

ALTIBASE ADMINISTRATION II

CONTENTS

- BACKUP & RECOVERY
- UTILITIES
- NEW FEATURE

ALTIBASE ADMINISTRATION II

BACKUP & RECOVERY

BACKUP & RECOVERY

❖ CONTENTS

- CHECKPOINT
- FULL BACKUP
- RECOVERY
- INCREMENTAL BACKUP & RECOVERY
- SNAPSHOT BACKUP
- FLASHBACK & PURGE

BACKUP & RECOVERY

CHECKPOINT

CHECKPOINT

❖ 체크포인트(CHECKPOINT)

- 데이터베이스에 대한 변경 및 현재 상황을 데이터 파일에 저장
- 체크포인트 완료 시의 처리 과정
 - 기존 복구시작 시점을 완료시점으로 변경
 - 데이터 파일에 반영이 완료된 불필요한 리두로그 파일 삭제
- 주기적인 체크포인트는 시스템 장애 시 데이터베이스 복구시간 단축

❖ ALTIBASE 체크포인트의 특징

- fuzzy & ping-pong 체크포인트
 - 체크포인트 수행 중에도 성능저하 없이 TRANSACTION을 처리 가능
 - 메모리 테이블스페이스에 한하여 데이터 파일을 두벌로 유지

CHECKPOINT

❖ 체크포인트 수행

➤ 자동수행

- 프로퍼티 CHECKPOINT_ENABLED(default 1)에 의해 자동수행 여부 설정
 - ◆ 시간주기 : CHECKPOINT_INTERVAL_IN_SEC(default 6000)
 - ◆ 리두로그 파일개수 : CHECKPOINT_INTERVAL_IN_LOG(default 100)

➤ 수동수행

- 사용자명령 - 사용자 "SYS"로만 실행이 가능한 명시적인 체크포인트 명령어

```
iSQL> ALTER SYSTEM CHECKPOINT;
```

❖ 체크포인트 수행 완료되어도 리두 로그 파일이 삭제되지 않는 경우

- 완료되지 않은 TRANSACTION의 리두 로그 파일인 경우
- 아카이브 로그 모드에서, 아카이브 되지 않은 리두 로그 파일인 경우
- 이중화로 전송되지 않은 리두 로그 파일인 경우

BACKUP & RECOVERY

FULL BACKUP

FULL BACKUP

❖ 데이터베이스 모드(DATABASE MODE)

- 아카이브 로그파일을 관리하는 방식에 따라 두 가지로 분류
 - 아카이브로그 모드 (archivelog mode)
 - ◆ 지정된 경로로 축적
 - 노아카이브로그 모드 (noarchivelog mode)
 - ◆ 지정된 경로로 축적하지 않음
- 데이터베이스 생성구문(CREATE DATABASE)에 의해 최초 지정
 - 데이터베이스 생성 이후에도 CONTROL 단계에서 변경은 가능하나 재 구동 필요

❖ 데이터베이스 모드의 비교

데이터베이스 모드	특징	고려사항
아카이브로그 모드 (archivelog mode)	<ul style="list-style-type: none">✓ 매체복구(media recovery) 가능✓ 아카이브로그 파일의 손실이 없다면 현재시점까지 복구 가능	<ul style="list-style-type: none">✓ 아카이브로그 파일을 저장하기 위한 디스크 공간 필요✓ 아카이브로그 파일에 대한 DBA 관리 필요
노아카이브로그 모드 (noarchivelog mode)	<ul style="list-style-type: none">✓ 아카이브로그 파일을 저장하기 위한 디스크 공간 불필요✓ 아카이브로그 파일에 대한 DBA 관리 불필요	<ul style="list-style-type: none">✓ 매체복구 불가능✓ 오프라인 백업을 받은 시점까지만 복구 가능

FULL BACKUP

❖ 현재 데이터베이스 모드 확인

- 성능뷰 v\$archive 를 통하여 보다 상세한 정보 확인 가능

```
iSQL(sysdba)> SELECT archive_mode, archive_dest FROM V$ARCHIVE;  
ARCHIVE_MODE    ARCHIVE_DEST  
-----  
0                /home/alti2/altibase_home/arch_logs
```

❖ 데이터베이스 생성 이후 데이터베이스 모드 변경

- 데이터베이스 구동 종료 후 CONTROL 단계에서 데이터베이스 모드 변경
 - SYSDBA 자격의 사용자 "SYS"로만 수행 가능

```
iSQL> CONNECT sys/manager AS sysdba  
iSQL(sysdba)> SHUTDOWN IMMEDIATE;  
iSQL(sysdba)> STARTUP CONTROL;  
iSQL(sysdba)> ALTER DATABASE ARCHIVELOG;  
iSQL(sysdba)> STARTUP;
```

FULL BACKUP

❖ 데이터베이스 백업 종류와 대상 객체

백업 종류	대상 객체	복구시점	데이터베이스 모드
온라인 데이터베이스 백업	리두로그 파일을 제외한 데이터베이스 전체 (데이터 파일, 로그엔커 파일)	✓ 최신시점	✓ 아카이브로그 필수
온라인 테이블스페이스 백업	특정 테이블스페이스의 데이터 파일		
온라인 테이블스페이스 백업 by DBA			
온라인 리두로그엔커 백업	리두로그엔커 파일		
오프라인 백업	데이터베이스 전체 (데이터 파일, 로그엔커파일, 리두로그 파일)	✓ 백업시점	✓ 노아카이브로그 ✓ 아카이브로그
aexport, iloader 백업	스키마생성 스크립트, 특정 테이블의 데이터		

- 온라인 백업 관련 구문은 SYSDBA 자격의 사용자 "SYS"로만 수행 가능
 - SYSDBA 자격으로 접속 가능한 세션은 하나이므로 동시에 여러 세션에서 수행 불가

FULL BACKUP

❖ 온라인 데이터베이스 백업

- 데이터베이스 구동 중 특정 시점에 전체 데이터베이스 백업
- 구문수행 시 모든 데이터 파일과 로그엔커 파일이 지정한 경로로 자동백업
 - 메모리 테이블스페이스 데이터 파일은 시간 단축을 위해 0, 1 중 한 벌만 백업
 - 임시 테이블스페이스관련 데이터 파일은 쿼리 연산용이므로 백업하지 않음

```
iSQL(sysdba)> ALTER DATABASE BACKUP DATABASE TO 'backup_dir';
```

FULL BACKUP

❖ 온라인 테이블스페이스 백업

- 데이터베이스 구동 중 특정 테이블스페이스 백업
- 구문 수행 시 테이블스페이스와 관련된 데이터 파일이 지정한 경로로 자동 백업
 - 메모리 테이블스페이스 데이터 파일은 시간 단축을 위해 0, 1 중 한 벌만 백업

```
iSQL(sysdba)> ALTER DATABASE BACKUP TABLESPACE tablespace_name TO 'backup_dir';
```

FULL BACKUP

❖ 온라인 테이블스페이스 백업 by DBA

- 데이터베이스 구동 중 특정 테이블스페이스 백업
- 구문 수행으로 백업 시작과 종료 시점을 통제 후 OS copy 명령어를 사용하여 테이블스페이스 관련 데이터 파일 수동 복사
- 관련 리두로그 파일을 강제로 아카이빙 하기 위해 로그 스위치 수행 절차 필요
- 동시에 여러 테이블스페이스를 백업 가능하므로 3rd party 백업 솔루션과 연동 가능

1. iSQL(sysdba)> ALTER TABLESPACE *tablespace_name* BEGIN BACKUP;
2. OS copy 명령어로 테이블스페이스 관련 데이터 파일 복사
3. iSQL(sysdba)> ALTER TABLESPACE *tablespace_name* END BACKUP;
4. iSQL(sysdba)> ALTER SYSTEM SWITCH LOGFILE;

FULL BACKUP

❖ 온라인 로그앵커 백업

- 데이터베이스 구동 중 로그앵커 파일 백업
- 구문수행 시 로그앵커 파일 3개 모두 지정한 경로로 자동 백업
- 테이블스페이스 변경(ALTER) 시 복구를 대비하여 변경된 테이블스페이스, 디렉터리 테이블스페이스와 함께 백업

```
iSQL(sysdba)> ALTER DATABASE BACKUP LOGANCHOR TO 'backup_dir';  
iSQL(sysdba)> ALTER DATABASE BACKUP TABLESPACE tablespace_name TO 'backup_dir';  
iSQL(sysdba)> ALTER DATABASE BACKUP TABLESPACE SYS_TBS_MEM_DIC TO 'backup_dir';
```

FULL BACKUP

❖ 오프라인 데이터베이스 백업

- 데이터베이스 종료 후 데이터베이스 전체 백업
- 정상종료확인 후 OS 명령어를 사용하여 데이터 파일, 리두로그 파일, 로그앵커 파일 복사

1. `iSQL(sysdba)> shutdown immediate;`
2. OS copy 명령어로 데이터 파일, 리두로그 파일, 로그앵커 파일 복사
3. `iSQL(sysdba)> startup;`

FULL BACKUP

❖ 오프라인 데이터베이스 백업 예제

- 프로퍼티 파일(altibase.properties)의 설정 상태

```
MEM_DB_DIR = ?/mem_dbs  
DEFAULT_DISK_DB_DIR = ?/dbs  
LOG_DIR = ?/logs  
LOGANCHOR_DIR = ?/logs
```

- 수행 절차

- 테이블스페이스 관련 데이터 파일은 기본 경로에만 위치한 경우를 가정

```
iSQL(sysdba)> shutdown immediate;  
shell> cp -r $ALTIBASE_HOME/mem_dbs/* /backup  
shell> cp -r $ALTIBASE_HOME/dbs/* /backup  
shell> cp -r $ALTIBASE_HOME/logs /backup  
iSQL(sysdba)> startup;
```

FULL BACKUP

❖ 백업 관련 유의사항

- 오프라인 데이터베이스 백업 시 프로퍼티 파일상의 기본경로만 확인해서는 안됨
 - 데이터 파일의 위치가 기본 경로가 아닌 경우 누락될 수 있음
 - 데이터 파일 관련 성능뷰(v\$datafiles, v\$mem_tablespace_checkpoint_paths)를 통하여 누락 여부 확인 필요
- 온라인 백업과 체크포인트는 동시에 수행될 수 없음
 - 체크포인트 수행 중에 온라인 백업 명령을 수행하면 체크포인트 완료까지 대기
 - 온라인 백업 중에는 체크포인트가 수행되지 않으며 수행 시도 시 에러발생
[ERR-110A4 : Backup is in progress. Please wait until the current backup process completes.]
- 테이블스페이스 변경 시 복구를 대비하여 아래 백업 중 하나 선택 수행
 - 해당 테이블스페이스, 디렉터리 테이블스페이스, 로그앵커 파일 백업
 - 온라인/오프라인 데이터베이스 백업
- 이중화가 걸려있는 경우 이중화 정보도 같이 백업
 - 백업한 데이터베이스를 다른 시스템에서 복구 시 IP 변경으로 인해 이중화 문제 발생 가능

BACKUP & RECOVERY

RECOVERY

RECOVERY

❖ 자동복구(RESTART RECOVERY)

- 정전, 시스템 crash 같은 비정상 종료 후 재 구동 시 자동 수행되는 복구

```
.....  
TRANSITION TO PHASE : META  
[SM] Recovery Phase - 1 : Preparing Database  
                        : Dynamic Memory Version => Parallel Loading  
[SM] Recovery Phase - 2 : Loading Database  
[SM] Recovery Phase - 3 : Starting Recovery  
                        Initializing Active Transaction List  
                        Redo  
                        Refine Disk Table..  
                        Undo  
[SM] Refine Memory Table :  
.....  
..... [SUCCESS]
```

RECOVERY

❖ 매체복구(MEDIA RECOVERY)

- 데이터 파일이 유실되거나 손상되었을 때 아카이브로그 파일과 이전에 백업한 데이터 파일 및 로그앵커 파일을 사용하여 수동 수행하는 복구
- CONTROL 단계에서 구문으로 수행하는 오프라인 매체복구 (온라인 매체복구 불가)
- 아카이브로그 모드로 운영될 때에만 가능
- 완전복구와 불완전복구로 분류

RECOVERY

❖ 완전복구

- 구문을 수행하여 현재 시점까지 데이터베이스 복구

```
iSQL(sysdba)> ALTER DATABASE RECOVER DATABASE;
```

❖ 요구사항

- 데이터베이스 전체 백업 받은 시점 이후의 아카이브로그 파일은 물론 온라인 리두 로그 파일까지 모두 손상이 없어야 복구 가능
- 현재 시점의 최신 로그앵커 파일 필요

RECOVERY

❖ 불완전복구

- 구문을 수행하여 특정 시점으로 데이터베이스 복구
 - 온라인 리두로그 파일 또는 아카이브 로그 파일이 손상 or 유실된 경우

```
iSQL(sysdba)> ALTER DATABASE RECOVER DATABASE UNTIL CANCEL;
```

- 데이터베이스 전체를 특정시점으로 복구 하는 경우

```
iSQL(sysdba)> ALTER DATABASE RECOVER DATABASE UNTIL TIME '2019-07-10:17:55:00';
```

- 불완전복구 수행 후에는 온라인 리두로그를 초기화 해야만 함
 - "Meta 단계로 전이", "다음 재구동"시 자동복구 되지 않도록 하기 위한 필수 과정

```
iSQL(sysdba)> ALTER DATABASE db_name META RESETLOGS;
```

❖ 요구사항

- 복구하려는 시점에 대응하는 로그앵커 파일 필요

RECOVERY

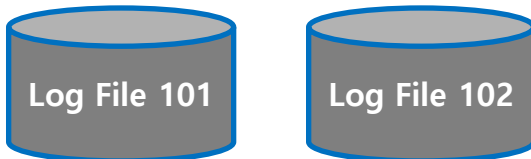
❖ 데이터 파일 유실 복구 예제 - 백업 받은 데이터 파일을 이용한 복구

- 백업 받은 데이터파일의 LSN(Log Sequence Number)부터 현재 시점의 LSN 사이의 Logfile 들을 사용

Log Anchor



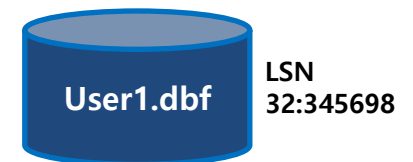
Online Logs 디렉토리



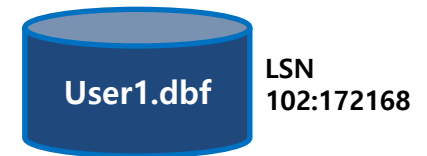
Archive Logs 디렉토리



백업한 User1.dbf



Recovery 구문 실행



RECOVERY

❖ 매체복구를 위한 빈 데이터 파일 생성

- 백업된 데이터 파일이 없는 경우 CONTROL 단계에서 구문 수행을 통해 생성
 - 메모리 테이블스페이스 관련 빈 데이터 파일 생성

```
iSQL(sysdba)> ALTER DATABASE CREATE CHECKPOINT IMAGE 'user_mem_data-1-0';
```

- 디스크 테이블스페이스 관련 빈 데이터 파일 생성

```
iSQL(sysdba)> ALTER DATABASE CREATE DATAFILE 'user_disk_data001.dbf';
```

❖ 요구사항

- 데이터 파일 경로가 로그엔커 파일에 기록된 경로와 반드시 일치

BACKUP & RECOVERY

INCREMENTAL BACKUP & RECOVERY

INCREMENTAL BACKUP & RECOVERY

❖ 증분 백업(INCREMENTAL BACKUP)

- 이전의 특정 백업 이후 변경된 페이지(Page)만 백업 받는 방법
 - 레벨 0 증분 백업 - 모든 페이지들을 백업
 - 레벨 1 증분 백업 - 증분 백업 이후 변경된 페이지만 백업
 - 차등 증분 백업: 레벨 1 이후 변경된 페이지를 백업
 - 누적 증분 백업: 레벨 0 이후 변경된 페이지를 백업

❖ 특징

- 아카이브 모드 필수
- 데이터베이스 구동 중 백업 수행
- 전체 백업에 종속적
- 데이터베이스, 테이블스페이스 단위로 백업 가능
- 백업 양 작고, 백업 소요 시간 짧음
- 증분 백업이 많을 수록(백업 파일이 많을 수록) 복구 소요 시간 증가

INCREMENTAL BACKUP & RECOVERY

❖ 변경 추적 기능

- Level 0 백업 이후 수정된 Data page(block)을 확인하기 위해 변경 발생시 Change tracking 파일에 기록
- 증분 백업에 필수 기능(비활성화 시 증분 백업 사용 불가)

❖ 변경 추적 파일(CHANGE TRACKING 파일)

- 변경된 페이지의 정보가 비트맵으로 저장
- 증분 백업을 수행하기 위해 반드시 필요
- \$ALTIBASE_HOME/dbs 디렉토리에 위치

❖ backupInfo 파일

- 증분 백업이 수행된 일시 순으로 백업 정보 저장
 - 레벨, 백업종류, 백업태그, 시작 일시, 완료 일시 및 백업 파일 위치
- 매체 복원(Media Restore)시 복원할 백업 파일 순서 정보 제공
- \$ALTIBASE_HOME/dbs 디렉토리에 위치

INCREMENTAL BACKUP & RECOVERY

❖ 변경 추적 기능 활성화

- ALTIBASE_HOME/dbs를 기본 경로로 사용

```
iSQL(sysdba)> ALTER DATABASE ENABLE[DISABLE] INCREMENTAL CHUNK CHANGE TRACKING;
```

- Change tracking 파일이 소실되거나 유효하지 않다면 SYS 사용자가 SYSDBA권한으로 Change tracking 을 활성화하는 SQL구문을 실행하여 재생성
- 활성화 시, ALTIBASE_HOME/dbs 에 변경 추적 파일과 backupinfo 파일이 생성
- 비활성화 시, ALTIBASE_HOME/dbs 에 변경 추적 파일은 삭제되며, 모든 구동 단계에서 sysdba 권한으로 수행 가능
- 변경추적파일과 backupinfo 파일 확인

```
[alti1@localhost dbs]$ ls -alt
total 489820
-rw-r----- 1 alti1 alti1          512 Aug 11 21:00 backupinfo
-rw-r----- 1 alti1 alti1 10486272 Aug 11 21:00 changeTracking
-rw-r----- 1 alti1 alti1 135274496 Aug 11 21:00 undo001.dbf
-rw-r----- 1 alti1 alti1 104865792 Aug 11 21:00 system001.dbf
```

INCREMENTAL BACKUP & RECOVERY

❖ 레벨0 증분 백업

- 데이터 파일의 모든 페이지들을 백업
 - 데이터베이스
 - 테이블스페이스
- 데이터베이스

```
iSQL(sysdba)> ALTER DATABASE BACKUP INCREMENTAL LEVEL 0 DATABASE;
```

- 데이터베이스 - TAG 지정

```
iSQL(sysdba)> ALTER DATABASE BACKUP INCREMENTAL LEVEL 0 DATABASE WITH TAG 'MONDAY';
```

- 테이블스페이스

```
iSQL(sysdba)> ALTER DATABASE BACKUP INCREMENTAL LEVEL 0 TABLESPACE SYS_TBS_MEM_DIC;
```

- 테이블스페이스 - TAG 지정

```
iSQL(sysdba)> ALTER DATABASE BACKUP INCREMENTAL LEVEL 0 TABLESPACE SYS_TBS_MEM_DIC WITH TAG 'MONDAY';
```

INCREMENTAL BACKUP & RECOVERY

❖ 레벨1 증분 백업

- 차등 증분 백업
 - 가장 최근에 수행된 레벨1 증분 백업 이후에 변경된 페이지 백업
 - 레벨1이 없을경우 레벨0 백업 이후로 변경된 페이지 백업
 - 복구 소요 시간 큼

```
iSQL(sysdba)> ALTER DATABASE BACKUP INCREMENTAL LEVEL 1 DATABASE WITH TAG 'SUN' ;
```

	일	월	화	수	목	금	토	일
Data	..a	b	c	d	e	f	g	..h
Full Backup								
Incremental Backup (Differential)								
Level	0	1	1	1	1	1	1	0
Backup Data	..a	b	c	d	e	f	g	..h

INCREMENTAL BACKUP & RECOVERY

❖ 레벨1 증분 백업

➤ 누적 증분 백업

- 가장 최근에 수행된 레벨0 백업 이후에 변경된 페이지 백업
 - 현재 level1이면 가장 최근의 level0 백업을 기준으로 백업 수행
- 복구 소요 시간 작음

```
iSQL(sysdba)> ALTER DATABASE BACKUP INCREMENTAL LEVEL 1 CUMULATIVE DATABASE WITH TAG 'SUN' ;
```

	일	월	화	수	목	금	토	일
Data	..a	b	c	d	e	f	g	..h
Full Backup	←							
Incremental Backup (Cumulative)	←							
	←							
	←							
	←							
	←							
	←							
	←							
Level	0	1C	1C	1C	1C	1C	1C	0
Backup Data	...a	b	bc	bcd	bcde	bcdef	bcdefg	...h

INCREMENTAL BACKUP & RECOVERY

❖ 증분 백업 종류와 대상 객체

	백업 레벨	백업 종류	설명	대상 객체	데이터베이스 모드
증분 백업	Level 0	전체백업	전체백업	데이터베이스, 테이블스페이스	아카이브로그
	Level 1	차등 증분 백업	✓가장 최근의 레벨1 증분 백업 이후 변경된 페이지 백업 ✓레벨1이 없을 경우, 레벨0 백업 이후 변경된 페이지 백업		
		누적 증분 백업	✓가장 최근의 레벨0 백업 이후 변경된 페이지 백업		

- SYSDBA 자격의 사용자 "SYS"로만 수행 가능
 - SYSDBA 자격으로 접속 가능한 세션이 하나이므로 동시에 여러 세션에서 수행 불가

INCREMENTAL BACKUP & RECOVERY

❖ 예제

- 변경 추적 기능 활성화

```
iSQL(sysdba)> ALTER DATABASE ENABLE INCREMENTAL CHUNK CHANGE TRACKING;
```

- 백업 경로 지정 또는 변경

```
iSQL(sysdba)> ALTER DATABASE CHANGE BACKUP DIRECTORY '$ALTIBASE_HOME/BACKUP';
```

- 레벨 0 전체 증분 백업 수행

```
iSQL(sysdba)> ALTER DATABASE BACKUP INCREMENTAL LEVEL 0 DATABASE;
```

- 레벨 1 차등 증분 백업 태그와 같이 수행

```
iSQL(sysdba)> ALTER DATABASE BACKUP INCREMENTAL LEVEL 1 DATABASE WITH TAG 'SUN';
```

INCREMENTAL BACKUP & RECOVERY

❖ 백업 경로 지정/변경

- 증분 백업 수행으로 생성되는 백업 파일들의 위치 지정

```
iSQL(sysdba)> ALTER DATABASE CHANGE BACKUP DIRECTORY '/backup_dir';
```

- 만약 처음에 지정한 경로에 디스크 공간이 부족하면, 위의 구문을 사용하여 새로운 백업 경로로 변경 가능
- 백업 파일을 이동하는데 시간이 오래 걸리거나, 생성되는 백업 파일들의 크기가 하나의 백업 디바이스에 유지할 수 없는 상황일 때에는 디스크 공간 관리를 위해 백업 경로 변경 권고

INCREMENTAL BACKUP & RECOVERY

❖ 백업 파일 이동

- ▶ 백업 경로의 디스크 공간이 부족한 경우, 백업 파일들을 다른 디바이스의 경로로 이동 가능
 - SQL 구문으로 backupInfo 파일 내에서 백업 파일 경로만 변경하고, 기존 백업 파일은 관리자가 복사 명령(cp)을 사용해서 수동으로 이동하는 방법

```
iSQL(sysdba)> ALTER DATABASE MOVE BACKUP FILE TO '/backup_dir2';
```

```
[alti1@localhost dbs]$ cp ... /backup_dir2
```

- SQL 구문으로 backupInfo 파일 내의 백업 파일 경로 변경과 백업 파일의 이동을 동시에 수행하는 방법

```
iSQL(sysdba)> ALTER DATABASE MOVE BACKUP FILE TO '/backup_dir2' WITH CONTENTS;
```

INCREMENTAL BACKUP & RECOVERY

❖ 백업 파일 삭제

- 유효 기간이 지난 백업 파일을 삭제하여 디스크의 여유 공간을 확보

```
iSQL(sysdba)> ALTER DATABASE DELETE OBSOLETE BACKUP FILES;
```

- 이 구문을 수행하면 V\$OBSOLETE_BACKUP_INFO 성능 뷰에 나타나는 백업 파일만 삭제
- V\$OBSOLETE_BACKUP_INFO 성능 뷰에서 아무 것도 조회되지 않는다면 삭제되는 파일 없음

INCREMENTAL BACKUP & RECOVERY

❖ 유의사항

- 노아카이브 로그 모드에서는 변경 추적 기능 구문 실행 불가
 - iSQL(sysdba)> ALTER DATABASE ENABLE INCREMENTAL CHUNK CHANGE TRACKING;
- 변경 추적 기능 비활성화 시 \$ALTIBASE_HOME/dbs 에서 change tracking 파일 삭제
- Backupinfo 파일이 존재하지 않으면, 백업 파일이 존재하더라도 복구 불가
- Change Tracking 파일 재생성 시, 전에 추적하여 변경된 페이지 정보 삭제

INCREMENTAL BACKUP & RECOVERY

❖ 매체 복원(MEDIA RESTORE)

- 매체(Media)에 장애가 발생하여 데이터베이스 파일이 소실된 경우 백업 파일을 복사해서 소실된 파일 대체
- 복원 후, 로그 파일을 이용하여 매체 복구 진행

❖ 매체 복원 종류

- 완전복원:
 - 가장 최근의 레벨0 증분 백업시점으로 복원이 되고 가장 최근의 누적 증분 백업, 차등 증분 백업이 된 시점의 순서로 복원

```
iSQL(sysdba)> ALTER DATABASE RESTORE DATABASE;
```

- 불완전복원
 - 태그를 지정하여 지정된 태그 이전에 수행된 가장 가까운 레벨0 증분 백업시점으로 복원이 되고 지정된 태그 시점이랑 가장 가까운 누적 증분 백업, 차등 증분 백업이 된 시점의 순서로 복원

```
iSQL(sysdba)> ALTER DATABASE RESTORE DATABASE FROM TAG 'tag_name' ;
```

INCREMENTAL BACKUP & RECOVERY

❖ 완전 복원 수행

- 증분 백업을 아래와 같이 먼저 수행

```
iSQL(sysdba)> ALTER DATABASE BACKUP INCREMENTAL LEVEL 0 DATABASE WITH TAG 'THURSDAY';  
iSQL(sysdba)> ALTER DATABASE BACKUP INCREMENTAL LEVEL 1 DATABASE WITH TAG 'FRIDAY';  
iSQL(sysdba)> ALTER DATABASE BACKUP INCREMENTAL LEVEL 1 CUMULATIVE DATABASE WITH TAG 'SATURDAY';  
iSQL(sysdba)> ALTER DATABASE BACKUP INCREMENTAL LEVEL 1 DATABASE WITH TAG 'SUNDAY';
```

- 완전 복구 구문 수행

```
iSQL(sysdba)> ALTER DATABASE RESTORE DATABASE;
```

- 가장 최근의 레벨 0 증분 백업 태그(THURSDAY)로부터 데이터 파일 복원
- 그런 다음 레벨 1 누적 증분 백업(태그 이름 SATURDAY)으로 복원되고, 마지막으로 레벨 1 차등 증분 백업(태그 이름 SUNDAY)으로 복원

INCREMENTAL BACKUP & RECOVERY

❖ 완전 복원 / 완전 복구

- 매체 복구를 이용하여 최근 시점까지 아카이브 로그 적용

```
iSQL(sysdba)> ALTER DATABASE RECOVER DATABASE ;
```

- 시스템 임시 테이블스페이스를 위한 파일은 백업이 되지 않기 때문에 수동으로 파일을 생성한 다음 서비스 시작

```
iSQL(sysdba)> ALTER DATATBASE CREATE DATAFILE ' temp001.dbf ' ;  
iSQL(sysdba)> STARTUP SERVICE;
```

INCREMENTAL BACKUP & RECOVERY

❖ 불완전 복구 수행

- 완전 복원 후 불완전 복구
- 불완전 복구에 필요한 loganchor와 backupinfo 파일로 복원

```
$ cp /backup_dir/TAG_SATURDAY/loganchor* $ALTIBASE_HOME/logs  
$ cp /backup_dir/TAG_SATURDAY/backupinfo $ALTIBASE_HOME/dbs
```

- 과거 시점의 loganchor 로 복원했기 때문에 change tracking 파일은 유효하지 않음 (PROCESS 단계에서 수행)

```
iSQL(sysdba)> ALTER DATABASE DISABLE INCREMENTAL CHUNK CHANGE TRACKING;
```

- 'SUNDAY'까지 데이터파일 완전 복원

```
iSQL(sysdba)> ALTER DATABASE RESTORE DATABASE;
```

- 완전 복원 후, 불완전 복구 수행

```
iSQL(sysdba)> ALTER DATABASE RECOVER DATABASE UNTIL CANCEL;
```

INCREMENTAL BACKUP & RECOVERY

- ▶ 시스템 임시 테이블스페이스 SYS_TBS_DISK_TEMP를 위한 파일은 백업이 되지 않기 때문에, 수동으로 파일을 생성한 다음 로그를 리셋 후 SERVER 시작

```
iSQL(sysdba)> ALTER DATABASE CREATE DATAFILE 'temp001.dbf';  
iSQL(sysdba)> ALTER DATABASE mydb META RESETLOGS;  
iSQL(sysdba)> STARTUP SERVICE;
```

INCREMENTAL BACKUP & RECOVERY

❖ 불완전 복원 후 완전 복구 수행

- 원하는 시점으로 데이터파일 불완전 복원

```
iSQL(sysdba)> ALTER DATABASE RESTORE DATABASE FROM TAG 'WEDNESDAY';
```

- 지정된 백업 태그 이전의 가장 최신 레벨 0 파일부터 복원하여, 백업 태그에 해당하는 레벨 1 파일들까지 순차적으로 복원

- 'WEDNESDAY'부터 최근 시점까지 아카이브 로그 적용

```
iSQL(sysdba)> ALTER DATABASE RECOVER DATABASE;
```

- 시스템 임시 테이블스페이스 SYS_TBS_DISK_TEMP를 위한 파일은 백업이 되지 않기 때문에, 수동으로 파일 생성 후 SERVER 시작

```
iSQL(sysdba)> ALTER DATABASE CREATE DATAFILE 'temp001.dbf';  
iSQL(sysdba)> STARTUP SERVICE;
```

INCREMENTAL BACKUP & RECOVERY

❖ 불완전 복원 후 불완전 복구

- 원하는 시점으로 데이터파일 불완전 복원

```
iSQL(sysdba)> ALTER DATABASE RESTORE DATABASE FROM TAG ' WEDNESDAY ' ;
```

- 불완전 복구에 필요한 loganchor, backupinfo 파일 복원

```
$ cp /backup_dir/TAG_WEDNESDAY/ loganchor* $ALTIBASE_HOME/logs  
$ cp /backup_dir/TAG_WEDNESDAY/ backupinfo $ALTIBASE_HOME/dbs
```

- 불완전 복구 수행

```
iSQL(sysdba)> ALTER DATATBASE RECOVER DATABASE UNTIL CANCEL ;
```

- Resetlog 를 수행 후 SERVER 시작

```
iSQL(sysdba)> ALTER DATABASE CREATE DATAFILE ' temp001.dbf ' ;  
iSQL(sysdba)> ALTER DATATBASE MYDB META RESETLOGS ;  
iSQL(sysdba)> STARTUP SERVICE ;
```

INCREMENTAL BACKUP & RECOVERY

❖ 유의사항

➤ Change tracking 파일

- 매체복원은 변경추적기능과 상관없이 실행이 가능 하지만 복원시간 단축을 위해서 아래의 구문을 PROCESS 단계에서 실행하여 SERVER가 더 이상 Change Tracking 파일을 검사하지 않도록 설정

```
iSQL(sysdba)> ALTER DATABASE DISABLE INCREMENTAL CHUNK CHANGE TRACKING;
```

➤ Backupinfo 파일

- 이 파일은 매체 복원에 반드시 필요
- backupInfo 파일은 증분 백업을 수행할 때 자동으로 백업
- 가장 최근에 수행된 증분 백업 경로에서 copy 명령어를 이용하여 backupInfo 파일을 복원

```
alti1@localhost dbs]$ cp /backup_dir/BACKUP_TAG/backupInfo $ALTIBASE_HOME/dbs
```

➤ 복원(Restore) 수행 후, 복구(Recover)도 수행 필요

➤ TAG 를 안 사용할 경우, TAG 이름은 날짜와 시간으로 자동 설정(Default)

➤ 증분 백업을 사용하여 매체 복원을 할 때에는 로그 파일이 사용되지 않기 때문에 ALTER DATABASE RESTORE DATABASE UNTIL CANCEL 구문 미지원(UNTIL TIME 가능)

BACKUP & RECOVERY

SNAPSHOT BACKUP

SNAPSHOT BACKUP

❖ 스냅샷 백업(SNAPSHOT BACKUP)

- 스냅샷 백업은 데이터베이스의 특정 시점을 SCN 으로 스냅샷 지정 후, iLoader 유틸리티를 사용하여 데이터 백업
- 스냅샷 지정 및 해제는 SYSDBA 권한을 가진 DBA 만 가능
- Foreign Key나 Trigger 가 걸려 있는 테이블의 경우 서비스 중 iloader를 통해 백업을 받게 되면 각각 테이블의 일관성 보장 안됨

❖ 백업 구문

```
* Snapshot 시작 및 SCN 지정
iSQL(sysdba)> alter database begin snapshot;
Alter success.

* Snapshot 종료
iSQL(sysdba)> alter database end snapshot;
Alter success.
```


SNAPSHOT BACKUP

❖ SNAPSHOT 관련 프로퍼티

프로퍼티	설 명	기본값
SNAPSHOT_MEM_THRESHOLD	현재 까지 사용된 메모리 테이블스페이스 사용량(%)이 해당 값 이상이면 SNAPSHOT 중단	80
SNAPSHOT_DISK_UNDO_THRESHOLD	현재 까지 사용된 언두 테이블스페이스 사용량(%)이 해당 값 이상이면 SNAPSHOT 중단	80

❖ V\$SNAPSHOT(성능 뷰)

Column	설 명
SCN	스냅샷에 설정된 스냅샷 SCN 값
BEGIN_TIME	스냅샷 설정 시 UNIX_TIME
BEGIN_MEM_USAGE	스냅샷 설정 시 메모리 테이블스페이스 사용률
BEGIN_DISK_UNDO_USAGE	스냅샷 설정 시 디스크 테이블스페이스 사용률
CURRENT_TIME	현재 시간 (Unix Time)
CURRENT_MEM_USAGE	현재 메모리 테이블스페이스 사용률
CURRENT_DISK_UNDO_USAGE	현재 디스크 언두 테이블스페이스 사용률

SNAPSHOT BACKUP

❖ 유의 사항

- Snapshot SCN 을 지정하고 나면 지정된 SCN 이후로 Data를 삭제하지 않기 때문에 메모리 / 디스크 언두 테이블스페이스 사용량 증가 가능성
- 메모리 / 디스크 언두 테이블스페이스 FULL 방지를 위한 프로퍼티 설정 필요
 - 해당 프로퍼티 설정 값 이상일 경우 스냅샷 백업 자동 종료
 - 해당 에러 발생시 altibase_boot.log 에 기록
- 스냅샷 설정 이후 생성된 테이블은 iloader를 통한 Data out 불가
- 스냅샷 설정 이후 기존 테이블에 DDL이 실행되면 iloader를 통한 Data out 불가

SNAPSHOT BACKUP

❖ 메모리 테이블 SNAPSHOT 백업 테스트 케이스

➤ 테이블 생성 후 임의의 값 INSERT

```
iSQL> CRAETE TABLE t1 ( c1 char(100));  
Create success.  
iSQL> INSERT INTO t1 SELECT LEVERL FROM DUAL CONNECT BY LEVEL <= 100000;  
100000 rows inserted.
```

➤ Snapshot 시작 및 altibase_boot.log 확인

```
iSQL(sysdba)> ALTER DATABASE BEGIN SNAPSHOT;  
Alter success.  
iSQL(sysdba)> SELECT * FROM V$SNAPSHOT;  
SCN : 643511 → Begin Snapshot 직후이므로 SCN 값 확인 가능  
BEGIN_TIME : 1554790588  
BEGIN_MEM_USAGE : 7  
BEGIN_DISK_UNDO_USAGE : 6  
CURRENT_TIME : 1554796985  
CURRENT_MEM_USAGE : 7  
CURRENT_DISK_UNDO_USAGE : 6  
1 row selected.  
  
-- altibase_boot.log  
[2019/04/09 15:16:28 7FF][PID:9864][Thread-140206437345024][LWP-9889]  
BEGIN SNAPSHOT [TID:1527441] [SCN:643511]
```

SNAPSHOT BACKUP

❖ 메모리 테이블 SNAPSHOT 백업 테스트 케이스

- 임의의 값을 추가로 INSERT (5만 건)

```
iSQL(sysdba)> INSERT INTO t1 SELECT LEVEL FROM DUAL CONNECT BY LEVEL <=50000;  
50000 rows inserted.
```

- iLoader을 이용한 DATA 백업

```
.....  
100000 record download(T1)  
  
Total 100000 record download(T1)  
  
→ Snapshot을 시작한 시점까지의 DATA만 Down이 가능하므로 100000건만 iloader out
```

SNAPSHOT BACKUP

❖ 메모리 테이블 SNAPSHOT 백업 테스트 케이스

➤ Snapshot 종료 및 altibase_boot.log

```
iSQL(sysdba)> ALTER DATABASE END SNAPSHOT;
```

```
Alter success.
```

```
iSQL(sysdba)> SELECT * FROM V$SNAPSHOT;
```

```
SCN                : 0
```

→ End snapshot 으로 인해 SCN 값 초기화

```
BEGIN_TIME         : 0
```

```
BEGIN_MEM_USAGE    : 0
```

```
BEGIN_DISK_UNDO_USAGE : 0
```

```
CURRENT_TIME       : 1554797091
```

```
CURRENT_MEM_USAGE  : 7
```

```
CURRENT_DISK_UNDO_USAGE : 6
```

```
[2019/04/09 17:04:47 970][PID:9864][Thread-140206437345024][LWP-9889]
```

```
END SNAPSHOT
```

BACKUP & RECOVERY

FLASHBACK & PURGE

FLASHBACK & PURGE

※ 6.5.1 버전부터 제공

❖ 개념

- DROP 구문으로 제거된 테이블을 휴지통으로 이동
- 휴지통의 테이블을 플래시백 구문으로 복원 or 퍼지 구문으로 완전 삭제
- DBMS_RECYCLEBIN 시스템 정의 저장 패키지 제공
 - PURGE_USER_RECYCLEBIN
 - PURGE_ALL_RECYCLEBIN
 - PURGE_TABLESPACE
 - PURGE_ORIGINAL_NAME

❖ 플래시백 및 퍼지 시작 및 종료

- 플래시백 및 퍼지 시작

```
iSQL> ALTER SYSTEM SET RECYCLEBIN_ENABLE = 1 ;  
Alter success.
```

- 플래시백 및 퍼지 종료

```
iSQL> ALTER SYSTEM SET RECYCLEBIN_ENABLE = 0 ;  
Alter success.
```

FLASHBACK & PURGE

❖ 플래시백 구문

```
FLASHBACK TABLE table_name  
TO BEFOR DROP  
[RENAME TO table_name];
```

❖ 플래시백 예제

```
iSQL> SELECT * FROM test_table;  
C1          C2  
-----  
1           1  
1 row selected.  
iSQL> DROP TABLE test_table;  
Drop success.  
iSQL> SELECT * FROM test_table;  
[ERR-31031 : Table or view was not found :]  
iSQL> FLASHBACK TABLE test_table TO BEFORE DROP;  
Flashback success.  
iSQL> SELECT * FROM test_table;  
C1          C2  
-----  
1           1  
1 row selected.
```


FLASHBACK & PURGE

❖ 퍼지 구문

```
PURGE TABLE [user_name] table_name;
```

❖ 퍼지 예제

```
iSQL> DROP TABLE test_table;
Drop success.
iSQL> SELECT * FROM SYSTEM_.SYS_RECYCLEBIN_;
USER_NAME           : SYS
TABLE_NAME          : TEST_TABLE549F350216D44E7D115219F85C36D0C2
ORIGINAL_TABLE_NAME : TEST_TABLE
TBS_NAME            : SYS_TBS_MEM_DATA
MEMORY_SIZE         : 32768
DISK_SIZE           : 0
DROPPED             : 10-JAN-2019
1 row selected.
iSQL> PURGE TABLE sys.test_table;
Purge success.
iSQL> SELECT * FROM SYSTEM_.SYS_RECYCLEBIN_;
No rows selected.
iSQL> SEELCT * FROM test_table;
[ERR-31031 : Table or view was not found :]
```

ALTIBASE ADMINISTRATION II

UTILITIES

❖ CONTENTS

- ALTIPROFILE
- ALTIMON
- ALTIERR
- DUMPTRC
- ILOADER
- AEXPORT
- MIGRATION CENTER
- DATACOMPJ
- JDBCADAPTER
- JOB SCHEDULER
- ORANGE for ALTIBASE

UTILITIES

ALTIPROFILE

ALTIPROFILE

❖ altiProfile

- \$ALTIBASE_HOME/bin 에 위치
- 프로파일링 수행 결과 파일은 \$ALTIBASE_HOME/trc/alti-xxx.prof 형태로 저장
- 프로파일링을 수행한 이후 생성된 결과물을 분석할 수 있게 변환
- 프로파일링으로 저장되는 정보
 - SQL를 수행한 CLIENT 정보
 - SQL 처리 통계 정보 (처리 시간, 실행 계획, 수행 누적 횟수, 바인딩 정보)

```
Shell::~/home/alti1> altiProfile alti-xxx-x.prof > res.txt
```

- 수행된 결과를 화면에 출력하기 때문에 위와 같이 redirection으로 저장

ALTIPROFILE

❖ altiProfile 사용을 위한 사전 수행 절차

- 수행 시간 확인을 위한 설정 변경 (iSQL 이용)

```
iSQL> ALTER SYSTEM SET TIMED_STATISTICS = 1;
```

- SQL 처리 통계를 파일에 저장하도록 프로파일링 개시 (iSQL 이용)

```
iSQL> ALTER SYSTEM SET QUERY_PROF_FLAG = 1;
```

- QUERY_PROF_FLAG 설정 값 정보
 - 아래값을 조합하여 기록할 정보를 선택
(예 1+2+4=7 로 설정하면 쿼리, BIND 파라미터, PLAN정보 저장)

값	이름	설명
0		기록하지 않음
1	STATEMENT	SQL문, 실행시간, 실행정보, 색인 및 디스크 접근 정보
2	BIND	SQL문이 실행될 때마다의 BIND 파라미터
4	PLAN	SQL문이 실행될 때마다의 실행계획
8	SESSION STAT	3초마다 세션정보 출력(V\$SESSION 정보)
16	SYSTEM STAT	3초마다 시스템 정보 출력(V\$SYSSTAT 정보)
32	MEMORY_STAT	3초마다 메모리 정보 출력(V\$MEMSTAT 정보)

ALTIPROFILE

❖ altiProfile 실행 결과

- \$ALTIBASE_HOME/trc/에 프로파일링 결과 누적 저장

```
Shell::~/home/alti1> ls -lrt $ALTIBASE_HOME/trc
-rw----- 1 alti1 alti1 3400 Jul 16 19:08 alti-1279274882-0.prof (파일명은 수행 시마다 변동)
```

- altiProfile을 수행 시 유의사항
 - 프로파일링 정보를 파일로 계속 저장하기 때문에 디스크 Full이 발생하지 않도록 유의
 - ◆ 중지 시에는 "ALTER SYSTEM SET QUERY_PROF_FLAG = 0;" 를 iSQL에서 반드시 실행
 - 실시간으로 파일에 분석정보를 기록하기 때문에 성능 저하 발생 가능

ALTIPROFILE

❖ altiProfile 실행 및 분석결과

- 생성된 alti-xxxx-y.prof 파일을 입력 파일명으로 지정
- 프로파일링 결과 파일은 사용자 해석을 위해 변환 필요

```
Shell::~/home/alti1/altibase_home/trc> altiProfile alti-127974882-0.prof > res.txt
```

- 생성된 분석파일 결과 예제

```
Shell::~/home/alti1/altibase_home/trc> cat res.txt
[STATEMENT] 2010/07/16 19:08:02 (57/5/0)
SQL
=> [alter SYSTEM set QUERY_PROF_FLAG = 1]
User Info
  User   ID   = 2
  Client PID   = 31764
  Client Type  = [CLI-64LE]
  Client AppInfo = [isql]
```

← 실행시간

← SQL

← 세션정보

ALTIPROFILE

❖ altiProfile 실행 (계속)

Elapsed Time			← 수행시간
Total =	0 sec	195 usec	
SoftP =	0 sec	86 usec	
Parse =	0 sec	0 usec	
Valid =	0 sec	0 usec	
Optim =	0 sec	0 usec	
Execu =	0 sec	58 usec	
Fetch =	0 sec	51 usec	
Query Execute Info			← 수행결과
EXECUTE Result =	1 (0:failure, 1:success)		
Optimizer Mode =	0		
Used Memory =	0		
SUCCESS SUM =	4		
FAILURE SUM =	0		
PROCESSED ROW =	0		

ALTIPROFILE

❖ altiProfile 실행 (계속)

Index Access Info	←	인덱스 비용
Memory Full Scan Count	=	1
Memory Index Scan Count	=	1
Disk Full Scan Count	=	0
Disk Index Scan Count	=	0
Disk Access Info	←	버퍼캐쉬/디스크 접근 비용
READ DATA PAGE	=	0
WRITE DATA PAGE	=	0
GET DATA PAGE	=	0
CREATE DATA PAGE	=	0
READ UNDO PAGE	=	0
WRITE UNDO PAGE	=	0
GET UNDO PAGE	=	0
CREATE UNDO PAGE	=	0

ALTIPROFILE

❖ altiProfile 실행 (계속)

```
[PLAN] 2010/07/21 15:36:56(82/2/620673)
```

← 실행계획

```
[-----
```

```
PROJECT ( COLUMN_COUNT: 2, TUPLE_SIZE: 12 )
```

```
SCAN ( TABLE: T1, INDEX: __SYS_IDX_ID_102, ACCESS: 1, SELF_ID: 2 )
```

```
-----
```

```
]
```

ALTIPROFILE

❖ altiProfile 옵션

- stat 옵션을 사용하여 실행된 SQL문에 대한 통계 정보 구축, 출력 가능

```
Shell::~/home/alti1> altiProfile -stat query alti-xxx-x.prof > res.txt
```

- 출력정보

통계정보	설명
COUNT	QUERY 가 실행된 횟수
AVG	QUERY 수행에 걸린 평균 시간 (microseconds)
TOTAL	QUERY 수행에 걸린 시간 합계 (microseconds)
MIN	QUERY 수행에 걸린 최소 시간 (microseconds)
MAX	QUERY 수행에 걸린 최대 시간 (microseconds)
SUCCESS	QUERY 실행에 성공한 횟수
FAIL	QUERY 실행에 실패한 횟수
QUERY	수행된 SQL문

ALTIPROFILE

❖ altiProfile 옵션 출력 결과

➤ stat 옵션을 사용한 출력 결과

```
Shell::~/home/alti1> altiProfile -stat query alti-xxx-x.prof > res.txt
### Processing [alti-xxx-x.prof]...
100% [=====]
### Writing CSV File [alti-prof-stat-xxx.csv]...
### Writing TEXT File [alti-prof-stat-xxx.txt]...
### Successfully done.
Shell::~/home/alti1> cat alti-prof-stat-xxx.txt
```

COUNT	AVG	TOTAL	MIN	MAX	SUCCESS	FAIL	QUERY
1	0.013431	0.013431	0.013431	0.013431	1	0	select count(*) from tb_test1
1	0.001931	0.001931	0.001931	0.001931	1	0	alter system set
query_prof_flag=7							
1	0.000378	0.000378	0.000378	0.000378	1	0	alter system set
timed_statistics=1							
1	0.000234	0.000234	0.000234	0.000234	1	0	select * from tb_test1

```
[TOTAL] Count(Query): 4 Avg(Time): 0.003993 Sum(Time): 0.0160
```

UTILITIES

ALTIMON

❖ ALTIMON 지원 OS

- ALTIMON은 OS 정보를 수집하기 위해 C언어로 작성된 PICL 라이브러리 사용
- PICL 라이브러리를 사용할 수 있는 운영체제

OS	CPU	Version	PICL Library
AIX	Ppc64	OS version 5.3, 6.1	Aix-ppc64-5.so
HP-UX	la64	IA64	Hpux-ia64-11.sl
LINUX	X86_64 ppc64	OS Version 2.6 Glibc 2.5이상	Linux-x64.so Linux-ppc64.so

- 유의사항
 - ALTIMON은 java 1.5 이상 동작
 - Java 버전은 PICL c 라이브러리의 비트 수와 일치하는 것 선택

❖ ALTIMON 설정

- ▶ ALTIMON을 사용하기 위해 \$ALTIBASE_HOME/altiMon/conf 디렉토리에 있는 아래 파일 설정 필요

파일명	설명
Config.xml	모니터링 대상인 ALTIBASE SERVER 접속 정보와 ALTIMON 제어 정보 설정
Metrics.xml	사전 정의된 OS Metric, SQL Metric과 사용자 정의된 OS Metric(Command Metric)을 설정하는 파일
GroupMetrics.xml	Group Metric을 정의하는 파일 OS Metric, Command Metric의 조합으로 이루어짐 Metrics.xml에 정의한 metric을 사용해서 GroupMetric을 정의 함 Group Metric으로 수집한 데이터는 일반 Metric 측정값이 *.log파일에 저장되는 것과 달리 CSV형식의 파일로 저장

❖ ALTIMON 구동 및 종료

➤ ALTIMON 구동 및 종료

- altimon.sh { start | stop }

```
[kck@r920 bin]$ ./altimon.sh start
Starting up...
INFO | 2019-02-12 11:09:29,540 | Loading Configuration from config.xml...
INFO | 2019-02-12 11:09:29,889 | Loading Altibase Information from Monitoring Target Database...
INFO | 2019-02-12 11:09:29,910 | Initializing PICL(Platform Information Collection Library)...
INFO | 2019-02-12 11:09:29,971 | Success to open JDBC Driver.
INFO | 2019-02-12 11:09:29,972 | Success to load JDBC Driver Class.
INFO | 2019-02-12 11:09:29,980 | Success to instantiate JDBC Driver.
INFO | 2019-02-12 11:09:30,103 | Success to establish a connection to the database.
INFO | 2019-02-12 11:09:30,111 | Success to establish a connection to the database.
INFO | 2019-02-12 11:09:30,112 | Initializing the OS metric (DISK_FREE_PERCENTAGE.disk1)...
INFO | 2019-02-12 11:09:30,112 | Initializing and testing the SQL metric (MEMSTAT_LIST)...
INFO | 2019-02-12 11:09:30,137 | Initializing and testing the SQL metric (DATAFILE_STAT_2)...
INFO | 2019-02-12 11:09:30,139 | Initializing and testing the SQL metric (BUFFER_STAT)...
INFO | 2019-02-12 11:09:30,141 | Initializing and testing the SQL metric (LONG_RUN_QUERY)...
INFO | 2019-02-12 11:09:30,143 | Initializing and testing the SQL metric (SESSION_EVENT)...
INFO | 2019-02-12 11:09:30,146 | Initializing and testing the SQL metric (STATEMENT_COUNT)...
INFO | 2019-02-12 11:09:30,148 | Initializing and testing the SQL metric (MEM_DATABASE_USAGE)...
INFO | 2019-02-12 11:09:30,151 | Initializing and testing the SQL metric (LOCK_DESC)...
INFO | 2019-02-12 11:09:30,166 | Initializing and testing the SQL metric (DISK_TBL_USAGE)...
INFO | 2019-02-12 11:09:30,168 | Initializing and testing the SQL metric (DB_EVENT)...
INFO | 2019-02-12 11:09:30,170 | Initializing and testing the SQL metric (MEM_GC)...
INFO | 2019-02-12 11:09:30,171 | Initializing and testing the SQL metric (MEM_TBL_USAGE)...
INFO | 2019-02-12 11:09:30,174 | Initializing and testing the SQL metric (DATAFILE_STAT)...
INFO | 2019-02-12 11:09:30,175 | Initializing and testing the SQL metric (SERVICE_THREAD_STATE)...
INFO | 2019-02-12 11:09:30,176 | Initializing and testing the SQL metric (SYSTEM_WAIT_CLASS)...
INFO | 2019-02-12 11:09:30,178 | Initializing and testing the SQL metric (SYSTEMSTAT_LIST)...
INFO | 2019-02-12 11:09:30,187 | Initializing and testing the SQL metric (TABLESPACE_USAGE)...
INFO | 2019-02-12 11:09:30,192 | Initializing and testing the SQL metric (SERVICE_THREAD_MODE)...
INFO | 2019-02-12 11:09:30,193 | Initializing and testing the SQL metric (MEMSTAT_SUM)...
INFO | 2019-02-12 11:09:30,207 | Initializing and testing the SQL metric (UTRANS_QUERY)...
INFO | 2019-02-12 11:09:30,209 | Initializing and testing the SQL metric (LOGFILE_COUNT)...
INFO | 2019-02-12 11:09:30,210 | Initializing and testing the SQL metric (SESSION_COUNT)...
INFO | 2019-02-12 11:09:30,211 | Initializing and testing the target of a GroupMetric (TOTAL_CPU_USER)...
INFO | 2019-02-12 11:09:30,212 | Initializing and testing the target of a GroupMetric (PROC_CPU_USER)...
INFO | 2019-02-12 11:09:30,212 | Initializing and testing the target of a GroupMetric (LOGFILE_COUNT)...
INFO | 2019-02-12 11:09:30,216 | Initializing csv file for the group metric (group1)...
INFO | 2019-02-12 11:09:30,216 | Initializing and testing the target of a GroupMetric (PROC_MEM_USED_PERCENTAGE)...
INFO | 2019-02-12 11:09:30,216 | Initializing and testing the target of a GroupMetric (MEM_TBL_USAGE)...
INFO | 2019-02-12 11:09:30,217 | Initializing csv file for the group metric (group2)...
INFO | 2019-02-12 11:09:30,275 | Job execution threads will use class loader of thread: Thread-1
INFO | 2019-02-12 11:09:30,302 | Initialized scheduler signaller of type: class org.quartz.core.schedulerSignalerImpl
INFO | 2019-02-12 11:09:30,302 | Quartz scheduler v.1.6.6 created.
INFO | 2019-02-12 11:09:30,306 | RAMJobstore initialized.
INFO | 2019-02-12 11:09:30,306 | Quartz scheduler 'defaultQuartzScheduler' initialized from default resource file in Quartz package
: 'quartz.properties'
INFO | 2019-02-12 11:09:30,306 | Quartz scheduler version: 1.6.6
INFO | 2019-02-12 11:09:30,316 | Scheduler defaultQuartzScheduler_$_NON_CLUSTERED started.
INFO | 2019-02-12 11:09:30,317 | Profile Job Scheduler started.
INFO | 2019-02-12 11:09:30,341 | --- STARTUP altimon SUCCESS ---

[kck@r920 bin]$ ./altimon.sh stop
Shutting down...
INFO | 2019-02-12 11:09:39,351 | Stopping Profile Job Scheduler...
INFO | 2019-02-12 11:09:39,351 | Scheduler defaultQuartzScheduler_$_NON_CLUSTERED paused.
INFO | 2019-02-12 11:09:39,354 | Profile Job Scheduler is on standby mode.
INFO | 2019-02-12 11:09:39,355 | Disconnecting from the database.
INFO | 2019-02-12 11:09:39,356 | Disconnecting from the database.
INFO | 2019-02-12 11:09:39,356 | --- SHUTDOWN altimon SUCCESS ---
```

❖ ALTIMON 출력항목

➤ Logs 디렉토리

파일명	설명
altimon.log	altiMon 데몬의 각종 로그(info, warn, error) 기록
alert.log	<Alert> 설정에 해당하는 데이터 기록
OsMetrics.log	모든 OS Metric들에 의해 수집된 데이터 기록
[SQL_Metric_Name].log	SQL Metric에 의해 수집된 데이터가 Metric별로 log파일 기록
[GroupMetric_Name].csv	Group Metric에 의해 수집된 데이터가 Metric별로 csv파일 기록
report.html	사용자가 설정한 Configuration을 html 형식으로 리포트

➤ Logs/archive 디렉토리

- Altimon.log, *.cvs를 제외한 모든 파일이 YYYY-MM-DD 디렉토리 형태로 백업
- <MaintenancePeriod> 에 설정된 유지기간이 지나면 매일 1:50 am에 삭제

➤ Logs/csv_backup 디렉토리

- Group Metric 설정을 변경할 경우 기존 csv 파일이 해당 디렉토리 백업

UTILITIES

ALTIERR

❖ altierr

- \$ALTIBASE_HOME/bin 에 위치
- ALTIBASE SERVER 에러코드에 대한 상세한 설명을 검색하여 출력
 - 에러 번호나 에러 메시지 패턴을 이용하여 검색 가능
- 파라미터

파라미터	설 명
-w	지정한 검색 패턴을 포함하는 에러 메시지 검색 검색 패턴을 포함하는 모든 에러 메시지 출력
-n	에러 번호를 사용하여 검색 에러 번호는 16진수 / 양의 정수 / 음의 정수 형태로 입력 가능 에러 코드와 정확히 일치하는 레코드만 검색 에러 번호를 사용하여 검색 할 경우 파라미터 (-n) 생략 가능

❖ altierr 설명

- ALTIBASE SERVER는 에러 상황 발생 시 해당 코드를 "ERR-(에러번호)" 형태로 altibase_boot.log에 출력
- 에러 번호는 16진수 값

```
Ex) For 'ERR-00015'  
$ altierr 0x00015  
$ altierr -w 00015  
$ altierr 21
```

- C/C++ Precompiler 나 ODBC로 작성된 애플리케이션의 경우 SQL 관련 기능 수행 시 ODBC함수의 리턴 코드에 에러 코드가 설정

```
EX) For -266286  
$ altierr -266286  
$ altierr 266286  
$ altierr 0x4102E
```

- altierr는 에러 메시지에 포함된 키워드를 이용한 검색 가능

```
$ altierr -w connect  
$ altierr -w "does not"
```

UTILITIES

DUMPTRC

DUMPTRC

❖ dumptrc

- \$ALTIBASE_HOME/bin 에 위치
- ALTIBASE 비정상 종료 될 때 \$ALTIBASE_HOME/trc 디렉토리에 기록되는 트레이스 로그 파일을 사용자가 식별할 수 있는 형태로 변환 출력
- 유의 사항
 - dumptrc 가 정상적으로 동작하려면, ALTIBASE 의 실행 파일의 버전과 dumptrc 버전이 동일해야 정확한 콜스택 정보 확인 가능
- 사용 예제

```
$ dumptrc -i server -i error
$ dumptrc -e error
$ dumptrc -c -i error -i server -i sm -n 20
$ dumptrc -p /home/djin/work/altidev4/trunk/altibase_home/trc -c -n 20
```

DUMPTRC

❖ dumptrc

➤ 파라미터

파라미터	설 명
-h	도움말 출력
-p	트레이스 로그 파일의 경로 지정 경로를 지정하지 않으면 기본적으로 \$ALTIBASE_HOME/trc 디렉토리의 로그 파일을 read
-c	altibase_error.log에 기록된 ALTIBASE 프로세스 콜스택을 사용자가 식별할 수 있는 함수명으로 변환하여 출력
-s	트레이스 로그에 기록된 콜스택만 출력
-a	트레이스 로그 파일의 전부를 정렬하여 출력
-i	지정하는 트레이스 로그 파일을 출력, 복수의 로그 파일이 출력 가능 '-e' 파라미터와 함께 사용 불가능
-e	지정한 트레이스 로그 파일을 제외한 모든 파일을 출력, 반복하여 복수의 로그 파일 제거 '-i' 파라미터와 함께 사용 불가능
-n	한 번에 출력할 로그의 개수를 지정, 1~127개의 로그 출력 가능 지정하지 않으면 10개의 로그가 출력
-f	트레이스 로그 파일이 기록될 때마다 추가되는 로그 메시지 출력
-v	dumptrc 버전 출력

UTILITIES

ILOADER

ILOADER

❖ iloader

- \$ALTIBASE_HOME/bin 에 위치
- 데이터베이스의 데이터를 CSV 또는, 텍스트 형태의 파일로 다운로드 및 업로드
- 특정 시점의 테이블 백업 용도

```
Shell::~/home/alti1> iloader -u sys -p manager -s 127.0.0.1 -port 20301  
formout -T t1 -f t1.fmt
```

```
-----  
Altibase Data Load/Download utility.  
Release Version 7.1.0.1.0  
Copyright 2000, ALTIBASE Corporation or its subsidiaries.  
All Rights Reserved.  
-----
```

```
ISQL_CONNECTION : TCP
```

ILOADER

❖ il 스크립트

- \$ALTIBASE_HOME/bin 에 위치
- iSQL처럼 iLoader의 입력옵션을 생략하고 쓸 수 있도록 제공하는 스크립트

```
Shell::~/home/alti1> cat $ALTIBASE_HOME/bin/il
#!/bin/sh

${ALTIBASE_HOME}/bin/iloader -S 127.0.0.1 -U SYS -P MANAGER $*
```

- 접속과 관련된 부분만 생략할 수 있도록 제공되는 스크립트

ILOADER

❖ iloader 실행 시 기능 옵션

기능 옵션	설명
formout	iloader 동작 시에 필요한 form 파일을 생성하도록 동작
in	iloader 동작 시에 지정된 데이터파일을 읽어 데이터를 SERVER로 전송하는 동작
out	iloader 동작 시에 대상 테이블에서 데이터를 읽어 지정된 파일로 저장하는 동작

➤ form파일이란?

- 테이블에서 데이터를 다운로드 및 업로드를 수행할 때 대상 컬럼과 날짜형 컬럼의 형식 및 데이터 문자셋 등을 정의한 파일 (필요 시 사용자가 편집 가능한 구조)

➤ 데이터파일이란?

- iloader를 통해 저장된 파일 또는, 사용자가 타DBMS나 엑셀등으로부터 iloader가 인식할 수 있는 형태로 데이터를 저장한 파일

ILOADER

❖ iloader 실행 시 입력 옵션(접속 및 동작관련)

입력 옵션	설명
-s	접속할 ALTIBASE IP 지정
-u	ALTIBASE 데이터베이스 사용자 계정 명 지정
-p	계정의 패스워드 지정
-port	ALTIBASE Listen Port번호 지정
-log	수행결과를 저장할 로그파일명 지정 (오류도 같이 기록)
-bad	수행 중 오류가 발생한 데이터를 별도의 파일로 저장해야 할 경우 지정
-mode	APPEND : 기존의 테이블에 추가하여 삽입(기본값) REPLACE : DELETE 구문을 이용해 기존 테이블의 데이터를 모두 지우고 데이터 삽입 TRUNCATE : TRUNCATE 구문을 이용해 기존 테이블의 데이터를 모두 지우고 새로 생성 (옵션 미사용 시 일반 INSERT와 동일하게 동작)
-replication	true/false : 이중화 환경에서 상대방으로의 반영 여부 지정 (True : iloader로 입력된 데이터가 이중화로 상대방에 전송) (False : iloader로 입력된 데이터가 이중화로 전송되지 않음)

ILOADER

❖ iloader 실행 시 입력 옵션(성능관련)

입력 옵션	설명
-commit	지정된 숫자만큼 SERVER에서 처리된 후 COMMIT (COMMIT으로 인한 I/O비용을 감소시키는 효과, 기본값=1000)
-array	지정된 숫자만큼 통신버퍼에 담은 후 SERVER로 데이터 전송 (통신비용을 감소 시키는 효과, 기본값=1)
-parallel	지정된 숫자만큼 스레드가 테이블을 적절한 개수로 나누어 데이터를 다운로드 또는, 하나의 파일을 적절하게 나누어 업로드 수행 ※ array 옵션과 같이 쓰는 것이 성능상 유리 (파일은 datafile.dat0, datafile.dat1 과 같은 형식으로 생성) (업로드 시 개별 파일에 대한 parallel 옵션으로 동작)

ILOADER

❖ iloader 실행 시 입력 옵션(데이터/파일 관련)

입력 옵션	설명
-t	컬럼 간 구분자 지정
-r	레코드간 구분자 지정
-f	formout 동작 시 저장 될 form 파일명 지정
-d	in/out 동작 시 사용 될(저장 될) 데이터 파일명 지정
-rule	데이터파일을 csv 파일형식으로 생성하고자 할 경우 지정
-errors	지정된 숫자만큼 오류가 발생하면 실행을 멈춤 (기본값:50)
-split	데이터 파일마다 저장할 레코드의 개수 지정 (파일명은 datafile.dat0, datafile.dat1 같은 형식으로 생성)
-T	작업 대상 테이블 명 지정

ILOADER

❖ iloader 사용 예 (테이블 명을 T1이라고 가정)

- 데이터를 다운로드 받는 경우
 - `iloader -u sys -p manager -s 127.0.0.1 -port 20301 formout -T T1 -f T1.fmt`
 - `iloader -u sys -p manager -s 127.0.0.1 -port 20301 out -f T1.fmt -d T1.dat`
- 데이터를 업로드 하는 경우
 - `iloader -u sys -p manager -s 127.0.0.1 -port 20301 in -f T1.fmt -d T1.dat`
 - ◆ 업로드의 경우, 테이블의 형식이 동일하다면 이전에 받은 form파일 재사용 가능
- 데이터파일을 여러 개로 나누어 다운로드
 - `iloader -u sys -p manager -s 127.0.0.1 -port 20301 out -f T1.fmt -d T1.dat -parallel 4 -array 50`
 - `iloader -u sys -p manager -s 127.0.0.1 -port 20301 out -f T1.fmt -d T1.dat -split 50000`
- CSV가 아닌 명시적인 구분자를 지정하여 다운로드
 - `iloader -u sys -p manager -s 127.0.0.1 -port 20301 in -f T1.fmt -d T1.dat -t "!"@$" -r "$^%n"`

❖ iloader FORM 파일 수정 예 (테이블 명을 T1이라고 가정)

- 지정된 컬럼만 다운받고자 할 경우
 - 지정된 컬럼을 제외한 삭제 대상 컬럼을 form파일에서 삭제
- 조건 절을 명시하여 조건에 해당하는 데이터만 다운받고자 할 경우
 - form파일 내에 "DOWNLOAD CONDITION" 구문 사용
 - ◆ `DOWNLOAD CONDITION "WHERE C1 > 1000"`
- 날짜형식을 변경하고자 할 경우
 - form파일 내에 "DATEFORM" 구문 사용
 - ◆ `DATEFORM YYYY/MM/DD HH:MI:SS`
- 문자셋 변경
 - form파일 내에 "DATA_NLS_USE" 구문 사용 (데이터 다운로드 시에만 유효)
 - ◆ `DATA_NLS_USE=MS949`

ILOADER

❖ iloader FORM 파일 수정 예제 (테이블 명을 T1이라고 가정)

```
Shell::~/home/alti1> il formout -T T1 -f t1.fmt
Shell::~/home/alti1> vi t1.fmt
table t1
{
  A integer;
  B char(10);
  C date;
}
DOWNLOAD CONDITION "WHERE a > 1000 ORDER BY a DESC"
DATEFORM YYYY/MM/DD HH:MI:SS
DATA_NLS_USE=MS949
```

ILOADER

❖ iloader 업로드 시 시퀀스 사용

- 데이터를 업로드 할 때 순차적인 번호를 부여하고자 할 경우 사용
- 사용할 시퀀스 객체는 미리 생성되어 있어야 함
- 데이터파일에는 시퀀스에 해당하는 부분이 없어도 업로드 가능

```
isQL> CREATE SEQUENCE seq1 START WITH 1 INCREMENT BY 1;
isQL> QUIT;

Shell::~/~/altibase> vi t1.fmt
SEQUENCE seq1 NUM          # 사용자가 직접 수정
table t1
{
    NUM integer;           # 위에서 정의한 seq1의 Alias명을 사용
    B integer;
}
```

ILOADER

❖ iloader 업로드 시 함수 사용

- 데이터를 업로드 할 때, 함수를 이용한 결과값으로 데이터를 입력할 경우 사용
- 내장함수/사용자 정의함수를 모두 사용 가능
- fmt 파일에 사용자가 추가해야 하며 변수 부분을 "?" 바인딩 처리

```
Shell::home/alti1> il formout -T T1 -f t1.fmt
Shell::home/alti1> vi t1.fmt
table t1
{
  A integer      "TRIM (?)"          # 사용자가 직접 함수명을 추가
  B varchar(20) "CONCAT ('$ ', ?)";
}
```

ILOADER

❖ iloader 사용 시 발생할 수 있는 오류 사항 (1)

➤ 실행 시 입력 인자의 잘못된 사용

- [ERR-91031 : -T option(Table name) is not used or precedence option is not correct]
- [ERR-9103B : Option (-k) is invalid.]
- [ERR-9102C : Input Command Parser Error]

```
Shell::~/home/alti1> iloader -s 127.0.0.1 -u sys -p manager -port 20301 -T t1 -f t1.fmt
```

```
-----  
Altibase Data Load/Download utility.
```

```
Release Version 7.1.0.1.0
```

```
Copyright 2000, ALTIBASE Corporation or its subsidiaries.
```

```
All Rights Reserved.  
-----
```

```
[ERR-9102C : Input Command Parser Error]
```

```
Use help. iloader> help
```

- (formout, out, in) 옵션이 입력되어 동작구분이 지정되어야 하나 누락된 경우

❖ iloader 사용 시 발생할 수 있는 오류 사항 (2)

- 올바르지 않은 구분자의 사용 (-log 옵션을 사용할 경우 확인 가능)
 - [ERR-9103D : Data Parsing Error (Column : A)]
- 이미 존재하는 데이터를 중복 삽입하는 경우
 - [ERR-11058 : The row already exists in a unique index.]
- 날짜형 데이터에 잘못된 데이터 형식 또는 데이터가 존재하는 경우
 - [ERR-21038 : Literals in the input do not match format string.]
 - [ERR-21033 : Date format picture ends before converting entire input string.]
- 형 변환이 불가능한 데이터로 업로드를 수행하려 할 경우
 - [ERR-21011 : Invalid literal]
 - [ERR-2100C : Conversion not applicable]
- 숫자형 컬럼에 허용범위를 넘는 데이터로 업로드를 수행하려 할 경우
 - [ERR-21010 : Value overflow]

❖ iloader 사용 시 발생할 수 있는 오류 사항 (사례-1)

➤ iloader 동작 시 unique violation 오류가 발생한 경우

```
Shell::~/home/alti1> il in -T T1 -f t1.fmt -d t1.dat -log t1.log ☒ 에러가 발생했을 경우
Shell::~/home/alti1> cat t1.log
<DataLoad>
TableName : T1
Start Time : Wed Jul 21 14:49:50 2010
Recode 1 : 117783^%2010/07/21 14:49:03
[ERR-11058 : The row already exists in a unique index.]
Recode 2 : 217783^%2010/07/21 14:49:05
[ERR-11058 : The row already exists in a unique index.]
End Time : Wed Jul 21 14:49:50 2010
Total Row Count : 2
Load Row Count : 0
Error Row Count : 2
```

ILOADER

❖ iloader 사용 시 발생할 수 있는 오류 사항 (사례-2)

➤ iloader 동작 시 잘못된 날짜형식으로 오류가 난 경우

```
Shell::~/home/alti1> cat t1.fmt
table T1
{
A integer;
B date;
}
DATEFORM YYYY/MM/DD HH:MI:SS
DATA_NLS_USE=US7ASCII
```

```
Shell::~/home/alti1> cat a.log
<DataLoad>
TableName : T1
Start Time : Wed Jul 21 14:55:13 2010
Recode 1 : 117783^%2010/07/21 14:49:033
[ERR-21033 : Date format picture ends before converting entire input string.]
```

**Form 파일의 날짜형식과
실제 데이터형식 불일치**

ILOADER

❖ iloader 사용 시 발생할 수 있는 오류 사항 (사례-3)

- iloader 동작시 잘못된 구분자 지정으로 오류가 발생한 경우

사용자는 ^를 컬럼 구분자로 설정하고자 했으나
데이터 값에 존재하는 값이었거나 어떤 실수로 구분자가 변형된 상태

(TABLE T1: A, B, C 3개의 컬럼으로 구성된 경우 아래와 같이 데이터파일 존재)

```
1 ^_ 2010/07/21 14:49:03  
2 ^Hello^2010/07/21 14:49:05
```

```
Shell::~/home/alti1> cat a.log
```

```
<DataLoad>
```

```
TableName : T1
```

```
Start Time : Wed Jul 21 15:00:58 2010
```

```
Recode 1 :
```

```
[ERR-9103D : Data Parsing Error (Column : B)]
```

데이터에 구분자가
포함된 경우

UTILITIES

AEXPORT

AEXPORT

❖ aexport

- \$ALTIBASE_HOME/bin 에 위치
- ALTIBASE의 논리적인 구조 및 데이터를 텍스트로 저장하고, 이를 다시 새로운 데이터베이스 로 로딩하기 위한 스크립트 자동 생성
 - SQL 스크립트로 변환하여 저장하고, 모든 데이터를 텍스트로 내려 받기 때문에 상이한 ALTIBASE 버전간, 상이한 플랫폼 간의 데이터 이동 시 유용
- aexport 가 추출할 수 있는 데이터베이스 객체 및 구성 요소
 - 데이터베이스 사용자
 - 사용자 권한
 - 롤 (Role)
 - 테이블스페이스
 - 테이블
 - 테이블 제약 조건
 - 인덱스
 - 뷰
 - Materialized View
 - 저장 프로시저
 - 이중화 객체

AEXPORT

❖ aexport

➤ aexport 실행할 때 생성되는 SQL 스크립트 파일

SQL 파일명	설 명
SYS_CRT_DIR.sql	모든 디렉토리 객체 생성
SYS_CRT_USER.sql	모든 사용자와 롤(Role) 생성
SYS_CRT_SYNONYM.sql	모든 시노님(Synonym) 객체 생성
SYS_CRT_REP.sql	모든 이중화 객체 생성
ALL_CRT_VIEW_PROC.sql	모든 뷰와 프로시저 생성
ALL_CRT_TBS.sql	모든 테이블스페이스 생성
ALL_CRT_TBL.sql	모든 사용자 테이블 생성
ALL_CRT_INDEX.sql	모든 사용자 인덱스 생성
ALL_CRT_FK.sql	모든 사용자의 외래 키 생성
ALL_CRT_TRIG.sql	모든 사용자의 트리거 생성
ALL_CRT_SEQ.sql	모든 사용자의 시퀀스 생성
ALL_CRT_LINK.sql	모든 사용자의 데이터베이스 Link 생성
ALL_EXE_STATS.sql	모든 사용자의 테이블, 칼럼, 인덱스의 통계 정보 생성
ALL_REFRESH_MVIEW.sql	모든 사용자의 Materialized View를 리프레쉬
ALL_ALT_TBL.sql	모든 사용자의 테이블과 파티션에 대한 접근 모드를 변경

AEXPORT

❖ aexport

- ▶ aexport 수행 시 생성되는 쉘 스크립트 파일

스크립트	설 명
run_il_in.sh	데이터 로드 스크립트
run_il_out.sh	데이터 다운로드 스크립트
run_is.sh	스키마 생성 스크립트
run_is_con.sh	constraint 생성 스크립트 TWO_PHASE_SCRIPT 프로퍼티를 ON으로 설정하면, constraint 쉘 스크립트 파일 생성 이 파일은 인덱스, 외래 키, 트리거, 및 이중화 객체를 생성하는 SQL 스크립트 포함
run_is_fk.sh	외래키와 트리거 생성 스크립트
run_is_index.sh	인덱스 생성 스크립트
run_is_repl.sh	이중화 생성 스크립트
run_is_refresh_mview.sh	Materialized View를 refresh하는 스크립트
run_is_alt_tbl.sh	테이블과 파티션에 대한 접근 모드를 변경하는 스크립트

- ▶ TWO_PHASE_SCRIPT 프로퍼티를 ON으로 설정하면, index / repl / mview / alt_tbl
관련 쉘 스크립트 생성 불가

AEXPORT

❖ aexport

➤ aexport 파라미터

파라미터	설 명
-h	도움말 출력
-s	데이터를 다운로드할 SERVER의 호스트 명 또는 IP 주소 설정 생략시 호스트 명 입력 프롬프트 출력
-u	접속할 ALTIBASE 사용자명 설정, 생략 시 사용자명 입력 프롬프트 출력 전체 데이터베이스 모드로 실행하려면, 이 옵션에 SYS 사용자 지정
-p	접속할 ALTIBASE 사용자의 패스워드 설정 생략시 패스워드 입력 프롬프트 출력
-port	접속할 ALTIBASE 포트번호 설정 생략 시 환경 변수 ALTIBASE_PORT_NO, altibase.properties를 차례로 참조하며 설정되어 있지 않은 경우에는 포트 번호 입력 프롬프트 출력
-object	추출할 객체를 소유자 이름과 함께 명시
-tserver	추출한 데이터를 업로드 할 대상 SERVER 지정 이 정보는 aexport가 생성하는 쉘 스크립트 파일 안에 사용 -s 옵션과 마찬가지로 호스트 이름, IPv4 주소 또는 IPv6 주소 모두 가능
-tport	접속할 대상 SERVER의 포트번호 설정 이 정보는 aexport가 생성하는 쉘 스크립트 파일 안에 사용
-nls_use	데이터베이스에 데이터 저장 또는 데이터 추출 시에 사용되는 CLIENT 캐릭터 셋 지정
-prefer_ipv6	-s 옵션으로 호스트 이름을 입력했을 때, 접속할 IP 주소의 버전을 결정하는 옵션 이 옵션을 명시하면, 호스트 이름을 IPv6 주소로 바꾸어 접속

AEXPORT

❖ aexport 프로퍼티

- aexport.properties 파일의 프로퍼티 값 조정 필요
 - 이 파일은 \$ALTIBASE_HOME/conf 디렉토리에 위치

```
#####  
# Execute  
#####  
OPERATION = OUT # IN  
EXECUTE = OFF # ON  
  
#####  
# Script Option  
#####  
INVALID_SCRIPT = OFF # ON  
TWO_PHASE_SCRIPT = OFF # ON  
  
#####  
# Schema  
#####  
INDEX = ON # OFF  
DROP = OFF # ON  
#USER_PASSWORD = altibase  
VIEW_FORCE = OFF # ON
```

```
#####  
# Script File Name  
#####  
ISQL = run_is.sh  
ISQL_CON = run_is_con.sh  
ISQL_INDEX = run_is_index.sh  
ILOADER_OUT = run_il_out.sh  
ILOADER_IN = run_il_in.sh  
ISQL_FOREIGN_KEY = run_is_fk.sh  
ISQL_REPL = run_is_repl.sh  
  
#####  
# iloader option  
#####  
#ILOADER_FIELD_TERM = ^  
#ILOADER_ROW_TERM = %n  
ILOADER_PARTITION = OFF #ON
```

- ILOADER_FIELD_TERM = field_term
 - 테이블 데이터를 텍스트로 내릴 때 사용할 필드 구분자 설정
- ILOADER_ROW_TERM = row_term
 - 테이블 데이터를 텍스트로 내릴 때 사용할 레코드 구분자 설정

❖ aexport 사용 예제

➤ 전체 데이터베이스 모드로 실행

```
$ aexport -s 127.0.0.1 -u sys -p manager
```

```
-----  
Altibase Export Script Utility.  
Release Version 7.1.0.1.0  
Copyright 2000, ALTIBASE Corporation or its subsidiaries.  
All Rights Reserved.  
-----
```

```
##### TBS #####  
##### USER #####  
##### SYNONYM #####  
##### DIRECTORY #####  
##### TABLE #####  
##### QUEUE #####  
##### SEQUENCE #####  
##### DATABASE LINK #####  
##### VIEW #####  
##### MATERIALIZED VIEW #####  
##### STORED PROCEDURE #####  
##### STORED PACKAGE #####  
##### TRIGGER #####  
##### LIBRARY #####  
##### REPLICATION #####  
##### JOB #####
```

사용자 모드로 실행

```
$ aexport -s 127.0.0.1 -u user1 -p user1
```

```
-----  
Altibase Export Script Utility.  
Release Version 7.1.0.1.0  
Copyright 2000, ALTIBASE Corporation or its subsidiaries.  
All Rights Reserved.  
-----
```

```
##### USER #####  
##### SYNONYM #####  
##### TABLE #####  
##### QUEUE #####  
##### SEQUENCE #####  
##### DATABASE LINK #####  
##### VIEW #####  
##### MATERIALIZED VIEW #####  
##### STORED PROCEDURE #####  
##### STORED PACKAGE #####  
##### TRIGGER #####  
##### LIBRARY #####
```


❖ 유의사항

- SYS 사용자가 아닌 일반 사용자로 aexport 를 실행할 경우 해당 사용자가 생성한 스키마에 대해서만 스크립트 생성
- SYS 사용자가 아닌 일반 사용자로 aexport를 실행할 경우 이중화 객체 생성 불가
- 일반 사용자로 aexport를 실행할 경우 테이블 생성 권한 필요
 - aexport는 객체간 의존성 분석을 위해 임시 테이블을 생성하기 때문
- 동시에 여러 개의 aexport 프로세스를 실행 불가
 - aexport는 생성된 SQL 스크립트를 저장하기 위해 임시 테이블을 사용하기 때문에, 동시에 두 개 이상의 aexport 프로세스를 실행하면 비정상적인 결과 도출 가능

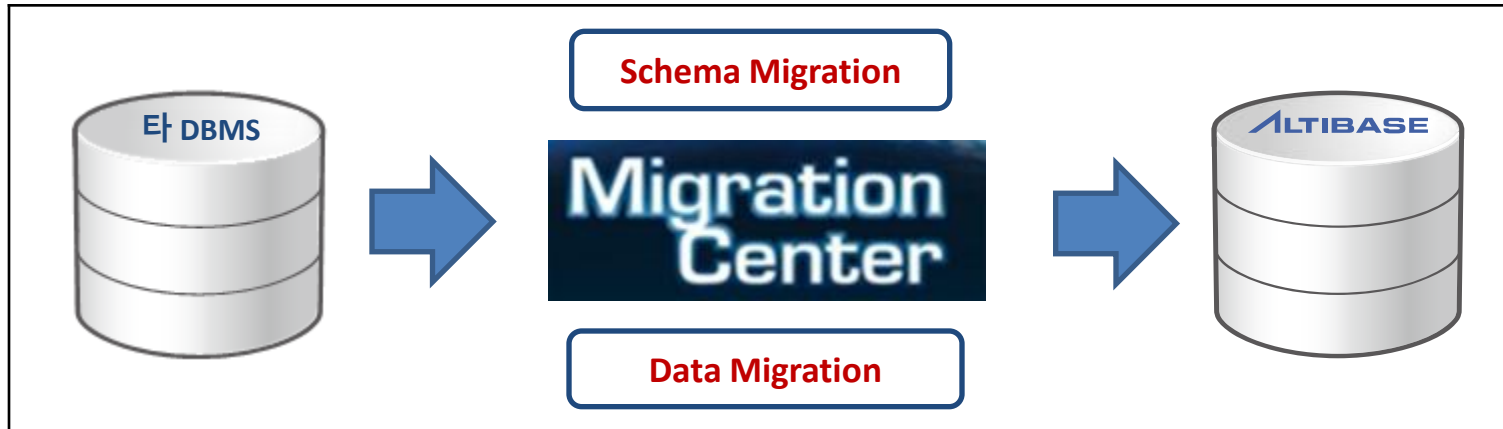
UTILITIES

MIGRATION CENTER

MIGRATION CENTER

❖ MIGRATION CENTER

- 이기종 데이터베이스 간의 데이터 Migration 수행
- 주요이점
 - 원본 데이터베이스의 객체들을 ALTIBASE로 마이그레이션 용이
 - 데이터를 외부파일로 내보내어(export), iLoader로 ALTIBASE에 가져올(import) 때 사용
 - 마이그레이션 과정을 단축시킬 수 있는 병렬 수행 옵션 제공
 - 이기종 데이터베이스 간에 다른 데이터 타입들의 데이터 타입 매핑을 제공하며 유연성을 위해 사용자 정의 타입 매핑 기능도 제공



MIGRATION CENTER

❖ 시스템 요구사항

- 하드웨어 요구사항
 - CPU : 800Mhz 펜티엄 III 이상
 - 메인메모리 : 512MB 이상
 - 디스크 : 150MB 이상의 여유공간
 - 화면 해상도 : 1024 x 768 화소 이상 (GUI모드 경우)
- 소프트웨어 요구사항
 - Oracle 또는 IBM Java 5 이상의 JRE
- 호환 가능한 데이터베이스 시스템
 - ALTIBASE : v 5.5.1 이상 버전
 - Oracle 데이터베이스 : 10g 이상 버전
 - Microsoft SQL SERVER : 2005 이상 버전
 - Oracle MySQL : 5.0 이상 버전

MIGRATION CENTER

❖ 설치 및 제거

➤ 다운로드

- ALTIBASE 고객지원서비스 포털 (<http://support.altibase.com>) 에서 다운로드
- Zip 또는 tar.gz 파일 형태로 제공되며 이 파일은 실행파일과 JDBC드라이버 포함

➤ 설치

- Migration Center 설치 파일의 압축 해제(디렉토리 변경 가능)

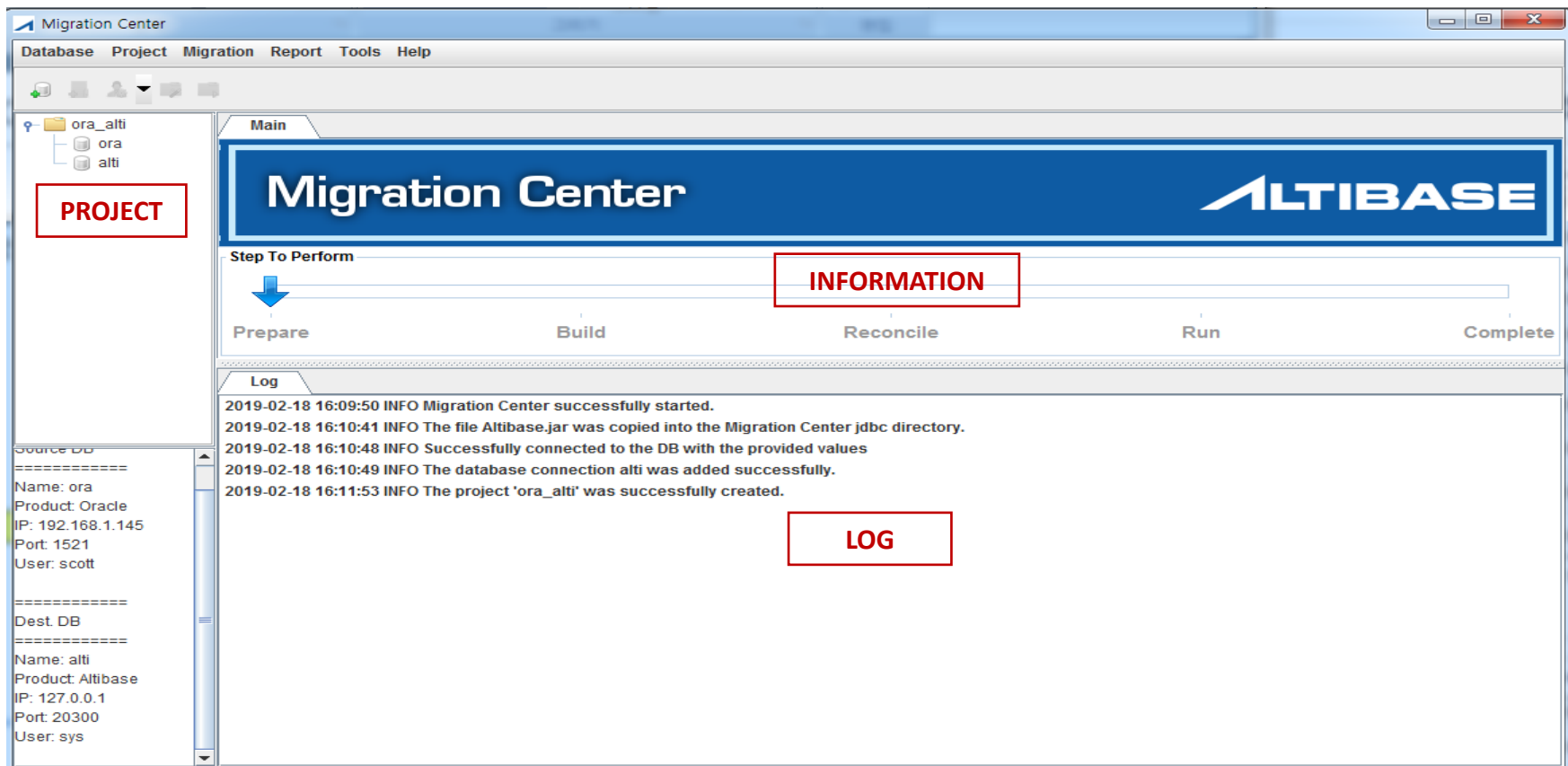
➤ 제거

- Migration Center 설치 디렉토리 삭제

MIGRATION CENTER

❖ GUI 모드

- Migration Center GUI는 "프로젝트(Project)", "정보(Information)", "로그(Log)" 3개의 창으로 구성



MIGRATION CENTER

❖ MIGRATION CENTER 시작

- 윈도우 환경에서는 "migcenter.bat" 아이콘을 더블 클릭해서 실행하고, 유닉스 계열의 운영체제에서는 "migcenter.sh" 실행

❖ 데이터베이스 연결 등록

- "Database" 메뉴 아래의 "Add Database Connection" 메뉴항목을 선택하거나 도구 모음에서 "Add Database Connection" 아이콘 클릭

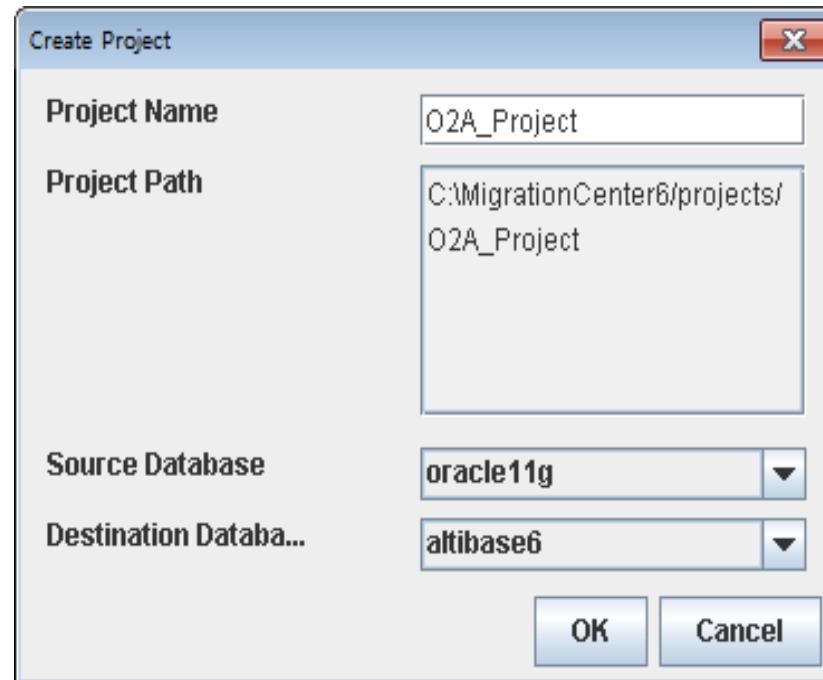
The screenshot shows the 'Add Database Connection' dialog box for Oracle. The 'DB Product' is set to 'Oracle'. The 'Connection Name' is 'Oracle'. The 'IP' is '127.0.0.1', 'Port' is '1521', 'User' is 'scott', and 'Password' is masked with dots. The 'SID' is 'orac1' and 'IP Version' is 'IPv4'. The 'URL' field contains 'jdbc:oracle:thin:@//127.0.0.1:1521/orac1'. There are 'Test', 'OK', and 'Cancel' buttons at the bottom.

The screenshot shows the 'Add Database Connection' dialog box for Altibase. The 'DB Product' is set to 'Altibase'. The 'Connection Name' is 'ALTI'. The 'IP' is '127.0.0.1', 'Port' is '20300', 'User' is 'sys', and 'Password' is masked with dots. The 'JDBC Driver' is './jdbc/Altibase.jar' with a 'File' button. The 'DB Name' is 'mydb'. The 'Option' section includes 'Encoding', 'IP Version' (set to 'IPv4'), and 'Property'. The 'URL' field contains 'jdbc:Altibase://127.0.0.1:20300/mydb'. There are 'Test', 'OK', and 'Cancel' buttons at the bottom.

MIGRATION CENTER

❖ 프로젝트 생성

- "Project" 메뉴 아래의 "Create Project" 메뉴 항목을 선택하거나, 도구 모음에서 "Create Project" 아이콘 클릭



Create Project

Project Name: O2A_Project

Project Path: C:\MigrationCenter6/projects/O2A_Project

Source Database: oracle11g

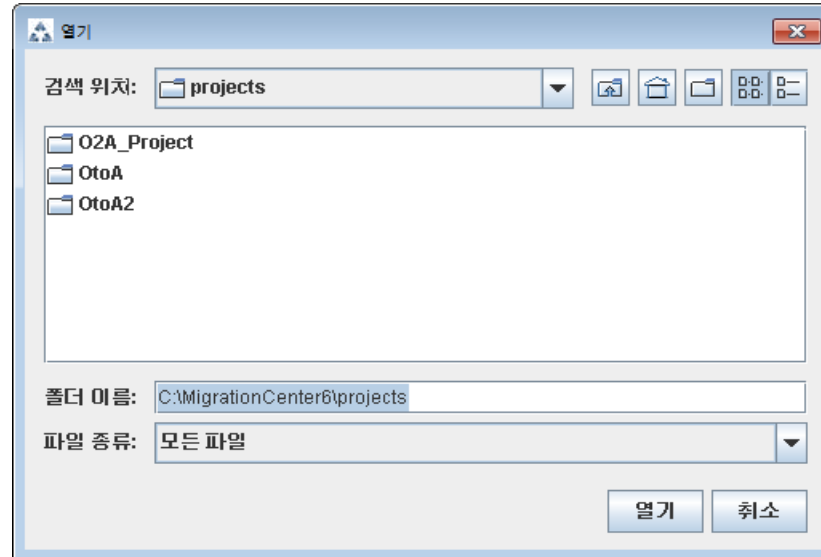
Destination Databa...: altibase6

OK Cancel

MIGRATION CENTER

❖ 프로젝트 열기

- "Project" 메뉴 아래의 "Open Project" 메뉴 항목 선택



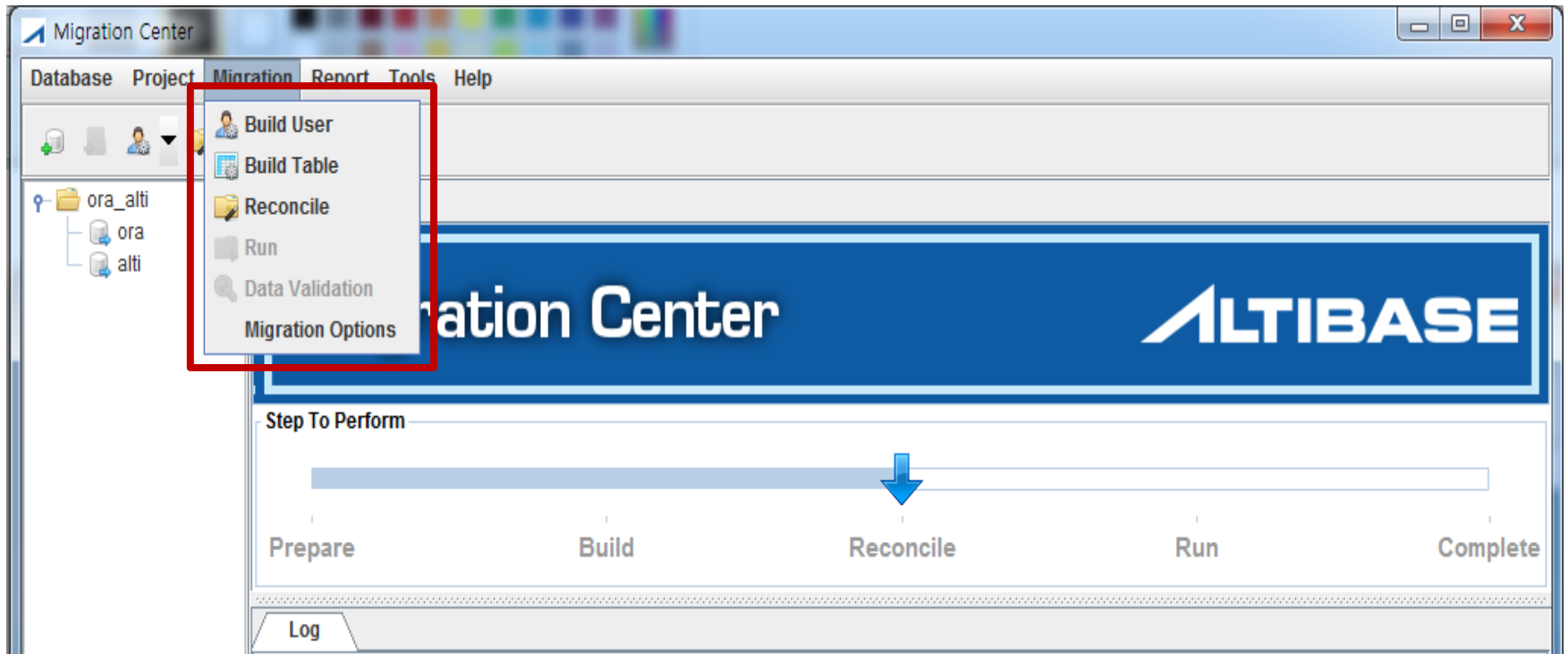
❖ 원본 및 대상 데이터베이스로 연결

- 원본 및 대상 데이터베이스로 연결은 "구축" 단계로 진행하기 위해 필수적 "Project" 메뉴 아래의 "Connect" 메뉴 항목을 선택하거나 열려진 프로젝트에 오른쪽 마우스 버튼 클릭 후 "Connect" 선택

MIGRATION CENTER

❖ 프로젝트 구축

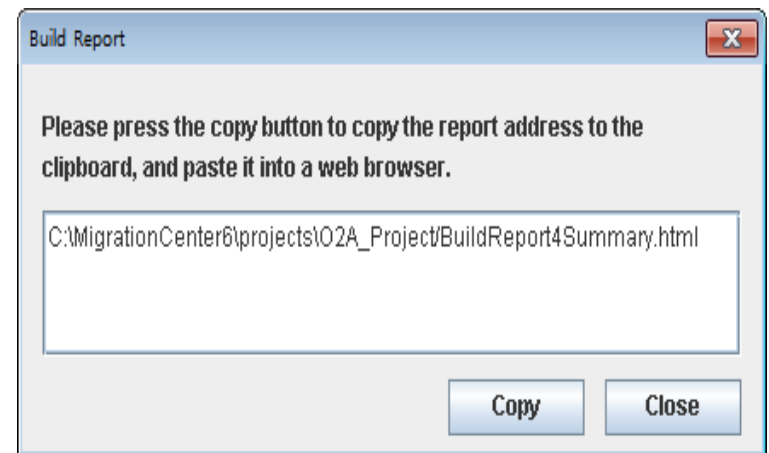
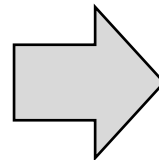
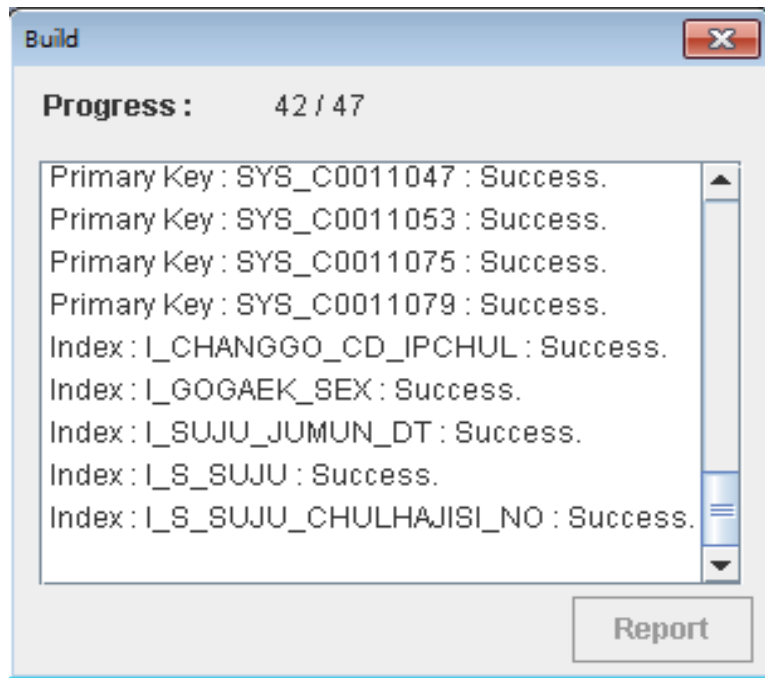
- "Migration" 메뉴 아래의 "Build User" 또는 "Build Table" 모드 선택 가능
- 사용자가 원하는 모드 클릭



MIGRATION CENTER

❖ 프로젝트 구축

- "Migration" 메뉴 아래의 "Build" 메뉴 아이템을 선택하거나, 도구 모음에 "Build" 아이콘 클릭



MIGRATION CENTER

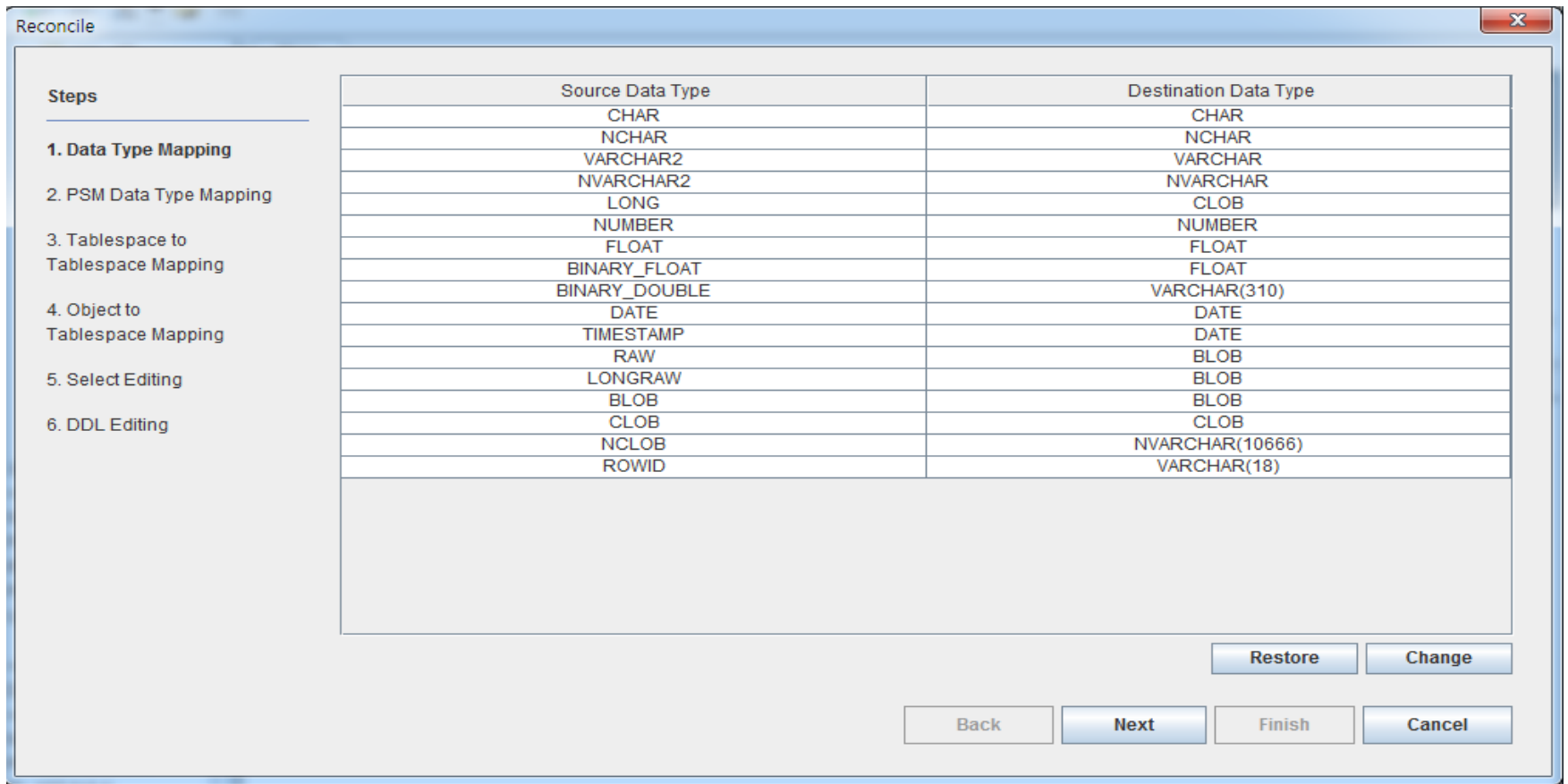
❖ 프로젝트 구축(BUILD REPORT)

Summary	Table	Constraint	Index	Sequence	Synonym	Library	Package	Package_Body	View	Materialized_View	Procedure	Function	Trigger	Comment	Missing
Contents 1. Time Information 2. Execution Environment Information 3. Product Information 4. Tablespaces 5. Migration Target Candidates															
1. Time Information [Top]															
Start Time	End Time	Duration													
2019/02/25 13:38:40	2019/02/25 13:39:10	0:00:30.105													
2. Execution Environment Information [Top]															
Title	Value														
Operating System	Windows 7 6.1 (amd64)														
Java Runtime Environment	Java HotSpot(TM) 64-Bit Server VM 1.7.0_80-b15 (Java(TM) SE Runtime Environment)														
VM Arguments	-XX:MaxPermSize=512m -Xms1g -Xmx4g -Duser.language=en														
Environment Variable 'JAVA_HOME'	C:\Users\user\Desktop\MigrationCenter7.6\jre\jre1.7.0_80_win_x64														
User Working Directory	C:\Users\user\Desktop\MigrationCenter7.6														
3. Product Information [Top]															
Properties	Source DBMS	Destination DBMS													
Connection Name	ora	alti													
Product	Oracle	Altibase													
DBMS Version	11.2.0.1.0	6.5.1.4.2													
JDBC	ojdbc5.jar (11.2.0.3.0)	Altibase.jar (6.5.1.4.2)													

MIGRATION CENTER

❖ 프로젝트 조정

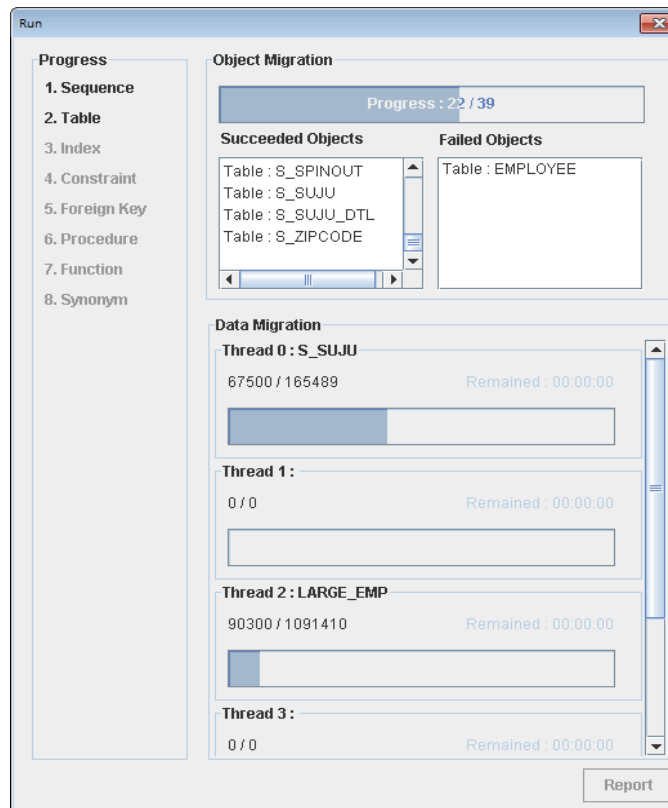
- "Migration" 메뉴 아래의 "Reconcile" 메뉴 항목을 선택하거나, 도구 모음에서 "Reconcile" 아이콘 클릭



MIGRATION CENTER

❖ 프로젝트 실행

- "Migration" 메뉴 아래의 "Run" 메뉴 항목을 선택하거나, 도구 모음에서 "Run" 아이콘 클릭



MIGRATION CENTER

❖ 커맨드 라인 인터페이스(CLI) 모드

➤ CLI 모드

- GUI 모드 실행이 불가능할 경우 사용
- 마이그레이션 절차 중 "실행" 단계만 수행 가능

➤ CLI 모드 수행 방법

- "project_path" 는 상대경로, 절대경로 모두 사용 가능

```
# ./migcenter.sh run project_path
```

MIGRATION CENTER

❖ 마이그레이션 가능한 데이터베이스 객체

➤ 타 DBMS(O사) to ALTIBASE

데이터베이스 객체 유형	User 모드에서 마이그레이션 가능 여부	Table 모드에서 마이그레이션 가능 여부	비고
Table	○	○	
Primary Key Constraint	○	○	
Unique Constraint	○	○	
Check Constraint	○	○	
Foreign Key Constraint	○	○	
Index	○	○	
Sequence	○	X	
Private Synonym	부분 지원	X	
Procedure	부분 지원	X	지원 예정
Function	부분 지원	X	지원 예정
View	부분 지원	X	지원 예정
Trigger	부분 지원	X	지원 예정

MIGRATION CENTER

❖ 데이터 타입 매핑

➤ 오라클 데이터베이스 to ALTIBASE

인덱스	소스	대상	특이 사항
1	CHAR	CHAR	
2	NCHAR	NCHAR	원본 및 대상 데이터베이스의 NCHAR 칼럼의 명시적인 크기는 같음(예.NCHAR(10) -> NCHAR(10)) 그러나, 오라클 JDBC 드라이버에서는 NCHAR 칼럼의 크기가 사용되는 바이트의 개수로 정의되는 반면, ALTIBASE의 JDBC 드라이버에서는 NCHAR 칼럼의 크기가 저장되는 문자의 개수로 정의되는 ALTIBASE에서 생성되는 NCHAR 칼럼이 필요에 따라 오라클보다 2 배 또는 3 배 정도 클 것이라는 의미로 유의 필요
3	VARCHAR2	VARCHAR	
4	NVARCHAR2	NVARCHAR	NCHAR와 같은 이유로, 컬럼 크기 서로 상이
5	LONG	CLOB	
6	NUMBER	NUMERIC	
7	FLOAT	FLOAT	
8	BINARY FLOAT	FLOAT	

MIGRATION CENTER

❖ 데이터 타입 매핑

➤ 오라클 데이터베이스 to ALTIBASE

인덱스	소스	대상	특이 사항
9	BINARY DOUBLE	VARCHAR(310)	ALTIBASE는 오라클 BINARY DOUBLE 타입과 호환되는 데이터 타입이 없으므로 데이터 손실을 막기 위해 문자형으로 저장
10	DATE	DATE	
11	TIMESTAMP	DATE	스케일의 차이로 인해서 소량의 데이터 손실이 발생 가능 오라클에서는 타임스탬프 값의 스케일이 나노초(9 자리수)인 반면, ALTIBASE는 타임스탬프 값의 스케일이 마이크로초(6 자리 수)
12	RAW	BLOB	
13	LONG RAW	BLOB	
14	BLOB	BLOB	
15	CLOB	CLOB	
16	NCLOB	NVARCHAR(10666)	ALTIBASE는 오라클 NCLOB 타입과 호환 가능한 데이터 타입이 없으므로, 최대 크기의 NVARCHAR 타입 사용 실제 데이터 크기가 NVARCHAR 최대 크기를 초과하는 경우, 데이터를 마이그레이션하는 동안 데이터 손실 발생 가능

MIGRATION CENTER

❖ 데이터 타입 매핑

➤ MS SQL SERVER to ALTIBASE

인덱스	소스	대상	특이 사항
1	BIGINT	BIGINT	
2	DECIMAL	NUMERIC	
3	INT	INTEGER	
4	NUMERIC	NUMERIC	
5	SMALLINT	SMALLINT	
6	MONEY	FLOAT	
7	TINYINT	SMALLINT	
8	SMALLINTMONEY	FLOAT	
9	BIT	CHAR(1)	
10	FLOAT	VARCHAR(310)	ALTIBASE는 SQL SERVER FLOAT 타입과 호환 가능한 데이터 타입이 없으므로, 데이터 손실을 막기 위해 VARCHAR(310)으로 사용
11	REAL	FLOAT	
12	DATE	DATE	

MIGRATION CENTER

❖ 데이터 타입 매핑

➤ MS SQL SERVER to ALTIBASE

인덱스	소스	대상	특이 사항
13	DATETIME2	DATE	스케일의 차이로 인해서 시간의 fraction 손실 발생 가능 SQL SERVER의 DATETIME2 타입 스케일이 나노초의 100 배(7 자리 수)인 반면, ALTIBASE는 DATE 타입의 스케일이 마이크로초(6 자리 수)
14	DATETIME	DATE	
15	SMALLDATETIME	DATE	
16	CHAR	CHAR	
17	TEXT	CLOB	
18	VARCHAR	VARCHAR	
19	VARCHAR(MAX)	CLOB	
20	NVARCHAR	NVARCHAR	
21	NVARCHAR(MAX)	NVARCHAR(10666)	ALTIBASE는 SQL SERVER NTEXT 타입과 호환 가능한 데이터 타입이 없으므로 최대 크기의 VARCHAR 타입이 사용 실제 데이터 길이가 최대 NVARCHAR 크기를 초과하는 경우, 데이터를 마이그레이션하는 동안 데이터 손실 발생 가능

MIGRATION CENTER

❖ 데이터 타입 매핑

➤ MS SQL SERVER to ALTIBASE

인덱스	소스	대상	특이 사항
22	BINARY	BYTE	
23	IMAGE	BLOB	
24	VARBINARY	BLOB	
25	ALLIDENTITY	NUMERIC(38,0)	
26	UNIQUEIDENTIFIER	VARCHAR(40)	ALTIBASE는 SQL SERVER UNIQUEIDENTIFIER 타입과 호환 가능한 데이터 타입이 없으므로, 데이터 손실을 방지하기 위해 VARCHAR 타입 사용
27	SYSNAME	NVARCHAR(128)	

MIGRATION CENTER

❖ 데이터 타입 매핑

➤ MySQL to ALTIBASE

인덱스	소스	대상	특이 사항
1	TINYINT	SMALLINT	
2	SMALLINT	INTEGER	
3	MEDIUMINT	INTEGER	
4	INT(INTEGER)	INTEGER	
5	BIGINT	BIGINT	
6	DECIMAL (NUMERIC)	VARCHAR(70)	ALTIBASE는 MySQL DECIMAL 타입과 호환 가능한 데이터 타입이 없으므로, 데이터 손실을 막기 위해 VARCHAR 타입 사용
7	FLOAT	FLOAT	
8	DOUBLE	VARCHAR(310)	ALTIBASE는 MySQL DOUBLE 타입과 호환 가능한 데이터 타입이 없으므로, 데이터 손실을 막기 위해 VARCHAR 타입 사용
9	BIT	BIT	
10	DATETIME	DATE	시각 부분이 0으로 설정
11	TIMESTAMP	DATE	TIMEZONE 제외

MIGRATION CENTER

❖ 데이터 타입 매핑

➤ MySQL to ALTIBASE

인덱스	소스	대상	특이 사항
12	CHAR	CHAR	
13	VARCHAR	VARCHAR	
14	CAHR with National Character	NCHAR	
15	VARCHAR with National Character	NVARCHAR	
16	BINARY	BYTE	
17	VARBINARY	BLOB	
18	TINYBLOB	BLOB	
19	TINYTEXT	VARCHAR(255)	
20	BLOB	BLOB	
21	TEXT	CLOB	
22	MEDIUMBLOB	BLOB	

MIGRATION CENTER

❖ 데이터 타입 매핑

➤ MySQL to ALTIBASE

인덱스	소스	대상	특이 사항
23	MIDIUMTEXT	CLOB	
24	LONGBLOB	BLOB	
25	LONGTEXT	CLOB	
26	ENUM	VARCHAR (10666)	ALTIBASE는 MySQL ENUM 타입과 호환 가능한 데이터 타입이 없으므로, 데이터 손실을 막기 위해 VARCHAR 타입 사용
27	SET	VARCHAR (10666)	ALTIBASE는 MySQL SET 타입과 호환 가능한 데이터 타입이 없으므로, 데이터 손실을 막기 위해 VARCHAR 타입 사용

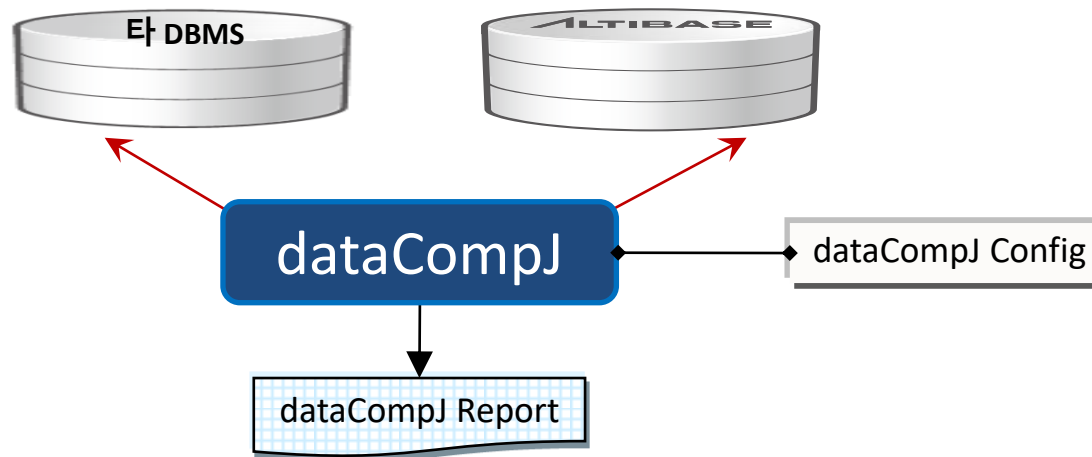
UTILITIES

DATACOMPJ

DATACompJ

❖ dataCompJ

- 이기종 데이터베이스를 테이블 단위로 데이터 비교 및 동기화 관련 기능을 수행하는 유틸리티
- ALTIBASE Adapter for Oracle 또는 ALTIBASE jdbcAdapter 로 인한 데이터불일치를 사용자 판단에 의해 일괄적 해결 목적
- 기본적으로 마스터 (MASTER) 데이터베이스를 기준으로 슬레이브(SLAVE) 데이터베이스를 일치시키는 정책 채택
- 대상 데이터베이스가 변경중에는 비 정상적으로 수행 가능성 존재



❖ dataCompJ 요구사항

- dataCompJ의 경우 자바 어플리케이션으로 JAVA_HOME 환경변수 설정 필요
 - Oracle이 Slave인 경우 : Java 5 이상의 Java Runtime Environment(JRE)
 - MariaDB가 Slave 인 경우 : Java 6 이상의 Java Runtime Environment(JRE)

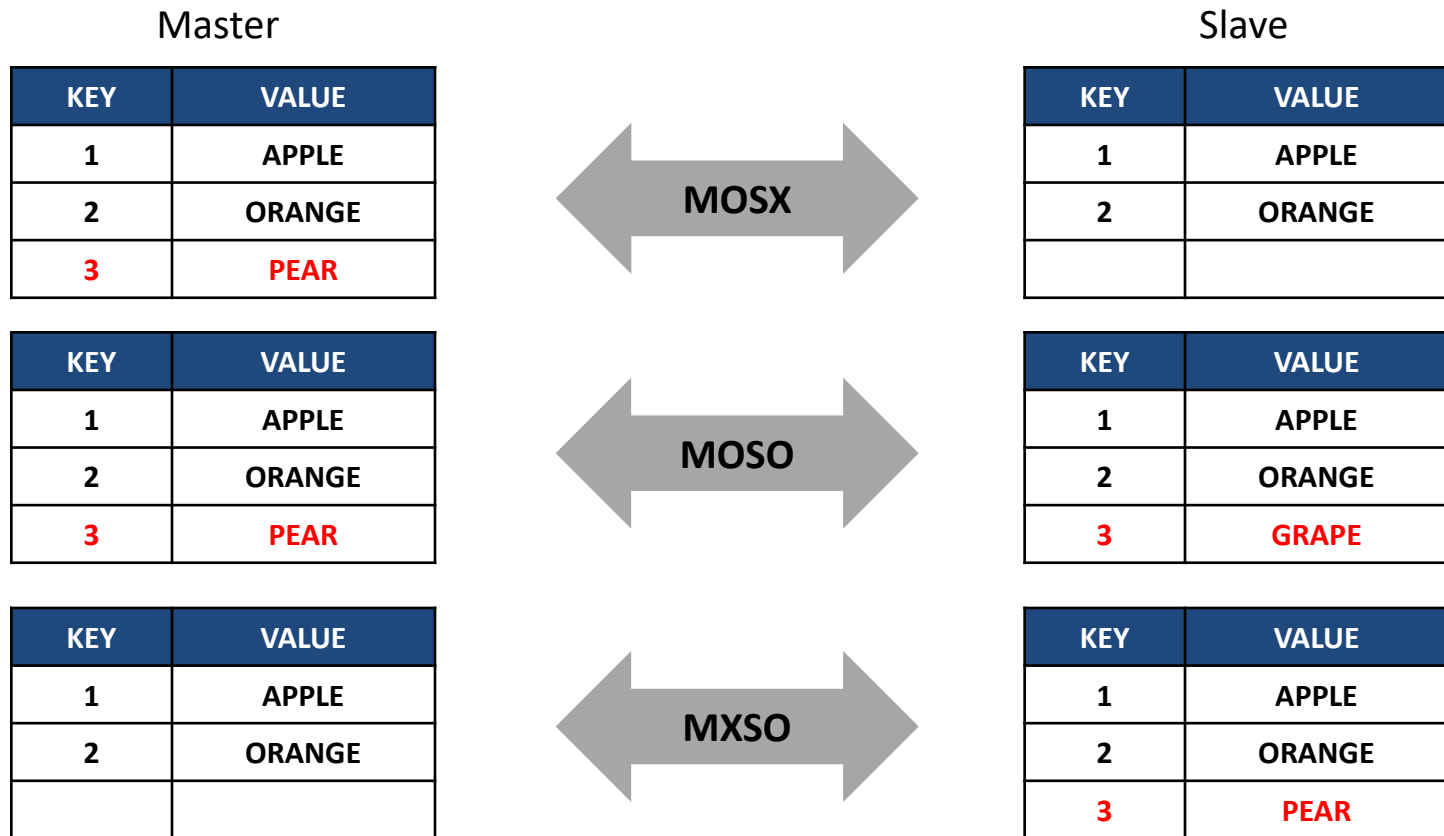
❖ dataCompJ 호환 데이터베이스

- Master 데이터베이스 : ALTIBASE 5.3.3 이상
- Slave 데이터베이스 : Oracle 9i 이상 , MariaDB 5.5.X 버전 이상

DATACOMPJ

❖ dataCompJ 데이터 불일치 종류

- Master 데이터베이스와 Slave 데이터베이스간의 데이터 불일치가 발생하는 세가지 경우



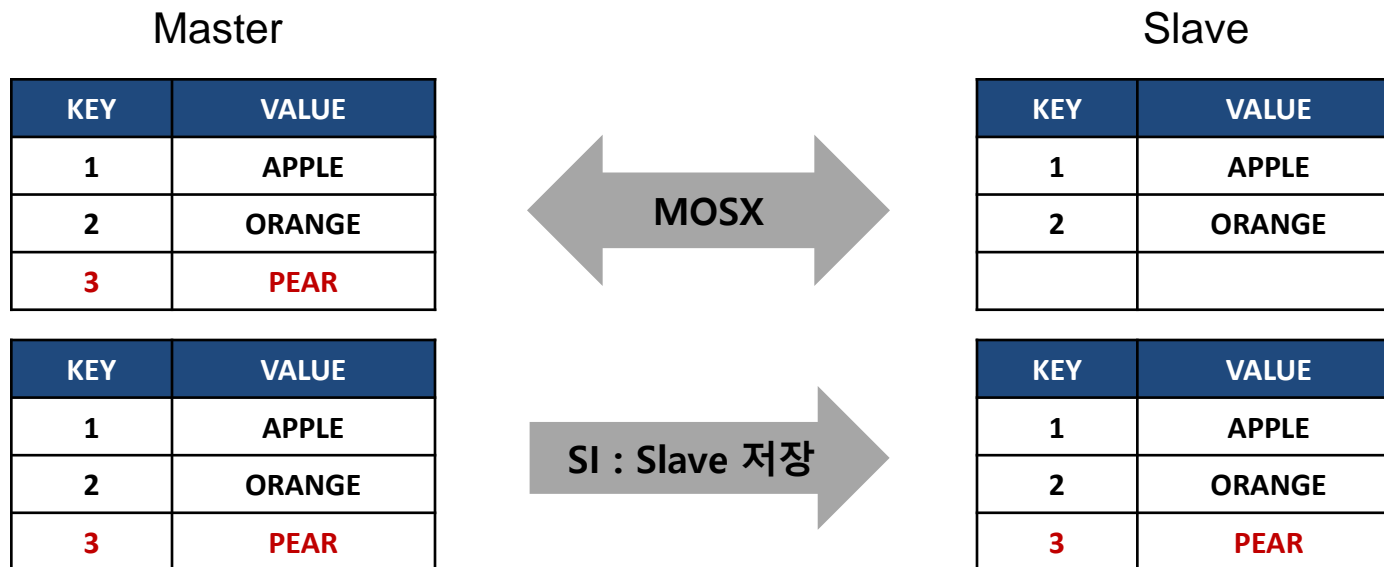
DATACOMPJ

❖ 데이터 동기화 정책

- DATACOMPJ 이후 Master 데이터베이스와 Slave 데이터베이스간의 데이터 불일치가 발생하는 세 가지 경우에 따라 다음의 데이터 동기화 정책 사용

❖ SI(SLAVE DATABASE INSERT)

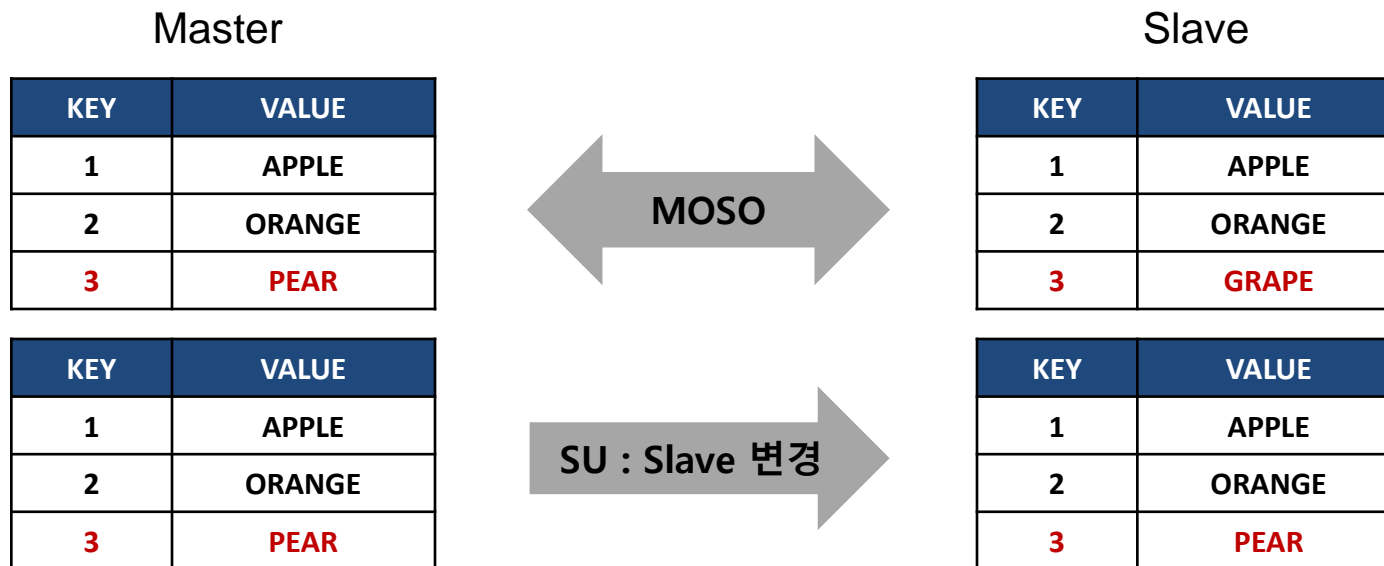
- MOSX 불일치를 해소하는 정책으로, Master 데이터베이스의 레코드를 Slave 데이터베이스에 삽입(Insert)



DATACOMPJ

❖ SU(SLAVE DATABASE UPDATE)

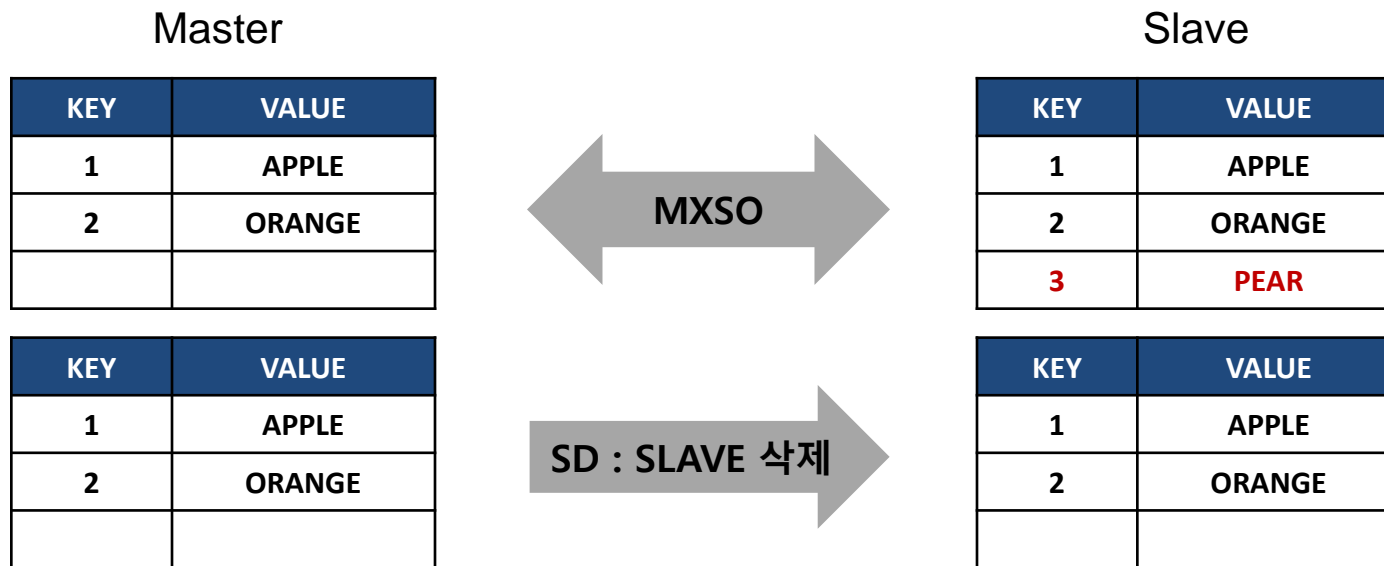
- MOSO 불일치를 해소하는 정책으로, Master 데이터베이스의 레코드 내용으로 Slave 데이터베이스 변경(Update)



DATACOMPJ

❖ SD(SLAVE DATABASE DELETE)

- MXSO 불일치를 해소하는 정책으로, Slave 데이터베이스의 레코드 삭제>Delete)



DATACompJ

❖ dataCompJ 의 사용 방법

- dataCompJ는 ALTIBASE 고객 서비스포털 <http://support.altibase.com>에서 다운로드 가능하며 zip 또는 tar.gz 형식으로 제공
- 압축 해제 후 바로 실행 가능

❖ dataCompJ 의 실행 방법

- Linux : `dataCompJCli.sh -f dataCompJ_env_file_path`
- Widows : `C:\W\dataCompJ> dataCompJ.bat -f dataCompJ_env_file_path`

❖ dataCompJ 의 수행 단계

- 구축(Bulid)
 - 환경파일을 기반으로 수행가능한지 판단
 - 문제발생시 `dataCompJ_report.txt`로 리포트 출력
- 실행(Run)
 - 테이블 데이터를 비교하여 비교(DIFF) 또는 일치(SYNC) 기능 수행

DATACompJ

❖ dataCompJ 의 환경파일 설정

- dataCompJ를 수행하기 위한 환경파일은 dataCompJ 설치 시 dataCompJ.xml sample 파일 제공
- dataCompJ 환경파일은 Connection, Options, TablePairs 3개 부분으로 제공

구분	설명
Connection	Master /Slave 데이터베이스의 접속을 위한 정보 기록
Options	dataCompJ 수행(DIFF/SYNC 등) 수행에 필요한 사용자 설정값 입력
TablePaires	TablePairs 부분은 비교 대상테이블이나 테이블이름이 나열되어있는 텍스트 파일의 경로를 지정 할때 사용

❖ dataCompJ SYNC 예제

➤ MASTER/SLAVE TABLE 구성

<MASTER-ALTIBASE>

```
iSQL> DESC tb_test1;  
[ TABLESPACE : SYS_TBS_DISK_DATA ]  
[ ATTRIBUTE ]
```

NAME	TYPE	IS NULL
C1	INTEGER	NOT NULL
C2	CHAR(10)	
C3	INTEGER	
C4	CHAR(20)	
C5	CHAR(10)	
C6	CHAR(10)	
C7	CHAR(10)	
C8	CHAR(10)	
C9	INTEGER	
C10	DATE	

```
iSQL> INSERT INTO tb_test1(c1,c10) VALUES(1,sysdate);
```

1 row inserted.

```
iSQL> INSERT INTO tb_test1(c1,c10) VALUES(2,sysdate);
```

1 row inserted.

```
iSQL> INSERT INTO tb_test1(c1,c10) VALUES(3,sysdate);
```

1 row inserted.

<SLAVE-ORACLE>

```
SQL> DESC tb_Test1;
```

Name	Null?	Type
C1	NOT NULL	NUMBER(38)
C2		CHAR(10)
C3		NUMBER(38)
C4		CHAR(10)
C5		CHAR(10)
C6		CHAR(10)
C7		CHAR(10)
C8		CHAR(10)
C9		NUMBER(38)
C10		DATE

```
SQL> SELECT count(*) FROM tb_test1;
```

COUNT(*)

0

❖ dataCompJ SYNC 예제

➤ 환경파일 (dataCompJ.xml)

```
<?xml version="1.0" encoding="UTF-8" ?>

<dataCompJ>

<Connections>
  <MasterDB>
    <JdbcUrl>jdbc:Altibase://192.168.1.35:26977/mydb</JdbcUrl>
    <JdbcFilePath>./jdbc/Altibase.jar</JdbcFilePath>
    <UserId>sys</UserId>
    <Password>manager</Password>
  </MasterDB>

  <SlaveDB>
    <JdbcUrl>jdbc:oracle:thin:@//192.168.1.145:1521/ORCL11G</JdbcUrl>
    <JdbcFilePath>./jdbc/ojdbc5.jar</JdbcFilePath>
    <UserId>scott</UserId>
    <Password>tiger</Password>
  </SlaveDB>
</Connections>
```

```
<Options>
  <Operation>SYNC</Operation>
  <Diff>
    <DirPath>./diff</DirPath>
  </Diff>
  <Sync>
    <MOSO UPDATE_TO_SLAVE="true"/>
    <MOSX INSERT_TO_SLAVE="true"/>
    <MXSO DELETE_FROM_SLAVE="true"/>
  </Sync>
  <Log>
    <DirPath>./</DirPath>
    <TraceInconsistentRecord>>false</TraceInconsistentRecord>
  </Log>
  <FileEncoding>UTF-8</FileEncoding>
  <MaxThread>0</MaxThread>
</Options>

<TablePairs>
  <TablePair>
    <MasterTable>SYS.TB_TEST1</MasterTable>
    <SlaveTable>SCOTT.TB_TEST1</SlaveTable>
  </TablePair>
</TablePairs>
</dataCompJ>
```

❖ dataCompJ SYNC 예제

➤ dataCompJ 실행

```
C:\Users\USER\Desktop\dataCompJCli7.1\dataCompJ>dataCompJ.bat -f dataCompJ.xml
```

```
-----  
Data Comparison Utility for Java (dataCompJ)  
Copyright 2000, ALTIBASE Corporation or its subsidiaries.  
All Rights Reserved.  
-----
```

```
* Build started at 2019-02-27 15:06:34.985  
* Build finished at 2019-02-27 15:06:37.522 (Elapsed: 0:00:02.550)  
* Run started at 2019-02-27 15:06:37.548  
  
* Run finished at 2019-02-27 15:06:37.994 (Elapsed: 0:00:00.446)
```

```
-----  
Refer to the following output files for details.  
-----
```

```
- dataCompJ_report.txt: Report file to provide summarized execution result.  
- dataCompJ.log: Log file to keep track of detailed program events during execu  
tion.  
- dataCompJ_data.log: Log file to keep track of inconsistent record events duri  
ng diff/sync operation.
```

❖ dataCompJ SYNC 예제

➤ dataCompJ 결과 리포트

```
=====
Build
- Started : 2019-02-27 15:06:34.985
- Finished : 2019-02-27 15:06:37.522
- Elapsed : 0:00:02.550
=====

[ User input information ]
--Connections--
<MASTER>
JobcUrl      : jdbc:Altibase://192.168.1.35:26977/mydb
JobcFilePath : ./jdbc/Altibase.jar
UserId       : sys
Password     : *****
FetchSize    : 1000
BatchSize    : 1000
<SLAVE>
JobcUrl      : jdbc:oracle:thin:@//192.168.1.145:1521/ORCL11G
JobcFilePath : ./jdbc/ojdbc5.jar
UserId       : scott
Password     : *****
FetchSize    : 1000
BatchSize    : 1000

--Options--
Operation          : Sync
FileEncoding       : UTF-8
Log-DirPath        : ./
Log-TraceInconsistentRecord: false
MaxThread          : 0
```

```
MaxThread          : 0
MaxThread(Adjusted) : 3
Sync               :
MOSO               : true
MOSX               : true
MXSO               : true

[ Problematic table(s): 0 ]
[ Candidate table(s) for data comparison: 1 ]
1: SYS.TB_TEST1 -> SCOTT.TB_TEST1
* SELECT SQL       : SELECT C1,C2,C3,C4,C5,C6,C7,C8,C9,C10 FROM
SYS.TB_TEST1 ORDER BY C1 ASC
* Excluded columns :
* Where condition  :
* N/A data type columns:

=====

Run
- Started : 2019-02-27 15:06:37.548
- Finished : 2019-02-27 15:06:37.994
- Elapsed : 0:00:00.446

=====

1: SYS.TB_TEST1 -> SCOTT.TB_TEST1
The fetched record count from MASTER:      3
The fetched record count from SLAVE:       0

=====

Type  Resolution          Try    Fail
-----
MOSO  UPDATE TO SLAVE        0      0
MOSX  INSERT TO SLAVE        3      0
MXSO  DELETE FROM SLAVE      0      0
=====
```

❖ dataCompJ SYNC 예제

➤ MASTER/SLAVE TABLE 결과 확인

```
<MASTER-ALTIBASE>
iSQL> select c1,c10 from tb_test1;
C1          C10
-----
1           27-FEB-2019
2           27-FEB-2019
3           27-FEB-2019
3 rows selected.
```

```
<SLAVE-ORACLE>
SQL> select c1,c10 from tb_test1;

          C1 C10
-----
1 27-FEB-19
2 27-FEB-19
3 27-FEB-19
```

❖ dataCompJ 사용시 유의 사항

- MasterDB와 SlaveDB는 반드시 동일하게 설정
 - 단, 이 기존 데이터베이스의 경우 비교가능한 타입이면 가능
- 지원하지 않는 데이터 타입의 칼럼은 비교 대상에서 자동으로 제외
 - LOB과 같은 binary type
- 주요 키(Primary key) 외에 값을 비교할 수 있는 칼럼이 적어도 하나 이상 필요

UTILITIES

JDBC ADAPTER

JDBC ADAPTER

❖ JDBC ADAPTER

- 이 기종 데이터베이스 간 혹은 ALTIBASE 타 버전간 데이터 복제 수행
- ALTIBASE Log Analysis API (ALA) 와 JDBC 사용하여 구현
 - ALA는 ALTIBASE 에서 변경된 데이터를 수신할 때 사용
 - JDBC는 변경된 데이터를 보낼 대상이 되는 데이터베이스로 데이터 전송할 때 사용

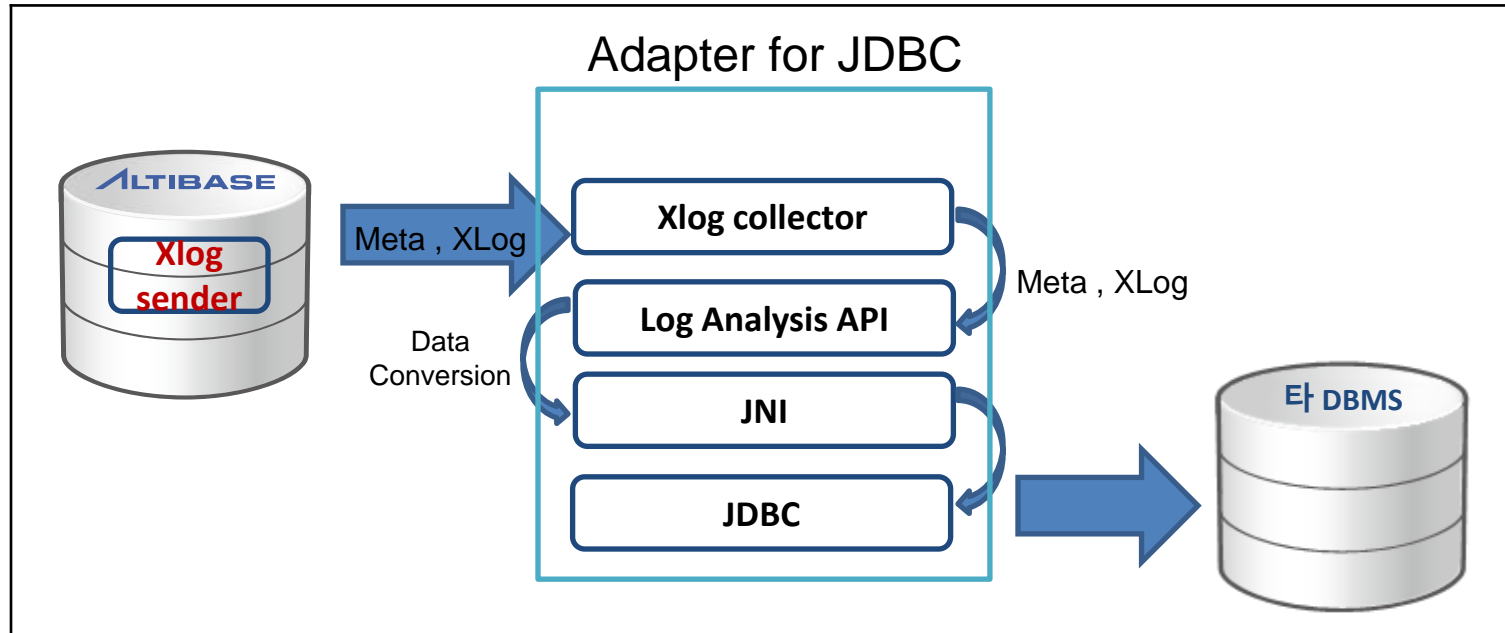
❖ 요구 사항

- 운영체제
 - LINUX : x86-64bit
- jdbcAdapter가 지원하는 데이터베이스 버전
 - ALTIBASE : v 6.3.1 이상
 - OTHER 데이터베이스 : JDBC 4.1 이하 버전 DBMS
- 데이터베이스 문자 집합이 지원하지 않는 언어에 해당하는 데이터는 NCHAR 또는 NVARCHAR 타입 사용해서 저장
- JRE 설치
 - JRE 7 버전 이상에서 동작
 - JAVA_HOME, CLASSPATH 등 환경 변수 설정

JDBC ADAPTER

❖ JDBC ADAPTER

- JDBC Adapter 구조 및 개념



- Xlog sender는 Xlog와 메타 정보를 생성하여 Xlog collector 에게 전송
- Xlog collector는 Xlog 와 메타 정보를 사용자에게 제공하기 위해 ALA 사용
- ALA는 획득한 데이터를 타 DBMS 적용을 위해 데이터 변환
- jdbcadapter는 변환된 데이터를 JNI → JDBC를 통해 타 DBMS에 적용

JDBC ADAPTER

❖ JDBC ADAPTER Property(jdbcAdapter.conf)

```
#####  
# Altibase Connection for Checking Constraints  
#####  
ALTIBASE_USER = sds  
ALTIBASE_PASSWORD = sds123  
ALTIBASE_IP = 127.0.0.1  
ALTIBASE_PORT = 20300
```

```
#####  
# for ALA  
#####  
ALA_SENDER_IP = 127.0.0.1  
ALA_SENDER_REPLICATION_PORT = 0  
ALA_RECEIVER_PORT = 25090  
ALA_REPLICATION_NAME = ALA  
ALA_SOCKET_TYPE = TCP  
ALA_XLOG_POOL_SIZE = 1000000  
ALA_LOGGING_ACTIVE = 1  
ALA_RECEIVE_XLOG_TIMEOUT = 300
```

Property	설명
ALA_SENDER_IP	ALTIBASE가 설치된 장비의 IP 주소
ALA_RECEIVER_PORT	Jdbcadapter가 데이터를 수신하기 위해 사용하는 포트 번호
ALA_REPLICAION_NAME	ALTIBASE에 존재(생성)하는 이중화 객체의 이름
ALA_XLOG_POOL_SIZE	Xlog 풀의 최대 크기를 설정 (def : 100,000)
ALA_SOCKET_TYPE	ALA가 사용할 통신 소켓 타입을 설정 (TCP/IP 와 Unix Domain 지원)
ALA_LOGGING_ACTIVE	ALA 가 트레이스 로그를 남길것인지 여부를 지정 (def :1)

JDBC ADAPTER

❖ JDBC ADAPTER Property

Property	설명
OTHER_DATABASE_USER	데이터를 보낼 대상이 되는 other 데이터베이스에 접속할 사용자 계정
OTHER_DATABASE_PASSWORD	데이터를 보낼 대상이 되는 other 데이터베이스에 접속할 사용자 계정 암호
OTHER_DATABASE_JDBC_MAX_HEAP_SIZE	JVM이 사용하는 Heap의 최대 크기 (def : 2048)
OTHER_DATABASE_JDBC_DRIVER_PATH	Other 데이터베이스 용 JDBC 드라이버 경로 지정
OTHER_DATABASE_JDBC_CONNECTION_URL	Other 데이터베이스의 연결 URL 지정
OTHER_DATABASE_GROUP_COMMIT	복수의 TRANSACTION을 한 번에 처리 원본 ALTIBASE SERVER에서 커밋을 해도 대상 Target 데이터베이스 는 일정한 수준의 TRANSACTION이 처리 될때 까지 커밋 연기 (def : 1) (0 = 그룹 커밋 미 실행, 1 = 그룹 커밋 실행)
OTHER_DATABASE_BATCH_DML_MAX_SIZE	Batch DML은 같은 종류인 복수의 DML구문을 일괄 처리 하는 것을 의미
OTHER_DATABASE_ERROR_RETRY_COUNT	레코드를 반영할 때 오류가 발생할 경우 재시도 횟수
OTHER_DATABASE_ERROR_RETRY_INTERVAL	레코드를 반영할 때 오류가 발생할 경우 재시도 간격

JDBC ADAPTER

❖ JDBC ADAPTER Property

```
#####  
# for JDBC Server Database  
#####  
OTHER_DATABASE_USER = sds  
OTHER_DATABASE_PASSWORD = sds123  
OTHER_DATABASE_SKIP_INSERT = 0  
OTHER_DATABASE_SKIP_UPDATE = 0  
OTHER_DATABASE_SKIP_DELETE = 0  
OTHER_DATABASE_SKIP_ERROR = 1  
OTHER_DATABASE_BATCH_DML_MAX_SIZE = 10  
OTHER_DATABASE_CONFLICT_LOGGING_LEVEL = 1  
OTHER_DATABASE_GROUP_COMMIT = 1  
OTHER_DATABASE_SET_USER_TO_TABLE = 0  
OTHER_DATABASE_ERROR_RETRY_COUNT = 0  
OTHER_DATABASE_ERROR_RETRY_INTERVAL = 0  
OTHER_DATABASE_JDBC_MAX_HEAP_SIZE = 2048  
OTHER_DATABASE_JDBC_DRIVER_PATH = /sas_home/mbw2/jdbcAdapter/Altibase4.jar  
OTHER_DATABASE_JDBC_DRIVER_CLASS = Altibase4.jdbc.driver.AltibaseDriver  
OTHER_DATABASE_JDBC_CONNECTION_URL = "jdbc:Altibase://192.168.1.35:52254/mydb"
```

JDBC ADAPTER

❖ JDBC ADAPTER 사용법

➤ ALTIBASE 내 Xlog Sender 생성

```
iSQL> CREATE REPLICATION ala1 FOR ALNANALYSIS WITH '127.0.0.1',25090 FROM sys.t1 to sys.t1;  
Create success.
```

➤ jdbcadapter 실행

```
$ pwd  
/sas_home/home/jdbcAdapter/bin  
$ ./jdbcAdapter
```

ALA 실행

```
iSQL> ALTER REPLICATION ala1 START;
```

➤ 확인

```
iSQL> SELECT rep_name,rep_Gap FROM V$REPGAP;  
REP_NAME                REP_GAP  
-----  
ALA                      25687513  
1 row selected.  
  
$ ps -ef | grep jdbc  
mbw2      29501      1  0 10:30 ?          00:00:00 /sas_home/home/jdbcAdapter/bin/jdbcAdapter --daemon
```

JDBC ADAPTER

❖ JDBC ADAPTER 사용법

- Adapter for JDBC 유틸리티(oaUtility)
 - jdbcAdapter를 데몬으로 실행 및 상태 확인 스크립트
 - 타 DBMS 확인 불가능
- sample

```
$ oaUtility start
Start /sas_home/home/jdbcAdapter/bin/jdbcAdapter...
JDBC Adapter started.

$ oaUtility status
/sas_home/home/jdbcAdapter/bin/jdbcAdapter is alive.

$ oaUtility check                → jdbcadapter 가 동작 중인지 확인하다가 종료되면 재 시작
/sas_home/home/jdbcAdapter/bin/jdbcAdapter is alive.
/sas_home/home/jdbcAdapter/bin/jdbcAdapter is alive.
/sas_home/home/jdbcAdapter/bin/jdbcAdapter is alive.

$ oaUtility stop
/sas_home/home/jdbcAdapter/bin/jdbcAdapter's PID is 29501.
Shutdown /sas_home/home/jdbcAdapter/bin/jdbcAdapter...
```

UTILITIES

JOB SCHEDULER

JOB SCHEDULER

※ 6.3.1 버전부터 제공

❖ 개념

- 저장 프로시저에 실행 일정을 결합한 JOB 객체 제공
- 한 개의 JOB에는 한 개의 프로시저만 등록 가능
- JOB_SCHEDULER_ENABLE 과 JOB_THREAD_COUNT 프로퍼티 설정 필요
- SYS 사용자만이 JOB을 생성, 변경, 삭제 가능

❖ JOB SCHEDULER 시작 및 종료

➤ Job scheduler 시작

```
iSQL> ALTER SYSTEM SET JOB_SCHEDULER_ENABLE = 1;  
Alter success.  
iSQL> ALTER SYSTEM SET JOB_THREAD_COUNT = 8 ; => property 파일을 수정해야 함  
Alter success.
```

➤ Job scheduler 종료

```
iSQL> ALTER SYSTEM SET JOB_SCHEDULER_ENABLE = 0 ;  
Alter success.
```

JOB SCHEDULER

❖ 생성 구문

```
CREATE JOB job_name
EXEC procedure_name
START expr1
END expr2
INTERVAL number {YEAR | MONTH | DAY | HOUR | MINUTE}
{DISABLE | ENABLE};
```

❖ 생성 예제

- 이름이 proc1인 프로시저가 현재부터 실행되어 1시간 주기로 작업을 실행 후 3일 후에 끝나도록 JOB1 생성(job1의 상태는 DISABLE)

```
iSQL> CREATE JOB job1
      2 EXEC proc1
      3 START sysdate
      4 END sysdate + 3
      5 INTERVAL 1 HOUR;
Create success.
```

JOB SCHEDULER

❖ 변경 구문

```
ALTER JOB job_name
SET execute_procedure_statement |
    START expr1 |
    END expr2 |
    INTERVAL number {YEAR | MONTH | DAY | HOUR | MINUTE}
ENABLE | DISABLE;
```

❖ 변경 예제

- Job1이 실행되는 시작 시간을 '2019년 1월 1일' 로 변경

```
iSQL> ALTER JOB job1
      2 SET START to_date( ' 20190101 ' , ' YYYYMMDD ' );
Alter success.
```

- Job1의 실행 주기를 10분 간격으로 변경(job1의 상태는 ENABLE로 변경)

```
iSQL> ALTER JOB job1
      2 SET INTERVAL 10 MINUTE
      3 ENABLE;
Alter success.
```

JOB SCHEDULER

❖ 삭제 구문

```
DROP JOB job_name ;
```

❖ 삭제 예제

- ▶ 이름이 job1인 JOB 제거

```
iSQL> DROP JOB job1;  
Drop success.
```

UTILITIES

ORANGE FOR ALTIBASE

ORANGE FOR ALTIBASE

❖ ORANGE for ALTIBASE

- ALTIBASE 운영환경에서 쉽고, 빠르게 개발의 효율성과 생산성을 높이고 성능의 최적화가 가능한 어플리케이션 개발 및 성능 관리 DBA 툴
- SQL 및 PSM 개발을 지원하는 다양한 도구들을 제공하여 쉽고 빠르게 개발 할 수 있는 환경 제공
- 시스템의 안정적인 운영 및 관리를 위해 세션(session), SQL, 락(Lock)들에 대한 모니터링 기능 제공

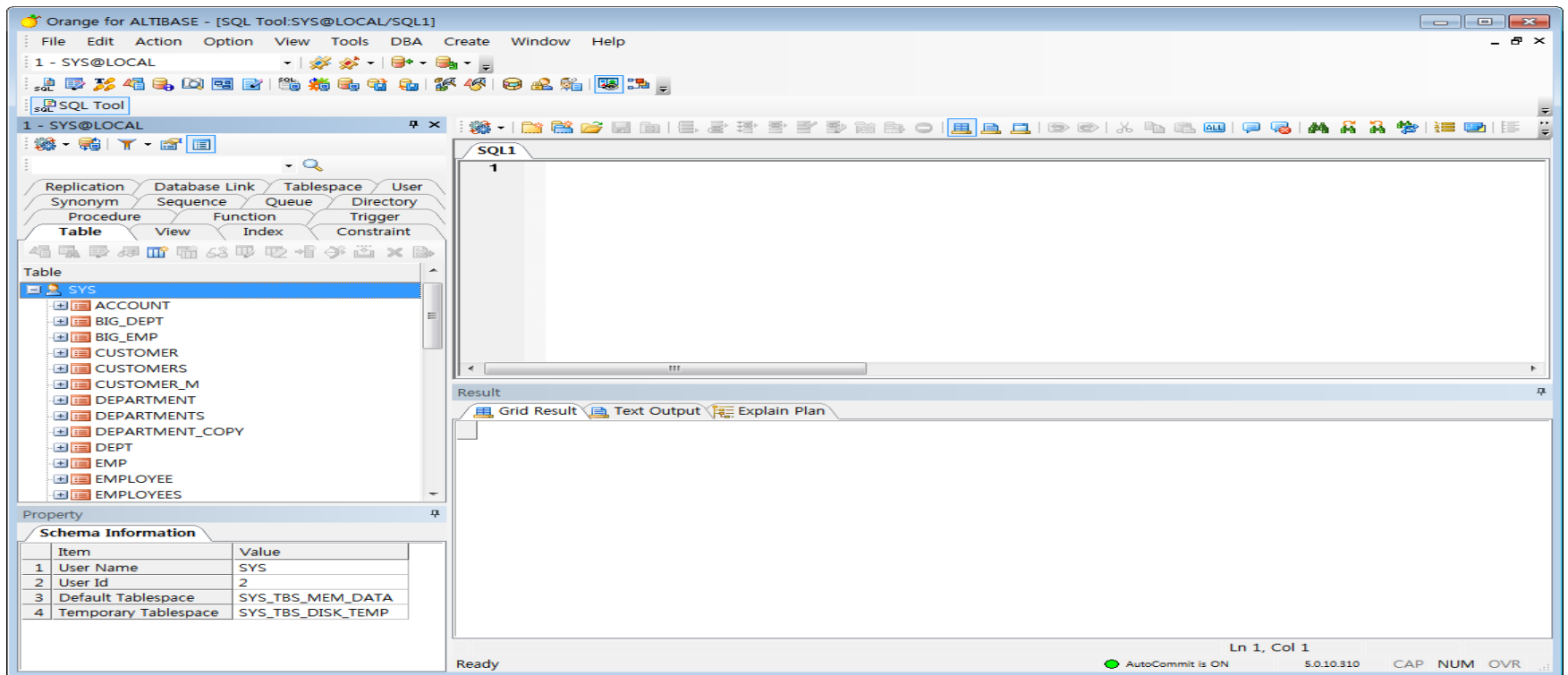
❖ 제품 구성

- Standard Edition : 개발자 / 컨설턴트를 위한 필수 기능
- DBA Edition : DBA 혹은 성능 분석을 위한 고급 기능

ORANGE FOR ALTIBASE

❖ ORANGE 실행 화면

- ▶ [시작] → [모든 프로그램] → [Orange for ALTIBASE 5.0 DBA] 실행



ORANGE FOR ALTIBASE

❖ ORANGE 상세 기능

➤ SQL Tool

- 다수의 SQL을 일괄 또는 순차 실행하거나 특정 SQL만을 선택적 실행할 수 있고 실행 결과를 직접 수정하거나 파일로 저장 가능

The screenshot displays the Orange for ALTIBASE SQL Tool interface. The main window shows a SQL query in the SQL1 editor:

```
1 SELECT eno, ename, dno, emp_job
2 FROM employee
3 WHERE dno IN (SELECT dno
4 FROM employee
5 WHERE eno IN (13, 15))
6 AND emp_job IN (SELECT emp_job
7 FROM employee
8 WHERE eno IN (13, 15));
9
```

Below the editor, the 'Result' tab is active, showing a grid of data:

ENO	ENAME	DNO	EMP_JOB
13	KWKIM	1002	PM
11	MSKIM	1003	WEBMASTER
14	KCIJUNG	1003	PM
15	JHSEOUNG	1003	WEBMASTER

The status bar at the bottom indicates: Ready, Ln 9, Col 17, 0.13 sec., 4 rows, AutoCommit is ON, 5.0.10.310, CAP, NUM, OVR.

ORANGE FOR ALTIBASE

❖ ORANGE 상세 기능

➤ Table Editor

- 데이터의 DML 작업을 GUI환경에서 편리하게 할 수 있으며, 조건에 의한 일부 데이터의 조회, 편집, 저장 기능 제공

The screenshot displays the 'Table Editor' window for the 'SYS.EMPLOYEES' table. The interface includes a menu bar (File, Action, Option, View, Tools, DBA, Create, Window, Help), a toolbar, and a left-hand pane for 'Accessible Tables'. The main area shows a table with 20 rows of employee data. The status bar at the bottom indicates 'Ready', 'AutoCommit is ON', and '20 rows'.

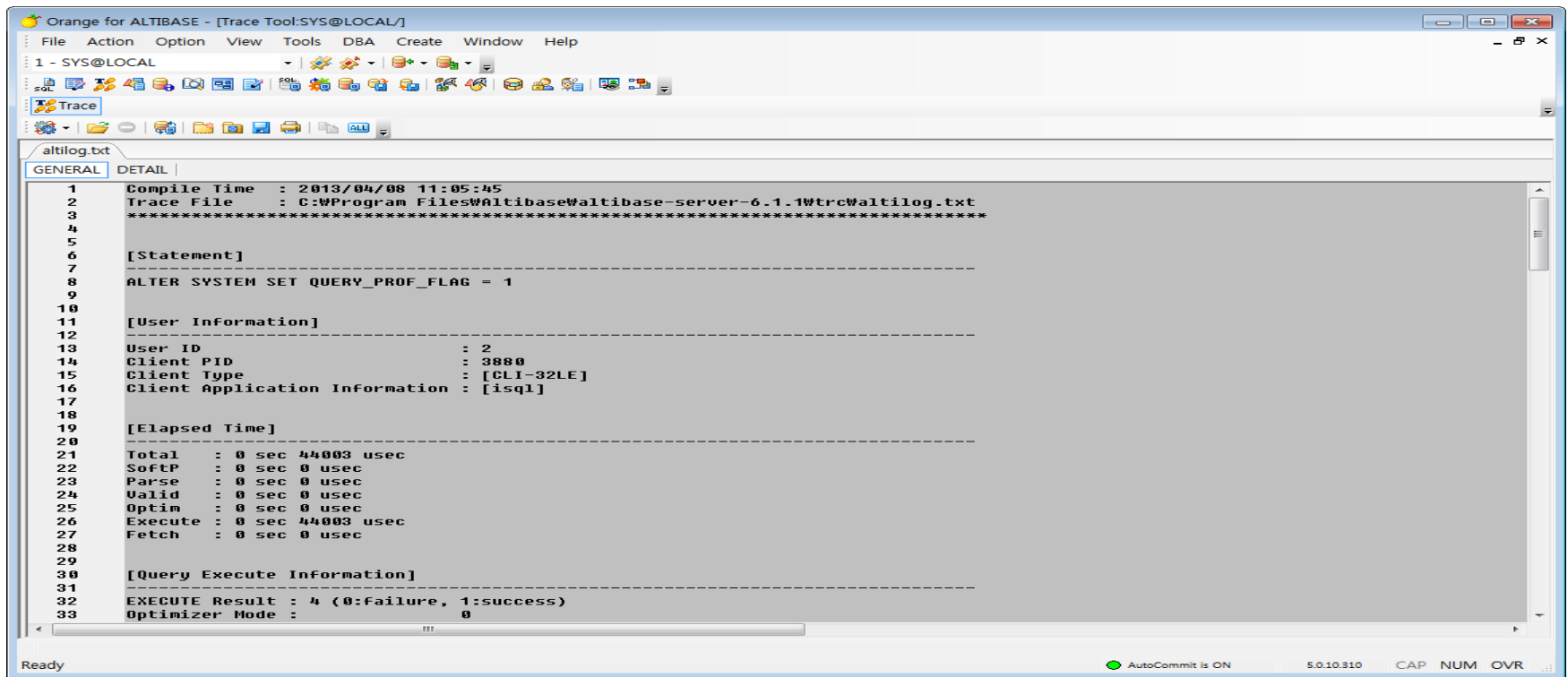
	ENO	E_LASTNAME	E_FIRSTNAME	EMP_JOB	EMP_TEL	DNO	SALARY	SEX	BIRTH	JOIN_DATE	STATUS
1	1	Moon	Chan-seung	CEO	01195662365	3002		M			R
2	2	Davenport	Susan	designer	0113654540		1500	F	721219	2009/11/18 00:00:00	H
3	3	Kobain	Ken	engineer	0162581369	1001	2000	M	650226	2010/01/11 00:00:00	H
4	4	Foster	Aaron	PL	0182563984	3001	1800	M	820730		H
5	5	Ghorbani	Farhad	PL	01145582310	3002	2500	M		2009/12/20 00:00:00	H
6	6	Momoi	Ryu	programmer	0197853222	1002	1700	M	790822	2010/09/09 00:00:00	H
7	7	Fleischer	Gottlieb	manager	0175221002	4002	500	M	840417	2004/01/24 00:00:00	H
8	8	Wang	Xiong	manager	0178829663	4001		M	810726	2009/11/29 00:00:00	H
9	9	Diaz	Curtis	planner	0165293668	4001	1200	M	660102	2010/06/14 00:00:00	H
10	10	Bae	Elizabeth	programmer	0167452000	1003	4000	F	710213	2010/01/05 00:00:00	H
11	11	Liu	Zhen	webmaster	0114553206	1003	2750	M		2011/04/28 00:00:00	H
12	12	Hammond	Sandra	sales rep	0174562330	4002	1890	F	810211	2009/12/14 00:00:00	H
13	13	Jones	Mitch	PM	0187636550	1002	980	M	801102		H
14	14	Miura	Yuu	PM	0197664120	1003	2003	M			H
15	15	Davenport	Jason	webmaster	0119556884	1003	1000	M	901212		H
16	16	Chen	Wei-Wei	manager	0195562100	1001	2300	F	780509		H
17	17	Fubuki	Takahiro	PM	0165293886	2001	1400	M	781026	2010/05/07 00:00:00	H
18	18	Huxley	John	planner	01755231044	4001	1900	M		2007/10/30 00:00:00	H
19	19	Marquez	Alvar	sales rep	0185698550	4002	1800	M		2010/11/18 00:00:00	H
20	20	Blake	William	sales rep	01154112366	4002		M		2006/11/18 00:00:00	H

ORANGE FOR ALTIBASE

❖ ORANGE 상세 기능

➤ Trace Tool

- Trace 파일을 분석하여 파싱 정보, 실행 계획 정보 실시간 제공



The screenshot displays the Orange for ALTIBASE Trace Tool interface. The window title is "Orange for ALTIBASE - [Trace Tool:SYS@LOCAL]". The menu bar includes File, Action, Option, View, Tools, DBA, Create, Window, and Help. The toolbar contains various icons for file operations and database actions. The main area shows a trace file named "altilog.txt" with the following content:

```
1 Compile Time : 2013/04/08 11:05:45
2 Trace File : C:\Program Files\Altibase\altibase-server-6.1.1\trc\altilog.txt
3 *****
4
5
6 [Statement]
7 -----
8 ALTER SYSTEM SET QUERY_PROF_FLAG = 1
9
10
11 [User Information]
12 -----
13 User ID : 2
14 Client PID : 3880
15 Client Type : [GLI-32LE]
16 Client Application Information : [isql]
17
18
19 [Elapsed Time]
20 -----
21 Total : 0 sec 44003 usec
22 SoftP : 0 sec 0 usec
23 Parse : 0 sec 0 usec
24 Valid : 0 sec 0 usec
25 Optim : 0 sec 0 usec
26 Execute : 0 sec 44003 usec
27 Fetch : 0 sec 0 usec
28
29
30 [Query Execute Information]
31 -----
32 EXECUTE Result : 4 (0:failure, 1:success)
33 Optimizer Mode : 0
```

The status bar at the bottom shows "Ready", "AutoCommit is ON", "5.0.10.310", "CAP NUM OVR", and a small icon.

ORANGE FOR ALTIBASE

❖ ORANGE 상세 기능

➤ Description Tool

- 데이터베이스 접속 사용자에게 허용된 모든 Object의 상세 Description을 제공
- 다양한 검색 조건에 의한 특정 객체 찾기 기능 제공

Orange for ALTIBASE - [Description Tool:SYS@LOCAL]

File Action Option View Tools DBA Create Window Help

1 - SYS@LOCAL

Desc

Search

Search by criteria below.

Object Name
employee

Object Type
<< All Object >>

Which word is included in source?

When was it modified?

What size is it?

Search

Drag a column header here to group by that column.

Type	Owner	Name	DB Link	Size(KB)	Created	Last DDL Time
TABLE	SYS	EMPLOYEE		32	2013/01/28 11:...	2013/01/28 11:...

Description

Script Properties Reference Referenced By Column Segment Index Constraint Trigger Grant

```
1 -- DDL Script was generated by Orange for ALTIBASE
2 -- using session 'SYS@LOCAL' on '2013/04/04 15:35:17'.
3
4 CREATE TABLE SYS.EMPLOYEE
5 (
6     ENO          INTEGER,
7     ENAME        CHAR(20) NOT NULL,
8     EMP_JOB      VARCHAR(15) FIXED,
9     EMP_TEL      CHAR(15),
10    DNO           SMALLINT,
11    SALARY        NUMERIC(10,2) DEFAULT 0,
12    SEX           CHAR(1) DEFAULT 'M' NOT NULL,
13    BIRTH         CHAR(4),
14    JOIN_DATE     DATE,
15    STATUS        CHAR(1) DEFAULT 'H',
16    MGR           INTEGER
17 )
```

Ready AutoCommit is ON 5.0.10.310 1 Object(s) CAP NUM OVR

ORANGE FOR ALTIBASE

❖ ORANGE 상세 기능

➤ Load Tool

- Txt, csv, xls 등 다양한 포맷의 외부 파일로부터 데이터베이스의 테이블에 데이터로 Load 할 수 있는 기능 제공

Orange for ALTIBASE - [Load Tool-<SYS@LOCAL>]

File Action Option View Tools DBA Create Window Help

3 - SYS@LOCAL

Load Unload

Target

Schema: SYS
Table: EMP

Data File

File: E:\W2222.csv
File Format: Symbol Separated
 1st row is column name 1st col is row #
Separator: Comma(,)
Excel Sheet: All

Pre-Process

Truncate before loading

Table Column	Type	File Column	If NULL	Always	Terminated by	Predefined Value
1 EMPNO	NUMERIC	Column 1				
2 ENAME	VARCHAR	[NONE]				
3 JOB	VARCHAR	[NONE]				
4 MGR	NUMERIC	[NONE]				
5 HIREDATE	DATE	[NONE]				
6 SAL	NUMERIC	[NONE]				
7 COMM	NUMERIC	[NONE]				
8 DEPTNO	NUMERIC	[NONE]				

	EMPNO	ENAME	JOB	MGR	HIREDATE	SAL	COMM	DEPTNO
1	#ENO							
2	11Moon							
3	22Davenport							
4	33Kobain							
5	44Foster							
6	55Ghorbani							

Ready

0.02 sec. 0 rows 21 errors
AutoCommit is ON 5.0.10.310 CAP NUM OVR

ORANGE FOR ALTIBASE

❖ ORANGE 상세 기능

➤ Unload Tool

- 데이터베이스의 테이블이나 VIEW를 선택하여 txt, csv, xls 등 다양한 포맷의 파일로 데이터 내려 받기 가능

Orange for ALTIBASE - [Unload Tool-SYS@LOCAL>]

File Action Option View Tools DBA Create Window Help

2 - SYS@LOCAL

Desc Unload

Unload Method

Using Table Using Query

Source

Schema: SYS

Table: EMPLOYEE

Data File

File Format

Symbol Separated Fixed Width

Insert into statement

Include Column Name Include Row Number

Separator

Tab Comma(,) User-Defined

Excel Sheet

Sheet Name

	Column Order	Type	Terminated By
1	ENO	INTEGER	
2	ENAME	CHAR	
3	EMP_JOB	VARCHAR	
4	EMP_TEL	CHAR	
5	DNO	SMALLINT	
6	SALARY	NUMERIC	
7	SEX	CHAR	
8	BIRTH	CHAR	
9	JOIN_DATE	DATE	
10	STATUS	CHAR	
11	MGR	INTEGER	

	ENO	ENAME	EMP_JOB	EMP_TEL	DNO	SALARY	SEX	BIRTH	JOIN_DATE	STATUS	MGR
1	1	EJUNG	CEO	01195662365	3002		M			R	
2	2	HJNO	DESIGNER	0113654540		1500000	F	1219	1999/11/18 00:00:00	H	16
3	3	HSCOI	ENGINEER	0162581369	1001	2000000	M	0226	2000/01/11 00:00:00	H	16
4	4	KSKIM	PL	0182563984	3001	1800000	M	0730		H	17
5	5	SJKIM	PL	01145582310	3002	2500000	M		1999/12/20 00:00:00	H	17

Ready

0.00 sec.

AutoCommit is ON

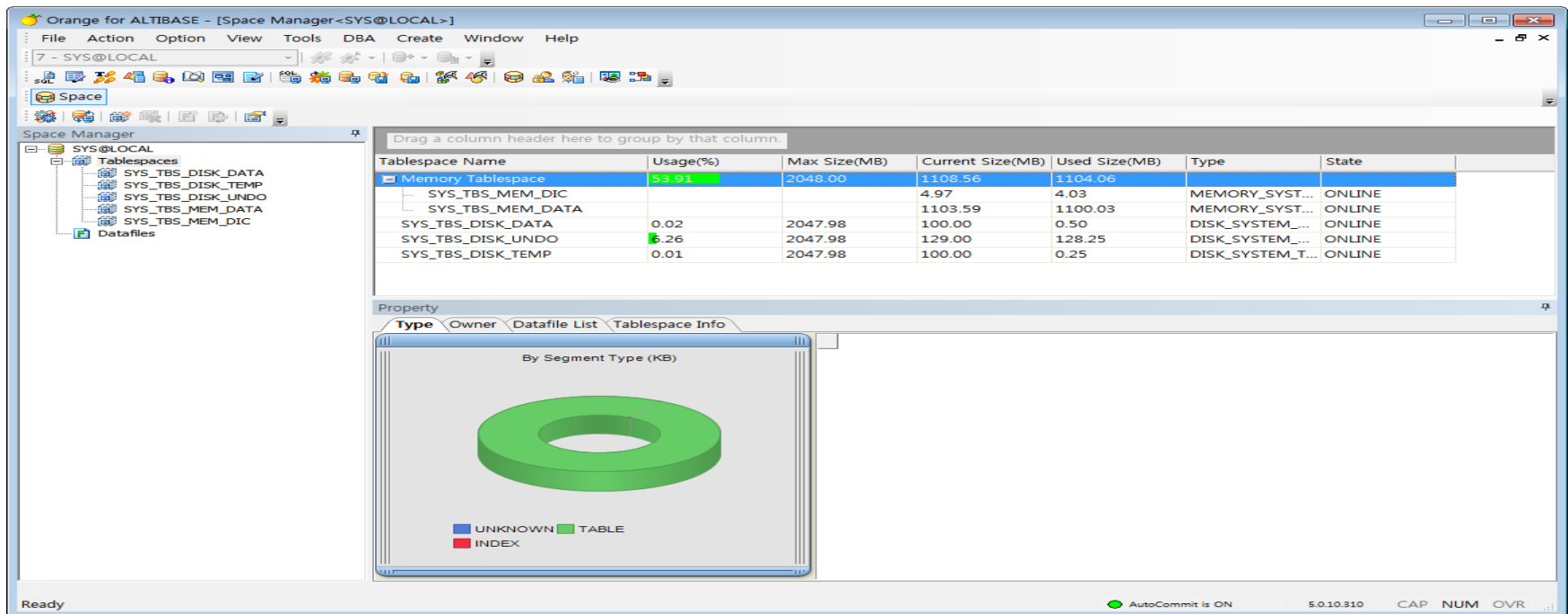
5.0.10.310 CAP NUM OVR

ORANGE FOR ALTIBASE

❖ ORANGE 상세 기능

➤ Space Manager

- 테이블스페이스와 데이터 파일에 대한 사용 현황을 다양한 방식으로 제공



ORANGE FOR ALTIBASE

❖ ORANGE 상세 기능

➤ Schema Browser

- Schema Object와 테이블스페이스, 이중화, 테이블 등 객체 정보를 Tree 형태로 가시화

Table Data View

Item	Value
1 User Name	SYS
2 User Id	2
3 Default Tablespace	SYS_TBS_MEM_DATA
4 Temporary Tablespace	SYS_TBS_DISK_TEMP

Index Data View

Column	Type	Sort
1 MGR_NO	INTEGER	ASC

Tablespace Data View

Item	Value
1 Db Name	mydb
2 Product Signature	WIN_NT_5.0-32bit-6.1.1.4
3 Db Signature	2C4138081BD0-50863E
4 Version Id	100728833
5 Endian	Little endian
6 Logfile Size	10485760

ALTIBASE ADMINISTRATION II

NEW FEATURE

NEW FEATURE

❖ CONTENTS

- ALTIBASE SHARDING
- COMPRESSION COLUMN
- HYBRID PARTITIONED TABLE
- IPCDA PROTOCOL
- ROLE

NEW FEATURE

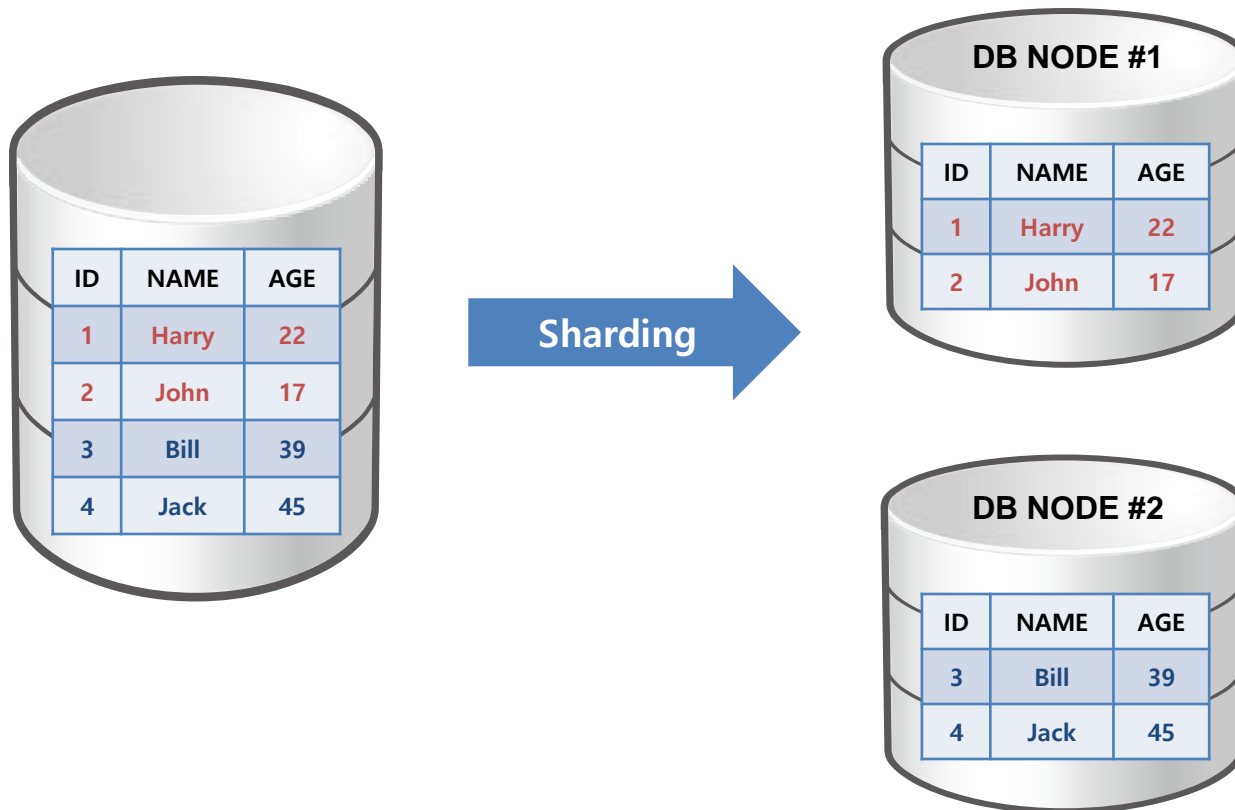
ALTIBSE SHARDING

ALTIBASE SHARDING

※ 7.1.0 버전부터 제공

❖ 개념

- ▶ 한 대의 데이터베이스에 저장했던 데이터를 여러 대의 데이터베이스에 분산하여 저장 및 처리하는 스케일 아웃 기술



ALTIBASE SHARDING

❖ SHARDING 종류

- SERVER측 샤딩
- CLIENT측 샤딩
- HYBRID 샤딩

❖ ALTIBASE SHARDING

- HYBRID 샤딩 방식 채택
 - 자동으로 SERVER측, CLIENT측 샤딩 중 최적의 경로 선택
- 다양한 분상 방식, 분산 객체, 유틸리티 지원
- 기존의 SQL을 수정하지 않고, 라이브러리 교체만으로 쉽게 적용 가능

NEW FEATURE

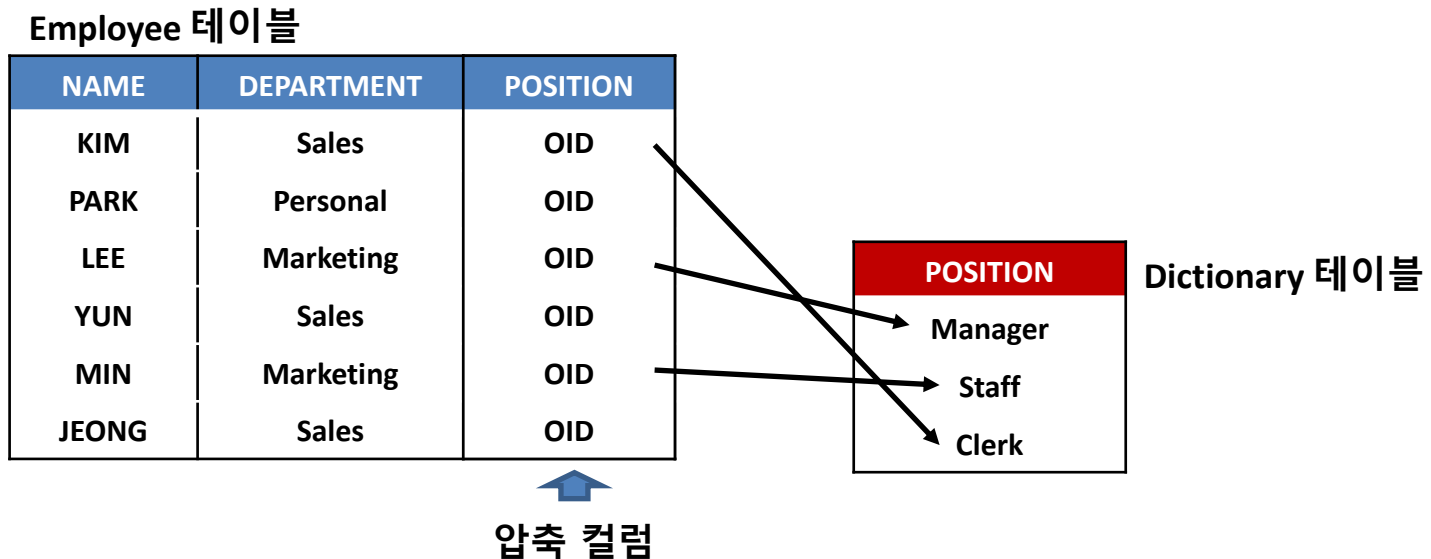
COMPRESSION COLUMN

COMPRESSION COLUMN

※ 6.3.1 버전부터 제공

❖ 개념

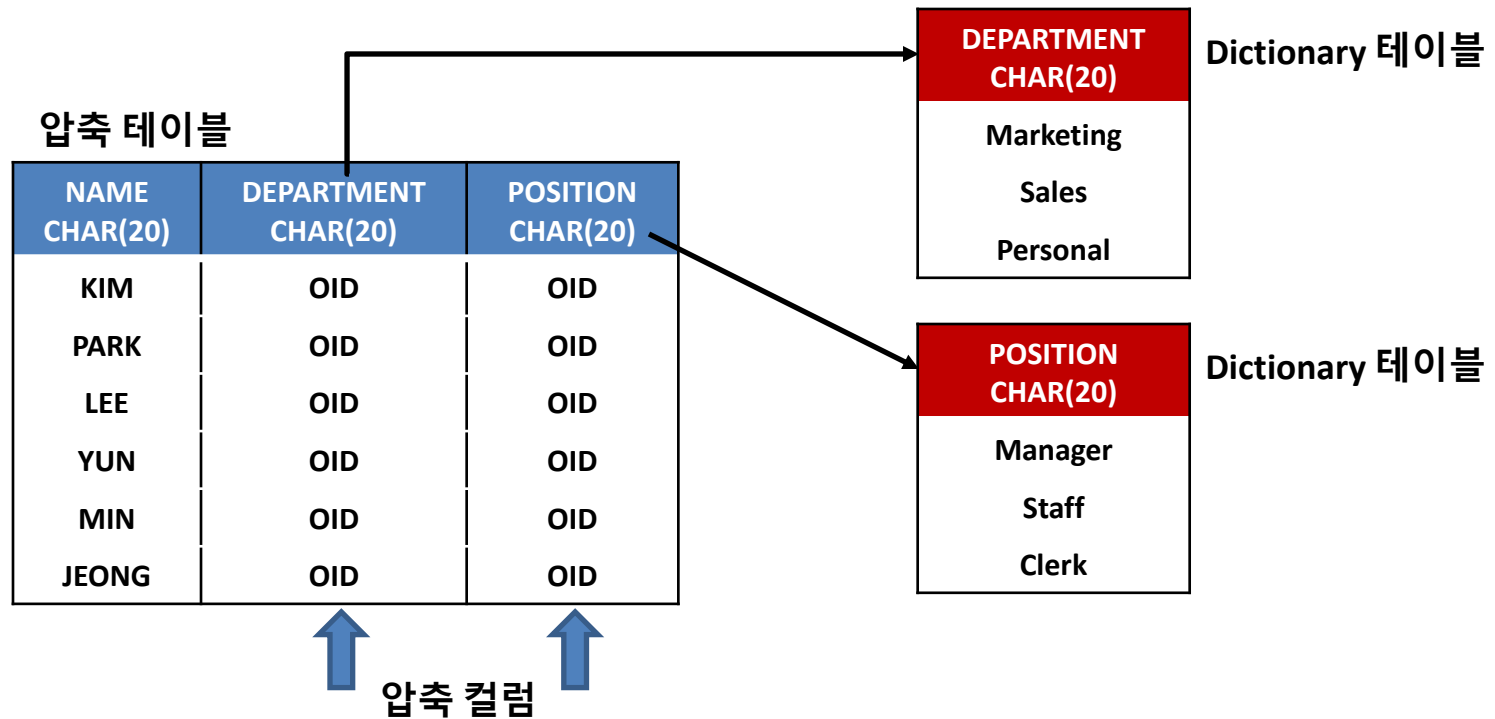
- 압축 컬럼
 - 중복된 데이터를 디셔너리 테이블에 별도로 저장하고, 그 데이터를 가리키는 OID로 대체
- 압축 테이블
 - 압축 컬럼이 속해 있는 테이블
 - 압축 컬럼을 포함하는 테이블 생성 시, SERVER에서 압축 컬럼 별로 디셔너리 테이블과 유니크 인덱스를 자동으로 생성 관리



COMPRESSION COLUMN

❖ COMPRESSION COLUMN 생성 예제

```
iSQL> CREATE TABLE employee(  
2 name CHAR(20),  
3 department CHAR(20),  
4 position CHAR(20)  
5 ) COMPRESS ( department, position MAXROWS 100 );
```



COMPRESSION COLUMN

❖ COMPRESSION COLUMN INSERT / UPDATE 시 유의사항

- Insert 하는 값이 Dictionary 테이블에 존재하는지 조회
 - 존재한다면 해당 OID 값 저장
 - 존재하지 않으면 Dictionary 테이블에 해당 값을 Insert 후, OID를 얻어 테이블에 저장 (성능저하 발생)

```
INSERT ( 'Yu', 'Sales', 'Manager' );
```

```
INSERT ( 'Yu', 'Consulting', 'Manager' );
```

압축 테이블

NAME CHAR(20)	DEPARTMENT CHAR(20)	POSITION CHAR(20)
KIM	OID	OID
PARK	OID	OID
LEE	OID	OID
YUN	OID	OID
MIN	OID	OID
JEONG	OID	OID
YU	OID	OID

DEPARTMENT CHAR(20)
Marketing
Sales
Personal
Consulting

Dictionary 테이블

POSITION CHAR(20)
Manager
Staff
Clerk

Dictionary 테이블

COMPRESSION COLUMN

❖ COMPRESSION COLUMN DELETE 시 유의사항

- DELETE 실행 후, 딕셔너리 테이블에 unpointed data 존재 가능
- 테이블 Reorganize 작업을 통해 unpointed data 삭제 가능

```
DELETE FROM EMPLOYEE WHERE  
NAME != 'JEONG';
```

압축 테이블

NAME CHAR(20)	DEPARTMENT CHAR(20)	POSITION CHAR(20)
JEONG	OID	OID
PARK	OID	OID
LEE	OID	OID
YUN	OID	OID
MIN	OID	OID
JEONG	OID	OID

DEPARTMENT CHAR(20)
Marketing
Sales
Personal

Dictionary 테이블

POSITION CHAR(20)
Manager
Staff
Clerk

Dictionary 테이블

COMPRESSION COLUMN

❖ COMPRESSION COLUMN REORGANIZE

- 테이블에 X Lock 을 획득 후, Reorganize 작업 실행
- 딕셔너리 테이블과 인덱스의 unpointed 데이터 삭제

압축 테이블

NAME CHAR(20)	DEPARTMENT CHAR(20)	POSITION CHAR(20)
JEONG	OID	OID

```
iSQL> ALTER TABLE EMPLOYEE  
REORGANIZE COLUMN ( DEPARTMENT, POSITION );
```

DEPARTMENT
CHAR(20)

Marketing ✖
Sales
Personal ✖

Dictionary 테이블

POSITION
CHAR(20)

Manager ✖
Staff
Clerk ✖

Dictionary 테이블

COMPRESSION COLUMN

❖ COMPRESSION COLUMN 추가

- ALTER TABLE ~ ADD COLUMN 구문을 사용하여 압축 컬럼 추가 가능
- 컬럼 추가 없이 기존 컬럼 압축 불가능(DROP COLUMN → ADD COLUMN)

ALTER TABLE EMPLOYEE

ADD COLUMN (JOIN_DATE CHAR(20)) **COMPRESS** (JOIN_DATE) ;

NAME CHAR(20)	DEPARTMENT CHAR(20)	POSITION CHAR(20)	JOIN_DATE CHAR(20)
KIM	Sales	Clerk	OID
PARK	Personal	Manager	OID
LEE	Marketing	Manager	OID
YUN	Sales	Clerk	OID
MIN	Marketing	Staff	OID
JEONG	Personal	Staff	OID

JOIN_DATE CHAR(20)
20120701
20131003

Dictionary 테이블

❖ 지원되는 DATA TYPE

- CHAR, VARCHAR, NCHAR, NVARCHAR, BYTE, BIT, VARBIT, NIBBLE, DATE
- CHAR, VARCHAR 의 경우 precision이 8 이상인 경우에만 압축 허용

NEW FEATURE

HYBRID PARTITIONED TABLE

HYBRID PARTITIONED TABLE

※ 7.1.0 버전부터 제공

❖ 개념

- ▶ 파티션드 테이블이 저장공간(디스크/메모리)에 관계 없이 다른 테이블 스페이스로 변환 가능
- ▶ 디스크 파티션드 테이블에서 메모리/휘발성 파티션드 테이블로 변환 될 때 컬럼의 타입을 fixed 또는 variable로 지정하더라도 fixed 처리
- ▶ 메모리 파티션드 테이블을 디스크 파티션드 테이블로 변환될 때는 타입이 동일하게 적용

❖ 제약 사항

- ▶ 글로벌 인덱스는 하이브리드 파티션드 테이블 지원 안함
- ▶ 하이브리드 파티션 테이블은 Direct Key Index와 CONJOIN, DISJOIN 불가능
- ▶ ALTER TABLE ~ MODIFY COLUMN 구문 불가능

HYBRID PARTITIONED TABLE

❖ 생성 예제

```
CREATE TABLE part_table
(
  sales_date DATE,
  sales_id NUMBER,
  sales_city VARCHAR(20)
)
PARTITION BY LIST(sales_city)
(
  PARTITION part_1 VALUES ( 'SEOUL' , 'INCHEON' ) TABLESPACE mem_tbs_0,
  PARTITION part_2 VALUES ( 'PUSAN' , 'JUNJU' ) TABLESPACE disk_tbs_1,
  PARTITION part_3 VALUES ( 'CHUNGJU' , 'DAEJUN' ) TABLESPACE
mem_tbs_2,
  PARTITION part_def VALUES DEFAULT TABLESPACE disk_tbs_3
)
TABLESPACE mem_tbs_4;
```

```
iSQL> SELECT U.USER_NAME
      , T.TABLE_NAME
      , P.PARTITION_NAME
      , TBS.NAME TABLESPACE_NAME
FROM SYSTEM_.SYS_TABLES_ T
      , SYSTEM_.SYS_TABLE_PARTITIONS_ P
      , SYSTEM_.SYS_PART_TABLES_ PM
      , SYSTEM_.SYS_USERS_ U
      , V$TABLESPACES TBS
```

```
WHERE 1=1
      AND TBS.ID=P.TBS_ID
      AND U.USER_ID = T.USER_ID
      AND T.TABLE_ID = P.TABLE_ID
      AND PM.TABLE_ID = T.TABLE_ID
```

```
;
```

USER_NAME	TABLE_NAME	PARTITION_NAME	TABLESPACE_NAME
-----------	------------	----------------	-----------------

SYS	PART_TABLE	PART_1	MEM_TBS_0
SYS	PART_TABLE	PART_2	DISK_TBS_1
SYS	PART_TABLE	PART_3	MEM_TBS_2
SYS	PART_TABLE	PART_DEF	DISK_TBS_3

4 rows selected.

HYBRID PARTITIONED TABLE

❖ 변경 예제

```
ALTER TABLE part_table ALTER PARTITION part_1 TABLESPACE  
disk_tbs_0;
```

```
ALTER TABLE part_table ALTER PARTITION part_2 TABLESPACE  
mem_tbs_1;
```

```
ALTER TABLE part_table ALTER PARTITION part_3 TABLESPACE  
disk_tbs_2;
```

```
ALTER TABLE part_table ALTER PARTITION part_def TABLESPACE  
mem_tbs_3;
```

```
SQL> SELECT U.USER_NAME  
      , T.TABLE_NAME  
          , P.PARTITION_NAME  
          , TBS.NAME TABLESPACE_NAME  
FROM SYSTEM_SYS_TABLES_T  
      , SYSTEM_SYS_TABLE_PARTITIONS_P  
      , SYSTEM_SYS_PART_TABLES_PM  
      , SYSTEM_SYS_USERS_U  
      , V$TABLESPACES TBS  
WHERE 1=1  
      AND TBS.ID=P.TBS_ID  
      AND U.USER_ID = T.USER_ID  
      AND T.TABLE_ID = P.TABLE_ID  
      AND PM.TABLE_ID = T.TABLE_ID  
;  
USER_NAME      TABLE_NAME      PARTITION_NAME  
TABLESPACE_NAME
```

```
-----  
SYS      PART_TABLE      PART_1      DISK_TBS_0  
SYS      PART_TABLE      PART_2      MEM_TBS_1  
SYS      PART_TABLE      PART_3      DISK_TBS_2  
SYS      PART_TABLE      PART_DEF     MEM_TBS_3
```

4 rows selected.

NEW FEATURE

IPCDA PROTOCOL

❖ 개념

- IPCDA(Inter Process Communication Direct Attach)
- 공유 메모리를 이용하여 CLIENT와 데이터베이스 SERVER 간 데이터 교환
- 데이터 읽기, 쓰기를 단순화
- CLIENT와 SERVER 사이의 유휴 시간(idle time)을 줄여 성능 향상

❖ 제약 사항

- CLI, ODBC 만 지원(JDBC 미지원)
- LOB 데이터 사용 불가능
- Linux 운영 체제에서만 사용 가능

IPCD A PROTOCOL

❖ 설정 프로퍼티

- IPCDA_CHANNEL_COUNT
 - CLIENT와 SERVER 간 통신 채널의 최대 개수 지정(default : 0)
 - ◆ 값의 범위 : 0 ~ 65535
 - 채널 개수에 비례하여 공유 메모리와 세마포어 할당
 - CPU 코어 개수의 1/2 값을 권고
 - 적용하기 위해 데이터베이스 SERVER 재기동 필요
- IPCDA_DATABLOCK_SIZE(단위 : kb)
 - 통신 채널 한 개의 공유 메모리 크기 설정(default : 20480)
 - ◆ 값의 범위 : 32 ~ 102400
 - 시스템 메모리 크기에 따라 공유 메모리를 사용하는 다른 프로그램 운영에 장애가 되지 않도록 적절한 값 설정
 - $IPCDA_CHANNEL_COUNT * IPCDA_DATABLOCK_SIZE$ 로 계산
 - ◆ $IPCDA_CHANNEL_COUNT$ 가 24이고, $IPCDA_DATABLOCK_SIZE$ 가 102400 kb 인 경우
 $24 * 102400 \text{ kb} = 2457600 \text{ kb}$ (약 2.4 gb)

NEW FEATURE

ROLE

ROLE

※ 6.5.1 버전부터 제공

❖ 개념

- 권한(Privilege)들의 묶음
- 여러 개의 권한을 사용자들에게 부여할 때 롤을 사용하는 것이 용이
- 처음 생성한 롤에는 아무런 권한이 없음
- GRANT 구문을 사용하여 롤에 권한들을 추가하고, 롤을 사용자에게 부여

❖ 생성 구문

```
CREATE ROLE role_name ;
```

❖ 생성 예제

- 이름이 `alti_role` 인 롤 생성

```
iSQL> CREATE ROLE alti_role;  
Create success.
```

ROLE

❖ 룰에 권한 부여 예제

- 이름이 alti_role인 룰에게 create user, drop user 시스템 권한 부여

```
iSQL> GRANT CREATE USER, DROP USER TO alti_role;
Grant success.
iSQL> CREATE USER user01 IDENTIFIED BY user01;
Create success.
iSQL> GRANT alti_role to user01;
Grant success.
iSQL> CONNECT user01/user01
Connect success.
iSQL> CREATE USER user02 IDENTIFIED BY user02;
Create success.
iSQL> DROP USER user02;
Drop success.
```

ROLE

❖ 삭제 구문

```
DROP ROLE role_name ;
```

❖ 삭제 예제

- 이름이 `alti_role`인 롤 제거

```
iSQL> DROP ROLE alti_role;  
Drop success.
```

Thank you!