로세스 확인 (window의 프로세스처럼)*

$**ps –ef | grep LISTENER --ps: process, grep:보여줘**

- /u01/app/oracle/product/11.2/bin/tnslsnr …..

$**echo** $ORACLE\_HOME

$**cd $ORACLE\_HOME/network/admin**

$**ls**

- **listener.ora** 확인

$**cat** **listener.ora** *-- 리스너가 올라와 있어야 오라클 DB가 설치됨!!*

- ADR\_BASE\_LISTENER = /u01/app/oracle

\* **자동완성기능**: 앞의 몇문자+tab

**!--- 11. ORACLE Database 생서(dbca 실행) ---**

*! 오라클 생성 @oracle 유저 -------------*

$**dbca** *-- @oracle유저, DB생성명령*

🡪"11.Oracle Database 생성" **설치매뉴얼** 순서 참조

**! ---2교시------ vi 편집기 연습-----------------------------------------------------**

🡪 **0129-2 02\_linux\_vi 편집기.pdf**

$exit -- 원래 root로 로그인해서 oracle로 넘어왔으므로

#**su – oracle**

$**cd $ORACLE\_BASE/diag/rdbms/orcl/ORCL/trace** ---- 오라클 로그화일 폴더

*- $ORACLE\_BASE = /u01/app/oracle*

$ls –alTAB

$vi a.txt

!**스왑파일 삭제**

- vi 편집중 강제 종료시키면, swap 파일이 해당 폴더에 남아있게됨

- 다시 같은 파일을 vi편집기로 실행하려면 🡪 메시지

$**rm –rf** .a.txt\* -- 스왑파일 삭제

**! ---3교시------오라클 구조----------------------------------------------------------**

**0131-2 dml\_parsing.pdf**

**! 오라클영역**

- 인스턴스영역 *-- @메모리*

- 데이터베이스영역 *-- @디스크*

**! 인스턴스영역**

- SGA영역 *--(System Global Area)*

- BP영역 *--(Background process)*

**! 데이터베이스 파일 종류** *@데이터베이스영역*

- **control** file

- **redo log** file

- **Data** file

**! Data file**

- System화일

- 일반data 파일

**! 데이터 저장단위** = BLOCK (8192 byte)

- employees테이블 🡪 /u01/oradata/ORCL/example.dbf 어딘가에 저장

**! 유저가 SELECT ename FROM emp WHERE empno=7788**

**🡪 유저프로세스 생성**

**! 서버는 서버 프로세스 생성**

- 1단계: parsing

- 2단계: excute

- 3단계: fetch

**! PARSING** 처리순서 -- parsing: 어구의 해석

1)**구문검사** -- 문법검사, 오타검사 …

2)**의미검사** -- 권한 검사, 테이블 존재여부, 컬럼 존재여부 …

3)**ASCII 코드화** 🡪 hashvalue생성 (예:0001)

4)똑 같은 hashvalue값이 메모리에 있는지 확인 **@library cache 영역**

5)library cache에 있으면 🡪 재활용(soft caching)

메모리에 없으면, 최적의 **실행계획** 작성 (hard parsing)

6)**hashvalue** (0002)와 **sql text**와 **parsing code**와 **실행계획**을

🡪 메모리의 library cache에 저장 -- called hard parsing!!

**! EXCUTE** (실행) : 디스크로부터 data를 가져오는 행위

1)메모리에서 employees data가 있는지 확인 **@Buffer cache 영역**

2)**메모리 있으면 재사용**, 없으면 디스크에서 buffer cache로 데이터를 가져옴

- High Water Mark

- 데이터는 row단위로 메모리로 가져가는게 아니고(**IO단위=** BLOCK)

**block 단위**로 HWM설정 범위 전체를 탑재

(60번 직원 데이터가 속해있는 block 의 HWM범위 전체를 탑재)

- block 단위 탑재는 무거운 작업

*\* buffer cache 히트율이 높으면 좋음 – DBA 튜닝 목표*

**!** **FETCH** 단계 *-- fetch 명령문을 꺼내다, 가서 가져오다*

- only select 문 실행시에만 있음

- 유저프로세스에게 결과를 전달

- Server Process 🡪 User Process 🡪 Display

! cursor: library cache에 공간을 차지하고 있는 것

- 0001 a-l~~

**!--------------- oracle database 영역 -------------------------------------**

**INSTANCE @ 메모리**

redo log buffer

SGA영역

buffer

cache

Shared pool

library cache

BP영역

**Database @ 디스크**

**redo log**

**file**

control file

**Data file**

- System file

일반 Data file

-----------------------------------------------------------------------------!

**! ---4교시------오라클 구조----------------------------------------------------------**

! cache는 **LRU-MRU 알고리즘** 적용

- 자주쓰는건 MRU쪽으로, 안쓰는건 LRU 쪽으로 가서 aging out됨

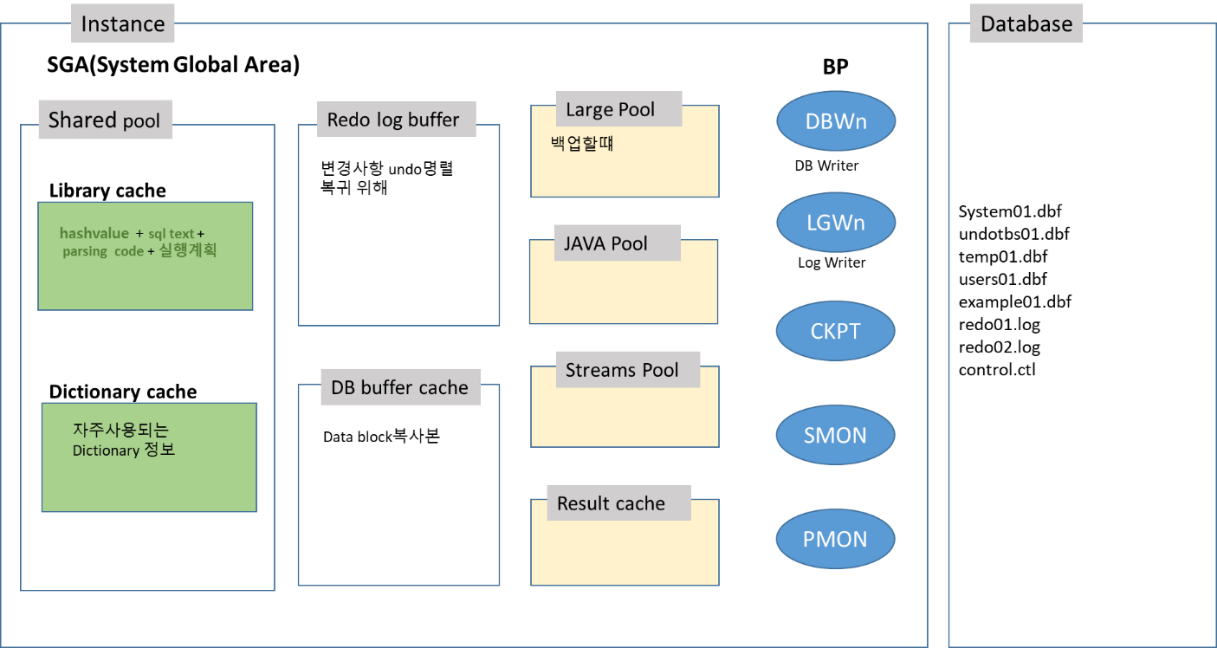
LRU MRU

**! SYSTEM file *@ /u01/oradata/ORCL***

- **system01**.dbf 🡨 딕셔너리 등 관리정보 저장

- **undotbs01**.dbf 🡨 update, delete시 변경전 이미지 저장

- **temp01**.dbf 🡨 정렬할수 있는 공간 (메모리에 정렬공간이 부족할시 사용)

! **메모리공간**

update employees set sal=100 where emp\_id=200;

(이전값=50)

1. parsing

2. excute

1)DB buffer cache 에 있는지 찾아

2)transaction slot 확보 from **undotbs01.dbf (**이전이미지 저장공간 확보, if not 🡪 실패)

3)Row level Rock

4)redo log buffer에 **명령문**과 **undo할명령문** 같이 저장

5)undotbs01.dbf에 **이전이미지**(값=50)저장

Commit하면   
🡪 data buffer에 값을 변경완료 flag 표시(dirty상태)   
🡪 DBWn(DB Writer 프로세스) 가 특정시기에 buffer 캐시내용을 디스크에 새값(100)으로 저장  
🡪 redo log 내용(명령문+undo명령문)을 LGWR 프로세스가 **Redo log 파일**에 기록

! **redolog 파일**: 복구시 사용

- redo log buffer는 entry(행)단위로 IO실행 - 속도 빠름

-------------- 정리 ----------------

**0131-2 DML parsing**

SQL Parsing

SELECT SQL Parsing

SELECT SQL Hard Parsing

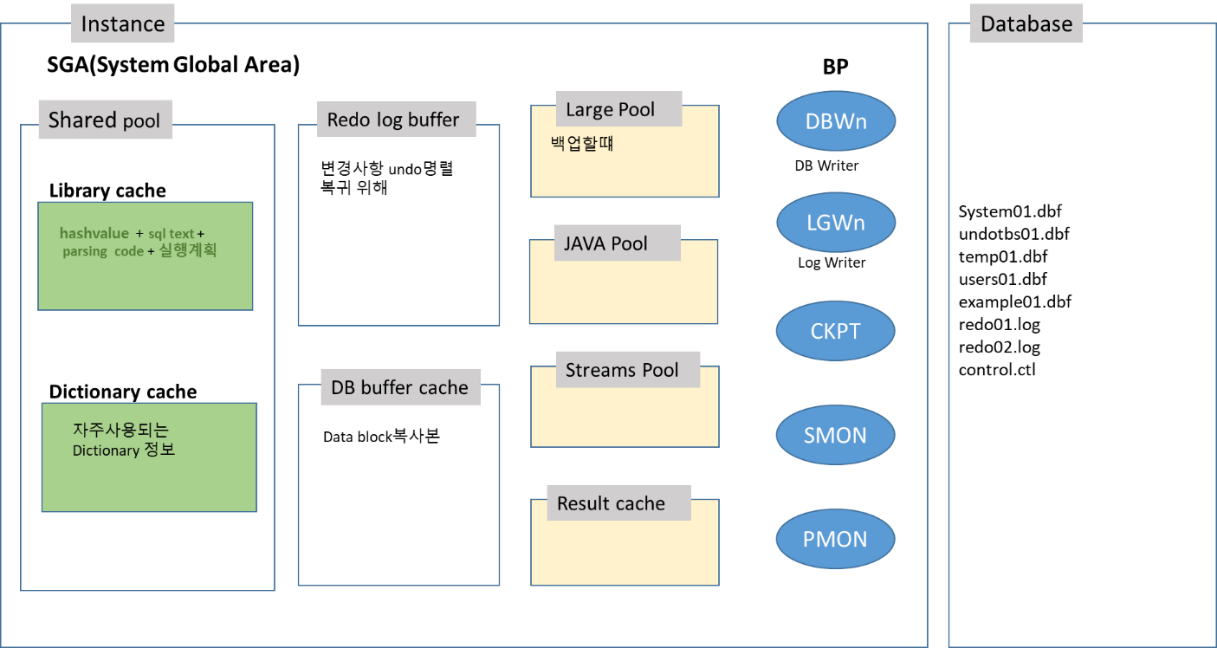
Hard Parsing

SELECT SQL Execute

**===== 190201 (금) Day 15 ===================================================**

**🡪 0131-3 0.Oracle\_Database\_Architecture.pdf**

**🡪 0131-1 Oracle\_11g\_architecture.png**



**---------< Background Processes >---------------------------------------------------------------**

**! DBW :** DB Writer -- DB Buffer상 **dirty block**을 주기적으로 disk의 **datafile**에 기록하는 프로세스

- 발생 시기

. checkpoint 발생 시

. free buffer 부족 시

. dirty block 수가 임계치에 도달했을 때

**! LGW :** log writer **-- redo log buffer**의 내용을 **redo log file** 에 내려쓰는 프로세스

- DBW가 일하기 전에 복구를 위해 Log Writer가 먼저 일을 함

- 발생시기

. 트랜잭션 commit 시

. redo log buffer 의 1/3이 찼을 때

. 매 3초 마다, 1MB 이상의 redo 발생 시

. DBWn가 dirty block을 datafile에 쓰기 직전

**! CKPT :** Checkpoint Process: 동기화 (메모리와 디스크 내용을 동일하게 유지)

- Checkpoint event 발생 시

. DBWn에 신호를 보내고 dirty block을 datafile에 기록

. Control file과 Datafile Header 정보를 최신의 물리적 정보로 갱신

- 발생시기

. Database 정상 종료 시 *-- 동기화*

. alter system checkpoint 명령 수행 시

. Tablespace가 offline 될 때

. log switch 발생 시

. checkpoint 주기마다

! **동기화**: DB buffer내용 datafile 내용과 동일하게

- **CKPT**가 DBW에게 지시 🡪 DBW는 우선 LGW에게 로그기록 시킨후 작업

**! SMON Process** : System Monitor proces

- **Instance Recovery 역할** (비정상 종료시)

- 사용하지않는 공간정리 (temp01.dbf화일)

- 비정상종료시 startup하면   
🡪 SMON이 date file header를 조회: SCN=100번, redo log파일의 scn=102번

🡪 SMON이 redo 로그 파일의 100~102번 발생된 명령문 재실행

🡪 메모리내 내용이 비정상종료 전으로 복귀

- **temp01.dbf** : 정렬을 위한 임시공간

🡪 SMON이 비사용 공간 정리

! **PMON Process** : Process Monitor Process

\* user가 접속하면 User Process 생성

🡪 대응하는 Server Process생성 (with **Process Grobal Area 공간** 확보)

\* user가 SQL문장 수행시

1)transaction slot 배정

2)row Lock

3)실행

- user가 비정상 종료시 🡪 **PMON이 프로세스 Recovery 수행**

. DB Buffer Cache 정리

. User Process가 사용하던 리소스 정리: transaction slot, row lock ..)

- Idel session time-out에 대한 session monitor 수행

- 리스너에 동적으로 DB서비스 등록

! 5개 백그라운드 프로세스(BP)중 하나만 죽어도 DB는 비정상 종료

- **DBW, LGW, CKPT, SMON, PMON**

------------- \*--------------------------------------------------

**!** Windwos > 탐색기 > e:\oracle>**orcl** *-- data 파일들이 저장되어 있는곳*

! e:\oracle>**orcl 🡪 redo01.log, redo02.log, redo03.log**

! **redo file**: redo buffer 내용을 내려쓰는 파일 🡪 50MB세개화일이 순환하면서 덮어씀

- called **NO Archive 모드**

- log switching : 파일 순환

- log switching 발생시 별도 파일로 저장하면 **Archive모드** ---(복사) >1-1.arc, 1-2.arc

**!** **ARCn**: ! Archive Processes *\* archive : 기록보관소, 데이터의 보관/보존*

- 세개 Redo파일간에 로그 스위치가 발생시 🡪 Redo Log File을 지정된 기억장치로 복사

! 권장사항: **Archieve mode**

**---------< Storage Architecture >---------------------------------------------------------------**

! **System Data File** : 필수화일

- **system01.dbf** : dictionary정보

- **sysaux01.dbf** : 10g부터 생성, AWR (automatic work load repository) 저장

- **undotbs01.dbf** : 이전 이미지 저장

- **temp01.dbf**  : 정렬용 공간, 임시테이블 저장(주문정보)

! **일반 Data File** : no필수

- **users01.dbf** : 유저 생성 data (expert유저의 employees 테이블, departments테이블…)

- **example01.dbf**:

! **Control File**: DB에 존재하는 모든화일, redo로그파일에 대한 **위치정보 + 상태 + 최신SCN정보**

- **control01.ctl, control02.ctl**

-startup 시 🡪 control file로 모든 파일의 위치정보 + 최신 SCN정보 확인

**---------< Oracle Database Administration Workshop 교재 >--------------------------------**

**🡪 0201-1 Oracle Database 11g Administration Workshop I\_sg1.pdf**

p3. 목차

1 **Exploring the Oracle Database Architecture**

2 Installing your Oracle Software

3 Creating an Oracle Database Using DBCA

4 Managing the Database Instance

5 Managing the ASM Instance -- 스토리지관리기능 (생략!!!)

6 Configuring the Oracle Network Environment

7 **Managing Database Storage Structures** -- 어제오늘 공부한거 후속 , 중요!

8 Administering User Security -- 유저생성, 권한관리

9 Managing Data Concurrency -- 락 관리

10 Managing Undo Data

11 Implementing Oracle Database Auditing --감사, 접속정보-저장부하

12 Database Maintenance – AWR(유료기능)

13 Performance Management -- 성능, 메모리 관리

14 Backup and Recovery Concepts -- 백업방법론

15 Performing Database Backups --실제백업

16 Performing Database Recovery -- 몇가지 실습

17 Moving Data --자료만 제공

**---------1장 Exploring the Oracle Database Architecture-------------------**

**🡪 0131-3 0.Oracle\_Database\_Architecture.pdf**

! Linux start : root/oracle

! Oracle112040 시스템 확인 (@학원PC)

----------------------------------------------------

1. root 로 log-in \_ passwd=oracle

2. oracle 유저로 스위칭

# **su** **– oracle**

3. sqlplus로 오라클 접속

$ **sqlplus "/as sysdba"**

SQL> select instance\_name from **v$Instance**

*orcl*

SQL> **show sga**; *-- sga영역 보여줘*

*- Total system area : 676MB*

*- Fixed size : 2.5MB*

*- Variable size : 440MB*

*- Data buffers : 230MB*

*- Redo buffers : 3MB*

**---------4장 Managing the Databse Instance------------------**

**I4-12 Initializing Parameter Files**

# **su – oracle** -- '-' 오라클 유저의 환경화일을 읽고 들어가라 :

$cat **.bash\_profile**

! sqlplus 실행파일 🡪 path 지정된곳에

# su – oracle

$**splplus "/as sysdba"**

🡪 ORACLE\_HOME/dbs/**spfileorcl.ora** 를 읽어서 환경설정

\* (화일내용) 메모리사이즈, block size, db이름, **Control File위치, Undo정보,** process갯수

**I4-14 Simplified Initialization Parameters**

! 초기화 파라메터들

CONTROL\_FILES

DB\_BLOCK\_SIZE

PROCESSES

UNDO\_TABLESPACE

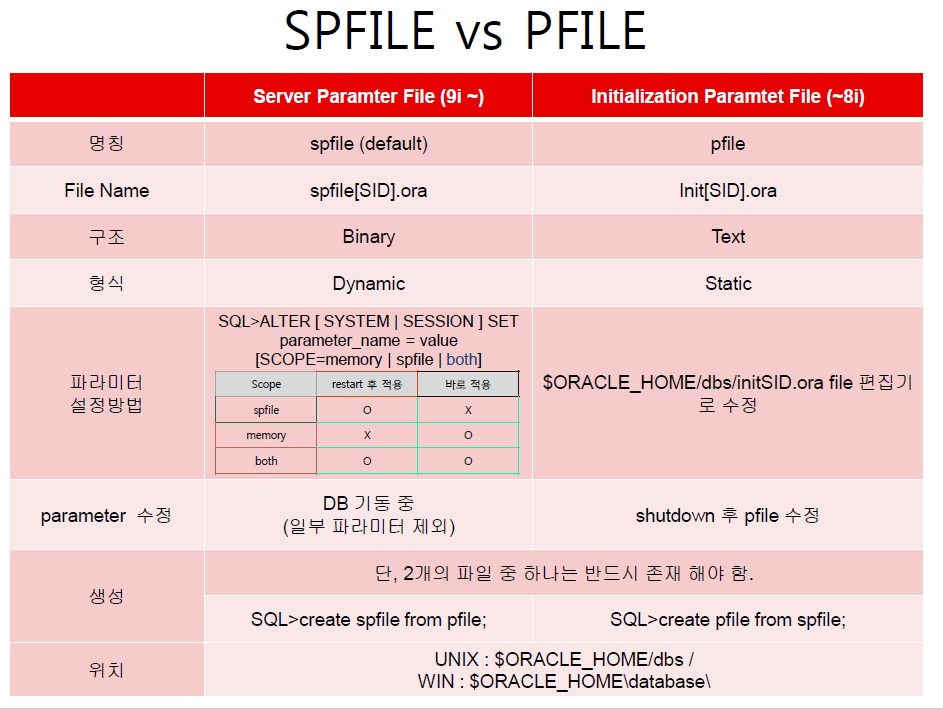
…

DB\_CACHE\_SIZE

DB\_FILE\_MULTIBLOCK

\_READ\_COUNT

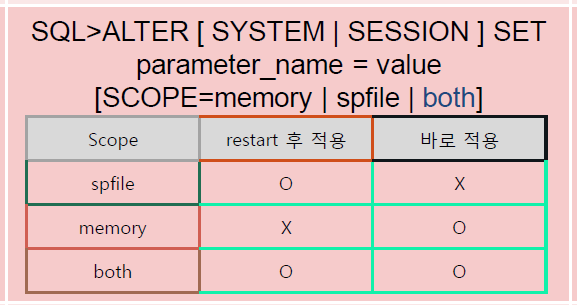
SHARED\_POOL\_SIZE

**!----- 0201-3 SPFILE vs PFILE.pdf -----------------------------**

! 파라메터화일은 **SPFILE**과 **PFILE**중 하나만 선택가능

- pfile: text 파일 - parameter수정하면 나갔다 들어와야

- spfile: binary file - 운영중에 SQL>Alter명령으로 즉시 수정



SQL> alter system set share\_pool\_size = 1G scope=both – 메모리와 spfile 동시적용

SQL> alter system set share\_pool\_size = 2G scope=spfile ; -- 파일만수정🡪내렸다 올렸을떄 적용

SQL> alter system set share\_pool\_size = 3G scope=memory; -- 임시로 메모리에만 적용

!-------------------------------------------------------

**SQL>startup**

**#su – oracle**

**$splplus "/as sysdba"**

🡪 $ORACLE\_HOME/dbs/**초기화파라미터** 읽어들이는데,

1. **spfileSID**.ora 찾아(spfileORCL.ora), 없으면

2. **spfile**.ora 찾아, 없으면

3. **initSID**.ora 찾아 (init**ORCL**,but 샘플화일), 없으면

Error

! **초기화 파일 위치**

- LINUX : **$ORACLE\_HOME/dbs** (= /u01/app/oracle/product/11.2.0)

- WIN : $ORACLE\_HOME\database\ (=E:\oracle\product\11.2.0\dbhome\_1\database)

! **초기화파일 생성**

- SQL>create spfile from pfile;

- SQL>create pfile from spfile;

----------------------- -----------------------------------

spfileORCL.ora 운영중

alter 명령으로 spfile의 파라미터 수정 (잘못수정!)

database shutdown 발생

startup

error

create pfile from spfile

initORCL.ora 수정

----------------------- -----------------------------------

$cd $ORACLE\_HOME/dbs;

$ls

- spfileORCL.ora확인

$cat spfileORCL.ora

SQL>create pfile from spfile

- $ORACLE\_HOME/dbs 밑에 initORCL.ora 생성 확인

---------------------------- ----------------------------

**!현재 운영중이 DB가 pfile/spfile어떤걸로 운영중인지 확인**

SQL>show parameter spfile

- value절에 경로를 포함한 spfile이 나오면 spfile로 운영중

- value절에 null 값이면 pfile로 운영중

---------------------------- ----------------------------

**! parameter변경**

1)system레벨: **alter system set**

- 전체 시스템에 영향

2)session레벨: **alter session set**

- 현재 접속한 세션만 적용

sql>**show parameter process**

*🡪 processes: 150*

sql>alter system set processes=200

*🡪 오류*

sql>alter system set processes=120 scope=spfile

*🡪 system altered*

sql>shutdown

sql>start

sql>show parameter processes

------------------------------------------------- ---

sql>show parameter sga\_target

*🡪 sga\_target 0*

! sga\_target : 여러 파라메터를 자동관리

! memory\_target

sql>exit

$vi spfileORCL.ora ($oracle\_hoe/dbs)

- memory\_target : 500M로 변경

**===== 190207 (목) Day 16 ===================================================**

4-12 Initialization Parameter Files

- spfile : 10g부터, binary file, alter system set으로 변경, 기동중 변경

. scope= memory에만적용, file에만 적용, both 적용

- pfile : 9i 까지, text file, shutdown후 restart 필요

- spfile, pfile 둘중 하나도 없으면 DB start 못함

- 기동에 필요한 control file에 spfile에 기록

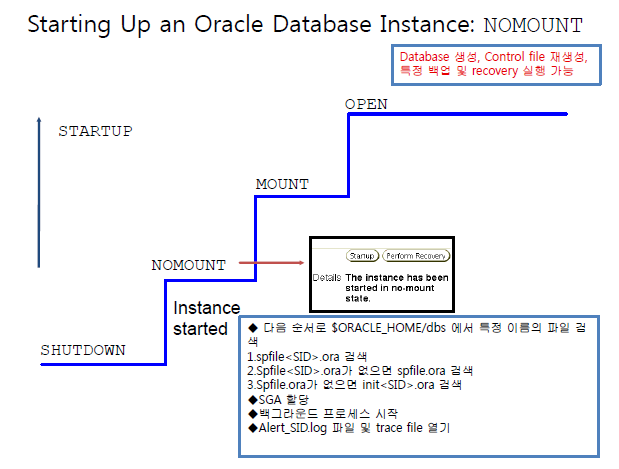
- create spfile from pfile :

- create pfile from spfile: db가 내려진 상태에서도 수정 목적

- spfile위치 : $ORACLE\_HOME/dbs

**4-26 Database Startup and Shutdown: Credentials**

**🡪 0207-1 Oracle Startup,Shutdown 단계.pdf**



startup 🡪 nomout 🡪 mount 🡪 open

startup nomount -- nomount단계 까지만

--------------------------------- ------------------------

#**su – oracle**

.**bash\_profile을** 읽어서 필수 경로 읽어들임,

oracle\_SID

ORACLE\_HOME

ORACLE\_BASE

PATH= $ORACLE\_HOME/bin -- 실행화일이 있는곳

등의 정보를 읽어들임

$sqlplus "/as sysdba"

에러나면 🡪 path 설정 점검 ….

--------------------------------- ------------------------

**🡪 0207-1 Oracle Startup,Shutdown 단계.pdf**

SQL>**startup** 명령치면

**nomount 단계로**

**$ORACLE\_HOME/dbs/spfileOACL.ora 파일**을 찾음

없으면 **spfile.ora** 파일을 찾음

없으면 **initORCL.ora** 파일을 찾음 -- UNIX는 대소문자 구분

🡪**SGA영역 할당** -- ora화일 설정대로

🡪**BP영역 할당** -- SMON, PMON, LGW, DBW, CKPT

🡪alert\_ORCL.log기록 시작 – 시스템에 영향을 미치는 로그 기록

🡪trace file기록시작

- 할수 있는일 : database 생성, control file 재생성, 특정백업 및 recovery

**mount 단계로** 진입

**control file 읽어들임**

-- control file이 깨지거나, 위치가 잘못기록되어 있으면 오류!!

-- control file: 모든 데이터파일과 redo로그파일, 최신 SCN 정보 위치 기록

-- but file의 **유효성 검사는 하지 않음**

\* password: system01.dbf 에 저장

-- 데이터파일을 아직 앍지 않았으므로 🡪 아직 일반유저 로그인 불가

-- sysdba는 로그인 가능

-- DB가 어떤단계로 살아있는지가 중요

- 할수 있는일: system data file 이름바꾸기, online redo log file 옵션 활성화/비활성화

전체 database recovery 수행

-- system01.dbf 파일 이름변경(운영중에는 변경불가)

-- redo log archive : 기본모드는 no-archive (운영중 변경 불가)

-- database recovery

**open 단계로** 올라가면

- **유효성 검사**

-- datafile, redologfile file열기 : header가 open

-- 일반유저 접속가능

- 할수 있는일: database 일관성 체크, SMON이 Instance recovery , 모든유저 접속가능

-- 기록된 SCN정보와 실제 메모리상의 scn number 확인

-- SMON이 redo로그파일을 읽어서(open단계) 필요시 instance recovery 수행

(시간 오래 걸림 🡪 )

------------------------- ---------------------------------------------------------------

**🡪 0207-1 Oracle Startup,Shutdown 단계.pdf - 4page**

터미널창 2개 open

! OS에서 oracle 백그라운드 확인

w1 oracle> ps –ef | grep smon

w2 oracle> **cd $ORACLE\_BASE/dig/rdbms/orcl/ORCL/trace**

w2>**tail –f alert\_ORCL.log --** 실시간 모니터링

w1>sqlplus "/as sysdba"

w1 sql>startup -- 창2 monitoring

w1 sql>select status from v$Instance -- DB open상태 보기

open

w1 sql **>! ps –ef | grep smon;** -- **!** : 오라클에서 OS명령어 입력시

w1 sql >alter user scott identified by tiger account unlock

w1 sql >conn scott/tiger

w1 sql >shutdown immediate – 일반유저는 권한없음

w1 sql >conn /as sysdba; -- sysdba는 모든권한을 가진 ROLE -- OS인증방식 사용

w1 sql >**shutdown immediate**; -- *w2>모니터링 – shutting down instance (immediate)*

w1 sql **>! ps –ef | grep smon**

smon 죽어있는거 확인 (ora\_smon)

---------------------------------------------------- --

w1 sql > **startup nomount** *-- starting oracle instance(normal) ~~~*

*-- processes=102 (초기화파라미터에서 설정한것들)*

*-- PMON started ~~~*

**w1** sql **>!ps –ef | grep smon**

smon 살아있는거 확인 (ora\_smon)

터미널 하나더 open : by Ctrl+Alt+T

w3> sqlplus scott/tiger -- 에러

w1 sql >select status from v$Instance

w1 sql >**alter database mount**; -- db를 nomount 🡪 mount단계로

w1 sql >select status from v$Instance

w3> sqlplus scott/tiger -- 에러

w1 sql > **alter database** **open**

w1 sql > conn scott/tiger; -- 일반유저 접속 ok

------------------- ----------------------------------

**🡪 0207-1 Oracle Startup,Shutdown 단계.pdf - 5page**

! **shutdown mode 4가지 연습**

scott> update emp set sal=20;

20개 변경

w3> sysdba

w3>shutdown -- 실무에서는 절대 사용 안함

w2> -- monitoring창

-- shutdown 대기중 : normal모드는 트랜젝션 종료까지 대기

w1>exit

w2> -- 정상 shutdown됨

w1 $> ps –ef | grep smon

아직 살아있음

w1 sql>sqlplus "/as sysdba"

w1 sql> shutdown abort

w1 sql>startup

w1> conn scott/tiger

w1> update emp set sal=20;

w3> conn /as sysdba

w3> shutdown transactional --transaction 종료시까지 대시

w1>rollback;

w3> -- shutdown 완료

! 실무에서는 shutdown transactional, normal 모두 사용안함

w3>startup

w1> conn scott/tiger

w1>update emp set sal=20;

w3>**shutdown immediate -- 실무에서 일반적으로 씀 (abort도 가끔)**

**-- commit되지 않은 세션은 rollback시킴**

w3> -- database 내려감, no waiting

w3>startup

w1>conn scott/tiger

w1>select sal from emp; -- sal값이 변경안되고 남아있음

! echeck point 발생 = 정상종료 – 동기화상태(memory와 disk가)

---------------------------------------------------------------------- --

클라우드에 oracle admin 한글교재 – 암호: **@oraclewikis!**

------------------- 메모리 관리방식----------------------------------

**🡪 0207-2 ORACLE\_MEMORY\_관리.pdf**

<메모리구조>

SGA영역: System global areas (Shared~)

Share pool

Library cache : 명령문+hassh value+실행계획 저장

Dictionary cache: 자주사용되는 dictionary 정보들

Log buffer: 명령문+undo명령문 저장

DB buffer: data블럭 복사본 저장

Large pool

Java pool

Streams pool

Backgrount Process

SMON

PMON

DBW

LGW

CKPT

ARC

SP(+PGA) : Server Process + Program global area

--------------- ----------------------------------- --------

<9i version>

shared\_pool\_size =150M

db\_cache\_size =300M

java\_pool\_size=0 ~~~~ 각각 fixed\_size로 설정해서 사용

9시: update 많이 사용 🡪 dba가 shared pool size 증가

15시: insert 많이 사용 🡪 dba가 db\_cachje\_size 증가

~~~~~

🡪 과거 수시로 library cache 부족으로 에러 발생

<10g version>

메모리관리기능을 자동화

**sga\_target** parameter 신설

- sga\_target =1G

- 나머지 parameter = 0 으로 설정

🡪 SGA 내부 메모리 운영을 1G안에서 시스템이 알아서 관리

\* sga\_target =0 이면 🡪 각각 수동으로 메모리 설정하겠다

\* sga\_target = xG이면 🡪 시스템이 알라서 하겠다 (**ASMM방식 @10g**)

\* ga\_target = xG + shared pool=100M 🡪 shared pool은 최소값 지정

----PGA-------------------------------------- ------------------------------

user> select \* from employees where dept\_id=50 order by empl\_id

🡪 Server process에게 전달

server> parsing~~~

server> deptid+50에 해당하는 buffer cache 데이터를 **PGA공간**으로 가져와서 (cursor station?)

정렬 (soft area)

🡪 자동관리 : **pga\_aggregate\_target parameter**로 설정

\* user생성 --- 1:1로 server process 생성 (pga공간 각각 500MB)

? sga 모자랄떄 pga메모리를 전용하면 좋겠다

🡪 11g부터 **memory\_target** parameter 신설 (**AMM방식**)

- **memory target = sga\_target(5개) + pga\_aggregate target**

- memory taget = 4G : 자동관리

- memory\_target =0 : 수동관리

- memory target=4G, sga\_target=1G : sga는 최소값 1G으로

- "memory\_target”, “memory\_max\_target” parameter 설정하면 활성화,

-----------실습------------------------------------------------------------------ --

conn /as sysdba

show parameter spfile -- 경로가 나타나면 spfile

경로가 안나타나면 pfile 로 운영중 🡪

create spfile from pfile ;

shutdown immediate

startup; -- startup force = shutdown abort+start (실무에서는 쓰지말 것!)

show parameter sga\_target

0

show parameter memory\_target;

648M – 값이 있으므로 현재 AMM기능으로 운영중

show parameter shared\_pool\_size;

0 -- 최소값 미설정

show parameter db\_cache\_size;

0

alter system set memory\_target=0; -- alert log확인

show parameter sag\_target

388M -- 자동으로 asmm방식으로 변경

show parameter pga\_aggregate\_tartget

260M

show parameter shared\_pool\_size

0 -- 진짜0이아니고, 확인하려면

**show component, current\_size, max\_size** from **v$memory\_dynamic\_components**;

**===== 190208 (금) Day 17 ===================================================**

ch5 asm은 single 구성에서는 사용하지 않으므로 생략

**🡪 0208-1 WS1\_ch06.ORACLE\_Network.pdf**

server -------------------------------------------- client

listener 필요 – netca명령/or수동구성

1)Oracle설치

- $ORACLE\_HOME 생성

2)Network구성 (=리스너생성작업)

- 자동구성 NETCA

- 수동구성: listener.ora 직접 작성

3)UP와 SP연결 -- 리스너는 빠짐

! 리스너=프로세스

- CLIENT의 접속요청이 오면 **확인**하고, SERVER프로세스를 생성하는 역할

확인정보: **$ORACLE\_HOME/network/admin**/**listener.ora** 정보 확인

**. HOST이름 or IP**

**. Port번호**

**. 프로토콜**

**. SID (DB이름)**

이미 서버측에서 이런정보 보유중이므로 이정보와

client가 보내온정보를 확인해서 연결

\* 오라클 instance start/shutdown 과 listener는 개별로 작동

- **서버단**: 리스너🡨 🡪 **클라이언트단: client용 오라클엔진** (가벼움)

! 클라이언트단 보유정보: $ORALCE\_HOME/network/admin/tnsnames.or 파일

IP 192.168.0.206

포트번호 1521

프로토콜 tcp

SID orcl 🡪 alias k\_ora

-- 복수정보 가능

IP 192.168.0.100

포트번호

프로토콜

SID 🡪 alias shared

! client>**sqlplus expert/expert**@k\_ora

🡪 User\_process가 tnsname.ora에 가서 alias.ora정보를 가지고

Listener 에게 연결

🡪 **Listener**는 서버정보와 가져온 정보를 확인해서 맞으면

**SP**와 **UP**를 direct 연결

\* 아직 login이 안되고 **connect**되어서, session이 구성된상태

\* 리스너는 id/password, 권한 등을 확인하지 않음

\* DB 죽어도 connect는 결개로 가능

🡪 Server Process는 ID/password, 권한등을 확인 from dictionary

(DB가 온전히 연결되어서 살아있는 상태)

! SP가 생성되면, 유저정보를 보관하는 PGA공간 할당

! 서버가 살아있는지 확인: **tnsping** (오라클에서 지원하는 OS명령~ 윈도우의 ping과 같음)

. 리스너와의 연결가능성만 확인

$tnsping

! 리스너 프로세스: 서버단에서 올려야 함

! >sqlplus scott/tiger@k\_ora

🡪 리스너가 양쪽정보 확인

🡪 SP와 UP연결 전용연결: dedicated connection

. resource점용

! 접속방식: Dedicate방식, Shared방식

- 기본: dedicated 방식 –리소스점유 한계

- 추가: shared 방식

! Share server 방식

- 우선 dispatcher 생성(중재자)

- 다음 공유가능한 Server process 여러 개 생성 (Shared Server Process)

- client가 접속시 dedicated와 shared방식 선택

- >sqlplus scott/tiger@shared

dispatcher가 빈 SP를 찾아서 Shared Server Proces와 User Process연결

**SP가 query** **결과를 회송하면 connection 끊음**

- Web서버 WAS단은 shared 방식으로 구성

------------------- ------------------------------

! 서버단 실습작업

/listener.ora수정 (서버입장)

/tnsnames.or수정 (클라이언트입장) – 자신

\* $sqlplus demo/demo – 로칼접속

\* $sqlplus demo/demo@k\_ora – 자신이 클라이언트가 돼서 리스너경유해서 자신의 서버 접속

\* 외부에서는 접속이 안되는데, 자신이 리스너로 접속가능하면 🡪 대부분 방화벽문제

---------------------------------- ------------------

**🡪 0208-1 WS1\_ch06.ORACLE\_Network.pdf 설명**

2page

3page

4page – 네트워크 구성 및 관리도구: 첫번째, 두번째 방식은 안씀 🡪 세번쨰 네번째만 알면됨

5page – 리스너 구성명령어 Listener Control Utility Syntax

$ps –ef | grep tns

root ~~~

oracle ~~~

$lsnrctl

LSNRCTL>start

-- listener parameter file : listener.ora 내용 확인

LSNRCTL>status

LSNRCTL>exit

$ lsnrctl status -- 동일방식