

ALTIBASE HDB ADMINISTRATION II



ALTIBASE HDB ADMINISTRATION II

MONITORING

TRACE FILES

META TABLES

PERFORMANCE VIEWS

통계정보 수집



TRACE FILES

모니터링 항목

❖ 모니터링 항목

정상 시 혹은 문제가 발생했을 때 확인 해야 하는 항목

➤ trace file

- 알티베이스가 구동 중 발생하는 여러 사항들에 대한 trace 정보들이 기록된 파일
- \$ALTIBASE_HOME/trc 디렉토리에 위치

➤ meta table

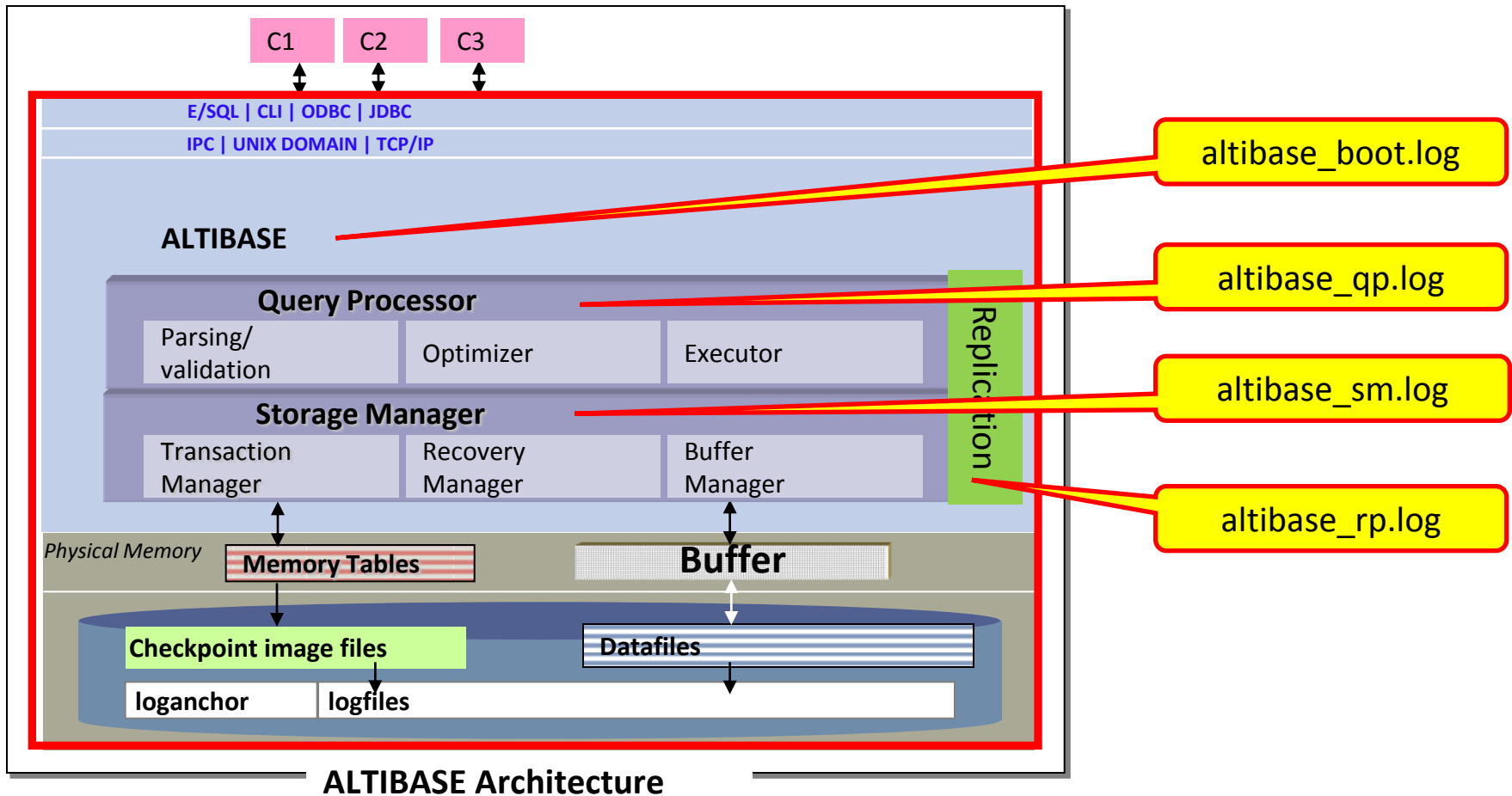
- 데이터베이스 객체에 관한 모든 정보를 기록하기 위한 시스템 정의 테이블
- 소유자는 SYSTEM_
- DDL 수행 시 시스템에 의해 변경

➤ performance view

- 시스템 메모리, 프로세스 상태, 세션, 버퍼 등의 최신 정보를 제공
- read only

TRACE FILES

❖ Trace file 종류



TRACE FILES

❖ Trace file 종류

파일이름	주요 내용
altibase_boot.log	Altibase HDB가 동작된 상태를 기록 구동 및 종료에 관련된 각종 step 정보를 기록 구동 중 디스크 부족으로 인한 에러 등 여러 가지 에러상황을 기록 비정상 종료가 발생할 경우 Call-Stack 정보
altibase_rp.log	Replication 모듈에서 발생하는 경고 메시지나 트레이스 메시지 이중화 상태, 이중화 conflict 등의 메시지
altibase_sm.log	Storage Manager 모듈에서 발생하는 경고 메시지나 트레이스 메시지 checkpoint, 백업 등의 메시지
altibase_qp.log	Query Processor 모듈에서 발생하는 경고 메시지나 트레이스 메시지 등이 기록 DDL 문의 성공/실패 메시지



META TABLES

META TABLE

❖ 주요 meta table

분류	관련 메타 테이블
사용자 관련	SYS_USERS_ SYS_TBS_USERS_
테이블 관련	SYS_TABLES_ SYS_COLUMNS_ SYS_CONSTRAINTS_ SYS_CONSTRAINT_COLUMNS_
인덱스 관련	SYS_INDICES_ SYS_INDEX_COLUMNS_
권한 관련	SYS_GRANT_OBJECT_ SYS_GRANT_SYSTEM_ SYS_PRIVILEGES_
뷰 관련	SYS_VIEWS_ SYS_VIEW_PARSE_ SYS_VIEW_RELATED_
트리거 관련	SYS_TRIGGERS_ SYS_TRIGGER_DML_TABLES_ SYS_TRIGGER_STRINGS_

META TABLE

❖ SYS_USERS_

DB에 생성된 사용자 정보

컬럼	설명
USER_ID	사용자의 고유번호
USER_NAME	사용자의 이름
PASSWORD	암호화된 패스워드
LAST_DDL_TIME	사용자 정보에 최종적으로 변경이 수행된 시각
DEFAULT_TBS_ID	TABLESPACE절을 명시하지 않은 테이블 생성 시 사용되는 기본 테이블스페이스의 고유번호

META TABLE

❖ SYS_TABLES_

DB에 생성된 테이블, 뷰, 시퀀스, 큐 정보

컬럼	설명
USER_ID	테이블이 속한 사용자의 고유번호
TABLE_ID	테이블의 고유번호
TABLE_OID	내부적으로 사용되는 테이블의 고유 참조번호
COLUMN_COUNT	테이블 내의 컬럼의 총 개수
TABLE_NAME	테이블의 이름
TABLE_TYPE	T: (Table), V: (View), Q:(Queue), S:(Sequence)
MAXROW	0: (무제한), 기타: (최대로 저장될 수 있는 레코드 개수)
TBS_ID	테이블이 저장되어 있는 테이블스페이스의 고유번호
LAST_DDL_TIME	테이블에 변경이 수행된 최종 시각

META TABLE

❖ 사용 예

<전제>

'SYS' 사용자 소유의 테이블 목록을 보고자 한다.

<질의>

```
iSQL> SELECT a.user_name, b.table_name , b.table_type  
2 FROM system_.sys_users_ a, system_.sys_tables_ b  
3 WHERE a.user_id = b.user_id  
4 AND a.user_name = 'SYS' ;
```

USER_NAME	TABLE_NAME	TABLE_TYPE
-----------	------------	------------

SYS	PERSON	T
SYS	CUSTOMER	T
SYS	PERSON_SEQ	S
SYS	GOODS	T
SYS	DEPARTMENT	T
SYS	ORDERS	T
.....		

META TABLE

❖ SYS_COLUMNS_

테이블, 뷰, 시퀀스에 정의된 컬럼 정보

컬럼	설명
COLUMN_ID	컬럼 식별자
DATA_TYPE	CHAR: 1, VARCHAR: 12, NUMERIC: 2, DATE: 9, BLOB:30, CLOB:40, ...
TABLE_ID	테이블 식별자
SIZE	컬럼의 물리적 길이
PRECISION	지정한 PRECISION
SCALE	지정한 SCALE, 숫자타입만 저장됨
COLUMN_NAME	컬럼의 이름
IS_NULLABLE	T: NULL 허용, F: NULL 불허
DEFAULT_VALUE	테이블 생성 시 지정한 기본 값

META TABLE

❖ 사용 예

<전제>

T1 테이블의 컬럼 정보와 해당 컬럼의 기본 값을 보고자 한다.

<질의>

```
iSQL> SELECT column_id, column_name, data_type, default_val, is_nullable  
2 FROM system_.sys_columns_  
3 WHERE table_id = (SELECT table_id FROM system_.sys_tables_ WHERE table_name='T1');
```

```
COLUMN_ID COLUMN_NAME DATA_TYPE DEFAULT_VAL IS_NULLABLE
```

```
-----  
2270208 C1 2 F  
2270209 C2 1 T  
2270210 C3 9 SYSDATE T  
...
```

META TABLE

❖ SYS_INDICES_ 인덱스 정보

컬럼	설명
INDEX_ID	인덱스 식별자
INDEX_NAME	인덱스 이름
TABLE_ID	테이블 식별자
IS_UNIQUE	UNIQUE 인덱스 인지
COLUMN_CNT	인덱스를 구성하는 컬럼의 개수
TBS_ID	테이블스페이스 식별자
LAST_DDL_TIME	마지막으로 DDL 변경이 일어난 시간
CREATED	인덱스가 생성된 시간

META TABLE

❖ SYS_INDEX_COLUMNS_ 인덱스 구성 컬럼의 정보

컬럼	설명
INDEX_ID	인덱스 식별자
COLUMN_ID	컬럼 식별자
TABLE_ID	테이블 식별자
INDEX_COL_ORDER	인덱스 구성 칼럼의 순서
SORT_ORDER	A: 오름차순, B: 내림차순

META TABLE

❖ 사용 예

<전제>

T1 테이블에 생성된 인덱스와 칼럼의 정보를 보고자 한다.

<질의>

```
iSQL> SELECT a.index_name, b.index_col_order, c.column_name,  
2 DECODE(b.sort_order,'A','ASC','D','DESC') AS sort_order  
3 FROM system_.sys_indices_ a, system_.sys_index_columns_ b, system_.sys_columns_ c  
4 WHERE a.index_id = b.index_id  
5 AND b.column_id = c.column_id  
6 AND a.table_id = (SELECT table_id FROM system_.sys_tables_ WHERE table_name='T1')  
7 ORDER BY 1, 2;
```

INDEX_NAME	INDEX_COL_ORDER	COLUMN_NAME	SORT_ORDER
T1_IDX1	0	C1	ASC
T1_IDX1	1	C2	DESC

META TABLE

❖ SYS_CONSTRAINTS_ 제약 조건 정보

컬럼	설명
CONSTRAINT_ID	제약조건 식별자
CONSTRAINT_NAME	제약조건 이름
CONSTRAINT_TYPE	제약조건 타입 0: FOREIGN KEY, 1: NOT NULL, 2: UNIQUE, 3: PRIMARY KEY, 4: NULL
TABLE_ID	테이블 식별자
INDEX_ID	제약조건의 인덱스 식별자
COLUMN_CNT	제약조건의 칼럼 개수
REFERENCED_TABLE_ID	parent table의 식별자
DELETE_RULE	Delete CASCADE 동작 여부 0: No action, 1: Delete CASCADE

META TABLE

❖ **SYS_CONSTRAINT_COLUMNS_** 제약 조건에 구성된 컬럼의 정보

컬럼	설명
CONSTRAINT_ID	제약조건 식별자
CONSTRAINT_COL_ORDER	제약 조건 컬럼 순서
COLUMN_ID	컬럼 식별자

META TABLE

❖ 사용 예

<전제>

T1 테이블에 생성된 제약조건과 컬럼의 정보를 보고자 한다.

<질의>

```
iSQL> SELECT a.constraint_name, b.constraint_col_order, c.column_name,  
2 DECODE(a.constraint_type,0,'forieign',1,'not null',2,'unique',3,'primary',4,'null') AS constraint_type  
3 FROM system_.sys_constraints_ a, system_.sys_constraint_columns_ b, system_.sys_columns_ c  
4 WHERE a.constraint_id = b.constraint_id  
5 AND b.column_id = c.column_id  
6 AND a.table_id = (SELECT table_id FROM system_.sys_tables_ WHERE table_name='T1')  
7 ORDER BY 1, 2;
```

CONSTRAINT_NAME	CONSTRAINT_COL_ORDER	COLUMN_NAME	CONSTRAINT_TYPE
-----------------	----------------------	-------------	-----------------

__SYS_CON_NOT_NULL_ID_207	0	C3	not null
__SYS_CON_PRIMARY_ID_208	0	C1	primary
__SYS_CON_PRIMARY_ID_208	1	C2	primary



PERFORMANCE VIEWS

Performance View

❖ Performance View

사용자가 질의를 통해 쉽게 조회할 수 있도록 테이블처럼 만든 형태

- performance view 목록을 조회하는 방법

```
iSQL> SELECT * FROM v$tab;
```

- 주요 Performance view

분류	연관 Performance View
Session 관련	v\$session
Statement 관련	v\$statement, v\$sqltext
Lock 관련	v\$lock, v\$lock_wait
Transaction 관련	v\$transaction, v\$memgc
Property 관련	v\$property
Replication 관련	v\$repgap, v\$repsender, v\$repreceiver
Resource 관련	v\$memstat, v\$memtbl_info, v\$tablespaces, v\$mem_tablespaces, v\$vol_tablespaces, v\$segment, v\$disktbl_info, v\$lfg, v\$buffpool_stat, v\$datafiles, v\$sysstat, v\$sesstat

Performance View

❖ V\$SESSION

DB에 접속 중인 세션의 정보

컬럼	설명
ID	세션의 고유번호
CLIENT_PID	사용자 프로그램의 process id
TRANS_ID	마지막 수행 중인 트랜잭션의 고유번호
COMM_NAME	접속한 방식, IP 주소, 포트번호
AUTOCOMMIT_FLAG	AUTOCOMMIT 여부 (0: NON-AUTOCOMMIT)
DB_USERNAME	세션을 사용중인 DB User
CURRENT_STMT_ID	마지막 수행 중인 질의문의 고유번호

Performance View

❖ V\$STATEMENT

접속이 유효한 세션에서 실행한 최종 질의의 수행정보

컬럼	설명
ID	세션별로 수행한 질의문의 고유번호
SESSION_ID	세션의 고유번호
TX_ID	해당 질의가 속해있는 트랜잭션의 고유번호
QUERY	실제 질의 문장 (16Kbyte까지만 표현)
TOTAL_TIME	질의가 최종 수행된 시점의 전체 소요시간
EXECUTE_TIME	질의가 최종 수행시점에 DBMS내부에서 처리된 소요시간 (DB의 SM모듈에서 수행된 시간)
FETCH_TIME	최종 질의에 대해 DB와 프로그램과 발생한 통신의 누적 소요시간 (통신 비용의 합)
EXECUTE_SUCCESS	동일한 질의가 성공적으로 수행된 누적횟수
EXECUTE_FAILURE	동일한 질의가 실패한 누적횟수

Performance View

❖ V\$SQLTEXT

질의문 전체를 저장

컬럼	설명
SID	세션의 고유번호
STMT_ID	세션별 질의문의 고유번호
PIECE	질의문의 조각 순서 (64byte단위로 나누어 저장되어 있음)
TEXT	실제 질의문의 내용

Performance View

❖ 특정 세션에 수행한 질의를 찾고자 할 경우

<전제>

프로그램의 PID는 1234 이다. 이 프로그램이 수행 중인 SQL문과 처리시각을 보려고 한다.

<질의>

```
iSQL> SELECT a.query , a.execute_time, a.fetch_time
      2 FROM v$statement a, v$session b
      3 WHERE a.session_id = b.id
      4 AND  b.client_pid = 1234;
```

❖ 수행된 횟수가 가장 많고, 처리시간이 긴 질의를 찾는 경우

<질의>

```
iSQL> SELECT a.client_pid, a.comm_name, b.execute_success, b.execute_time, .bquery,
      2 FROM v$session a, v$statement b
      3 WHERE a.id = b.session_id
      4 ORDER BY b.execute_success DESC, b.execute_time DESC
      5 LIMIT 1;
```

Performance View

❖ V\$LOCK

테이블 단위의 Lock 정보

컬럼	설명
LOC_ITEM_TYPE	LOCK이 획득 된 대상의 종류 (TBL, TBS)
TBS_ID	LOCK이 획득 된 대상이 속한 테이블스페이스 고유번호
TABLE_OID	LOCK이 획득 된 테이블의 고유번호
TRANS_ID	트랜잭션의 고유번호
LOCK_DESC	LOCK의 유형 (X_LOCK, IX_LOCK, IS_LOCK)
IS_GRANT	LOCK을 획득했는지 여부 (1: 획득)

Performance View

❖ V\$LOCK_WAIT

Lock의 상호 대기 정보

컬럼	설명
TRANS_ID	트랜잭션의 고유번호
WAIT_FOR_TRANS_ID	TRANS_ID가 대기하고 하는 트랜잭션의 고유번호

```
iSQL> SELECT * FROM v$lock_wait;  
TRANS_ID      WAIT_FOR_TRANS_ID  
-----  
17856         139648
```

17856 트랜잭션은 139648 트랜잭션이 완료되기를 기다리고 있다는 의미임.

Performance View

❖ 내가 발생시킨 트랜잭션을 기다리고 있는 트랜잭션의 개수는?

<전제>

자신의 세션ID를 리턴하는 함수는 SESSION_ID() 라는 함수로 제공된다.

<질의>

```
iSQL> SELECT COUNT(*)  
2 FROM v$sqllock_wait  
3 WHERE wait_for_trans_id IN  
4     ( SELECT tx_id FROM v$sqlstatement  
5       WHERE session_id = SESSION_ID() )
```

Performance View

❖ V\$PROPERTY

DBMS내에 설정된 프로퍼티 정보

컬럼	설명
NAME	프로퍼티의 명칭
MIN	설정할 수 있는 최소값
MAX	설정할 수 있는 최대값
VALUE1	현재 설정된 상태값

```
iSQL> SELECT value1 FROM v$property WHERE name LIKE '%DB_NAME%';
```

```
-----  
mydb
```

```
iSQL> SELECT value1 FROM v$property WHERE name = 'PORT_NO';
```

```
-----  
20302
```

Performance View

❖ V\$TABLESPACES

테이블스페이스에 대한 정보

칼럼	설명
ID	테이블스페이스 ID
NAME	테이블스페이스 이름
TYPE	테이블스페이스 타입
STATE	테이블스페이스 상태
SEGMENT_MANAGEMENT	테이블스페이스에서 세그먼트를 생성할 때 어떤 타입으로 생성할 것인지 나타냄
TOTAL_PAGE_COUNT	테이블스페이스의 전체 페이지 개수
ALLOCATE_PAGE_COUNT	테이블스페이스에 할당된 페이지 개수
PAGE_SIZE	테이블스페이스의 페이지 크기
EXTENT_PAGE_COUNT	extent의 페이지 개수

Performance View

❖ V\$DATAFILES

디스크 테이블스페이스의 데이터파일에 대한 정보
주로 데이터파일의 사용량을 확인하고 싶을 때 사용

칼럼	설명
ID	데이터파일 ID
NAME	데이터파일 경로와 이름
SPACEID	데이터파일이 속한 테이블스페이스 ID
MAXSIZE	데이터파일 생성 시 지정한 MAXSIZE. AUTOEXTEND OFF 이면 0
INITSIZE	데이터파일 생성 시 지정한 SIZE
NEXTSIZE	데이터파일 생성 시 지정한 NEXT AUTOEXTEND OFF 이면 0
CURRSIZE	데이터파일의 현재 크기 AUTOEXTEND OFF이면 INITSIZE와 같음
AUTOEXTEND	0: OFF, 1: ON
STATUS	1: 오프라인, 2: 온라인, 4: 백업시작, 8: 백업종료, 128: 삭제(dropped)

Performance View

❖ disk tablespace의 데이터파일 별 사용량 확인

<질의>

```
iSQL> SELECT b.name tbs_name, a.id 'FILE#', a.name datafile_name,  
2         currsz*8/1024 'ALLOC(M)',  
3         ROUND(CASE2(a.maxsz=0, currsz, a.maxsz)*8/1024) 'MAX(M)',  
4         DECODE(autoextend, 0, 'OFF', 'ON') 'AUTOEXTEND'  
5 FROM v$datafiles a,  
6      v$tablespaces b  
7 WHERE b.id = a.spaceid  
8 ORDER BY b.name, a.id;
```

•ALLOC(M) - 데이터파일의 현재 크기

•MAX(M) - 데이터파일이 AUTOEXTEND ON으로 생성되었을 경우에 확장될 수 있는 최대 크기
AUTOEXTEND OFF 로 생성되었다면 생성 시 크기

Performance View

❖ V\$MEM_TABLESPACES

메모리 테이블스페이스에 대한 정보

메모리 테이블스페이스의 사용량을 확인하고 싶을 때 사용

칼럼	설명
SPACE_ID	테이블스페이스 ID
SPACE_NAME	테이블스페이스 이름
MAXSIZE	테이블스페이스 최대 크기 DECODE(MAXSIZE, 0, ALLOC_PAGE_COUNT*PAGE_SIZE, MAXSIZE)
ALLOC_PAGE_COUNT	테이블스페이스의 전체 페이지 개수
FREE_PAGE_COUNT	테이블스페이스의 free 페이지 개수

Performance View

❖ V\$VOL_TABLESPACES

휘발성 테이블스페이스에 대한 정보

휘발성 테이블스페이스의 사용량을 확인하고 싶을 때 사용

칼럼	설명
SPACE_ID	테이블스페이스 ID
SPACE_NAME	테이블스페이스 이름
MAXSIZE	테이블스페이스 최대 크기 DECODE(MAXSIZE, 0, ALLOC_PAGE_COUNT*PAGE_SIZE, MAXSIZE)
ALLOC_PAGE_COUNT	테이블스페이스의 전체 페이지 개수
FREE_PAGE_COUNT	테이블스페이스의 free 페이지 개수

Performance View

❖ 메모리 테이블스페이스의 사용량 조회

<질의>

```
iSQL> SELECT space_id, space_name, autoextend_mode,  
2          DECODE(maxsize, 140737488322560, 'UNDEFINED',  
3              0, alloc_page_count*32/1024,  
4              maxsize/1024/1024) 'MAX(M)',  
5          alloc_page_count * 32/1024 'TOTAL(M)',  
6          (alloc_page_count-free_page_count)*32/1024 'ALLOC(M)',  
7          ((alloc_page_count-free_page_count)*32/1024  
8          /DECODE(maxsize, 0, alloc_page_count*32/1024, maxsize/1024/1024) 'USAGE(%)'  
9 FROM v$mem_tablespaces;
```

- MAX(M) - 메모리 테이블스페이스 maxsize. 테이블스페이스 생성 시 지정
- TOTAL(M) - 메모리 테이블스페이스가 현재까지 할당 받은 크기
- ALLOC(M) - 메모리 테이블스페이스가 현재까지 할당 받은 페이지 중 빈 페이지를 제외한 사용 공간
- USAGE(%) - 메모리 테이블스페이스가 최대로 할당할 수 있는 크기 대비 사용중인 공간에 대한 백분율

Performance View

❖ V\$MEMTBL_INFO

메모리 테이블 정보

어느 테이블이 메모리를 많이 사용하는지 확인하고 싶을 때 사용

칼럼	설명
TABLESPACE_ID	테이블스페이스 ID
TABLE_OID	테이블 식별자. sys_tables_와 조인하여 table_name 확인
FIXED_ALLOC_MEM	테이블에서 할당한 고정 영역의 메모리 크기
FIXED_USED_MEM	테이블에서 실제 사용하고 있는 고정 영역의 메모리 크기
VAR_ALLOC_MEM	테이블에서 할당한 가변 영역의 메모리 크기
VAR_USED_MEM	테이블에서 실제 사용하고 있는 가변 영역 메모리 크기
UNIQUE_VIOLATION_COUNT	UNIQUE 제약조건이 위반된 횟수
UPDATE_RETRY_COUNT	UPDATE 시 재시도한 횟수
DELETE_RETRY_COUNT	DELETE 시 재시도한 횟수

- 할당공간은 (FIXED_ALLOC_MEM+VAR_ALLOC_MEM),
사용공간은 (FIXED_USED_MEM+VAR_USED_MEM)을 질의

Performance View

❖ 휘발성 테이블스페이스와 메모리 테이블스페이스에 속한 테이블의 실제 사용 메모리를 확인

<질의>

```
iSQL> SELECT a.user_name, b.table_name, d.name tablespace_name,  
2         (c.fixed_alloc_mem + c.var_alloc_mem)/(1024*1024) 'ALLOC(M)',  
3         (c.fixed_used_mem + c.var_used_mem)/(1024*1024) 'USED(M)',  
4         (c.fixed_used_mem + c.var_used_mem)/(c.fixed_alloc_mem + c.var_alloc_mem)*100  
5         'EFFICIENCY(%)'  
6 FROM system_.sys_users_ a, system_.sys_tables_ b, v$membtbl_info c, v$tablespaces d  
7 WHERE a.user_name <> 'SYSTEM_'  
8 AND b.table_type = 'T'  
9 AND a.user_id = b.user_id  
10 AND b.table_oid = c.table_oid  
11 AND b.tbs_id = d.id  
12 ORDER BY 4 DESC ;
```

- ALLOC(M) - 테이블이 할당 받은 메모리 합계
- USED(M) - 테이블이 할당 받은 페이지중에서 “실제로 데이터가 적재된 페이지”의 메모리 합계
예를 들어, ALLOC이 100M 크기인 메모리 테이블에 전체 DELETE를 수행하면 ALLOC은 100M로 변함없으나 USED는 0에 가깝게 됨
- EFFICIENCY(%) - 테이블이 소유한 페이지중에서 “실제로 데이터가 적재된 페이지”에 대한 백분율로 공간 효율성을 나타냄

Performance View

❖ V\$DISKTBL_INFO

디스크 테이블 정보

어느 테이블이 디스크를 많이 사용하는지 확인하고 싶을 때 사용

칼럼	설명
TABLESPACE_ID	테이블스페이스 ID
TABLE_OID	테이블 식별자. sys_tables_와 조인하여 table_name 확인
DISK_PAGE_CNT	테이블에서 데이터를 갖고 있는 페이지 개수

Performance View

❖ 디스크 테이블의 사용량 확인

<질의>

```
iSQL> SELECT user_name, a.table_name, d.name tbs_name,  
2         ROUND((b.disk_page_cnt*8)/1024) 'ALLOC(M)'  
3 FROM system_.sys_tables_ a, v$disktbl_info b, system_.sys_users_ c, v$tablespaces d  
4 WHERE a.table_oid = b.table_oid  
5 AND a.user_id = c.user_id  
6 AND a.tbs_id=d.id  
7 AND c.user_name <> 'SYSTEM_'
```

Performance View

❖ V\$MEMSTAT

ALTIBASE 내부 모듈들이 사용하는 메모리 사용량 정보

컬럼	설명
NAME	메모리를 할당하여 사용하는 모듈의 명칭
ALLOC_SIZE	현재 할당된 메모리의 사용량
MAX_TOTAL_SIZE	DB구동 이후 최대로 사용했던 메모리의 크기

- V\$MEMSTAT의 정보는 적절한 주기를 가지고 모니터링하여 메모리의 증가현상이 발생할 때 ALTIBASE HDB에 의한 문제일 경우 어떤 모듈에 의해 메모리가 증가했는지를 추적하는데 유용한 정보로 활용할 수 있음

가장 많은 메모리를 사용했던 모듈을 5개만 출력하라

```
iSQL> SELECT name, max_total_size, alloc_size
2 FROM v$memstat
3 ORDER BY max_total_size
4 LIMIT 5;
```


Performance View

❖ V\$SYSSTAT

DB의 전체 세션들의 활동을 누적한 통계정보

컬럼	설명
SEQNUM	통계항목의 고유번호
NAME	통계항목의 명칭
VALUE	현재까지 누적된 수치

➤ 주기적인 모니터링을 통해 성능이슈에 대한 분석정보로 활용

현재 질의수행의 성공, 실패 누적횟수를 출력하라

```
iSQL> SELECT name, value  
2 FROM v$sysstat  
3 WHERE seqnum IN (25, 26);
```

```
-----  
execute success count  2039404  
execute failure count  201
```

Performance View

❖ V\$LFG

로그파일이 미처 생성되지 못해 트랜잭션이 로그파일이 생성될 때까지 기다렸는지 확인할 때 사용

칼럼	설명
CUR_WRITE_LF_NO	기록 로그 파일번호
LF_PREPARE_COUNT	미리 생성한 로그파일의 수
LF_PREPARE_WAIT_COUNT	로그 스위치 시 대기한 횟수
END_LSN_FILE_NO	restart redo가 시작될 LSN의 파일번호
END_LSN_OFFSE	restart redo가 시작될 LSN의 오프셋

Performance View

❖ logfile switch 시 트랜잭션이 얼마나 대기했는지 확인

```
<질의>  
iSQL> SELECT lf_prepare_wait_count  
         2 FROM v$log;  
LF_PREPARE_WAIT_COUNT  
-----  
0
```



통계정보 수집

통계정보 수집

❖ DBMS stat

- 데이터베이스에 대한 통계 자료를 수집하는 기능
- 시스템 저장 프로시저 형태로 제공
- SYS사용자만 수행할 수 있으며 이 저장 프로시저를 사용해서 통계자료를 수집하고 실행 계획을 재구축.

이름	설명
GATHER_SYSTEM_STATS	데이터베이스 시스템에 대한 통계 자료를 수집한다.
GATHER_DATABASE_STATS	모든 테이블에 대한 통계 자료를 수집한다.
GATHER_TABLE_STATS	특정 테이블에 대한 통계 자료를 수집한다.
GATHER_INDEX_STATS	특정 인덱스에 대한 통계 자료를 수집한다.

통계정보 수집

❖ GATHER_SYSTEM_STATS

데이터베이스 시스템에 대한 통계 자료를 수집한다.

➤ 구문

```
GATHER_SYSTEM_STATS( );
```

➤ 결과값

저장 프로시저이므로 결과값을 반환하지 않는다.

➤ 예제

```
iSQL> EXEC GATHER_SYSTEM_STATS ( );  
Execute success.
```

통계정보 수집

❖ GATHER_DATABASE_STATS

데이터베이스에 존재하는 모든 테이블에 대한 통계 자료를 수집한다.

➤ 구문

```
EXEC GATHER_DATABASE_STATS (  
    estimate_percent  FLOAT  DEFAULT  NULL,  
    degree             INTEGER DEFAULT  NULL,  
    no_invalidate     BOLLEAN DEFAULT FALSE);
```

➤ 결과값

저장 프로시저이므로 결과값을 반환하지 않는다.

➤ 예제

```
iSQL> EXEC GATHER_DATABASE_STATS ();  
SYSTEM_.SYS_TABLES_  
SYSTEM_.SYS_COLUMNS_  
SYSTEM_.SYS_DATABASE_  
SYSTEM_.SYS_USERS_  
.....  
Execute success.
```

통계정보 수집

❖ GATHER_TABLE_STATS

특정 테이블과 그 테이블에 정의된 인덱스들에 대한 통계 자료를 수집한다.

➤ 구문

```
EXEC GATHER_TABLE_STATS (  
    ownname      VARCHAR,  
    tablename    VARCHAR,  
    partname     VARCHAR DEFAULT NULL,  
    estimate_percent  FLOAT  DEFAULT NULL,  
    degree       INTEGER DEFAULT NULL,  
    no_invalidate  BOLLEAN DEFAULT FALSE);
```

➤ 결과값

저장 프로시저이므로 결과값을 반환하지 않는다.

➤ 예제

```
iSQL> EXEC GATHER_TABLE_STATS ( 'SYS', 'T1' );  
Execute success.
```


통계정보 수집

❖ GATHER_INDEX_STATS

특정 인덱스에 대한 통계 자료를 수집한다.

➤ 구문

```
EXEC GATHER_TABLE_STATS (  
    ownname      VARCHAR,  
    idxname      VARCHAR,  
    estimate_percent  FLOAT  DEFAULT NULL,  
    degree       INTEGER DEFAULT NULL,  
    no_invalidate  BOLLEAN DEFAULT FALSE);
```

➤ 결과값

저장 프로시저이므로 결과값을 반환하지 않는다.

➤ 예제

```
iSQL> EXEC GATHER_INDEX_STATS ( 'SYS', 'T1_IDX' );  
Execute success.
```



ALTIBASE HDB ADMINISTRATION II

BACKUP & RECOVERY

CHECKPOINT
BACKUP
RECOVERY

ALTIBASE

CHECKPOINT

CHECKPOINT

❖ 체크포인트 (Checkpoint)

- 데이터베이스에 대한 변경 및 현재 상황을 데이터 파일에 저장하는 것
- 체크포인트 완료 시의 처리 과정
 - 기존 복구시작 시점을 완료시점으로 변경
 - 데이터 파일에 반영이 완료된 불필요한 리두로그 파일을 삭제
- 주기적인 체크포인트는 시스템 장애 시 데이터베이스 복구시간을 줄일 수 있음

❖ ALTIBASE HDB 체크포인트의 특징

- fuzzy & ping-pong checkpoint
 - 체크포인트 수행 중에도 성능저하 없이 트랜잭션을 처리할 수 있음
 - 메모리 테이블스페이스에 한하여 데이터 파일을 두벌로 유지

CHECKPOINT

❖ 체크포인트 수행

➤ 자동수행

- 프로퍼티 CHECK_POINT_ENABLED에 의해 자동수행 여부를 우선적으로 설정
 - ◆ 시간주기 - CHECK_POINT_INTERVAL_IN_SEC
 - ◆ 리두로그 파일개수 - CHECK_POINT_INTERVAL_IN_LOG

➤ 수동수행

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

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

❖ 체크포인트 수행이 완료되더라도 리두로그 파일이 삭제되지 않는 경우

- 완료되지 않은 트랜잭션의 리두로그 파일인 경우
- 아카이브로그 모드에서, 아카이브 되지 않은 리두로그 파일인 경우
- 이중화로 전송되지 않은 리두로그 파일인 경우



BACKUP

데이터베이스 모드

❖ 데이터베이스 모드 (database mode)

- 체크포인트 완료 시 리두로그 파일을 관리하는 방식에 따라 두 가지로 분류
 - 아카이브로그 모드 (archivelog mode) - 지정된 경로로 축적(archiving)후 삭제하는 방식
 - 노아카이브로그 모드 (noarchivelog mode) - 삭제하는 방식
- 데이터베이스 생성구문(CREATE DATABASE)에 의해 최초로 지정됨
 - 데이터베이스 생성 이후에도 CONTROL 단계에서 변경은 가능하나 재구동 필요

❖ 데이터베이스 모드의 비교

데이터베이스 모드	특징	고려사항
아카이브로그 모드 (archivelog mode)	<ul style="list-style-type: none">▪ 매체복구(media recovery)가 가능.▪ 아카이브로그 파일의 손실이 없다면 현재시점까지 복구가 가능하다.	<ul style="list-style-type: none">▪ 아카이브로그 파일을 저장하기 위한 디스크 공간이 필요하다.▪ 아카이브로그 파일에 대한 DBA의 관리가 필요하다.
노아카이브로그 모드 (noarchivelog mode)	<ul style="list-style-type: none">▪ 아카이브로그 파일을 저장하기 위한 디스크 공간이 필요 없다.▪ 아카이브로그 파일에 대한 DBA의 관리가 필요 없다.	<ul style="list-style-type: none">▪ 매체복구가 불가능.▪ 오프라인 백업을 받은 시점까지만 복구가 가능하다.

데이터베이스 모드

❖ 현재 데이터베이스 모드 확인

- 성능뷰 v\$archive 를 통하여 보다 상세한 정보를 확인할 수 있음

```
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

- DB 구동 종료 후 CONTROL 단계에서 데이터베이스 모드를 변경
 - SYSDBA 자격의 사용자 "SYS"로만 수행이 가능

```
iSQL> connect sys/manager as sysdba  
iSQL(sysdba)> shutdown immediate;  
iSQL(sysdba)> startup control;  
iSQL(sysdba)> ALTER DATABASE ARCHIVELOG;  
iSQL(sysdba)> startup;
```


데이터베이스 백업

❖ 데이터베이스 백업 종류와 대상 객체

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

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

데이터베이스 백업

❖ 온라인 데이터베이스 백업

- DB 구동 중 특정 시점에 전체 데이터베이스를 백업
- 구문수행 시 모든 데이터 파일과 리두로그앵커 파일이 지정한 경로로 자동 백업 됨
 - 메모리 테이블스페이스관련 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';
```

데이터베이스 백업

❖ 온라인 테이블스페이스 백업

- DB 구동 중 특정 테이블스페이스를 백업
- 구문수행 시 테이블스페이스와 관련된 데이터 파일이 지정한 경로로 자동 백업 됨
 - 메모리 테이블스페이스 데이터 파일은 시간 단축을 위해 0, 1 중 한 번만 백업

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

데이터베이스 백업

❖ 온라인 테이블스페이스 백업 by DBA

- DB 구동 중 특정 테이블스페이스를 백업
- 구문수행으로 백업 시작과 종료 시점을 통제 후 OS copy 명령어를 사용하여 테이블스페이스 관련 데이터 파일을 수동으로 복사
- 관련 리두로그 파일을 강제로 아카이빙 하기 위해 로그스위치 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;

데이터베이스 백업

❖ 온라인 로그앵커 백업

- DB 구동 중 리두로그앵커 파일을 백업
- 구문수행 시 리두로그앵커 파일 3개 모두 지정한 경로로 자동백업 됨
- 테이블스페이스 변경(ALTER) 시 복구를 대비하여 변경된 테이블스페이스, 디렉터리 테이블스페이스와 함께 백업 함

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

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

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

데이터베이스 백업

❖ 오프라인 데이터베이스 백업

- DB 종료 후 데이터베이스 전체를 백업
- 정상종료확인 후 OS 명령어를 사용하여 데이터 파일, 리두로그 파일, 리두로그앵커 파일을 복사

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

데이터베이스 백업

❖ 오프라인 데이터베이스 백업 예제

➤ 프로퍼티 파일(alibase.properties)의 설정상태

```
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;
```

데이터베이스 백업

❖ 백업 관련 유의사항

- 오프라인 데이터베이스 백업 시 프로퍼티 파일상의 기본경로만 확인해서는 안 된다.
 - 데이터 파일의 위치가 기본 경로가 아닌 경우 누락될 수 있음
 - 데이터 파일 관련 성능뷰를 통하여 누락 여부를 확인하는 과정이 필요
- 온라인 백업과 체크포인트는 동시에 수행될 수 없다.
 - 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.]
- 테이블스페이스에 대한 변경 시는 복구를 대비하여 아래 백업 중 하나를 해야 한다.
 - 해당 테이블스페이스, 디렉터리 테이블스페이스, 리두로그앵커 파일 백업
 - 온라인/오프라인 데이터베이스 백업
- 이중화가 걸려있는 경우 이중화 정보도 같이 백업된다.
 - 백업한 데이터베이스를 다른 시스템에서 복구 시 IP 변경으로 인해 이중화 문제 발생 가능



RECOVERY

데이터베이스 복구

❖ 자동복구 (Restart Recovery)

- 정전, 시스템 crash 같은 비정상종료 후 재구동 시 자동으로 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]
```

데이터베이스 복구

❖ 매체 복구 (Media recovery)

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

매체복구

❖ 완전복구

- 구문을 수행하여 현재 시점까지 데이터베이스를 복구

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

❖ 요구사항

- 데이터베이스 전체 백업 받은 시점 이후의 아카이브로그 파일은 물론 온라인 리두로그 파일까지 모두 손상이 없어야 함
- 현재 시점의 최신 리두로그앵커 파일이 필요

매체복구

❖ 불완전복구

- 구문을 수행하여 특정 시점으로 데이터베이스를 복구
 - 온라인 리두로그 파일 또는 아카이브로그 파일이 손상/유실된 경우

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

- 데이터베이스 전체를 특정시점으로 되돌리고자 하는 경우

```
iSQL(sysdba)> ALTER DATABASE RECOVER DATABASE UNTIL TIME '2010-09-10:17:55:00';
```

- 불완전복구 수행 후에는 온라인 리두로그를 초기화 해야만 함
 - “Meta 단계로 전이”, “다음 재구동”시 자동복구가 수행되지 않도록 위한 필수과정

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

❖ 요구사항

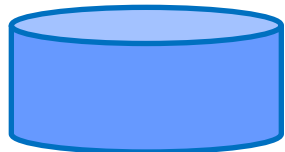
- 복구하려는 시점에 대응하는 리두로그앵커 파일이 필요

매체복구

❖ 데이터 파일 유실 복구 예제(1) - 백업 받은 데이터 파일을 이용한 복구

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

Log Anchor



User1.dbf 정보
파일 생성 LSN(20:012284)
Checkpoint LSN(102:172168)

Online Logs 디렉토리

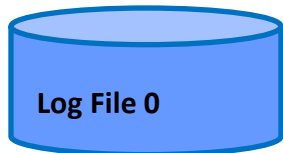


Log File 101



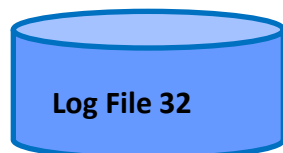
Log File 102

Archive Logs 디렉토리



Log File 0

...



Log File 32

...



Log File 100

백업한 User1.dbf



LSN
32:345698



Recovery 구문 실행



Media Recovery 프로세스

Redolog
재수행

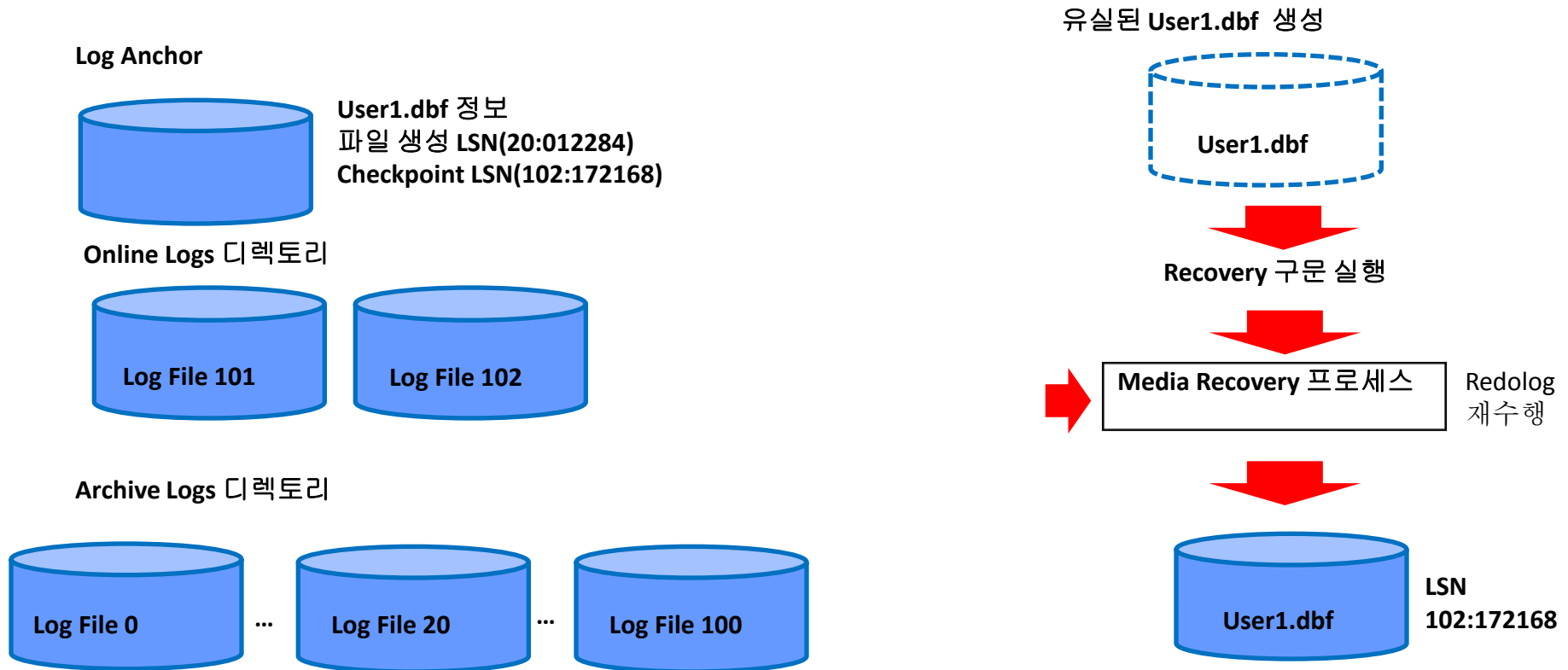


LSN
102:172168

매체복구

❖ 데이터 파일 유실 복구 예제(2) - 빈 데이터 파일 생성을 통한 복구

- 데이터 파일 생성한 후, Archive Log와 Active Log 를 사용하여 Recovery
- 데이터 파일 생성 시점의 LSN(Log Sequence Number)과 현재 시점의 LSN 사이의 Log File 들을 사용함



매체복구

❖ 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 리두로그앵커 파일에 기록된 경로와 일치해야만 함

매체복구

❖ 매체복구 시 유의사항

- 일부 데이터 파일은 백업 정책으로 인하여 사용자가 직접 생성
 - 메모리 테이블스페이스 관련 데이터 파일의 경우 OS 명령어를 사용하여 나머지 한 벌을 생성해야 함

```
shell> cp USER_MEM_DATA-1-0 USER_MEM_DATA-0-0;
```

- 임시 테이블스페이스 관련 데이터 파일의 경우 CONTROL 단계에서 빈 데이터 파일을 생성해야 함

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

❖ 매체복구 후 유의사항

- 불완전 복구 후에는 반드시 데이터베이스 백업
 - 또 다시 복구 수행 시 리두로그 초기화 이후의 데이터베이스 변경사항은 복구가 불가능



ALTIBASE HDB ADMINISTRATION II

UTILITIES

ILOADER

AEXPORT

ALTIPASSWD

ALTIPROFILE

DUMP

ALTIERR

ORANGE for ALTIBASE

MIGRATION CENTER



ILOADER

iLoader

❖ iLoader

- \$ALTIBASE_HOME/bin 에 위치
- DB의 데이터를 CSV 또는, 텍스트 형태의 파일로 다운로드 및 업로드
- 특정 시점의 테이블을 백업하는 용도로도 활용

```
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 6.1.1.0.10

Copyright 2000, ALTIBASE HDB 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 서버로 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 데이터의 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 저장해 놓은 파일

❖ iLoader 실행 시 입력 옵션(접속 및 동작관련)

입력 옵션	설명
-s	접속할 ALTIBASE HDB IP를 지정
-u	ALTIBASE HDB DB 사용자 계정 명을 지정
-p	계정의 패스워드를 지정
-port	ALTIBASE HDB Listen Port번호를 지정
-log	수행결과를 저장할 로그파일명을 지정 (오류도 같이 기록됨)
-bad	수행 중 오류가 발생한 데이터를 별도의 파일로 저장해야 할 경우
-mode	truncate 옵션: 테이블을 먼저 TRUNCATE 시킨 후 입력 (옵션 미사용 시 일반 INSERT와 동일하게 동작)
-replication	true/false : 이중화 환경에서 상대방으로의 반영여부를 지정 (True : iLoader로 입력된 데이터가 이중화로 상대방에 전송됨) (False : iLoader로 입력된 데이터가 이중화로 전송되지 않음)

❖ iLoader 실행 시 입력 옵션(성능관련)

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

❖ iLoader 실행 시 입력 옵션(데이터/파일 관련)

입력 옵션	설명
-t	컬럼 간의 구분자를 지정
-r	레코드간의 구분자를 지정
-f	formout 동작 시 저장 될 form 파일명을 지정
-d	in/out 동작 시 사용 될(저장 될) 데이터 파일명을 지정
-rule	데이터파일을 csv 파일형식으로 생성하고자 할 경우 지정
-error	지정된 숫자만큼 오류가 발생하면 실행을 멈춤 (기본값:50)
-split	데이터 파일마다 저장할 레코드의 개수를 지정 (파일명은 datafile.dat0, datafile.dat1 과 형식으로 생성됨)
-T	작업 대상 테이블 명을 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 경우는 테이블의 형식이 동일하다면 이전에 받은 form파일을 재 사용하여도 된다.

➤ 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 Form파일의 수정 예 (테이블 명을 T1이라고 가정)

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

iLoader

❖ iLoader Form파일의 수정 예 (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 시에 Sequence의 사용

- 데이터를 업로드 할때 순차적인 번호를 부여하고자 할 경우 사용
- 사용할 Sequence 객체는 미리 생성되어 있어야 함
- 데이터파일에는 Sequence에 해당하는 부분이 없어도 업로드가 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 시에 함수의 사용

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

```
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 6.1.1.0.10
```

```
Copyright 2000, ALTIBASE HDB 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.]
```

**Form파일의 날짜형식
과 실제 데이터형
식 불일치**

❖ 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

데이터에 구분자가
포함 되어 경우

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)]

ALTIBASE

AEXPORT

AEXPORT

❖ aexport

- \$ALTIBASE_HOME/bin 에 위치
- DB 간의 자동화된 데이터 마이그레이션(migration)을 지원하기 위한 도구
- DB 에 생성된 객체들의 생성 스크립트를 SQL문 형식으로 파일에 저장

```
Shell::~/home/alti1> aexport
```

```
-----  
Altibase Export Script Utility.
```

```
Release Version 6.1.1.0.10
```

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

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

```
Write Server Name (default:127.0.0.1) :
```

```
Write UserID : sys
```

```
Write Password : manager
```

AEXPORT

❖ aexport

- 수행 전 \$ALTIBASE_HOME/conf/aexport.properties 파일이 존재해야 함
 - 존재하지 않을 경우 아래와 같이 오류가 발생
 - ◆ [ERR-91066 : Property File Loading Error.(conf/aexport.properties)]
 - \$ALTIBASE_HOME/conf/aexport.properties.sample을 복사하여 사용
 - 설정파일 내에서는 "#" 기호가 주석으로 사용됨
 - 사용자가 수정할 옵션은 다음 2가지 항목으로 iLoader 스크립트를 생성할 때 구분자로 사용되는 부분임으로 데이터에 중복되지 않게 복잡한 문자열을 지정 (10자 이내)
 - ◆ ILOADER_FIELD_TERM = ^
 - ◆ ILOADER_ROW_TERM = %n
 - ◆ 이 옵션을 주석으로 처리할 경우 aexport에 생성된 iLoader 쉘 스크립트의 데이터파일 저장형식은 CSV를 사용함

AEXPORT

❖ aexport 실행 시 입력 옵션

입력 옵션	설명
-s	접속할 ALTIBASE HDB 서버의 IP
-u	ALTIBASE HDB DB 사용자 명 (sys 이외의 경우 해당 사용자 정보만 추출)
-p	사용자의 패스워드
-port	ALTIBASE HDB Listen Port 번호
-object	추출할 객체를 소유자와 함께 명시
-tserver	추출할 데이터를 업로드할 대상 서버를 지정
-tport	접속할 대상 서버의 포트 번호
-nls_use	ALTIBASE HDB 문자셋 정보 (주석 등에 한글을 사용한 경우를 대비해 올바른 문자셋 입력 필요)

※ -tserver, -tport 옵션 사용시 생성된 스크립트의 서버 IP 정보와 Port 번호 정보는 로컬 서버정보가 아닌 -tserver, -tport 에 명시된 타겟 서버의 정보로 대체

AEXPORT

❖ aexport에 의해 생성되는 파일 : SQL파일

➤ 전체 DB 모드: SYS 사용자만 실행 가능

파일명	설명
SYS_CRT_DIR.sql	모든 디렉토리를 생성하는 SQL문을 저장
SYS_CRT_USER.sql	모든 DB 사용자를 생성하는 SQL문을 저장
SYS_CRT_REP.sql	이중화 객체 생성 정보를 저장
SYS_CRT_SYNONYM.sql	모든 시노님(synonym) 객체 생성 SQL문을 저장
ALL_CRT_VIEW_PROC.sql	모든 뷰와 프로시저 생성 SQL문을 저장
ALL_CRT_TBS.sql	모든 테이블스페이스 생성 SQL문을 저장
ALL_CRT_TBL.sql	모든 사용자 테이블 생성 SQL문을 저장
ALL_CRT_INDEX.sql	모든 사용자 인덱스 생성 SQL문을 저장
ALL_CRT_FK.sql	모든 사용자의 외래키 생성 SQL문을 저장
ALL_CRT_TRIG.sql	모든 사용자의 트리거 생성 SQL문을 저장
ALL_CRT_SEQ.sql	모든 사용자의 시퀀스 생성 SQL문을 저장
ALL_CRT_LINK.sql	모든 사용자의 데이터베이스 LINK 생성 SQL문을 저장

AEXPORT

❖ aexport에 의해 생성되는 파일 : SQL파일

- 사용자 모드: SYS 사용자와 지정 사용자가 실행 가능

파일명	설명
사용자명_CRT_TBS.sql	지정한 사용자의 모든 테이블 생성 SQL문을 생성
사용자명_CRT_INDEX.sql	지정한 사용자의 모든 인덱스 생성 SQL문을 생성
사용자명_CRT_FK.sql	지정한 사용자의 모든 외래키 생성 SQL문을 생성
사용자명_CRT_TRIG.sql	지정한 사용자의 모든 트리거 생성 SQL문을 생성
사용자명_CRT_SEQ.sql	지정한 사용자의 모든 시퀀스 생성 SQL문을 생성
사용자명_CRT_LINK.sql	지정한 사용자의 모든 데이터베이스 LINK 생성 SQL문을 생성

- 객체 모드 : SYS 사용자와 지정 사용자가 실행 가능

파일명	설명
사용자명_객체명_CRT.sql	지정한 사용자의 객체 생성 SQL문을 저장

AEXPORT

❖ aexport에 의해 생성되는 파일 : 쉘 실행파일

파일명	설명
run_is.sh	테이블스페이스, DB사용자, 테이블, 프로시저, VIEW, 사용자 권한 등을 순차적으로 생성하는 스크립트
run_is_cons.sh	제약사항(constraints)을 순차적으로 생성하는 스크립트 (TWO_PHASE_SCRIPT 프로퍼티 ON 생성시 생성되며 인덱스, 외래키, 트리거, 이중화 객체 생성 스크립트를 포함)
run_is_fk.sh	참조키를 순차적으로 생성하는 스크립트 (TWO_PHASE_SCRIPT 프로퍼티 ON 설정시 생성되지 않음)
run_is_index.sh	인덱스를 순차적으로 생성하는 스크립트 (TWO_PHASE_SCRIPT 프로퍼티 ON 설정시 생성되지 않음)
run_is_repl.sh	이중화 객체를 생성하는 스크립트 (TWO_PHASE_SCRIPT 프로퍼티 ON 설정시 생성되지 않음)
run_il_in.sh	데이터를 순차적으로 업로드 하는 스크립트
run_il_out.sh	데이터를 순차적으로 다운로드 하는 스크립트

- ※ -tserver, -tport 옵션 사용시 생성된 스크립트중 run_il_out.sh (데이터다운로드 스크립트)의 경우 입력옵션의 서버 IP정보를 로컬서버(데이터를 추출할 서버) 정보로 수정이 필요

AEXPORT

❖ aexport를 이용한 전체 DB 마이그레이션 절차

1. 소스 장비에서 aexport를 실행하여 객체 정보를 스크립트 파일로 백업
 2. 소스 장비에서 aexport를 통해 만들어진 run_il_out.sh 를 실행하여 데이터를 백업
 3. 소스 장비에서 (1, 2번) 과정에서 생성된 파일을 모두 대상 장비로 이관
 4. 대상 장비의 DB 재생성 및 구동
 5. 대상 장비에서 run_is.sh 을 수행 (테이블스페이스, DB사용자, 테이블 등을 생성)
 6. 대상 장비에서 run_il_in.sh 을 수행 (데이터를 업로드)
 7. 대상 장비에서 run_is_index.sh 을 수행 (인덱스 정보를 생성)
 8. 대상 장비에서 run_is_fk.sh 을 수행 (참조키를 생성)
- ※ 이미 대상 장비에 동일한 DB객체가 존재할 경우에는 5번 과정을 생략하고 대상 장비의 모든 테이블을 TRUNCATE 한 후 데이터를 업로드 해야 함
- ※ TWO_PHASE_SCRIPT 프로퍼티가 ON일 경우는 6번, 8번 과정 대신 run_is_con.sh 스크립트를 실행

ALTIBASE

ALTIPASSWD

ALTIPASSWD

❖ altiPasswd

- \$ALTIBASE_HOME/bin 에 위치
- SYS계정의 패스워드를 "ALTER USER" 구문으로 변경한 경우 반드시 수행해야 함
 - 변경하지 않은 경우 ALTIBASE HDB 구동/종료 등 모든 DBA권한 작업에서 패스워드 오류 발생
 - 수행방법의 예

아래와 같이 iSQL상에서 "SYS" 계정의 패스워드를 변경한 후
iSQL> **ALTER USER sys IDENTIFIED BY ppp1234;**

아래와 같이 사용자 프롬프트 상에서 altipasswd를 이용하여 동일하게 변경

Shell:~/home/alti1> **altipasswd**

Previous Password : 기존 패스워드 입력

New Password : 새로운 변경 패스워드 입력

Retype New Password : 패스워드 확인

ALTIBASE

ALTIPROFILE

ALTIPROFILE

❖ altiProfile

- \$ALTIBASE_HOME/bin 에 위치
- 프로파일링을 수행한 이후 생성된 결과물을 분석할 수 있게 변환하는 유틸리티
- 프로파일링으로 저장되는 정보
 - 질의를 수행한 클라이언트 정보
 - 질의 처리의 통계정보 (처리시간, 실행계획, 수행누적횟수, 바인딩정보)

```
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

- 질의처리 통계를 파일에 저장하도록 프로파일링을 개시 (iSQL을 이용)

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

- \$ALTIBASE_HOME/trc/에 프로파일링 결과가 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 변동됨)
```

- altiProfile을 수행 시 주의사항

- 프로파일링 정보를 파일로 계속 저장하기 때문에 디스크 Full이 발생하지 않도록 주의
 - ◆ 중지 시에는 "ALTER SYSTEM SET QUERY_PROF_FLAG = 0;" 를 iSQL에서 반드시 실행
- 실시간으로 파일에 분석정보를 기록하기 때문에 성능저하가 발생할 수 있음

ALTIPROFILE

❖ altiProfile 실행 및 분석결과

- 생성된 alti-xxxx-y.prof 파일을 입력 파일명으로 지정
- 프로파일링 결과물은 사용자가 해석할 수 없으므로 변환이 필요

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

- 생성된 분석파일 결과의 예

```
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]
```

실행 시각

세션정보

질의문

ALTIPROFILE

❖ altiProfile 실행 (계속)

Elapsed Time		수행시각
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		실행결과
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	
Memory Full Scan Count =	1
Memory Index Scan Count =	1
Disk Full Scan Count =	0
Disk Index Scan Count =	0

인덱스 비용

Disk Access Info	
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

버퍼캐쉬/디스크 접근비용

ALTIPROFILE

❖ altiProfile 실행 (계속)

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

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

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

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

```
-----
```

```
]
```



실행계획

ALTIBASE

DUMP

DUMPLA

❖ dump 유틸리티

구분	설명
dumpla	로그앵커파일을 분석 가능하도록 화면에 출력 복구나 DB의 일부 정보를 임의로 확인하기 위해 사용
dumplf	리두로그 파일에 기록된 트랜잭션 로그를 분석하는 유틸리티
dumpddf	디스크 데이터파일에 대한 복구 시작점을 확인 디스크 데이터파일이 갖고 있는 특정 페이지의 무결성을 임의로 체크해야 할 경우 사용

DUMPLA

❖ dumpla

- \$ALTIBASE_HOME/bin 에 위치
- 로그앵커파일의 정보를 해석해주는 유틸리티

```
Shell::~/home/alti1/altibase_home/logs> dumpla loganchor0
```

```
-----  
Altibase Client Dump LogAnchor utility.
```

```
Release Version 6.1.1.0.10
```

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

```
All Rights Reserved.
```

```
-----  
<< DUMP OF LOGANCHOR FILE - loganchor0 >>
```

DUMPLA

❖ dumpla 결과의 이해 (1)

- 로그앵커파일의 기본 속성 정보
- 각 정보들을 로그앵커에 기록할 때 해당 정보가 저장되는 구조체의 길이를 의미

```
-----  
[LOGANCHOR ATTRIBUTE SIZE  
Loganchor Static Area      [ 1280 ]  
Tablespace Attribute       [ 600 ]  
Checkpoint Path Attribute  [ 520 ]  
Checkpoint Image Attribute [ 404 ]  
Disk Datafile Attribute    [ 584 ]
```


DUMPLA

❖ dumpla 결과의 이해 (2)

➤ 로그앵커 헤더정보

[LOGANCHOR HEADER]	
Binary DB Version	[5.4.1]
Archivelog Mode	[Archivelog]
Transaction Segment Entry Count	[512]
Begin Checkpoint LSN	[0, 3, 4796393]
End Checkpoint LSN	[0, 3, 4797952]
Disk Redo LSN	[0, 3, 4796393]
Global SN	[29473]
SN for Recovery from Replication	[NULL]
Server Status	[SERVER STARTED]
Log File Group Count	[1]
Log File Group	[0]

archive모드

복구 시작점


ALTIBASE HDB 구동정보

DUMPLA

❖ dumpla 결과의 이해 (3)

➤ 로그앵커 헤더정보 (계속)

```
-----  
[LOGANCHOR HEADER]  
  
.....  
.....  
End LSN                [ 0, 3,4796393 ]  
ResetLog LSN           [ 4294967295, 4294967295, 4294967295 ]  
Last Created Logfile Num [ 6 ]  
Delete Logfile(s) Range [ 2 ~ 1 ]  
Update And Flush Count [ 399 ]  
New Tablespace ID      [ 6 ]
```



체크포인트 시
삭제된
로그파일 범위

DUMPLA

❖ dumpla 결과의 이해 (4)

➤ 로그엔커에 저장된 테이블스페이스 정보

```
[ TABLESPACE ATTRIBUTE ]
Tablespace ID          [ 1 ]
Tablespace Name        [ SYS_TBS_MEM_DATA ]
New Database File ID   [ 0 ]
Extent Management      [ FREE EXTENT BITMAP TABLESPACE ]
Tablespace Status      [ ONLINE ]
TableSpace Type        [ 1 ]
Checkpoint Path Count  [ 0 ]
Autoextend Mode        [ Autoextend ]
Shared Memory Key      [ 0 ]
Stable Checkpoint Image Num. [ 1 ]
Init Size              [ 8 MBytes ( 257 Pages ) ]
Next Size              [ 4 MBytes ( 128 Pages ) ]
Maximum Size           [ 134217727 MBytes ( 4294967295 Pages ) ]
Split File Size        [ 1024 MBytes ( 32768 Pages ) ]
```

DUMPLA

❖ dumpla 결과의 이해 (5)

➤ 로그엔커에 저장된 디스크 데이터파일의 정보

[DISK DATABASE FILE ATTRIBUTE]

Tablespace ID	[5]
Database File ID	[0]
Database File Path	[/home/alti1/dbs/user_data001.dbf]
Create LSN	[0, 0, 509526]
Database File Status	[ONLINE]
Autoextend Mode	[Autoextend]
Create Mode	[1]
Initialize Size	[100 MBytes (12800 Pages)]
Current Size	[100 MBytes (12800 Pages)]
Next Size	[1 MBytes (128 Pages)]
Maximum Size	[2048 MBytes (262144 Pages)]

생성 시점의
리두로그파일
정보

DUMPLA

❖ dumpla 결과의 이해 (6)

- 로그엔커에 저장된 메모리 체크포인트 패스 정보

[MEMORY CHECKPOINT PATH ATTRIBUTE]

Tablespace ID [1]

Checkpoint Path [/home/alti1/altibase_home/dbs]

[MEMORY CHECKPOINT IMAGE ATTRIBUTE]

Tablespace ID [1]

File Number [0]

Create LSN [0, 0, 1380]

Create On Disk (PingPong 0) [Created]

Create On Disk (PingPong 1) [Created]

DUMPLF

❖ dumplf

- \$ALTIBASE_HOME/bin 에 위치
- 리두로그파일을 분석하기 위한 유틸리티 (redirection 방법으로 결과를 저장)

```
Shell:~/home/alti1/altibase_home/logs> dumplf logfile1234 > 1234.txt
```

```
Shell:~/home/alti1/altibase_home/logs> cat 1234.txt
```

```
-----  
Altibase Client Dump LogAnchor utility.  
Release Version 6.1.1.0.10  
Copyright 2000, ALTIBASE HDB Corporation or its subsidiaries.  
All Rights Reserved.  
-----
```

```
.....
```

```
.....
```

DUMPLF

❖ dumpLf

➤ 로그파일의 시작로그

```
SN=<251745>,LSN=< 0,0>, COMP:N, MAGIC:12288, TID: 4294967295,BE: N, REP: N,  
ISVP: N PLSN=< 4294967295, 4294967295, 4294967295 >, LT: SMR_LT_FILE_BEGIN,  
SZ: 49 LFGID: 0, FileNo: 3
```

➤ 로그파일의 종료로그

```
SN=<251744>,LSN=< 0,10485648>, COMP:N, MAGIC:12176, TID: 4294967295,  
BE: N, REP: N, ISVP: N PLSN=< 4294967295, 4294967295, 4294967295 >,  
LT: SMR_LT_FILE_END, SZ: 41
```

- 사용자가 임의로 확인할 경우는 일반적으로 없지만 특정 문제가 발생한 시점의 트랜잭션의 종류나 대상 테이블을 확인하고자 할 경우 유용
 - 각 명칭에 대해서는 슬라이드 노트를 참조

DUMPLF

❖ dumplf

➤ 주요 로그타입의 분류

LT	설명
SMR_LT_DUMMY	트랜잭션과 무관한 로깅 발생 시 (내부 리소스 로깅 등)
SMR_LT_CHKPT_BEGIN	체크포인트 개시
SMR_CHKPT_END	체크포인트 종료
SMR_LT_TRANS_COMMIT	트랜잭션 COMMIT 로그
SMR_LT_TRANS_ABORT	트랜잭션 ROLLBACK 로그
SMR_LT_UPDATE	트랜잭션 로그임을 표기

DUMPLF

❖ dumplf

➤ 트랜잭션 로그의 예 (DELETE문 로그)

```
SN=<570331>,LSN=< 0,5584455>, COMP:N, MAGIC:22087, TID: 22336, BE: N, REP: N, ISVP: N PLSN=< 0, 5, 5584372 >, LT: SMR_LT_UPDATE, SZ: 83  
UTYPE: SMR_SMC_PERS_DELETE_VERSION_ROW,  
UPOS( SPACEID: 1, PID: 328, OFFSET: 15304 => OID: 10763208 )  
TTbOID: 2916376
```

대상
테이블의
고유번호

DELETE
수행

트랜잭션
고유번호

DUMPDDF

❖ dumpddf

- \$ALTIBASE_HOME/bin 에 위치
- 데이터파일의 헤더 정보 및 페이지의 무결성을 임의로 체크해보고자 할 경우 사용

```
Shell::~/home/alti1> dumpddf -m -f $ALTIBASE_HOME/dbs/system001.dbf
[BEGIN DATABASE FILE HEADER]

Binary DB Version      [ 5.4.1 ]      # DB 버전
Redo LSN                [ 0, 7, 2701724 ] # Recovery시에 필요한 로그파일 정보
Create LSN              [ 0, 0, 2036 ] # 데이터파일 생성시점의 로그파일 정보

[END DATABASE FILE HEADER]
```

DUMPDDF

❖ dumpddf 입력옵션

옵션	설명
-p	페이지의 고유번호를 명시할 경우
-f	대상 데이터파일명을 지정
-m	대상 데이터파일의 헤더를 출력

DUMPDDF

❖ dumpddf 사용 결과 예

```
Shell::~/home/alti1/altibase_home/dbs> dumpddf -m -f system001.dbf -p 27
```

```
[END DATABASE FILE HEADER]
```

```
PID(27), Start Offset(229376)
```

```
----- Physical Page Begin -----
```

```
Checksum           : 903803392
```

```
Page ID            : 0
```

```
.....
```

```
Page Start Pointer : 7fbfffc000
```

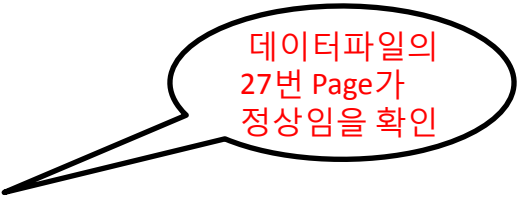
```
Page Type          : 0
```

```
Page State         : 0
```

```
Index SMO Number   : 0
```

```
----- Physical Page End -----
```

```
This page(27) is not corrupted.
```



데이터파일의
27번 Page가
정상임을 확인

ALTIBASE

ALTIERR

ALTIERR

❖ Altterr

- ATLIBASE HDB 에러 코드에 대한 상세한 설명을 검색하여 출력
- 에러 메시지 패턴을 이용하여 검색이 가능

➤ 구문

```
altterr { -w keyword pattern | [-n] error number }
```

➤ 파라미터

파라미터	설명
-w	지정한 검색 패턴을 포함하는 에러 메시지를 찾는다. 검색 패턴을 포함하는 모든 에러 메시지를 출력한다.
-n	에러 번호를 사용하여 검색한다. 에러 번호는 16진수/양의 정수/음의 정수 형태로 입력 가능하다. 에러 코드와 정확히 일치하는 레코드만 검색된다. 에러 번호를 사용하여 검색할 경우 파라미터("-n")를 생략할 수 있다.



ORANGE for ALTIBASE

ORANGE FOR ALTIBASE

❖ ORANGE FOR ALTIBASE

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

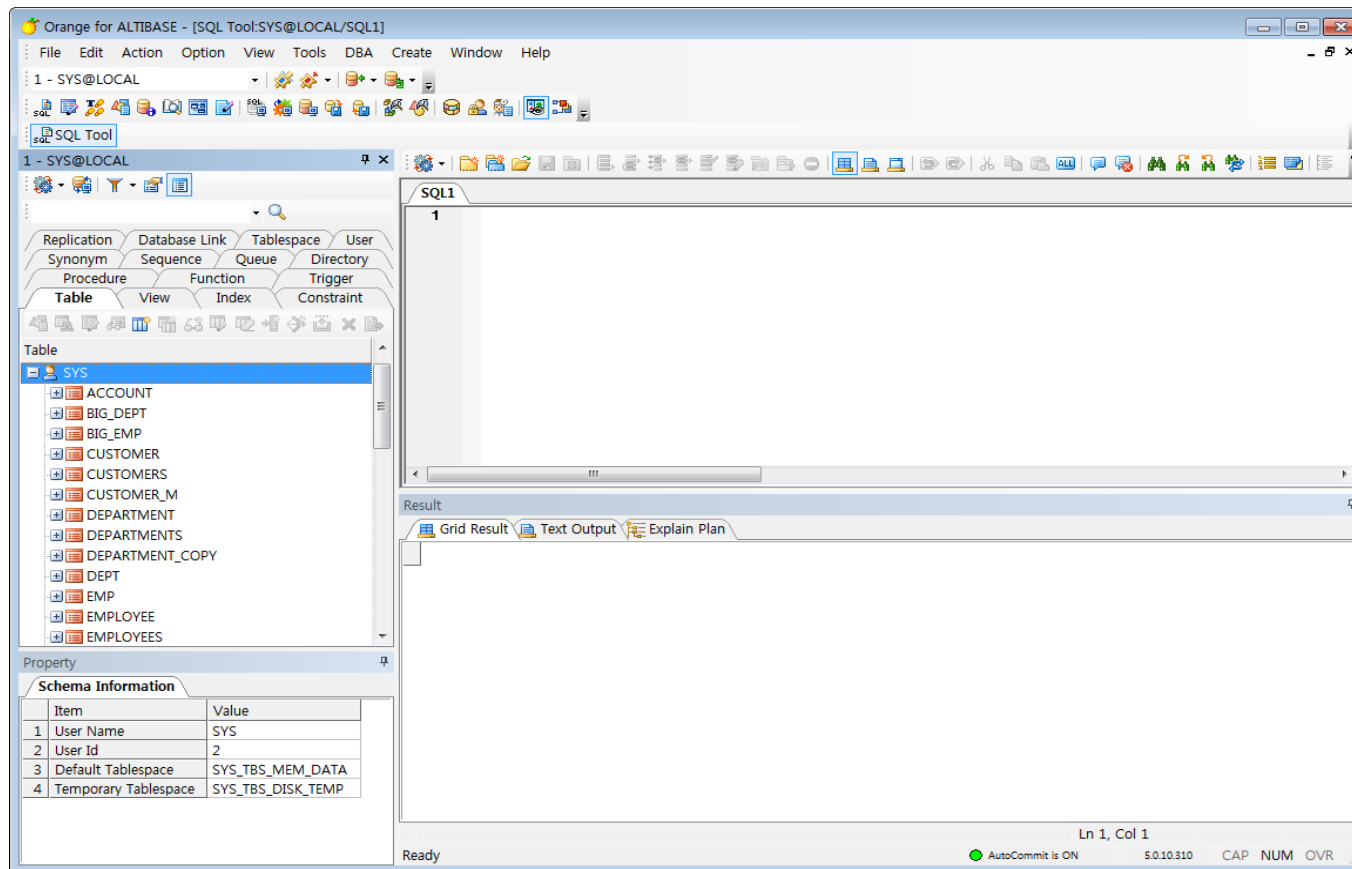
❖ 제품 구성

- 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] 를 실행

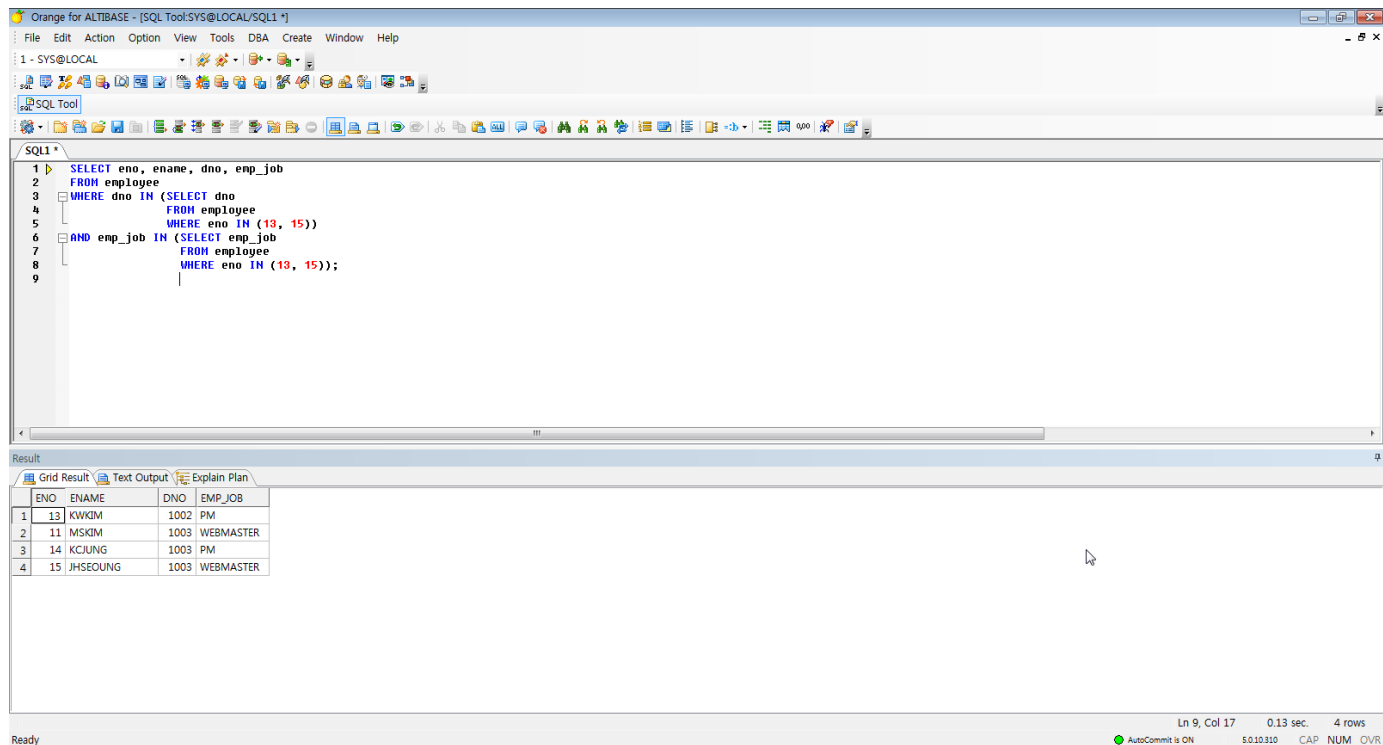


ORANGE FOR ALTIBASE

❖ ORANGE 상세 기능

➤ SQL Tool

- 다수의 SQL을 일괄 또는 순차 실행하거나 특정 SQL만을 선택적으로 실행 할 수 있고 실행 결과를 직접 수정하거나 파일로 저장할 수 있음



ORANGE FOR ALTIBASE

❖ ORANGE 상세 기능

➤ Table Editor

- 데이터의 DML 작업을 GUI환경에서 편리하게 할 수 있으며, 조건에 의한 일부 데이터의 조회, 편집, 저장 기능을 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 'AutoCommit is ON', version '5.0.10.310', and column counts for 'CAP', 'NUM', and 'OVR'.

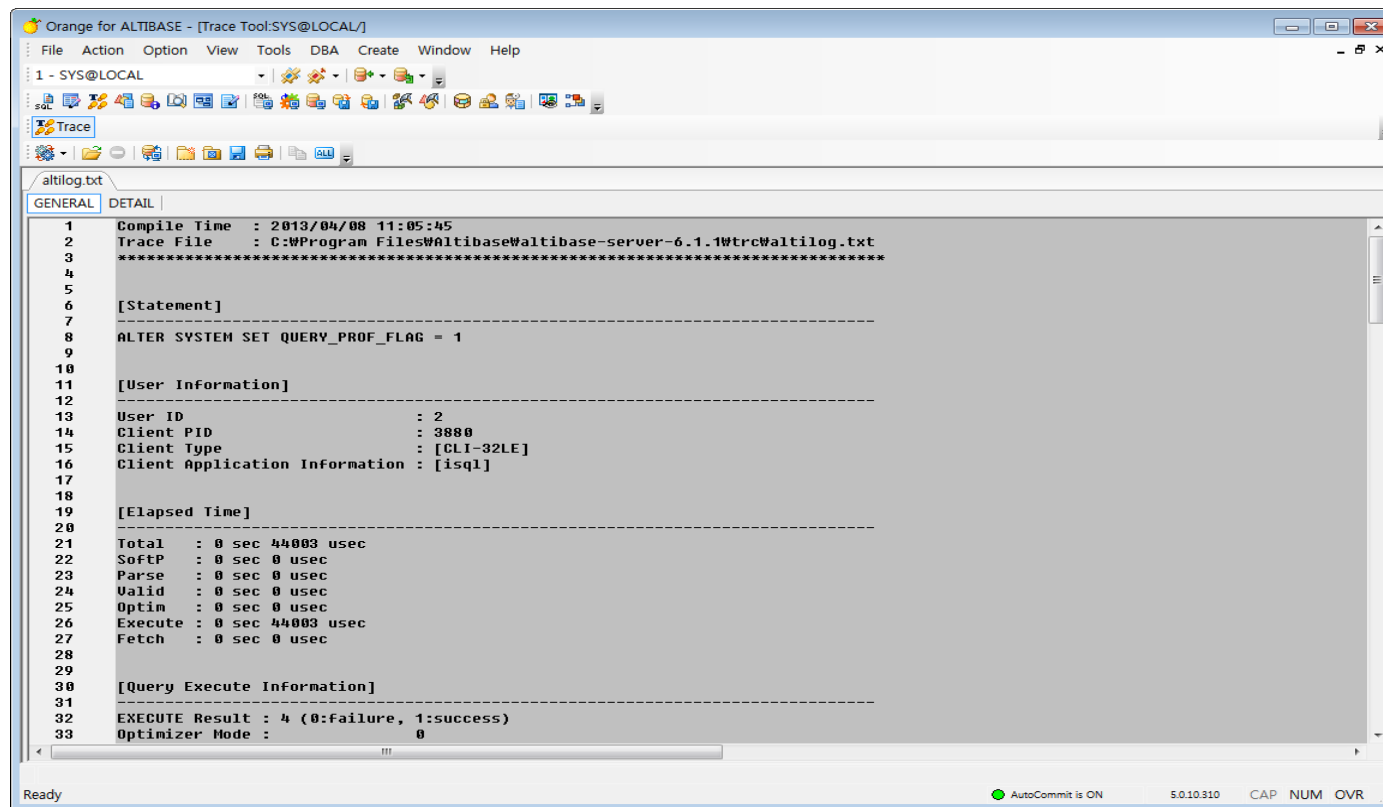
	ENO	E_LASTNAME	E_FIRSTNAME	EMP_JOB	EMP_TEL	DNO	SALARY	SEX	BIRTH	JOIN_DATE	STATUS
1	1	Moon	Chan-seung	CEO	01195662365	3002		M			R
2	2	Davenport	Susan	designer	0113654540		1500	F	721219	2009/11/18 00:00:00	H
3	3	Kobain	Ken	engineer	0162581369	1001	2000	M	650226	2010/01/11 00:00:00	H
4	4	Foster	Aaron	PL	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 정보와 실행 계획 정보를 실시간으로 제공



The screenshot displays the 'Orange for ALTIBASE - [Trace Tool:SYS@LOCAL]' application window. The main area shows the contents of a trace file named 'altilog.txt'. The trace data is as follows:

```
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 : [CLI-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
```

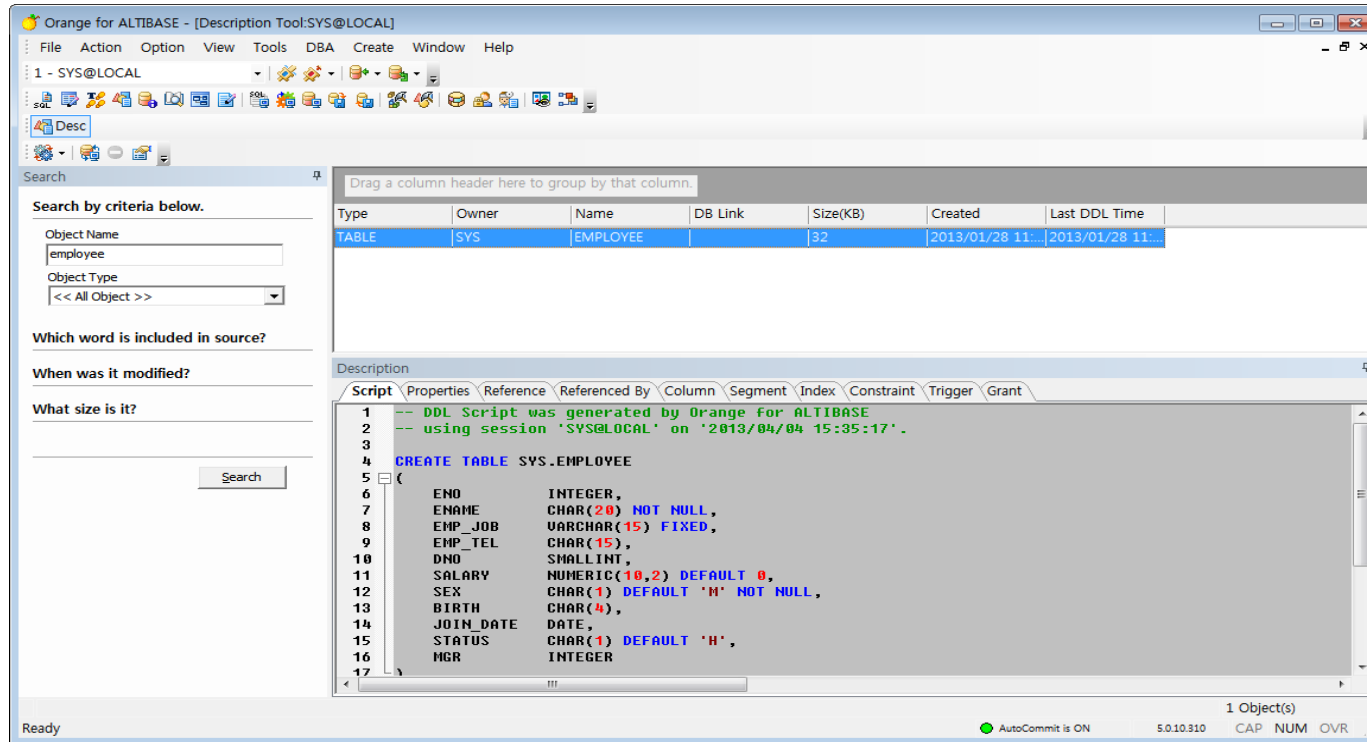
The status bar at the bottom indicates 'Ready', 'AutoCommit is ON', and version '5.0.10.310'.

ORANGE FOR ALTIBASE

❖ ORANGE 상세 기능

➤ Description Tool

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

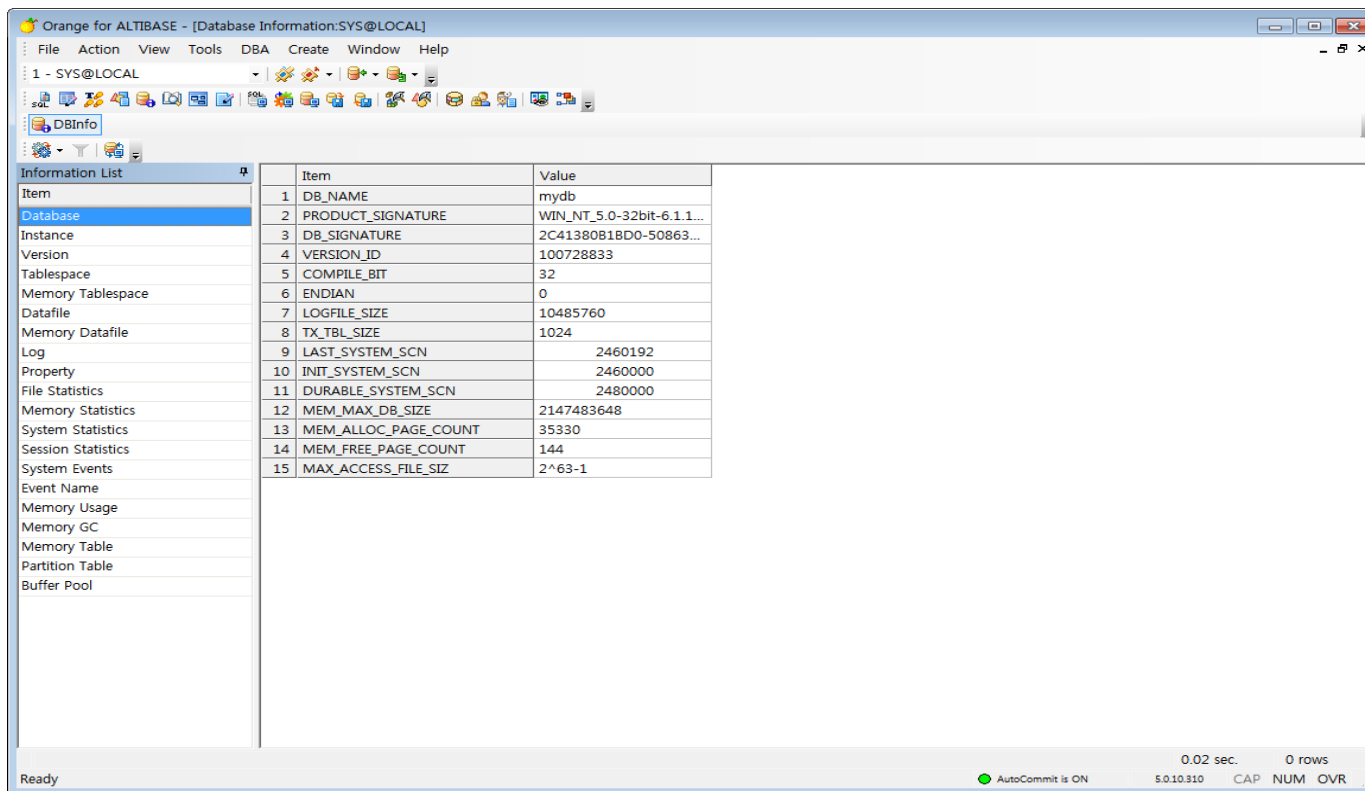


ORANGE FOR ALTIBASE

❖ ORANGE 상세 기능

➤ Database Information

- 데이터베이스의 전반적인 구성 정보를 제공



Orange for ALTIBASE - [Database Information:SYS@LOCAL]

File Action View Tools DBA Create Window Help

1 - SYS@LOCAL

DBInfo

Item	Value
1 DB_NAME	mydb
2 PRODUCT_SIGNATURE	WIN_NT_5.0-32bit-6.1.1...
3 DB_SIGNATURE	2C41380B1BD0-50863...
4 VERSION_ID	100728833
5 COMPILE_BIT	32
6 ENDIAN	0
7 LOGFILE_SIZE	10485760
8 TX_TBL_SIZE	1024
9 LAST_SYSTEM_SCN	2460192
10 INIT_SYSTEM_SCN	2460000
11 DURABLE_SYSTEM_SCN	2480000
12 MEM_MAX_DB_SIZE	2147483648
13 MEM_ALLOC_PAGE_COUNT	35330
14 MEM_FREE_PAGE_COUNT	144
15 MAX_ACCESS_FILE_SIZ	2^63-1

Information List

Item

Database

Instance

Version

Tablespace

Memory Tablespace

Datafile

Memory Datafile

Log

Property

File Statistics

Memory Statistics

System Statistics

Session Statistics

System Events

Event Name

Memory Usage

Memory GC

Memory Table

Partition Table

Buffer Pool

Ready

AutoCommit is ON

0.02 sec. 0 rows

5.0.10.310 CAP NUM OVR

ORANGE FOR ALTIBASE

❖ ORANGE 상세 기능

➤ Data Dictionary Tool

- 자료사전에 존재하는 다양한 뷰를 성격 및 역할에 따라 카테고리화 하고 관련설명을 제공

Orange for ALTIBASE - [Data Dictionary Tool:SYS@LOCAL]

File Action View Tools DBA Create Window Help

1 - SYS@LOCAL

Dict

Dictionary List

Performance / Dictionary / Search

Search

#	Column Name	Data Type	Nullable	Comments
1	NAME	DOUBLE		
2	MIN	DOUBLE		
3	ATTR	CHAR(0)		
4	MAX	DOUBLE		
5	VALUE1	DOUBLE		
6	VALUE2	DOUBLE		
7	VALUE3	DOUBLE		
8	VALUE4	DOUBLE		
9	VALUE5	DOUBLE		
10	VALUE6	DOUBLE		
11	VALUE7	DOUBLE		
12	VALUE8	DOUBLE		

Result

Where

	NAME	STOREDCOUNT	ATTR	MIN	MAX	VALUE1	VALUE2	VALUE3	VALUE4	VALUE5	VALUE6	VALUE7	VALUE8
1	PORT_NO	1	385875970	1024	65535	20300							
2	IPC_PORT_NO	1	385875970	1024	65535	20350							
3	IPC_CHANNEL_COUNT	1	385875970	0	65535	0							
4	NET_CONN_IP_STACK	1	385875970	0	2	0							
5	NLS_COMP	1	385875970	0	1	0							
6	NLS_NCHAR_CONV_EXCP	1	385875968	0	1	0							
7	NLS_NCHAR_LITERAL_REPLACE	1	385875970	0	1	0							

Ready

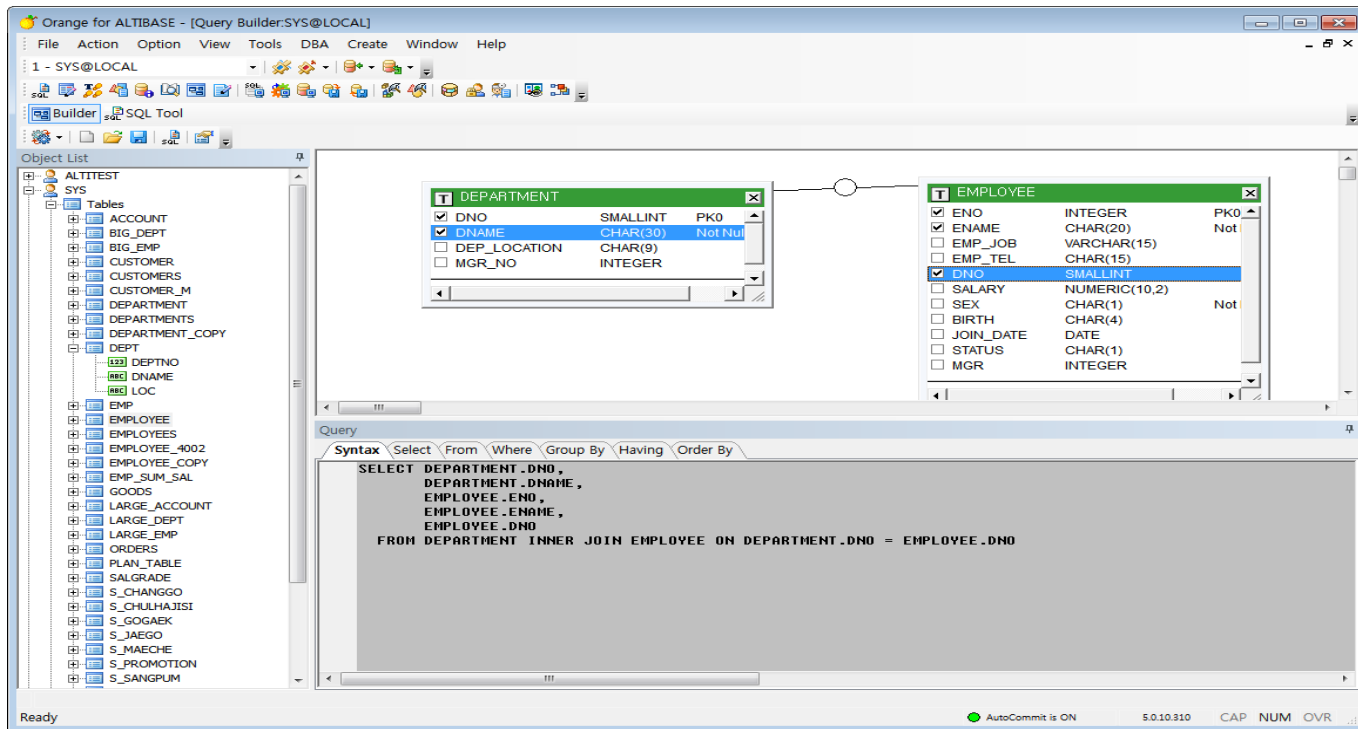
AutoCommit is ON 0.10 sec. 100+ rows 5.0.10.310 CAP NUM OVR

ORANGE FOR ALTIBASE

❖ ORANGE 상세 기능

➤ Query Builder

- GUI 환경에서 논리적인 데이터베이스 객체 모델을 생성 할수 있도록 개발된 사용자 중심의 SQL 쿼리 생성기

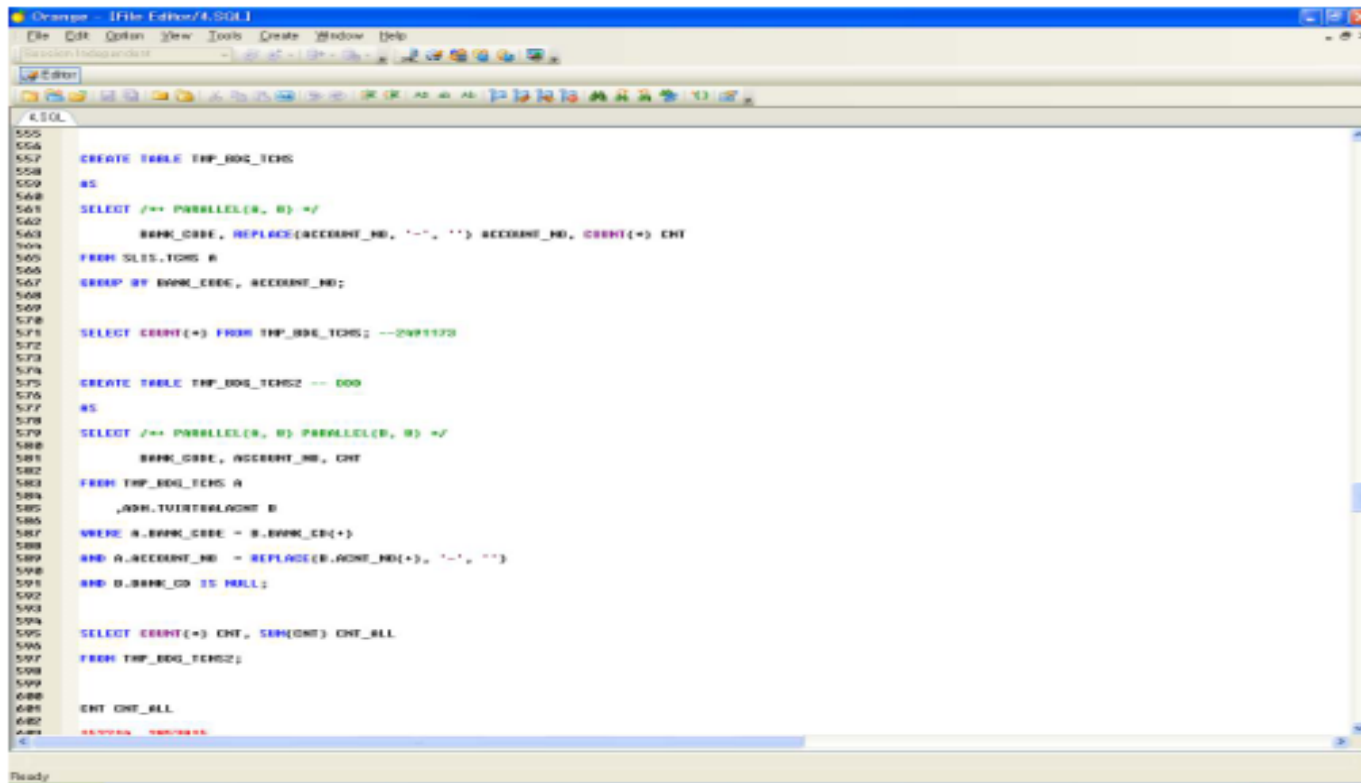


ORANGE FOR ALTIBASE

❖ ORANGE 상세 기능

➤ File Editor

- 텍스트 데이터 파일을 편집하는 등의 작업시 전용으로 쓸수 있는 에디터



The screenshot shows the Orange File Editor window with the following SQL code:

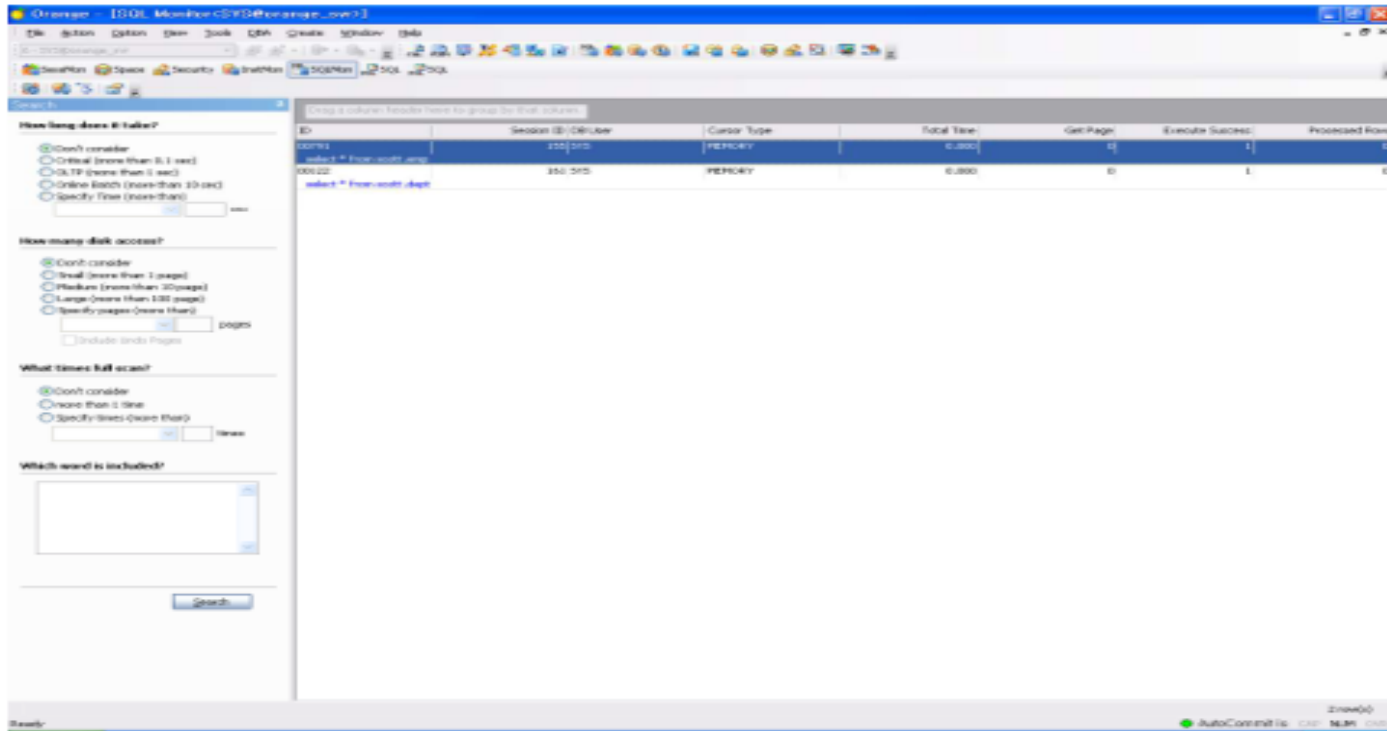
```
555
556
557 CREATE TABLE TMP_BDG_TCHS
558
559
560
561 SELECT /*+ PARALLEL(A, B) */
562
563     BANK_CODE, REPLACE(ACCOUNT_NO, '-', '') ACCOUNT_NO, COUNT(*) CNT
564
565 FROM SLS.TCHS A
566
567 GROUP BY BANK_CODE, ACCOUNT_NO;
568
569
570
571 SELECT COUNT(*) FROM TMP_BDG_TCHS; --298173
572
573
574
575 CREATE TABLE TMP_BDG_TCHS2 -- 000
576
577
578
579 SELECT /*+ PARALLEL(A, B) PARALLEL(C, D) */
580
581     BANK_CODE, ACCOUNT_NO, CNT
582
583 FROM TMP_BDG_TCHS A
584
585     ,ADM.TURTALOGS B
586
587 WHERE A.BANK_CODE = B.BANK_CD(+)
588
589 AND A.ACCOUNT_NO = REPLACE(B.AGNT_NO(+), '-', '')
590
591 AND B.BANK_CD IS NULL;
592
593
594
595 SELECT COUNT(*) CNT, SUM(CNT) CNT_ALL
596
597 FROM TMP_BDG_TCHS2;
598
599
600
601 CNT CNT_ALL
602
603
```

ORANGE FOR ALTIBASE

❖ ORANGE 상세 기능

➤ SQL Monitor

- 실행된 Dynamic SQL중에서 성능저하를 유발시키는 Top N SQL을 쉽고 빠르게 찾아주는 기능을 제공



ORANGE FOR ALTIBASE

❖ ORANGE 상세 기능

➤ Session Monitor

- DB에 현재 접속되어 있는 모든 Agent 리스트와 각 Agent 의 여러가지 상세정보를 제공

The screenshot displays the ORANGE Session Monitor interface. The top section shows a table of active sessions with columns for SID, Username, Connection Info, Wait Event, Active Flag, Open/Wait Count, Client Type, Client App Info, Status, Commit Flag, Data page read, memory table c..., disk table curs..., and write endlog... The bottom section shows a detailed view of a session with tabs for Current Statement, Blocked Locks, Blocking Locks, Transaction, Statistics, Session, Current Wait, and Wait Event. The Current Statement tab is selected, showing a SQL query: SELECT * FROM Gantt_data.

SID	Username	Connection Info	Wait Event	Active Flag	Open/Wait Count	Client Type	Client App Info	Status	Commit Flag	Data page read	memory table c...	disk table curs...	write endlog...
100 (SYS)	TOP	PRC-300.0...	no wait event	ACTIVE	0 (0/0-000)	Orange for ALT...	ORANGE	AutoCommit	0	0	0	0	0
104 (SYS)	TOP	PRC-300.0...	no wait event	ACTIVE	1 (0/1-000)	Orange for ALT...	ORANGE	AutoCommit	0	17	0	0	0
107 (SYS)	TOP	PRC-300.0...	no wait event	INACTIVE	0 (0/0-000)	Orange for ALT...	ORANGE	AutoCommit	0	0	0	0	0
108 (SYS)	TOP	PRC-300.0...	no wait event	INACTIVE	0 (0/0-000)	Orange for ALT...	ORANGE	AutoCommit	0	0	0	0	0
109 (SYS)	TOP	PRC-300.0...	no wait event	INACTIVE	0 (0/0-000)	Orange for ALT...	ORANGE	AutoCommit	0	15	0	0	0
102 (SYS)	TOP	PRC-300.0...	no wait event	INACTIVE	0 (0/0-000)	Orange for ALT...	ORANGE	AutoCommit	0	0	0	0	0
103 (SYS)	TOP	PRC-300.0...	no wait event	ACTIVE	0 (0/0-000)	Orange for ALT...	ORANGE	AutoCommit	0	0	0	0	0

Session Information

Current Statement

```
SELECT * FROM Gantt_data
```

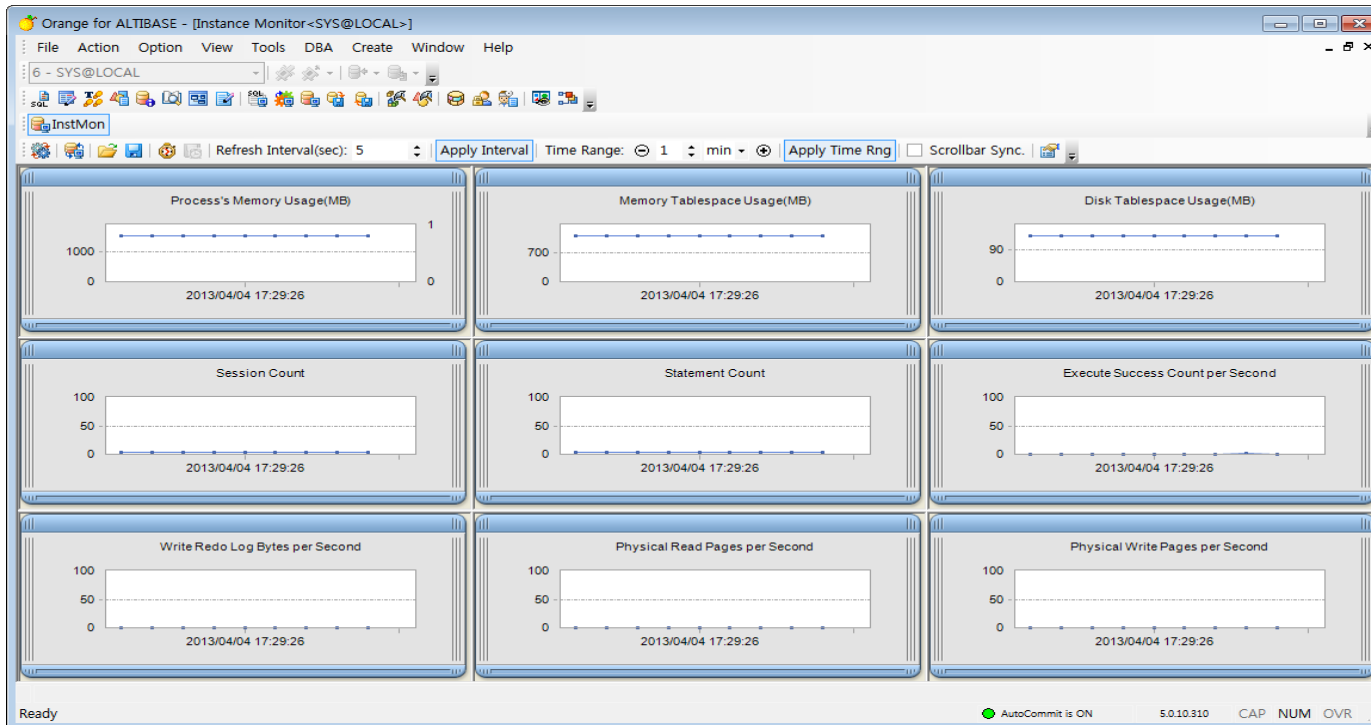
Total: 7 Active: 3
AutoCommit is: 0/0 0/0

ORANGE FOR ALTIBASE

❖ ORANGE 상세 기능

➤ Instance Monitor

- 데이터베이스 운영에 관련된 많은 통계정보 중 가장 중요하고 자주 보는 핵심적인 통계정보를 그래프를 이용하여 서버 현황을 파악

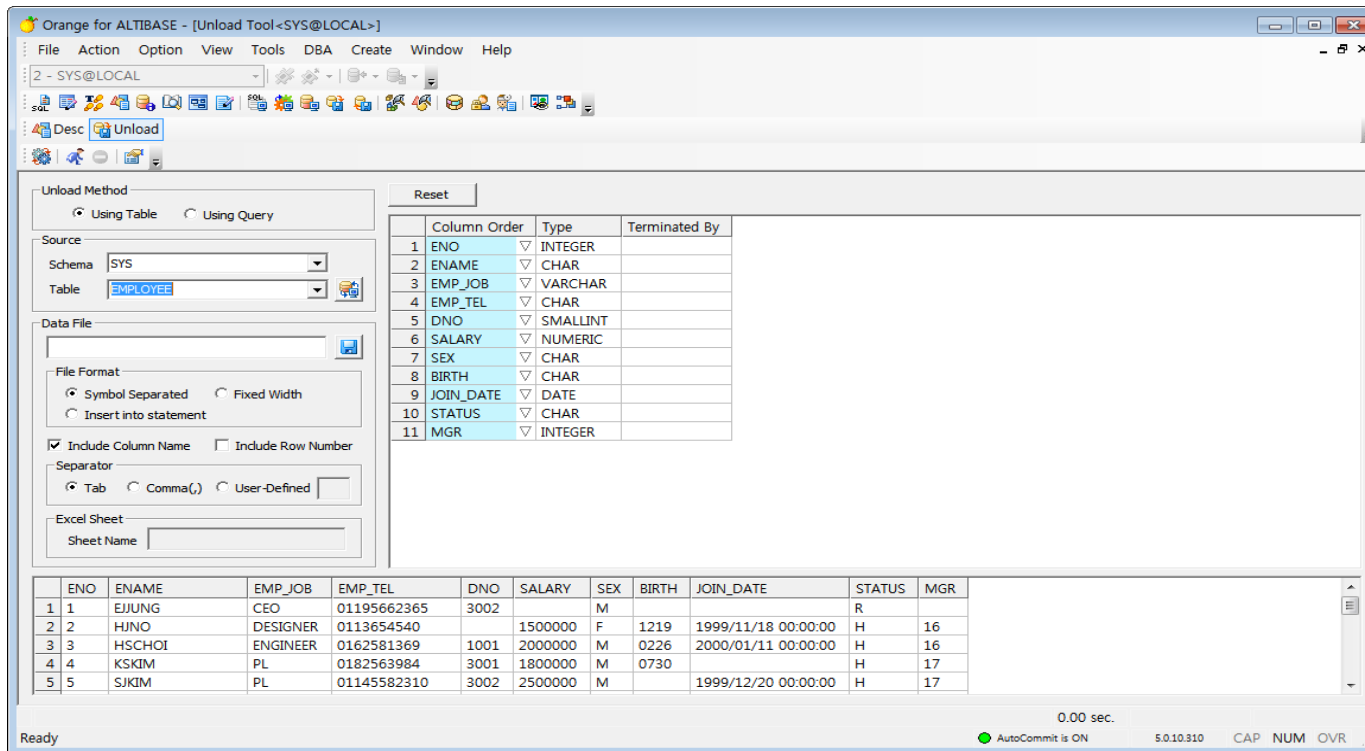


ORANGE FOR ALTIBASE

❖ ORANGE 상세 기능

➤ Unload Tool

- 데이터베이스의 테이블이나 VIEW를 선택하여 txt, csv, xls 등 다양한 포맷의 파일로 데이터를 내려 받을수 있는 기능을 제공

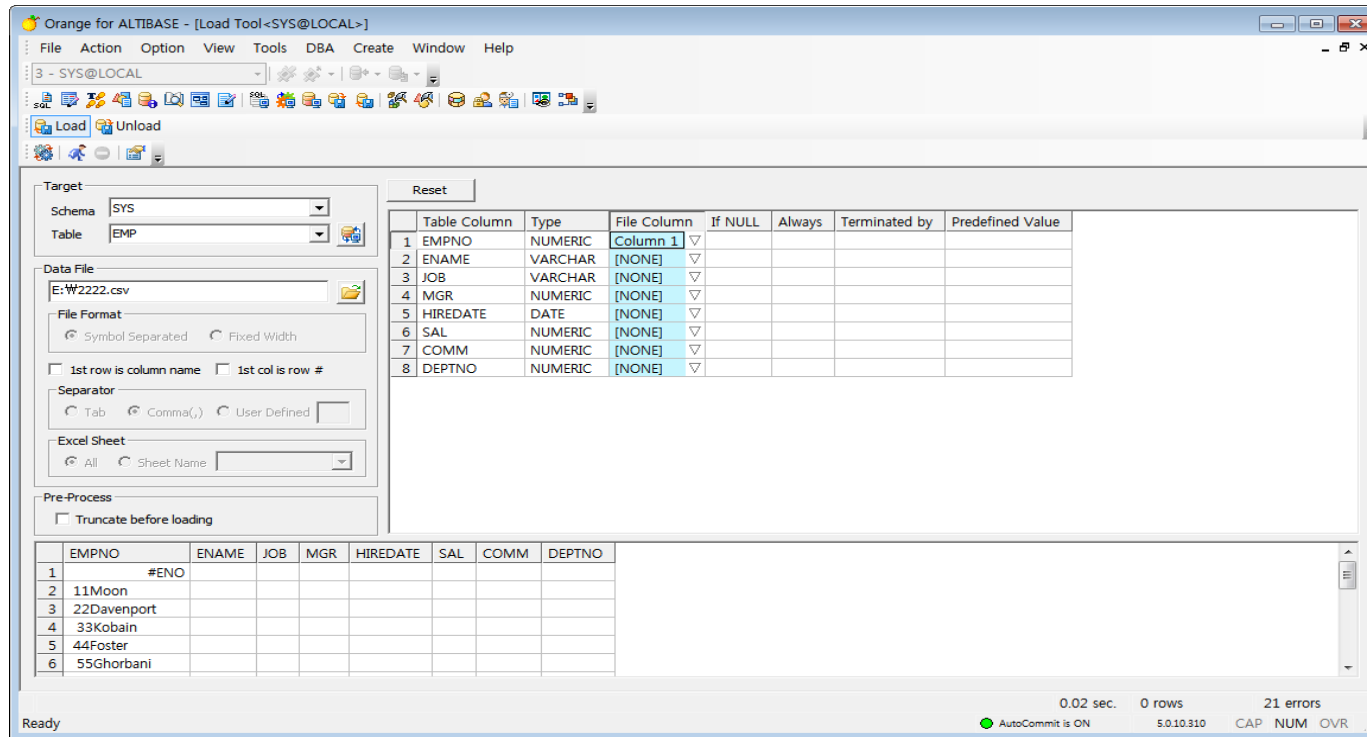


ORANGE FOR ALTIBASE

❖ ORANGE 상세 기능

➤ Load Tool

- Txt, csv, xls 등 다양한 포맷의 외부 파일로부터 데이터베이스의 테이블에 데이터로 Load 할수 있는 기능을 제공



ORANGE FOR ALTIBASE

❖ ORANGE 상세 기능

➤ Data Generation Tool

- 임의의 데이터들을 생성하여 원하는 테이블에 Insert 하는 역할

Configuration

User: USER1
Table: EMPLOYEE

Column Information

Column Name	Column Type	Nullable
1 ENO	INTEGER	Not Null
2 ENAME	CHAR	Not Null
3 EMP_JOB	VARCHAR	
4 EMP_TEL	CHAR	
5 DNO	SMALLINT	
6 SALARY	NUMERIC	
7 SEX	CHAR	Not Null

Table Option

If the Data exists on the table
 Truncate Statement Delete Statement
 #of Record to create: 10 All Random Column

Column Option

Generator: Sequence Format
 Allow NULLs: 1 % NULL
Initial Number: 1
Increment Value: 1

Output

```
1 INSERT
2 INTO "USER1"."EMPLOYEE" ( "ENO",
3 "ENAME",
4 "EMP_JOB",
5 "EMP_TEL",
6 "DNO",
7 "SALARY",
8 "SEX",
9 "BIRTH",
10 "JOIN_DATE",
11 "STATUS",
```

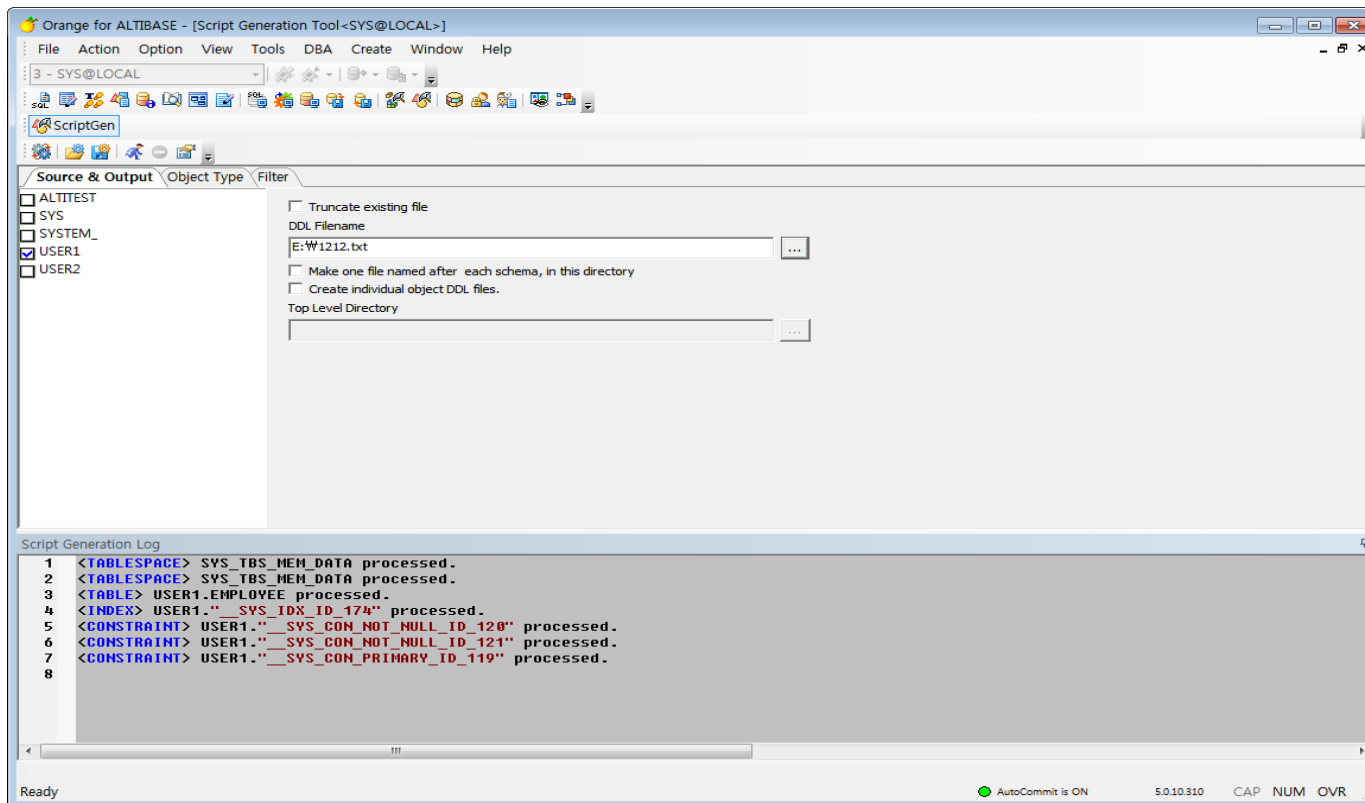
Ready AutoCommit is ON 5.0.10.310 CAP NUM OVR

ORANGE FOR ALTIBASE

❖ ORANGE 상세 기능

➤ Script Generation Tool

- 유저가 소유하고 있는 오브젝트를 스크립트 파일로 생성



ORANGE FOR ALTIBASE

❖ ORANGE 상세 기능

➤ Space Manager

- 테이블스페이스와 데이터 파일에 대한 사용현황을 다양한 방식으로 제공

The screenshot displays the 'Space Manager' window in Orange for ALTIBASE. The main area shows a table with the following data:

Tablespace Name	Usage(%)	Max Size(MB)	Current Size(MB)	Used Size(MB)	Type	State
Memory Tablespace	53.91	2048.00	1108.56	1104.06		
SYS_TBS_MEM_DIC			4.97	4.03	MEMORY_SYST...	ONLINE
SYS_TBS_MEM_DATA			1103.59	1100.03	MEMORY_SYST...	ONLINE
SYS_TBS_DISK_DATA	0.02	2047.98	100.00	0.50	DISK_SYSTEM_...	ONLINE
SYS_TBS_DISK_UNDO	0.26	2047.98	129.00	128.25	DISK_SYSTEM_...	ONLINE
SYS_TBS_DISK_TEMP	0.01	2047.98	100.00	0.25	DISK_SYSTEM_T...	ONLINE

Below the table, a 3D donut chart titled 'By Segment Type (KB)' is displayed. The chart shows a single green segment representing 'TABLE' segments. A legend below the chart identifies the colors: UNKNOWN (blue), TABLE (green), and INDEX (red).

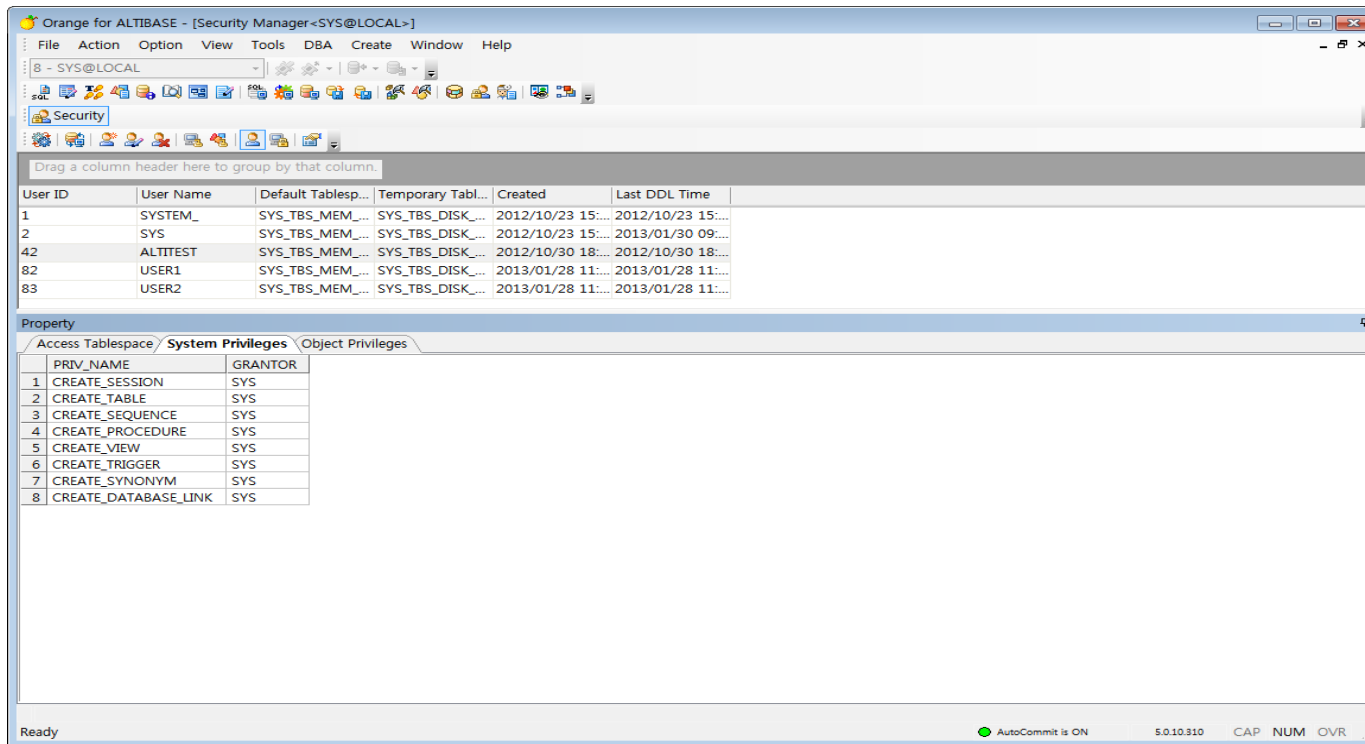
The interface also shows a tree view on the left with 'Tablespaces' and 'Datafiles' under 'SYS@LOCAL'. The status bar at the bottom indicates 'Ready', 'AutoCommit is ON', and version '5.0.10.310'.

ORANGE FOR ALTIBASE

❖ ORANGE 상세 기능

➤ Security Manager

- 데이터베이스에 정의된 사용자, 권한등 보안 관련 사항을 다양한 관점에서 손쉽게 조회하고 사용자 변경작업을 용이하게 할수 있는 기능을 제공



ORANGE FOR ALTIBASE

❖ ORANGE 상세 기능

➤ Health Check

- 데이터베이스가 가동이 된 이후 Session, Statement, Lock, Memory DB, Memory, Replication, GC, Logfile, Tablespace 등의 정보를 Reporting

The screenshot shows the 'Orange for ALTIBASE' application window with the 'Health Check' tool active. The main display area shows a 'Health Check Report' for 'Friday April 05 2013 17:40:14'. The report is divided into sections: 'Session', 'SYSDBA Session', and 'Progress'. The 'Session' section shows a 'Session Count' of 2. The 'SYSDBA Session' section shows a 'Purpose' of 'SYSDBA 자격으로 접속중인 세션 정보를 확인한다.' and a 'Criteria' of '200 Maximum'. The 'Progress' section shows 'Writing report...' and 'Elapsed Time (00:00:01)'. The left sidebar shows a tree view of 'Health Check Item' categories, including Memory DB, Memory, Replication, GC, Logfile, and Tablespace. The bottom status bar shows 'AutoCommit is ON', '5.0.10.310', 'CAP NUM OVR', and '완료'.

Orange for ALTIBASE - [Health Check<-SYS@LOCAL>]

File Action Option View Tools DBA Create Window Help

2 - SYS@LOCAL

Trace Health Chk

Health Check Item

- Memory DB
 - Memory DB Usage
 - Memory Table Maximum Size Top 10
 - Memory Table Efficiency
- Memory
 - Memory Total
 - Memory (by Module)
- Replication
 - Replication Sender
 - Replication Receiver
 - Replication Gap
- GC
 - Memory GC Gap
 - Disk GC Gap
- Logfile
 - Redo Logfile
 - Redo Logfile prepare wait count
- Tablespace
 - Disk Tablespace
 - Temo Tablespace

Description

Purpose:

Reference Value:

Health Check Report

Friday April 05 2013 17:40:14

Session

Session Count

Purpose	전체 Session 개수를 확인한다.
Criteria	200 Maximum

Session Count

2

SYSDBA Session

Purpose	SYSDBA 자격으로 접속중인 세션 정보를 확인한다.
Criteria	이 수치를 초과할 때는 사용자에게 알림을 하고, 그 후 해당 세션의 상태를 확인하여 종료한다.

Progress

Writing report...

Elapsed Time (00:00:01)

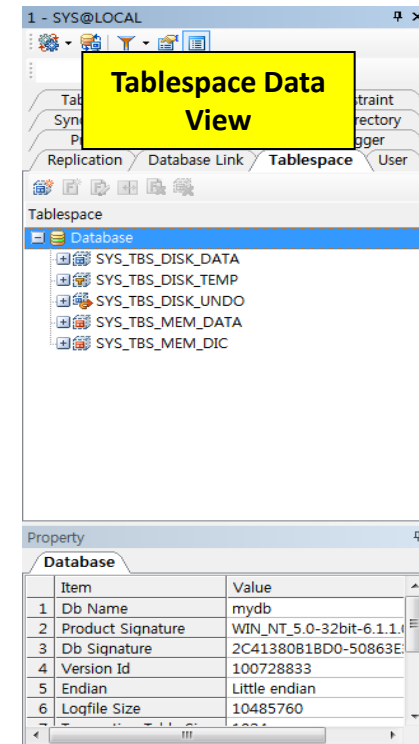
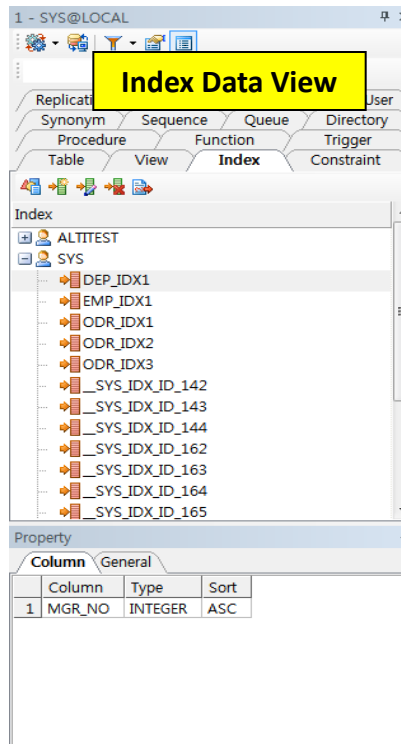
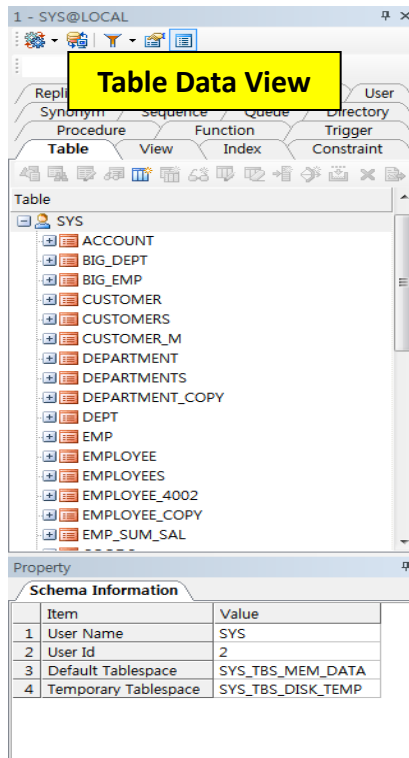
완료 AutoCommit is ON 5.0.10.310 CAP NUM OVR

ORANGE FOR ALTIBASE

❖ ORANGE 상세 기능

➤ Schema Browser

- Schema Object와 Tablespace, Replication, Schema 객체 정보등을 Tree 형태로 일목요연하게 보여줌

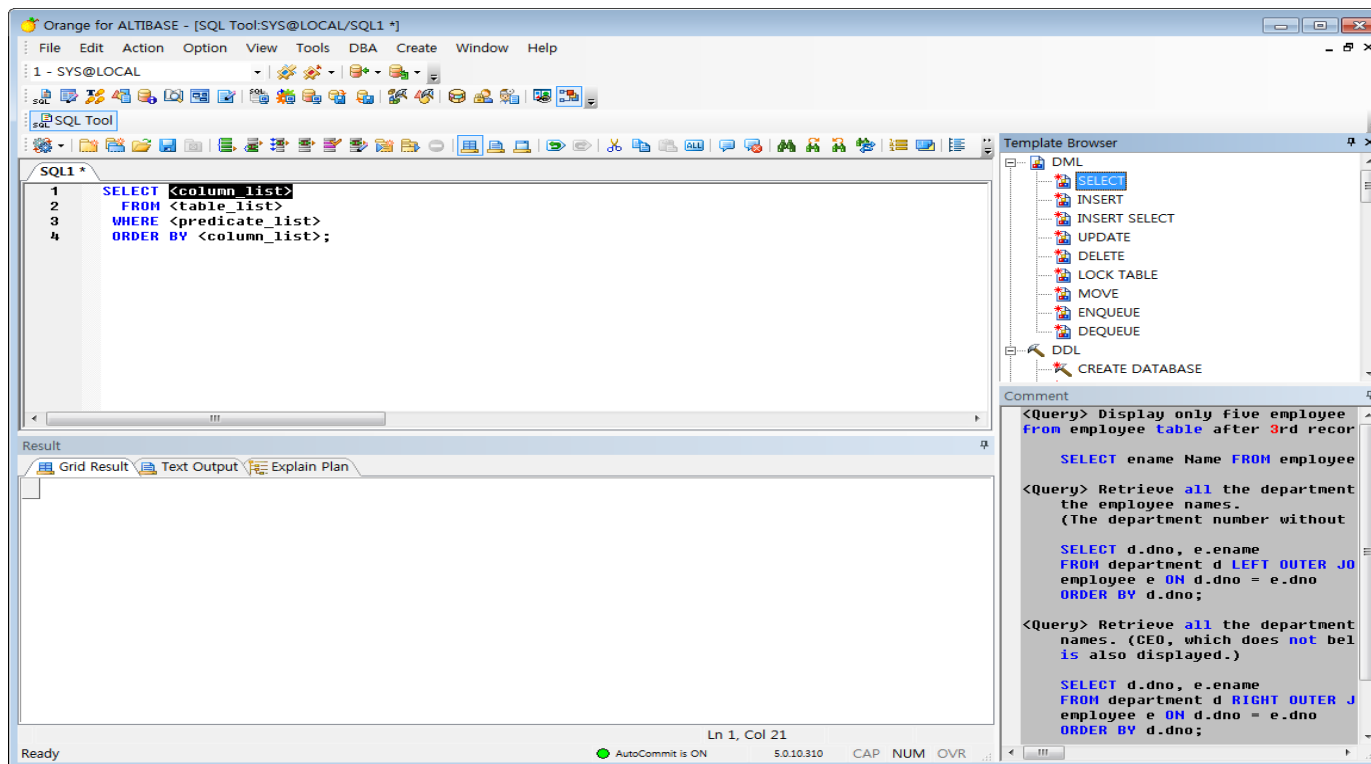


ORANGE FOR ALTIBASE

❖ ORANGE 상세 기능

➤ Template Browser

- 사용자가 DML, DDL, PL/SQL, SQL Function, SQL Optimizer와 같이 복잡한 SQL 문법을 쉽게 만들수 있도록 Template을 제공



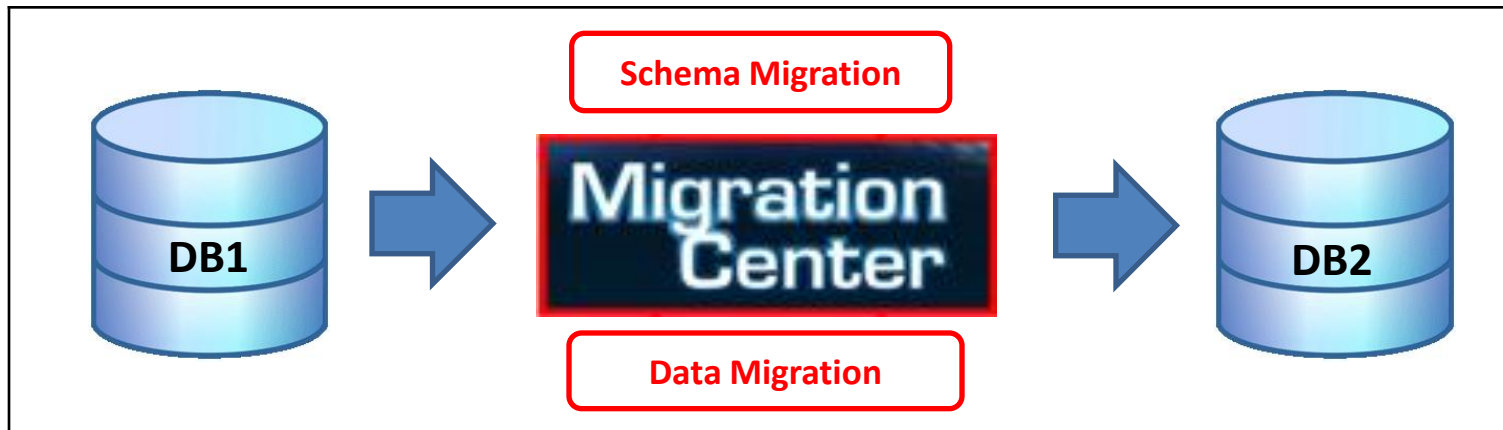


MIGRATION CENTER

MIGRATION CENTER

❖ MIGRATION CENTER

- 이기종 데이터베이스 간의 데이터 Migration을 간편하게 수행할수 있는 도구
- 주요이점
 - 원본 데이터베이스의 객체들을 ALTIBASE HDB로 쉽게 마이그레이션.
 - 데이터를 외부파일로 내보내어(export), iLoader로 ALTIBASE HDB에 가져올(import) 때 사용.
 - 마이그레이션 과정을 단축시킬 수 있는 병렬 수행 옵션을 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 HDB : 5.5.1 이상 버전
- Oracle Database : 10g 이상 버전
- Microsoft SQL Server : 2005 이상 버전
- Oracle MySQL : 5.0 이상 버전

MIGRATION CENTER

❖ 설치 및 제거

➤ 다운로드

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

➤ 설치

- Migration Center 설치 파일의 압축을 해제하고 디렉토리를 원하는 위치로 이동

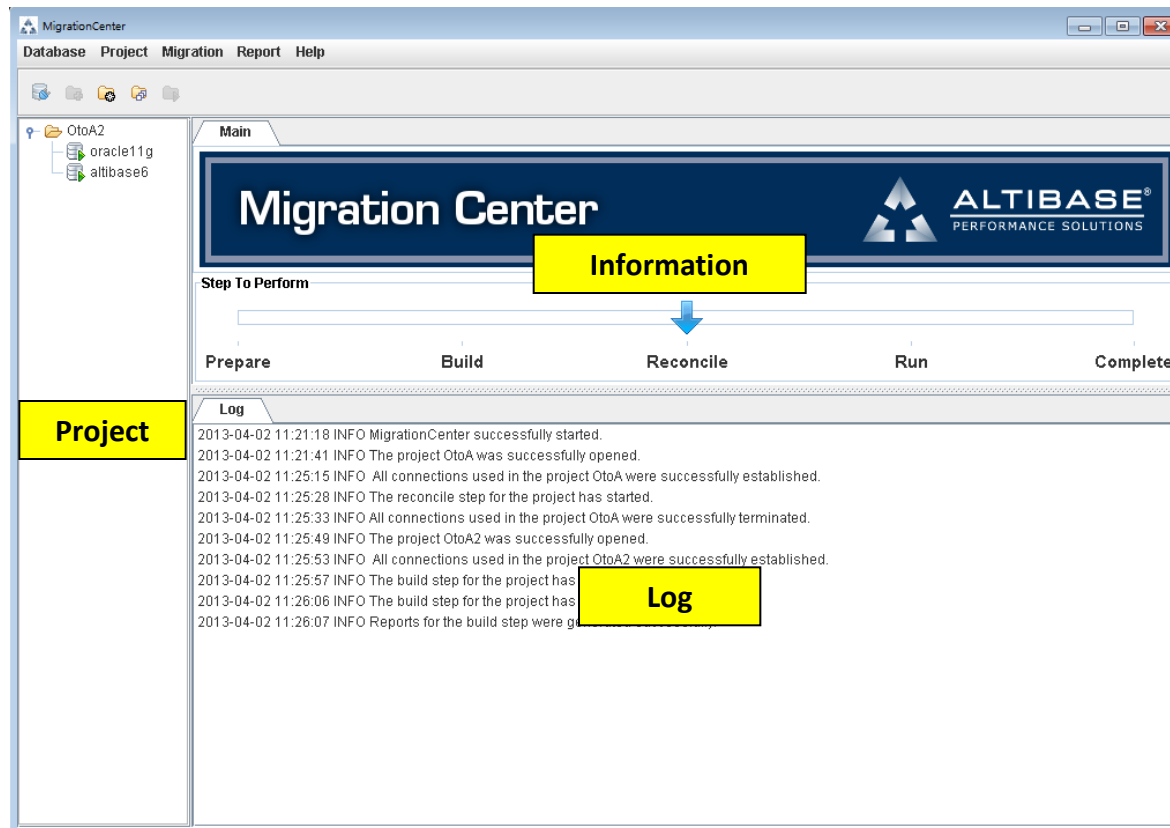
➤ 제거

- 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” 를 실행

❖ 데이터베이스 연결 등록

- “Database” 메뉴 아래의 “Add Database Connection” 메뉴항목을 선택하거나 도구 모음에서 “Add Database Connection” 아이콘을 클릭

The screenshot shows the 'Add Database Connection' dialog box for ORACLE. The fields are filled with the following information:

DB Product	ORACLE
Connection Name	Oracle11g
IP	127.0.0.1
Port	1521
User	scott
Password	*****
SID	ORCL
Option	IP Version: IPv4

Buttons at the bottom: Test, OK, Cancel.

The screenshot shows the 'Add Database Connection' dialog box for ALTIBASE. The fields are filled with the following information:

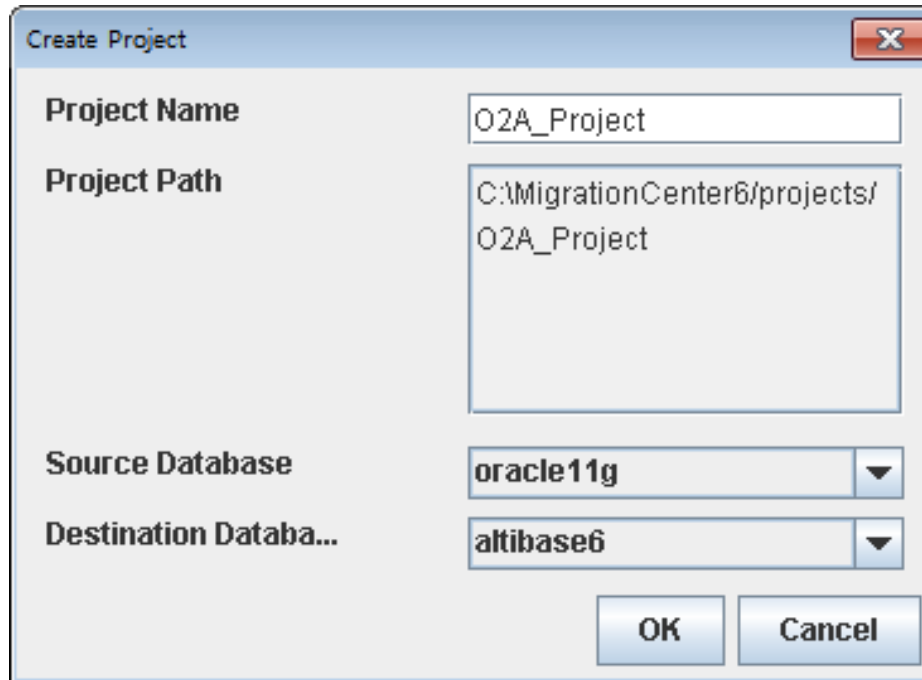
DB Product	ALTIBASE
Connection Name	Altibase6
IP	127.0.0.1
Port	20300
User	sys
Password	*****
JDBC Driver	/jdbc/Altibase_6.1.1.0.jar File
Option	Encoding: MS949, IP Version: IPv4

Buttons at the bottom: Test, OK, Cancel.

MIGRATION CENTER

❖ 프로젝트 생성

- “Project” 메뉴 아래의 “Create Project” 메뉴 항목을 선택하거나, 도구 모음에서 “Create Project” 아이콘을 클릭



Create Project

Project Name: O2A_Project

Project Path: C:\MigrationCenter6/projects/O2A_Project

Source Database: oracle11g

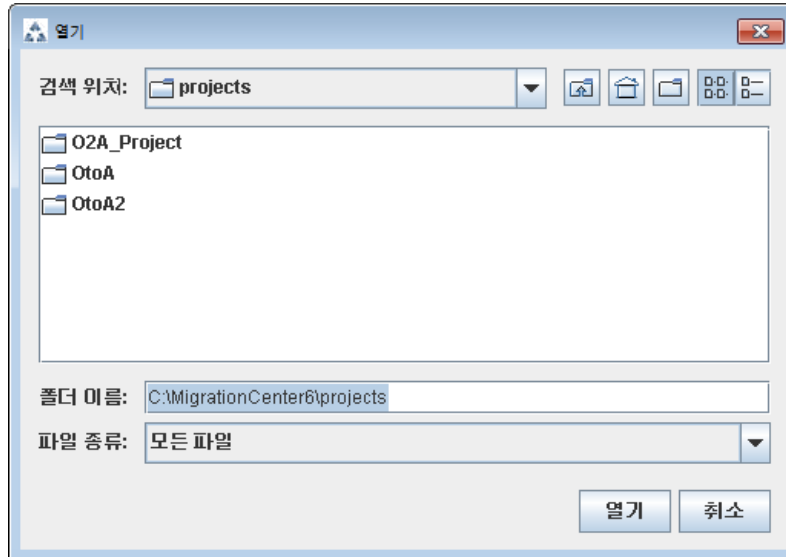
Destination Databa...: altibase6

OK Cancel

MIGRATION CENTER

❖ 프로젝트 열기

- “Project” 메뉴 아래의 “Open Project” 메뉴 항목을 선택



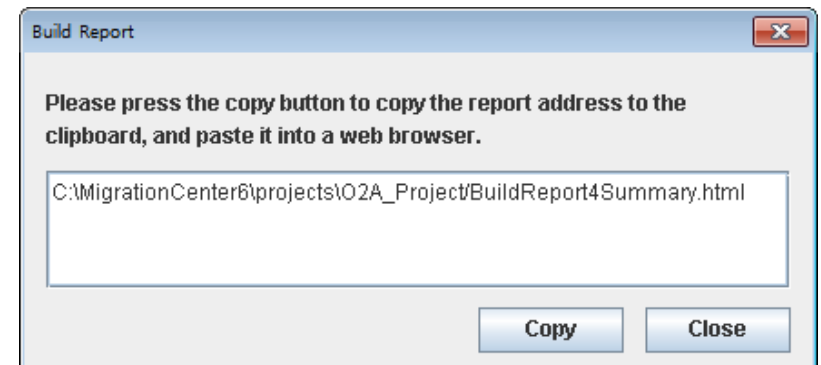
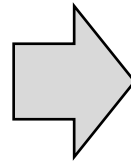
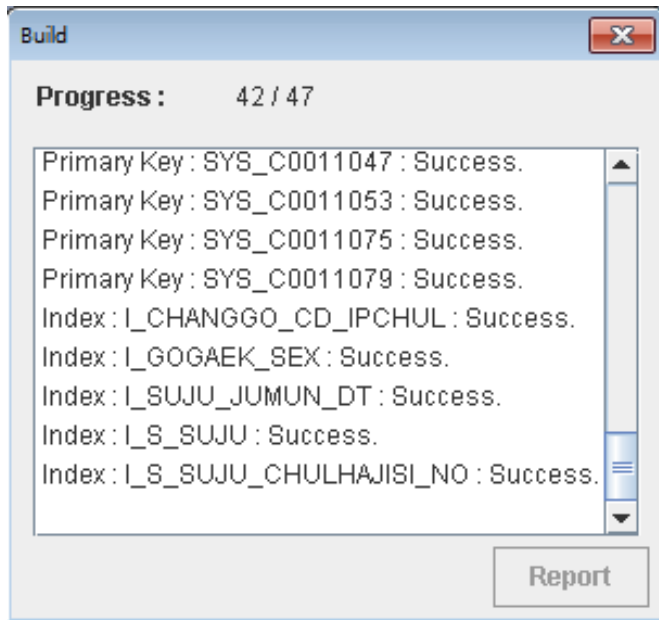
❖ 원본 및 대상 데이터베이스로 연결

- 원본 및 대상 데이터베이스로 연결은 “구축” 단계로 진행하기 위해 필수적 “Project” 메뉴 아래의 “Connect” 메뉴 항목을 선택하거나 열려진 프로젝트에 오른쪽 마우스 버튼을 클릭해서 “Connect”를 선택

MIGRATION CENTER

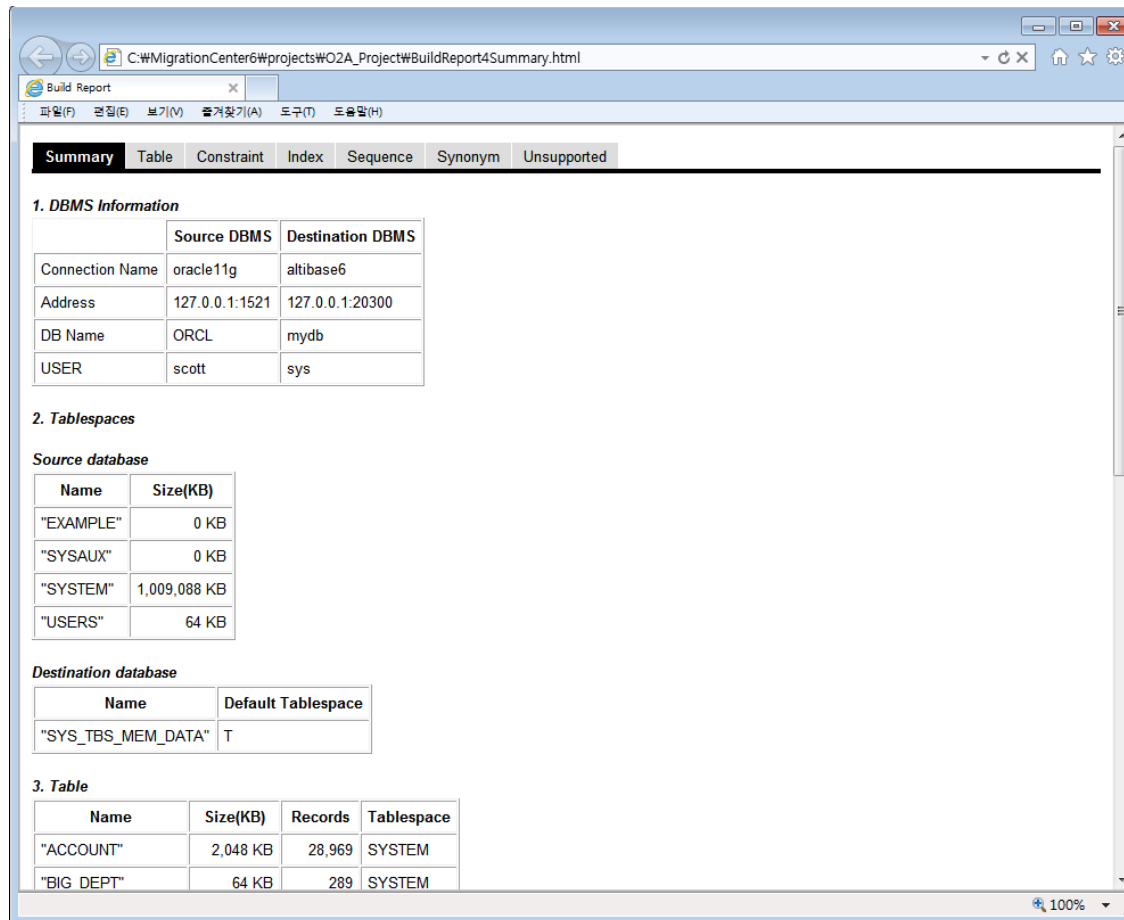
❖ 프로젝트 구축

- “Migration” 메뉴 아래의 “Build” 메뉴 아이템을 선택하거나, 도구 모음에서 “Build” 아이콘을 클릭



MIGRATION CENTER

❖ 프로젝트 구축 (Build Report)



The screenshot shows a web browser window displaying a "Build Report" summary page. The browser's address bar shows the file path: C:\MigrationCenter6\projects\#O2A_Project\#BuildReport4Summary.html. The page has a navigation menu with tabs for Summary, Table, Constraint, Index, Sequence, Synonym, and Unsupported. The "Summary" tab is selected.

1. DBMS Information

	Source DBMS	Destination DBMS
Connection Name	oracle11g	altibase6
Address	127.0.0.1:1521	127.0.0.1:20300
DB Name	ORCL	mydb
USER	scott	sys

2. Tablespaces

Source database

Name	Size(KB)
"EXAMPLE"	0 KB
"SYSAUX"	0 KB
"SYSTEM"	1,009,088 KB
"USERS"	64 KB

Destination database

Name	Default Tablespace
"SYS_TBS_MEM_DATA"	T

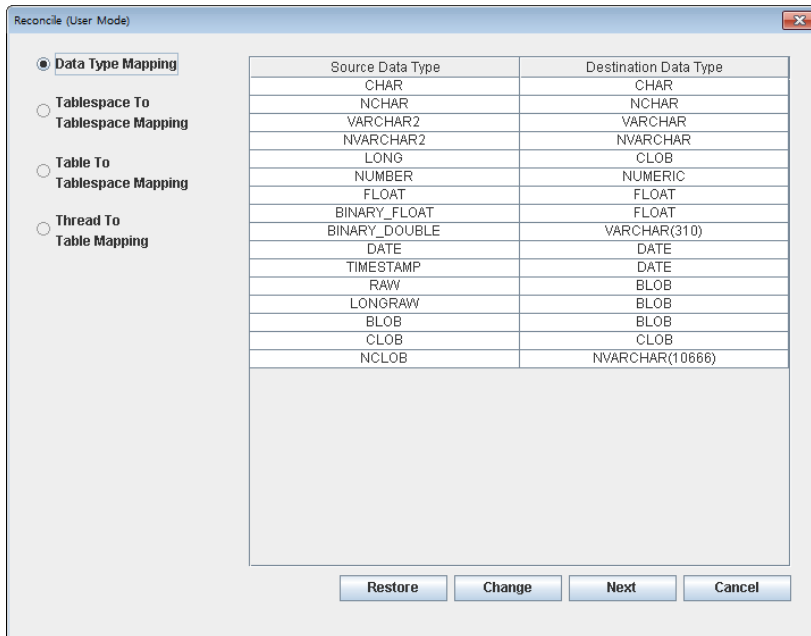
3. Table

Name	Size(KB)	Records	Tablespace
"ACCOUNT"	2,048 KB	28,969	SYSTEM
"BIG DEPT"	64 KB	289	SYSTEM

MIGRATION CENTER

❖ 프로젝트 조정

- “Migration” 메뉴 아래의 “Reconcile” 메뉴 항목을 선택하거나, 도구 모음에서 “Reconcile” 아이콘을 클릭



User Mode

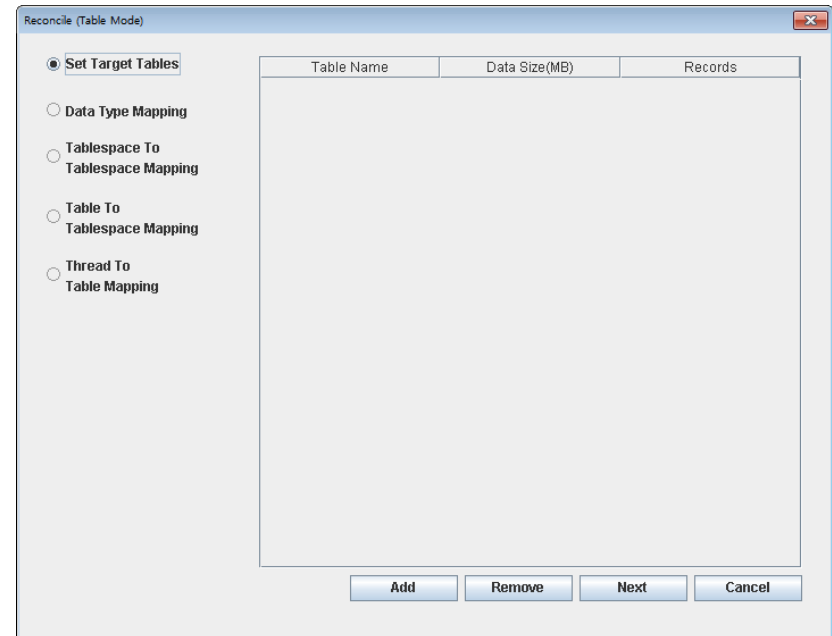
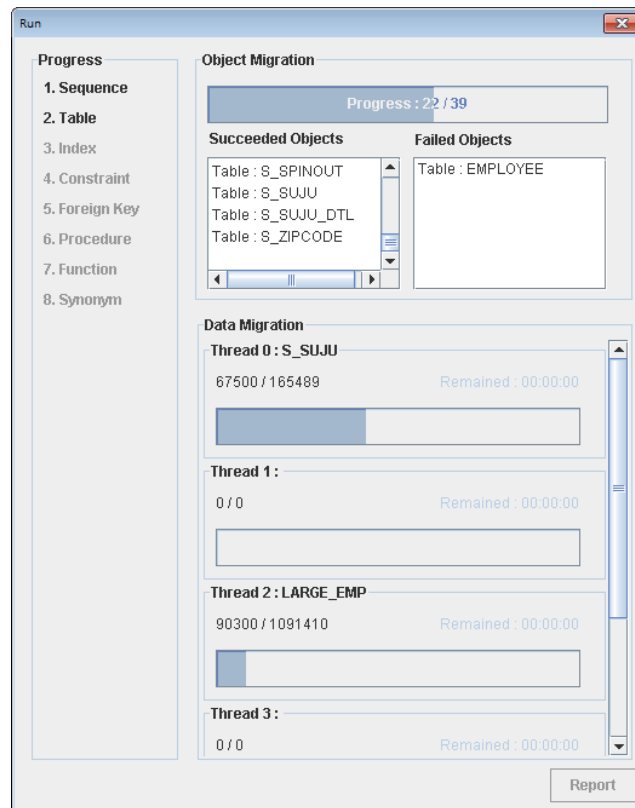


Table Mode

MIGRATION CENTER

❖ 프로젝트 실행

- “Migration” 메뉴 아래의 “Run” 메뉴 항목을 선택하거나, 도구 모음에서 “Run” 아이콘을 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 데이터베이스 to ALTIBASE HDB

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

MIGRATION CENTER

❖ 데이터 타입 매핑

➤ 오라클 데이터베이스 to ALTIBASE HDB

인덱스	소스	대상	특이 사항
1	CHAR	CHAR	
2	NCHAR	NCHAR	원본 및 대상 데이터베이스의 NCHAR 칼럼의 명시적인 크기는 같다(예.NCHAR(10) -> NCHAR(10)). 그러나, 오라클 JDBC 드라이버에서는 NCHAR 칼럼의 크기가 사용되는 바이트의 개수로 정의되는 반면, ALTIBASE HDB의 JDBC 드라이버에서는 NCHAR 칼럼의 크기가 저장되는 문자의 개수로 정의된다. 이는 ALTIBASE HDB에서 생성되는 NCHAR 칼럼이 필요에 따라 오라클보다 2 배 또는 3 배 정도 클 것이라는 의미이므로, 이런 점을 유의하도록 한다.
3	VARCHAR2	VARCHAR	
4	NVARCHAR2	NVARCHAR	NCHAR와 같은 이유로, 칼럼 크기가 서로 다르다.
5	LONG	CLOB	
6	NUMBER	NUMERIC	
7	FLOAT	FLOAT	
8	BINARY FLOAT	FLOAT	

MIGRATION CENTER

❖ 데이터 타입 매핑

➤ 오라클 데이터베이스 to ALTIBASE HDB

인덱스	소스	대상	특이 사항
9	BINARY DOUBLE	VARCHAR(310)	ALTIBASE HDB에는 오라클 BINARY DOUBLE 타입과 호환되는 데이터 타입이 없으므로 데이터 손실을 막기 위해 문자형으로 저장된다.
10	DATE	DATE	
11	TIMESTAMP	DATE	스케일의 차이로 인해서 소량의 데이터 손실이 발생할 수 있다. 오라클에서는 타임스탬프 값의 스케일이 나노초(9 자리수)인 반면, ALTIBASE HDB에서는 타임스탬프 값의 스케일이 마이크로초(6 자리 수)이다.
12	RAW	BLOB	
13	LONG RAW	BLOB	
14	BLOB	BLOB	
15	CLOB	CLOB	
16	NCLOB	NVARCHAR(10666)	ALTIBASE HDB에는 오라클 NCLOB 타입과 호환 가능한 데이터 타입이 없으므로, 최대 크기의 NVARCHAR 타입으로 변환된다. 실제 데이터 크기가 NVARCHAR 최대 크기를 초과하는 경우, 데이터를 마이그레이션하는 동안 데이터 손실이 발생할 수도 있다.

MIGRATION CENTER

❖ 데이터 타입 매핑

➤ MS SQL Server to ALTIBASE HDB

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)	ALTIBASE HDB에는 SQL Server FLOAT 타입과 호환 가능한 데이터 타입이 없으므로, 데이터 손실을 막기 위해 VARCHAR(310)으로 매핑된다.
11	REAL	FLOAT	
12	DATE	DATE	

MIGRATION CENTER

❖ 데이터 타입 매핑

➤ MS SQL Server to ALTIBASE HDB

인덱스	소스	대상	특이 사항
13	DATETIME2	DATE	스케일의 차이로 인해서 시간의 fraction 손실이 발생할 수 있다. SQL Server의 DATETIME2 타입 스케일이 나노초의 100 배(7 자리 수)인 반면, ALTIBASE HDB에서는 DATE 타입의 스케일이 마이크로초(6 자리 수)이다.
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)	ALTIBASE HDB에는 SQL Server NTEXT 타입과 호환 가능한 데이터 타입이 없으므로 최대 크기의 VARCHAR 타입이 사용된다. 실제 데이터 길이가 최대 NVARCHAR 크기를 초과하는 경우, 데이터를 마이그레이션하는 동안 데이터 손실이 발생할 수도 있다.

MIGRATION CENTER

❖ 데이터 타입 매핑

➤ MS SQL Server to ALTIBASE HDB

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

MIGRATION CENTER

❖ 데이터 타입 매핑

➤ MySQL to ALTIBASE HDB

인덱스	소스	대상	특이 사항
1	TINYINT	SMALLINT	
2	SMALLINT	INTEGER	
3	MEDIUMINT	INTEGER	
4	INT(INTEGER)	BIGINT	
5	BIGINT	NUMERIC(20,0)	
6	DECIMAL (NUMERIC)	VARCHAR(70)	ALTIBASE HDB에는 MySQL DECIMAL 타입과 호환 가능한 데이터 타입이 없으므로, 데이터 손실을 막기 위해 VARCHAR 타입으로 매핑된다.
7	FLOAT	FLOAT	
8	DOUBLE	VARCHAR(310)	ALTIBASE HDB에는 MySQL DOUBLE 타입과 호환 가능한 데이터 타입이 없으므로, 데이터 손실을 막기 위해 VARCHAR 타입으로 매핑된다.
9	BIT	BIT	
10	DATETIME	DATE	시각 부분이 0으로 설정된다.
11	TIMESTAMP	DATE	TIMEZONE 제외

MIGRATION CENTER

❖ 데이터 타입 매핑

➤ MySQL to ALTIBASE HDB

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	BYTE	
19	TINYTEXT	VARCHAR(255)	
20	BLOB	BLOB	
21	TEXT	CLOB	
22	MEDIUMBLOB	BLOB	

MIGRATION CENTER

❖ 데이터 타입 매핑

➤ MySQL to ALTIBASE HDB

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



실습교재 II



PERFORMANCE VIEW

PERFORMANCE VIEW

❖ Performance View 실습

- iSQL로 접속한 후 해당 iSQL의 세션정보를 확인하라 (process id, 접속IP정보)

```
iSQL> SELECT client_pid, comm_name FROM v$session  
2 WHERE id = SESSION_ID();
```

- 임의의 테이블과 질의를 수행한 후 어떠한 정보가 v\$statement를 통해 조회가 되는지 확인하라

```
iSQL> SELECT  
iSQL> SELECT LPAD(query, 100, ' '), execute_time, execute_success  
2 FROM v$statement  
3 WHERE session_id = SESSION_ID();
```

- 현재 ALTIBASE HDB의 각 모듈별 메모리 사용량 중 상위 5개를 출력하라

```
iSQL> SELECT name, max_total_size FROM v$memstat  
2 ORDER BY max_total_size DESC LIMIT 5;
```

PERFORMANCE VIEW

❖ Performance View 실습

- 다음의 질의가 2개의 iSQL을 실행해 순차적으로 처리된 후 LOCK정보를 확인하라

Time	세션 A	세션 B
t1	iSQL> autocommit off;	iSQL> autocommit off;
t2	iSQL> DROP TABLE t1 ; iSQL> CREATE TABLE t1 (a INTEGER); iSQL> INSERT INTO t1 VALUES (10); iSQL> COMMIT;	
t3	iSQL> UPDATE t1 SET a = a + 1;	
t4		iSQL> UPDATE t1 SET a = a + 1; (Lock Wait상태 유발)
t5	iSQL> SELECT * FROM v\$lock_wait; iSQL> SELECT query, tx_id FROM v\$statement 2 WHERE tx_id= 조회된 ID 값 ;	
t6	iSQL> commit ;	



BACKUP & RECOVERY

ALTIBASE

iLOADER

ILOADER

❖ iLoader 실습

➤ 간단한 데이터 다운로드

■ Form파일 생성

```
Shell::~/home/alti1> mkdir work
```

```
Shell::~/home/alti1> cd work
```

```
Shell::~/home/alti1/work> iloader -u sys -p manager -port 20301 -s 127.0.0.1 formout -T TB_TEST1 -f TB_TEST1.fmt
```

■ 다운로드

```
Shell::~/home/alti1/work> iloader -u sys -p manager -port 20301 -s 127.0.0.1 out -f TB_TEST1.fmt -d TB_TEST1.dat
```

■ 결과 확인

```
Shell::~/home/alti1/work> vi TB_TEST1.fmt
```

```
Shell::~/home/alti1/work> vi TB_TEST1.dat
```

ILOADER

❖ iLoader 실습

- 간단한 데이터 업로드
 - 업로드

```
iSQL> SELECT COUNT(*) FROM tb_test1;
```

```
iSQL> TRUNCATE TABLE tb_test1;
```

```
Shell::~/home/alti1/work> iloader -u sys -p manager -port 20301 -s 127.0.0.1 in -f TB_TEST1.fmt -  
d TB_TEST1.dat -log TB_TEST1.log
```

- 결과 확인

```
Shell::~/home/alti1/work> cat TB_TEST1.log
```

ILOADER

❖ iLoader 실습

- 데이터 업로드(성능 옵션 사용)
 - 슬라이드 노트의 실습 실행 후, 진행

```
Shell::/home/alti1/work> il formout -T t1 -f t1.fmt
```

```
Shell::/home/alti1/work> il out -f t1.fmt -d t1.dat -log t1.log -bad t1.bad
```

```
Shell::/home/alti1/work> is
```

```
iSQL> TRUNCATE TABLE t1 ;
```

```
Shell::/home/alti1/work> il in -f t1.fmt -d t1.dat → 성능 옵션 미사용 시 시간측정
```

```
iSQL> TRUNCATE TABLE t1 ;
```

```
Shell::/home/alti1/work> il in -f t1.fmt -d t1.dat -array 50 -parallel 4 → 성능 옵션 사용시 시간측정
```

ILOADER

❖ iLoader 실습

- 잘못된 데이터파일에 의한 오류 확인
 - 이미 받은 TB_TEST1.dat 파일에서 쌍따옴표 구분자 하나를 지우고 저장

```
Shell::/home/alti1/work> vi TB_TEST1.dat
```

- 업로드 수행

```
iSQL> SELECT COUNT(*) FROM tb_test1;
```

```
iSQL> TRUNCATE TABLE tb_test1;
```

```
Shell::/home/alti1/work> il in -f TB_TEST1.fmt -d TB_TEST1.dat -log TB_TEST1.log -bad  
TB_TEST1.bad
```

- 결과 확인

```
Shell::/home/alti1/work> vi TB_TEST1.log
```

```
Shell::/home/alti1/work> vi TB_TEST1.bad → 지웠던 “ 를 추가하고 저장
```

```
Shell::/home/alti1/work> il in -f TB_TEST1.fmt -d TB_TEST1.bad
```

```
iSQL> SELECT COUNT(*) FROM tb_test1;
```

ILOADER

❖ iLoader 실습

➤ 폼파일의 편집 실습 (조건절 사용)

- TB_TEST1에서 (C1 > 5000) 인 조건의 데이터만 다운로드

```
Shell:~/home/alti1/work> vi TB_TEST1.fmt
```

맨 마지막에

DOWNLOAD CONDITION "WHERE c1 > 5000" 을 추가

- 다운로드 수행

```
iSQL> SELECT COUNT(*) FROM tb_test1 WHERE c1 > 5000;
```

```
Shell:~/home/alti1/work> iloader -u sys -p manager -port 20301 -s 127.0.0.1 out -f TB_TEST1.fmt  
-d TB_TEST1.dat
```

- 결과 확인

```
Shell:~/home/alti1/work> wc -l TB_TEST1.dat
```

```
5000 TB_TEST1.dat
```

ILOADER

❖ iLoader 실습

➤ 폼파일의 편집 실습 (DATE 포맷의 변경)

- DATE 포맷의 변경

```
Shell::/home/alti1/work> vi TB_TEST1.fmt  
맨 마지막에  
DATEFORM YYYY-MM-DD HH:MI:SS.SSSSSS 으로 변경
```

- 다운로드 수행

```
Shell::/home/alti1/work> il out -f TB_TEST1.fmt -d TB_TEST1.dat
```

- 결과 확인 (사용자가 변경한 DATE형식으로 다운로드 되었는지 확인)

```
Shell::/home/alti1/work> vi TB_TEST1.dat
```

ILOADER

❖ iLoader 실습

- 폼파일의 편집 실습 (Sequence의 사용)
 - 실습 테이블 다운로드 및 폼파일 편집(슬라이드 노트에서 시퀀스 생성부터 실습)

```
Shell::/home/alti1/work> il formout -T seqTable -f seqTable.fmt
```

```
shell::/home/alti1/work> il out -f seqTable.fmt -d seqTable.dat
```

```
Shell::/home/alti1/work> vi seqTable.fmt
```

```
SEQUENCE seq1 C1      ← 사용자 추가
```

```
table seqTable
```

```
{
```

```
C1 integer;
```

```
C2 VARCHAR(20);
```

```
}
```

```
Shell::/home/alti1/work> il in -f seqTable.fmt -d seqTable.dat -mode truncate
```

```
iSQL> SELECT * FROM seqTable;
```


ILOADER

❖ iLoader 실습

- 폼파일의 편집 실습 (함수의 사용을 통한 데이터 변환)
 - 실습 테이블 다운로드 및 폼파일 편집

```
Shell::~/home/alti1/work> vi seqTable.fmt
```

```
SEQUENCE seq1 C1 → 기존에 있던 구문 제거
```

```
table seqTable
```

```
{
```

```
C1 integer;
```

```
C2 VARCHAR(20) "CONCAT (TRIM(?), '/TEST')"; ← 추가
```

```
}
```

```
Shell::~/home/alti1/work> il in -f seqTable.fmt -d seqTable.dat -mode truncate
```

```
iSQL> SELECT * FROM seqTable;
```

ALTIBASE

AEXPORT

AEXPORT

❖ aexport 실습

➤ 실행

- 모든 DB객체의 생성 스크립트를 백업

```
Shell:~/home/alti1> mkdir work1
```

```
Shell :~/home/alti1> cd work1
```

```
Shell :~/home/alti1/work1> aexport (실행 후 user: sys, pw: manager 입력)
```

- 실행결과

```
Shell:~/home/alti1/work1> ls
```

ALL_CRT_PROC.sql	SYS_CRT_REP.sql	run_il_out.sh
ALL_CRT_TBS.sql	SYS_CRT_SYNONYM.sql	run_is.sh
ALL_CRT_VIEW.sql	SYS_CRT_TBL.sql	run_is_fk.sh
SYS_CRT_DIRECTORY.sql	SYS_CRT_USER.sql	run_is_index.sh
SYS_CRT_FK.sql	SYS_PRIV_USER.sql	run_is_repl.sh
SYS_CRT_INDEX.sql		run_il_in.sh

AEXPORT

❖ aexport 실습

➤ 실행

- 사용자 user1객체의 생성 스크립트를 백업

```
Shell:~/home/alti1> is -f schema_edu.sql  
Shell:~/home/alti1> mkdir work2  
Shell :~/home/alti1> cd work2  
Shell :~/home/alti1/work2> aexport -s 127.0.0.1 -u user1 -p user1
```

- 실행결과

```
Shell:~/home/alti1/work2> ls  
USER1_CRT_DIR.sql          USER1_CRT_TBL.sql          run_is.sh  
USER1_CRT_FK.sql          USER1_CRT_TRIG.sql         run_is_fk.sh  
USER1_CRT_INDEX.sql       USER1_CRT_USER.sql        run_is_index.sh  
USER1_CRT_LINK.sql        USER1_CRT_VIEW_PROC.sql   run_is_repl.sh  
USER1_CRT_SEQ.sql         run_il_in.sh  
USER1_CRT_SYN.sql         run_il_out.sh
```

AEXPORT

❖ aexport 실습

➤ 실행

- User1 사용자의 employee 테이블과 t1_view 의 생성 스크립트를 백업

```
iSQL> CONNECT user1/user1
```

```
iSQL> CREATE VIEW t1_view
```

```
2 AS
```

```
3 SELECT eno, ename, salary
```

```
4 FROM employee;
```

```
Shell ::/home/alti1> cd work2
```

```
Shell ::/home/alti1/work2> aexport -s 127.0.0.1 -u user1 -p user1 -object user1.employee
```

```
Shell ::/home/alti1/work2> aexport -s 127.0.0.1 -u user1 -p user1 -object user1.t1_view
```

- 실행결과

```
Shell:~/home/alti1/work> ls
```

```
user1_employee_CRT.sql  user1_t1_view_CRT.sql
```

ALTIBASE

ALTIPASSWD

❖ altiPasswd 실습

➤ 실행

- DBA권한을 가진 SYS계정의 패스워드 변경 실습

```
iSQL> ALTER USER sys IDENTIFIED BY ppp1234;
```

```
Alter success.
```

```
shell::/home/alti1> altipasswd
```

```
Previous Password : manager
```

```
New Password : ppp1234
```

```
Retype New Password : ppp1234
```

```
shell::/home/alti1> vi $ALTIBASE_HOME/bin/server, is, il 스크립트를 수정
```

```
➔ SYS계정의 패스워드 지정 부분을 수정
```

ALTIBASE

ALTIPROFILE

ALTIPROFILE

❖ altiProfile 실습

➤ profile의 시작

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

➤ DML을 자유롭게 수행

```
iSQL> DELETE FROM tb_test1 WHERE c1<5100;  
iSQL> SELECT c1, c2 FROM tb_test1 WHERE c1 > 5100 LIMIT 10 ;
```

➤ profile의 종료

```
iSQL> ALTER SYSTEM SET QUERY_PROF_FLAG = 0 ;
```

ALTIPROFILE

❖ altiProfile 실습

➤ profile의 결과 생성

```
shell::/home/alti1> cd $ALTIBASE_HOME/trc
```

```
shell::/home/alti1/altibase_home/trc> altiProfile alti-xxxxxxx-0.prof > prof.txt (*.prof 파일명은 실행 시  
점 별로 다름)
```

```
shell::/home/alti1/altibase_home/trc> vi prof.txt
```

(실습생이 실행한 쿼리들에 대한 분석결과를 확인)

ALTIBASE

DUMP

DUMPLA

❖ dumpla 실습

➤ 실행

- dumpla 실습

```
Shell::~/home/alti1> cd $ALTIBASE_HOME/logs  
Shell::~/home/alti1/altibase_home/logs> dumpla loganchor0 > res.txt  
Shell::~/home/alti1/altibase_home/logs> vi res.txt
```

- dumpla를 수행한 결과파일에서 End LSN 정보를 확인
- dumpla를 수행한 결과파일에서 Disk Redo LSN 정보를 확인
- dumpla를 수행한 결과파일에서 각 데이터파일의 정보들을 확인

DUMPLF

❖ dumplf 실습

➤ 실행

- 마지막 LSN 정보를 통해 어떤 로그파일을 dump해볼지 선정

```
Shell::/home/alti1/altibase_home/logs> dumpla loganchor0 > an.txt  
Shell::/home/alti1/altibase_home/logs> grep "End LSN" an.txt  
Shell::/home/alti1/altibase_home/logs> dumplf -f logfileXX > XX_before.txt  
iSQL> DROP TABLE t1;  
iSQL> CREATE TABLE t1(c1 integer);  
iSQL> INSERT INTO t1 VALUES(1);  
iSQL> INSERT INTO t1 VALUES(2);
```

- iSQL상에서 임의의 DDL, DML을 수행해본다.
- 위에서 확인한 동일 로그파일을 dump한 후 DDL/DML 수행 전/후의 결과를 비교

```
Shell::/home/alti1/altibase_home/logs> dumplf -f logfileXX > XX_after.txt
```

DUMPDDF

❖ dumpddf 실습

➤ 실행

- system001.dbf 파일을 대상으로 선정

```
shell::~/home/alti1> cd $ALTIBASE_HOME/dbs
```

```
shell::~/home/alti1/altibase_home/dbs> dumpddf -m -f system001.dbf
```

```
shell::~/home/alti1/altibase_home/dbs> dumpddf -m -f system001.dbf -p 27 > res.log
```



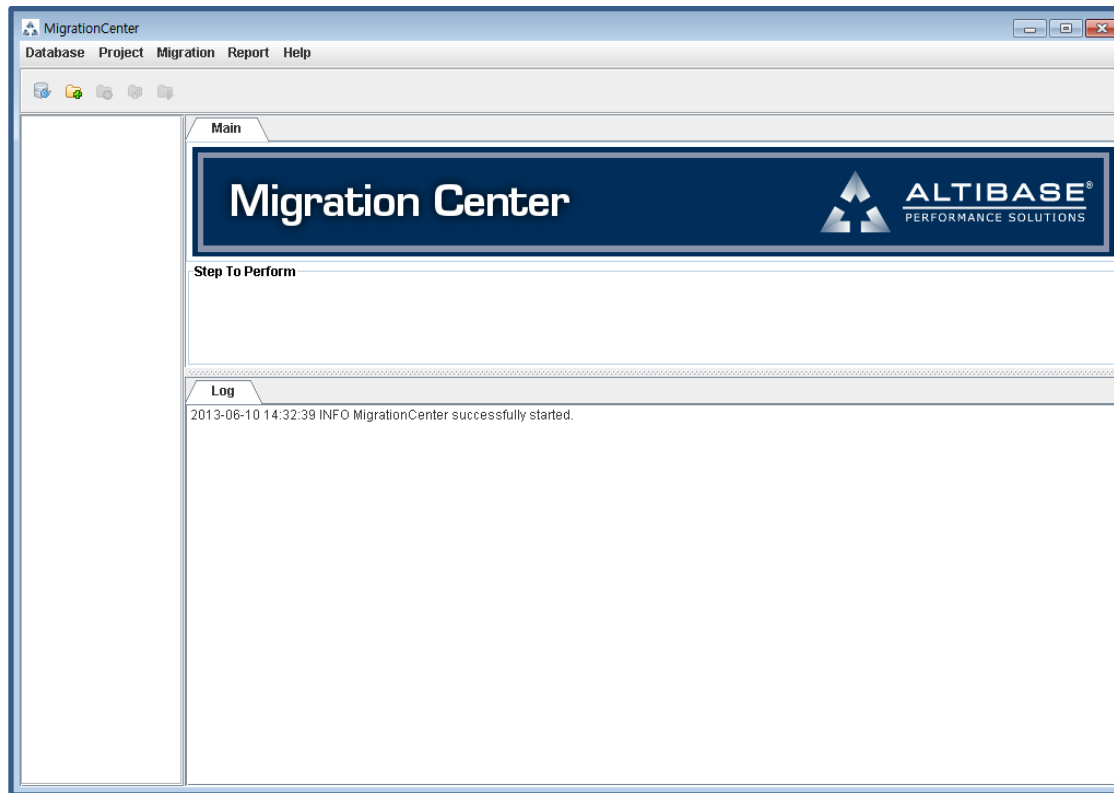
MIGRATION CENTER

Migration Center

❖ Migration Center 실습

➤ 실행

- migcenter.bat 파일을 더블 클릭하여 실행



Migration Center

❖ Migration Center 실습

- 데이터베이스 등록
 - 오라클 DB 정보 등록 후 접속 테스트

Add Database Connection

DB Product: ORACLE

Connection Name: ORA11G

IP: 127.0.0.1

Port: 1521

User: scott

Password: *****

SID: ORCL

Option: IP Version: IPv4

Test OK Cancel

Add Database Connection

DB Product: ORACLE

Connection Name: ORA11G

IP: localhost

Port: 1521

Information: Successfully connected to the DB with the provided values

확인

IP Version: IPv4

Test OK Cancel

Migration Center

❖ Migration Center 실습

➤ 데이터베이스 등록

- ALTIBASE HDB 정보 등록 후 접속 테스트

Add Database Connection

DB Product: ALTIBASE

Connection Name: ALT16

IP: 127.0.0.1

Port: 20300

User: sys

Password: *****

JDBC Driver: /jdbc/Altibase_6.1.1.0.jar **File**

Option

Encoding: MS949

IP Version: IPv4

Test OK Cancel

Add Database Connection

DB Product: ALTIBASE

Connection Name: ALT16

IP: 127.0.0.1

Port: 20300

Encoding: MS949

IP Version: IPv4

Test OK Cancel

Information

Successfully connected to the DB with the provided values

확인

Migration Center

❖ Migration Center 실습

➤ 프로젝트 등록

- 등록된 데이터베이스를 기준으로 신규 프로젝트를 등록

Create Project

Project Name: OtoA_Proj

Project Path: C:\MigrationCenter6\projects\OtoA_Proj

Source Database: ORA11G

Destination Database: ALTI 6

OK Cancel

MigrationCenter

Database Project Migration Report Help

OtoA_Proj
ORA11G
ALTI 6

Migration Center

ALTIBASE®
PERFORMANCE SOLUTIONS

Step To Perform

Information

A dialog box for setting migration options will now appear, but you can change these options later by selecting the Options menu subitem from the Migration menu item.

확인

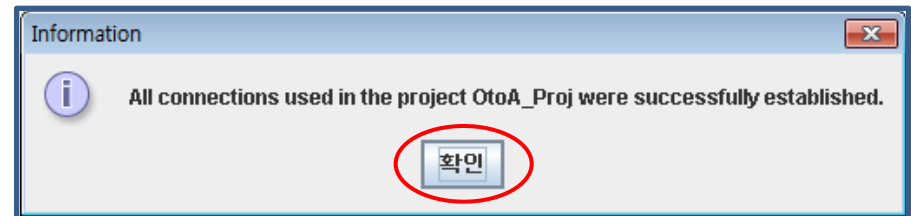
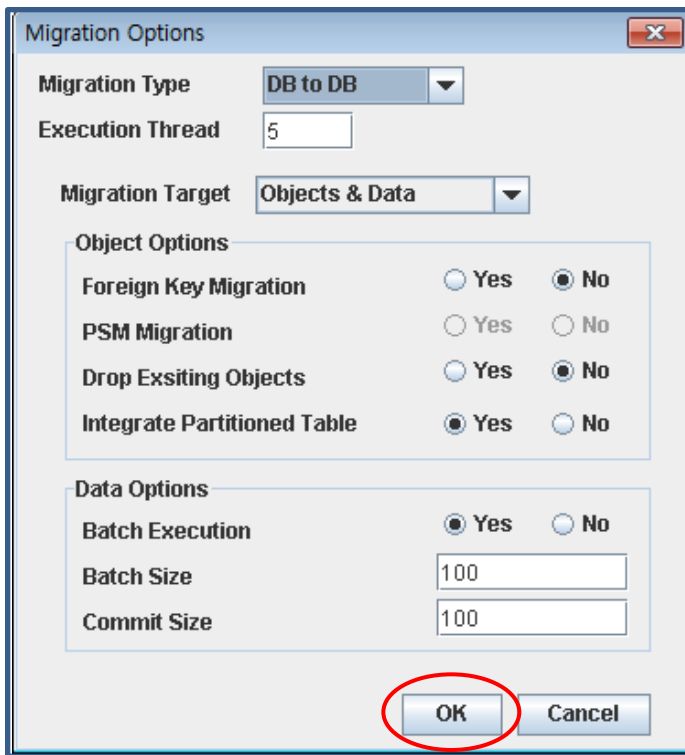
2013-06-14 10:16:17 WARN [ERROR] Unable to connect to DB.
[CAUSE] Unknown host: 127.0.0.1
[ACTION]
2013-06-14 10:18:17 ERROR Fail to fetch DB version: jdbc:Altibase://127.0.0.1:20300/mydb
2013-06-14 10:18:45 INFO Successfully connected to the DB with the provided values
2013-06-14 10:18:46 WARN [ERROR] Fail to add DB Connection.
[CAUSE] Duplicate name found:ALTI6
[ACTION]
2013-06-14 10:18:56 INFO [ERROR] Fail to add DB Connection.
[CAUSE] Duplicate name found:ALTI6
[ACTION]
Please review your settings and correct any errors.
2013-06-14 10:19:11 INFO Successfully connected to the DB with the provided values
2013-06-14 10:19:12 INFO The database connection ALTI 6 was added successfully.
2013-06-14 10:34:01 INFO The project OtoA_Proj was successfully created.

Migration Center

❖ Migration Center 실습

➤ 프로젝트 등록

- Migration Options 을 지정후 “Project” - “Connect” 클릭

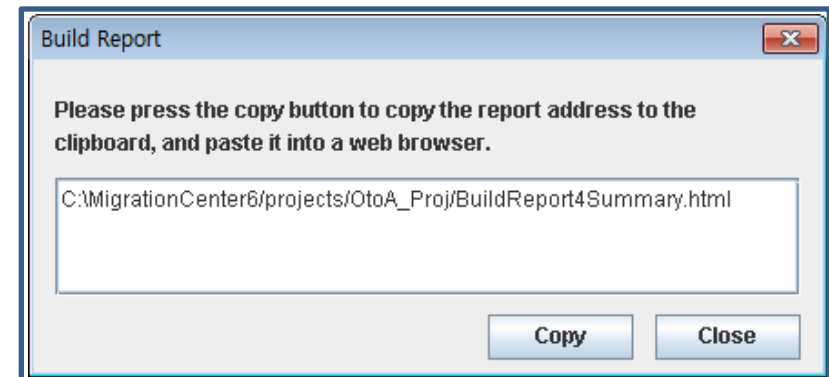
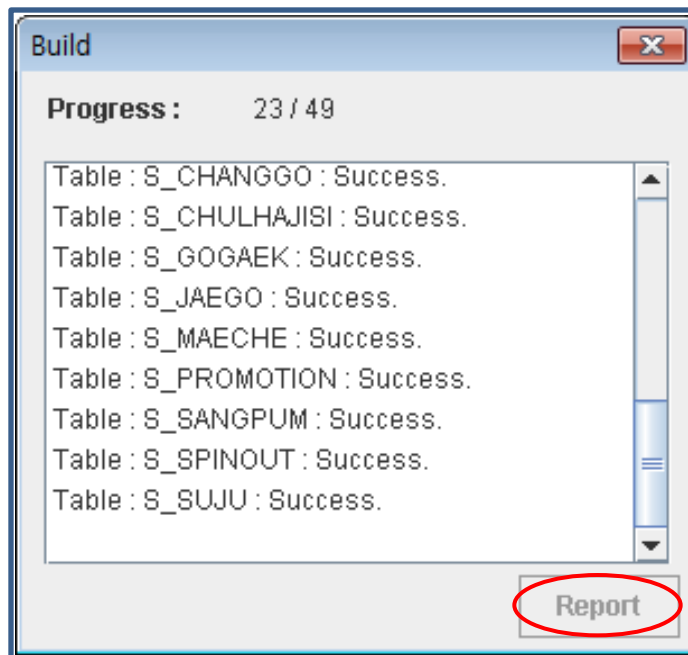


Migration Center

❖ Migration Center 실습

➤ 프로젝트 Build

- 신규 등록된 프로젝트를 Build 하고 Report 결과를 확인

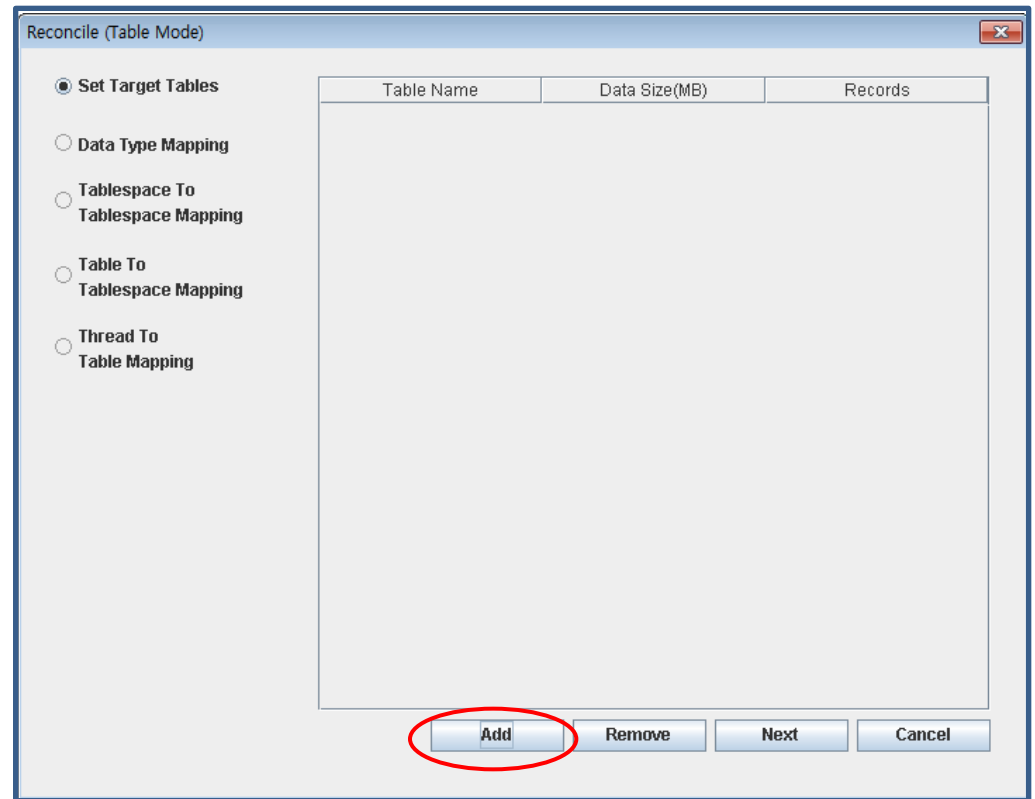
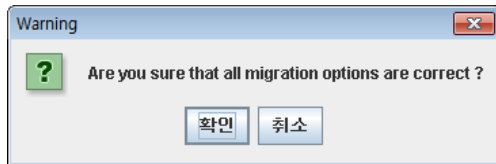


Migration Center

❖ Migration Center 실습

➤ 프로젝트 Reconcile

- Reconcile 메뉴에서 마이그레이션 대상을 지정

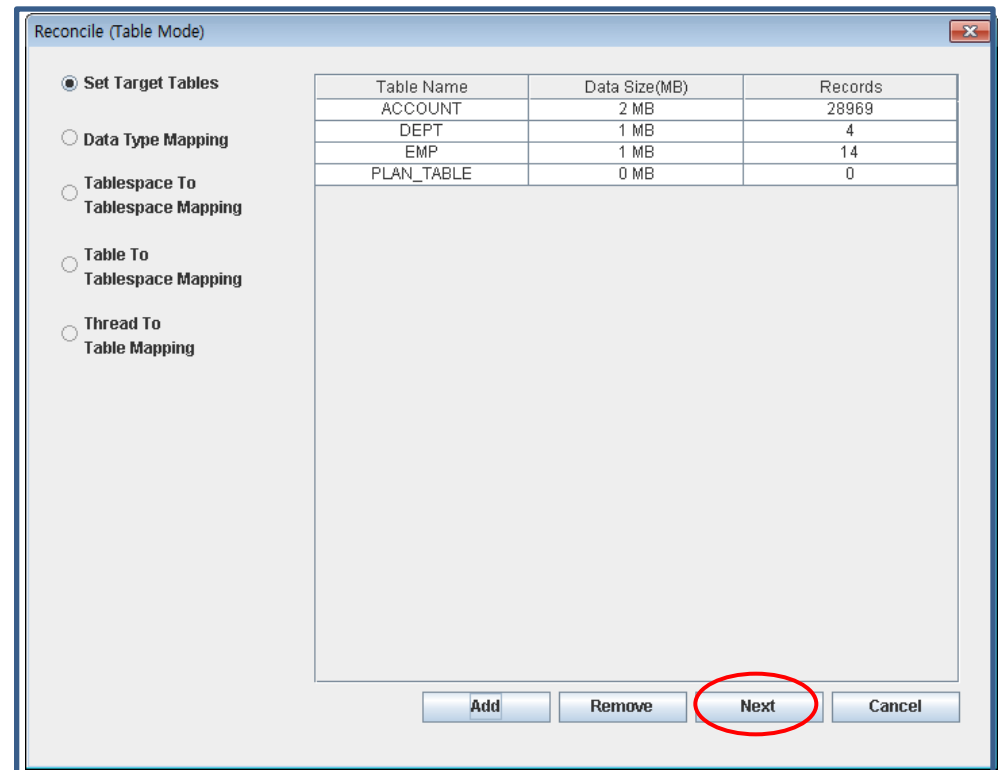
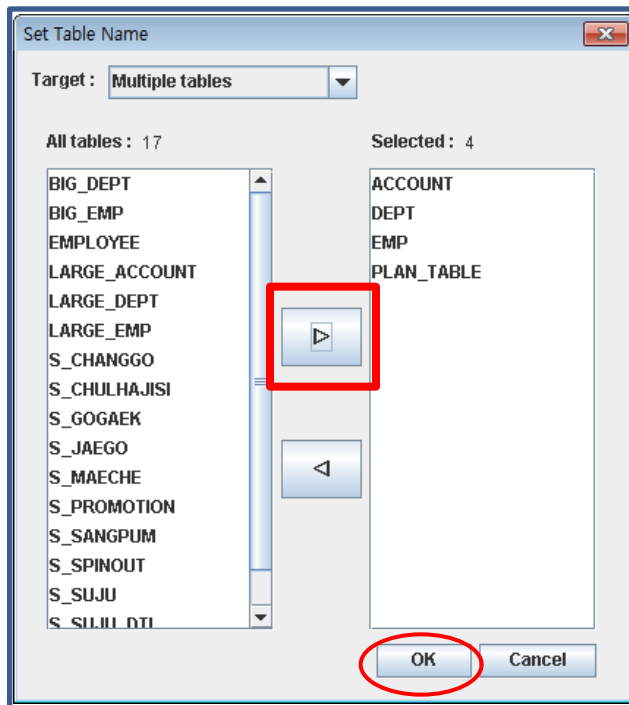


Migration Center

❖ Migration Center 실습

➤ 프로젝트 Reconcile

- 마이그레이션 대상 테이블을 지정후 “NEXT” 클릭

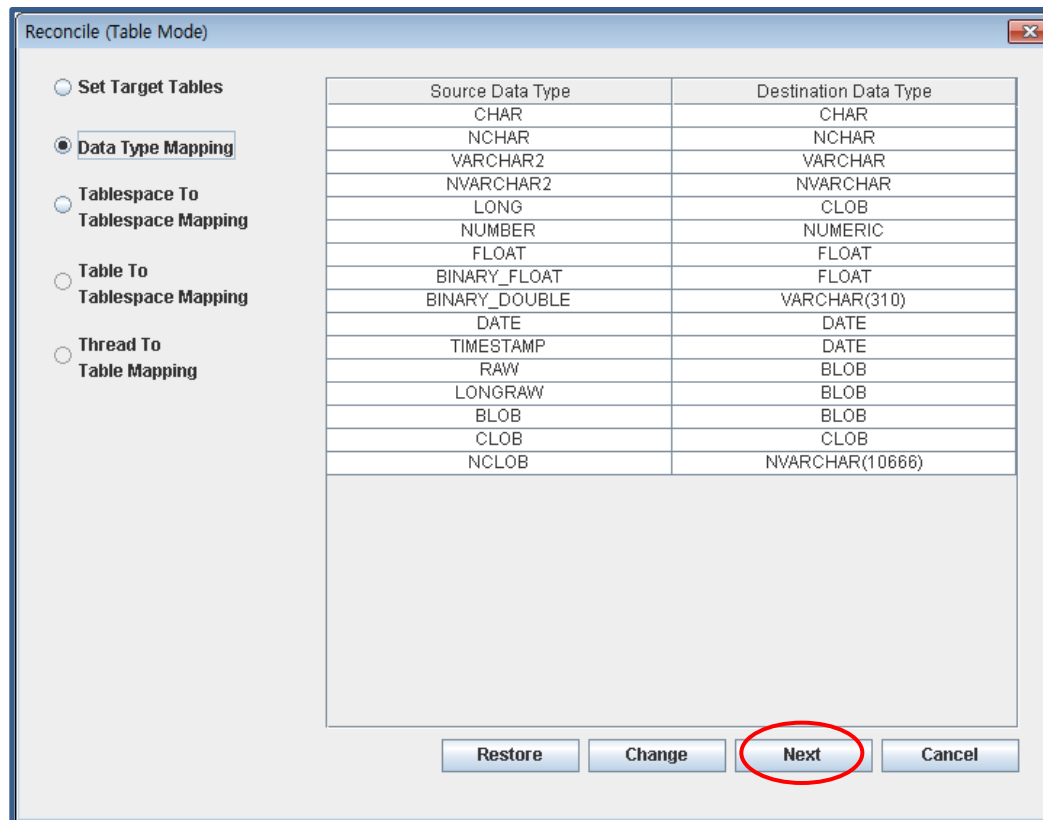


Migration Center

❖ Migration Center 실습

➤ 프로젝트 Reconcile

- 데이터 타입 Mapping 정보 확인후 “Next” 클릭

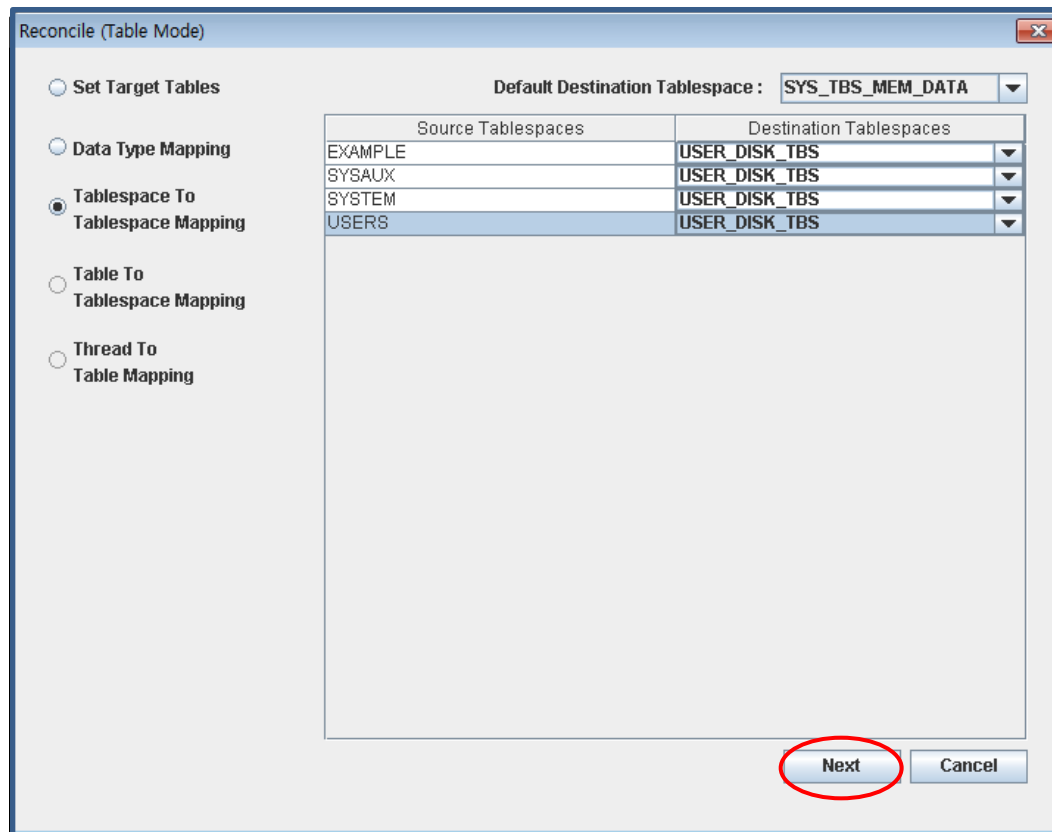


Migration Center

❖ Migration Center 실습

➤ 프로젝트 Reconcile

- 테이블스페이스 Mapping 정보 확인후 “Next” 클릭

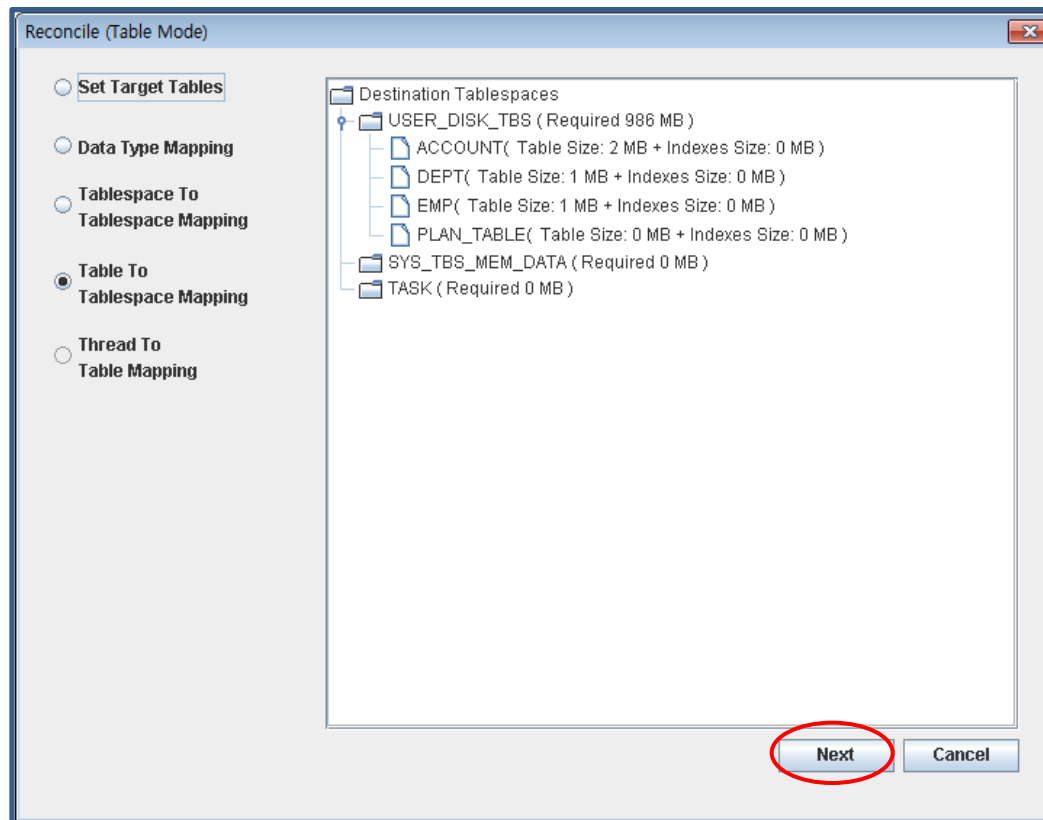


Migration Center

❖ Migration Center 실습

➤ 프로젝트 Reconcile

- 테이블, 테이블스페이스 Mapping 정보 확인후 “Next” 클릭

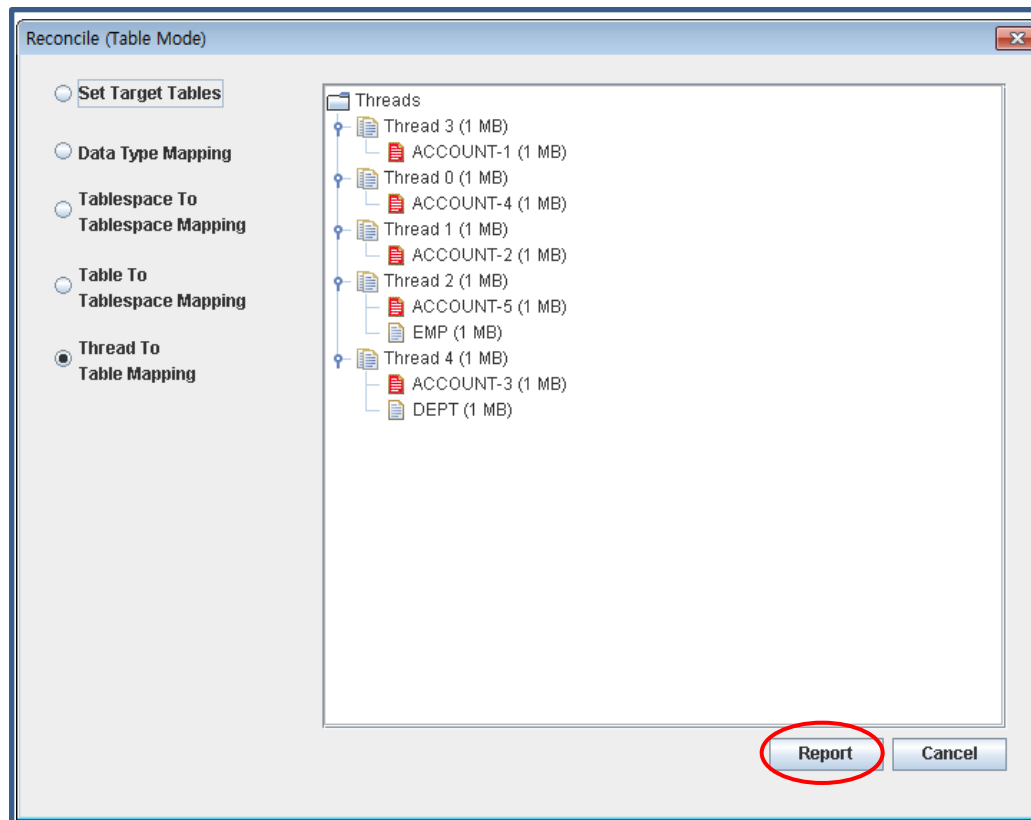


Migration Center

❖ Migration Center 실습

➤ 프로젝트 Reconcile

- Thread, 테이블 Mapping 정보 확인 후 “Report” 클릭. 최종정보 확인

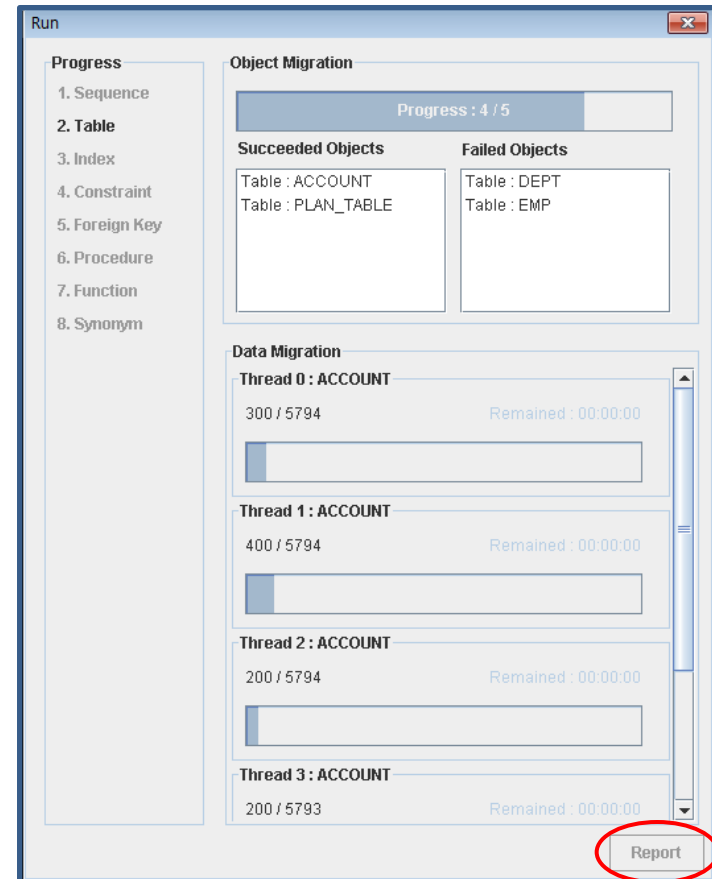
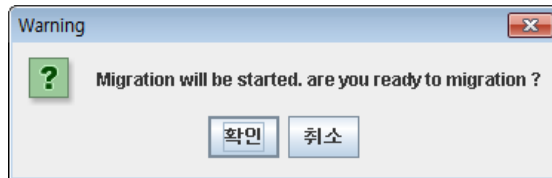


Migration Center

❖ Migration Center 실습

➤ 프로젝트 실행

- 프로젝트 실행후 결과 확인



Migration Center

❖ Migration Center 실습

➤ 마이그레이션 결과 조회

- isql 실행하여 테이블 조회 (본인이 마이그레이션 수행한 테이블 정보를 확인)

```
shell::~/home/alti1> is
iSQL> DESC account
iSQL> DESC dept
iSQL> DESC emp
iSQL> DESC plan_table

iSQL> SELECT * FROM account;
iSQL> SELECT * FROM dept;
iSQL> SELECT * FROM emp;
iSQL> SELECT * FROM plan_table;
```

Thank you!

Altibase Education Center

서울시 구로구 구로동 대륭포스트타워2차 9층 152-050

Tel : 02-2082-0439

Fax : 02-2082-1099

E-mail : education@altibase.com

Homepage : <http://edu.altibase.com>