

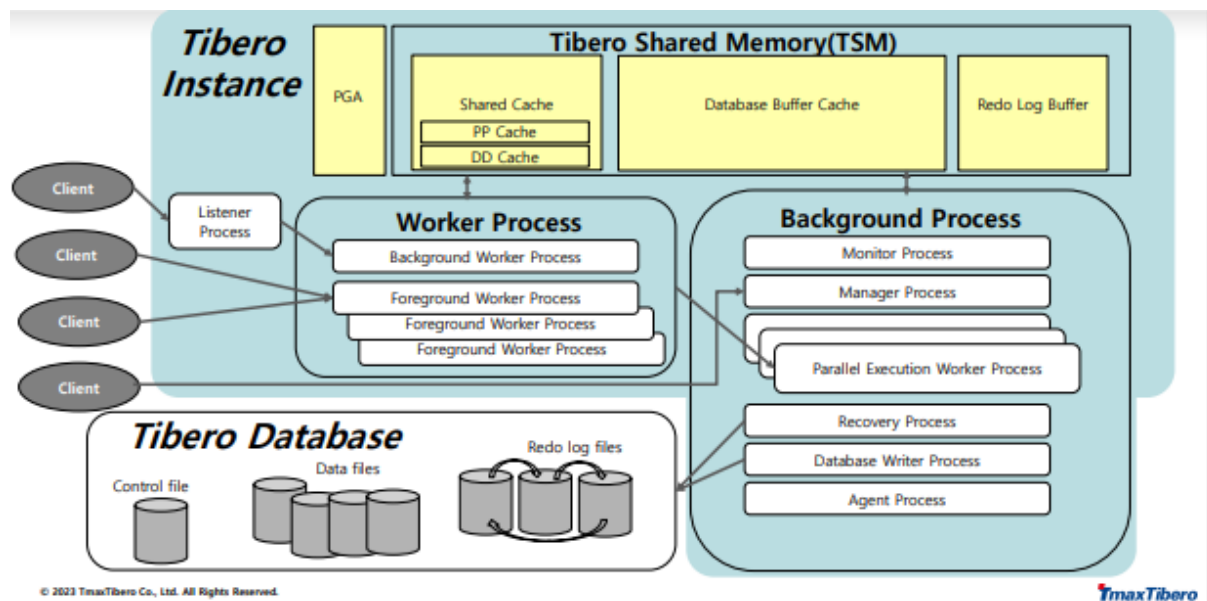
# 티베로

-구조

😂 티베로 6설치 (CENT OS).

Tibero 백업&리커버리

티베로 실습



주요 구성원으로는 티베로 데이터베이스, 워커프로세스, 백그라운드프로세스, 티베로 웨어드 메모리가있다.

가장중요한거는 리스너이다

사용자가 접근시 워커프로세스에게 작업을 할당한다.

이후 워커프로세스와 백그라운드 프로세스가

쿼리문이나 데이터베이스에 관한 작업을 처리한다.

## 워커 프로세스

WTHR이 사용자와 일대일로 연결된다.

워커 프로세스는 언제 시작할까? -> 한 번에 최대 개수가 모두 시작된다. 대기상태로 있다가 사용자가 접속하고 sql문을 보내면 sql을 처리해서 동작한다. 리스너, 작업 프로세스 등은 중간에 시작하지 않는다. 최대갯수만큼 시작했다가 대기한다.

Foreground 워커 프로세스: 리스너를 통해 요청 처리

Background 워커 프로세스: 리스너의 요청 처리를 받지 않는다.

## LISTENER

### 파라미터 파일

파라미터 파일의 역할

- 프로세스 동작의 제어
- 리스너 포트도 이 파일에서 제어

리스너 포트가 여러 포트를 리스닝하게 할 수 있다.

정적 파라미터 파일: 파라미터 파일을 편집해서 **tbBoot** (인스턴스 재시작)로 포트 적용

동적 파라미터 파일: tbBoot를 사용하지 않고 sql쿼리를 통해 포트 적용

정적으로 설정해주어야 할 파라미터가 있고 동적으로 설정해주어야 할 파라미터가 있다. 예를 들어 워커 스레드의 개수는 동적으로 설정할 수 없고 정적으로 인스턴스 재시작을 통해 설정해주어야 한다.

파라미터는 약 2000여개 정도. 대부분 기본값이 설정되어 있다.

Tibero6 온라인 매뉴얼에 따르면 파라미터는 300여가지인데, 매뉴얼에 실리지 않은 1500여 가지의 파라미터는 히든 파라미터이다.

### WTHR(워커 스레드)

- 클라이언트가 보내는 메시지 (String)을 받아 파싱 등의 처리를 하고 결과를 리턴한다.
- 파일에서 데이터를 찾아 메모리에 올리고 메모리 작업을 처리하는 것
- SQL Parsing, 최적화, 수행 등 DBMS가 해야하는 대부분의 일을 처리한다.
- 워커 스레드의 개수가 1000개라면 1000개의 클라이언트가 동시에 작업을 처리가능하다.

동시에 많은 양의 워커 스레드 사용이 발생하는 경우

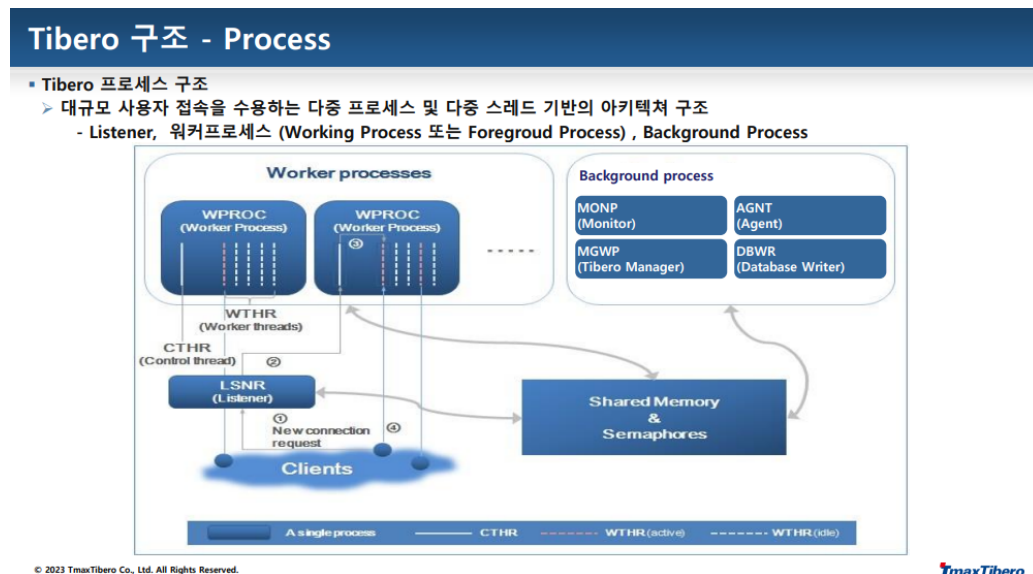
WAS서버는 sql형태로 처리한다.

동시에 많은 insert쿼리가 발생한다

파라미터 파일에 워커 스레드 관련 설정을 입력해서 조절한다.

만약 워커스레드를 너무 큰 값으로 설정해둔다면 메모리가...

(하나의 워커스레드는 약 2MB)



워커 프로세스: 사용자와 직접 통신을 하며 사용자 요구 사항을 처리하는 프로세스

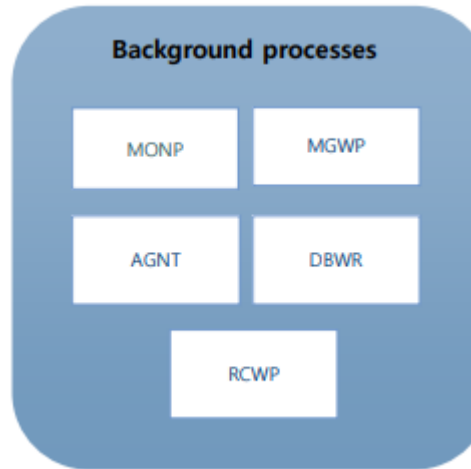
CTHR (Control Thread)

- 각 Working Process마다 하나씩 생성. 서버 시작 시에 지정된 개수의 Worker Thread를 생성.
- 시그널 처리 담당.
- I/O Multiplexing을 지원하며, 필요한 경우 워커 스레드 대신 메시지 송/수신 역할 수행

WTHR (Worker Thread).

- 각 Worker Process마다 여러 개 생성.
- Client 가 보내는 메시지를 받아 처리하고 그 결과를 리턴.
- SQL Parsing, 최적화, 수행 등 DBMS가 해야 하는 대부분의 일 처리.

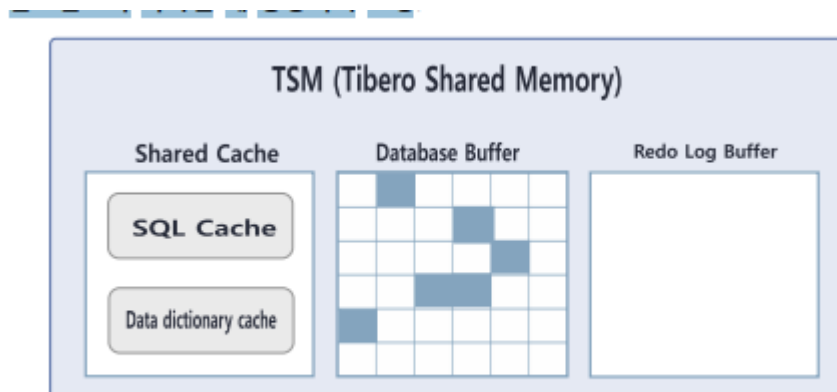
백그라운드 프로세스 : 사용자의 요청을 직접 받아들이진 않고 다른 배경 프로세스가 요청시 동작하며 주로 시간이 오래걸리는 디스크 작업을 담당한다.

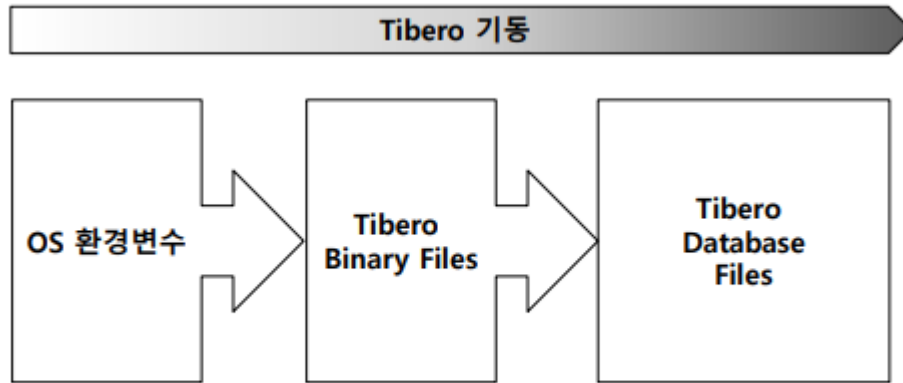


## TSM(TIBEERO SHARED MEMORY)

- 인스턴스에 대한 데이터와 제어 정보를 가지는 공유 메모리 영역
- 사용자가 동시에 데이터를 공유
- Background Process는 인스턴스가 시작될 때 TSM 영역을 할당하고, 인스턴스가 종료하면 할당 해제

TSM의 전체 크기는 인스턴스가 시작될 때 생성되어 고정





## 기동구조

ibero 는 OS 환경변수를 참조하여 구동 되므로, 적절한 값이 설정되어 있어야 한다

환경변수	설명
TB_HOME	Tibero가 설치된 디렉터리이다. <b>예: /tibero/tibero6</b>
TB_SID	Tibero 인스턴스 식별자 이며, 파라미터 파일의 이름, 각종 로그 생성 경로 등에 영향을 준다 <b>예: tibero</b>
LD_LIBRARY_PATH	Tibero를 사용할 때 필요한 공유 라이브러리가 위치한 경로이다. 필요한 라이브러리는 모두 \$TB_HOME/lib와 \$TB_HOME/client/lib 안에 있고, OS별로 환경변수가 다르게 지정된다.  SunOS, Linux : LD_LIBRARY_PATH HP-UX : SHLIB_PATH AIX : LIBPATH
PATH	Tibero 의 실행파일 디렉터리 경로이다. OS 기본 \$PATH 에 \$TB_HOME/bin 와 \$TB_HOME/client/bin 을 추가하여 설정한다 <b>예: /tibero/tibero6/bin , /tibero/tibero6/client/bin</b>

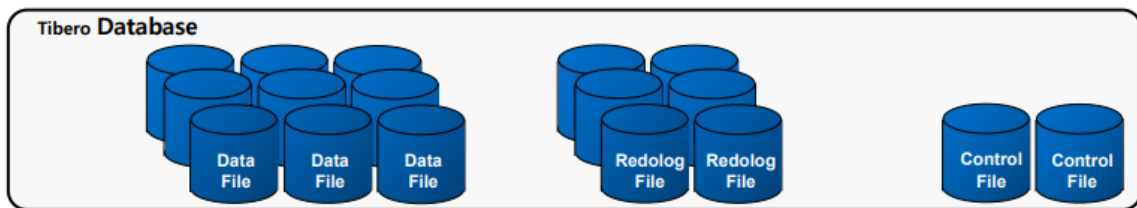
© 2023 TmaxTibero Co., Ltd. All Rights Reserved.

**TmaxTibero**

## 폴더구조

디렉토리	설명
<b>bin</b>	실행 파일과 서버 관리를 위한 유틸리티가 위치한 디렉터리이다. 이 디렉터리에 속한 파일 중에서 tbsvr과 tblastener는 Tibero를 구성하는 실행 파일이며, tbboot와 tbdownd는 각각 Tibero를 기동하고 종료하는 역할을 담당한다.
<b>client</b>	클라이언트 실행 파일이 있는 디렉터리이다. 이 디렉터리에는 다음과 같은 유틸리티가 있다.( tbSQL, T-Up, tbExport, tbImport, tbLoader, tbpc )
<b>config</b>	환경설정 파일이 위치하는 디렉터리이다. 이 위치에 존재하는 \$TB_SID.tip 파일이 Tibero의 환경설정을 결정한다.
<b>database</b>	데이터베이스 저장 디렉터리 기본 경로 ( 데이터베이스 정보를 별도로 설정하지 않는 한 모든 데이터베이스 정보가 이 디렉터리와 그 하위 디렉터리에 저장된다 )
<b>instance</b>	인스턴스 동작시 발생하는 각종 로그를 기록하는 디렉터리이다
<b>lib</b>	공간(Spatial) 쿼리와 관련된 함수를 사용하기 위한 라이브러리 파일이 있는 디렉터리이다
<b>license</b>	라이선스 파일(license.xml)이 있는 디렉터리이다. XML 형식이므로 일반 텍스트 편집기로도 라이선스의 내용을 확인할 수 있다.
<b>nls</b>	시간대 파일이 있는 디렉터리이다
<b>scripts</b>	데이터베이스를 생성할 때 사용하는 각종 SQL script 가 있는 디렉터리이다

TIBERO server 기동을 위해서는 데이터베이스 파일이 정상적으로 생성되어야만한다.



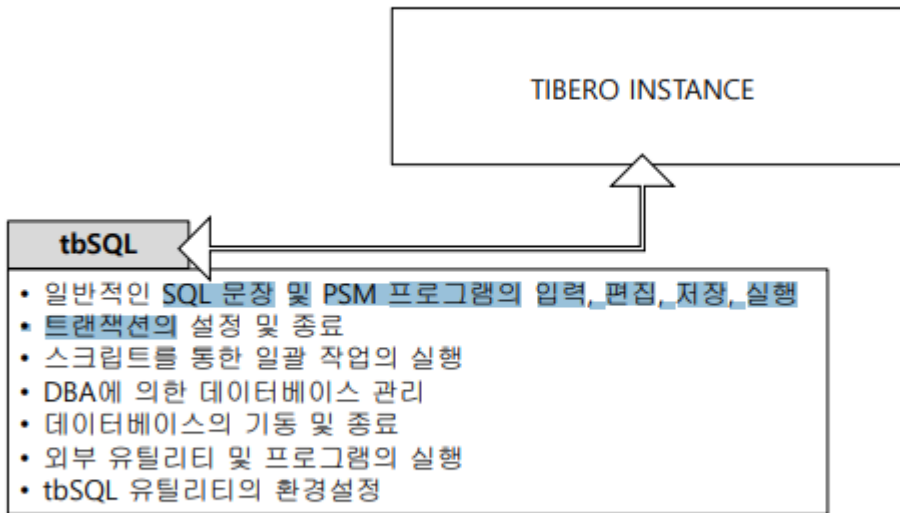
- datafiles: table indexes들과 같은 데이터 저장 객체를 물리적으로 저장하고 있다.
- REDO log files: 복구를 위하여 데이터베이스에서 변경된 모든 것을 기록하는 파일
- Control files: 데이터베이스의 물리적구조와 | 상태를 기록하는 파일

## TBSQL

티베로에서 기본적으로 제공하는 sql툴이다..

비교하자면 오라클의 sql플러스이다.

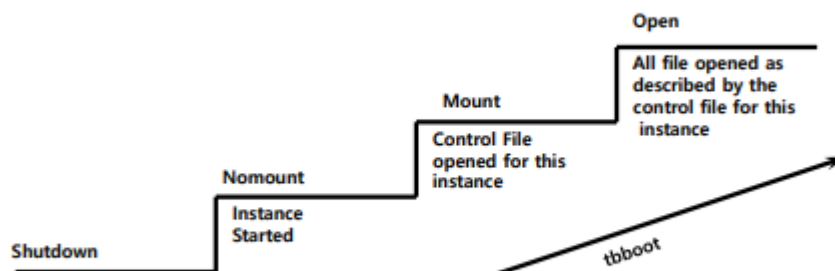
티이다. SQL 질의, 데이터 정의어, 트랜잭션과 관련된 SQL 문장 등을 실행할 수 있다.



옵션	설명
	옵션이 없는 경우 Tibero를 부트 모드(bootmode) 중 NORMAL로 기동하는 옵션이다.
-h	tbboot를 사용하기 위한 간단한 도움말을 보여주는 옵션이다.
-v	Tibero의 버전 정보를 보여주는 옵션이다.
-C	Tibero에서 사용 가능한 character set 정보와 nls_date_lang 정보를 보여주는 옵션이다.
-c	Tibero가 replication mode로 설정되어 있을 경우 replication mode를 사용하지 않는 옵션이다.
-t	Tibero 서버를 기동할 수 있는 옵션이다. 이 옵션은 생략이 가능하다.

부팅명령어 : tbboot  
다운명령어 : tbdwnA

Tibero Instance 기동



## 다운모드 옵션

다운모드	설명	
NORMAL	모든 세션이 끝날때까지 기다린 이후 종료하는 모드이다.	
POST_TX	모든 트랜잭션이 끝날 때까지 기다린 이후 티베로를 종료하는 모드이다.	
IMMEDIATE	현재 수행 중인 모든 작업을 강제로 종료하고 트랜잭션을 전부 종료하며 끝내는 모드이다.	
ABORT	tibero의 프로세스를 강제로 종료하는 모드이다. (정상적인 상태에선 사용되지 않으며) 장애 발생시 강제로 종료할때 사용된다.	

## 기동종료.

- tbdownd clean

옵션	설명	
clean	티베로 서버가 종료된 상태에서 운영중에 사용하였던 공유 메모리나 세마포어 자원들을 해제하는 옵션 티베로 서버가 운영중일때 사용할 수 없는 옵션이다.	

kill 명령으로 종료한 경우, tbdownd clean 를 실행

만약 Tibero 서버가 Kill 과 같은 시스템 내부 명령어에 의해서 비정상적으로 종료된 경우, 운영 중에 사용하였던 공유 메모리나 세마포어 자원들이 해제가 안 될 수 있다. 이런 경우 Tibero 가 정상적으로 재기동 하지않게 되므로, tbdownd clean 으로 기존 자원을 해제시킨 후 tboot 를 실행.

### BOOT\_WITH\_AUTO\_DOWN\_CLEAN 파라미터

비정상적으로 종료되더라도 초기화 파라미터 BOOT\_WITH\_AUTO\_DOWN\_CLEAN 으로 이전 운영 중에 사용하였던 자원을 해제시켜 부팅 가능.

관리자가 서버의 비정상 종료 상황을 제대로 인지하지 못하고 서버 운영을 할 나 프로세스가 제대로 정리가 안 되는 예외적이 상황이 발생하여 충돌이 날

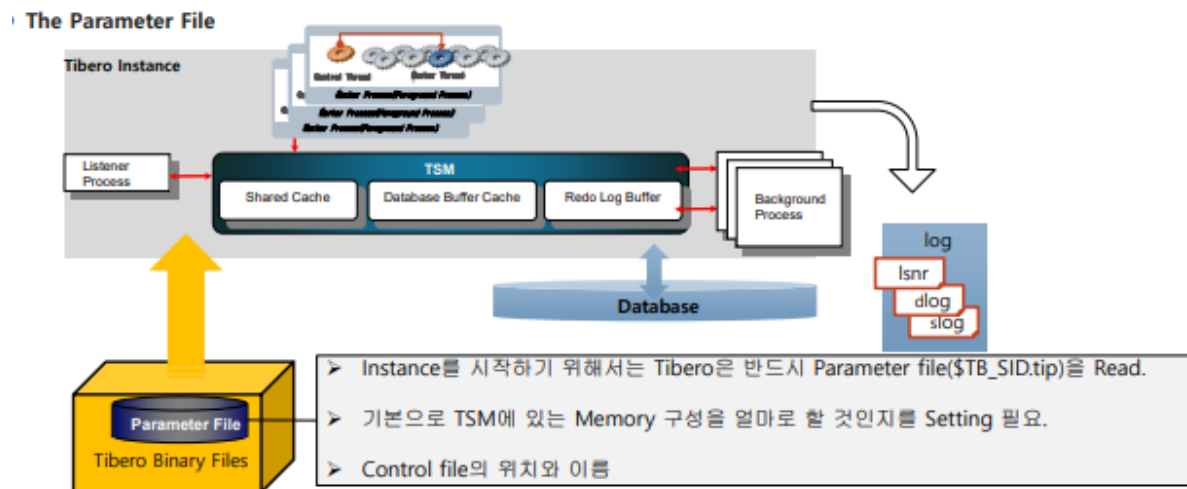
- tbdownd POST\_TX: 모든 트랜잭션이 끝나는 상황일때 기다리고 나서 티베로를 종료하는 모드이다.



## tbdown immediate

- 현재 수행중인 모든 작업을 강제로 중단시키며, 진행중인 모든 트랜잭션을 롤백하고 티베로를 종료하는 모드이다.

## Parameter File



인스턴스를 시작하기 위해서 티베로는 파라미터파일 **\$TB\_SID**을 리드  
기본적으로 **TSM**에 있는 메모리를 얼마나 사용할건지를 세팅한다.

## 컨트롤 파일

### Controlfile

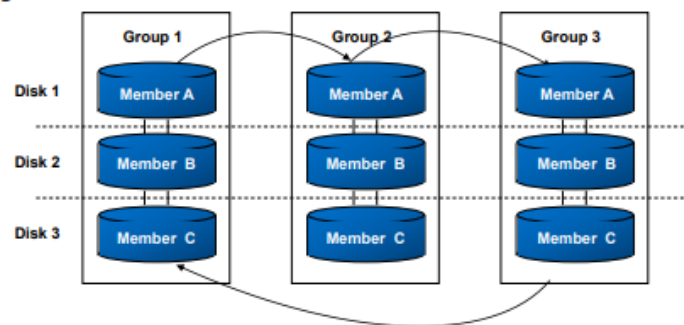
- 데이터베이스 이름과 데이터베이스의 고유 **식별자 (DBID)**
- 데이터베이스 생성시각
- 데이터파일, 온라인 **REDO**로그, 아카이브 **REDO**로그 정보
- 테이블스페이스 정보
- **RMGR** 백업 정보

- Control File은 작은 Binary File로 Database의 구조.
- 모든 Database Files & log files들은 Control file에서 식별 가능.

- Database name은 Control file에 저장.
- Control File은 Mount, Open시에 필요.
- Recovery시 필요한 동기화된 정보를 저장.
- 최소 다른 Disk상에 2개 이상의 Control Files를 가지도록 권장.
- Control File은 Database가 Open되어 졌을 때 반드시 Tibero Server가 Writing 가능하도록 설정.

## Redo log

### ● Multiplexed Redo Log Files



- Redo log File을 구성하는데 1 Group에 2개 이상의 Member를 동일 Machine상에서 다른 Disk에 분리하여 구성하기 권장.
- Multiplexed Redo Log Files는 특정 그룹에서 하나의 파일을 손실 시에 특이 사항 없음.
- 그룹의 모든 Member들은 동일한 정보와 Size 를 갖음

- Redo log File을 구성하는데 1 Group에 2개 이상의 Member를 동일 Machine상에서 다른 Disk에 분리하여 구성하기 권장.

-관리

## ● Archived Redo Log Files

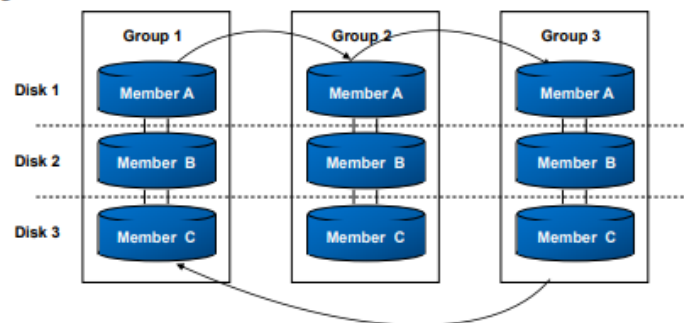


- 아카이브 리두 로그 파일은 온라인 리두로그 그룹 멤버의 복사본임
- 아카이브 리두 로그 파일은 데이터베이스 파일에 속하지는 않지만 복구를 위해 사용되는 중요한 파일임
- 아카이브 리두 로그의 용도
  - 데이터베이스 백업 복구
  - TSC( Tiberio Standby Cluster) 의 Standby DB 의 데이터 업데이트
  - Tmax 의 데이터복제 솔루션인 ProSync 에서 사용

온라인을 카피해서 아카이브 리두로그 파일이 만들어진다.

- 아카이브 리두로그 파일은 온라인 리두로그 그룹멤버의 복사본이다.
- 복사가 안될시 모든 동작을 멈추게 된다.

## ● Multiplexed Redo Log Files



- Redo log File을 구성하는데 1 Group에 2개 이상의 Member를 동일 Machine상에서 다른 Disk에 분리하여 구성하기 권장.
- Multiplexed Redo Log Files는 특정 그룹에서 하나의 파일을 손실 시에 특이 사항 없음.
- 그룹의 모든 Member들은 동일한 정보와 Size 를 갖음

## 테이블 스페이스

- 하나 이상의 논리적 저장 영역으로 구성되어 있어 티베로의 모든 데이터를 저장한다.
- 각 테이블 스페이스는 물리적인 저장 영역인 데이터파일이라는 하나 이상의 파일로 구성이 된다.
- TABLESPACE 관리 지침
  - Using Multiple Tablespaces
    - Data Dictionary Data 와 유저 Data의 분리
    - 어플리케이션별 분리

- I/O 경합을 줄이기 위하여 데이터 파일의 분산(디스크)
- 유저 Data 와 Rollback Segment의 분리
- 개별 테이블스페이스를 오프라인 할 수 있도록 할 것.
- 업데이트가 많은것, 읽기 전용, 임시용 등 특별한 형태별 지정
- 개별 테이블스페이스의 백업 고려

## 테이블 스페이스 유형

구성요소	설명	
SYSTEM TABLESPACE	데이터베이스 작업을 위해 항상 있어야함(온라인 상태유지) 포함 내용	
NON SYSTEM TABLESPACE	응용프로그램 데이터/인덱스 임시 세그먼트 롤백 세그먼트	

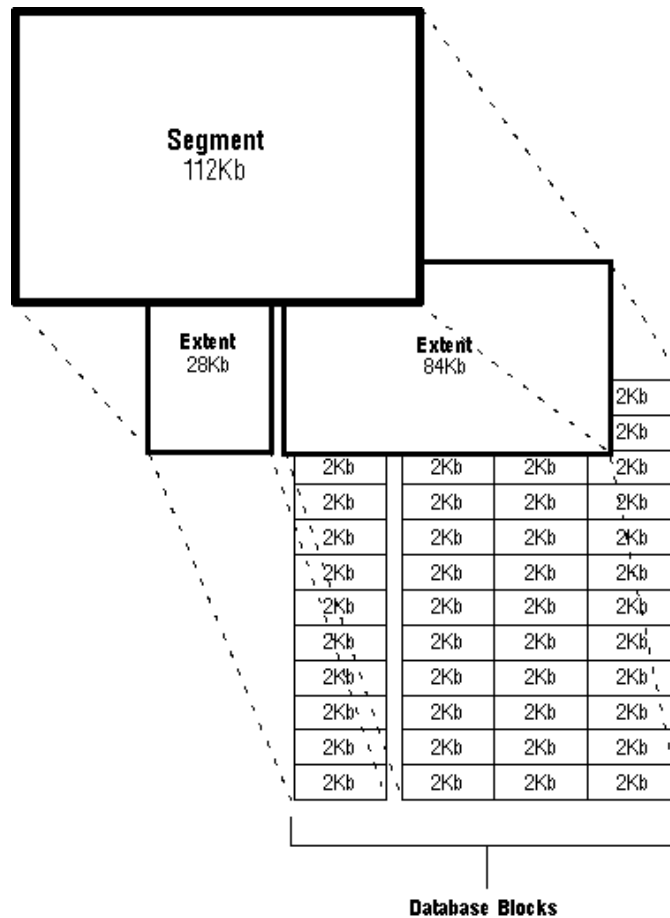
## 세그먼트 개념

- 세그먼트는 일반적으로 디스크 공간을 차지하며, 하나 이상의 데이터베이스 객체(테이블, 인덱스 등)를 포함할 수 있습니다. 세그먼트는 보통 데이터베이스 관리 시스템(DBMS)에 의해 관리되며, 데이터베이스 객체들을 저장하고 접근할 수 있는 물리적인 메커니즘을 제공합니다.
- 하나의 세그먼트는 다수의 데이터 파일에 나누어 저장될 수도 있다. 단 그 데이터 파일들이 같은 테이블스페이스 내부에 할당되어 있는 경우에만 가능하다.

## 테이블 스페이스 개념

- 테이블 스페이스란 테이블,인덱스 프로시저,뷰 등의 객체들을 저장해놓은 오라클의 논리적인 공간을 말한다.
- 테이블 스페이스(Tablespace)는 데이터베이스에서 데이터를 물리적으로 저장하는 단위입니다.
- 데이터베이스가 사용하는 공간(객체)을 할당하는 것이다.

대부분의 관계형 데이터베이스 관리 시스템(RDBMS)에서 사용되며, 테이블 스페이스는 하나 이상의 데이터 파일로 구성되어 있습니다. 이러한 파일들은 실제로는 디스크 상에 있는 파일이지만, 데이터베이스에서는 테이블, 인덱스, 뷰, 등의 데이터베이스 객체들을 저장하는 공간으로 보입니다.



각 테이블 스페이스는 데이터베이스 관리 시스템에 따라 다르지만 일반적으로 여러 개의 세그먼트(Segment)로 구성되어 있습니다. 세그먼트는 여러 개의 익스텐트(Extent)로 구성되어 있고, 익스텐트는 다시 여러 개의 데이터 블록으로 구성됩니다.

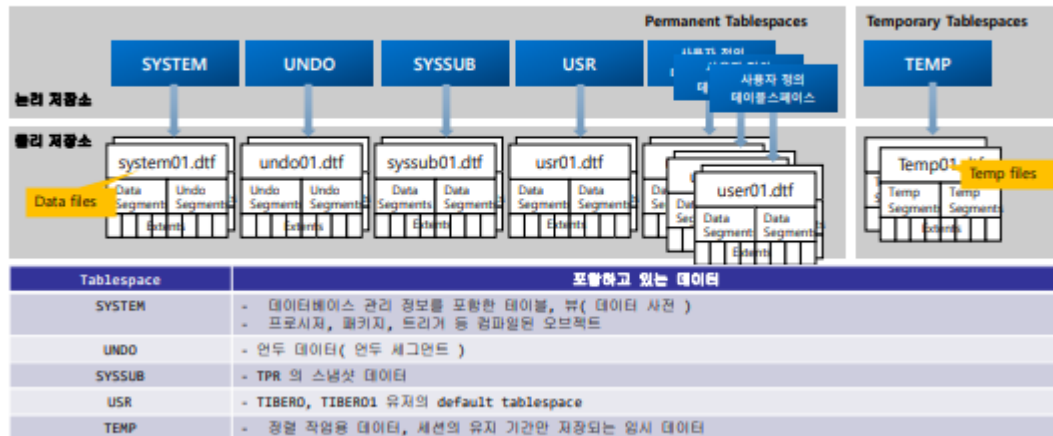
테이블 스페이스를 사용하는 주된 이유 중 하나는 데이터 관리를 용이하게 하기 위해서입니다. 예를 들어, 특정 애플리케이션에 관련된 데이터를 독립적으로 관리하려면 해당 애플리케이션의 데이터를 포함하는 테이블 스페이스를 생성하면 됩니다. 이렇게 하면 데이터베이스 관리 작업(예: 백업, 복구 등)을 효율적으로 수행할 수 있습니다.

## 테이블 스페이스 와 데이터파일 관리

- 테이블 스페이스는 세그먼트를 저장하는 논리 저장소, 한개 이상의 데이터파일이나 임시파일을 사용하여 데이터를 저장함.

## ● 테이블스페이스

- 테이블스페이스는 세그먼트를 저장하는 논리 저장소, 한개 이상의 데이터 파일이나 임시파일을 사용하여 데이터를 저장함



© 2022 TmaxTibero Co., Ltd. All Rights Reserved.

TmaxTibero

## 티베로 사용자관리

### ● 사용자 관리

- Tibero가 설치되면 아래와 같은 사용자 계정이 자동으로 생성됨
- 데이터베이스 관리자는 기본 계정 이외에 업무에 필요한 사용자 계정을 추가 생성하고, 관리하여야 함

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

## 사용자 계정

- 내부 데이터 접근하기 위해선 계정이 필요하다.
- 스키마가 먼저 존재해야지만 테이블을 생성하는게 가능하다.

## 프로파일

- 데이터베이스 프로파일은 사용자 또는 사용자 그룹이 데이터베이스 시스템에서 사용할 수 있는 리소스 및 특권을 정의하는 설정의 모음입니다. 이를 통해 데이터베이스 관리자는 사용자에게 적절한 권한을 부여하고, 시스템 리소스를 효율적으로 관리할 수 있습니다.

## 권한

- 데이터베이스나 데이터베이스 오브젝트에 접근하여 sql문은 실행하기 위한 귀니
- 권한: 스키마 오브젝트 권한, SYSTEM 권한

- 특권을 부여하는 예

```
SQL> conn Peter/abcdef ... ① ... Connected.  
SQL> CREATE TABLE EMPLOYEE  
      (ID NUMBER,  
       EMPLOYEE_NAME VARCHAR(20),  
       ADDRESS VARCHAR(50)); ... ② ... Created.  
SQL> GRANT SELECT ON EMPLOYEE TO Smith; ... ③ ... Granted.
```

- ① Peter 라는 사용자 계정으로 데이터베이스에 접속한다.
- ② CREATE TABLE 문을 사용하여 EMPLOYEE 테이블을 생성한다. 총 3개의 컬럼(ID, EMPLOYEE\_NAME, ADDRESS)을 생성한다.
- ③ Smith 라는 사용자가 Peter 사용자가 생성한 EMPLOYEE 테이블에 GRANT 명령을 실행하여 SELECT 특권을 부여한다.

이제 사용자 Smith는 Peter의 EMPLOYEE 테이블을 조회할 수 있다.

## 스키마 오브젝트 권한(Schema object privilege)

- 오브젝트에 대해 실행할 수 있는 명령어 사용을 위한 권한

```
SQL> GRANT SELECT, UPDATE ( PROD_NAME , PROD_DATE ) ON PRODUCT  
TO tiberio WITH GRANT OPTION;
```

권한	설명	
alter	오브젝트 특성을 변경	
delete	테이블에서 row를 삭제	
ExCUTE	함수,프로시저를 실행.	
index	테이블에 대한 인덱스를 생성	
insert	table에 row를 삽입	
select	table에 대하여 질의	
update	테이블의 row를 갱신	

## 롤 (role)

권한의 집합

바구니처럼 가지고 있다가 부여하는 개념.

작업에 맞는 권한이 필요하다.

-롤 부여

```
SQL> GRANT SELECT, UPDATE ( PROD_NAME , PROD_DATE ) ON PRODUCT  
TO tiberio WITH GRANT OPTION;
```

```
SQL> CREATE ROLE APP_USER2;  
SQL> GRANT SELECT, INSERT ON tiberio.prduct TO APP_USER2
```

```
SQL> GRANT APP_USER TO tiberio;
```

```
QL> GRANT APP_USER2 TO tiberio
```

WITH Admin opTION

- 롤을 부여 받은 사용자가 부여 받은 롤을 다른 사용자에게 다시 부여가 가능하다.
- 롤취소

```
SQL> GRANT APP_USER TO tiberio WITH ADMIN OPTION;
```

```
SQL> REVOKE APP_USER FROM tiberio;
```

미리 정의된 롤



롤	설명
CONNECT	•CREATE SESSION
RESOURCE	•CREATE PROCEDURE •CREATE SEQUENCE •CREATE TABLE •CREATE TRIGGER
DBA	•모든 시스템 권한이 WITH ADMIN OPTION으로 부여됨

#### ● SET ROLE 명령

- 동적으로 변경 가능 롤 변경 가능

```
SET ROLE APP_USER, TMAX;      /* APP_USER, TMAX 롤을 컴 */
SET ROLE ALL EXCEPT TMAX;   /* TMAX 롤 제외한 모든 롤을 컴 */
SET ROLE ALL;                 /* 모든 롤을 컴 */
SET ROLE NONE;                /* 모든 롤을 끄 */
```

예제\_ROLL조회\_사용자 생성 및 권한 부여.sql