

목차

- [Unix / Linux 구조 및 파일 시스템](#)
 - [OS](#)
 - [Virtual Machine](#)
 - [VIM](#)
 - [Shell Script](#)
- [Oracle](#)
 - [RDBMS의 특징](#)
 - [Oracle Standalone 설치](#)
 - [Oracle Admin](#)
 - [Oracle Architecture](#)
 - [Backup & Recovery](#)
 - [ASM](#)
 - [RAC](#)
 - [Multitenant](#)
- [Other Database](#)
 - [MySQL / MariaDB](#)
 - [PostgreSQL](#)
 - [Tibero](#)
- [Ark Product](#)
 - [Ark for FR](#)
 - [Ark for Oracle](#)
 - [Ark for CDC](#)

Unix / Linux 구조 및 파일 시스템

OS

OS란 Operating System의 줄임말이다.

OS는 한국어로 운영체제라고 한다.

개요

운영체제를 정리하면 하드웨어와 시스템 리소스를 제어하고 프로그램에 대한 일반적 서비스를 지원하는 시스템 소프트웨어이다.

시스템 하드웨어를 관리할 뿐 아니라 응용 소프트웨어를 실행하기 위해 하드웨어 추상화 플랫폼과 공통 시스템 서비스를 제공한다.

운영체제는 입출력과 메모리 할당 같은 하드웨어 기능의 경우 응용 프로그램과 컴퓨터 하드웨어 사이의 중재 역할을 한다.

응용 프로그램 코드는 일반적으로 하드웨어에서 직접 실행된다.

운영체제는 실행되는 응용 프로그램들이 메모리와 CPU, 입출력 장치 등의 자원을 사용할 수 있도록 만들어 준다.

위 자원들을 추상화하여 파일 시스템 등의 서비스를 제공한다.

멀티태스킹을 지원하는 경우 여러 개의 응용 프로그램을 실행하고 있는 동안 운영체제는 모든 프로세스들을 스케줄링하여 동시에 수행되는 것 처럼 보여주는 효과를 한다.

운영체제는 대표적으로 네 가지 역할을 수행한다.

| 역할 | 설명 |
|----------|------------------------------------------------------------------|
| 자원 관리 | 컴퓨터의 한정적인 자원을 효율적으로 관리한다. |
| 보안 | 소프트웨어나 외부 사용자가 데이터를 삭제하거나, 중요한 파일에 접근하는 것을 막거나, Read Only로 보호한다. |
| 인터페이스 제공 | GUI, CLI 등 사용자가 편리하게 커널의 자원을 사용할 수 있도록 한다. |
| 프로세스 관리 | 컴퓨터가 여러 작업을 동시에 효율적으로 처리할 수 있게 한다. |

종류

싱글태스킹 / 멀티태스킹

- 싱글태스킹 운영체제
 - 한번에 오직 하나의 프로그램만 실행할 수 있다.
- 멀티태스킹 운영체제
 - 시분할을 통해 하나 이상의 프로그램이 동시에 실행할 수 있다.
 - 선점형과 비선점형(협동형)이 있다.
 - 선점형 멀티태스킹의 경우 CPU 시간을 쪼개 프로그램들 각각에 슬롯을 할당한다.
 - Solaris, Linux 등의 Unix 계열 운영체제

- 비선점형 멀티태스킹의 경우 정해진 방식에 따라 다른 프로세스들에 시간을 제공하기 위해 각 프로세스에 의존한다.
- 16bit 버전의 Microsoft Windows

단일 사용자 / 다중 사용자

- 단일 사용자 운영체제
 - 사용자 구별이 없으나 여러 프로그램이 나란히 실행하는 것은 가능하다.
- 다중 사용자 운영체제
 - 디스크 공간과 같은 리소스와 프로세스를 식별하는 기능을 갖춘 멀티태스킹의 기본개념을 확장한 것이다.
 - 여러 사용자에게 속해 있으며 여러 사용자가 동시에 시스템과 상호 작용할 수 있게 한다.
 - 시분할 운영체제들은 시스템의 효율적인 이용을 위해 태스크를 스케줄링한다.

분산 운영체제

- 구별된 컴퓨터 그룹을 관리해 하나의 컴퓨터인 것처럼 보이게 만든다.
- 서로 통신하는 네트워크화된 컴퓨터들이 개발되며 분산 컴퓨팅이 활성화되었다.
- 분산되는 연산들은 하나 이상의 컴퓨터에서 수행된다.
- 하나의 그룹에 속하는 컴퓨터들이 협업할 때 분산 시스템을 형성한다.

판형 운영체제

- 배포 형식이나 클라우드 컴퓨팅 환경에서 하나의 가상 머신 이미지를 게스트 운영체제로 만드는 것을 가리킨다.
- 실행 중인 여러 개의 가상 머신을 위한 도구로 이를 저장한다.
 - 가상화와 클라우드 컴퓨팅 관리 둘 다에 사용된다.
 - 대형 서버 웨어하우스 환경에서 흔히 볼 수 있다.

임베디드 운영체제

- 임베디드 컴퓨터 시스템에서 사용할 수 있게 설계된 운영체제이다.
 - 임베디드 컴퓨터 시스템: 내장형 시스템, 기계나 기타 제어가 필요한 시스템에 대해 제어를 위한 특정 기능을 수행하는 컴퓨터 시스템
- 조그마한 기계에 동작하도록 설계되어 있다.
- 제한된 수와 자원으로 동작한다.
 - 매우 크기가 작고 효율적으로 설계되어 있다.

실시간 운영체제

- 특정한 짧은 시간 내 이벤트나 데이터의 처리를 보증하는 운영체제이다.
- 실시간 운영체제는 싱글태스킹일 수도 있고, 멀티태스킹일 수도 있다.

라이브러리 운영체제

- 네트워크 등 일반적인 운영체제가 제공하는 서비스들이 라이브러리 형태로 제공되는 운영체제를 의미한다.

Unix / Linux

Unix

- AT&T 벨 연구소에서 1970년대 C와 함께 탄생했다.
- 교육, 연구 기관에서 주로 사용하는 다중 사용자, 대화식, 시분할처리 시스템용 운영체제이다.
- 많은 OS의 본보기가 되었다.
- 하드웨어와 같이 배포되는게 일반화되었다.
 - 지원되는 하드웨어와의 높은 호환성과 신뢰성을 기대할 수 있다.

배포판 요약

- Minix
 - 교육용으로 개발된 Unix, Linux에 영감을 주었다.
- MacOS
 - 최초 GUI 환경인 Mac Classic의 후속작이며 개인 컴퓨터 시장에서 Unix 중 가장 점유율이 높다.
- BSD
 - 캘리포니아 대학의 CSRG에서 개발한 범용 OS, 과거에는 Linux 보다 높은 안정성으로 인기가 있었다.
- AIX
 - IBM사의 Power 제품군에 올라가는 OS로 주로 대기업과 금융권 등의 대형 서버로 이용된다.
 - System-V 계열 Unix 중 유일하게 지속적으로 개발 중이다.
- HP-UX
 - HP 사에서 개발한 HP Server용 OS이며 2021년부터 업데이트를 하지 않는다.

Linux

- Linux Is Not Unix로 재귀약자로 설명하기도 한다.

기술적인 특징

| 특징 | 설명 |
|------------|-----------------------------------------------------------------------|
| 계층적인 파일 구조 | /를 기준으로 그 하위 디렉터리에 다시 디렉터리가 존재하는 구조이다. 이런 구조를 계층적 파일 구조라 한다.(Tree 구조) |
| 장치의 파일화 | Device를 파일화하여 사용한다. 특정 하드웨어에 명령을 수행하려면 해당 장치 파일에 명령을 내리는 형식이다. |
| 가상 메모리 사용 | Virtual Memory는 Hard Disk의 일부를 메모리처럼 사용하는 것을 말한다. 이런 영역을 SWAP이라고 부른다. |

| 특징 | 설명 |
|-------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 동적 라이브러리 지원 | <p>프로그램에서 특정 기능을 실행하기 위한 명령어인 Routine들을 모아 놓은 것을 Library라 한다.</p> <p>프로그램 개발 시 Library 중 필요한 Routine들을 받아 Link 시킨다.</p> <p>이런 Routine들을 공유하는 것을 Shared Library라 한다.</p> <p>Dynamic Shared Library는 실행 파일 내부에 넣지 않고 프로그램을 실행할 때 가져다 사용하므로 메모리 효율성이 높다.</p> |
| 가상 콘솔 | <p>Virtual Console은 말 그래도 가상 콘솔을 제공하는 기능이다.</p> <p>리눅스는 기본적으로 6개의 가상 콘솔을 제공한다.</p> <p><code>ctrl + alt + F1 ~ F6</code></p> |
| 파이프 | <p>Pipe는 Process의 통신을 위해 도입된 것이다.</p> <p>어떤 Process의 표준 출력이 다른 프로세스의 표준 입력으로 쓰는 것을 말한다.</p> <p><code> </code></p> |
| Redirection | <p>Process의 I/O를 Standard I/O가 아닌 다른 I/O로 변경할 때 사용한다.</p> <p>출력 결과를 파일로 저장하거나 파일의 내용을 Process의 입력으로 사용하는 기법이다.</p> |

배포판 요약

- Redhat 계열
 - Redhat Linux와 커뮤니티 버전인 Fedora에서 파생된 배포판들이다.
 - 패키지 형식은 .rpm이며 패키지 관리자로 yum을 사용한다.
 - CentOS, Oracle Linux 등이 있다.
- Debian 계열
 - 데비안에서 파생된 배포판들이다.
 - 패키지 형식은 .deb이며 패키지 관리자로 apt를 사용한다.
 - Chrome OS, TmaxOS 등이 있다.
- Ubuntu 계열
 - Kubuntu, Vanilla OS 등이 있다.
- Arch 계열
 - 아치 리눅스에서 파생된 배포판들이다.
 - 패키지 관리자는 pacman이며 형식은 특정 확장자 없이 tar.gz나 tar.xz이나 관례적으로 압축 용 확장자 앞에 .pkg가 붙는다.
 - Steam OS 등이 있다.
- SUSE 계열
 - 슬랙웨어 기반으로 시작했으나 현재는 관계가 멀어지고 .rpm을 사용하여 독자적 계열로 취급한다.
 - openSUSE 등이 있다.

기본 구조 및 구성



Hardware <-> Kernel <-> Shell <-> Application <-> User

| 구조 | 설명 |
|-------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Hardware | 컴퓨터의 하드웨어이다. CPU, 모니터, 키보드, RAM, GPU, Sound Card, 메인보드와 같은 물리적 부품을 의미한다. |
| Kernel | OS의 다른 부분이나 Application 수행에 필요한 여러 서비스를 제공한다. 프로그램 실행 과정에서 가장 핵심적인 연산이 이루어지는 부분이다. Core라고도 부른다. Hardware를 직접 제어하고, Process 관리, Memory 관리, File System 관리 등을 수행하는 OS의 핵심이다. Application과 Hardware 사이의 관리자 역할을 수행하며 Shell과 연관되어 실행하는 명령을 수행하고 수행 결과를 Shell로 보내는 역할을 한다. |
| Shell | OS 상 다양한 OS 기능과 서비스를 구현하는 Interface를 제공한다. 사용자의 명령을 해석해 Kernel로 전달하는 프로그램이다. OS의 내부를 감싸는 층이어서 Shell이라고 부른다. |
| Application | OS에서 실행되는 모든 Software |

Group / User

리눅스는 다중 사용자 운영체제이다.

사용자는 별도의 권한을 가지고 있으며 권한에 따라 파일을 읽고 쓰고 실행할 수 있다.

유저를 확인하는 방법은 `/etc/passwd`를 조회하면 된다.

```
cat /etc/passwd

root:x:0:0:root:/root:/bin/bash
...
ark:x:1000:1000:ark:/home/ark:/bin/bash
...
```

`/etc/passwd` 파일의 뜻은 다음과 같다.

| | | | | | | |
|----------|----------|------|------|-------------|----------------|--------------------|
| root | :x | :0 | :0 | :root | :/root | :/bin/bash |
| Username | Password | :UID | :GID | Description | Home Directory | :Shell Information |

위 Password의 x는 비밀번호가 보안되어있다는 뜻이다.

리눅스에는 Group이라는 개념이 있다.

어떤 파일이나 디렉터리를 특정 권한의 사용자들만 사용할 수 있게 하는 데에 사용한다.

그룹을 확인하는 방법은 `/etc/group`을 조회하면 된다.

```
cat /etc/group

root:x:0:
...
ark:x:1000:
...
```

`/etc/group` 파일의 뜻은 다음과 같다.

| | | | |
|-----------|-----------|------|--------------|
| root | :x | :0 | : |
| Groupname | :Password | :GID | :Group Users |

기본적으로 Group의 경우 그냥 생성했을 때 1000번 부터 생성되는 것을 확인할 수 있다.

따로 옵션을 주어 해당하는 그룹을 만들 수도 있다.

```
groupadd -g 54321 oinstall
```

생성한 그룹은 사용자에게 부여할 수 있다.

```
usermod -G oinstall ark

id ark

uid=1000(ark) gid=1000(ark) groups=1000(ark),54321(oinstall)
```

권한

Linux는 하나의 컴퓨터를 여러 사람이 사용할 수 있는 Multi User OS기에 권한 관리가 중요하다. 파일이나 디렉터리의 소유권을 변경하거나 권한을 변경할 수 있다.

파일이나 디렉터리의 소유권과 권한을 확인하는 방법은 `ls -l`을 통해 확인할 수 있다.

```
ls -l
-rw----- 1 oracle dba 115983 Nov  3 12:36 nohup.out
```

첫 10자리는 파일의 종류와 권한들, 이어서 링크 개수, 소유자, 소유 그룹, 용량, 최종 편집 일자, 파일명이다.

첫 10자리 중 2~10번째가 권한에 관한 값이고 소유자와 소유 그룹이 소유권과 관련된 값이다.

파일이나 디렉터리의 소유권을 바꾸는 명령어는 `chown`이다.

```
chown root:root nohup.out      # <user>:<group> 쌍으로 변경, <user>.<group>으로 대체 가능
chown root:root -R /directory  # 하위 경로의 소유권을 모두 변경
```

파일이나 디렉터리의 권한은 Read, Write, eXecute로 표시한다.

단어 그대로 r은 읽기, w는 쓰기, x는 실행하는 권한을 말한다.

권한은 소유자 User, 그룹 Group, 나머지 Other, 모두 All의 앞자를 따 부여 가능하다.

해당 권한을 2진수로 나열하여 권한을 변경할 수 있다.

권한 변경 명령어는 `chmod`이다.

```
chmod u=rwx nohub.out
chmod 700 nohub.out
chmod -R 700 /directory      # 하위 경로의 권한을 모두 변경
```

File System

다양한 UNIX 계열 운영체제가 등장하며 운영체제 간 호환성과 이식성을 높이기 위해 POSIX(Portable Operating System Interface in unix)가 탄생했다.

POSIX는 운영체제 자체가 아닌 응용 프로그램과 운영체제 간 인터페이스를 정의하는 개념이다.

Linux는 POSIX 표준을 만족한다.

파일시스템에는 Ext, Ext2, Ext3, Ext4, XFS 등이 있다.

Virtual Machine

Linux 설치

1. 홈페이지(<https://virtualbox.org/>) 접속한다.
2. VirtualBox를 다운로드한다.
 1. 좌측 Downloads 클릭
 2. OS에 맞게 선택(Windows → Windows hosts)
 3. 다운로드 진행
3. 설치 파일 실행 및 진행

1. Next → Next → Yes → Install → Yes → Finish

4. 정상 설치 시 화면



5. Centos iso 파일 다운로드

1. Centos 다운로드 페이지(<https://www.centos.org/download/>) 접속

2. 원하는 버전의 mirrors 클릭해 원하는 iso 파일 다운로드

6. 새로 만들기를 눌러 가상머신 생성

1. 가상머신 운영 체제 설정시 종류와 버전을 Linux / Red Hat(64-bit)로 설정
(이름을 CentOS 7로 설정 시 자동 설정)

2. 가상머신 메모리 크기 설정(4096MB)

3. 하드 디스크 설정(새 가상 하드 디스크 만들기)

4. 디스크 파일 종류 설정(VDI)

5. 물리적 하드 드라이브 저장 방식 설정(동적 할당)

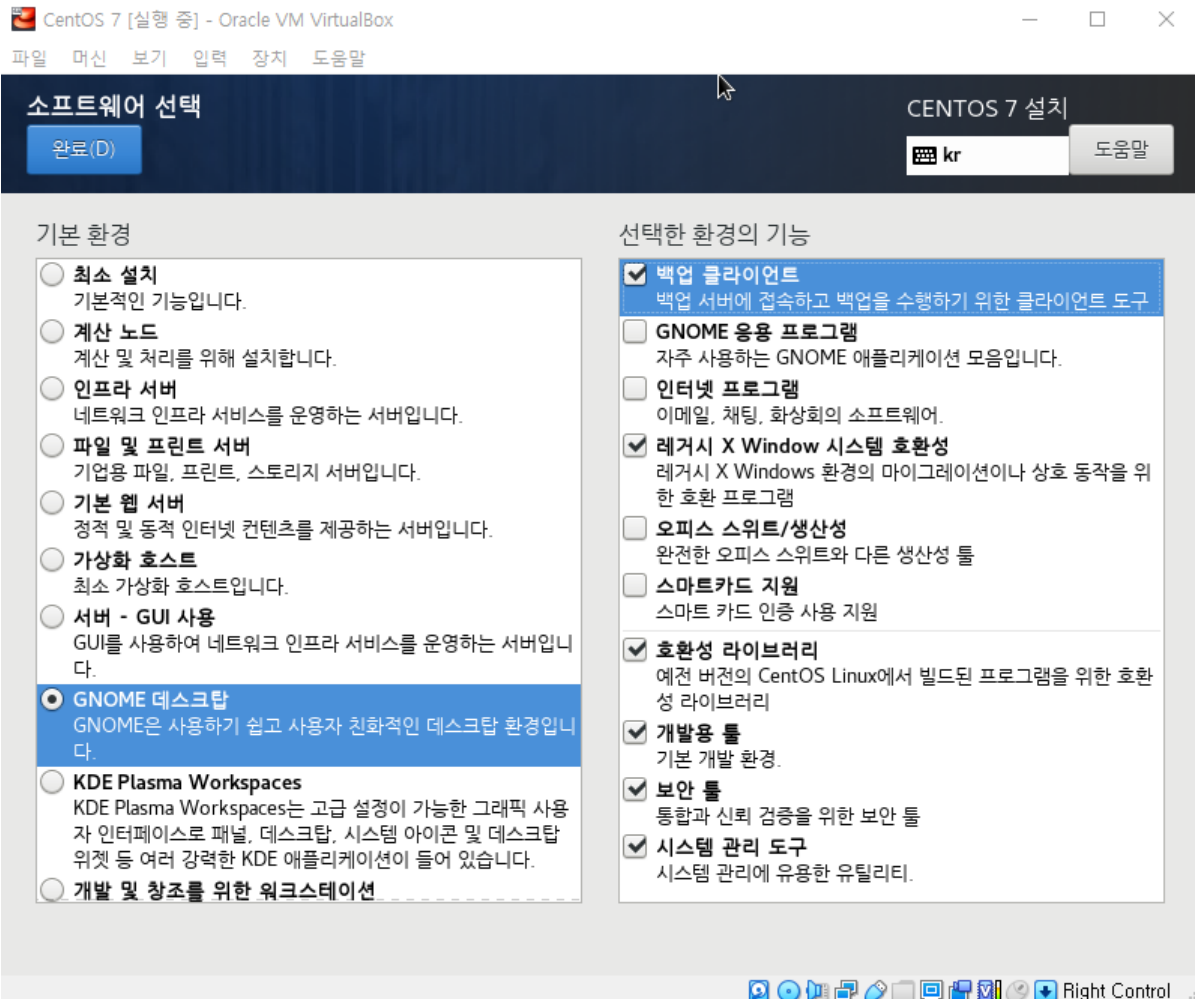
6. 위치 및 크기 설정(60GB)

7. 설정 버튼을 눌러 생성을 위한 설정 진행

1. 저장소 → 컨트롤러 : IDE → 비어 있음 → Choose a disk file... → CentOS iso 파일

2. 네트워크 → 어댑터 1,2 → 네트워크 어댑터 사용하기 → NAT / 호스트 전용 어댑터 → 무작위 모드 모두 허용

8. Install CentOS 7 → 하단 이미지와 같이 소프트웨어 선택



9. 이더넷(enp0s3) 켜기

10. root와 사용자 생성 → 설치 완료 → 재부팅

11. 라이선스 동의 → 설정 완료 → 끝

리눅스 주요 명령어

- **ls**

List Segments: 현재 위치의 파일 목록을 조회한다.

| | |
|-----|------------------------------------------------------------|
| -l | 파일의 상세정보 |
| -a | 숨김 파일 표시 |
| -t | 파일들을 생성시간 역순으로 표시 |
| -rt | 파일들을 생성시간순으로 표시 |
| -f | 파일 표시 시 마지막 유형에 나타내는 파일명을 끝에 표시(/: 디렉터리, *: 실행파일, @: 링크 등) |

- **cd**

Change Directory: 디렉터리를 이동한다.

| [route] | 기입한 루트의 디렉터리로 이동 |
|---------|------------------|
| ~ | 홈 디렉터리로 이동 |
| / | 최상위 디렉터리로 이동 |
| . | 현재 디렉터리 |
| .. | 상위 디렉터리로 이동 |
| - | 이전 경로로 이동 |

- **touch**

0 byte 파일 생성, 파일의 날짜, 시간을 수정한다.

| [file] | file을 생성 |
|--------------------------|----------------------------------|
| -c [file] | file의 시간을 현재시간으로 갱신 |
| -t [YYYYMMDDhhmm] [file] | file의 시간을 YYYYMMDDhhmm으로 갱신 |
| -r [oldfile] [newfile] | newfile의 날짜 정보를 oldfile과 동일하게 변경 |

- **mkdir**

MaKe DIRtory: 디렉터리 생성

| [dir] | dir인 디렉터리 생성 |
|-----------------------|-------------------------------|
| [dir1] [dir2] | dir1, dir2 디렉터리 생성 |
| -p [dir1]/[dir2] | dir1 디렉터리 생성, dir2 하위 디렉터리 생성 |
| -m [Permission] [dir] | Permission을 갖는 dir인 디렉터리 생성 |

- Permission

2진수 기준 자리 수가 1일 때 권한이 있다.

1의 자리 → 실행

2의 자리 → 쓰기

4의 자리 → 읽기

Permission은 8진수 기준 3자리이다.

1의 자리 → 일반 사용자

8의 자리 → 소유 그룹

64의 자리 → 소유자

- **cp**

CoPy: 파일을 복사한다

| [file1] [file2] | file1을 file2라는 이름으로 복사 |
|--------------------|-------------------------------|
| -f [file1] [file2] | 강제 복사(file2가 존재 시 대치) |
| -r [dir1] [dir2] | 디렉터리 복사(폴더 안 모든 하위 경로, 파일 복사) |

- **mv**

MoVe: 파일을 이동한다.

| [file1] [file2] | file1을 file2로 변경 |
|------------------------|-----------------------|
| [file1] /[dir] | file1을 dir로 이동 |
| [file1] [file2] /[dir] | file1, file2를 dir로 이동 |
| /[dir1] /[dir2] | dir1을 dir2로 이름 변경 |

- **rm**

ReMove: 파일을 삭제한다.

| [file] | file을 삭제 |
|-----------|-------------------------------|
| -f [file] | file을 강제삭제 |
| -r dir | dir를 삭제(디렉터리는 -r 옵션 없이 삭제 불가) |

- **cat**

CATenate: 파일의 내용을 화면에 출력
redirection 기호를 사용하여 새 파일 생성

| [file] | file의 내용 출력 |
|-----------------|---------------------|
| [file1] [file2] | file1과 file2의 내용 출력 |
| [file1] [file2] | more |
| [file1] [file2] | head |
| [file1] [file2] | tail |

- redirection

- **>**: 기존에 있는 파일 내용을 지우고 저장한다.
- **>>**: 기존 파일 내용 뒤에 덧붙여서 저장한다.
- **<**: 파일의 데이터를 명령에 입력한다.

ex)

| cat [file1] [file2] > [file3] | file1, file2의 명령 결과를 합쳐 file3에 저장 |
|-------------------------------|-----------------------------------|
| cat [file1] >> [file2] | file2에 file1의 내용 추가 |
| cat < [file] | file의 결과 출력 |
| cat < [file1] > [file2] | file1의 출력 결과를 file2에 저장 |

- **alias**

별칭을 지정할 수 있다.

```
alias [nick] = '[command]'
```

#nick을 실행하면 command가 실행

```
unalias [nick]
```

#nick이라는 alias 해제

```
#ex)
alias lsa = 'ls -a'
unalias lsa
```

- **tail**

파일의 뒷부분을 보여준다.

옵션 없이 사용 시 마지막 10줄 보여준다.

| [file] | file의 마지막 10줄을 보여줌 |
|-------------------|-------------------------------------|
| -f [file] | tail을 종료하지 않고 file의 업데이트 내용을 실시간 출력 |
| -n (라인 수) [file] | file의 마지막 줄부터 지정한 라인 수 까지 출력 |
| -c (바이트 수) [file] | file의 마지막부터 지정한 바이트만큼 출력 |
| -q [file] | file의 헤더와 상단의 파일 이름을 출력하지 않고 내용만 출력 |
| -v [file] | file의 헤더와 이름먼저 출력한 후 내용 출력 |

- **grep**

특정 패턴 찾는 명령어이다.

| -A [N] | 특정 문자열부터 N 이후 라인까지 출력 |
|----------------|------------------------------------------------|
| -B [N] | 특정 문자열부터 N 이전 라인까지 출력 |
| -C [N] | -A [N] -B [N] |
| --color=[when] | 특정 문자열을 특정 색으로 표시([when]: never, always, auto) |
| -d [action] | 특정 디렉터리에서 특정 문자열 검색 |
| -e [pattern] | 여러 특정 문자열로 검색 |
| -i | 특정 문자열을 대소문자 구별 없이 검색 |
| -v | 특정 문자열을 제외한 나머지 행 검색 |
| -w | 다른 문자열이 포함되지 않은 문자열만 검색 |

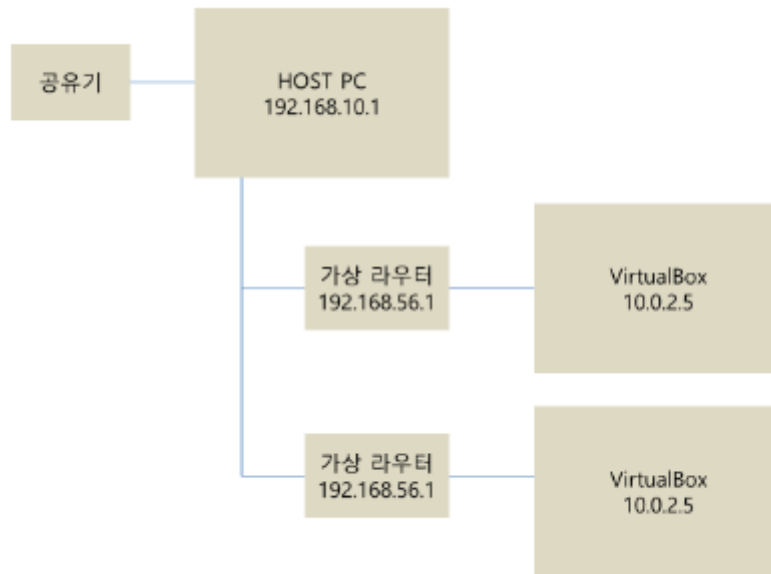
Network

요약

| Type | VM <> VM | VM > Host | VM < HOST | VM > LAN | VM < LAN |
|--------------|-------------|--------------|-----------|-------------|----------|
| Not attached | N | N | N | N | N |

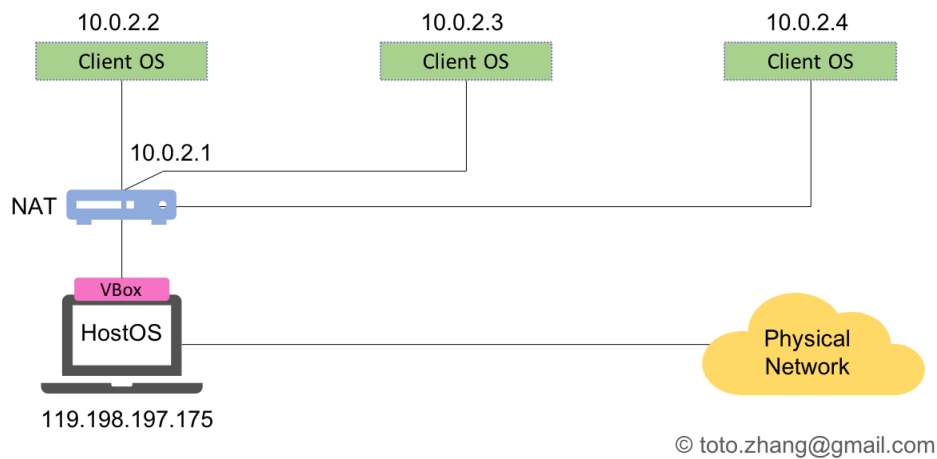
| Type | VM <> VM | VM > Host | VM < HOST | VM > LAN | VM < LAN |
|-------------------|----------|-----------|--------------|----------|--------------|
| NAT | N | Y | Port Forward | Y | Port Forward |
| NAT Network | Y | Y | Port Forward | Y | Port Forward |
| Bridged Adapter | Y | Y | Y | Y | Y |
| Internal Network | Y | N | N | N | N |
| Host-only Network | Y | Y | Y | N | N |

NAT



Host/Guest 설정이 불필요하여 간단하게 외부 네트워크에 접근 가능하다.
가상 라우터 환경을 조성하여 Host OS에서 포트포워딩하는 구조이다.
각 VM 머신은 가상 IP를 할당해 통신한다.
라우터를 공유하지 않으므로 IP가 중복될 수 있다.

NAT Network



NAT Network는 기존 NAT에서 가상 환경을 공유하는 구조이다.

VirtualBox Host-Only 드라이버가 가상의 공유기 역할을 해 가상머신끼리의 통신이 가능하다.

Bridge Adapter

Bridge Adapter 방식은 Host PC와 동일한 네트워크를 사용한다.

Host와 동일한 대역의 IP를 할당해야 한다.

MAC 제한이 걸린 환경의 경우 IP가 할당되지 않을 수 있다.

OS X와 Linux에서는 무선 네트워크에 대해 사용이 제한될 수 있다.

Internal Network

Internal Network는 Host PC의 영향 없이 가상의 내부망 환경을 만들 때 사용한다,

가상의 스위치를 구성한다.

Host PC, 외부 네트워크 연결이 차단되어 가상머신끼리의 통신만 가능하다.

보안과 테스트 환경 구성에 이점이 있다.

Host-Only Network

Host-Only Network는 Internal Networking과 기능면에서 동일하나 내부망에서 Host PC에 IP를 가상으로 한번 더 할당해 가상머신과 통신 가능하다.

외부 네트워크는 Host PC만 접근 가능하며 가상머신에서는 외부 네트워크가 차단된다.

Generic Driver

UDP Tunnel과 VDE(Virtaul Distributed Ethernet) Networking을 지원한다.

잘 사용하지 않는다.

- UDP Tunnel: 서로 다른 Host에서 실행되는 가상머신을 기존 네트워크 인프라를 사용해 통신 가능하다.
- VDE: Linux Host의 가상 스위치에 연결 가능.

SSH

Secure Shell의 약자이다.

원격으로 로그인하여 그 안에 있는 명령들을 실행할 수 있는 프로그램이다.

SSH는 암호화 기법을 사용하기에 안전하게 통신할 수 있다.

SSH 설치

```
yum install -y openssh-server openssh-clients openssh-askpass
```

SSH 설정

서버 관련 주 설정 파일은 `/etc/ssh/sshd_config`에 존재한다.

```
# $OpenBSD: sshd_config,v 1.103 2018/04/09 20:41:22 tj Exp $

# This is the sshd server system-wide configuration file. See
# sshd_config(5) for more information.

# This sshd was compiled with
PATH=/usr/local/bin:/usr/bin:/usr/local/sbin:/usr/sbin

# The strategy used for options in the default sshd_config shipped with
# OpenSSH is to specify options with their default value where
# possible, but leave them commented. Uncommented options override the
# default value.

# If you want to change the port on a SELinux system, you have to tell
# SELinux about this change.
# semanage port -a -t ssh_port_t -p tcp #PORTNUMBER
#
#Port 22
#AddressFamily any
#ListenAddress 0.0.0.0
#ListenAddress ::
...
```

TCP 통신

TCP는 응용 프로그램이 데이터를 교환할 수 있는 네트워크 대화를 설정하고 유지하는 방법을 정의하는 표준이다.

TCP는 IP Network를 통해 통신하는 Host에 실행하는 애플리케이션 간에 신뢰할 수 있고, 순서가 정해졌으며, 오류를 체크하고 전송할 수 있다.

SCP

Secure CoPy의 약자이다.

SSH Protocol 기반 파일 전송 수단이다.

```
scp <OPTION> [ID]@[IP]:[PATH] [FILE]
scp <OPTION> [FILE] [ID]@[IP]:[PATH]
```


X Window

Unix 계열 운영체제에서 사용되는 윈도 시스템과 X Window GUI 환경을 뜻한다.
X11 Window의 Forwarding을 활성화 하여 원격으로 GUI를 활용할 수 있다.

Disk Mount

Linux에서의 Mount는 하드디스크의 파티션, CD/DVD, USB Memory 등을 사용하기 위해 특정 위치에 연결을 해주는 것을 말한다.
쉽게 요약해 물리적 장치를 특정 위치(디렉터리)에 연결시켜주는 과정이다.

LVM

Logical Volumn Manager의 약자이다.
여러 디스크들을 하나의 디스크처럼 사용할 수 있게 해준다.

VI(VIM)

Unix 환경에서 가장 많이 쓰이는 문서 편집기이다.
한 화면을 편집하는 Vsual editor라는 뜻에서 유래했다.
모드는 총 3가지가 있다.

명령 모드(Command Mode)

처음 VI를 사용하면 진행중인 모드이다.
방향키나 **h**, **j**, **k**, **l** 을 이용해 이동할 수 있다.
입력 모드에서 **esc** 를 눌러 명령 모드로 돌아올 수 있다.
명령어 사용이 가능하다.
명령어의 경우 **enter** 를 누르지 않아도 실행된다.

| 명령어 | 동작 |
|-------|---------------------------------|
| i | 현재 커서 위치에 삽입(입력 모드로 전환) |
| a | 현재 커서 바로 다음 위치에 삽입(입력 모드로 전환) |
| o | 현재 줄 다음 위치에 삽입(입력 모드로 전환) |
| (N)x | 커서 위치의 글자 N개 삭제(기입 없으면 1개) |
| dw | 커서 위치의 단어 삭제 |
| (N)dd | 커서 위치의 N개의 줄 잘라내기(기입 없으면 1개) |
| u | 이전 명령 취소 |
| (N)yy | 커서 위치의 N개의 줄을 버퍼로 복사(기입 없으면 1개) |
| p | 현재 커서가 있는 줄 바로 아래에 버퍼 내용 붙여넣기 |
| k | 커서가 한 줄 위로 올라감 |
| j | 커서가 한줄 아래로 내려감 |
| l | 커서가 한칸 우측으로 감 |

| 명령어 | 동작 |
|------|-------------------|
| h | 커서가 한칸 좌측으로 감 |
| 0 | 커서가 있는 줄의 맨 앞으로 감 |
| \$ | 커서가 있는 줄의 맨 뒤로 감 |
| (| 현재 문장의 처음 |
|) | 현재 문장의 끝 |
| { | 현재 문단의 처음 |
| } | 현재 문단의 끝 |
| [N]- | N개의 줄만큼 위로 이동 |
| [N]+ | N개의 줄만큼 아래로 이동 |
| G | 파일의 끝으로 이동 |
| r | 한 문자 변경 |
| cc | 커서가 있는 줄의 내용 변경 |

입력 모드(Insert Mode)

명령 모드에서 **i** 나 **a** 명령을 통해 입력 모드로 넘어갈 수 있다.

마지막 행 모드(Last Line Mode)

명령 모드에서 **:**을 입력해 마지막 행 모드로 넘어갈 수 있다.

enter를 입력해야 명령이 들어간다.

| 명령어 | 동작 |
|---------------|-------------------------------------|
| w [file name] | 기입한 파일명으로 파일 저장(기입 없을시 현재 파일명으로 저장) |
| q | vi 종료 |
| q! | vi 강제 종료 |
| wq | 저장 후 종료 |
| wq! | 강제 저장 후 종료 |
| f [file name] | 기입한 파일명으로 파일명 변경 |
| [N] | N번 줄로 이동 |
| \$ | 파일의 맨 끝 줄로 이동 |
| e! | 마지막 저장 이후 모든 편집 취소 |
| /[String] | 현재 커서 위치부터 앞쪽으로 기입한 문자열 검색 |
| ?[String] | 현재 커서 위치부터 뒤쪽으로 기입한 문자열 검색 |

| 명령어 | 동작 |
|----------|-------------|
| set nu | vi 라인 번호 출력 |
| set nonu | vi 라인 출력 취소 |

Shell Script

Shell Script를 사용하면 Unix 커맨드들을 나열해 실행할 수 있다.
다른 스크립트 언어와 마찬가지로 제어문과 반복문 사용이 가능하다.

Bash 셸 기준 예시이다.

```
#                                # 주석
var="Variable"                  # 변수
echo "$var은 변수입니다."      # 변수를 사용하기 위해 변수명 앞 $를 기입한다.
readonly var                    # 변경 불가한 읽기 전용 변수를 만들 수 있다.
echo $0                         # 스크립트 명이다.
echo $1                         # $n = n번째 파라미터이다.
echo $#                         # 파라미터의 개수이다.
echo $*                         # 모든 파라미터를 하나로 처리한다.
echo @$                         # 모든 파라미터를 각각 처리한다.
echo $?                         # 직전 실행한 커맨드의 종료 값이다. 0은 성공이며 1은 실패이다.
echo $$                         # 이 셸 스크립트의 Process ID이다.
echo $!                         # 마지막으로 실행한 Backgroud Process ID이다.

ARRAY=(1 2 3 4)                # 배열
echo ${ARRAY[0]}                # 배열의 아이템 하나에 액세스하는 방법이다.
echo ${ARRAY[*]}                # 배열의 모든 아이템에 액세스하는 방법이다.

var1="20"
var2="10"
echo `expr 20 + 10`             # 덧셈
echo `expr 20 - 10`             # 뺄셈
echo `expr 20 \* 10`             # 곱셈
echo `expr 20 / 10`             # 나눗셈

if [ $var1 == $var2 ]           # 조건문
then
    echo "equal"
elif [ $var1 != $var2 ]
then
    echo "not equal"
fi

case "$var1" in                  # Switch문
    10)
        echo "10!"
        ;;
    20)
        echo "20!"
        ;;
    *)
        echo "other!"
        ;;
```

```
esac

while [ $var2 -lt $var1 ]    # 반복문
do
    echo $var2
    var2=`expr $var2 + 1`
done

for var in 0..4             # 반복문
do
    echo $var
done

MyFunction() {
    echo "This is Function."
}

MyFunction
```

위 예시들을 이용해서 원하는 대로 스크립트를 작성해 보다 쉽게 명령어들을 사용할 수 있다.

Oracle

RDBMS의 특징

DBMS

DataBase Management System의 약자이다.
DB를 관리하는 시스템을 말한다.

사용자와 DB 사이에서 사용자의 요구에 따라 데이터를 생성, 삭제, 변경해주고 DB를 관리해주는 소프트웨어이다.

RDB

Relational DataBase의 약자이다.
관계형 모델에 기초를 둔 DB이다.
모든 데이터를 2차원 테이블 형태로 표현한다.

RDBMS

Relational DataBase Management System의 약자이다.
RDBMS는 관계형 모델을 기반으로 하는 DBMS 유형이다.

RDBMS의 테이블은 서로 연관되어 있어 일반 DBMS보다 효율적으로 데이터를 저장, 구성, 관리할 수 있다.

정규화를 통해 데이터의 중복성을 최소화 할 수 있다.

데이터의 원자성, 일관성, 독립성, 내구성을 유지하며 데이터 무결성을 높인다.

MSSQL, PostgreSQL, Tiberio, MySQL, Oracle등이 예시이다.

Transaction

DB의 상태를 변화시키기 위해 수행하는 작업의 단위이다.

SQL문을 이용해 DB에 접근할 경우 DB의 상태가 바뀔 수 있다.

작업의 단위는 SQL문 하나가 아닌 Transaction의 시작과 종료로 Commit이라고 생각하는 것이 편하다.

트랜잭션은 네 가지 특징이 있다.

| 특징 | 설명 |
|------------------|-----------------------------------------------------------------------------------------------------------------|
| Atomicity(원자성) | Transaction이 DB에 모두 반영되거나 전혀 반영되지 않는다. Transaction이 수행되다가 중간에 종료되는 경우 이전 내용들을 Rollback한다. |
| Consistency(일관성) | Transaction의 작업 처리 결과가 항상 일관성이 있어야 한다. Transaction이 진행되는 동안에 DB가 변경되어도 처음 Transaction을 진행하기 위해 참조한 DB로 진행된다. |
| Isolation(독립성) | 둘 이상의 Transaction이 동시에 실행되는 경우 하나의 Transaction이 나머지 Transaction의 연상에 끼어들 수 없다. |
| Durability(지속성) | Transaction이 성공적으로 완료되었을 경우 결과가 영구적으로 DB에 반영이 된다. |

SQL

구조적 쿼링 언어(Structured Query Language)의 약어이다.

RDB에 정보를 저장하고 처리하기 위한 프로그래밍 언어이다.

SQL문(SQL Statement)를 사용하여 DB에서 정보를 삽입, 변경, 제거, 검색할 수 있다.

SQL의 종류

SQL의 종류로는 DDL, DML, DCL, DQL, TCL 등이 있다.

주로 분류를 할 때는 DDL, DML, DCL로만 분류한다.

Data Definition Language(데이터 정의어)

- CREATE
- ALTER
- DROP
- TRUNCATE

Data Manipulation Language(데이터 조작어)

- SELECT
- INSERT
- UPDATE
- DELETE

Data Control Language(데이터 제어어)

- GRANT
- REVOKE
- COMMIT
- ROLLBACK

Data Query Language(데이터 쿼리어)

정해진 스키마 내에서 쿼리할 수 있는 언어로 DML의 일부분으로 취급하기도 한다.

- SELECT
- WHERE
- DISTINCT
- GROUP BY
- ORDER BY

Transaction Control Language(트랜잭션 제어어)

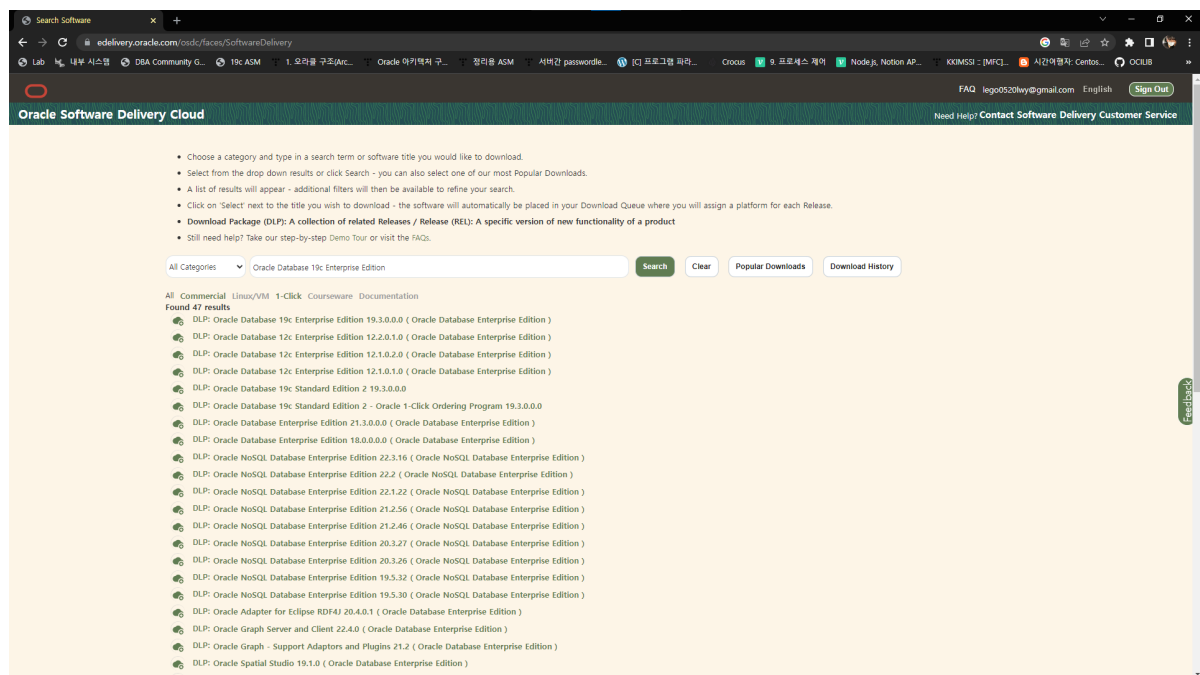
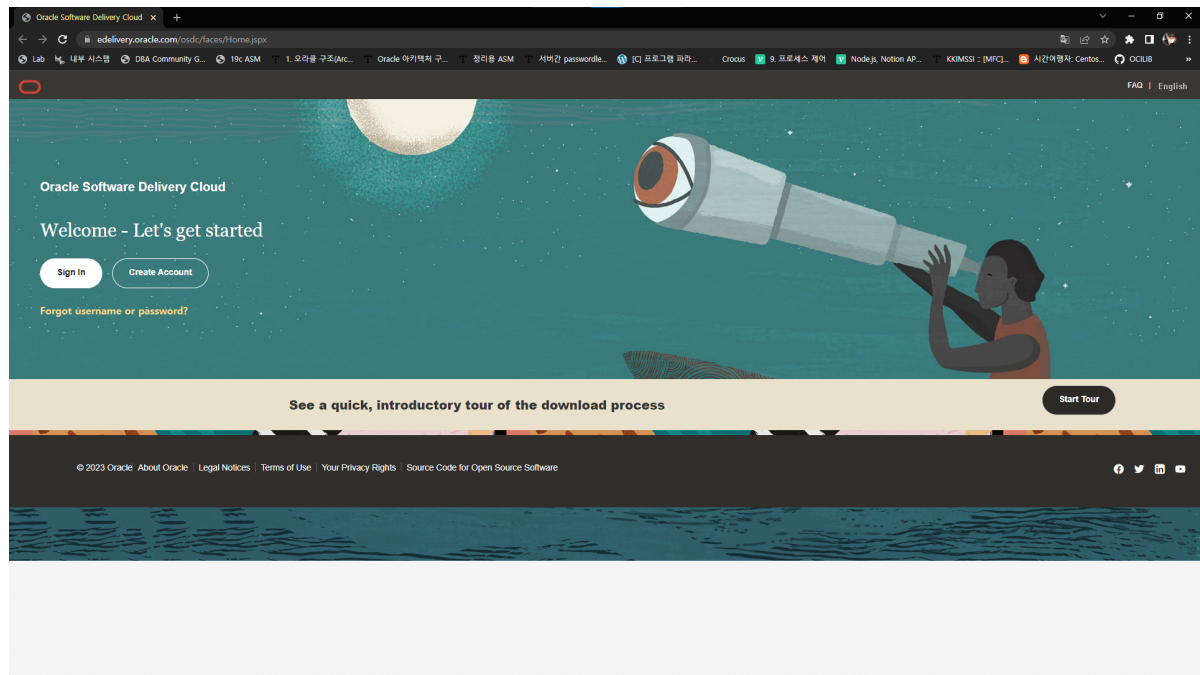
DML을 거친 데이터의 변경사항을 수정할 수 있다.

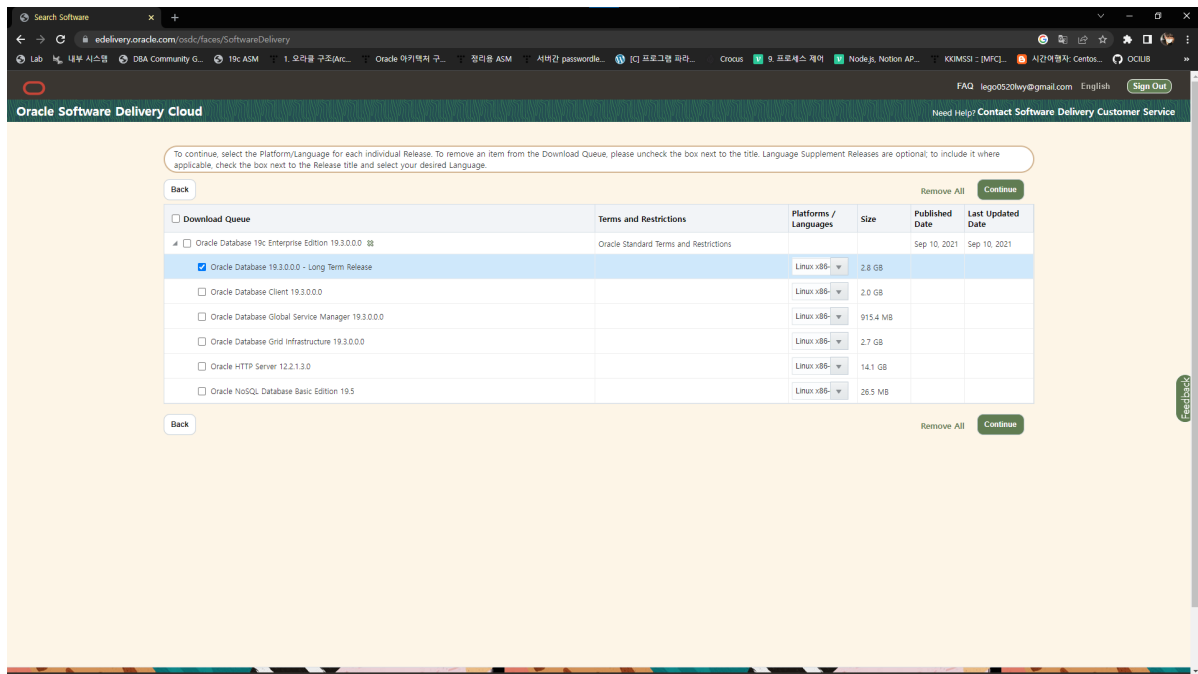
- COMMIT
- ROLLBACK

Oracle Standalone 설치

Download

Oracle Software Delivery Cloud에 가서 원하는 설치 파일을 다운로드 받는다.





User, Group 생성

```
groupadd dba          # dba group 생성
useradd -g dba oracle # dba group에 oracle user 생성
passwd oracle          # oracle의 비밀번호 설정
```

Oracle User Environment Variable 설정

```
vi ~oracle/.bash_profile
```

```
export TMP=/tmp
export TMPDIR=$TMP
export ORACLE_BASE=<oracle_base> # ex) /app/oracle
export ORACLE_HOME=<oracle_home> # ex) $ORACLE_BASE/product/19.3/db_home
export ORACLE_SID=<oracle_sid>   # ex) orcl
export NLS_LANG=<nls_lang>       # ex) AMERICAN_AMERICA.AL32UTF8
export TNS_ADMIN=<tns_admin>     # ex) $ORACLE_HOME/network/admin
export ORACLE_HOSTNAME=<hostname> # ex) localhost
export LD_LIBRARY_PATH=<path>    # ex) $ORACLE_HOME/lib:$LD_LIBRARY_PATH
export PATH=<path>              # ex) $ORACLE_HOME/bin:$PATH
```

Oracle 설치 Directory 생성

```
source ~oracle/.bash_profile
mkdir -p $ORACLE_HOME
chown -R oracle:dba /app
```


의존 Library 설치

```
yum -y install compat-libstdc++-33.x86_64 binutils elfutils-libelf elfutils-  
libelf-devel  
yum -y install glibc glibc-common glibc-devel glibc-headers gcc gcc-c++ libaio  
libaio-devel  
yum -y install libgcc libstdc++ libstdc++-devel make sysstat unixODBC unixODBC-  
devel  
yum -y install unzip wget ksh  
yum -y install  
https://yum.oracle.com/repo/OracleLinux/OL7/latest/x86\_64/getPackage/oracle-  
database-preinstall-19c-1.0-3.el7.x86\_64.rpm
```

Download한 File Unzip

```
su - oracle
```

```
mv <oracle_zip> $ORACLE_HOME  
cd $ORACLE_HOME  
unzip <oracle_zip>
```

Run Installer

```
./runInstaller
```

Oracle Database 19c Installer - Step 1 of 9@localhost.localdomain

Select Configuration Option

19^c ORACLE Database

Select any of the following install options.

☒ Create and configure a single instance database.
This option creates a starter database.

☐ Set Up Software Only

Note 1: For RAC install, do 'Set Up Software Only' and then execute DBCA (Database Configuration Assistant) from the oracle home.
Note 2: To upgrade an Oracle Database, do 'Set Up Software Only' and then execute DBUA (Database Upgrade Assistant) from the oracle home.

Configuration Option
Database Installation Options
Install Type
Typical Installation
Root script execution
Prerequisite Checks
Summary
Install Product
Finish

Help < Back Next > Install Cancel

Oracle Database 19c Installer - Step 2 of 8@localhost.localdomain

Select System Class

19^c ORACLE Database

☐ Desktop class
Choose this option if you are installing on a laptop or desktop class system. This option includes a starter database and allows minimal configuration.

☒ Server class
Choose this option if you are installing on a server class system, which Oracle defines as a system used in a production data center. This option allows for more advanced configuration options.

Configuration Option
System Class
Typical Installation
Root script execution
Prerequisite Checks
Summary
Install Product
Finish

Help < Back Next > Install Cancel

Oracle Database 19c Installer - Step 3 of 18@localhost.localdomain

Select Database Edition

19^c ORACLE Database

Which database edition do you want to install?

☒ Enterprise Edition
Oracle Database 19c Enterprise Edition is a self-managing database that has the scalability, performance, high availability, and security features required to run the most demanding, mission-critical applications.

☐ Standard Edition 2
Oracle Database 19c Standard Edition 2 is a full-featured data management solution ideally suited to the needs of medium-sized businesses.

Configuration Option
System Class
Database Edition
Installation Location
Create Inventory
Configuration Type
Database Identifiers
Configuration Options
Database Storage
Management Options
Recovery Options
Schema Passwords
Operating System Groups
Root script execution
Prerequisite Checks
Summary
Install Product
Finish

Help < Back Next > Install Cancel

Oracle Database 19c Installer - Step 4 of 18@localhost.localdomain

Specify Installation Location

19c ORACLE Database

Configuration Option

System Class

Database Edition

Installation Location

Create Inventory

Configuration Type

Database Identifiers

Configuration Options

Database Storage

Management Options

Recovery Options

Schema Passwords

Operating System Groups

Root script execution

Prerequisite Checks

Summary

Install Product

Finish

Specify a path to place all Oracle software and configuration-related files installed by this installation owner. This location is the Oracle base directory for the installation owner.

Oracle base:

This software directory is the Oracle Database home directory.

Software location: `/app/oracle/product/19.3.0.0.0/dbhome_1`

Help

< Back

Next >

Install

Cancel

Oracle Database 19c Installer - Step 5 of 18@localhost.localdomain

Create Inventory

19c ORACLE Database

Configuration Option

System Class

Database Edition

Installation Location

Create Inventory

Configuration Type

Database Identifiers

Configuration Options

Database Storage

Management Options

Recovery Options

Schema Passwords

Operating System Groups

Root script execution

Prerequisite Checks

Summary

Install Product

Finish

You are starting your first installation on this host. Specify a directory for installation metadata files (for example, install log files). This directory is called the "inventory directory". The installer automatically sets up subdirectories for each product to contain inventory data. The subdirectory for each product typically requires 150 kilobytes of disk space.

Inventory Directory:

Specify an operating system group whose members have write permission to the inventory directory (orainventory).

orainventory_Group Name:

Help

< Back

Next >

Install

Cancel

Oracle Database 19c Installer - Step 6 of 18@localhost.localdomain

Select Configuration Type

19c ORACLE Database

Configuration Option

System Class

Database Edition

Installation Location

Create Inventory

Configuration Type

Database Identifiers

Configuration Options

Database Storage

Management Options

Recovery Options

Schema Passwords

Operating System Groups

Root script execution

Prerequisite Checks

Summary

Install Product

Finish

Select the type of database that you want to create.

☒ General Purpose / Transaction Processing

A starter database designed for general purpose use, or for transaction-heavy applications.

☐ Data Warehousing

A starter database optimized for data warehousing applications.

Help

< Back

Next >

Install

Cancel

Oracle Database 19c Installer - Step 7 of 18@localhost.localdomain

Specify Database Identifiers

19c ORACLE Database

Configuration Option

System Class

Database Edition

Installation Location

Create Inventory

Configuration Type

Database Identifiers

Configuration Options

Database Storage

Management Options

Recovery Options

Schema Passwords

Operating System Groups

Root script execution

Prerequisite Checks

Summary

Install Product

Finish

Provide the identifier information required to access the database uniquely. An Oracle database is uniquely identified by a Global database name, typically of the form "name.domain". A database is referenced by at least one Oracle instance which is uniquely identified from any other instance on this computer by an Oracle system identifier (SID).

Global database name:

Oracle system identifier (SID):

☐ Create as Container database

Creates a database container for consolidating multiple databases into a single database and enables database virtualization.

Pluggable database name:

Help

< Back

Next >

Install

Cancel

Oracle Database 19c Installer - Step 8 of 18@localhost.localdomain

Specify Configuration Options

19c ORACLE Database

Configuration Option

System Class

Database Edition

Installation Location

Create Inventory

Configuration Type

Database Identifiers

Configuration Options

Database Storage

Management Options

Recovery Options

Schema Passwords

Operating System Groups

Root script execution

Prerequisite Checks

Summary

Install Product

Finish

Memory

Character sets

Sample schemas

Enabling Automatic Memory Management allows the database to distribute memory automatically between the system global area (SGA) and the program global area (PGA), based on user-specified overall database memory target size. If automatic memory management is not enabled, then the SGA and PGA must be sized manually.

☐ Enable Automatic Memory Management

Allocate memory: 40 %

SGA target: 1797 MB

PGA aggregate target: 600 MB

Target database memory: 2397 MB

Help

< Back

Next >

Install

Cancel

Oracle Database 19c Installer - Step 8 of 18@localhost.localdomain

Specify Configuration Options

19c ORACLE Database

Configuration Option

System Class

Database Edition

Installation Location

Create Inventory

Configuration Type

Database Identifiers

Configuration Options

Database Storage

Management Options

Recovery Options

Schema Passwords

Operating System Groups

Root script execution

Prerequisite Checks

Summary

Install Product

Finish

Memory

Character sets

Sample schemas

The database character set determines how character data is stored in the database.

☒ Use Unicode (AL32UTF8)

Setting character set to Unicode (AL32UTF8) enables you to store multiple language groups.

☐ Use OS character set (WE8MSWIN1252)

Character set is based on the language setting of this operating system: WE8MSWIN1252.

☐ Choose from the following list of character sets

Select Database character set:

Help

< Back

Next >

Install

Cancel

Oracle Database 19c Installer - Step 8 of 18@localhost.localdomain

19c ORACLE Database

Specify Configuration Options

[Configuration Option](#)
[System Class](#)
[Database Edition](#)
[Installation Location](#)
[Create Inventory](#)
[Configuration Type](#)
[Database Identifiers](#)
Configuration Options
[Database Storage](#)
Management Options
Recovery Options
Schema Passwords
Operating System Groups
Root script execution
Prerequisite Checks
Summary
Install Product
Finish

MemoryCharacter setsSample schemas

You can choose to install sample schema in the starter database. If you choose this option, Human Resources schema will be installed in the starter database, in the SYSAUX tablespace.

☐ Install sample schemas in the database

Help

< BackNext >InstallCancel

Oracle Database 19c Installer - Step 9 of 18@localhost.localdomain

19c ORACLE Database

Specify Database Storage Options

[Configuration Option](#)
[System Class](#)
[Database Edition](#)
[Installation Location](#)
[Create Inventory](#)
[Configuration Type](#)
[Database Identifiers](#)
[Configuration Options](#)
Database Storage
[Management Options](#)
Recovery Options
Schema Passwords
Operating System Groups
Root script execution
Prerequisite Checks
Summary
Install Product
Finish

☒ File system

Use a file system for database storage. For optimal database organization and performance, Oracle recommends that you install data files and the Oracle database software on different disks.

Specify database file location: [Browse...](#)

☐ Oracle Automatic Storage Management

Oracle Automatic Storage Management (Oracle ASM) simplifies database storage administration, and places database files for optimal I/O performance. Select this option if you intend to use Oracle ASM.

Help

< BackNext >InstallCancel

Oracle Database 19c Installer - Step 10 of 18@localhost.localdomain

19c ORACLE Database

Specify Management Options

[Configuration Option](#)
[System Class](#)
[Database Edition](#)
[Installation Location](#)
[Create Inventory](#)
[Configuration Type](#)
[Database Identifiers](#)
[Configuration Options](#)
[Database Storage](#)
Management Options
[Recovery Options](#)
Schema Passwords
Operating System Groups
Root script execution
Prerequisite Checks
Summary
Install Product
Finish

Oracle Database 12c is managed by Oracle Database Express by default. You can use Oracle Enterprise Manager 12c Cloud Control to manage each Oracle Database 19c centrally. Specify the details of the Cloud Control configuration to manage your database.

☐ Register with Enterprise Manager (EM) Cloud Control

OMS Host:

OMS Port:

EM Admin User Name:

EM Admin Password:

Help

< BackNext >InstallCancel

Oracle Database 19c Installer - Step 11 of 18@localhost.localdomain

19c ORACLE Database

Specify Recovery Options

[Configuration Option](#)
[System Class](#)
[Database Edition](#)
[Installation Location](#)
[Create Inventory](#)
[Configuration Type](#)
[Database Identifiers](#)
[Configuration Options](#)
[Database Storage](#)
[Management Options](#)
Recovery Options
[Schema Passwords](#)
Operating System Groups
Root script execution
Prerequisite Checks
Summary
Install Product
Finish

Enable or disable recovery for your database. If you choose to enable recovery, then the location specified will be used as recovery area storage.

☒ **Enable Recovery**

☐ File system

Recovery area location:

☐ Oracle Automatic Storage Management

Help

< Back **Next >** Install Cancel

Oracle Database 19c Installer - Step 12 of 18@localhost.localdomain

19c ORACLE Database

Specify Schema Passwords

[Configuration Option](#)
[System Class](#)
[Database Edition](#)
[Installation Location](#)
[Create Inventory](#)
[Configuration Type](#)
[Database Identifiers](#)
[Configuration Options](#)
[Database Storage](#)
[Management Options](#)
[Recovery Options](#)
Schema Passwords
[Operating System Groups](#)
Root script execution
Prerequisite Checks
Summary
Install Product
Finish

The starter database contains pre-loaded schemas, most of which have passwords that are expired and locked at the end of installation. After installation is complete, you must unlock and set new passwords for those accounts you want to use. Schemas used for database management and postinstallation functions are left unlocked, and passwords for these accounts will not expire. Specify the passwords for these accounts.

☐ Use different passwords for these accounts

| | Password | Confirm password |
|--------|----------------------|----------------------|
| SYS | <input type="text"/> | <input type="text"/> |
| SYSTEM | <input type="text"/> | <input type="text"/> |

☒ **Use the same password for all accounts**

Password: Confirm password:

Messages:

Password [INS-30011] The ADMIN password entered does not conform to the Oracle recommended standards.

Help

< Back **Next >** Install Cancel

Oracle Database 19c Installer - Step 13 of 18@localhost.localdomain

19c ORACLE Database

Privileged Operating System groups

[Configuration Option](#)
[System Class](#)
[Database Edition](#)
[Installation Location](#)
[Create Inventory](#)
[Configuration Type](#)
[Database Identifiers](#)
[Configuration Options](#)
[Database Storage](#)
[Management Options](#)
[Recovery Options](#)
[Schema Passwords](#)
Operating System Groups
[Root script execution](#)
Prerequisite Checks
Summary
Install Product
Finish

SYS privileges are required to create a database using operating system (OS) authentication. Membership in OS Groups grants the corresponding SYS privilege, eg. membership in OSDBA grants the SYSDBA privilege.

Database Administrator (OSDBA) group:

Database Operator (OSOPER) group (Optional):

Database Backup and Recovery (OSBACKUPDBA) group:

Data Guard administrative (OSDGDBA) group:

Encryption Key Management administrative (OSKMDBA) group:

Real Application Cluster administrative (OSRACDBA) group:

Help

< Back **Next >** Install Cancel

Oracle Database 19c Installer - Step 14 of 18@localhost.localdomain

Root script execution configuration

19c ORACLE Database

[Configuration Option](#)
[System Class](#)
[Database Edition](#)
[Installation Location](#)
[Create Inventory](#)
[Configuration Type](#)
[Database Identifiers](#)
[Configuration Options](#)
[Database Storage](#)
[Management Options](#)
[Recovery Options](#)
[Schema Passwords](#)
[Operating System Groups](#)
Root script execution
[Prerequisite Checks](#)
Summary
Install Product
Finish

During the software configuration, certain operations have to be performed as "root" user. You can choose to have the installer perform these operations automatically by specifying inputs for one of the options below. The input specified will also be used by the installer to perform additional prerequisite checks.

☒ Automatically run configuration scripts

☒ Use "root" user credential

Password :

☐ Use sudo

Program path : [Browse...](#)

User name :

Password :

[Help](#) < Back **Next >** Install Cancel

Oracle Database 19c Installer - Step 15 of 18@localhost.localdomain

Perform Prerequisite Checks

19c ORACLE Database

[Configuration Option](#)
[System Class](#)
[Database Edition](#)
[Installation Location](#)
[Create Inventory](#)
[Configuration Type](#)
[Database Identifiers](#)
[Configuration Options](#)
[Database Storage](#)
[Management Options](#)
[Recovery Options](#)
[Schema Passwords](#)
[Operating System Groups](#)
[Root script execution](#)
Prerequisite Checks
[Summary](#)
Install Product
Finish

Verification Result

Some of the minimum requirements for installation are not completed. Review and fix the issues listed in the following table, and recheck the system.

[Check Again](#) [Fix & Check Again](#) [Show Failed](#) ☒ [Ignore All](#)

| Checks | Status | Fixable |
|-----------|---------|---------|
| Swap Size | Ignored | No |

This is a prerequisite condition to test whether sufficient total swap space is available on the system. [more details](#)

Expected Value : 5.8538GB (6138180.0KB)
Actual Value : 3.875GB (4063228.0KB)

[Help](#) < Back **Next >** Install Cancel

Oracle Database 19c Installer - Step 16 of 18@localhost.localdomain

Summary

19c ORACLE Database

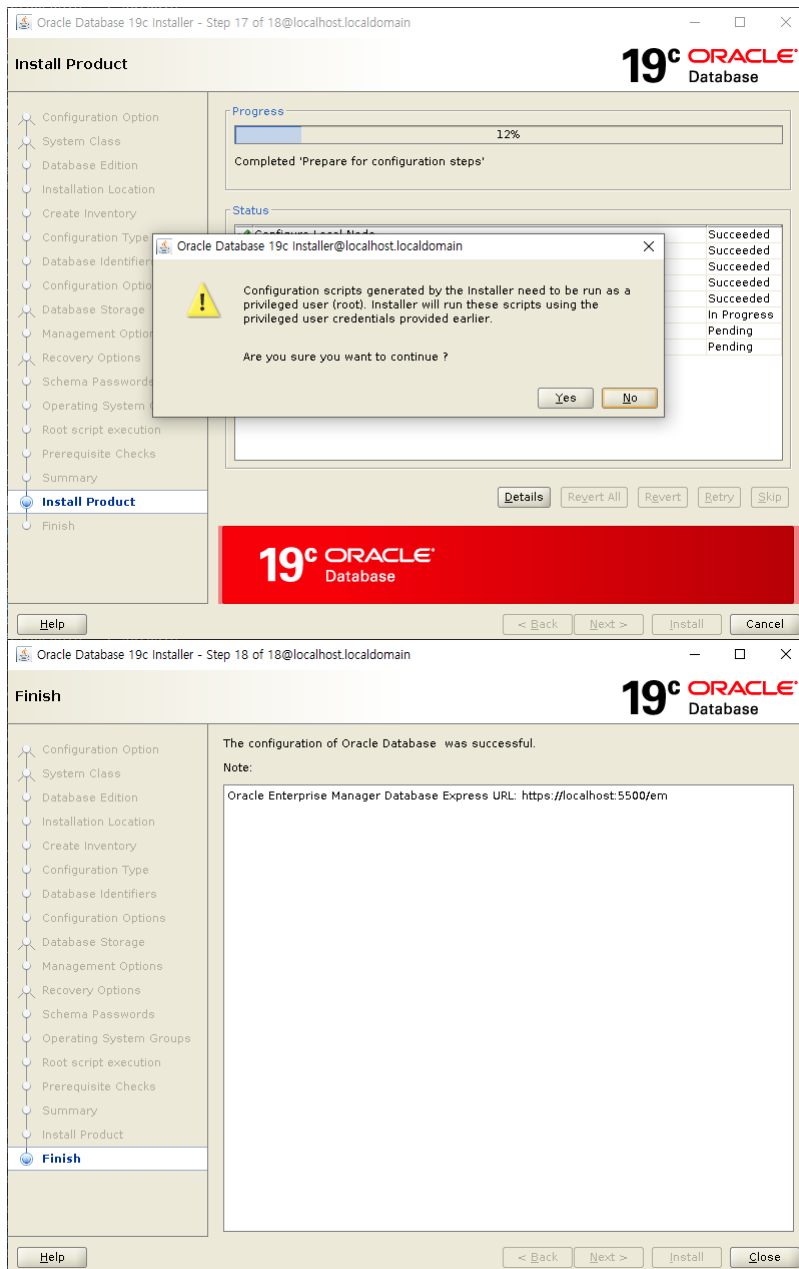
[Configuration Option](#)
[System Class](#)
[Database Edition](#)
[Installation Location](#)
[Create Inventory](#)
[Configuration Type](#)
[Database Identifiers](#)
[Configuration Options](#)
[Database Storage](#)
[Management Options](#)
[Recovery Options](#)
[Schema Passwords](#)
[Operating System Groups](#)
[Root script execution](#)
[Prerequisite Checks](#)
Summary
Install Product
Finish

Oracle Database 19c Installer

- Global settings
 - Install method: Advanced installation
 - Database edition: Enterprise Edition (Create and configure a database) [\[Edit\]](#)
 - Oracle base: /app/oracle [\[Edit\]](#)
 - Software location: /app/oracle/product/19.3.0.0.0/dbhome_1
 - Privileged Operating System groups: dba (OSDBA), dba (OSOPER), dba (OSBACKUPDB)
 - Root script execution configuration: Root user credential [\[Edit\]](#)
- Inventory information
 - Inventory location: /app/orainventory [\[Edit\]](#)
 - orainventory group: dba [\[Edit\]](#)
- Database information
 - Configuration: General Purpose / Transaction Processing [\[Edit\]](#)
 - Global database name: orcl [\[Edit\]](#)
 - Oracle system identifier (SID): orcl [\[Edit\]](#)
 - Allocated memory: 2397 MB [\[Edit\]](#)
 - Automatic memory management option: FALSE [\[Edit\]](#)
 - Database character set: AL32UTF8 (Unicode UTF-8 Universal character set) [\[Edit\]](#)
 - Management method: Database express [\[Edit\]](#)
 - Database storage mechanism: File system [\[Edit\]](#)
 - Database file location: /app/oracle/oradata [\[Edit\]](#)
 - Recovery: Enabled [\[Edit\]](#)

Save Response File...

[Help](#) < Back **Next >** Install Cancel



설치되었는지 확인

```
sqlplus -version
```

```
sqlplus / as sysdba
```

```
shutdown immediate
```

```
startup
```

```
exit
```

```
lsnrctl start
```

```
lsnrctl status
```


Oracle Admin

Startup

```
STARTUP [FORCE] [RESTRICT] [PFILE=<filename>] [OPEN [RECOVER] [<db>] | MOUNT | NOMOUNT]
```

- **FORCE**
 - Oracle DB가 시작된 상태에서 다시 재시작할 때만 사용한다.
- **RESTRICT**
 - DBA 권한을 가진 USER만이 connect하여 Oracle DB를 이용할 수 있다.
- **PFILE**
 - Oracle이 기본으로 제공되는 파라미터 파일인 INIT.ORA파일이 아닌 관리자가 생성한 파라미터 파일을 사용하여 오라클 서버를 시작할 때 사용한다.

Shutdown

```
SHUTDOWN [NORMAL | TRANSACTIONAL | IMMEDIATE | ABORT]
```

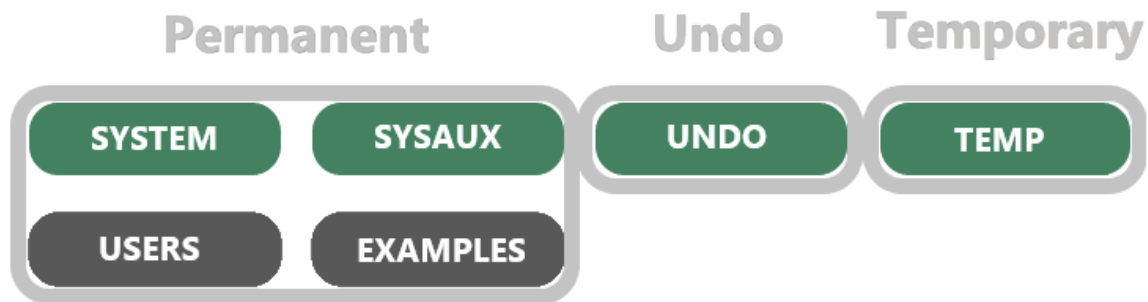
- **NORMAL**
 - 새로운 DB Connection을 허락하지 않는다.
 - 현재 USER들이 맺고 있는 Connection들은 Disconnect할 때 까지 기다린다.
 - 다음 DB STARTUP 시 Instance Recovery 절차가 필요하지 않다.
- **TRANSACTIONAL**
 - 모든 Client가 새로운 Transaction을 시작할 수 없다.
 - Client들의 진행중인 Transaction을 종료하면 서버가 종료된다.
 - 다음 DB STARTUP 시 Instance Recovery 절차가 필요하지 않다.
- **IMMEDIATE**
 - 현재 처리중인 SQL Statement가 있을 시 모두 Stop 시킨다.
 - Uncommitted Transaction이 있다면 다 Rollback 시킨다.
 - USER들이 Disconnect 할 때 까지 기다리지 않고 DB를 Close & Dismount하여 Oracle Instance를 SHUTDOWN시킨다.
 - 다음 DB STARTUP 시 Instance Recovery 절차가 필요하지 않다.
- **ABORT**
 - 현재 처리중인 SQL Statement들을 모두 Abort 시키고 Rollback 시키지 않는다.
 - USER들을 Disconnect 시키며 Close와 Dismount 없이 강제 종료한다.
 - DB가 비정상 종료된 후 다음 DB STARTUP 시 Instance Recovery 절차가 필요하다.
 - SMON에 의해 Instance Recovery 절차가 자동으로 수행된다.

Tablespace 및 Data File 관리

Tablespace

- 하나의 DB 안에 가장 큰 논리적 공간이다.
- 업무의 단위나 사용 용도에 따라 여러 개의 Tablespace로 분리되어 관리된다.
- Segment(Object)라는 논리적 저장 공간의 집합이다.

Tablespace의 종류



종류는 크게 3개로 나뉘며 필수 Tablespace는 4개이다.

- Permanent Tablespace
 - 영구 Tablespace이다.
 - 가장 일반적인 Tablespace로 데이터 축적 용도로 사용한다.
 - 임의로 USERS나 EXAMPLES 처럼 원하는 데이터를 저장할 수 있다.
 - SYSTEM
 - DB의 기본 정보들을 담고 있는 Data Dictionary Table이 저장되는 공간이다.
 - 일반 사용자들의 Object들을 저장하지 않는 것을 권장한다.
 - 사용자들의 Object에 문제가 생겨 DB가 종료되면 완벽한 복구가 불가능해지기 때문이다.
 - SYSAUX
 - SYSTEM Tablespace의 보조이다.
 - SYSAUX Tablespace에 문제가 생기면 시스템 상 문제가 없으나 SYSAUX Tablespace에 저장되어 있는 요소들의 기능은 사용 불가해진다.
- Undo Tablespace
 - 읽기 일관성을 유지하기 위해 사용한다.
 - Rollback하는 경우를 대비해 DML 발생 때 수정 이전 값을 UNDO Segment에 저장한다.
- Temporary Tablespace
 - 사용자 쿼리 요청으로 정렬하는 작업이 필요할 때 메모리의 부담을 덜어주기 위해 사용된다.

Data File

- Tablespace의 물리적 파일 형태이다.
- 하나 이상의 Data File이 모여 Tablespace를 형성한다.

Tablespace 생성

CREATE PERMANENT TABLESPACE

```
CREATE [BIGFILE | SMALLFILE(DEFAULT)] TABLESPACE <tablespace>
DATAFILE '<path>' SIZE <size> [EXTENT MANAGEMENT [DICTIONARY | LOCAL(DEFAULT)
[AUTOALLOCATE(DEFAULT) | UNIFORM SIZE <size>]]]
[SEGMENT SPACE MANAGEMENT [AUTO(DEFAULT) | MANUAL]];
```

```
-- EX)
CREATE TABLESPACE TS1 DATAFILE '/app/tablespace1.dbf' SIZE 100M;

SELECT TABLESPACE_NAME, CONTENTS, EXTENT_MANAGEMENT, ALLOCATION_TYPE,
SEGMENT_SPACE_MANAGEMENT, BIGFILE
FROM DBA_TABLESPACES WHERE TABLESPACE_NAME = 'TS1';
```

CREATE TEMPORARY TABLESPACE

```
CREATE [BIGFILE | SMALLFILE] TEMPORARY TABLESPACE <temp_tablespace>
TEMPFILE '<path>' SIZE <size> [EXTENT MANAGEMENT [DICTIONARY | LOCAL
[AUTOALLOCATE | UNIFORM SIZE <size>]]];
```

```
-- EX)
CREATE TEMPORARY TABLESPACE TEMPTS1 TEMPFIL 'app/temp_tablespace1.dbf' SIZE
100M;
```

CREATE UNDO TABLESPACE

```
CREATE [BIGFILE | SMALLFILE] UNDO TABLESPACE <undo_tablespace>
DATAFILE '<path>' SIZE <size> [EXTENT MANAGEMENT [DICTIONARY | LOCAL
[AUTOALLOCATE]]];
```

- UNDO Tablespace는 사이즈를 지정할 수 없다.
- SEGMENT SPACE MANAGEMENT는 MANUAL만 가능하다.

ADD / DROP TABLESPACE DATAFILE

```
-- ADD)
ALTER TABLESPACE <tablespace> ADD DATAFILE '<path>' SIZE <size>;

-- DROP)
ALTER TABLESPACE <tablespace> DROP DATAFILE <file_id>;
```

DROP TABLESPACE

```
DROP TABLESPACE <table_spacename> INCLUDING CONTENTS AND DATAFILES;
```

- CONTENTS: 모든 Segment들을 삭제한다.
- DATAFILES: 모든 Data File들을 삭제한다.

BIGFILE

- Data File을 하나만 사용할 수 있다.
- ASM이 생기며 만들어졌다.

SMALLFILE

- 여러 개의 디스크에 균등히 수동으로 Data File을 생성한다.

EXTENT MANAGEMENT

- Tablespace의 공간 할당은 Extent 단위로 진행된다.
- DML 작업이 반복되며 Extent의 할당과 반환이 발생하는 데 어느 Extent를 사용해도 되는 지에 관한 정보 관리가 필요하다.
- 해당 방법에는 DICTIONARY와 LOCAL 방법이 있다.
 - DICTIONARY
 - 사용 가능한 Extent에 대한 정보를 Data Dictionary에서 관리하는 방법이다.
 - Segment마다 다른 Extent 크기를 설정할 수 있다.
 - Data Dictionary에 대한 경합 발생 가능성이 높아 사용되지 않는다.
 - LOCAL
 - Data File Header에 Bitmap을 통해 Extent의 사용 유무를 관리한다.
 - Resource의 사용량이 높아지나 중요한 Object 경합을 줄이는 것이 더 중요해 생겼다.
 - AUTOALLOCATE 옵션을 사용하면 자동으로 Extent의 크기를 정하도록 위임 가능하다.
 - UNIFORM 옵션을 사용하면 모든 Extent의 크기를 동일하게 설정 가능하다.

SEGMENT SPACE MANAGEMENT

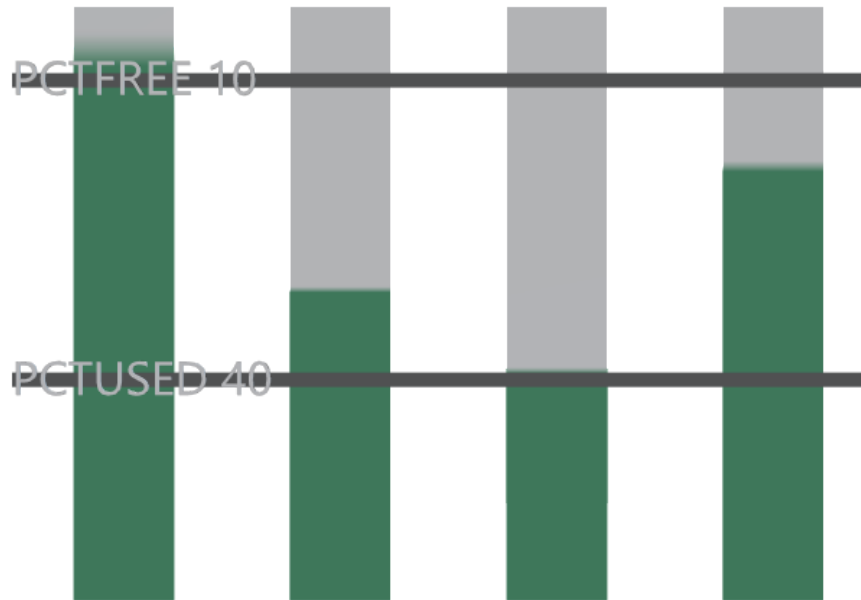
- Segment의 공간 관리에 대한 옵션이다.
- Tablespace의 공간 관리를 어떻게 할 것인가를 묻는 옵션이다.

```
SELECT TABLESPACE_NAME, SEGMENT_SPACE_MANAGEMENT FROM DBA_TABLESPACES;
```

| TABLESPACE_NAME | SEGMENT_SPACE_MANAGEMENT |
|-----------------|--------------------------|
| SYSTEM | MANUAL |
| SYSAUX | AUTO |
| UNDOTBS1 | MANUAL |
| TEMP | MANUAL |
| USERS | AUTO |

- 해당 방법에는 MANUAL과 AUTO가 있다.
 - MANUAL

- Freelist를 사용해 INSERT가 가능한 블록을 확인 가능하다.
- Free List Management라고도 한다.
 - PCTUSED: 일정 백분위 이하로 사이즈가 줄어들면 Freelist에 블록을 등록하는 크기이다.
 - PCTFREE: 일정 백분위 이하로 데이터 변경에 대비해 확보해 놓은 Block Size 크기이다.



- PCTFREE가 넘어갈 경우 Freelist에서 사용 가능한 다음 블록을 사용한다.
PCTUSED 이상인 블록들 밖에 없어 Freelist에서 사용 가능한 블록이 없을 경우 새로운 Extent를 할당해야 한다.
- AUTO
 - Bitmap을 이용해 비어있는 블록을 확인한다.
 - 네 개의 등급으로 나누고 총 여섯 가지 상태를 나타내는 Bitmap 블록을 사용해 Segment를 관리한다.
 - ASSM(Automatic Space Segment Management)이라고도 부른다.
 - Full: INSERT가 더이상 일어날 수 없다.
 - FS1: 0~25%의 여유 공간이 Block에 존재
 - FS2: 25~50%의 여유 공간이 Block에 존재
 - FS3: 50~75%의 여유 공간이 Block에 존재
 - FS4: 75~100%의 여유 공간이 Block에 존재
 - Never Used: 비어 있음
 - ASSM은 Freelist 대신 3단계 Bitmap Block을 이용해 효율적으로 Segment 공간을 관리한다.
 - L1BMB(Level 1 BitMap Block): 각 Block의 Freeness Status를 관리하는 역할을 한다.
Segment의 크기에 따라 16~1024개의 Block 상태를 관리한다.

- L2BMB: L1BMB의 목록을 관리한다.
L1BMB DBA, L1BMB가 관리하는 Block들의 Maximun Freeness, Owning Instance 등을 관리한다.
- L3BMB: L2BMB의 목록을 관리한다.
대부분 별도의 물리적인 Block으로 존재하지 않고 Segment Header Block 내부에 존재한다.
Segment 크기가 커서 하나의 L3BMB로 관리 불가능할 때 별도의 L3BMB가 물리적으로 분리된다.

유저 생성 / 삭제 / 권한 부여 / 권한 강탈

유저 생성

```
CREATE USER <id> IDENTIFIED BY <pw> [DEFAULT TABLESPACE <tablespace>] [TEMPORARY
TABLESPACE <temp_tablespace>];
```

유저 삭제

```
DROP USER <id> [CASCADE];
```

- CASCADE: 해당 스키마에 Object가 존재하면 삭제할 수 없다.
CASCADE 키워드를 통해 스키마에 속한 모든 Object를 함께 삭제할 수 있다.

유저 권한 부여

```
GRANT <privilege> TO <id>;
```

유저 권한 강탈

```
REVOKE <privilege> TO <id>;
```

테이블 생성 / 삭제

테이블 생성

```
CREATE TABLE <table>(
    <column> <type> <constraint>,
    ...
);
```

테이블 삭제

```
-- 데이터와 테이블 삭제
DROP TABLE <table>;

-- 테이블과 데이터, 제약조건 삭제
DROP TABLE <table> CASCADE CONSTRAINTS;

-- 테이블과 데이터를 휴지통에 넣지 않고 삭제
DROP TABLE <table> PURGE;

-- 테이블, 데이터, 제약조건 휴지통에 넣지 않고 삭제
DROP TABLE <table> CASCADE CONSTRAINTS PURGE;
```

테이블 변경

```
-- 이름 변경
ALTER TABLE <table> RENAME TO <new_table>;

-- Column 이름 변경
ALTER TABLE <table> RENAME COLUMN <column> TO <new_column>;

-- Column Data Type 변경
ALTER TABLE <table> MODIFY <column> TO <datatype>;

-- Column 추가
ALTER TABLE <table> ADD <column> <datatype>;

-- Column 삭제
ALTER TABLE <table> DROP COLUMN <column>;

-- 제약조건 추가
ALTER TABLE <table> ADD CONSTRAINTS <constraint_name> <constraint> (<column>, ...);

-- 제약조건 삭제
ALTER TABLE <table> DROP CONSTRAINTS <constraint_name>;
```

테이블 복사(CTAS)

- 테이블 구조 변경 전 테스트를 위해 복사하거나 백업 용도로 사용된다.

```
-- TABLE 구조와 데이터 모두 복사
CREATE TABLE <table> AS SELECT <column>, ... FROM <source_table>;
```

데이터 조회 / 생성 / 변경 / 삭제

데이터 조회

```
SELECT [<column> | * | <aggregate_function>] FROM <table> [WHERE <expression>]
[GROUP BY <column> [HAVING <expression>]]
[ORDER BY <column> [(ASC(DEFAULT) | DESC)]];
```

데이터 생성

```
INSERT INTO <table> [(<column>, ...) | ] VALUES (<value>, ...);
```

데이터 변경

```
UPDATE <table> SET <column> = <value>, ... [WHERE <expression>];
```

데이터 삭제

```
DELETE FROM <table> [WHERE <expression>];
```

Procedure

DB에 대한 일련의 작업을 정리한 절차를 RDBMS에 저장한 것이다.

넓은 의미로 어떤 업무를 수행하기 위한 절차를 뜻한다.

쿼리문을 하나의 메소드 형식으로 만들고 어떤 동작을 일괄적으로 처리하는 용도로 쓰인다.

| Procedure | Function |
|------------------------|--------------------|
| 특정한 작업을 수행한다. | 특정한 계산을 수행한다. |
| 리턴 값을 가질 수도, 아닐 수도 있다. | 리턴 값을 반드시 가진다. |
| 리턴 값을 여러 개 가질 수 있다. | 리턴 값을 하나만 가질 수 있다. |
| 서버(DB)에서 기술한다. | 화면(Client)에서 기술한다. |
| 수식 내에서 사용 불가하다. | 수식 내에서만 사용 가능하다. |
| 단독으로 문장 구성이 가능하다. | 단독으로 문장 구성이 불가능하다. |

PL/SQL

Procedural Language extension to Structured Query Language의 약자이다.

Oracle에서 지원하는 프로그래밍 언어의 특성을 수용한 SQL의 확장이다.

PL/SQL Block 내에서 SQL의 DML문과 Query문, 절차형 언어(if / Loop) 등을 사용하여 절차적 프로그래밍을 가능하게 한 트랜잭션 언어이다.

선언부, 실행부, 예외처리부로 총 세 개의 Section으로 구성된다.

```
DECLARE      -- 선언부
BEGIN        -- 실행부 시작
END;         -- 실행부 종료
EXCEPTION   -- 예외처리부
```

PL/SQL의 종류는 크게 다섯 개로 나눌 수 있다.

1. 익명 Procedure

- 이름 없이 사용되는 PL/SQL Block이다.
- DB에 저장되지 않고 사용자가 필요할 때마다 반복적으로 작성하고 실행한다.

2. Stored Procedure

- 생성 이후 DB에 정보가 저장된다.
- 실행하려는 로직을 처리하고 PL/SQL Block의 흐름을 제어한다.
- 인자를 받아서 호출되고 실행된다.

3. Stored Function

- Stored Procedure와 동일한 개념이나 기능이나 처리 결과를 사용자에게 전달한다.

4. Package

- 특정 업무에 사용되는 프로시저나 함수를 묶어 생성해 관리한다.

5. Trigger

- 지정된 이벤트 발생 시 자동적으로 호출되어 실행되는 특수한 형태의 Procedure이다.

Procedure 생성 / 변경 / 변경

생성 및 변경

```
CREATE OR REPLACE PROCEDURE <procedure>
(<parameter> IN <type>, ...)
IS
    <variable> <type> := <value>;
    ...
BEGIN
    ...
END;
EXCEPTION
/
```

삭제

```
DROP PROCEDURE <procedure>
```

Procedure 실행

```
EXEC <procedure>(<parameter_value>, ...);
```

Oracle Architecture

Oracle Server

Oracle Server란 Oracle Instance와 Database의 합집합이다.

Oracle Server에 대한 이해 이전에 Host / Server / Client를 구별하고 시작하겠다.

Host란 네트워크 주소가 할당된 네트워크 노드를 의미한다.

따라서 요청을 하는 노드와 요청을 처리해주는 노드는 모두 Host이다.

Server란 요청받는 정보를 제공해주는 장치이다.

따라서 Oracle Database가 설치되어 Instance가 가동되어 정보를 제공해주는 노드가 Oracle Server이다.

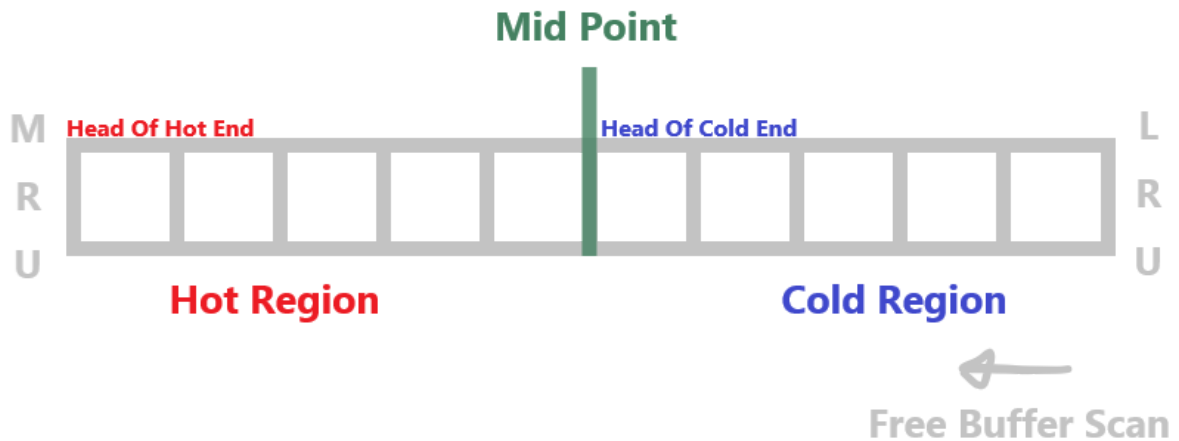
Client는 Host 중 정보를 요청하는 장치이다.

따라서 SQL Plus, SQL Developer, TOAD, DataGrip과 같은 소프트웨어가 설치된 노드는 모두 Client이다.

Oracle Queuing Algorithm

- 기본적으로 Oracle은 Memory에서 LRU(Least Recently Used) List를 사용한다.
- 사용 빈도가 높은 Buffer일수록 오래 DB Buffer Cache에 존재할 수 있는 Algorithm이다.
- LRU 보조 -> LRU 메인 -> LRUW 메인 -> LRUW 보조 순으로 순환하며 버퍼를 탐색한다.

LRU List



- Dirty Buffer를 제외한 모든 Buffer를 관리한다.
- 메인 리스트 : 사용된 버퍼들의 리스트가 hot, cold로 분류된다.
- 보조 리스트 : 미사용된 버퍼나 DBWR에 의해 기록된 버퍼들의 리스트이다.

LRUW List

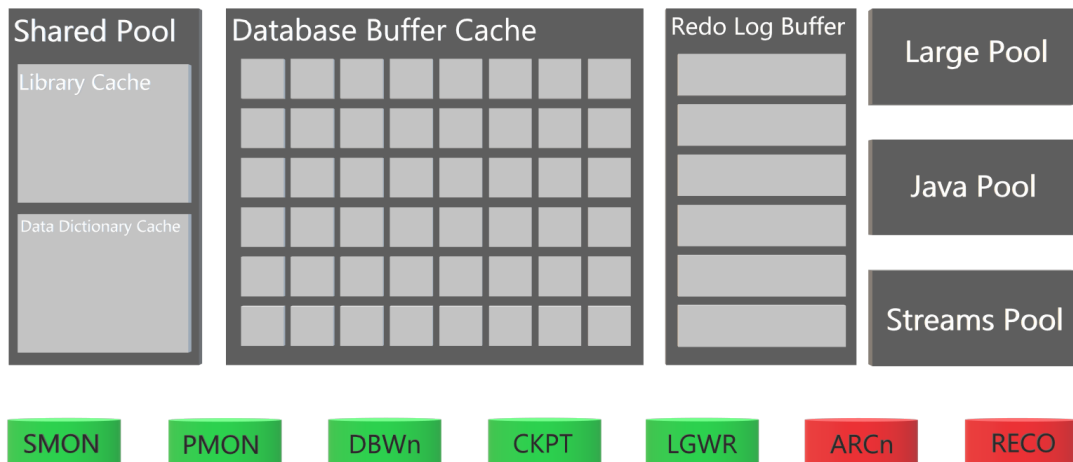
- 같은 Data Block에 대한 DB Buffer Cache에 저장된 Buffer Image와 Data File에 저장되어 있는 물리적인 Block Image가 서로 다른 Buffer들을 관리하는 리스트이다.
- 메인 리스트 : 변경된 버퍼들의 리스트이다.
- 보조 리스트 : DBWR에 의해 기록중인 버퍼들의 리스트이다.

Oracle Instance

Oracle Instance는 SGA와 Background Process의 집합이다.

PGA를 넣지 않은 이유는 서버의 설정마다 위치가 다르기 때문이다.

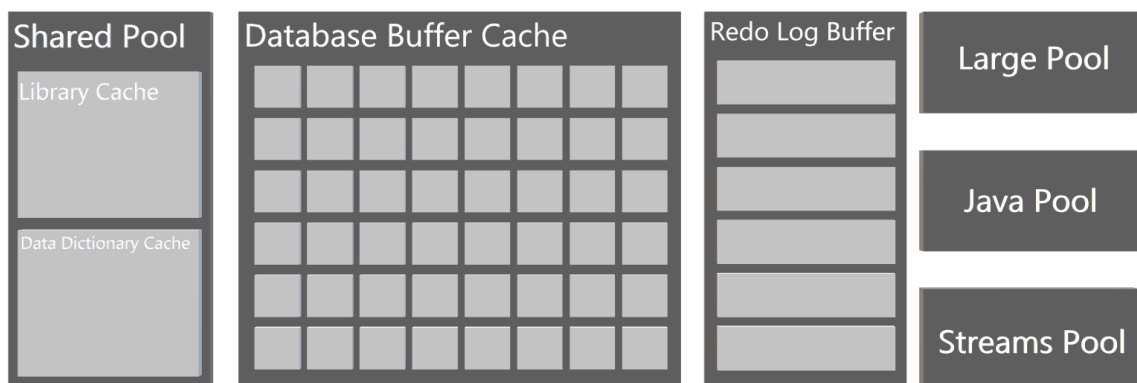
Oracle Instance



SGA(System Global Area)

- SGA는 간단하게 오라클서버의 메모리영역이다.
- SGA는 Oracle의 인스턴스에 대한 데이터와 제어 정보를 가지는 공유 메모리 영역의 집합이다.
- 목적의 따라 오라클 파라미터 파일(init.ora)의 조정으로 SGA의 각 부분의 크기를 조절 가능하다.
- Oracle9i부터 오라클 서버의 종료 없이 SGA의 구성을 SGA_MAX_SIZE 파라미터 값 범위 내에서만 각각의 크기를 동적으로 변경 가능하다.
- Oracle 서버를 동시에 사용하고 있는 사용자는 시스템 글로벌 영역의 데이터 공유한다.
- 전체 SGA를 실제 메모리 크기가 허용하는 범위에서 가장 크게 잡으면 디스크 I/O를 줄이고 메모리에 가능한 많은 데이터를 저장할 수 있으므로 최적의 성능을 낼 수 있다.
- SGA는 Shared Pool, DB Buffer Cache, Redo Log Buffer, Large Pool, Java Pool, Streams Pool로 구성되어 있다.

SGA



Shared Pool

- Library Cache와 데이터 사전 캐시(Data Dictionary Cache)로 구성한다.
- 하나의 데이터베이스에 실행되는 모든 SQL 문을 처리하기 위해 사용한다.
- 문장 실행을 위해 그 문장과 관련된 실행 계획과 구문 분석 정보가 포함된다.
- 사이즈는 `SHARED_POOL_SIZE` 파라미터 값으로 결정한다.

Library Cache

- 가장 최근에 사용된 SQL 문장의 명령문, 구문 분석 트리, 실행 계획 정보를 가진다.
- LRU 알고리즘으로 관리된다.
- Shared SQL과 Shared PL/SQL 영역으로 구분한다.
 - Shared SQL 영역: SQL문장에 대한 실행계획과 파싱 트리를 저장하고 공유한다.
동일한 문장이 다시 실행되면 Shared SQL 영역에 저장되어 있는 실행 계획과 파싱 트리를 그대로 이용하기에 SQL 문장 처리 속도가 향상된다.
 - Shared PL/SQL 영역: 가장 최근에 실행한 PL/SQL 문장을 저장하고 공유한다.
파싱 및 컴파일 된 프로그램 및 프로시저(함수, 패키지, 트리거)가 저장한다.

Data Dictionary Cache

- 테이블, 컬럼, 사용자 이름, 사용 권한 같은 가장 최근에 사용된 데이터 사전의 정보를 저장한다.
- 구문 분석 단계에서 서버 프로세스는 SQL문에 지정된 오브젝트 이름을 찾아내고 접근 권한을 검증하기 위해 Dictionary Cache의 정보를 찾는다.

DB Buffer Cache

- 가장 최근에 사용된 데이터를 저장하는 메모리 공간이다.
- 디스크에 완전히 쓰여지지 않는 수정된 데이터를 보유할 수도 있다.
- DB Buffer Cache에서 찾고 있으면 반환한다.(Logical Read)
- DB Buffer Cache에 없어서 Free Buffer를 확보 후 Disk에서 찾아 Cache하여 반환한다.(Physical Read)
- LRU 알고리즘에 의하여 가장 오래전에 사용된 것은 Disk에 저장, Memory에는 가장 최근에 사용된 데이터를 저장함으로써, Disk I/O이 줄어들고, DBS의 성능이 증가한다.
 - LRU List: Buffer Block들의 상태를 관리하는 리스트이다.
 1. 많은 사용자가 동시에 Physical Read를 하여 동시에 DB Buffer Cache의 Free Buffer를 찾으려 할 때 LRU List 참조한다.
 2. 동시성 관리를 위해 순번 제공한다.(Latch)
 3. 본인 순번이 올 때까지 대기한다.
- Buffer Status
 - **Free**: 사용해도 되는 Buffer이다.
 - **Clean**: Buffer의 Data와 DB File 내의 Data가 일치하는 상태이다.
 - **Pinned**: 현재 사용중인 Buffer, 누군가 읽거나 변경하고 있는 상태이다.
 - **Dirty**: Buffer의 Data와 DB File 내의 Data가 일치하지 않는 상태이다.

Redo Log Buffer

- 데이터베이스에서 일어난 모든 변화를 저장하는 메모리 공간이다.
- 장애 발생 시 Recovery를 위해 존재한다.
 - 만약 Log를 남길 수 없는 경우에는 DB가 종료되거나 대기한다.
- Redo Log Buffer에 기록되지 않는 경우는 아래와 같다.
 - Direct Load
 - table이나 index의 nologging 옵션인 경우

- table nologging 시 DML의 경우 제한적으로 Redo Log에 기록한다.
- DB에서 발생한 모든 변화는 LGWR에 의해 리두 로그 파일에 저장한다.
- Redo Log Buffer는 Database의 변경 사항 정보를 유지하는 SGA에 있는 Circular(순환) 버퍼이다.
- Redo Log Buffer의 크기는 Oracle Parameter `LOG_BUFFER` 에서 지정한다.

Large Pool

- Oracle 백업 및 복원 작업에 대한 대용량 메모리 할당, I/O 서버 프로세스 및 다중 스레드 서버, Oracle XA에 대한 세션 메모리를 제공하는 SGA의 선택적인 영역이다.
- `LARGE_POOL_SIZE` 파라미터로 관리되며, 기본 크기는 0 byte이다.

Java Pool

- 자바로 작성된 프로그램을 실행할 때 실행 계획을 저장하는 영역이다.
- `JAVA_POOL_SIZE` 파라미터로 관리되며, 기본 크기 24MB로 할당한다.

Streams Pool

- Oracle Streams 전용으로 사용되며 버퍼링된 Queue Message를 저장하고 Oracle Streams 캡처 Process 및 적용 Process에 대해 메모리를 제공하는 선택적인 영역이다.
- `STREAMS_POOL_SIZE` 파라미터로 관리되며, 기본 크기는 0 byte이다.

Oracle 필수 Background Process

Oracle DB가 시작되기 위해 꼭 필요하며 DB 종료 시 모두 종료된다.

SMON(System MONitor)

- Oracle Instance를 관리한다.
 - Instance Recovery 수행한다,
 - Startup 중 싱크 정보를 확인해 어긋날 경우 Redo Log Entires를 재실행 하여 서버의 싱크를 맞추는 과정이 Instance Recovery이다.
 - 인스턴스 복구는 저장되는 것까지 고려해야 한다.
 - 아래는 순서이다.
 1. DB 비정상 종료
 2. STARTUP
 3. MOUNT 단계에서 Data File의 SCN번호가 일치하지 않음 확인
 4. Roll Forward
 - Redo Log File의 정보를 Data File에 적용
 5. OPEN 단계에서 Roll Back
 - Undo Tablespace의 Undo Data를 사용해 Commit 되지 않은 내용 Roll Back
- 데이터 파일의 빈 공간을 연결해 하나의 큰 빈공간으로 만든다.
- 더 이상 사용하지 않는 임시 세그먼트 제거하여 재사용 가능하게 만든다.
- 오라클 인스턴스 fail시 복구하는 역할을 한다.

PMON(Process MONitor)

- 오라클 서버에서 사용되는 각 프로세스들을 감시한다.
- 비정상 종료된 DB 접속을 정리한다.
- 정상적으로 작동하지 않는 프로세스를 감시해 종료하여 비정상적 종료된 프로세스들에게 할당된 SGA 리소스를 재사용 가능하게 만든다.
- 커밋되지 않은 트랜잭션을 **ROLLBACK** 시킨다.

DBWn(DataBase WRiter)

- DB Buffer Cache에 있는 Dirty Block의 내용을 데이터 파일에 기록한다.
- DB Buffer Cache내의 충분한 수의 Free Buffer가 사용 가능해진다.
- LRU 알고리즘을 사용한다.
- n은 숫자로 DB Writer를 여러개 구성 가능하다.
 - Default 1 or CPU_CONT/8 중 큰 쪽 1~100
 - **DB_WRITER_PROCESSES** Parameter를 통해 설정 가능하다.
 - 처음 36개의 DB Writer Process의 이름은 DBW0-DBW9 및 DBWa-DBWz, 37~100번째 DB Writer Process의 이름은 BW36-BW99
 - 보통은 DBW0으로 충분하나 시스템에서 데이터를 많이 수정할 때 추가 Process를 구성 가능하다.
 - uniprocessor system(단일 프로세서 시스템)에서는 사용하지 않는다.
- 발생하는 이벤트는 아래와 같다.
 - Dirty Buffer 수가 임계값 도달
 - 프로세스가 지정된 개수의 블록을 스캔 하고도 Free Buffer를 발견하지 못했을 때
 - 시간 초과
 - CKPT가 발생 시
 - RAC ping이 요청되었을 때
 - Tablespace가 offline이나 read only로 변경되었을 때
 - TABLESPACE BEGIN BACKUP 명령 실행했을 때

LGWR(Log WRiter)

- DB Buffer Cache의 모든 변화를 기록한다.
- SGA의 Redo Log Buffer에 생겨나며 트랜잭션이 완료되었을 때 Redo Log Buffer의 내용을 Online Redo Log File에 기록한다.

CKPT(Check Point)

- 모든 변경된 DB Buffer를 디스크 내의 데이터 파일로 저장하는 것을 보장한다.
- 변화된 데이터 블록 수, 일정 간격을 두어 DBWn이 Dirty Buffer를 데이터 파일로 저장하도록 명령한다.
- 발생시 데이터 파일과 컨트롤 파일의 헤더를 갱신한다.
- 관련 오라클 파라미터는 아래와 같다.
 - **LOG_CHECKPOINT_TIMEOUT** : CKPT가 발생할 시간 간격 설정(단위: Sec)
 - **LOG_CHECKPOINT_INTERVAL** : CKPT가 발생할 Redo Log File의 블록 수 지정

- 발생하는 이벤트는 아래와 같다.
 - LOG SWITCH CHANGE
 - LOG_CHECKPOINT_TIMEOUT
 - 마지막 Redo Log 작성(tail of the log)으로 부터 설정한 시간(초 단위)
 - 해당 초 이후 Checkpoint 발생
 - LOG_CHECKPOINT_INTERVAL
 - Redo Log File Block 수로 Checkpoint 빈도 지정
 - DB Block이 아닌 OS Block 의 개수로 작동
 - 해당 OS Block 수 이후 Checkpoint 발생
 - SHUTDOWN
 - TABLESPACE OFFLINE

Ark 제품과 관련있는 Process

ARCn(ARCiver)

- LOG SWITCH 발생 시 Redo Log File들을 지정된 저장장치로 저장한다.
- 발생 이벤트는 아래와 같다.
 - Online Redo Log File이 꽉 찼을 때
 - ~DBA가 ALTER SYSTEM SWITCH LOGFILE의 명령어 실행
- n은 숫자로 Archiver를 여러개 구성 가능하다.
 - Default 2, 1~30
 - 데이터의 벌크 로딩과 같은 무거운 워크로드가 많을 경우 여러 개 사용 가능 하다.
 - LOG_ARCHIVE_MAX_PROCESS 파라미터를 통해 설정 가능하다.
- ARCHIVELOG 모드 일 때만 작동한다.

RECO(RECOVer)

- DB 복구 시 시작되는 프로세스이다.

Server Process

- 사용자가 오라클 Application Program을 실행 시켰을 때 사용되는 프로세스이다.
- 아래는 예시이다.
 - SQLPlus
 - Forms
 - ProC
 - DataGrip
 - DBeaver
- 사용자가 오라클 서버에 접속할 때마다 사용자 프로세스가 생성된다.
- 사용자가 실행시킨 SQL문을 Server Process에 전달하고, 그 결과를 Server Process에게 받는다.

User Process

- Oracle은 Server Process를 생성하여 접속된 User Process의 요구 사항을 처리한다.
- User Process와의 통신과 요구 사항을 수행하는 Oracle과의 상호 작용 담당한다.
- Oracle은 Server Process당 User Process 수를 조정하도록 구성 가능하다.
- **전용 서버(Dedicated Server)** 구성에서 Server Process는 단일 User Process에 대한 요구 사항을 처리한다.
- **공유 서버(Shared Server)** 구성에서는 여러 개의 User Process가 적은 수의 Server Process를 공유하여 Server Process 수를 최소화하는 동시에 사용 가능한 시스템 자원 활용도를 최대화한다.
- 오라클 Server Process는 사용자로부터 받은 요구사항(SQL문)을 처리한다.
- 전달받은 SQL문을 Parse, Bind, Execute, Fetch 작업을 통해 실행시키는 역할을 수행한다.

Parse, Bind, Execute, Fetch

1. Parse - 동일한 쿼리인지 검색한다.
 - SQL문 문법 검사
 - 사용자 인증 및 권한 검사
 - 객체의 사용 가능 여부 검사
2. Bind
 - bind 할 값이 있다면 값을 치환해 변수값을 적용해 Execute 과정으로 넘긴다.
 - 없을 경우 바로 Execute 과정으로 넘긴다.
3. Execute
 - Parse 과정에서 만들어진 Parse Tree로 원하는 데이터 찾는다.
 - DB Buffer Cache에서 데이터를 찾은 후 있다면 재사용한다.
 - DB Buffer Cache에 존재하지 않으면 Data File에서 필요한 Block 적재 후 사용한다.
 - 필요할 경우 데이터를 수정한다.
4. Fetch
 - 데이터를 User Process에게 전달한다.

PGA(Program Global Area)





- 하나의 단일 프로세스에 대한 데이터와 제어 정보를 가지고 있는 메모리 공간이다.
- `PGA_AGGREGATE_TARGET` parameter 값을 통해 사이즈 조절 가능하다.
- USER PROCESS가 Oracle Database에 접속하고 Session이 생성될 때 Oracle에 의해 할당된다.
- 각 SERVER PROCESS에 하나만 할당한다.(1 : 1)
- 다른 프로세스와 공유되지 않는 독립적으로 사용하는 non-shared 메모리 영역이다.
- 세션 변수, 배열, 다른 정보를 저장하기 위해 스택 영역을 사용한다.
- PGA는 프로세스가 생성될 때 할당, 프로세스가 종료될 때 해제된다.
- PGA는 모드 구성에 따라 저장 위치가 다르다.
 - Dedicated Server
 - User Session Data, Cursor State, Sort Area 영역을 PGA 공간에 저장한다.
 - Shared Server
 - User Session Data 영역을 SGA에 저장한다.
- Memory가 가득 찰 시 Temp Tablespace로 간다.

UGA

User Session Data

- 추출된 결과 값을 전달하기 위해 User Process의 Session 정보를 저장한다.
- SQL문 결과를 User Process에게 전달하기 위해 User Session Address를 저장한다.

Cursor State

- 해당 SQL의 Parsing 정보가 기록되어 있는 주소를 저장한다.
 - 실행한 SQL문의 위치이다.

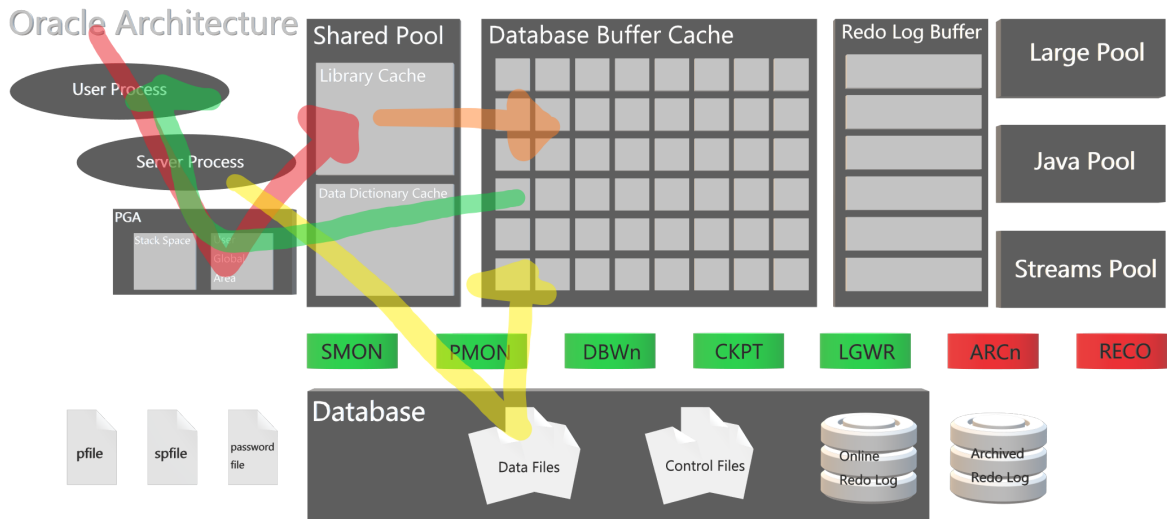
Sort Area

- 정렬 시 사용하는 공간이다.
- SQL의 작업 공간이며 가장 많은 공간을 할당한다.

Stack Space

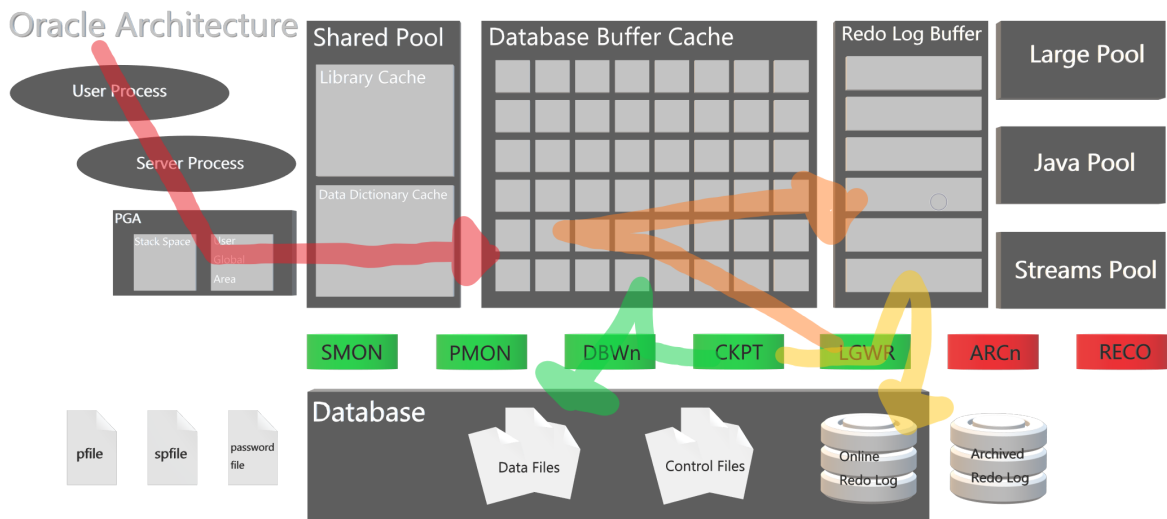
- SQL문의 Bind 변수를 사용할 때 저장하는 공간이다.

Select 흐름



1. Client가 SELECT절을 날리면 Server Process는 Shared Pool에 Library Cache를 확인해 Execute Plan이 있으면 Soft Parsing, 없을 경우 Optimizer가 Execute Plan을 만들어 Hard Parsing한다.
2. Server Process는 DB Buffer Cache를 읽는다.
3. 없을 경우 Data File로부터 읽어와 DB Buffer Cache에 올린다.
4. 해당 결과를 Client에게 전달한다.

INSERT 흐름



1. Client가 INSERT절을 날리면 Server Process가 DB Buffer Cache에 데이터를 담는다.
2. 데이터를 담을 때 LGWR는 변경내용을 Redo Log Buffer에 담는다.
3. Redo Log Buffer에 담긴 내용은 commit이나 특정 시간마다 LGWR가 Redo Log File에 내린다.
4. DBWn는 Checkpoint 발생 시 DB Buffer Cache상에 모든 Dirty Buffer를 Data File에 저장한다.

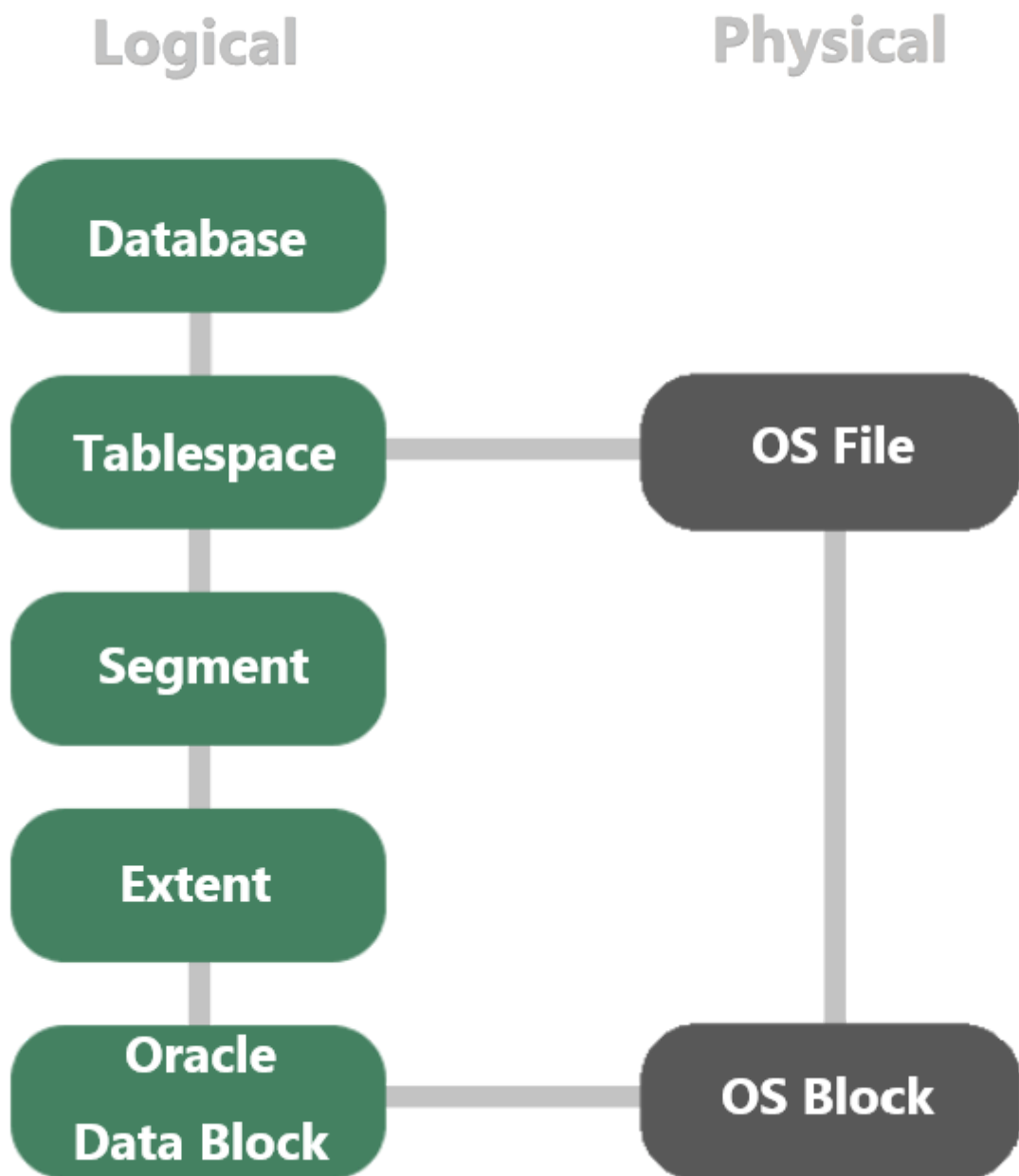
Oracle Database

Oracle Database는 Data File들과 Control File들과 Redo Log File들의 집합이다.

Data Files

- 실제 데이터가 저장되는 하드디스크상의 물리적 파일이다.
- 테이블이나 인덱스 같은 DB의 논리적 구조는 DB를 위해 할당된 Data Files에 물리적으로 저장한다.
- 생성시 그 크기를 명시하고 더 필요할 경우 확장 가능하다.
- Oracle에 의해 생성 및 삭제 되어야 한다.(운영 체제 명령으로 삭제 및 이동 금지)

Oracle 논리적 / 물리적 구조



Oracle의 논리적 구조는 DB, Tablespace, Segment, Extent, Oracle Data Block으로 이루어져 있다. 논리적 구조에 해당하는 물리적 구조는 Oracle Data Block은 OS Block에 해당하며 OS에 생성된 File(.dbf 등등)들 N개가 하나의 Tablespace에 해당한다. Tablespace의 경우 여러 Tablespace가 하나의 OS File에 존재할 수 없으며 Tablespace 당 N개의 OS File을 가진다.

Oracle Data Block

- 실제 데이터가 기록된다.
- Block Header에는 Block을 관리하기 위한 데이터가 있다.
- Oracle에서 공간을 할당하는 최소 단위이다.
- 2, 4, 8, 16, 32KB 등 크기가 다양하다.
- Oracle Data Block 한 개는 OS Block N개가 모여 생성된다.

Extent

- 여러 개의 연속된 Oracle Data Block의 집합이다.
- 테이블에 데이터가 없어도 자동 할당되며 최초 Extent 할당을 다 사용했을 시 추가로 생성할 수 있다.
- Extent의 크기는 Segment 생성 시 `STORAGE` 라는 Parameter로 수동으로 지정 가능하다.
 - `STORAGE` Parameter 생략 시 Tablespace의 기본 설정 값을 적용 받는다(Minimal 64KB)
- 테이블 별로 Block의 구역을 나눠 저장하여 검색 범위를 줄이는 데에 Extent가 사용된다.

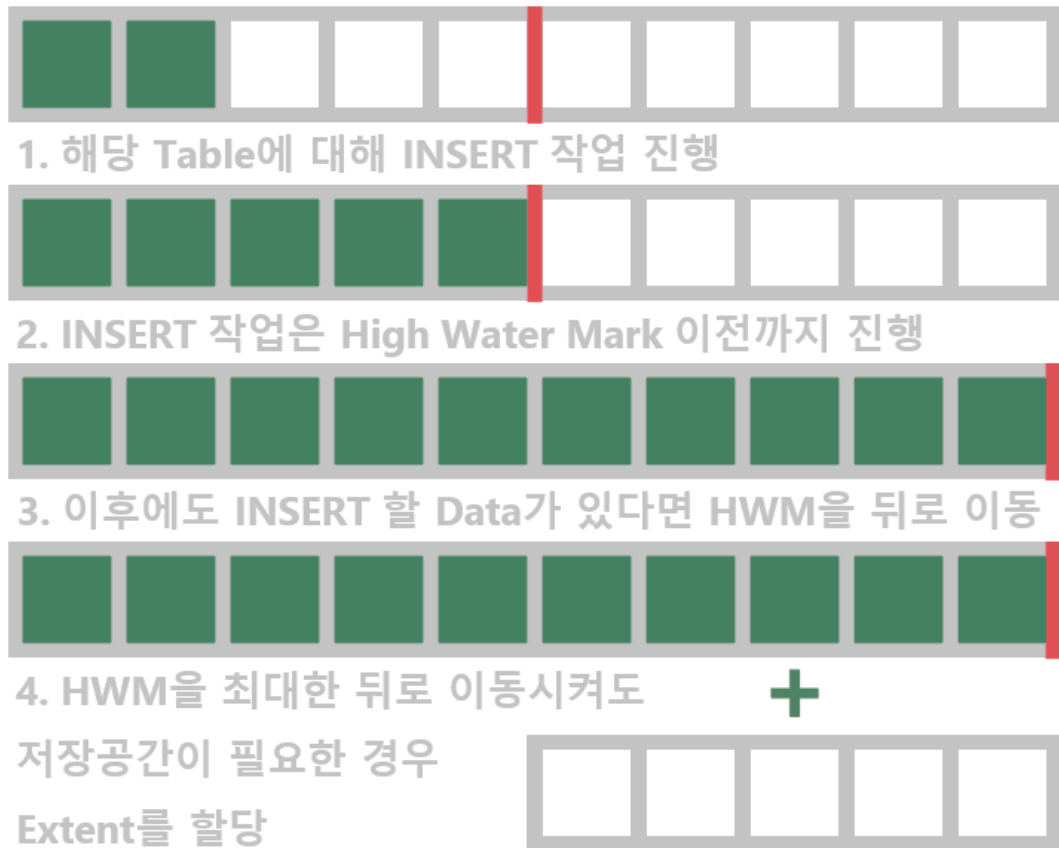
Segment

- 여러개의 Extent들의 집합이다.
- Table, Index, Undo, Temp와 같이 저장 공간을 필요로 하는 Object이다.
- Segment의 여러 Extent 중 가장 첫 번째의 첫 Block에 Segment Header가 존재한다.
- Segment Header에 해당 Segment의 종류에 대한 정보가 들어가며 Extent의 할당 상태와 공간 사용 내역이 들어간다.

Tablespace

- Segment들의 집합이다.
- Table이 존재하는 공간이다.

Segment의 증가



Segment에는 High Water Mark(이하 HWM)가 존재한다.

Segment Header에 있으며 모든 Segment 당 하나씩 존재한다.

저장 공간을 갖는 Segment 영역에서 사용한 적이 있는 Block과 사용한 적이 없는 Block의 경계점을 나타낸다.

HWM 이전 블록에만 저장 가능하며 데이터를 넣기 위해 HWM는 증가할 수 있으나 데이터가 제거되어도 감소하지 않는다.

DB를 스캔할 때 HWM까지 Data Block 전체를 확인한다.

Table Drop이나 Truncate를 통해 Table을 초기화하여 HWM를 초기화할 수 있다.

| 명령어 | 종류 | 설명 | Rollback 여부 | HWM 여부 |
|----------|-----------|---------|---------------|------------|
| DELETE | DML | 데이터 삭제 | 가능(COMMIT 이전) | 유지 |
| TRUNCATE | DML / DDL | 테이블 초기화 | 불가 | 해제 |
| DROP | DDL | 테이블 삭제 | 불가 | 삭제로 인해 사라짐 |

Data Block의 옵션

- **INITRANS**: Block이 생성될 때 동시 접근 가능한 트랜잭션의 슬롯 개수 지정한다.
트랜잭션이 많이 발생하면 **MAXTRANS** 까지 늘어나며 **PCTFREE** 로 확보된 영역에 추가 확장 가능하다.
- **INITRANS** 값을 크게 설정하면 블록 공간이 감소한다.

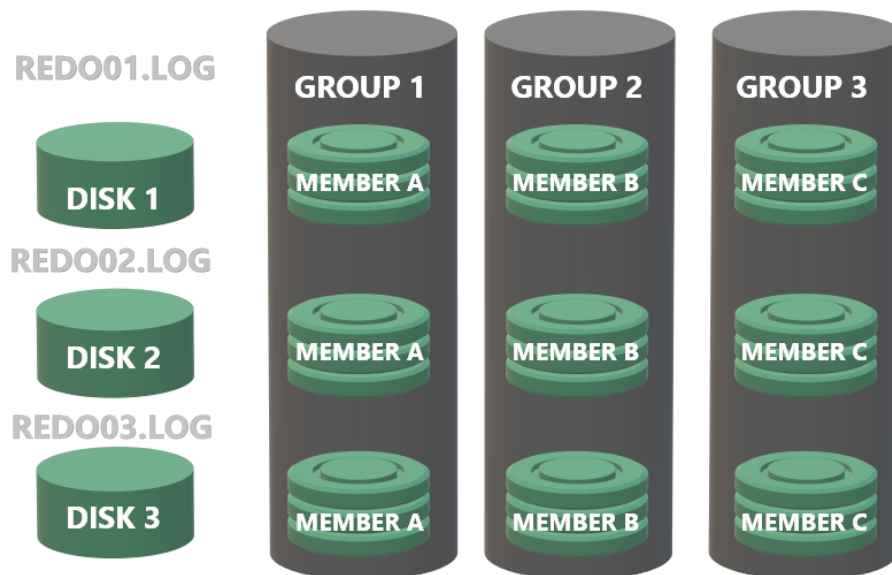
- **MAXTRANS**: Block에 접근 가능한 최대 트랜잭션 개수이다.
트랜잭션의 수가 **MAXTRANS** 을 초과할 때 앞 트랜잭션이 **COMMIT** or **ROLLBACK** 해야 사용 가능하다.
- **PCTUSED**: Block 재사용 여부 결정 Default = 40% 지정값 이하의 Block 사용량이면 저장 가능하다.
PCTUSED 가 높으면 공간 활용도는 높아지나 Free List에 등록 제거를 반복해 처리 비용이 증가한다.
 - Free List: 데이터가 입력될 수 있는 Block List이다.
- **PCTFREE**: Block Row Data의 길이가 늘어날 것을 대비하는 여유 공간의 확률이다. 기본값은 10%이다.
PCTFREE 가 높으면 Row Migration과 같은 문제를 줄이나 저장 공간이 줄어 비효율적이다.
 - Row Migration: **UPDATE** 로 인해 행 길이가 증가했을 때 저장 공간이 부족하면 발생한다.
원래 정보를 기존 Block에 남겨두고 실제 데이터는 다른 Block에 저장한다.
 - Row Chaining: 데이터가 커 여러 블록에 나누어 저장하는 현상이다.

Control Files

- DB의 제어 정보를 가지고 있는 파일이다.
- DB 이름이 Control File에 저장된다.
- Oracle DB를 **MOUNT**, **OPEN** 하여 사용하는데 필수적인 파일이다.(Control File을 백업해놓는 것이 좋다.)
- Binary File이라 직접 접근이 불가능하다.

Redo Log Files

- DB의 모든 변화를 기록하는 파일이다.
- 수정 내용을 Data Files에 반영하지 못해도 변경 사항이 저장되어 있어 유실되지 않는다.
- DB 장애를 보호하기 위해 필수적이다.
- 데이터 복구에 사용된다.
- SGA 내의 Redo Log Buffer Cache에 저장된 데이터들은 Redo Log Buffer가 일정 수준 이상 채워질 때 LGWR에 의해 Redo Log File로 저장된다.
- 적어도 두개 이상의 그룹을 가지며 한 그룹 내 각 멤버들은 모두 동일한 데이터를 가진다.



- Redo Log File Group이 가득 찼을 때 LGWR은 다음 그룹에 기록한다.(Log Switch)
- Online Redo Log File과 Archived Redo Log File이 있다.
 - Online Redo Log File: 모든 변경사항을 저장하는 공간, 복구를 위한 필수적인 공간이다.
 - Archived Redo Log File: Online Redo Log File가 덮어쓰이기 전 반영구적 보관을 위해 백업한다. 선택적인 공간이다.
- Redo Log File 상태는 아래와 같다.
 - **UNUSED**: 생성 이후 사용하지 않은 상태이다.
 - **CURRENT**: 현재 Redo Log File을 LGWR이 내용을 기록하는 상태이다.(활성 상태)
 - **ACTIVE**: 데이터가 찼으나 디스크에 저장하지 않은 상태이다.(활성 상태)
 - **INACTIVE**: 데이터를 디스크에 저장하여 삭제되어도 되는 상태이다.(비활성 상태)
 - **CLEARING**: 초기화 상태, 이후 **UNUSED** 상태로 변경된다.
 - **CLEARING_CURRENT**: **CURRENT** 상태에서 초기화가 진행중인 상태이다. 이후 **UNUSED** 상태로 변경된다.
- 순환형으로 사용한다.(재사용한다는 뜻)
 - Group은 최소 2개 이상 존재한다.
 - Group당 Member가 최소 1개 이상 존재한다.

SCN

- System Commit Number의 약자이다.
- **commit** 발생 시 Transaction이 부여받는 고유한 번호이다.
- Instance Recovery때나 USER가 **RECOVER** 명령을 수행할 때 DB에 문제가 있는지 판단하는 지표이다.
- DB를 다시 생성하지 않는 이상 RESET되지 않는다.

- SCN Base(4 bytes) + SCN Wrap(2 bytes)로 구성되어 있다.
- Sequence에서 발상하는 것이 아니라 kcmgas라는 function에서 구현된다.

SCN 기록 Solution

1. Control File Header

- Checkpoint 발생 시
- Resetlogs 발생 시
- Incomplete Recovery 수행 시

2. Data Blocks(Cache Layer)

- Block Cleanout 시 마지막 SCN을 각 Block에 기록

3. Data Blocks(ITL entires)

- Data Block의 Transaction Layer 안에 있는 Interested Transaction List Entries에 `commit` 된 SCN 정보 기록

4. Data File Headers

- 마지막 Checkpoint 발생 시
- Begin Backup 수행 시
- Recovery 되었다면 사용자의 마지막 SCN 기록

5. Redo Records / Log Buffer

- `commit` 수행 시 commit record에 SCN을 포함하여 저장

6. Rollback Segment(Undo Segment)와 Tablespace Headers에도 기록

Backup & Recovery

Backup

Recovery

ASM

RAC

Multitenant

Other Database

MySQL / MariaDB

PostgreSQL

Tibero

Ark Product

Ark for FR

Ark for ORacle

Ark for CDC
