

General Reference-2.The Data Dictionary

Altibase 7.1

Altibase® Administration



Altibase Administration General Reference

Release 7.1

Copyright © 2001~2023 Altibase Corp. All Rights Reserved.

본 문서의 저작권은 (주)알티베이스에 있습니다. 이 문서에 대하여 당사의 동의없이 무단으로 복제 또는 전용할 수 없습니다.

(주)알티베이스

08378 서울시 구로구 디지털로 306 대륭포스트타워 II 10층

전화 : 02-2082-1114

팩스 : 02-2082-1099

고객서비스포털 : <http://support.altibase.com>

홈페이지 : <http://www.altibase.com>

목차

- [서문](#)
 - [이 매뉴얼에 대하여](#)
- [1.데이터 디렉터리](#)
 - [메타 테이블](#)
 - [SYS AUDIT](#)
 - [SYS AUDIT_OPTS](#)
 - [SYS COLUMNS](#)
 - [SYS COMMENTS](#)
 - [SYS COMPRESSION TABLES](#)
 - [SYS CONSTRAINTS](#)
 - [SYS CONSTRAINT COLUMNS](#)
 - [SYS CONSTRAINT RELATED](#)
 - [SYS DATABASE](#)
 - [SYS DATABASE LINKS](#)
 - [SYS DIRECTORIES](#)
 - [SYS ENCRYPTED COLUMNS](#)
 - [SYS GRANT OBJECT](#)
 - [SYS GRANT SYSTEM](#)
 - [SYS INDEX COLUMNS](#)
 - [SYS INDEX PARTITIONS](#)
 - [SYS INDEX RELATED](#)
 - [SYS INDICES](#)
 - [SYS JOBS](#)
 - [SYS LIBRARIES](#)
 - [SYS LOBS](#)
 - [SYS MATERIALIZED VIEWS](#)
 - [SYS PACKAGES](#)
 - [SYS PACKAGE PARAS](#)
 - [SYS PACKAGE PARSE](#)
 - [SYS PACKAGE RELATED](#)
 - [SYS PART INDICES](#)
 - [SYS PART KEY COLUMNS](#)
 - [SYS PART LOBS](#)
 - [SYS PART TABLES](#)
 - [SYS PASSWORD HISTORY](#)
 - [SYS PASSWORD LIMITS](#)

- [SYS PRIVILEGES](#)
- [SYS PROCEDURES](#)
- [SYS PROC PARAS](#)
- [SYS PROC PARSE](#)
- [SYS PROC RELATED](#)
- [SYS RECYCLEBIN](#)
- [SYS REPLICATIONS](#)
- [SYS REPL HOSTS](#)
- [SYS REPL ITEMS](#)
- [SYS REPL OFFLINE DIR](#)
- [SYS REPL OLD CHECKS](#)
- [SYS REPL OLD CHECK COLUMNS](#)
- [SYS REPL OLD COLUMNS](#)
- [SYS REPL OLD INDEX COLUMNS](#)
- [SYS REPL OLD INDICES](#)
- [SYS REPL OLD ITEMS](#)
- [SYS REPL TABLE OID IN USE](#)
- [SYS REPL RECOVERY INFOS](#)
- [SYS SECURITY](#)
- [SYS SYNONYMS](#)
- [SYS TABLES](#)
- [SYS TABLE PARTITIONS](#)
- [SYS TABLE SIZE](#)
- [SYS TBS USERS](#)
- [SYS TRIGGERS](#)
- [SYS TRIGGER DML TABLES](#)
- [SYS TRIGGER STRINGS](#)
- [SYS TRIGGER UPDATE COLUMNS](#)
- [SYS USERS](#)
- [DBA USERS](#)
- [SYS USER ROLES](#)
- [SYS VIEWS](#)
- [SYS VIEW PARSE](#)
- [SYS VIEW RELATED](#)
- [SYS XA HEURISTIC TRANS](#)
- [SYS GEOMETRIES](#)
- [SYS GEOMETRY COLUMNS](#)
- [USER SRS](#)

- [성능 뷰](#)
- [V\\$ACCESS LIST](#)
- [V\\$ALLCOLUMN](#)
- [V\\$ARCHIVE](#)
- [V\\$BACKUP INFO](#)
- [V\\$BUFFPAGEINFO](#)
- [V\\$BUFFPOOL STAT](#)
- [V\\$CATALOG](#)
- [V\\$DATABASE](#)
- [V\\$DATAFILES](#)
- [V\\$DATATYPE](#)
- [V\\$DBA 2PC PENDING](#)
- [V\\$DBLINK ALILINKER STATUS](#)
- [V\\$DBLINK DATABASE LINK INFO](#)
- [V\\$DBLINK GLOBAL TRANSACTION INFO](#)
- [V\\$DBLINK LINKER CONTROL SESSION INFO](#)
- [V\\$DBLINK LINKER DATA SESSION INFO](#)
- [V\\$DBLINK LINKER SESSION INFO](#)
- [V\\$DBLINK NOTIFIER TRANSACTION INFO](#)
- [V\\$DBLINK REMOTE STATEMENT INFO](#)
- [V\\$DBLINK REMOTE TRANSACTION INFO](#)
- [V\\$DBMS STATS](#)
- [V\\$DB FREEPAGELISTS](#)
- [V\\$DB PROTOCOL](#)
- [V\\$DIRECT PATH INSERT](#)
- [V\\$DISKTBL INFO](#)
- [V\\$DISK BTREE HEADER](#)
- [V\\$DISK TEMP INFO](#)
- [V\\$DISK TEMP STAT](#)
- [V\\$DISK UNDO USAGE](#)
- [V\\$DR CONNECTION INFO](#)
- [V\\$DR GAP](#)
- [V\\$DR SERVERS](#)
- [V\\$DR STATUS](#)
- [V\\$EVENT NAME](#)
- [V\\$EXTPROC AGENT](#)
- [V\\$FILESTAT](#)
- [V\\$FLUSHER](#)

- [V\\$FLUSHINFO](#)
- [V\\$INDEX](#)
- [V\\$INSTANCE](#)
- [V\\$INTERNAL_SESSION](#)
- [V\\$LATCH](#)
- [V\\$LIBRARY](#)
- [V\\$LFG](#)
- [V\\$LOCK](#)
- [V\\$LOCK STATEMENT](#)
- [V\\$LOG](#)
- [V\\$LOCK WAIT](#)
- [V\\$MEMGC](#)
- [V\\$MEMSTAT](#)
- [V\\$MEMTBL INFO](#)
- [V\\$MEM BTREE HEADER](#)
- [V\\$MEM BTREE NODEPOOL](#)
- [V\\$MEM RTREE HEADER](#)
- [V\\$MEM RTREE NODEPOOL](#)
- [V\\$MEM TABLESPACES](#)
- [V\\$MEM TABLESPACE CHECKPOINT PATHS](#)
- [V\\$MEM TABLESPACE STATUS DESC](#)
- [V\\$MUTEX](#)
- [V\\$NLS PARAMETERS](#)
- [V\\$NLS TERRITORY](#)
- [V\\$OBSOLETE BACKUP INFO](#)
- [V\\$PKGTEXT](#)
- [V\\$PLANTEXT](#)
- [V\\$PROCINFO](#)
- [V\\$PROCTEXT](#)
- [V\\$PROPERTY](#)
- [V\\$QUEUE DELETE OFF](#)
- [V\\$REPEXEC](#)
- [V\\$REPGAP](#)
- [V\\$REPGAP PARALLEL](#)
- [V\\$REPLOGBUFFER](#)
- [V\\$REPOFFLINE STATUS](#)
- [V\\$REPRECEIVER](#)
- [V\\$REPRECEIVER COLUMN](#)

- [V\\$REPRECEIVER PARALLEL](#)
- [V\\$REPRECEIVER PARALLEL APPLY](#)
- [V\\$REPRECEIVER STATISTICS](#)
- [V\\$REPRECEIVER TRANSTBL](#)
- [V\\$REPRECEIVER TRANSTBL PARALLEL](#)
- [V\\$REPRECOVERY](#)
- [V\\$REPSENDER](#)
- [V\\$REPSENDER PARALLEL](#)
- [V\\$REPSENDER SENT LOG COUNT](#)
- [V\\$REPSENDER SENT LOG COUNT PARALLEL](#)
- [V\\$REPSENDER STATISTICS](#)
- [V\\$REPSENDER TRANSTBL](#)
- [V\\$REPSENDER TRANSTBL PARALLEL](#)
- [V\\$REPSYNC](#)
- [V\\$REPL REMOTE META REPLICATIONS](#)
- [V\\$REPL REMOTE META ITEMS](#)
- [V\\$REPL REMOTE META COLUMNS](#)
- [V\\$REPL REMOTE META INDEX COLUMNS](#)
- [V\\$REPL REMOTE META INDICES](#)
- [V\\$REPL REMOTE META CHECKS](#)
- [V\\$RESERVED WORDS](#)
- [V\\$SBUFFER STAT](#)
- [V\\$SEGMENT](#)
- [V\\$SEQ](#)
- [V\\$SERVICE THREAD](#)
- [V\\$SERVICE THREAD MGR](#)
- [V\\$SESSION](#)
- [V\\$SESSION EVENT](#)
- [V\\$SESSION WAIT](#)
- [V\\$SESSION WAIT CLASS](#)
- [V\\$SESSIONMGR](#)
- [V\\$SESSTAT](#)
- [V\\$SFLUSHER](#)
- [V\\$SFLUSHINFO](#)
- [V\\$SNAPSHOT](#)
- [V\\$SQLTEXT](#)
- [V\\$SQL PLAN CACHE](#)
- [V\\$SQL PLAN CACHE PCO](#)

- [V\\$SQL PLAN CACHE SQLTEXT](#)
- [V\\$STABLE MEM DATAFILES](#)
- [V\\$STATEMENT](#)
- [V\\$STATNAME](#)
- [V\\$SYSSTAT](#)
- [V\\$SYSTEM CONFLICT PAGE](#)
- [V\\$SYSTEM EVENT](#)
- [V\\$SYSTEM WAIT CLASS](#)
- [V\\$TABLE](#)
- [V\\$TABLESPACES](#)
- [V\\$TIME ZONE NAMES](#)
- [V\\$TRACELOG](#)
- [V\\$TRANSACTION](#)
- [V\\$TRANSACTION MGR](#)
- [V\\$TSSEGS](#)
- [V\\$TXSEGS](#)
- [V\\$UDSEGS](#)
- [V\\$UNDO BUFF STAT](#)
- [V\\$USAGE](#)
- [V\\$VERSION](#)
- [V\\$VOL TABLESPACES](#)
- [V\\$WAIT CLASS NAME](#)
- [V\\$XID](#)
- [2.샘플 스키마](#)
 - [예제 테이블 정보](#)
 - [E-R 다이어그램과 샘플 데이터](#)

서문

이 매뉴얼에 대하여

이 매뉴얼은 Altibase의 기능, 제품 구성요소, 그리고 사용법에 대해 설명한다.

대상 사용자

이 매뉴얼은 다음과 같은 Altibase 사용자를 대상으로 작성되었다.

- 데이터베이스 관리자
- 성능 관리자
- 데이터베이스 사용자
- 응용 프로그램 개발자
- 기술지원부

다음과 같은 배경 지식을 가지고 이 매뉴얼을 읽는 것이 좋다.

- 컴퓨터, 운영 체제 및 운영 체제 유틸리티 운용에 필요한 기본 지식
- 관계형 데이터베이스 사용 경험 또는 데이터베이스 개념에 대한 이해
- 컴퓨터 프로그래밍 경험
- 데이터베이스 서버 관리, 운영 체제 관리 또는 네트워크 관리 경험

소프트웨어 환경

이 매뉴얼은 데이터베이스 서버로 Altibase 버전 7.1을 사용한다는 가정 하에 작성되었다.

이 매뉴얼의 구성

이 매뉴얼은 다음과 같이 구성되어 있다.

- 제 1장 데이터 디렉터리
이 장은 Altibase 데이터 디렉터리에 대해 설명한다. Altibase의 데이터 디렉터리는 데이터베이스 객체 정보를 저장하는 메타 테이블과 시스템 프로세스 정보를 저장하는 프로세스 테이블로 나뉘어 있다.
- 제 2장 샘플 스키마
이 장은 샘플로 제공되는 테이블 정보와 ER 다이어그램을 제공한다.

문서화 규칙





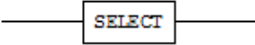
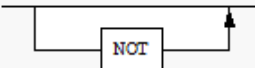
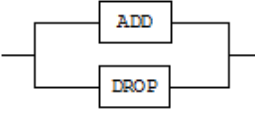
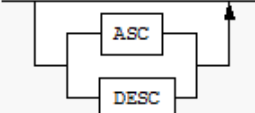
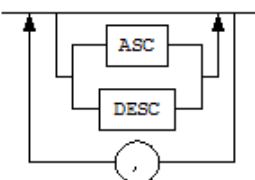
이 절에서는 이 매뉴얼에서 사용하는 규칙에 대해 설명한다. 이 규칙을 이해하면 이 매뉴얼과 설명서 세트의 다른 매뉴얼에서 정보를 쉽게 찾을 수 있다.

여기서 설명하는 규칙은 다음과 같다.

- 구문 다이어그램
- 샘플 코드 규칙

구문 다이어그램

이 매뉴얼에서는 다음 구성 요소로 구축된 다이어그램을 사용하여, 명령문의 구문을 설명한다.

| 구성 요소 | 의미 |
|---|---|
|  | 명령문이 시작한다. 완전한 명령문이 아닌 구문 요소는 화살표로 시작한다. |
|  | 명령문이 다음 라인에 계속된다. 완전한 명령문이 아닌 구문 요소는 이 기호로 종료한다. |
|  | 명령문이 이전 라인으로부터 계속된다. 완전한 명령문이 아닌 구문 요소는 이 기호로 시작한다. |
|  | 명령문이 종료한다. |
|  | 필수 항목 |
|  | 선택적 항목 |
|  | 선택사항이 있는 필수 항목. 한 항목만 제공해야 한다. |
|  | 선택사항이 있는 선택적 항목. |
|  | 선택적 항목. 여러 항목이 허용된다. 각 반복 앞부분에 콤마가 와야 한다. |

샘플 코드 규칙

코드 예제는 SQL, Stored Procedure, iSQL, 또는 다른 명령 라인 구문들을 예를 들어 설명한다.

아래 테이블은 코드 예제에서 사용된 인쇄 규칙에 대해 설명한다.

| 규칙 | 의미 | 예제 |
|-----|---------------------------------|---|
| [] | 선택 항목을 표시 | VARCHAR [(size)][[FIXED \]] VARIABLE] |
| { } | 필수 항목 표시. 반드시 하나 이상을 선택해야 되는 표시 | { ENABLE DISABLE COMPILE } |
| | 선택 또는 필수 항목 표시의 인자 구분 표시 | { ENABLE DISABLE COMPILE } [ENABLE DISABLE COMPILE] |

| 규칙 | 의미 | 예제 |
|-----------------|---|---|
| ... | 그 이전 인자의 반복 표시 예제 코드들의 생략되는 것을 표시 | <pre>SQL> SELECT ename FROM employee; ENAME ----- SWNO HJNO HSCHOI . . . 20 rows selected.</pre> |
| 그 밖 에 기 호 | 위에서 보여진 기호 이 외에 기호들 | EXEC :p1 := 1; acc NUMBER(11,2); |
| 기울임 꼴 | 구문 요소에서 사용자가 지정해야 하는 변수, 특수한 값을 제공해야만 하는 위치 | SELECT * FROM <i>table_name</i> ; CONNECT <i>userID/password</i> ; |
| 소문자 | 사용자가 제공하는 프로그램의 요소들, 예를 들어 테이블 이름, 칼럼 이름, 파일 이름 등 | SELECT ename FROM employee; |
| 대문자 | 시스템에서 제공하는 요소들 또는 구문에 나타나는 키워드 | DESC SYSTEM.SYS_INDICES; |

관련 자료

자세한 정보를 위하여 다음 문서 목록을 참조한다.

- Installation Guide
- Getting Started Guide
- Administrator's Manual
- Replication Manual

Altibase는 여러분의 의견을 환영합니다.

이 매뉴얼에 대한 여러분의 의견을 보내주시기 바랍니다. 사용자의 의견은 다음 버전의 매뉴얼을 작성하는데 많은 도움이 됩니다. 보내실 때에는 아래 내용과 함께 고객센터포털(<http://support.altibase.com/kr/>)로 보내주시기 바랍니다.

- 사용 중인 매뉴얼의 이름과 버전
- 매뉴얼에 대한 의견
- 사용자의 성함, 주소, 전화번호

이 외에도 Altibase 기술지원 설명서의 오류와 누락된 부분 및 기타 기술적인 문제들에 대해서 이 주소로 보내주시면 정성껏 처리하겠습니다. 또한, 기술적인 부분과 관련하여 즉각적인 도움이 필요한 경우에도 고객센터포털을 통해 서비스를 요청하시기 바랍니다.

여러분의 의견에 항상 감사드립니다.

1.데이터 디렉터리

Altibase의 데이터 디렉터리는 데이터베이스 객체 정보를 저장하는 메타 테이블과 시스템 프로세스 정보를 저장하는 프로세스 테이블로 나뉘어진다. 프로세스 테이블은 다시 고정 테이블 (Fixed Table)과 성능 뷰 (Performance View)로 나뉘어진다.

본 장은 데이터베이스 객체 및 Altibase 시스템 정보를 제공하는 데이터 디렉터리에 대해 설명한다.

메타 테이블

메타 테이블이란 데이터베이스에 생성된 객체에 대한 모든 정보를 저장하고 있는 시스템 정의 테이블이다.

이 절에서는 메타 테이블의 종류 및 그 구조, 그리고 메타 테이블의 조회 및 변경에 대하여 설명한다.

구조 및 기능

메타 테이블은 데이터베이스 객체를 관리하기 위해 시스템에 의해 정의된 테이블이다. 메타 테이블의 데이터 타입 및 레코드 저장 형태는 사용자가 생성하는 일반 테이블과 동일하다.

Altibase는 구동시 데이터베이스 객체 정보를 로딩하고, DDL 문을 수행할 때 데이터베이스 객체 정보를 조회, 저장 및 변경하기 위해 메타 테이블을 사용한다.

메타 테이블의 소유자는 시스템 사용자 (SYSTEM_)로 일반 사용자는 메타 테이블에 대한 접근이 제한된다.

메타 테이블 조회

DDL 문으로 데이터베이스 객체를 생성, 삭제 및 변경 시 메타 테이블의 레코드가 시스템에 의해 생성, 삭제 또는 변경된다.

DDL 문 수행 후, 변경된 데이터베이스 객체 정보는 메타 테이블을 조회함으로써 확인할 수 있다. 메타 테이블의 레코드는 일반 테이블과 같이 SELECT 문으로 조회가 가능하다.

메타 테이블 데이터 변경

사용자는 시스템에서 정의된 시스템 사용자(SYSTEM_) 계정으로 DML문을 사용하여 메타 테이블의 데이터를 명시적으로 변경할 수 있다. 그러나 메타 테이블 정보가 변경되면 시스템 구동이 실패하거나, 데이터베이스 객체 정보를 상실하여 시스템에 치명적인 손상이 발생할 수 있다. 따라서 가급적 메타 테이블 데이터에 대한 사용자의 명시적인 변경은 피해야 한다. 불가피한 사정으로 메타 테이블 데이터 변경 시에는 변경 전에 반드시 데이터베이스 백업을 해야 하며, 사용자의 명시적인 메타 테이블 데이터 변경으로 인해 발생하는 데이터베이스의 손상은 전적으로 사용자 책임이다.

메타 테이블 스키마 변경

새로운 종류의 DDL문이 제공되거나 기존 구문의 기능 변경 시 메타 테이블 스키마가 변경될 수 있다. 메타 테이블 스키마의 변경 특성에 따라 데이터베이스 마이그레이션이 필요한 경우와 Altibase 구동 시 자동으로 메타 테이블 스키마를 변경하는 두 가지 경우로 구분된다.

Altibase 하위 버전에서 상위 버전으로 업그레이드 시 이를 고려해야 한다.

메타 테이블 종류

다음 표는 메타 테이블의 목록이다. 메타 테이블의 이름은 SYS_로 시작한다.

| 메타 테이블 이름 | 설명 |
|-------------------------|---|
| SYS_AUDIT_ | 감사의 동작 상태가 저장되는 메타 테이블 |
| SYS_AUDIT_OPTS_ | 감사 조건이 저장되는 메타 뷰. SYS_AUDIT_ALL_OPTS_가 이 뷰의 베이스 메타 테이블이다. |
| SYS_COLUMNS_ | 칼럼에 대한 정보를 저장하는 메타 테이블 |
| SYS_COMMENTS_ | 설명을 달기 위한 주석 메타 테이블 |
| SYS_COMPRESSION_TABLES_ | 압축 칼럼에 대한 정보가 저장되는 메타 테이블 |
| SYS_CONSTRAINTS_ | 제약 조건에 대한 정보를 저장하는 메타 테이블 |
| SYS_CONSTRAINT_COLUMNS_ | 제약 조건을 가지는 칼럼에 대한 정보를 저장하는 메타 테이블 |
| SYS_CONSTRAINT_RELATED_ | 제약조건(constraints)이 참조하는 저장 함수에 대한 정보를 저장하는 메타 테이블 |
| SYS_DATABASE_ | 데이터베이스 이름과 버전에 대한 정보를 저장하는 메타 테이블 |
| SYS_DATABASE_LINKS_ | 데이터베이스 링크에 대한 정보를 저장하는 메타 테이블 |
| SYS_DIRECTORIES_ | 저장프로시저 내 파일 제어용 디렉터리에 대한 정보를 저장하는 메타 테이블 |
| SYS_DN_USERS_ | 향후 확장 예정 |
| SYS_DUMMY_ | 내부 용도 |
| SYS_ENCRYPTED_COLUMNS_ | 보안 설정에 기반한 부가적인 보안 정보를 암호화된 칼럼 별로 저장하는 메타 테이블 |
| SYS_GRANT_OBJECT_ | 객체 권한에 대한 정보를 저장하는 메타 테이블 |
| SYS_GRANT_SYSTEM_ | 시스템 권한에 대한 정보를 저장하는 메타 테이블 |
| SYS_INDEX_COLUMNS_ | 인덱스 키 칼럼에 대한 정보를 저장하는 메타 테이블 |
| SYS_INDEX_PARTITIONS_ | 인덱스 파티션에 대한 정보를 저장하는 메타 테이블 |
| SYS_INDEX_RELATED_ | 함수 기반 인덱스가 기반하는 저장 함수에 대한 정보를 저장하는 메타 테이블 |
| SYS_INDICES_ | 인덱스에 대한 정보를 저장하는 메타 테이블 |
| SYS_JOBS_ | JOB에 대한 정보를 저장하는 메타 테이블 |
| SYS_LIBRARIES_ | 외부 라이브러리 객체에 대한 정보를 저장하는 메타 테이블 |

| 메타 테이블 이름 | 설명 |
|-------------------------|--|
| SYS_LOBS_ | LOB 칼럼에 대한 정보를 저장하는 메타 테이블 |
| SYS_MATERIALIZED_VIEWS_ | Materialized view에 대한 정보가 기록되어 있는 메타 테이블 |
| SYS_PACKAGES_ | 패키지에 대한 정보가 저장되는 메타 테이블 |
| SYS_PACKAGE_PARAS_ | 패키지에 포함된 서브프로그램(저장 프로시저와 저장 함수)들의 인자 (parameter)들에 대한 정보가 저장되는 메타 테이블 |
| SYS_PACKAGE_PARSE_ | 사용자가 정의한 패키지의 구문 텍스트가 저장되는 메타 테이블 |
| SYS_PACKAGE_RELATED_ | 패키지 내에 포함된 저장 프로시저와 저장 함수들이 참조하는 테이블, 시퀀스, 저장 프로시저, 저장 함수, 또는 뷰들에 대한 정보가 저장되는 메타 테이블 |
| SYS_PART_INDICES_ | 파티션드 인덱스에 대한 정보를 저장하는 메타 테이블 |
| SYS_PART_KEY_COLUMNS_ | 파티셔닝 키에 대한 정보를 저장하는 메타 테이블 |
| SYS_PART_LOBS_ | 파티션별 LOB 칼럼에 대한 정보를 저장하는 메타 테이블 |
| SYS_PART_TABLES_ | 파티션드 테이블에 대한 정보를 저장하는 메타 테이블 |
| SYS_PASSWORD_HISTORY_ | 패스워드 관리 정책을 설정한 사용자의 패스워드 변경 내역을 저장하는 메타 테이블 |
| SYS_PASSWORD_LIMITS_ | 사용자 생성 시 계정에 대해 지정한 패스워드 관리 정책과 계정의 현재 상태를 저장하는 메타 뷰 |
| SYS_PRIVILEGES_ | 권한에 대한 정보를 저장하는 메타 테이블 |
| SYS_PROCEDURES_ | 저장 프로시저 및 함수에 대한 정보를 저장하는 메타 테이블 |
| SYS_PROC_PARAS_ | 저장 프로시저 및 함수의 파라미터에 대한 정보를 저장하는 메타 테이블 |
| SYS_PROC_PARSE_ | 저장 프로시저 및 함수의 구문에 대한 정보를 저장하는 메타 테이블 |
| SYS_PROC_RELATED_ | 저장 프로시저 및 함수가 접근하는 테이블에 대한 정보를 저장하는 메타 테이블 |
| SYS_RECYCLEBIN_ | 휴지통에 있는 테이블의 정보를 저장하는 메타 테이블 |
| SYS_REPLICATIONS_ | 이중화에 대한 정보를 저장하는 메타 테이블 |
| SYS_REPL_HOSTS_ | 이중화 호스트에 대한 정보를 저장하는 메타 테이블 |
| SYS_REPL_ITEMS_ | 이중화 테이블에 대한 정보를 저장하는 메타 테이블 |
| SYS_REPL_OFFLINE_DIR_ | 이중화 오프라인 옵션 관련 로그 디렉터리에 대한 정보를 저장하는 메타 테이블 |

| 메타 테이블 이름 | 설명 |
|-----------------------------|--|
| SYS_REPL_OLD_CHECKS_ | 이중화 송신 쓰레드가 복제중인 이중화 대상 칼럼 중 CHECK 제약조건에 대한 정보를 가진 메타 테이블 |
| SYS_REPL_OLD_CHECK_COLUMNS_ | 이중화 송신 쓰레드가 복제 중인 이중화 대상 칼럼에 설정된 CHECK 제약조건에 대한 정보를 가진 메타 테이블 |
| SYS_REPL_OLD_COLUMNS_ | 이중화 송신 쓰레드가 이중화하는 칼럼에 대한 정보를 저장하는 메타 테이블 |
| SYS_REPL_OLD_INDEX_COLUMNS_ | 이중화 송신 쓰레드가 이중화하는 인덱스 칼럼에 대한 정보를 저장하는 메타 테이블 |
| SYS_REPL_OLD_INDICES_ | 이중화 송신 쓰레드가 이중화하는 인덱스에 대한 정보를 저장하는 메타 테이블 |
| SYS_REPL_OLD_ITEMS_ | 이중화 송신 쓰레드가 이중화하는 테이블에 대한 정보를 저장하는 메타 테이블 |
| SYS_REPL_TABLE_OID_IN_USE_ | 이중화가 아직 처리하지 않은 DDL 로그에 포함된 테이블의 테이블 객체 식별자(TABLE OID) 정보를 관리하는 메타 테이블 |
| SYS_REPL_RECOVERY_INFOS_ | 원격 서버의 복구를 위한 로그 정보를 저장하는 메타 테이블 |
| SYS_SECURITY_ | 보안 모듈에 대한 정보를 저장하는 메타 테이블 |
| SYS_SYNONYMS_ | 시노ним에 대한 정보를 저장하는 메타 테이블 |
| SYS_TABLES_ | 테이블에 대한 정보를 저장하는 메타 테이블 |
| SYS_TABLE_PARTITIONS_ | 테이블의 파티션에 대한 정보를 저장하는 메타 테이블 |
| SYS_TABLE_SIZE_ | 시스템에 있는 디스크 테이블과 메모리 테이블의 실제 크기 정보를 저장하는 메타 테이블 |
| SYS_TBS_USERS_ | 사용자 정의 테이블스페이스에 대한 사용자 접근 정보를 저장하는 메타 테이블 |
| SYS_TRIGGERS_ | 트리거에 대한 정보를 저장하는 메타 테이블 |
| SYS_TRIGGER_DML_TABLES_ | 트리거가 접근하는 테이블에 대한 정보를 저장하는 메타 테이블 |
| SYS_TRIGGER_STRINGS_ | 트리거 구문을 저장하는 메타 테이블 |
| SYS_TRIGGER_UPDATE_COLUMNS_ | 그 값이 변경될 때마다 트리거를 시작시키는 칼럼들에 대한 정보를 저장하는 메타 테이블 |
| SYS_USERS_ | 사용자에 대한 정보를 저장하는 메타 테이블 |
| DBA_USERS_ | 사용자에 대한 정보를 저장하는 메타 테이블. SYS 사용자만 조회 가능. |

| 메타 테이블 이름 | 설명 |
|-------------------------|---|
| SYS_USER_ROLES_ | 사용자에게 부여된 롤(Role)에 대한 정보를 저장하는 메타 테이블 |
| SYS_VIEWS_ | 뷰에 대한 정보를 저장하는 메타 테이블 |
| SYS_VIEW_PARSE_ | 뷰 구문을 저장하는 메타 테이블 |
| SYS_VIEW_RELATED_ | 뷰가 접근하는 테이블에 대한 정보를 저장하는 메타 테이블 |
| SYS_XA_HEURISTIC_TRANS_ | 글로벌 (global) 트랜잭션에 대한 정보를 저장하는 메타 테이블 |
| SYS_GEOMETRIES_ | GEOMETRY 칼럼을 보유한 테이블의 정보를 저장하는 메타 테이블 |
| SYS_GEOMETRY_COLUMNS_ | GEOMETRY 칼럼에 대한 정보를 저장하는 메타 테이블; Synonym으로 GEOMETRY_COLUMNS가 있음 |
| USER_SRS_ | 공간 참조 시스템(SRS, Spatial Reference System)에 관한 정보를 저장하는 메타 테이블, Synonym으로 SPATIAL_REF_SYS가 있음 |

SYS_AUDIT_

감사(Auditing)의 동작 상태가 기록되는 메타 테이블이다.

| Column name | Type | Description |
|-------------|---------|-------------------|
| IS_STARTED | INTEGER | 감사가 실행 중인지 여부 |
| START_TIME | DATE | 감사 시작 일시 |
| STOP_TIME | DATE | 감사 종료 일시 |
| RELOAD_TIME | DATE | 감사 조건이 서버에 적용된 일시 |

칼럼 정보

IS_STARTED

현재 감사가 실행 중인지를 나타낸다.

0: 현재 감사가 실행 중이 아님

1: 현재 감사가 실행 중임

START_TIME

감사가 시작된 일시를 나타낸다.

STOP_TIME

감사가 종료된 일시를 나타낸다.

RELOAD_TIME

변경된 감사 조건을 Altibase 서버에 적용한 시각을 나타낸다. 아래의 경우에 이 칼럼의 값이 업데이트 된다.

- 데이터베이스 관리자가 ALTER SYSTEM START AUDIT문을 사용해서 감사를 시작한 경우
- 데이터베이스 관리자가 ALTER SYSTEM RELOAD AUDIT문을 사용해서 변경된 감사 조건이 감사 수행에 적용되도록 한 경우

SYS_AUDIT_OPTS_

감사 조건이 저장되어 있는 뷰이다. 이 뷰의 베이스 테이블은 SYS_AUDIT_ALL_OPTS_ 메타 테이블이다.

| Column name | Type | Description |
|------------------|--------------|-----------------------------|
| USER_NAME | VARCHAR(128) | 사용자 이름 |
| OBJECT_NAME | VARCHAR(128) | 객체 이름 |
| OBJECT_TYPE | VARCHAR(40) | 객체 타입 |
| SELECT_OP | CHAR(3) | 각 작업 구문에 대한 로그 기록 단위를 나타낸다. |
| INSERT_OP | CHAR(3) | |
| UPDATE_OP | CHAR(3) | |
| DELETE_OP | CHAR(3) | |
| MOVE_OP | CHAR(3) | |
| MERGE_OP | CHAR(3) | |
| ENQUEUE_OP | CHAR(3) | |
| DEQUEUE_OP | CHAR(3) | |
| LOCK_TABLE_OP | CHAR(3) | |
| EXECUTE_OP | CHAR(3) | |
| COMMIT_OP | CHAR(3) | |
| ROLLBACK_OP | CHAR(3) | |
| SAVEPOINT_OP | CHAR(3) | |
| CONNECT_OP | CHAR(3) | |
| DISCONNECT_OP | CHAR(3) | |
| ALTER_SESSION_OP | CHAR(3) | |
| ALTER_SYSTEM_OP | CHAR(3) | |
| DDL_OP | CHAR(3) | |

칼럼 정보

USER_NAME

감사 대상 객체의 소유자의 사용자 이름이다.

OBJECT_NAME

감사 대상 객체의 이름을 나타낸다.

OBJECT_TYPE

대상 객체의 타입을 나타낸다. 아래 타입들 중 하나일 것이다.

- TABLE
- VIEW
- QUEUE
- SEQUENCE
- PROCEDURE
- FUNCTION

XXX_OP

각 작업 구문에 대한 로그를 기록하는 단위를 나타낸다. '/' 앞 쪽은 수행 성공에 대한 로그 기록 단위이고, 뒤 쪽은 수행 실패에 대한 로그 기록 단위이다.

표시되는 로그가 기록되는 단위는 아래와 같다.

- -: 로그가 기록되지 않음
- S: 세션 단위 로그가 기록됨
- A: 액세스 단위 로그가 기록됨
- T: 세션 / 액세스 단위에 상관없이 로그가 기록됨

아래는 감사 조건 설정 후의 SYS_AUDIT_OPTS_ 뷰의 값을 보여준다.

```
isQL> AUDIT insert, select, update, delete on friends BY SESSION WHENEVER
SUCCESSFUL;
Audit success.

isQL> AUDIT insert, select, update, delete on friends BY ACCESS WHENEVER NOT
SUCCESSFUL;
Audit success.

USER_NAME : SYS
OBJECT_NAME : FRIENDS
OBJECT_TYPE : TABLE
SELECT_OP : S/A
INSERT_OP : S/A
UPDATE_OP : S/A
DELETE_OP : S/A
MOVE_OP : -/-
MERGE_OP : -/-
ENQUEUE_OP : -/-
DEQUEUE_OP : -/-
LOCK_TABLE_OP : -/-
```

```
EXECUTE_OP : -/-
COMMIT_OP : -/-
ROLLBACK_OP : -/-
SAVEPOINT_OP : -/-
CONNECT_OP : -/-
DISCONNECT_OP : -/-
ALTER_SESSION_OP : -/-
ALTER_SYSTEM_OP : -/-
DDL_OP : -/-
```

```
isQL> AUDIT DDL BY sys WHENEVER NOT SUCCESSFUL;
Audit success.
```

```
USER_NAME : SYS
OBJECT_NAME : ALL
OBJECT_TYPE :
SELECT_OP : -/-
INSERT_OP : -/-
UPDATE_OP : -/-
DELETE_OP : -/-
MOVE_OP : -/-
MERGE_OP : -/-
ENQUEUE_OP : -/-
DEQUEUE_OP : -/-
LOCK_TABLE_OP : -/-
EXECUTE_OP : -/-
COMMIT_OP : -/-
ROLLBACK_OP : -/-
SAVEPOINT_OP : -/-
CONNECT_OP : -/-
DISCONNECT_OP : -/-
ALTER_SESSION_OP : -/-
ALTER_SYSTEM_OP : -/-
DDL_OP : -/T
```

SYS_COLUMNS_

모든 테이블에 정의된 칼럼들의 정보, 뷰의 가상 칼럼 정보, 그리고 시퀀스의 가상 칼럼 정보를 저장하는 메타 테이블이다.

| Column name | Type | Description |
|-------------|---------|------------------|
| COLUMN_ID | INTEGER | 칼럼 식별자 |
| DATA_TYPE | INTEGER | 데이터 타입 |
| LANG_ID | INTEGER | 언어 식별자 |
| OFFSET | BIGINT | 레코드 내 칼럼의 오프셋 |
| SIZE | BIGINT | 레코드 내 칼럼의 물리적 길이 |
| USER_ID | INTEGER | 사용자 식별자 |
| TABLE_ID | INTEGER | 테이블 식별자 |

| Column name | Type | Description |
|------------------|---------------|---|
| PRECISION | INTEGER | 칼럼에 지정한 정밀도 (precision) |
| SCALE | INTEGER | 칼럼에 지정한 스케일 (scale) |
| COLUMN_ORDER | INTEGER | 테이블에서 칼럼의 위치 |
| COLUMN_NAME | VARCHAR(128) | 칼럼 이름 |
| IS_NULLABLE | CHAR(1) | 널 (NULL) 허용 여부 T: NULL 허용 F: NULL 불허 |
| DEFAULT_VAL | VARCHAR(4000) | 기본 값 또는 수식 |
| STORE_TYPE | CHAR(1) | 칼럼의 저장 타입 V: 가변 (Variable) 방식 F: 고정 (Fixed) 방식 L: LOB 칼럼 |
| IN_ROW_SIZE | INTEGER | 메모리 테이블의 가변 길이 칼럼에 데이터가 입력될 때, 고정 영역(fixed area)에 저장될 수 있는 데이터의 최대 길이 |
| REPL_CONDITION | INTEGER | Deprecated |
| IS_HIDDEN | CHAR(1) | Hidden 칼럼인지 여부 T: 숨기는 칼럼 F: 공개된 칼럼 |
| IS_KEY_PRESERVED | CHAR(1) | 데이터 변경이 가능한 칼럼인지 여부 T: 변경 가능 F: 변경 불가능 |

칼럼 정보

COLUMN_ID

칼럼 식별자로 시스템 시퀀스에 의해 자동으로 부여된다.

DATA_TYPE

데이터 타입 식별자이다. 각 데이터 타입별 식별자 값은 다음과 같다.

| Data Type | 값 |
|-----------|----|
| CHAR | 1 |
| VARCHAR | 12 |
| NCHAR | -8 |
| NVARCHAR | -9 |
| NUMERIC | 2 |
| DECIMAL | 2 |
| FLOAT | 6 |
| NUMBER | 6 |
| DOUBLE | 8 |
| REAL | 7 |

| Data Type | 값 |
|-----------|-------|
| BIGINT | -5 |
| INTEGER | 4 |
| SMALLINT | 5 |
| DATE | 9 |
| BLOB | 30 |
| CLOB | 40 |
| BYTE | 20001 |
| NIBBLE | 20002 |
| BIT | -7 |
| VARBIT | -100 |
| GEOMETRY | 10003 |

데이터 타입에 대한 자세한 내용은 [General Reference-1.Data Types & Altibase Properties](#)를 참조한다.

LANG_ID

데이터 타입 (CHAR, VARCHAR)의 언어 속성 정보를 나타내는 칼럼이다.

OFFSET

레코드 내에서 칼럼의 물리적 시작 위치이다. 레코드의 물리적 저장 크기를 계산할 때 칼럼의 오프셋과 사이즈 값이 이용된다.

SIZE

레코드 내의 칼럼의 물리적 저장 사이즈로, 칼럼의 타입 및 사용자가 지정하는 정밀도 (precision) 등을 기준으로 시스템에 의해 계산된다.

USER_ID

칼럼이 속한 테이블 소유자의 식별자로, SYS_USERS_ 메타 테이블의 한 USER_ID 값과 일치한다.

TABLE_ID

칼럼이 속한 테이블의 식별자로, SYS_TABLES_ 메타 테이블의 한 TABLE_ID 값과 일치한다.

PRECISION

데이터 타입의 정밀도 (precision)로, 사용자가 지정하거나 시스템의 의해 기본값이 부여된다. 데이터 타입의 경우 사용자가 정의한 데이터 타입의 길이와 일치한다.

SCALE

데이터 타입의 스케일로, 사용자가 지정하거나 시스템이 기본값으로 부여한다. 타입에 따라 이 값은 사용되지 않는다.

COLUMN_ORDER

한 테이블 내에서 해당 칼럼이 보여지는 순서이다.

CREATE TABLE문에서 기술한 칼럼의 순서대로 칼럼이 생성되고, 테이블 내에서의 위치가 된다. ALTER TABLE문으로 칼럼을 추가한 경우 이 칼럼은 그 테이블의 마지막 칼럼으로 생성된다.

COLUMN_NAME

사용자가 테이블 생성 또는 칼럼 추가 시 명시한 칼럼의 이름이다.

IS_NULLABLE

칼럼에 NULL을 허용할 지 여부를 나타낸다.

칼럼 생성 시 사용자가 명시적으로 칼럼의 NULL 허용 여부를 명시할 수 있으며, 명시하지 않을 경우 기본으로 NULL을 허용한다.

DEFAULT_VAL

사용자가 해당 칼럼에 지정한 기본값이 표시된다.

해당 칼럼이 함수 기반 인덱스 생성으로 인해 자동으로 추가된 hidden 칼럼일 경우 함수 기반 인덱스 생성에 사용된 수식이 저장된다.

STORE_TYPE

칼럼을 물리적으로 저장할 때 레코드의 한 부분으로 기록할 수도 있고, 레코드 내에는 칼럼의 저장 위치 정보만을 저장하고 실제 칼럼 값은 다른 페이지에 기록할 수도 있다.

한 칼럼의 물리적 저장 크기가 크거나 레코드별로 칼럼의 저장 크기의 변동이 잦은 경우, 칼럼 정의 시 VARIABLE 옵션을 사용하면 레코드와 물리적으로 다른 페이지에 해당 칼럼을 저장할 수 있다. 일반적으로 VARCHAR 타입의 경우 문자열 길이가 긴 칼럼의 경우 이 옵션을 사용한다.

이 칼럼은 이러한 VARIABLE 옵션 지정 여부를 나타낸다.

IN_ROW_SIZE

메모리 테이블의 가변(VARIABLE) 길이 칼럼에 데이터가 입력될 때의 기본 in row size를 나타낸다. 가변 길이 칼럼에 데이터가 삽입될 때 데이터 길이가 이 값보다 작거나 같으면 고정 (fixed) 영역에 저장되고, 이 보다 긴 경우에는 가변 (variable) 영역에 들어가게 된다. 디스크 테이블의 경우 이 값은 항상 0이다.

IN ROW 절이나 VARIABLE 옵션(가변 길이 칼럼)에 대한 자세한 사항은 [General Reference-1.Data Types & Altibase Properties](#)을 참조한다.

IS_HIDDEN

해당 칼럼이 hidden 속성을 갖는지 여부를 나타낸다. 함수 기반 인덱스 생성 시, 테이블에 hidden 속성의 칼럼이 자동으로 추가된다. 이 칼럼에는 아래 두개의 값 중에서 하나가 표시된다.

- T: 숨기는 칼럼
- F: 공개된 칼럼

IS_KEY_PRESERVED

조인 뷰의 칼럼이 DML문으로 변경(INSERT, UPDATE, DELETE) 가능한 칼럼인지를 나타낸다. 일반 테이블의 칼럼일 경우 이 값이 'T'로 표시될 것이다. 뷰의 경우 변경 가능한 칼럼은 'T'로 표시되고 변경이 불가능한 칼럼은 'F'로 표시될 것이다.

- T: 변경 가능한 칼럼

- F: 변경 불가능한 칼럼

참조 테이블

SYS_USERS_
SYS_TABLES_
SYS_GEOMETRIES_

SYS_COMMENTS_

사용자가 정의한 테이블, 뷰 및 그에 소속된 칼럼에 대한 설명, 즉 주석을 기록하는 메타 테이블이다.

| Column name | Type | Description |
|-------------|---------------|-------------|
| USER_NAME | VARCHAR(128) | 사용자 이름 |
| TABLE_NAME | VARCHAR(128) | 테이블 이름 |
| COLUMN_NAME | VARCHAR(128) | 칼럼 이름 |
| COMMENTS | VARCHAR(4000) | 주석 내용 |

칼럼 정보

USER_NAME

테이블 소유자 이름으로, 이 값은 SYS_USERS_ 메타 테이블의 한 USER_NAME 값과 일치한다.

TABLE_NAME

테이블 (또는 뷰)의 이름으로, 이 값은 SYS_TABLES_ 메타 테이블의 한 TABLE_NAME 값과 동일하다.

COLUMN_NAME

테이블 (또는 뷰)에 속한 칼럼의 이름으로, 이 값은 SYS_COLUMNS_ 메타 테이블의 한 COLUMN_NAME 값과 동일하다.

단, 주석이 테이블 (또는 뷰)에 대한 설명일 경우에는 COLUMN_NAME의 값은 NULL 일 것이다.

COMMENTS

사용자가 기록한 주석 내용이다.

참조 테이블

SYS_USERS_
SYS_TABLES_
SYS_COLUMNS_

SYS_COMPRESSION_TABLES_

압축 칼럼에 대한 정보가 저장되는 메타 테이블이다.

| Column name | Type | Description |
|--------------|---------|--|
| TABLE_ID | INTEGER | 압축 칼럼을 포함하는 테이블의 식별자 |
| COLUMN_ID | INTEGER | 압축 칼럼의 식별자 |
| DIC_TABLE_ID | INTEGER | 압축 칼럼의 데이터가 저장되어 있는 딕셔너리 테이블의 식별자 |
| MAXROWS | BIGINT | 압축 칼럼의 데이터가 저장되어 있는 테이블에 입력할 수 있는 행의 최대 개수(0: 제한 없음) |

칼럼 정보

TABLE_ID

압축 칼럼이 속한 테이블의 식별자를 나타낸다. 이 값은 SYS_TABLES_ 메타 테이블의 한 TABLE_ID 값과 일치한다.

COLUMN_ID

압축 칼럼의 식별자로, SYS_COLUMNS_ 메타 테이블의 한 COLUMN_ID 값과 일치한다.

DIC_TABLE_ID

압축 칼럼의 데이터가 실제로 저장되어 있는 딕셔너리 테이블의 식별자를 나타낸다.

MAXROWS

압축 칼럼의 데이터가 실제로 저장되어 있는 딕셔너리 테이블에 입력할 수 있는 행의 최대 개수를 나타낸다.

SYS_CONSTRAINTS_

테이블의 제약 조건에 관한 정보를 포함하는 메타 테이블이다.

| Column name | Type | Description |
|---------------------|--------------|----------------------------------|
| USER_ID | INTEGER | 사용자 식별자 |
| TABLE_ID | INTEGER | 테이블 식별자 |
| CONSTRAINT_ID | INTEGER | 제약조건 식별자 |
| CONSTRAINT_NAME | VARCHAR(128) | 제약조건 이름 |
| CONSTRAINT_TYPE | INTEGER | 제약조건 타입 |
| INDEX_ID | INTEGER | 제약조건의 인덱스 식별자 |
| COLUMN_CNT | INTEGER | 제약조건에 관련된 칼럼 개수 |
| REFERENCED_TABLE_ID | INTEGER | FOREIGN KEY 제약조건으로 참조하는 테이블의 식별자 |
| REFERENCED_INDEX_ID | INTEGER | FOREIGN KEY 제약조건으로 참조하는 인덱스의 식별자 |

| Column name | Type | Description |
|-----------------|---------------|--|
| DELETE_RULE | INTEGER | FOREIGN KEY 제약조건을 위한 삭제 규칙 0: 종속적으로 삭제하지 않음 1: 종속적으로 삭제 2: SET NULL, 외래 키 관계에 의해 종속되는 칼럼 값을 NULL로 변경 |
| CHECK_CONDITION | VARCHAR(4000) | CHECK 제약조건의 조건 문자열 |
| VALIDATED | CHAR(1) | 모든 데이터가 제약조건을 따르는지 여부 |

칼럼 정보

USER_ID

사용자 식별자로 SYS_USERS_ 메타 테이블의 한 USER_ID 값과 동일하다.

TABLE_ID

제약 조건을 정의한 테이블 식별자로 SYS_TABLES_ 메타 테이블의 TABLE_ID 중 한 값과 동일하다.

CONSTRAINT_ID

제약 조건 식별자로 시스템 시퀀스에 의해 자동으로 부여된다.

CONSTRAINT_NAME

제약 조건의 이름을 나타낸다.

CONSTRAINT_TYPE

제약 조건의 타입을 나타내는 값으로 종류는 다음과 같다.

- 0: FOREIGN KEY
- 1: NOT NULL
- 2: UNIQUE
- 3: PRIMARY KEY
- 5: TIMESTAMP
- 6: LOCAL UNIQUE
- 7: CHECK

각 제약 조건의 기능에 대한 설명은 *SQL Reference*의 CREATE TABLE문에 있는 column constraint 설명을 참조한다.

INDEX_ID

UNIQUE 또는 PRIMARY KEY 제약 조건과 같이 제약조건을 정의하기 위해서 인덱스를 생성해야 할 때, 시스템은 내부적으로 인덱스를 생성한다. 이것은 이때 생성한 인덱스의 식별자로 SYS_INDICES_ 메타 테이블의 한 INDEX_ID 값과 동일하다.

COLUMN_CNT

제약 조건에 관련된 칼럼들의 개수를 나타낸다. 예를 들어 UNIQUE (i1, i2, i3) 과 같은 제약 조건을 생성하였다면 이 값은 3일 것이다.

REFERENCED_TABLE_ID

참조 제약조건 (Foreign key constraint)으로 참조하는 테이블의 식별자이다 (제약 조건이 정의된 테이블이 아니다). 이 식별자는 SYS_TABLES_ 메타 테이블의 한 TABLE_ID 값과 일치할 것이다.

REFERENCED_INDEX_ID

참조 제약조건 (Foreign key constraint)으로 참조하는 테이블에 존재해야 하는 UNIQUE 또는 PRIMARY KEY 제약조건의 식별자이다. 이 제약조건의 식별자 값은 SYS_CONSTRAINTS_ 메타 테이블의 한 CONSTRAINT_ID 값과 동일할 것이다.

CHECK_CONDITION

사용자가 CHECK 제약조건을 지정할 때 정의한 무결성 규칙(Integrity Rule)을 나타낸다.

VALIDATED

모든 데이터가 제약조건을 따르는지 여부를 나타낸다.

참조 테이블

```
SYS_USERS_
SYS_TABLES_
SYS_INDICES_
```

SYS_CONSTRAINT_COLUMNS_

사용자 테이블에 정의된 모든 제한조건에 관련된 칼럼의 정보를 기록하고 있는 메타 테이블이다.

| Column name | Type | Description |
|----------------------|---------|----------------|
| USER_ID | INTEGER | 사용자 식별자 |
| TABLE_ID | INTEGER | 테이블 식별자 |
| CONSTRAINT_ID | INTEGER | 제약조건 식별자 |
| CONSTRAINT_COL_ORDER | INTEGER | 제약조건내에서 칼럼의 순서 |
| COLUMN_ID | INTEGER | 칼럼 식별자 |

칼럼 정보**USER_ID**

테이블의 소유자 식별자로, SYS_USERS_ 메타 테이블의 한 USER_ID 값과 동일하다.

TABLE_ID

제약조건을 정의한 테이블의 식별자로, SYS_TABLES_ 메타 테이블의 한 TABLE_ID 값과 동일하다.

CONSTRAINT_ID

제약조건의 식별자로, SYS_CONSTRAINTS_ 메타 테이블의 어떤 CONSTRAINT_ID 값과 동일하다.

CONSTRAINT_COL_ORDER

제약조건 내에 정의된 칼럼의 위치이다. 예를 들어 UNIQUE (i1, i2, i3)과 같은 제약조건을 생성할 경우 SYS_CONSTRAINT_COLUMNS_ 메타 테이블에는 3개의 레코드가 삽입된다. 이 때 i1의 위치는 1, i2의 위치는 2, i3의 위치는 3이 각각 기록된다.

COLUMN_ID

제약조건에 정의된 칼럼의 식별자로, SYS_COLUMNS_ 메타 테이블의 한 COLUMN_ID 값과 동일하다.

참조 테이블

```
SYS_USERS_
SYS_TABLES_
SYS_CONSTRAINTS_
SYS_COLUMNS_
```

SYS_CONSTRAINT_RELATED_

제약조건(constraint)이 참조하고 있는 저장 함수에 대한 정보가 기록되어 있는 메타 테이블이다.

| Column name | Type | Description |
|-------------------|--------------|---------------------------|
| USER_ID | INTEGER | 사용자 식별자 |
| TABLE_ID | INTEGER | 테이블 식별자 |
| CONSTRAINT_ID | INTEGER | 제약조건 식별자 |
| RELATED_USER_ID | INTEGER | 제약조건이 참조하는 저장 함수의 소유자 식별자 |
| RELATED_PROC_NAME | VARCHAR(128) | 제약조건이 참조하는 저장 함수의 이름 |

칼럼 정보**USER_ID**

제약조건 소유자의 식별자로, SYS_USERS_ 메타 테이블의 한 USER_ID 값과 동일하다.

TABLE_ID

제약조건을 정의한 테이블의 식별자로, SYS_TABLES_ 메타 테이블의 한 TABLE_ID 값과 동일하다.

CONSTRAINT_ID

제약조건의 식별자로, SYS_CONSTRAINTS_ 메타 테이블의 한 CONSTRAINT_ID 값과 동일하다.

RELATED_USER_ID

제약조건이 참조하는 저장 함수 소유자의 식별자로, SYS_USERS_ 메타 테이블의 한 USER_ID 값과 동일하다.

RELATED_PROC_NAME

제약조건이 참조하는 저장 함수의 이름으로, SYS_PROCEDURES_ 메타 테이블의 한 PROC_NAME 값과 동일하다.

참조 테이블

```
SYS_USERS_
SYS_TABLES_
SYS_CONSTRAINTS_
SYS_PROCEDURES_
```

SYS_DATABASE_

데이터베이스 이름과 메타 테이블 버전 정보를 기록하는 테이블이다.

| Column name | Type | Description |
|----------------|---------------|-------------------------|
| DB_NAME | VARCHAR(40) | 데이터베이스 이름 |
| OWNER_DN | VARCHAR(2048) | 향후 확장 예정 |
| META_MAJOR_VER | INTEGER | 데이터베이스 메타 테이블 버전(주 버전) |
| META_MINOR_VER | INTEGER | 데이터베이스 메타 테이블 버전(부 버전) |
| META_PATCH_VER | INTEGER | 데이터베이스 메타 테이블 버전(패치 버전) |

칼럼 정보

DB_NAME

데이터베이스 생성시 지정한 데이터베이스 이름이 저장된다.

META_MAJOR_VER

메타 테이블의 주 버전을 나타낸다. 주 버전은 메타 테이블의 정의가 변경되거나 메타 테이블이 추가 또는 삭제 될 경우 증가한다. 데이터베이스의 이 버전과 Altibase 바이너리의 해당 버전이 일치하지 않은 경우 데이터베이스 마이그레이션 작업을 요한다.

META_MINOR_VER

메타 테이블의 부 버전을 나타낸다. 부 버전은 메타 테이블의 일부 스키마 또는 레코드 값이 변경될 경우 증가한다. 데이터베이스의 이 버전과 Altibase 바이너리의 해당 버전이 다른 경우, 내부적으로 값을 비교해 상위 버전으로 메타 테이블의 자동 업그레이드를 수행한다.

META_PATCH_VER

메타 테이블 패치 버전을 나타낸다.

SYS_DATABASE_LINKS_

데이터베이스 링크 정보를 기록하는 메타 테이블이다

| Column name | Type | Description |
|-------------|---------|------------------|
| USER_ID | INTEGER | 사용자 식별자 |
| LINK_ID | INTEGER | 데이터베이스 링크 식별자 |
| LINK_OID | BIGINT | 데이터베이스 링크 객체 식별자 |

| Column name | Type | Description |
|-----------------|--------------|---|
| LINK_NAME | VARCHAR(40) | 데이터베이스 링크 이름 |
| USER_MODE | INTEGER | 원격 서버로의 접근 방법 |
| REMOTE_USER_ID | VARCHAR(128) | 원격 데이터베이스의 사용자 계정 |
| REMOTE_USER_PWD | BYTE(40) | 원격 데이터베이스의 사용자 비밀번호 |
| LINK_TYPE | INTEGER | Heterogeneous Link인지 Homogeneous Link인지를 나타냄. |
| TARGET_NAME | VARCHAR(40) | 데이터베이스 링크 객체가 접근할 원격서버의 이름 |
| CREATED | DATE | 데이터베이스 링크 객체가 생성된 일시 |
| LAST_DDL_TIME | DATE | 데이터베이스 링크 객체에 마지막으로 DDL 변경 작업이 일어난 일시 |

칼럼 정보

USER_ID

데이터베이스 링크 소유자의 식별자이다.

LINK_ID

데이터베이스 링크 식별자이다.

LINK_OID

데이터베이스 링크의 객체 식별자이다.

LINK_NAME

사용자가 데이터베이스 링크 생성 시에 명시한 데이터베이스 링크 이름을 나타낸다.

USER_MODE

원격 서버로의 접근 방법을 나타낸다.

- 0: DEDICATE USER MODE
- 1: CURRENT USER MODE (향후 사용을 위해 예약됨)

REMOTE_USER_ID

원격 데이터베이스 서버에 접근할 때 사용하는 원격 서버 사용자 계정을 나타낸다.

REMOTE_USER_PWD

원격 데이터베이스 서버에 접근할 때 사용하는 원격 서버 사용자 비밀번호를 나타낸다. 비밀번호는 복호화가 가능한 암호화 알고리즘으로 암호화하여 저장한다.

LINK_TYPE

Heterogeneous Link인지 Homogeneous Link인지를 나타낸다.

TARGET_NAME

데이터베이스 링크 객체가 접근할 원격서버의 이름을 나타낸다.

CREATED

데이터베이스 링크 객체가 생성된 일시를 나타낸다.

LAST_DDL_TIME

데이터베이스 링크 객체에 마지막으로 DDL 변경 작업이 일어난 일시를 나타낸다.

SYS_DIRECTORIES_

저장프로시저 내에서 파일 제어를 하기 위해 사용하는 디렉터리에 대한 정보를 기록하는 테이블이다.

| Column name | Type | Description |
|----------------|---------------|--|
| DIRECTORY_ID | BIGINT | 디렉터리 식별자 |
| USER_ID | INTEGER | 사용자 식별자 |
| DIRECTORY_NAME | VARCHAR(128) | 디렉터리 이름 |
| DIRECTORY_PATH | VARCHAR(4000) | 시스템에서 디렉터리의 절대 경로 |
| CREATED | DATE | 디렉터리가 생성된 시간 |
| LAST_DDL_TIME | DATE | 디렉터리에 대해 가장 최근에 DDL 변경작업이 마지막으로 일어난 시간 |

칼럼 정보**DIRECTORY_ID**

디렉터리 식별자로 시스템 내에서 유일값을 가진다.

USER_ID

디렉터리 소유자의 사용자 식별자를 나타낸다.

DIRECTORY_NAME

디렉터리 이름으로 시스템 내 유일값을 가진다.

DIRECTORY_PATH

디렉터리가 위치하는 시스템 내 절대 경로로, CREATE DIRECTORY문 수행 시 사용자가 명시적으로 지정한다.

LAST_DDL_TIME

디렉터리 객체에 마지막으로 DDL 변경 작업이 일어난 시간을 나타낸다.

SYS_ENCRYPTED_COLUMNS_

보안 설정에 기반한 추가적인 보안 정보를 암호화된 칼럼별로 관리하기 위한 메타 테이블이다.

| Column name | Type | Description |
|-------------------|--------------|--------------------|
| USER_ID | INTEGER | 보안 칼럼이 속한 테이블의 소유자 |
| TABLE_ID | INTEGER | 보안 칼럼이 속한 테이블의 식별자 |
| COLUMN_ID | INTEGER | 보안 대상 칼럼의 식별자 |
| ENCRYPT_PRECISION | INTEGER | 보안 칼럼의 precision |
| POLICY_NAME | VARCHAR(16) | 보안 정책의 이름 |
| POLICY_CODE | VARCHAR(128) | 보안 정책에 대한 검증 코드 |

SYS_GRANT_OBJECT_

사용자에게 부여된 객체 권한 정보를 저장한다.

| Column name | Type | Description |
|-------------------|------------|---|
| GRANTOR_ID | INTEGER | 권한을 부여한 사용자의 식별자 |
| GRANTEE_ID | INTEGER | 권한이 부여된 사용자의 식별자 |
| PRIV_ID | INTEGER | 권한 식별자 |
| USER_ID | INTEGER | 객체 소유자의 식별자 |
| OBJ_ID | BIGINT | 객체 식별자 |
| OBJ_TYPE | VARCHAR(1) | 객체 타입 |
| WITH_GRANT_OPTION | INTEGER | 객체 접근 권한 부여시 WITH GRANT OPTION의 사용 유무 0: 사용 안 함 1: 사용함 |

칼럼 정보

GRANTOR_ID

권한을 부여한 사용자의 식별자로, SYS_USERS_ 메타 테이블의 한 USER_ID 값과 동일하다.

GRANTEE_ID

권한을 부여받은 사용자의 식별자로, SYS_USERS_ 메타 테이블의 한 USER_ID 값과 동일하다. 단, 객체 권한을 public에게 부여한 경우, SYS_USERS_ 메타 테이블에 존재하지 않는 USER_ID 값인 "0"이 이 칼럼에 나타난다.

PRIV_ID

권한 식별자로 SYS_PRIVILEGES_ 메타 테이블의 한 PRIV_ID 값과 동일하다.

USER_ID

해당 권한과 관련된 객체 소유자의 사용자 식별자로, SYS_USERS_ 메타 테이블의 한 USER_ID 값과 동일하다.

OBJ_ID

해당 권한과 관련된 객체의 식별자로, 메타 테이블에 저장된 대상 객체의 식별자와 1:1 관계이다.

대상 객체가 테이블, 뷰 또는 시퀀스인 경우에는 SYS_TABLES_ 메타 테이블의 한 TABLE_ID와 매핑되고, 대상 객체가 저장 프로시저이거나 저장 함수일 경우에는 SYS_PROCEDURES_ 메타 테이블의 한 PROC_OID와 매핑된다.

OBJ_TYPE

해당 권한과 관련된 객체의 종류를 나타낸다.

- A: 저장 패키지
- D: 디렉토리
- T: 테이블 또는 뷰
- S: 시퀀스
- P: 저장 프로시저 또는 저장 함수
- Y: 라이브러리

WITH_GRANT_OPTION

권한을 부여받은 사용자가 다른 사용자에게 해당 권한을 부여할 수 있는 권한이 있는지 여부를 나타낸다.

참조 테이블

```
SYS_USERS_
SYS_PRIVILEGES_
SYS_TABLES_
SYS_PROCEDURES_
```

SYS_GRANT_SYSTEM_

사용자에게 부여된 시스템 권한 정보를 포함한다.

| Column name | Type | Description |
|-------------|---------|------------------|
| GRANTOR_ID | INTEGER | 권한을 부여한 사용자의 식별자 |
| GRANTEE_ID | INTEGER | 권한이 부여된 사용자의 식별자 |
| PRIV_ID | INTEGER | 권한 식별자 |

칼럼 정보

GRANTOR_ID

권한을 부여한 사용자의 사용자 식별자로, SYS_USERS_ 메타 테이블의 한 USER_ID 값과 동일하다.

GRANTEE_ID

권한을 부여받은 사용자의 사용자 식별자로, SYS_USERS_ 메타 테이블의 한 USER_ID 값과 동일하다.

PRIV_ID

권한 식별자로 SYS_PRIVILEGES_ 메타 테이블의 한 PRIV_ID 값과 동일하다.

참조 테이블

SYS_USERS_
SYS_PRIVILEGES_

SYS_INDEX_COLUMNS_

모든 테이블에 정의된 인덱스에 연관된 칼럼의 정보를 기록하고 있는 메타 테이블이다.

| Column name | Type | Description |
|-----------------|---------|----------------|
| USER_ID | INTEGER | 사용자 식별자 |
| INDEX_ID | INTEGER | 인덱스 식별자 |
| COLUMN_ID | INTEGER | 칼럼의 식별자 |
| INDEX_COL_ORDER | INTEGER | 인덱스 내에서 칼럼의 위치 |
| SORT_ORDER | CHAR(1) | 정렬 순서 |
| TABLE_ID | INTEGER | 테이블 식별자 |

칼럼 정보

USER_ID

인덱스 소유자의 사용자 식별자로, SYS_USERS_ 메타 테이블의 한 USER_ID 값과 동일하다.

INDEX_ID

인덱스 식별자로, SYS_INDICES_ 메타 테이블의 한 INDEX_ID 값과 동일하다.

COLUMN_ID

인덱스를 생성한 칼럼의 식별자로, SYS_COLUMNS_ 메타 테이블의 한 COLUMN_ID 값과 동일하다.

INDEX_COL_ORDER

복합 인덱스 (composite index)의 경우 여러 개의 칼럼에 한 인덱스를 생성하므로, 이 때 해당 칼럼이 인덱스에서 몇 번째 위치하는지를 나타내는 값이다.

SORT_ORDER

인덱스가 오름차순 또는 내림차순으로 정렬되었는지를 나타낸다.

- A: 오름차순
- D: 내림차순

TABLE_ID

인덱스를 생성한 테이블의 식별자로, SYS_TABLES_ 메타 테이블의 한 TABLE_ID 값과 동일하다.

참조 테이블

```
SYS_USERS_
SYS_TABLES_
SYS_COLUMNS_
SYS_INDICES_
```

SYS_INDEX_PARTITIONS_

인덱스 파티션을 관리하는 메타 테이블이다.

| Column name | Type | Description |
|----------------------|---------------|------------------------------|
| USER_ID | INTEGER | 사용자 식별자 |
| TABLE_ID | INTEGER | 테이블 식별자 |
| INDEX_ID | INTEGER | 인덱스 식별자 |
| TABLE_PARTITION_ID | INTEGER | 테이블 파티션 식별자 |
| INDEX_PARTITION_ID | INTEGER | 인덱스 파티션 식별자 |
| INDEX_PARTITION_NAME | VARCHAR(128) | 인덱스 파티션 이름 |
| PARTITION_MIN_VALUE | VARCHAR(4000) | 사용되지 않음 |
| PARTITION_MAX_VALUE | VARCHAR(4000) | 사용되지 않음 |
| TBS_ID | INTEGER | 테이블스페이스 식별자 |
| CREATED | DATE | 인덱스 파티션이 생성된 시간 |
| LAST_DDL_TIME | DATE | 인덱스 파티션을 마지막으로 DDL 변경 작업한 시간 |

칼럼 정보**USER_ID**

인덱스 소유자의 사용자 식별자로, SYS_USERS_ 메타 테이블의 한 USER_ID 값과 동일하다.

TABLE_ID

인덱스를 생성한 테이블의 테이블 식별자로, SYS_TABLES_ 메타 테이블의 한 TABLE_ID 값과 동일하다.

INDEX_ID

인덱스 식별자로, SYS_INDICES_ 메타 테이블의 한 INDEX_ID 값과 동일하다.

TABLE_PARTITION_ID

테이블 파티션의 식별자이다.

INDEX_PARTITION_ID

인덱스 파티션의 식별자이다.

INDEX_PARTITION_NAME

인덱스 파티션의 이름으로, 사용자가 명시한 값이다.

TBS_ID

인덱스가 저장되는 테이블스페이스의 식별자이다.

참조 테이블

```
SYS_USERS_
SYS_TABLES_
SYS_INDICES_
SYS_TABLE_PARTITIONS_
```

SYS_INDEX_RELATED_

함수 기반 인덱스(Function-based Index)가 기반하고 있는 저장 함수들에 대한 정보가 기록되어 있는 메타 테이블이다.

| Column name | Type | Description |
|-------------------|--------------|--------------------------|
| USER_ID | INTEGER | 사용자 식별자 |
| TABLE_ID | INTEGER | 테이블 식별자 |
| INDEX_ID | INTEGER | 인덱스 식별자 |
| RELATED_USER_ID | INTEGER | 인덱스가 참조하는 저장 함수의 소유자 식별자 |
| RELATED_PROC_NAME | VARCHAR(128) | 인덱스가 참조하는 저장 함수의 이름 |

칼럼 정보**USER_ID**

인덱스 소유자의 식별자이다. SYS_USERS_ 메타 테이블의 한 USER_ID 값과 동일하다.

TABLE_ID

인덱스가 정의된 테이블의 식별자로, SYS_TABLES_ 메타 테이블의 한 TABLE_ID 값과 동일하다.

INDEX_ID

인덱스 식별자로, SYS_INDICES_ 메타 테이블의 한 INDEX_ID 값과 동일하다.

RELATED_USER_ID

인덱스가 참조하는 저장 함수 소유자의 식별자로, SYS_USERS_ 메타 테이블의 한 USER_ID 값과 동일하다.

RELATED_PROC_NAME

인덱스가 참조하는 저장 함수의 이름으로, SYS_PROCEDURES_ 메타 테이블의 한 PROC_NAME 값과 동일하다.

참조 테이블

```
SYS_USERS_
SYS_TABLES_
SYS_INDICES_
SYS_PROCEDURES_
```

SYS_INDICES_

모든 테이블에 정의된 모든 인덱스 정보를 기록하고 있는 메타 테이블이다.

| Column name | Type | Description |
|----------------|--------------|--|
| USER_ID | INTEGER | 사용자 식별자 |
| TABLE_ID | INTEGER | 테이블 식별자 |
| INDEX_ID | INTEGER | 인덱스 식별자 |
| INDEX_NAME | VARCHAR(128) | 인덱스 이름 |
| INDEX_TYPE | INTEGER | 인덱스 타입 |
| IS_UNIQUE | CHAR(1) | 중복 키 값 허용 여부 |
| COLUMN_CNT | INTEGER | 인덱스 칼럼 개수 |
| IS_RANGE | CHAR(1) | 범위 검색 가능 여부 |
| IS_PERS | CHAR(1) | 인덱스 영구 저장 여부 |
| IS_DIRECTKEY | CHAR(1) | 다이렉트 키 인덱스 여부 |
| TBS_ID | INTEGER | 테이블스페이스 식별자 |
| IS_PARTITIONED | CHAR(1) | 파티션드 인덱스인지 여부 |
| INDEX_TABLE_ID | INTEGER | 파티션드 테이블의 년파티션드 인덱스가 보조적으로 생성한 테이블 식별자 |

| Column name | Type | Description |
|---------------|------|--|
| CREATED | DATE | 인덱스가 생성된 시간 |
| LAST_DDL_TIME | DATE | DDL 구문을 사용해서 인덱스에 대해 마지막으로 변경 작업이 일어난 시간 |

칼럼 정보

USER_ID

인덱스 소유자의 사용자 식별자로, SYS_USERS_ 메타 테이블의 한 USER_ID 값과 동일하다.

TABLE_ID

인덱스를 생성한 테이블의 테이블 식별자로, SYS_TABLES_ 메타 테이블의 한 TABLE_ID 값과 동일하다.

INDEX_ID

인덱스 식별자로, 시스템 시퀀스에 의해 자동으로 부여된다.

INDEX_NAME

인덱스의 이름이다.

INDEX_TYPE

인덱스 타입을 나타낸다. 1이면 B-TREE 인덱스이고, 2이면 R-TREE 인덱스이다.

IS_UNIQUE

중복 키 값 허용여부를 나타낸다.

- T: 중복 키 값을 허용하지 않는다.
- F: 중복 키 값을 허용한다.

COLUMN_CNT

인덱스를 구성하는 칼럼의 개수를 나타낸다.

IS_RANGE

범위 검색 가능 여부를 나타낸다.

- T: 범위 검색 가능
- F: 범위 검색 불가능

IS_DIRECTKEY

다이렉트 키(Direct Key) 인덱스의 사용 여부를 나타낸다.

- T: 다이렉트 키 인덱스
- F: 일반 인덱스

TBS_ID

인덱스가 저장되는 테이블스페이스의 식별자이다.

IS_PARTITIONED

파티션드 인덱스인지 여부를 나타내는 식별자이다. 'T'는 파티션드 인덱스, 'F'는 파티션드 인덱스가 아니다.

참조 테이블

```
SYS_USERS_
SYS_TABLES_
```

SYS_JOBS_

JOB에 대한 정보가 기록되는 메타 테이블이다.

| Column name | Type | Description |
|----------------|---------------|--|
| JOB_ID | INTEGER | JOB의 식별자 |
| JOB_NAME | VARCHAR(128) | JOB의 이름 |
| EXEC_QUERY | VARCHAR(1000) | JOB에 등록한 프로시저 |
| START_TIME | DATE | JOB이 처음으로 시작하는 시간 |
| END_TIME | DATE | JOB이 끝나는 시간 |
| INTERVAL | INTEGER | 실행 주기 |
| INTERVAL_TYPE | CHAR(2) | 실행 주기의 단위(YY, MM, DD, HH, MI) |
| STATE | INTEGER | 현재 실행중인 JOB의 상태 0: 실행되지 않음 1: 실행되고 있음 |
| LAST_EXEC_TIME | DATE | 마지막으로 JOB을 실행한 시간 |
| EXEC_COUNT | INTEGER | JOB의 실행 횟수 |
| ERROR_CODE | CHAR(7) | 에러 코드(NULL은 성공을 의미함) |
| IS_ENABLE | CHAR(1) | 작업 스케줄러에서 JOB 실행 여부 T: 실행 가능 F: 실행 불가능 |
| COMMENT | VARCHAR(4000) | JOB에 대한 부가 설명 |

칼럼 정보**EXEC_QUERY**

JOB에 등록되어 실행되는 프로시저를 나타낸다.

INTERVAL_TYPE

JOB의 실행 주기가 설정되어 있는 경우, 시간의 단위를 나타낸다. 즉, INTERVAL 칼럼에 값이 있을 경우 그 값의 단위이다.

- YY: 년
- MM: 월
- DD: 일
- HH: 시
- MI: 분

STATE

JOB이 현재 실행되고 있는지 여부를 나타낸다.

- 0: 실행되지 않음
- 1: 실행되고 있음

EXEC_COUNT

JOB이 생성된 이후 등록된 프로시저가 몇 번 실행되었는지 총 횟수를 나타낸다.

ERROR_CODE

마지막으로 JOB이 실행되었을 때의 프로시저 수행이 실패하였을 때 에러 코드를 나타낸다. 성공하였을 때는 NULL이다.

IS_ENABLE

작업 스케줄러에서 JOB을 실행할 수 있는지 여부를 나타낸다.

- T: 실행 가능
- F: 실행 불가능

COMMENT

JOB에 대하여 설명을 나타낸다. 설명이 기술되지 않으면 NULL 값이 조회된다.

SYS_LIBRARIES_

외부 라이브러리 객체에 대한 정보를 기록하는 메타 테이블이다.

| Column name | Type | Description |
|--------------|---------------|------------------|
| LIBRARY_ID | BIGINT | 라이브러리 식별자 |
| USER_ID | INTEGER | 사용자 식별자 |
| LIBRARY_NAME | VARCHAR(128) | 라이브러리 이름 |
| FILE_SPEC | VARCHAR(4000) | 동적 라이브러리 파일의 경로 |
| DYNAMIC | VARCHAR(1) | 향후 사용 예약 |
| STATUS | VARCHAR(7) | 향후 사용 예약 |
| CREATED | DATE | 라이브러리 객체가 생성된 시간 |

| Column name | Type | Description |
|---------------|------|-----------------------------------|
| LAST_DDL_TIME | DATE | 라이브러리 객체에 마지막으로 DDL 변경 작업이 일어난 시간 |

칼럼 정보

LIBRARY_ID

라이브러리 식별자로서 시스템 내에서 유일값을 가진다.

USER_ID

라이브러리 소유자의 사용자 식별자를 나타낸다.

LIBRARY_NAME

라이브러리 객체의 이름으로 시스템 내에서 유일값을 가진다.

FILE_SPEC

라이브러리 객체가 가리키는 동적 라이브러리 파일의 경로를 나타낸다. 라이브러리 파일이 위치하는 기본 경로 (\$ALTIBASE_HOME/lib)에 대한 상대 경로로 표시된다.

SYS_LOBS_

테이블에 정의된 LOB 칼럼의 정보를 기록하는 메타 테이블이다.

| Column name | Type | Description |
|----------------|---------|---|
| USER_ID | INTEGER | 사용자 식별자 |
| TABLE_ID | INTEGER | 테이블 식별자 |
| COLUMN_ID | INTEGER | 칼럼 식별자 |
| TBS_ID | INTEGER | 테이블스페이스 식별자 |
| LOGGING | CHAR(1) | 향후 확장 예정 |
| BUFFER | CHAR(1) | 향후 확장 예정 |
| IS_DEFAULT_TBS | CHAR(1) | LOB 칼럼 저장용 테이블스페이스 여부 T: 지정함 F: 지정하지 않음 |

칼럼 정보

USER_ID

LOB 칼럼이 속한 테이블 소유자의 식별자로, SYS_USERS_ 메타 테이블의 한 USER_ID 값과 동일하다.

TABLE_ID

LOB 칼럼이 속한 테이블의 식별자로, SYS_TABLES_ 메타 테이블의 한 TABLE_ID 값과 동일하다.

COLUMN_ID

LOB 칼럼의 식별자이다.

TBS_ID

LOB 칼럼이 저장되는 테이블스페이스의 식별자이다.

IS_DEFAULT_TBS

LOB 칼럼 생성 시, 사용자가 LOB 칼럼이 저장될 테이블스페이스를 지정했는지를 나타낸다.

- T: 지정함
- F: 지정하지 않음

자세한 설명은 *SQL Reference*의 CREATE TABLE > LOB_STORAGE_CLAUSE 구문을 참조한다.

참조 테이블

```
SYS_USERS_
SYS_TABLES_
SYS_COLUMNS_
```

SYS_MATERIALIZED_VIEWS_

Materialized view에 대한 정보가 기록된 메타 테이블이다.

| Column name | Type | Description |
|-------------------|--------------|---|
| USER_ID | INTEGER | 사용자 식별자 |
| MVIEW_ID | INTEGER | Materialized view의 식별자 |
| MVIEW_NAME | VARCHAR(128) | Materialized view의 이름 |
| TABLE_ID | INTEGER | 테이블 식별자 |
| VIEW_ID | INTEGER | 뷰 식별자 |
| REFRESH_TYPE | CHAR(1) | Refresh 타입 |
| REFRESH_TIME | CHAR(1) | Refresh 시기 |
| CREATED | DATE | Materialized view가 생성된 시간 |
| LAST_DDL_TIME | DATE | Materialized view에 대해 마지막으로 DDL 변경 작업이 일어난 시간 |
| LAST_REFRESH_TIME | DATE | Materialized view를 마지막으로 refresh한 시각 |

칼럼 정보**USER_ID**

Materialized view 소유자의 사용자 식별자로, SYS_USERS_ 메타 테이블의 한 USER_ID 값과 동일하다.

MVIEW_ID

Materialized view 식별자로, 데이터베이스가 자동으로 부여한다.

MVIEW_NAME

사용자가 명시한 materialized view의 이름이다.

TABLE_ID

Materialized view의 데이터 유지를 위해 자동으로 생성되는 테이블의 식별자이다. SYS_TABLES_ 메타 테이블에서 이 식별자로 조회해 보면 해당 materialized view의 이름과 동일한 이름의 테이블이 존재하는 것을 확인할 수 있다.

VIEW_ID

Materialized view의 데이터 유지를 위해 자동으로 생성되는 뷰의 식별자이다. SYS_VIEWS_ 메타 테이블에서 이 식별자로 해당 뷰를 조회할 수 있다.

REFRESH_TYPE

Materialized view의 리프레쉬 방법을 나타내는 값이다.

- C: COMPLETE
- F: FAST
- R: FORCE

REFRESH_TIME

Materialized view의 리프레쉬 시기를 나타내는 값이다.

- D: ON DEMAND
- C: ON COMMIT

CREATED

Materialized view가 생성된 일시를 나타낸다.

LAST_DDL_TIME

Materialized view에 대해 DDL 변경 작업이 마지막으로 일어난 일시를 나타낸다.

LAST_REFRESH_TIME

Materialized view를 마지막으로 리프레쉬한 일시를 나타낸다.

참조 테이블

```
SYS_USERS_
SYS_TABLES_
SYS_VIEWS_
SYS_VIEW_PARSE_
```

SYS_PACKAGES_

패키지에 대한 정보가 기록된 메타 테이블이다.

| Column name | Type | Description |
|---------------|--------------|---|
| USER_ID | INTEGER | 패키지 소유자 식별자 |
| PACKAGE_OID | BIGINT | 패키지 식별자 |
| PACKAGE_NAME | VARCHAR(128) | 패키지 이름 |
| PACKAGE_TYPE | INTEGER | 패키지 유형. 패키지 스펙인지 패키지 바디인지를 나타낸다. 6: 패키지 스펙 7: 패키지 바디 |
| AUTHID | INTEGER | 패키지의 실행자 권한 0: 생성자 권한(DEFINER) 1: 사용자 권한(CURRENT_USER) |
| STATUS | INTEGER | 패키지의 상태를 나타낸다. INVALID이면 실행 불가능 상태이다. 0: VALID 1: INVALID |
| CREATED | DATE | 패키지를 생성한 일시 |
| LAST_DDL_TIME | DATE | 패키지에 DDL 변경 작업이 마지막으로 일어난 일시 |

칼럼 정보

USER_ID

패키지 소유자의 사용자 식별자로, SYS_USERS_ 메타 테이블의 한 USER_ID 값과 동일하다.

PACKAGE_OID

패키지의 식별자로, 시스템에 의해 자동으로 부여된다.

PACKAGE_NAME

패키지의 이름이다.

PACKAGE_TYPE

패키지 스펙인지 패키지 바디인지를 나타내는 값이다.

- 6: 패키지 스펙
- 7: 패키지 바디

AUTHID

패키지를 실행하는 권한을 나타내는 값이다.

- 0: 생성자 권한(DEFINER)
- 1: 사용자 권한(CURRENT_USER)

STATUS

패키지의 실행 가능 여부를 나타내는 값이다. 0 (VALID) 은 실행 가능함을 나타낸다.

- 0: VALID
- 1: INVALID

CREATED

패키지가 생성된 일시를 나타낸다.

LAST_DDL_TIME

패키지에 DDL 변경 작업이 마지막으로 일어난 일시를 나타낸다.

참조 테이블

SYS_USERS_

SYS_PACKAGE_PARAS_

패키지에 포함된 서브프로그램(저장 프로시저와 저장 함수)들의 인자 (parameter)들에 대한 정보가 기록된 메타 테이블이다.

| Column name | Type | Description |
|--------------|---------------|-------------------------------|
| USER_ID | INTEGER | 패키지 소유자 식별자 |
| OBJECT_NAME | VARCHAR(128) | 서브프로그램 이름 |
| PACKAGE_NAME | VARCHAR(128) | 패키지 이름 |
| PACKAGE_OID | BIGINT | 패키지 식별자 |
| SUB_ID | INTEGER | 서브프로그램 식별자 |
| SUB_TPYE | INTEGER | 서브프로그램 유형. 0: 프로시저 1: 함수 |
| PARA_NAME | VARCHAR(128) | 서브프로그램의 파라미터 이름 |
| PARA_ORDER | INTEGER | 파라미터의 순서. 첫번째 파라미터의 경우 1을 가짐. |
| INOUT_TYPE | INTEGER | 파라미터의 입력, 출력, 입출력 여부 |
| DATA_TYPE | INTEGER | 파라미터의 데이터 타입 |
| LANG_ID | INTEGER | 파라미터 타입 언어 식별자 |
| SIZE | INTEGER | 파라미터 타입의 크기 |
| PRECISION | INTEGER | 파라미터 타입의 precision |
| SCALE | INTEGER | 파라미터 타입의 scale |
| DEFAULT_VAL | VARCHAR(4000) | 파라미터의 기본값 |

칼럼 정보

USER_ID

저장 프로시저 또는 저장 함수 소유자의 사용자 식별자로, SYS_USERS_ 메타 테이블의 한 USER_ID 값과 동일하다.

OBJECT_NAME

서브프로그램의 이름이다.

PACKAGE_NAME

패키지의 이름이다.

PACKAGE_OID

패키지의 식별자로, SYS_PACKAGES_ 메타 테이블에서 패키지 스펙에 해당하는 PACKAGE_OID 값들 중 하나와 동일하다.

SUB_ID

서브프로그램의 식별자이다. 패키지 내에서 서브프로그램들의 식별자는 1부터 시작되며 작성한 순서대로 번호가 부여된다.

SUB_TYPE

서브프로그램이 저장 프로시저인지 또는 저장 함수인지를 나타낸다.

- 0: 프로시저
- 1: 함수

PARA_NAME

서브프로그램의 파라미터 이름이다.

PARA_ORDER

여러 파라미터들 중 해당 파라미터가 몇번째 정의된 파라미터인지를 나타내는 값이다.

INOUT_TYPE

저장 프로시저 또는 저장 함수의 파라미터가 입력인자, 출력인자, 또는 입출력인자인지를 나타낸다.

- 0: IN
- 1: OUT
- 2: IN OUT

DATA_TYPE

파라미터의 데이터 타입 식별자이다. 데이터 타입 식별자 값은 SYS_COLUMNS_ 메타 테이블의 DATA_TYPE 칼럼 설명을 참조한다.

데이터 타입에 대한 자세한 내용은 [General Reference-1.Data Types & Altibase Properties](#)를 참조한다.

LANG_ID

타입 (CHAR, VARCHAR)의 언어 속성 정보를 나타내는 칼럼이다.

SIZE

데이터 타입의 물리적 크기이다.

PRECISION

인자 데이터 타입의 정밀도 (precision)으로, 사용자가 지정하거나 또는 시스템이 기본 값으로 부여한다. 타입의 경우 사용자가 정의한 타입의 길이이다.

SCALE

인자 데이터 타입의 scale로, 사용자가 지정하거나 또는 시스템이 기본 값으로 부여한다. 타입에 따라 이 값은 사용하지 않을 수 있다.

데이터 타입의 precision 과 scale에 대한 상세한 내용은 [General Reference-1.Data Types & Altibase Properties](#)를 참조한다.

DEFAULT_VAL

파라미터 정의 시 사용자가 지정하는 파라미터 기본 값이다.

참조 테이블

SYS_USERS_
SYS_PACKAGES_

SYS_PACKAGE_PARSE_

사용자가 정의한 패키지의 구문 텍스트를 기록하는 메타 테이블이다.

| Column name | Type | Description |
|--------------|--------------|--|
| USER_ID | INTEGER | 패키지 소유자 식별자 |
| PACKAGE_OID | BIGINT | 패키지 식별자 |
| PACKAGE_TYPE | INTEGER | 패키지 유형. 패키지 스펙인지 패키지 바디인지를 나타낸다. 6: 패키지 스펙 7: 패키지 바디 |
| SEQ_NO | INTEGER | 나뉘어 여러 레코드로 저장된 구문들 중 레코드의 순서 |
| PARSE | VARCHAR(100) | 나뉘어 저장된 구문 |

칼럼 정보**USER_ID**

패키지 소유자의 사용자 식별자로, SYS_USERS_ 메타 테이블의 한 USER_ID 값과 동일하다.

PACKAGE_OID

패키지의 식별자로, SYS_PACKAGES_ 메타 테이블의 한 PACKAGE_OID 값과 동일하다.

PACKAGE_TYPE

패키지 스펙인지 패키지 바디인지를 나타내는 값이다.

- 6: 패키지 스펙
- 7: 패키지 바디

SEQ_NO

패키지의 구문 정보를 나누어서 SYS_PACKAGE_PARSE_에 여러 개의 레코드로 저장할 때, 각 레코드의 순서를 나타낸다.

PARSE

패키지 구문의 문자열의 조각이다. 한 PACKAGE_OID 값으로 레코드들을 검색하여 SEQ_NO 순서대로 PARSE 값을 합치면 패키지 생성 구문이 된다.

참조 테이블

SYS_USERS_
SYS_PACKAGES_

SYS_PACKAGE_RELATED_

패키지 내에 포함된 저장 프로시저와 저장 함수들이 참조하는 테이블, 시퀀스, 저장 프로시저, 저장 함수, 또는 뷰들에 대한 정보가 기록된 메타 테이블이다.

| Column name | Type | Description |
|---------------------|--------------|--------------------------|
| USER_ID | INTEGER | 패키지 소유자 식별자 |
| PACKAGE_OID | BIGINT | 패키지 식별자 |
| RELATED_USER_ID | INTEGER | 패키지 내에서 참조하는 객체의 소유자 식별자 |
| RELATED_OBJECT_NAME | VARCHAR(128) | 패키지 내에서 참조하는 객체의 이름 |
| RELATED_OBJECT_TYPE | INTEGER | 패키지 내에서 참조하는 객체의 타입 |

칼럼 정보**USER_ID**

패키지 소유자의 사용자 식별자로, SYS_USERS_ 메타 테이블의 한 USER_ID 값과 동일하다.

PACKAGE_OID

패키지의 식별자로, SYS_PACKAGES_ 메타 테이블의 한 PACKAGE_OID 값과 동일하다.

RELATED_USER_ID

저장 프로시저가 접근하는 객체 소유자의 사용자 식별자로, SYS_USERS_ 메타 테이블의 한 USER_ID 값과 동일하다.

RELATED_OBJECT_NAME

저장 프로시저가 접근하는 객체의 이름이다.

RELATED_OBJECT_TYPE

저장 프로시저가 접근하는 객체의 타입을 나타낸다. 가능한 값은 다음과 같다.

- 0: 저장 프로시저
- 1: 저장 함수
- 2: 테이블, 시퀀스, 뷰
- 3: 타입세트
- 4: 데이터베이스 링크

참조 테이블

SYS_USERS_
SYS_PACKAGES_
SYS_TABLES_

SYS_PART_INDICES_

파티션드 인덱스를 관리하기 위한 메타 테이블이다. SYS_INDICES_의 IS_PARTITIONED가 'T'로 되어 있는 파티션드 인덱스에 대한 정보이다.

| Column name | Type | Description |
|-----------------|---------|-----------------|
| USER_ID | INTEGER | 사용자 식별자 |
| TABLE_ID | INTEGER | 테이블 식별자 |
| INDEX_ID | INTEGER | 인덱스 식별자 |
| PARTITION_TYPE | INTEGER | 파티션 타입 |
| IS_LOCAL_UNIQUE | CHAR(1) | 로컬 유니크 인덱스인지 여부 |

칼럼 정보**USER_ID**

인덱스 소유자의 사용자 식별자로, SYS_USERS_ 메타 테이블의 한 USER_ID 값과 동일하다.

TABLE_ID

인덱스를 생성한 테이블의 테이블 식별자로, SYS_TABLES_ 메타 테이블의 한 TABLE_ID 값과 동일하다.

INDEX_ID

인덱스 식별자로, SYS_INDICES_ 메타 테이블의 한 INDEX_ID 값과 동일하다.

PARTITION_TYPE

파티션 타입이 지역 (LOCAL)인지 글로벌 (GLOBAL)인지를 나타낸다. 그러나 현재 글로벌 파티션 타입을 지원하지 않으므로, 이 값은 항상 0이다.

- 0: LOCAL
- 1: GLOBAL

IS_LOCAL_UNIQUE

인덱스가 로컬 유니크 인덱스인지 여부를 가리키는 것으로, 'T' 또는 'F'이다.

- T: 로컬 유니크 인덱스이다.
- F: 로컬 유니크 인덱스가 아니다.

참조 테이블

```
SYS_USERS_
SYS_TABLES_
SYS_INDICES_
```

SYS_PART_KEY_COLUMNS_

파티션드 객체의 파티셔닝 키 칼럼에 대한 정보를 저장하는 메타 테이블이다.

| Column name | Type | Description |
|------------------|---------|----------------------------|
| USER_ID | INTEGER | 사용자 식별자 |
| PARTITION_OBJ_ID | INTEGER | 파티션드 객체 식별자 |
| COLUMN_ID | INTEGER | 칼럼 식별자 |
| OBJECT_TYPE | INTEGER | 객체 타입 |
| PART_COL_ORDER | INTEGER | 파티셔닝 키 내에서 칼럼의 위치 (0부터 시작) |

칼럼 정보**USER_ID**

인덱스 소유자의 사용자 식별자로, SYS_PART_INDICES_ 메타 테이블의 한 USER_ID 값과 동일하다.

PARTITION_OBJ_ID

파티션드 객체 식별자로, SYS_PART_TABLES_ 메타 테이블의 한 TABLE_ID 값 또는 SYS_PART_INDICES_ 메타 테이블의 한 INDEX_ID 값과 동일하다.

COLUMN_ID

인덱스를 생성한 테이블의 테이블 식별자로, SYS_COLUMNS_ 메타 테이블의 한 COLUMN_ID 값과 동일하다.

OBJECT_TYPE

객체 타입을 나타내는 식별자이다.

- 0: 테이블 (TABLE)
- 1: 인덱스 (INDEX)

PART_COL_ORDER

파티셔닝 키 내에서 칼럼의 위치를 나타낸다 (0부터 시작).

참조 테이블

SYS_PART_INDICES_
SYS_TABLE_PARTITIONS_
SYS_COLUMNS_

SYS_PART_LOBS_

파티션별로 LOB 칼럼을 관리하기 위한 메타 테이블이다.

| Column name | Type | Description |
|--------------|---------|-------------|
| USER_ID | INTEGER | 사용자 식별자 |
| TABLE_ID | INTEGER | 테이블 식별자 |
| PARTITION_ID | INTEGER | 파티션 식별자 |
| COLUMN_ID | INTEGER | 칼럼 식별자 |
| TBS_ID | INTEGER | 테이블스페이스 식별자 |
| LOGGING | CHAR(1) | 향후 확장 예정 |
| BUFFER | CHAR(1) | 향후 확장 예정 |

칼럼 정보**USER_ID**

LOB 칼럼이 속한 테이블 소유자의 사용자 식별자로, SYS_USERS_ 메타 테이블의 한 USER_ID 값과 동일하다.

TABLE_ID

LOB 칼럼이 속한 테이블의 식별자, SYS_TABLES_ 메타 테이블의 한 TABLE_ID 값과 동일하다.

PARTITION_ID

LOB 칼럼이 저장되는 파티션의 식별자이다.

COLUMN_ID

LOB 칼럼의 식별자이다.

TBS_ID

LOB 칼럼이 저장되는 테이블스페이스의 식별자이다.

참조 테이블

```
SYS_USERS_
SYS_TABLES_
SYS_PART_TABLES_
SYS_COLUMNS_
```

SYS_PART_TABLES_

파티션드 테이블을 관리하기 위한 메타 테이블이다. SYS_PART_TABLE_에 들어가는 테이블 정보는 SYS_TABLES_에서 IS_PARTITIONED가 'Y'로 되어 있는 파티션드 테이블에 대한 정보이다.

| Column name | Type | Description |
|---------------------|---------|--------------------------|
| USER_ID | INTEGER | 사용자 식별자 |
| TABLE_ID | INTEGER | 테이블 식별자 |
| PARTITION_METHOD | INTEGER | 파티셔닝 메소드 |
| PARTITION_KEY_COUNT | INTEGER | 파티션 키 칼럼의 개수 |
| ROW_MOVEMENT | CHAR(1) | 갱신된 레코드에 대한 파티션 이동 허용 여부 |

칼럼 정보**USER_ID**

인덱스 소유자의 사용자 식별자로, SYS_USERS_ 메타 테이블의 한 USER_ID 값과 동일하다.

TABLE_ID

인덱스를 생성한 테이블의 테이블 식별자로, SYS_TABLES_ 메타 테이블의 한 TABLE_ID 값과 동일하다.

PARTITION_METHOD

파티셔닝 메소드를 나타낸다.

- 0: 범위 (RANGE)
- 1: 해시 (HASH)
- 2: 리스트 (LIST)
- 3: 해시를 사용한 범위 파티셔닝 (RANGE PARTITIONING USING HASH)

ROW_MOVEMENT

파티션 키 칼럼의 값이 갱신 (UPDATE)될 때, 갱신된 레코드를 다른 파티션으로 이동할 것인지에 대한 허가 여부를 결정하는 것이다.

- T: 이동 허가
- F: 이동 불허가

참조 테이블

SYS_USERS_
SYS_TABLES_

SYS_PASSWORD_HISTORY_

패스워드 관리 정책을 설정한 사용자의 패스워드 변경 내역을 기록하는 메타 테이블이다.

| Column name | Type | Description |
|---------------|--------------|--------------|
| USER_ID | INTEGER | 사용자 식별자 |
| PASSWORD | VARCHAR(256) | 사용자 패스워드 |
| PASSWORD_DATE | DATE | 사용자 패스워드 변경일 |

SYS_PASSWORD_LIMITS_

사용자 생성 시 계정에 대해 지정한 패스워드 관리 정책과 계정의 현재 상태를 기록하는 메타 테이블이다.

| Column name | Type | Description |
|-----------------------|--------------|--|
| USER_ID | INTEGER | 사용자 식별자 |
| USER_NAME | VARCHAR(128) | 사용자 이름 |
| ACCOUNT_STATUS | VARCHAR(30) | 계정의 현재 상태를 표시 EXPIRED EXPIRED(GRACE) LOCKED(TIMED) LOCKED EXPIRED & LOCKED(TIMED) EXPIRED(GRACE) & LOCKED(TIMED) EXPIRED & LOCKED EXPIRED(GRACE) & LOCKED |
| REMAIN_GRACE_DAY | VARCHAR(10) | 패스워드 만료 후 남은 유예기간 |
| FAILED_LOGIN_ATTEMPTS | VARCHAR(10) | 로그인 실패 허용 최대 횟수 |
| PASSWORD_LOCK_TIME | VARCHAR(10) | 계정이 한 번 잠긴 후 다시 풀리기 위해 경과되어야 하는 기간 |
| PASSWORD_LIFE_TIME | VARCHAR(10) | 패스워드 유효기간 |
| PASSWORD_GRACE_TIME | VARCHAR(10) | 패스워드 만료 후 유예기간 |
| PASSWORD_REUSE_TIME | VARCHAR(10) | 동일한 패스워드가 재사용 가능해지기 위해 경과해야 하는 기간 |

| Column name | Type | Description |
|--------------------------|--------------|------------------------------------|
| PASSWORD_REUSE_MAX | VARCHAR(10) | 동일한 패스워드의 재사용 가능 횟수 |
| PASSWORD_VERIFY_FUNCTION | VARCHAR(128) | 패스워드를 검증할 콜백 함수(Callback Function) |

SYS_PRIVILEGES_

Altibase가 지원하는 권한의 종류 정보를 기록하는 메타 테이블이다. 권한에 대한 자세한 설명은 데이터베이스 권한 관리 또는 *SQL Reference*의 GRANT문 설명을 참조한다.

| Column name | Type | Description |
|-------------|--------------|-------------|
| PRIV_ID | INTEGER | 권한 식별자 |
| PRIV_TYPE | INTEGER | 권한 타입 |
| PRIV_NAME | VARCHAR(128) | 권한 이름 |

칼럼 정보

PRIV_ID

권한 식별자로 시스템이 내부적으로 정의한 값이다.

PRIV_TYPE

권한의 타입을 나타낸다.

- 1: 객체 권한
- 2: 시스템 권한

PRIV_NAME

권한의 이름이다.

SYS_PROCEDURES_

저장 프로시저와 저장 함수들에 대한 정보로 저장 프로시저 이름, 리턴 타입, 파라미터 개수, 실행 가능 여부 등을 기록하는 테이블이다.

| Column name | Type | Description |
|-------------|--------------|--|
| USER_ID | INTEGER | 저장 프로시저 소유자 식별자 |
| PROC_OID | BIGINT | 저장 프로시저 식별자 |
| PROC_NAME | VARCHAR(128) | 저장 프로시저 이름 |
| OBJECT_TYPE | INTEGER | 저장 프로시저, 저장 함수 또는 타입셋 인지를 나타냄 |
| STATUS | INTEGER | 객체의 상태를 나타낸다. INVALID이면 실행 불가능 상태이다. 0: VALID 1: INVALID |

| Column name | Type | Description |
|------------------|---------|---|
| AUTHID | INTEGER | 프로시저 또는 함수의 실행자 권한 - 0: 생성자 권한(DEFINER) - 1: 사용자 권한(CURRENT_USER) |
| PARA_NUM | INTEGER | 저장 프로시저 파라미터 개수 |
| RETURN_DATA_TYPE | INTEGER | 저장 함수의 리턴 데이터 타입 |
| RETURN_LANG_ID | INTEGER | 리턴 타입 언어 식별자 |
| RETURN_SIZE | INTEGER | 저장 함수의 리턴 데이터 타입의 크기 |
| RETURN_PRECISION | INTEGER | 저장 함수의 리턴 데이터 타입의 precision |
| RETURN_SCALE | INTEGER | 저장 함수의 리턴 데이터 타입의 scale |
| PARSE_NO | INTEGER | SYS_PROC_PARSE_에 구문의 조각들을 저장하고 있는 레코드의 개수 |
| PARSE_LEN | INTEGER | SYS_PROC_PARSE_에 저장된 구문의 전체 길이 |
| CREATED | DATE | 저장 프로시저를 생성한 날짜 |
| LAST_DDL_TIME | DATE | 저장 프로시저에 DDL 변경 작업이 마지막으로 일어난 시간 |

칼럼 정보

USER_ID

저장 프로시저 또는 저장 함수 소유자의 사용자 식별자로, SYS_USERS_ 메타 테이블의 한 USER_ID 값과 동일하다.

PROC_OID

저장 프로시저 또는 저장 함수의 식별자로, 시스템에 의해 자동으로 부여된다.

PROC_NAME

저장 프로시저 또는 저장 함수의 이름이다.

OBJECT_TYPE

저장 프로시저와 저장 함수를 구별하는 값이다. 저장 함수는 저장 프로시저와 달리 하나의 리턴 값을 가진다.

- 0: 저장 프로시저
- 1: 저장 함수
- 3: 타입 세트

STATUS

저장프로시저 또는 함수의 실행 가능 여부를 나타내는 값이다. 0 (VALID) 은 실행가능함을 나타낸다.

저장 프로시저 또는 저장 함수가 접근하는 객체에 DDL문을 수행하면, 관련 저장 프로시저 또는 저장 함수는 무효한 상태가 된다. 예를 들어 저장 프로시저가 접근하는 테이블에 새로운 칼럼이 추가되면 관련 저장 프로시저는 재 컴파일 후 VALID 상태가 되면 실행할 수 있다.

- 0: VALID
- 1: INVALID

AUTHID

프로시저 또는 함수를 실행하는 권한을 나타내는 값이다.

- 0: 생성자 권한(DEFINER)
- 1: 사용자 권한(CURRENT_USER)

PARA_NUM

저장 프로시저 또는 저장 함수에 정의된 파라미터 개수를 나타낸다.

RETURN_DATA_TYPE

저장 함수의 리턴값에 대한 데이터 타입의 식별자이다. 데이터 타입 식별자 값은 SYS_COLUMNS_ 메타 테이블의 DATA_TYPE 칼럼 설명을 참조한다.

데이터 타입에 대한 자세한 내용은 [General Reference-1.Data Types & Altibase Properties](#)를 참조한다.

RETURN_LANG_ID

타입 (CHAR, VARCHAR)의 언어 속성 정보를 나타내는 칼럼이다.

RETURN_SIZE

리턴 데이터 타입의 물리적 크기이다.

RETURN_PRECISION

리턴 데이터 타입의 정밀도 (precision)로, 사용자가 지정하거나 또는 시스템이 기본 값으로 부여한다. 타입의 경우 사용자가 정의한 타입의 길이이다.

RETURN_SCALE

리턴 데이터 타입의 scale로, 사용자가 지정하거나 또는 시스템이 기본 값으로 부여한다. 타입에 따라 이 값은 사용하지 않을 수 있다.

데이터 타입의 precision 과 scale에 대한 상세한 내용은 [General Reference-1.Data Types & Altibase Properties](#)를 참조한다.

PARSE_NO

저장 프로시저 또는 저장 함수 구문은 SYS_PROC_PARSE_ 메타 테이블에 나뉘져 여러 레코드로 저장되는데, 이 값은 저장하는 레코드의 수를 나타낸다.

PARSE_LEN

저장 프로시저 또는 저장 함수 구문은 SYS_PROC_PARSE_ 메타 테이블에 나뉘져 여러 레코드로 저장되는데 저장하는 전체 구문의 문자열 길이이다.

LAST_DDL_TIME

저장 프로시저에 DDL 변경 작업이 마지막으로 일어난 시간을 나타낸다.

참조 테이블

SYS_USERS_

SYS_PROC_PARAS_

저장 프로시저와 저장 함수들의 인자 (parameter)들에 대한 정보를 기록하는 테이블이다.

| Column name | Type | Description |
|-------------|---------------|-------------------------------|
| USER_ID | INTEGER | 저장 프로시저 소유자 식별자 |
| PROC_OID | BIGINT | 저장 프로시저 식별자 |
| PARA_NAME | VARCHAR(128) | 파라미터 이름 |
| PARA_ORDER | INTEGER | 파라미터의 순서. 첫번째 파라미터의 경우 1을 가짐. |
| INOUT_TYPE | INTEGER | 파라미터의 입력, 출력, 입출력 여부 |
| DATA_TYPE | INTEGER | 파라미터의 데이터 타입 |
| LANG_ID | INTEGER | 파라미터 타입 언어 식별자 |
| SIZE | INTEGER | 파라미터 타입의 크기 |
| PRECISION | INTEGER | 파라미터 타입의 precision |
| SCALE | INTEGER | 파라미터 타입의 scale |
| DEFAULT_VAL | VARCHAR(4000) | 파라미터의 기본 값 |

칼럼 정보**USER_ID**

저장 프로시저 또는 저장 함수 소유자의 사용자 식별자로, SYS_USERS_ 메타 테이블의 한 USER_ID 값과 동일하다.

PROC_OID

저장 프로시저 또는 저장 함수의 식별자로, SYS_PROCEDURES_ 메타 테이블의 한 PROC_OID 값과 동일하다.

PARA_NAME

파라미터의 이름이다.

PARA_ORDER

여러 파라미터들 중 해당 파라미터가 몇번째 정의된 파라미터인지를 나타내는 값이다.

INOUT_TYPE

저장 프로시저 또는 저장 함수의 파라미터가 입력인자, 출력인자, 또는 입출력인자인지를 나타낸다.

- 0: IN
- 1: OUT
- 2: IN OUT

DATA_TYPE

파라미터의 데이터 타입 식별자이다. 데이터 타입 식별자 값은 SYS_COLUMNS_ 메타 테이블의 DATA_TYPE 칼럼 설명을 참조한다.

데이터 타입에 대한 자세한 내용은 [General Reference-1.Data Types & Altibase Properties](#)를 참조한다.

LANG_ID

타입 (CHAR, VARCHAR)의 언어 속성 정보를 나타내는 칼럼이다.

SIZE

데이터 타입의 물리적 크기이다.

PRECISION

인자 데이터 타입의 정밀도 (precision)으로, 사용자가 지정하거나 또는 시스템이 기본 값으로 부여한다. 타입의 경우 사용자가 정의한 타입의 길이이다.

SCALE

인자 데이터 타입의 scale로, 사용자가 지정하거나 또는 시스템이 기본 값으로 부여한다. 타입에 따라 이 값은 사용하지 않을 수 있다.

데이터 타입의 precision 과 scale에 대한 상세한 내용은 [General Reference-1.Data Types & Altibase Properties](#)를 참조한다.

DEFAULT_VAL

파라미터 정의 시 사용자가 지정하는 파라미터 기본 값이다.

참조 테이블

```
SYS_USERS_
SYS_PROCEDURES_
```

SYS_PROC_PARSE_

사용자가 정의한 저장 프로시저와 저장 함수들의 구문 텍스트를 기록하는 테이블이다.

| Column name | Type | Description |
|-------------|---------|---------------------------|
| USER_ID | INTEGER | 저장 프로시저 또는 저장 함수의 소유자 식별자 |
| PROC_OID | BIGINT | 저장 프로시저 객체 식별자 |

| Column name | Type | Description |
|-------------|--------------|-------------------------------|
| SEQ_NO | INTEGER | 나뉘어 여러 레코드로 저장된 구문들 중 레코드의 순서 |
| PARSE | VARCHAR(100) | 나뉘어진 저장 프로시저 또는 저장 함수의 구문 |

칼럼 정보

USER_ID

저장 프로시저 소유자의 사용자 식별자로, SYS_USERS_ 메타 테이블의 한 USER_ID 값과 동일하다.

PROC_OID

저장 프로시저 또는 저장 함수의 식별자로, SYS_PROCEDURES_ 메타 테이블의 한 PROC_OID 값과 동일하다.

SEQ_NO

한 저장 프로시저의 구문 정보를 나누어서 SYS_PROC_PARSE_에 여러 개의 레코드로 저장할 때, 각 레코드의 순서를 나타낸다.

PARSE

저장 프로시저 또는 저장 함수 구문의 문자열의 조각이다. 한 PROC_OID 값으로 레코드들을 검색하여 SEQ_NO 순서대로 PARSE 값을 합치면 저장 프로시저 전체 구문을 생성할 수 있다.

참조 테이블

SYS_USERS_
SYS_PROCEDURES_

SYS_PROC_RELATED_

저장 프로시저와 저장 함수들이 접근하는 테이블, 시퀀스, 저장 프로시저, 저장 함수, 또는 뷰들에 대한 정보를 기록하는 테이블이다.

| Column name | Type | Description |
|---------------------|--------------|------------------------------|
| USER_ID | INTEGER | 저장 프로시저 소유자 식별자 |
| PROC_OID | BIGINT | 저장 프로시저 식별자 |
| RELATED_USER_ID | INTEGER | 저장 프로시저 내에서 참조하는 객체의 소유자 식별자 |
| RELATED_OBJECT_NAME | VARCHAR(128) | 저장 프로시저 내에서 참조하는 객체의 이름 |
| RELATED_OBJECT_TYPE | INTEGER | 저장 프로시저 내에서 참조하는 객체의 타입 |

저장 프로시저 PROC1이 테이블 t1에 INSERT 작업을 수행하는 경우, PROC1의 소유자 식별자와 저장 프로시저 식별자가 각각 USER_ID와 PROC_OID에 저장되고, 테이블 t1의 소유자 ID와 테이블 이름은 각각 RELATED_USER_ID, RELATED_OBJECT_NAME에 저장되며, RELATED_OBJECT_TYPE에는 2 (TABLE을 나타냄)가 저장된다.

칼럼 정보

USER_ID

저장 프로시저 또는 저장 함수 소유자의 사용자 식별자로, SYS_USERS_ 메타 테이블의 한 USER_ID 값과 동일하다.

PROC_OID

저장 프로시저 또는 저장 함수의 식별자로, SYS_PROCEDURES_ 메타 테이블의 한 PROC_OID 값과 동일하다.

RELATED_USER_ID

저장 프로시저가 접근하는 객체 소유자의 사용자 식별자로, SYS_USERS_ 메타 테이블의 한 USER_ID 값과 동일하다.

RELATED_OBJECT_NAME

저장 프로시저가 접근하는 객체의 이름이다.

RELATED_OBJECT_TYPE

저장 프로시저가 접근하는 객체의 타입을 나타낸다. 가능한 값은 다음과 같다.

- 0: 저장 프로시저
- 1: 저장 함수
- 2: 테이블, 시퀀스, 뷰
- 3: 타입세트
- 4: 데이터베이스 링크

참조 테이블

```
SYS_USERS_
SYS_PROCEDURES_
SYS_TABLES_
```

SYS_RECYCLEBIN_

휴지통에서 관리하는 테이블의 정보를 저장하는 메타 테이블이다. RECYCLEBIN_ENABLE 프로퍼티의 값이 1로 설정된 경우 DROP 구문으로 휴지통으로 이동하는 테이블이 저장된다.

| Column name | Type | Description |
|---------------------|--------------|--|
| USER_NAME | VARCHAR(128) | 테이블 소유자 |
| TABLE_NAME | VARCHAR(128) | 삭제될 때 휴지통에서 관리하기 위해 시스템에서 생성되는 테이블 이름. 동일한 이름의 테이블이 여러 번 삭제될 수 있으며, 휴지통에서 이를 관리하기 위해 테이블 이름을 별도로 생성한다. |
| ORIGINAL_TABLE_NAME | VARCHAR(128) | 삭제되기 전의 테이블의 이름. |
| TBS_NAME | VARCHAR(128) | 테이블이 저장된 테이블스페이스 이름 |

| Column name | Type | Description |
|-------------|--------|-------------------------------|
| MEMORY_SIZE | BIGINT | 삭제된 메모리 테이블이 메모리에서 차지하는 전체 크기 |
| DISK_SIZE | BIGINT | 삭제된 디스크 테이블이 디스크에서 차지하는 전체 크기 |
| DROPPED | DATE | 테이블이 삭제(Drop)된 시간 |

칼럼 정보

TABLE_NAME

테이블이 DROP 될 때 휴지통에서 관리하기 위해 시스템에서 생성하는 테이블 이름이다. 동일한 이름의 테이블(ORIGINAL_TABLE_NAME)이 여러 번 삭제될 경우 휴지통에서 테이블을 관리하기 위해서 새로운 이름이 생성된다.

SYS_REPLICATIONS_

이중화 관련 정보를 기록하고 있는 메타 테이블이다.

| Column name | Type | Description |
|--------------------------|-------------|---|
| LAST_USED_HOST_NO | INTEGER | 가장 최근에 사용한 원격 서버 |
| HOST_COUNT | INTEGER | 원격 서버 개수 |
| IS_STARTED | INTEGER | 이중화 시작 여부 |
| XSN | BIGINT | 송신자가 XLog 전송을 재개할 재시작 SN(Sequence Number) ¹³ |
| ITEM_COUNT | INTEGER | 이중화 대상 테이블 개수 |
| CONFLICT_RESOLUTION | INTEGER | 이중화 충돌 해결 방법 |
| REPL_MODE | INTEGER | 기본 이중화 모드 |
| ROLE | INTEGER | 이중화 스레드의 역할 |
| OPTIONS | INTEGER | 부가적인 이중화 기능을 위한 플래그 |
| INVALID_RECOVERY | INTEGER | 이중화 복구 가능 여부 |
| REMOTE_FAULT_DETECT_TIME | DATE | 원격 서버의 장애 감지 시각 |
| GIVE_UP_TIME | DATE | 가장 최근에 이중화를 포기한 일시 |
| REPLICATION_NAME | VARCHAR(40) | 이중화 이름 |
| GIVE_UP_XSN | BIGINT | 가장 최근에 이중화를 포기했을 시점의 XSN |
| PARALLEL_APPLIER_COUNT | INTEGER | 병렬 적용자(Applier)의 수 |
| REMOTE_XSN | BIGINT | 원격 서버에서 가장 최근에 처리한 SN |
| APPLIER_INIT_BUFFER_SIZE | BIGINT | applier buffer 의 초기 사이즈 |

| Column name | Type | Description |
|-----------------------|-------------|---------------------------|
| PEER_REPLICATION_NAME | VARCHAR(40) | 로컬 이중화한 원격 이중화 이름 |
| REMOTE_LAST_DDL_XSN | BIGINT | 원격 서버에서 가장 최근에 처리한 DDL SN |

[¹³] SN(Sequence Number): 로그 레코드의 식별 번호

칼럼 정보

REPLICATION_NAME

이중화 이름으로, 이중화 생성 시 사용자가 명시한다.

LAST_USED_HOST_NO

가장 최근에 사용한 원격 서버의 번호로, SYS_REPL_HOSTS_ 메타 테이블의 한 HOST_NO 값과 동일하다.

HOST_COUNT

이중화에 참여하는 원격 서버의 개수로, SYS_REPL_HOSTS_ 에 저장된 IP의 개수와 동일하다.

IS_STARTED

이중화 동작 여부를 나타낸다.

- 0: 중지
- 1: 이중화 수행 중

XSN

이중화가 시작될 때, 송신 쓰레드에서 로그 전송을 시작해야 할 SN을 나타낸다.

ITEM_COUNT

이중화 대상 테이블의 개수이다. 해당 이중화에 대해 SYS_REPL_ITEMS_ 메타 테이블에 이 수만큼 레코드들이 존재한다.

CONFLICT_RESOLUTION

이중화 충돌 해결 방법을 기록한다.

- 0: 기본 값
- 1: Master Server로 동작
- 2: Slave Server로 동작

이중화 충돌 해결 방법에 대한 자세한 설명은 *Replication Manual*을 참조한다.

REPL_MODE

이중화 생성시에 지정한 기본 이중화 모드이다.

- 0: LAZY MODE (기본 값)
- 2: EAGER MODE

기본 이중화 모드는 ALTER SESSION SET REPLICATION 구문으로 세션의 이중화 모드를 설정하지 않았을 때 사용된다.

기본 이중화 모드에 관한 자세한 내용은 *Replication Manual*을 참조하며, ALTER SESSION SET REPLICATION 구문에 관한 내용은 *SQL Reference*을 참조한다.

ROLE

이중화 쓰레드의 롤(ROLE)을 의미한다.

- 0: 일반 이중화 (롤을 지정하지 않은 경우)
- 1: Log Analyzer 전용 이중화 (FOR ANALYSIS 만 사용한 경우)
- 2: Propagable Logging (FOR PROPAGABLE LOGGING을 사용한 경우)
- 3: Propagation (FOR PROPAGATION 을 사용한 경우)
- 4: Log analyzer 용 이중화에서 Propagation 설정 한 경우(FOR ANALYSIS PROPAGATION을 사용한 경우)

Log Analyzer 전용 이중화에 대한 자세한 내용은 Log Analyzer User's Manual을 참고한다.

OPTIONS

이중화 부가 기능을 나타내는 플래그이다. 이중화 옵션의 종류는 아래와 같으며, 각 옵션을 설정시 이진수로 제어되며, 십진수로 변환되어 표시된다. 두 개 이상의 옵션을 사용할 경우 각각의 옵션에 해당하는 이진수 합이 십진수로 반환된다.

- 0(00000000): 이중화 옵션을 사용하지 않음
- 1(00000001): 복구 옵션 사용
- 2(00000010): 오프라인 옵션 사용
- 4(00000100): 이중화 갭 해소 옵션 사용
- 8(00001000): 병렬 적용자 옵션 사용
- 16(00010000):이중화 트랜잭션 그룹 옵션 사용
- 32(00100000):로컬 이중화 옵션 사용
- 64(01000000):메타 로깅 옵션 사용

INVALID_RECOVERY

이중화를 이용하여 복구가 가능한지 여부를 나타낸다.

- 0: 복구 가능 상태
- 1: 복구 불가능 상태

REMOTE_FAULT_DETECT_TIME

이중화 동작 중에 원격 서버의 장애를 감지한 시점을 기록한다.

GIVE_UP_TIME

이 값은 가장 최근에 이중화를 포기했을 시점의 일시이다. 즉, 이중화 송신 쓰레드가 이중화 전송을 포기한 시점이다.

GIVE_UP_XSN

이 값은 가장 최근에 이중화를 포기했을 시점의 XSN이다.

PARALLEL_APPLIER_COUNT

병렬 적용자의 수를 나타낸다.

REMOTE_XSN

원격 서버에서 가장 최근에 처리한 SN 이다. Sender 재시작 시 해당 REMOTE_XSN보다 SN이 작은 로그는 보내지 않고 Skip한다..

APPLIER_INIT_BUFFER_SIZE

병렬 적용자 옵션(receiver applier option)을 설정하여 이중화를 수행할 경우, 병렬 적용자의 초기 버퍼 크기이다. 병렬 적용자의 큐(queue) 개수는 해당 값을 XLog Size 로 나눈 값으로 설정된다.

```
( applier queue size = applier_init_buffer_size / xlog size )
```

만약 병렬 적용자 큐의 수가 프로퍼티 REPLICATION_RECEIVER_APPLIER_QUEUE_SIZE 값보다 작다면 병렬 적용자 큐의 수는 프로퍼티 REPLICATION_RECEIVER_APPLIER_QUEUE_SIZE에 지정된 값으로 설정된다.

PEER_REPLICATION_NAME

로컬 이중화 옵션을 사용했을 때 원격 이중화의 이름이다.

REMOTE_LAST_DDL_XSN

원격 서버에서 가장 최근에 처리한 DDL의 SN 이다.

예제

<예제> 다음은 생성된 이중화 rep1에 이중화 갭 해소 옵션과 병렬 적용자 옵션을 함께 사용할 때의 값을 반환한다.

```
isQL> alter replication rep1 set gapless enable;
Alter success.
isQL> alter replication rep1 set parallel 4;
Alter success.
isQL> select options from system_.sys_replications_;
OPTIONS
-----
12
1 row selected.
```

SYS_REPL_HOSTS_

원격 서버에 관련된 정보를 가진 메타 테이블이다

| Column name | Type | Description |
|------------------|-------------|-----------------|
| HOST_NO | INTEGER | 호스트 식별자 |
| REPLICATION_NAME | VARCHAR(40) | 이중화 이름 |
| HOST_IP | VARCHAR(64) | 원격 서버 IP 주소 |
| PORT_NO | INTEGER | 원격 서버 이중화 포트 번호 |

| Column name | Type | Description |
|-------------|-------------|----------------------------|
| CONN_TYPE | VARCHAR(20) | 원격 서버 접속 방법 |
| IB_LATENCY | VARCHAR(10) | rsocket의 RDMA_LATENCY 옵션 값 |

칼럼 정보

HOST_NO

원격 서버의 일련 번호로, 시스템 시퀀스에 의해 자동으로 부여된다.

REPLICATION_NAME

사용자가 명시한 이중화 이름으로 SYS_REPLICATIONS_ 메타 테이블에서도 확인할 수 있다.

HOST_IP

원격 서버의 IP 주소이다.

PORT_NO

원격 서버의 이중화 포트 번호를 기록한다.

CONN_TYPE

원격 서버의 접속 방법을 나타낸다.

- TCP
- Unix Domain
- InfiniBand(IB)

IB_LATENCY

인피니밴드를 사용할 경우 rsocket의 RDMA_LATENCY 옵션 값을 나타낸다. CONN_TYPE이 IB가 아닌 경우에 이 값은 N/A이다.

참조 테이블

SYS_REPLICATIONS_

SYS_REPL_ITEMS_

이중화 대상 테이블에 관련된 정보를 가진 메타 테이블이다.

| Column name | Type | Description |
|----------------------|--------------|----------------------|
| REPLICATION_NAME | VARCHAR(40) | 이중화 이름 |
| TABLE_OID | BIGINT | 테이블 객체 식별자 |
| LOCAL_USER_NAME | VARCHAR(128) | 지역 서버의 대상 테이블 소유자 이름 |
| LOCAL_TABLE_NAME | VARCHAR(128) | 지역 서버의 대상 테이블 이름 |
| LOCAL_PARTITION_NAME | VARCHAR(128) | 지역 서버의 파티션 이름 |

| Column name | Type | Description |
|-----------------------|---------------|----------------------|
| REMOTE_USER_NAME | VARCHAR(128) | 원격 서버의 대상 테이블 소유자 이름 |
| REMOTE_TABLE_NAME | VARCHAR(128) | 원격 서버의 대상 테이블 이름 |
| REMOTE_PARTITION_NAME | VARCHAR(128) | 원격 서버의 파티션 이름 |
| IS_PARTITION | CHAR(1) | 파티션드 테이블인지 여부 |
| INVALID_MAX_SN | BIGINT | 건너 뛴 로그의 최대 SN |
| CONDITION | VARCHAR(1000) | Deprecated |
| REPLICATION_UNIT | CHAR(1) | 이중화 단위 |

하나의 이중화 객체는 한 개 이상의 테이블들을 포함할 수 있으며, 이들 테이블 각각에 대해 SYS_REPL_ITEMS_에 레코드가 존재한다. 예를 들어 한 이중화가 10개의 테이블을 가지고 있다면, 이 이중화에 대한 총 10개의 레코드가 이 메타 테이블에 기록된다.

칼럼 정보

REPLICATION_NAME

사용자가 명시한 이중화 이름으로 SYS_REPLICATIONS_ 메타 테이블에서도 확인할 수 있다.

TABLE_OID

이중화 대상 테이블 또는 파티션의 식별자로, SYS_TABLES_ 메타 테이블의 한 TABLE_OID 값 또는 SYS_TABLES_PARTITIONS_의 한 PARTITION_OID 값과 동일하다.

LOCAL_USER_NAME

지역 서버의 이중화 대상 테이블 소유자의 사용자 이름으로, SYS_USERS_ 메타 테이블의 한 USER_NAME 값과 동일하다.

LOCAL_TABLE_NAME

지역 서버의 이중화 대상 테이블의 이름으로, SYS_TABLES_ 메타 테이블의 한 TABLE_NAME 값과 동일하다.

LOCAL_PARTITION_NAME

지역 서버의 이중화 대상 파티션의 이름이다.

REMOTE_USER_NAME

원격 서버의 이중화 대상 테이블 소유자의 사용자 이름으로, 원격 서버의 SYS_USERS_ 메타 테이블의 한 USER_NAME 값과 동일하다.

REMOTE_TABLE_NAME

원격 서버의 이중화 대상 테이블의 이름으로, 원격 서버의 SYS_TABLES_ 메타 테이블의 한 TABLE_NAME 값과 동일하다.

REMOTE_PARTITION_NAME

원격 서버의 이중화 대상 파티션의 이름이다.

IS_PARTITION

테이블이 파티션드 테이블인지를 나타낸다. 'Y'는 파티션드 테이블이고, 'N'은 파티션드 테이블이 아니다.

INVALID_MAX_SN

이중화 대상 테이블에 DDL구문 또는 동기화 작업이 수행되는 시점에서 가장 최근에 기록된 SN이 저장된다. 해당 SN까지의 테이블 로그를 이중화에서 건너뛰는다.

REPLICATION_UNIT

이중화 대상 아이템이 무엇인지를 나타낸다. 이 칼럼에는 아래 두 개의 값 중에서 하나가 표시된다.

- T: 이중화 대상 아이템이 테이블임을 나타낸다.
- P: 이중화 대상 아이템이 파티션임을 나타낸다.

참조 테이블

```
SYS_REPLICATIONS_
SYS_USERS_
SYS_TABLES_
```

SYS_REPL_OFFLINE_DIR_

이중화 오프라인 옵션과 관련된 로그 디렉터리 정보를 가지는 메타 테이블이다.

| Column name | Type | Description |
|------------------|--------------|---------------|
| REPLICATION_NAME | VARCHAR(40) | 이중화 이름 |
| LFG_ID | INTEGER | 로그 파일 그룹의 식별자 |
| PATH | VARCHAR(512) | 오프라인 로그 경로 |

칼럼 정보**REPLICATION_NAME**

사용자가 명시한 이중화 이름으로 SYS_REPLICATIONS_ 메타 테이블에서도 확인할 수 있다.

LFG_ID

이는 0의 값을 가진 로그파일 그룹 고유번호이다.

PATH

로그 파일이 저장되는 시스템 내의 절대 경로를 나타낸다.

SYS_REPL_OLD_CHECKS_

이중화 송신 스레드가 복제중인 이중화 대상 칼럼 중 CHECK 제약조건에 대한 정보를 가진 메타 테이블이다.

| Column name | Type | Description |
|------------------|---------------|--------------------|
| REPLICATION_NAME | VARCHAR(40) | 이중화 이름 |
| TABLE_OID | BIGINT | 테이블 객체 식별자 |
| CONSTRAINT_ID | INTEGER | CHECK 제약조건 식별자 |
| CHECK_NAME | VARCHAR(40) | CHECK 제약조건 이름 |
| CONDITION | VARCHAR(4000) | CHECK 제약조건의 조건 문자열 |

칼럼 정보

REPLICATION_NAME

사용자가 명시한 이중화 이름으로 SYS_REPLICATIONS_ 메타 테이블에서도 확인할 수 있다.

TABLE_OID

이중화 송신 스레드가 처리 중인 테이블 객체 식별자이다. 이중화 송신 스레드가 이중화 로그를 처리 중인 시점에 이 테이블이 존재하지 않는다면 SYS_TABLES_ 메타 테이블에서 조회할 수 없다.

CONSTRAINT_ID

이중화 송신 스레드가 처리 중인 CHECK 제약조건 식별자로 SYS_CONSTRAINTS_ 메타 테이블에서 같은 칼럼으로 확인할 수 있다.

이중화 송신 스레드가 이중화 로그를 처리 중인 시점에 해당 CHECK 제약조건이 삭제된 경우 SYS_CONSTRAINTS_에서 조회할 수 없다.

CHECK_NAME

이중화 송신 스레드가 현재 사용 중인 CHECK 제약조건 이름으로 SYS_CONSTRAINTS_ 메타 테이블의 CONSTRAINT_NAME과 일치한다.

이중화 송신 스레드가 이중화 로그를 처리 중인 시점에 해당 CHECK 제약조건이 삭제된 경우 SYS_CONSTRAINTS_에서 조회할 수 없다.

CONDITION

이중화 송신 스레드가 현재 사용 중인 CHECK 제약조건의 조건 문자열로 SYS_CONSTRAINTS_ 메타 테이블의 CHECK_CONDITION과 일치한다.

이중화 송신 스레드가 이중화 로그를 처리 중인 시점에 해당 CHECK 제약조건이 삭제된 경우 SYS_CONSTRAINTS_에서 조회할 수 없다.

참조 테이블

```
SYS_REPLICATIONS_
SYS_TABLES_
SYS_CONSTRAINTS_
```

SYS_REPL_OLD_CHECK_COLUMNS_

이중화 송신 스레드가 복제 중인 이중화 대상 칼럼에 설정된 CHECK 제약조건에 대한 정보를 가진 메타 테이블이다.

| Column name | Type | Description |
|------------------|-------------|-----------------------|
| REPLICATION_NAME | VARCHAR(40) | 이중화 이름 |
| TABLE_OID | BIGINT | 테이블 객체 식별자 |
| CONSTRAINT_ID | INTEGER | CHECK 제약조건 식별자 |
| COLUMN_ID | INTEGER | CHECK 제약조건을 갖는 칼럼 식별자 |

칼럼 정보

REPLICATION_NAME

사용자가 명시한 이중화 이름으로 SYS_REPLICATIONS_ 메타 테이블에서도 확인할 수 있다.

TABLE_OID

이중화 송신 스레드가 처리 중인 테이블 객체 식별자이다. 이중화 송신 스레드가 이중화 로그를 처리 중인 시점에 이 테이블이 존재하지 않는다면 SYS_TABLES_ 메타 테이블에서 조회할 수 없다.

CONSTRAINT_ID

이중화 송신 스레드가 처리 중인 CHECK 제약 조건 식별자로 SYS_CONSTRAINTS_ 메타 테이블의 CONSTRAINT_ID와 일치한다.

이중화 송신 스레드가 이중화 로그를 처리 중인 시점에 해당 CHECK 제약조건이 삭제된 경우 SYS_CONSTRAINTS_에서 조회할 수 없다.

COLUMN_ID

이중화 송신 스레드가 처리 중인 CHECK 제약조건을 갖는 칼럼 식별자로 SYS_COLUMNS_ 메타 테이블의 COLUMN_ID와 일치한다.

이중화 송신 스레드가 이중화 로그를 처리 중인 시점에 해당 CHECK 제약조건이 삭제된 경우 SYS_COLUMNS_에서 조회할 수 없다.

참조 테이블

```
SYS_REPLICATIONS_
SYS_TABLES_
SYS_CONSTRAINTS_
SYS_COLUMNS_
```

SYS_REPL_OLD_COLUMNS_

이중화 송신 스레드가 현재 복제 중인 이중화 대상 칼럼의 정보를 가진 메타 테이블이다.

| Column name | Type | Description |
|------------------|-------------|-------------|
| REPLICATION_NAME | VARCHAR(40) | 이중화 이름 |

| Column name | Type | Description |
|----------------------|---------------|---|
| TABLE_OID | BIGINT | 테이블 객체 식별자 |
| COLUMN_NAME | VARCHAR(128) | 칼럼 이름 |
| MT_DATATYPE_ID | INTEGER | 데이터 타입 식별자 |
| MT_LANGUAGE_ID | INTEGER | 언어 식별자 |
| MT_FLAG | INTEGER | 내부 플래그 |
| MT_PRECISION | INTEGER | 정밀도 |
| MT_SCALE | INTEGER | 소수 자릿수 |
| MT_ENCRYPT_PRECISION | INTEGER | 암호화 칼럼 정밀도 |
| MT_POLICY_NAME | VARCHAR(16) | 암호화 칼럼에 사용된 정책의 이름 |
| SM_ID | INTEGER | 칼럼 식별자 |
| SM_FLAG | INTEGER | 내부 플래그 |
| SM_OFFSET | INTEGER | 내부 오프셋 |
| SM_VARORDER | INTEGER | 한 테이블 내에서 가변(Variable) 방식으로 저장된 칼럼의 저장 순서. 예외적으로 공간 데이터형은 VARORDER가 부여되지 않는다 (기본값 0). |
| SM_SIZE | INTEGER | 내부 크기 |
| SM_DIC_TABLE_OID | BIGINT | 압축 칼럼의 경우 디셔너리 테이블의 OID |
| SM_COL_SPACE | INTEGER | 테이블스페이스 식별자 |
| QP_FLAG | INTEGER | 내부 플래그 |
| DEFAULT_VAL | VARCHAR(4000) | 칼럼의 기본 값 |
| MT_SRID | INTEGER | GEOMETRY 칼럼에 적용된 SRID |

칼럼 정보

REPLICATION_NAME

사용자가 명시한 이중화 이름으로 SYS_REPLICATIONS_ 메타 테이블에서도 확인할 수 있다.

TABLE_OID

이중화 송신 스레드가 처리 중인 테이블 객체 식별자이다. 이중화 송신 스레드가 이중화 로그를 처리 중인 시점에 이 테이블이 존재하지 않는다면 SYS_TABLES_ 메타 테이블에서 조회할 수 없다.

COLUMN_NAME

이중화 송신 쓰레드가 현재 복제중인 이중화 대상 칼럼의 이름이다.

MT_DATATYPE_ID

데이터 타입 식별자로, 내부 값이다.

MT_LANGUAGE_ID

언어 식별자로, 내부 값이다.

MT_FLAG

Altibase 서버가 사용하는 내부 플래그이다.

MT_PRECISION

숫자 타입의 경우, 칼럼의 정밀도 (숫자 자리수)를 나타낸다. 타입의 경우, 문자형 데이터 타입의 길이를 나타낸다.

MT_SCALE

숫자 타입의 경우, 칼럼의 소수점 이하 자릿수를 나타낸다.

MT_ENCRYPT_PRECISION

암호화된 칼럼의 정밀도 (크기)를 나타낸다.

MT_POLICY_NAME

암호화된 칼럼의 경우, 칼럼에 적용된 보안 정책의 이름을 나타낸다.

MT_SRID

GEOMETRY 칼럼의 경우, 칼럼에 적용된 SRID를 나타낸다.

SM_ID

칼럼 식별자이다. 0부터 시작한다.

SM_FLAG

Altibase 서버가 사용하는 내부 플래그이다.

SM_OFFSET

Altibase 서버가 사용하는 내부 오프셋이다.

SM_VARORDER

한 테이블 내에서 가변(Variable) 방식으로 저장된 칼럼들 중 해당 칼럼이 저장되는 순서를 나타낸다. 예외적으로 공간 데이터형(GEOMETRY)은 VARORDER가 부여되지 않는다(기본값 0).

SM_SIZE

Altibase 서버가 사용하는 내부 크기이다.

SM_DIC_TABLE_OID

해당 칼럼이 압축 칼럼일 경우 압축 칼럼의 데이터가 실제로 저장되어 있는 디렉터리 테이블의 OID를 나타낸다.

SM_COL_SPACE

해당 칼럼의 데이터가 저장되는 테이블스페이스의 식별자이다.

QP_FLAG

Altibase 서버가 내부적으로 사용하는 플래그이다.

DEFAULT_VAL

칼럼의 기본값이 문자열로 저장된다. Altibase 서버가 내부적으로 사용한다.

참조 테이블

```
SYS_REPL_OLD_ITEMS_
SYS_REPL_OLD_INDICES_
SYS_REPL_OLD_INDEX_COLUMNS_
```

SYS_REPL_OLD_INDEX_COLUMNS_

이중화 송신 쓰레드가 현재 사용 중인 이중화 대상 인덱스 칼럼의 정보를 가진 메타 테이블이다.

| Column name | Type | Description |
|------------------|-------------|---------------|
| REPLICATION_NAME | VARCHAR(40) | 이중화 이름 |
| TABLE_OID | BIGINT | 테이블 객체 식별자 |
| INDEX_ID | INTEGER | 인덱스 식별자 |
| KEY_COLUMN_ID | INTEGER | 칼럼 식별자 |
| KEY_COLUMN_FLAG | INTEGER | 내부 플래그 |
| COMPOSITE_ORDER | INTEGER | 인덱스에서의 칼럼의 위치 |

칼럼 정보**REPLICATION_NAME**

사용자가 명시한 이중화 이름으로 SYS_REPLICATIONS_ 메타 테이블에서도 확인할 수 있다.

TABLE_OID

이중화 송신 쓰레드가 처리 중인 테이블 객체 식별자이다. 이중화 송신 쓰레드가 이중화 로그를 처리 중인 시점에 이 테이블이 존재하지 않는다면 SYS_TABLES_ 메타 테이블에서 조회할 수 없다.

INDEX_ID

이중화 송신 쓰레드가 현재 복제 중인 이중화 대상 인덱스의 식별자이다.

KEY_COLUMN_ID

인덱스를 구성하는 칼럼의 식별자이다.

KEY_COLUMN_FLAG

인덱스를 구성하는 칼럼의 내부 플래그이다.

COMPOSITE_ORDER

인덱스를 구성하는 칼럼의 순서이다.

참조 테이블

SYS_REPL_OLD_ITEMS_
SYS_REPL_OLD_COLUMNS_
SYS_REPL_OLD_INDICES_

SYS_REPL_OLD_INDICES_

이중화 송신 스레드가 현재 복제 중인 이중화 대상 인덱스의 정보를 가진 메타 테이블이다.

| Column name | Type | Description |
|------------------|--------------|----------------|
| REPLICATION_NAME | VARCHAR(40) | 이중화 이름 |
| TABLE_OID | BIGINT | 테이블 객체 식별자 |
| INDEX_ID | INTEGER | 인덱스 식별자 |
| INDEX_NAME | VARCHAR(128) | 인덱스 이름 |
| TYPE_ID | INTEGER | 인덱스 타입 식별자 |
| IS_UNIQUE | CHAR(1) | 글로벌 유니크 인덱스 여부 |
| IS_LOCAL_UNIQUE | CHAR(1) | 로컬 유니크 인덱스 여부 |
| IS_RANGE | CHAR(1) | 범위 검색 가능 여부 |

칼럼 정보**REPLICATION_NAME**

사용자가 명시한 이중화 이름으로 SYS_REPLICATIONS_ 메타 테이블에서도 확인할 수 있다.

TABLE_OID

이중화 송신 스레드가 처리 중인 테이블 객체 식별자이다. 이중화 송신 스레드가 이중화 로그를 처리 중인 시점에 이 테이블이 존재하지 않는다면 SYS_TABLES_ 메타 테이블에서 조회할 수 없다.

INDEX_ID

이중화 송신 스레드가 현재 복제 중인 이중화 대상 인덱스의 식별자이다.

INDEX_NAME

이중화 송신 쓰레드가 현재 복제 중인 이중화 대상 인덱스의 이름이다.

TYPE_ID

인덱스 유형 식별자로, 내부 값이다.

IS_UNIQUE

글로벌 유니크 인덱스인지 여부를 나타낸다. 'Y'는 글로벌 유니크를 나타내고, 'N'은 글로벌 유니크가 아님을 나타낸다.

IS_LOCAL_UNIQUE

로컬 유니크 인덱스인지 여부를 나타낸다. 'Y'는 로컬 유니크를 나타내고, 'N'은 로컬 유니크가 아님을 나타낸다.

IS_RANGE

범위 검색 가능 여부를 나타낸다. 'Y'는 범위 검색이 가능한 인덱스이고, 'N'은 범위 검색이 불가능한 인덱스임을 나타낸다.

참조 테이블

SYS_REPL_OLD_ITEMS_
SYS_REPL_OLD_COLUMNS_
SYS_REPL_OLD_INDEX_COLUMNS_

SYS_REPL_OLD_ITEMS_

이중화 송신 쓰레드가 현재 복제 중인 이중화 대상 테이블의 정보를 가진 메타 테이블이다.

| Column name | Type | Description |
|-----------------------|---------------|-------------------------------|
| REPLICATION_NAME | VARCHAR(40) | 이중화 이름 |
| TABLE_OID | BIGINT | 테이블 객체 식별자 |
| USER_NAME | VARCHAR(128) | 사용자 이름 |
| TABLE_NAME | VARCHAR(128) | 테이블 이름 |
| PARTITION_NAME | VARCHAR(128) | 파티션 이름 |
| PRIMARY_KEY_INDEX_ID | INTEGER | 프라이머리 키의 인덱스 식별자 |
| REMOTE_USER_NAME | VARCHAR(128) | 원격 서버의 대상 테이블 소유자 이름 |
| REMOTE_TABLE_NAME | VARCHAR(128) | 원격 서버의 대상 테이블 이름 |
| REMOTE_PARTITION_NAME | VARCHAR(128) | 원격 서버의 파티션 이름 |
| PARTITION_ORDER | INTEGER | 파티션 순서(해쉬 파티션일 경우 필요) |
| PARTITION_MIN_VALUE | VARCHAR(4000) | 파티션의 최소 기준값 (해쉬 파티션의 경우 NULL) |

| Column name | Type | Description |
|---------------------|---------------|-------------------------------|
| PARTITION_MAX_VALUE | VARCHAR(4000) | 파티션의 최대 기준값 (해쉬 파티션의 경우 NULL) |
| INVALID_MAX_SN | BIGINT | 건너 뛴 로그의 최대 SN |

칼럼 정보

REPLICATION_NAME

사용자가 명시한 이중화 이름으로 SYS_REPLICATIONS_ 메타 테이블에서도 확인할 수 있다.

TABLE_OID

이중화 송신 스레드가 처리 중인 테이블 객체 식별자이다. 이중화 송신 스레드가 이중화 로그를 처리 중인 시점에 이 테이블이 존재하지 않는다면 SYS_TABLES_ 메타 테이블에서 조회할 수 없다.

USER_NAME

지역 서버의 이중화 대상 테이블인 소유자의 이름이다. SYS_USERS_ 메타 테이블의 USER_NAME 값과 동일하다.

TABLE_NAME

지역 서버의 이중화 대상 테이블의 이름이다. SYS_TABLES_ 메타 테이블의 한 TABLE_NAME 값과 동일하다.

PARTITION_NAME

지역 서버의 이중화 대상 테이블이 속해 있는 파티션의 이름이다.

PRIMARY_KEY_INDEX_ID

프라이머리 키 (Primary Key)의 인덱스 식별자이다.

REMOTE_USER_NAME

원격 서버의 이중화 대상 테이블인 소유자의 이름이다.

REMOTE_TABLE_NAME

원격 서버의 이중화 대상 테이블의 이름이다.

REMOTE_PARTITION_NAME

원격 서버의 이중화 대상 테이블이 속해 있는 파티션의 이름이다.

PARTITION_ORDER

파티션들 중에서 이 파티션의 순서를 나타낸다. 해쉬 (HASH) 파티션인 경우에 필요하다.

PARTITION_MIN_VALUE

파티션의 최소 기준값을 문자열로 보여준다. 해쉬 (HASH) 파티션인 경우에는 널(NULL)이다.

PARTITION_MAX_VALUE

파티션의 최대 기준값을 문자열로 보여준다. 해쉬 (HASH) 파티션인 경우에는 널(NULL)이다.

INVALID_MAX_SN

이중화 대상 테이블에 DDL구문 또는 동기화 작업이 수행되는 시점에서 가장 최근에 기록된 SN이 저장된다. 해당 SN까지의 테이블 로그를 이중화에서 건너뛰다.

참조 테이블

```
SYS_REPL_OLD_COLUMNS_
SYS_REPL_OLD_INDICES_
SYS_REPL_OLD_INDEX_COLUMNS_
```

SYS_REPL_TABLE_OID_IN_USE_

이중화가 아직 처리하지 않은 DDL 로그에 포함된 테이블의 테이블 객체 식별자(TABLE OID) 정보를 관리하는 메타 테이블이다.

| Column name | Type | Description |
|------------------|--------------|---------------------|
| REPLICATION_NAME | VARCHAR(40) | 이중화 이름 |
| OLD_TABLE_OID | BIGINTBIGINT | DDL 수행 전 테이블 객체 식별자 |
| TABLE_OID | BIGINTBIGINT | 현재 테이블 객체 식별자 |

칼럼 정보**REPLICATION_NAME**

사용자가 명시한 이중화 이름으로 SYS_REPLICATIONS_ 메타 테이블에서도 확인할 수 있다.

OLD_TABLE_OID

이중화가 아직 처리하지 않은 DDL 로그에 포함된 테이블의 이전 테이블 객체 식별자이다.

TABLE_OID

이중화가 아직 처리하지 않은 DDL 로그에 포함된 테이블의 현재 테이블 객체 식별자이다. 이 값은 SYS_REPL_ITEMS_ 메타 테이블의 한 TABLE_OID 값과 동일하다.

SYS_REPL_RECOVERY_INFOS_

원격 서버의 복구에 사용하기 위해 로그 정보를 기록하는 메타 테이블이다.

| Column name | Type | Description |
|---------------------|-------------|-------------------|
| REPLICATION_NAME | VARCHAR(40) | 이중화 이름 |
| MASTER_BEGIN_SN | BIGINT | 주 트랜잭션의 시작 로그 번호 |
| MASTER_COMMIT_SN | BIGINT | 주 트랜잭션의 완료 로그 번호 |
| REPLICATED_BEGIN_SN | BIGINT | 복제 트랜잭션의 시작 로그 번호 |

| Column name | Type | Description |
|----------------------|--------|-------------------|
| REPLICATED_COMMIT_SN | BIGINT | 복제 트랜잭션의 완료 로그 번호 |

칼럼 정보

REPLICATION_NAME

사용자가 명시한 이중화 이름으로 SYS_REPLICATIONS_ 메타 테이블에서도 확인할 수 있다.

MASTER_BEGIN_SN

원격 서버에서 발생한 주 트랜잭션의 시작 로그 번호이다.

MASTER_COMMIT_SN

원격 서버에서 발생한 주 트랜잭션의 완료 로그 번호이다.

REPLICATED_BEGIN_SN

지역 서버에서 발생한 복제 트랜잭션의 시작 로그 번호이다.

REPLICATED_COMMIT_SN

지역 서버에서 발생한 복제 트랜잭션의 완료 로그 번호이다.

참조 테이블

SYS_REPLICATIONS_

SYS_SECURITY_

보안 모듈의 상태 정보를 관리한다.

| Column name | Type | Description |
|-----------------|-------------|------------------|
| MODULE_NAME | VARCHAR(24) | 보안 모듈의 이름 |
| MODULE_VERSION | VARCHAR(40) | 보안 모듈의 버전 |
| ECC_POLICY_NAME | VARCHAR(16) | ECC 정책의 이름 |
| ECC_POLICY_CODE | VARCHAR(64) | ECC 정책에 대한 검증 코드 |

이 테이블은 써드 파티에서 제공한 보안 모듈의 연동 여부를 보여준다.

써드 파티에서 제공한 보안 모듈이 정상적으로 연동되어 있는 경우, SYS_SECURITY_ 메타 테이블은 보안 모듈 프로퍼티들에 대한 정보를 저장한다. 반면, 보안 모듈이 연동되어 있지 않은 경우에는 SYS_SECURITY_ 메타 테이블에는 어떤 레코드도 존재하지 않는다.

SYS_SYNONYMS_

데이터베이스 객체에 대한 별칭 기능을 하는 시노ним에 대한 정보를 기록하는 테이블이다.

| Column name | Type | Description |
|------------------|---------|-------------|
| SYNONYM_OWNER_ID | INTEGER | 사용자 식별자 |

| Column name | Type | Description |
|-------------------|--------------|---------------------------------|
| SYNONYM_NAME | VARCHAR(128) | 시노님 이름 |
| OBJECT_OWNER_NAME | VARCHAR(128) | 객체 소유자 이름 |
| OBJECT_NAME | VARCHAR(128) | 시노님 대상 객체 이름 |
| CREATED | DATE | 시노님이 생성된 시간 |
| LAST_DDL_TIME | DATE | 시노님에 대해 마지막으로 DDL 변경 작업이 일어난 시간 |

칼럼 정보

SYNONYM_OWNER_ID

시노님 소유자의 사용자 식별자로, SYS_USERS_ 메타 테이블의 한 USER_ID 값과 동일하다.

SYNONYM_NAME

사용자가 명시한 시노님 이름이다.

OBJECT_OWNER_NAME

시노님 대상 객체가 소속된 스키마 소유자의 이름이다.

OBJECT_NAME

사용자가 명시한 시노님 대상 객체의 이름이다.

CREATED

시노님이 생성된 시간을 나타낸다.

LAST_DDL_TIME

시노님에 대해 DDL 변경 작업이 마지막으로 일어난 시간을 나타낸다.

참조 테이블

SYS_USERS_

SYS_TABLES_

메타 테이블들과 사용자가 정의한 테이블, 시퀀스 그리고 뷰에 대한 정보를 기록하는 테이블이다.

| Column name | Type | Description |
|--------------|--------------|-------------|
| USER_ID | INTEGER | 사용자 식별자 |
| TABLE_ID | INTEGER | 테이블 식별자 |
| TABLE_OID | BIGINT | 테이블 객체 식별자 |
| COLUMN_COUNT | INTEGER | 테이블 칼럼의 개수 |
| TABLE_NAME | VARCHAR(128) | 테이블 이름 |

| Column name | Type | Description |
|----------------------------|--------------|---|
| TABLE_TYPE | CHAR(1) | 객체 타입 |
| REPLICATION_COUNT | INTEGER | 테이블과 관련된 이중화 개수 |
| REPLICATION_RECOVERY_COUNT | INTEGER | 테이블과 관련된 복구 옵션을 사용하는 이중화 개수 |
| MAXROW | BIGINT | 입력할 수 있는 최대 레코드 개수(0: 제한 없음) |
| TBS_ID | INTEGER | 테이블스페이스 식별자 |
| TBS_NAME | VARCHAR(128) | 테이블이 저장된 테이블스페이스 이름 |
| PCTFREE | INTEGER | 아래 설명 참조 |
| PCTUSED | INTEGER | 아래 설명 참조 |
| INIT_TRANS | INTEGER | 한 페이지에서 동시에 갱신 가능한 트랜잭션의 초기 개수 |
| MAX_TRANS | INTEGER | 한 페이지에서 동시에 갱신 가능한 트랜잭션의 최대 개수 |
| INITEXTENTS | BIGINT | 테이블 생성시 초기 익스텐트 개수 |
| NEXTTEXTENTS | BIGINT | 테이블 확장시 추가될 익스텐트 개수 |
| MINEXTENTS | BIGINT | 테이블의 최소 익스텐트 개수 |
| MAXEXTENTS | BIGINT | 테이블의 최대 익스텐트 개수 |
| IS_PARTITIONED | CHAR(1) | 파티션드 테이블 여부 |
| TEMPORARY | CHAR(1) | 임시 테이블 여부 D: 트랜잭션에 한정되는 임시 테이블 P: 세션에 한정되는 임시 테이블 N: 임시 테이블이 아님 |
| HIDDEN | CHAR(1) | 숨김 속성을 갖는 테이블인지 여부 |
| ACCESS | CHAR(1) | 테이블 접근 모드 |
| PARALLEL_DEGREE | INTEGER | 병렬 질의를 처리하는 스레드의 개수 |
| CREATED | DATE | 테이블이 생성된 시간 |
| LAST_DDL_TIME | DATE | 테이블에 대해 마지막으로 DDL 변경 작업이 일어난 시간 |

칼럼 정보

USER_ID

테이블 소유자의 사용자 식별자로, SYS_USERS_ 메타 테이블의 한 USER_ID 값과 동일하다.

TABLE_ID

테이블 식별자로, 시스템 시퀀스에 의해 자동으로 부여된다.

TABLE_OID

시스템 내부에서 자동으로 부여되는 테이블 객체 식별자이다. 사용자가 메타 테이블 조회 시 사용하는 TABLE_ID와는 달리 시스템 내부 동작 시에만 사용된다.

COLUMN_COUNT

테이블에 정의된 칼럼의 개수이다.

TABLE_NAME

사용자가 명시한 테이블 이름이다.

TABLE_TYPE

SYS_TABLES_ 메타 테이블에는 테이블 외에 시퀀스, 뷰 정보 등도 함께 저장된다. 타입 식별자는 이들 객체를 구별하며, 아래의 타입 식별자로 표시된다.

- T: 테이블
- S: 시퀀스
- V: 뷰
- W: 큐(Queue) 전용 시퀀스
- Q: 큐
- M: Materialized view의 데이터 유지를 위해 자동으로 생성되는 테이블
- A: Materialized view의 데이터 유지를 위해 자동으로 생성되는 뷰
- G: 글로벌 인덱스를 위해 내부적으로 사용되는 테이블
- D: 압축 칼럼의 데이터를 실제로 저장하기 위해 내부적으로 사용되는 디렉터리 테이블
- R: 삭제(Drop)되어 휴지통에서 관리되고 있는 테이블

REPLICATION_COUNT

해당 테이블과 관련된 이중화 객체의 개수이다.

REPLICATION_RECOVERY_COUNT

해당 테이블에 대해 복구 옵션을 사용하는 이중화 객체의 개수이다.

MAXROW

테이블에 삽입가능한 최대 레코드 수이다.

TBS_ID

테이블이 저장되는 테이블스페이스의 식별자이다.

PCTFREE

한 페이지가 갱신 가능하기 위해 유지해야 하는 여유 공간의 최소 비율이다. 기존에 페이지에 저장된 행들을 갱신하기 위해 PCTFREE에서 명시한 비율만큼의 여유 공간을 페이지에서 유지하고 있다. 예를 들어 PCTFREE 값이 20이면, 한 페이지의 20%의 공간은 갱신 연산을 위해 남겨두고, 80%의 공간에 대해서만 데이터 삽입이 가능하다.

PCTFREE는 CREATE TABLE문 정의시 0에서 99사이의 값으로 사용자가 명시할 수 있다.

PCTUSED

한 페이지가 갱신만 가능한 상태에서 다시 삽입이 가능한 상태로 가기 위한 페이지 사용 공간의 최소 비율을 의미한다. 페이지의 여유 공간이 PCTFREE에 명시한 비율에 도달하면 더 이상 삽입 연산은 안되며, 갱신만 가능해진다. 이후 갱신과 삭제 등으로 페이지 사용 공간의 비율이 PCTUSED에서 정한 값보다 낮아지면 새로운 행을 삽입할 수 있게 된다.

CREATE TABLE문 정의시 0에서 99사이의 값으로 사용자가 명시할 수 있다.

* PCTFREE와 PCTUSED에 대한 자세한 설명은 *SQL Reference*의 CREATE TABLE문 설명을 참조한다.

INIT_TRANS

한 페이지에 동시에 갱신 연산을 수행할 수 있는 트랜잭션의 개수로, 페이지를 생성할 때 설정된다. 실제 트랜잭션의 개수는 페이지 내의 가용 공간이 허용하는 한 MAX_TRANS에 설정된 개수까지 증가할 수 있다.

MAX_TRANS

한 페이지에서 동시에 갱신 연산을 수행할 수 있는 트랜잭션의 최대 개수이다.

INITEXTENTS

테이블을 생성할 때 할당하는 가용 익스텐트 개수를 나타낸다.

NEXTEXTENTS

테이블의 공간을 확장할 때 할당할 수 있는 추가 익스텐트 개수를 나타낸다.

MINEXTENTS

테이블의 최소 가용 익스텐트 개수를 나타낸다.

MAXEXTENTS

테이블의 최대 가용 익스텐트 개수를 나타낸다.

IS_PARTITIONED

테이블이 파티션드 테이블인지 여부를 나타내는 식별자이다. 'Y'는 파티션드 테이블이고, 'N'은 파티션드 테이블이 아니다.

TEMPORARY

해당 테이블이 임시 테이블인지 여부를 나타낸다.

- D: 트랜잭션에 한정되는 임시 테이블임
- P: 세션에 한정되는 임시 테이블임
- N: 임시 테이블이 아님

HIDDEN

해당 테이블이 숨기는 테이블인지 여부를 나타낸다.

- Y: 사용자에게 숨기는 테이블임
- N: 사용자에게 공개된 테이블임 (일반 테이블)

PARALLEL_DEGREE

파티션드 테이블을 스캔할 때 병렬 질의를 처리하는 스레드의 개수를 나타낸다.

ACCESS

테이블의 데이터에 대한 접근 모드를 나타낸다. 기본 모드는 읽기/쓰기가 가능한 W이다.

- R: 데이터 읽기 전용 모드
- W: 데이터 읽기/쓰기 모드 (기본 모드)
- A: 데이터 읽기/추가 모드. 이 모드에서는 데이터 변경/삭제가 허용되지 않는다.

참조 테이블

SYS_USERS_

SYS_TABLE_PARTITIONS_

테이블의 파티션을 관리하는 메타 테이블이다.

| Column name | Type | Description |
|---------------------|---------------|-------------------------------|
| USER_ID | INTEGER | 사용자 식별자 |
| TABLE_ID | INTEGER | 테이블 식별자 |
| PARTITION_OID | BIGINT | 파티션 객체 식별자 |
| PARTITION_ID | INTEGER | 파티션 식별자 |
| PARTITION_NAME | VARCHAR(128) | 파티션 이름 |
| PARTITION_MIN_VALUE | VARCHAR(4000) | 파티션의 최소 기준값 (해쉬 파티션의 경우 NULL) |
| PARTITION_MAX_VALUE | VARCHAR(4000) | 파티션의 최대 기준값 (해쉬 파티션의 경우 NULL) |
| PARTITION_ORDER | INTEGER | 파티션 순서 (해쉬 파티션일 경우 필요) |

| Column name | Type | Description |
|----------------------------|---------|-------------------------------|
| TBS_ID | INTEGER | 테이블스페이스 식별자 |
| PARTITION_ACCESS | CHAR(1) | 파티션 접근 모드 |
| REPLICATION_COUNT | INTEGER | 파티션에 관련된 이중화 객체의 개수 |
| REPLICATION_RECOVERY_COUNT | INTEGER | 파티션에 대해 복구 옵션을 설정한 이중화 객체의 개수 |
| CREATED | DATE | 파티션이 생성된 시간 |
| LAST_DDL_TIME | DATE | 파티션을 마지막으로 DDL 변경 작업한 시간 |

칼럼 정보

USER_ID

테이블 소유자의 사용자 식별자로, SYS_USERS_ 메타 테이블의 한 USER_ID 값과 동일하다.

TABLE_ID

테이블 식별자로, 시스템 시퀀스에 의해 자동으로 부여된다.

PARTITION_OID

시스템 내부에서 자동으로 부여되는 파티션 객체 식별자이다. 메타 테이블 조회 시 사용하는 PARTITION_ID와 달리 시스템 내부 동작 시에만 사용된다.

PARTITION_ID

파티션 식별자이다.

PARTITION_NAME

사용자가 명시한 파티션 이름이다.

PARTITION_MIN_VALUE

파티션의 최소 기준값을 문자열로 보여준다. 해쉬 (HASH) 파티션인 경우에는 널(NULL)이다.

PARTITION_MAX_VALUE

파티션의 최대 기준값을 문자열로 보여준다. 해쉬 (HASH) 파티션인 경우에는 널(NULL)이다.

PARTITION_ORDER

파티션들 중에서 이 파티션의 순서를 나타낸다. 해쉬 (HASH) 파티션인 경우에 필요하다.

TBS_ID

테이블이 저장되는 테이블스페이스의 식별자이다.

PARTITION_ACCESS

파티션의 데이터에 대한 접근 모드를 나타낸다. 기본 모드는 읽기/쓰기가 가능한 W이다.

- R: 데이터 읽기 전용 모드
- W: 데이터 읽기/쓰기 모드(기본 모드)
- A: 데이터 읽기/추가 모드. 이 모드에서는 데이터 변경/삭제가 허용되지 않는다.

REPLICATION_COUNT

이 파티션에 관련된 이중화 객체의 개수를 나타낸다.

REPLICATION_RECOVERY_COUNT

이 파티션에 대해 복구 옵션을 설정한 이중화 객체의 개수를 나타낸다.

참조 테이블

SYS_USERS_
SYS_TABLES_
SYS_PART_TABLES_

SYS_TABLE_SIZE_

시스템에 있는 디스크 테이블과 메모리 테이블의 실제 크기 정보를 저장하는 메타 테이블이다.

| Column name | Type | Description |
|-------------|--------------|---------------------|
| USER_NAME | VARCHAR(128) | 테이블 소유자 |
| TABLE_NAME | VARCHAR(128) | 테이블 이름 |
| TBS_NAME | VARCHAR(128) | 테이블이 저장된 테이블스페이스 이름 |
| MEMORY_SIZE | BIGINT | 메모리 테이블의 크기 |
| DISK_SIZE | BIGINT | 디스크 테이블의 크기 |

SYS_TBS_USERS_

사용자와 사용자 정의 테이블스페이스간의 관계에 대한 정보가 저장된 테이블이다.

| Column name | Type | Description |
|-------------|---------|------------------|
| TBS_ID | INTEGER | 테이블스페이스 식별자 |
| USER_ID | INTEGER | 사용자 식별자 |
| IS_ACCESS | INTEGER | 테이블스페이스 접근 허용 여부 |

칼럼 정보

TBS_ID

테이블스페이스 식별자이다.

USER_ID

사용자 식별자로, SYS_USERS_ 메타 테이블의 한 USER_ID 값과 동일하다.

IS_ACCESS

사용자가 해당 테이블스페이스에 접근 가능한지를 나타낸다.

- 0: 접근불가
- 1: 접근가능

참조 테이블

SYS_USERS_

SYS_TRIGGERS_

트리거의 기본 정보를 저장하는 메타 테이블이다.

| Column name | Type | Description |
|-------------------|--------------|---------------------------------|
| USER_ID | INTEGER | 사용자 식별자 |
| USER_NAME | VARCHAR(128) | 사용자 이름 |
| TRIGGER_OID | BIGINT | 트리거 식별자 |
| TRIGGER_NAME | VARCHAR(128) | 트리거 이름 |
| TABLE_ID | INTEGER | 테이블 식별자 |
| IS_ENABLE | INTEGER | 트리거 수행 여부 |
| EVENT_TIME | INTEGER | 트리거 수행 시점 |
| EVENT_TYPE | INTEGER | 트리거 이벤트 타입 |
| UPDATE_COLUMN_CNT | INTEGER | UPDATE 시 트리거를 발생시키는 칼럼 개수 |
| GRANULARITY | INTEGER | 트리거 수행 단위 구분 |
| REF_ROW_CNT | INTEGER | REFERENCING 구문의 ALIAS 개수 |
| SUBSTRING_CNT | INTEGER | 트리거 구문을 저장하고 있는 레코드 수 |
| STRING_LENGTH | INTEGER | 트리거 구문의 전체 문자열 길이 |
| CREATED | DATE | 트리거가 생성된 시간 |
| LAST_DDL_TIME | DATE | 트리거에 대해 마지막으로 DDL 변경 작업이 일어난 시간 |

칼럼 정보

USER_ID

사용자 식별자로, SYS_USERS_ 메타 테이블의 한 USER_ID 값과 동일하다.

USER_NAME

사용자 이름으로, SYS_USERS_ 메타 테이블의 한 USER_NAME 값과 동일하다.

TRIGGER_OID

트리거 식별자로, 시스템에 의해 자동으로 부여된다.

TRIGGER_NAME

사용자가 명시한 트리거 이름이다.

TABLE_ID

트리거가 정의된 테이블의 식별자로, SYS_TABLES_ 메타 테이블의 한 TABLE_ID 값과 동일하다.

IS_ENABLE

트리거를 발생시킬지 여부를 나타내는 값으로, ALTER TRIGGER문을 사용해서 변경할 수 있다.

- 0: 발생시키지 않음
- 1: 발생시킴

EVENT_TIME

트리거를 발생시킬 시점을 나타낸다.

- 1: BEFORE
- 2: AFTER
- 3: INSTEAD OF

EVENT_TYPE

트리거를 발생시키는 이벤트의 타입을 나타낸다.

- 1: INSERT
- 2: DELETE
- 4: UPDATE

UPDATE_COLUMN_CNT

갱신 시 트리거를 발생시키는 칼럼 수를 나타낸다. 이 값은 SYS_TRIGGER_UPDATE_COLUMNS_ 메타 테이블의 해당 트리거와 관련된 레코드의 개수와 동일하다.

GRANULARITY

트리거를 발생시키는 단위를 나타낸다.

- 1: FOR EACH ROW
- 2: FOR EACH STATEMENT

REF_ROW_CNT

REFERENCING 구문에 정의된 ALIAS의 개수이다.

SUBSTRING_CNT

한 트리거 구문은 나뉘어져서 SYS_TRIGGER_STRINGS_ 메타 테이블에 여러 레코드로 저장된다. 이 값은 그 구문을 저장하는 레코드의 수를 나타낸다.

STRING_LENGTH

트리거 구문의 전체 문자열 길이이다.

참조 테이블

SYS_USERS_
SYS_TABLES_

SYS_TRIGGER_DML_TABLES_

트리거가 참조하고 접근하는 테이블의 정보를 저장하는 메타 테이블이다.

| Column name | Type | Description |
|--------------|---------|----------------|
| TABLE_ID | INTEGER | 테이블 식별자 |
| TRIGGER_OID | BIGINT | 트리거 식별자 |
| DML_TABLE_ID | INTEGER | 트리거 내의 테이블 식별자 |
| STMT_TYPE | INTEGER | 실행 구문 종류 |

칼럼 정보**TABLE_ID**

트리거의 기반 테이블의 식별자로, SYS_TABLES_ 메타 테이블의 한 TABLE_ID 값과 동일하다.

TRIGGER_OID

트리거 식별자로, SYS_TRIGGERS_ 메타 테이블의 한 TRIGGER_OID 값과 동일하다.

DML_TABLE_ID

트리거 내에서 DML문으로 접근하는 테이블의 식별자로, SYS_TABLES_ 메타 테이블의 한 TABLE_ID 값과 동일하다.

STMT_TYPE

테이블에 수행하는 DML 구문의 종류를 나타낸다.

- 8: DELETE
- 19: INSERT
- 33: UPDATE

참조 테이블

SYS_TABLES_
SYS_TRIGGERS_

SYS_TRIGGER_STRINGS_

트리거 구문을 저장하는 메타 테이블이다.

| Column name | Type | Description |
|-------------|--------------|------------------------------|
| TABLE_ID | INTEGER | 테이블 식별자 |
| TRIGGER_OID | BIGINT | 트리거 식별자 |
| SEQNO | INTEGER | 나뉘어 저장된 구문 조각의 트리거 구문내에서의 위치 |
| SUBSTRING | VARCHAR(100) | 나뉘어진 트리거 구문 |

칼럼 정보

TABLE_ID

테이블 식별자로, SYS_TABLES_ 메타 테이블의 한 TABLE_ID 값과 동일하다.

TRIGGER_OID

트리거 식별자로, SYS_TRIGGERS_ 메타 테이블의 한 TRIGGER_OID 값과 동일하다.

SEQNO

한 트리거의 전체 구문을 여러 레코드로 SYS_TRIGGER_STRINGS_에 저장할 때, 이들 레코드 중에서 이 레코드의 위치를 나타낸다.

SUBSTRING

트리거 구문의 문자열 조각이다. 한 TRIGGER_OID 값으로 레코드들을 검색하여 SEQNO 순서대로 SUBSTRING 값을 합치면 트리거 전체 구문을 생성할 수 있다.

참조 테이블

SYS_TABLES_
SYS_TRIGGERS_

SYS_TRIGGER_UPDATE_COLUMNS_

갱신시 트리거를 발생시키는 칼럼 정보를 저장하는 메타 테이블이다.

| Column name | Type | Description |
|-------------|---------|-------------|
| TABLE_ID | INTEGER | 테이블 식별자 |
| TRIGGER_OID | BIGINT | 트리거 식별자 |
| COLUMN_ID | INTEGER | 칼럼 식별자 |

칼럼 정보

TABLE_ID

테이블 식별자로, SYS_TABLES_ 메타 테이블의 한 TABLE_ID 값과 동일하다.

TRIGGER_OID

트리거 식별자로, SYS_TRIGGERS_ 메타 테이블의 한 TRIGGER_OID 값과 동일하다.

COLUMN_ID

칼럼 식별자로, SYS_COLUMNS_ 메타 테이블의 한 COLUMN_ID 값과 동일하다.

참조 테이블

SYS_TABLES_
SYS_TRIGGERS_

SYS_USERS_

데이터베이스 사용자에 대한 정보를 기록하는 테이블이다.

| Column name | Type | Description |
|-----------------------|--------------|---|
| USER_ID | INTEGER | 사용자 식별자 |
| USER_NAME | VARCHAR(128) | 사용자 이름 |
| PASSWORD | VARCHAR(256) | 사용자 패스워드 |
| DEFAULT_TBS_ID | INTEGER | 기본 테이블스페이스 식별자 |
| TEMP_TBS_ID | INTEGER | 임시 테이블스페이스 식별자 |
| ACCOUNT_LOCK | CHAR(1) | 계정의 잠김 여부 표시 N: UNLOCKED L: LOCKED |
| ACCOUNT_LOCK_DATE | DATE | 계정이 잠기게 된 날짜 |
| PASSWORD_LIMIT_FLAG | CHAR(1) | 패스워드 관리 정책의 사용 여부 표시 T: 패스워드 관리 정책 사용 F: 패스워드 관리 정책 미사용 |
| FAILED_LOGIN_ATTEMPTS | INTEGER | 로그인 실패 허용 최대 횟수 |
| FAILED_LOGIN_COUNT | INTEGER | 로그인 실패 횟수 |
| PASSWORD_LOCK_TIME | INTEGER | 계정이 한 번 잠긴 후 다시 풀리기 위해 경과되어야 하는 기간 |
| PASSWORD_EXPIRY_DATE | DATE | 패스워드 만료일 |
| PASSWORD_LIFE_TIME | INTEGER | 패스워드 유효기간 |
| PASSWORD_GRACE_TIME | INTEGER | 패스워드 만료 후 유예기간 |

| Column name | Type | Description |
|--------------------------|--------------|---|
| PASSWORD_REUSE_DATE | DATE | 동일한 패스워드가 재사용 가능해지는 날짜 |
| PASSWORD_REUSE_TIME | INTEGER | 미사용 |
| PASSWORD_REUSE_MAX | INTEGER | 동일한 패스워드의 재사용 가능 횟수 |
| PASSWORD_REUSE_COUNT | INTEGER | 미사용 |
| PASSWORD_VERIFY_FUNCTION | VARCHAR(128) | 패스워드를 검증할 콜백 함수(Callback Function) |
| USER_TYPE | CHAR(1) | 사용자 타입 표시 U: 사용자(User) R: 롤(Role) |
| DISABLE_TCP | CHAR(1) | 사용중인 TCP 접속의 제한 여부 표시 T: TCP 접속을 못하고, SSL이나 IPC로만 통신 F: TCP 접속 허용 |
| CREATED | DATE | 데이터베이스 사용자가 생성된 시간 |
| LAST_DDL_TIME | DATE | 사용자에 대해 마지막으로 DDL 변경 작업이 일어난 시간 |

칼럼 정보

USER_ID

사용자 식별자로, 시스템의 시퀀스에 의해 자동으로 부여된다.

USER_NAME

사용자가 명시한 사용자의 이름이다.

PASSWORD

사용자의 패스워드로 암호화 되어 있다.

DEFAULT_TBS_ID

기본 테이블스페이스 식별자로, 사용자가 객체 생성 시 테이블스페이스를 명시적으로 기술하지 않을 경우 사용된다.

TEMP_TBS_ID

사용자의 임시 테이블스페이스 식별자이다.

DISABLE_TCP

사용자의 TCP 접속을 허용하거나 제한하는 것을 나타낸다.

참조 테이블

DBA_USERS_

DBA_USERS_

데이터베이스 사용자에 대한 정보를 기록하는 테이블이다. SYS 사용자만이 조회할 수 있다.

| Column name | Type | Description |
|--------------------------|--------------|---|
| USER_ID | INTEGER | 사용자 식별자 |
| USER_NAME | VARCHAR(128) | 사용자 이름 |
| PASSWORD | VARCHAR(256) | 사용자 패스워드 |
| DEFAULT_TBS_ID | INTEGER | 기본 테이블스페이스 식별자 |
| TEMP_TBS_ID | INTEGER | 임시 테이블스페이스 식별자 |
| ACCOUNT_LOCK | CHAR(1) | 계정의 잠김 여부 표시 N: UNLOCKED L: LOCKED |
| ACCOUNT_LOCK_DATE | DATE | 계정이 잠기게 된 날짜 |
| PASSWORD_LIMIT_FLAG | CHAR(1) | 패스워드 관리 정책의 사용 여부 표시 T: 패스워드 관리 정책 사용 F: 패스워드 관리 정책 미사용 |
| FAILED_LOGIN_ATTEMPTS | INTEGER | 로그인 실패 허용 최대 횟수 |
| FAILED_LOGIN_COUNT | INTEGER | 로그인 실패 횟수 |
| PASSWORD_LOCK_TIME | INTEGER | 계정이 한 번 잠긴 후 다시 풀리기 위해 경과되어야 하는 기간 |
| PASSWORD_EXPIRY_DATE | DATE | 패스워드 만료일 |
| PASSWORD_LIFE_TIME | INTEGER | 패스워드 유효기간 |
| PASSWORD_GRACE_TIME | INTEGER | 패스워드 만료 후 유예기간 |
| PASSWORD_REUSE_DATE | DATE | 동일한 패스워드가 재사용 가능해지는 날짜 |
| PASSWORD_REUSE_TIME | INTEGER | 미사용 |
| PASSWORD_REUSE_MAX | INTEGER | 동일한 패스워드의 재사용 가능 횟수 |
| PASSWORD_REUSE_COUNT | INTEGER | 미사용 |
| PASSWORD_VERIFY_FUNCTION | VARCHAR(128) | 패스워드를 검증할 콜백 함수(Callback Function) |
| USER_TYPE | CHAR(1) | 사용자 타입 표시 U: 사용자(User) R: 롤(Role) |
| DISABLE_TCP | CHAR(1) | 사용중인 TCP 접속의 제한 여부 표시 T: TCP 접속을 못하고, SSL이나 IPC로만 통신 F: TCP 접속 허용 |
| CREATED | DATE | 데이터베이스 사용자가 생성된 시간 |

| Column name | Type | Description |
|---------------|------|---------------------------------|
| LAST_DDL_TIME | DATE | 사용자에 대해 마지막으로 DDL 변경 작업이 일어난 시간 |

칼럼 정보

USER_ID

사용자 식별자로, 시스템의 시퀀스에 의해 자동으로 부여된다.

USER_NAME

사용자가 명시한 사용자의 이름이다.

PASSWORD

사용자의 패스워드로 암호화 되어 있다.

DEFAULT_TBS_ID

기본 테이블스페이스 식별자로, 사용자가 객체 생성 시 테이블스페이스를 명시적으로 기술하지 않을 경우 사용된다.

TEMP_TBS_ID

사용자의 임시 테이블스페이스 식별자이다.

DISABLE_TCP

사용자의 TCP 접속을 허용하거나 제한하는 것을 나타낸다.

SYS_USER_ROLES_

사용자에게 부여된 롤(Role) 정보가 기록되는 메타 테이블이다.

| Column name | Type | Description |
|-------------|---------|-----------------|
| GRANTOR_ID | INTEGER | 롤을 부여한 사용자의 식별자 |
| GRANTEE_ID | INTEGER | 롤이 부여된 사용자의 식별자 |
| ROLE_ID | INTEGER | 롤 식별자 |

칼럼 정보

GRANTOR_ID

롤을 부여한 사용자의 식별자로, SYS_USERS_ 메타 테이블의 한 USER_ID 값과 동일하다.

GRANTEE_ID

롤이 부여된 사용자의 식별자로, SYS_USERS_ 메타 테이블의 한 USER_ID 값과 동일하다.

ROLE_ID

롤의 식별자로, SYS_USERS_ 메타 테이블의 한 USER_ID 값과 동일하다.

참조 테이블

SYS_USERS_
SYS_TABLES_

SYS_VIEWS_

뷰에 대한 기본 정보는 SYS_TABLES_ 메타 테이블에 기록된다. 이 메타 테이블은 그 외의 뷰에 대한 부가 정보를 저장한다.

| Column name | Type | Description |
|-------------|---------|-------------|
| USER_ID | INTEGER | 뷰의 소유자 식별자 |
| VIEW_ID | INTEGER | 뷰 식별자 |
| STATUS | INTEGER | 뷰의 상태 |
| READ_ONLY | CHAR(1) | 읽기전용 뷰인지 여부 |

칼럼 정보**USER_ID**

뷰 소유자의 사용자 식별자로, SYS_USERS_ 메타 테이블의 한 USER_ID 값과 동일하다.

VIEW_ID

뷰 식별자로, SYS_TABLES_ 메타 테이블의 한 TABLE_ID 값과 동일하다.

STATUS

뷰 상태를 나타내는 값이다.

- 0: VALID
- 1: INVALID

참조 테이블

SYS_USERS_
SYS_TABLES_

SYS_VIEW_PARSE_

사용자가 정의한 뷰의 구문 텍스트를 기록하는 테이블이다.

| Column name | Type | Description |
|-------------|---------|-------------|
| USER_ID | INTEGER | 뷰의 소유자 식별자 |

| Column name | Type | Description |
|-------------|--------------|---|
| VIEW_ID | INTEGER | 뷰 식별자 |
| SEQ_NO | INTEGER | 뷰 생성문 텍스트를 여러 개의 텍스트 조각으로 SYS_VIEW_PARSE_에 저장할 때, 여러 레코드 중에서 이 레코드의 위치이다. |
| PARSE | VARCHAR(100) | 뷰 생성문 텍스트 조각 |

칼럼 정보

USER_ID

뷰 소유자의 사용자 식별자로, SYS_USERS_ 메타 테이블의 한 USER_ID 값과 동일하다.

VIEW_ID

뷰 식별자로, SYS_TABLES_ 메타 테이블의 한 TABLE_ID 값과 동일하다.

SEQ_NO

한 뷰의 생성 구문 텍스트를 SYS_VIEW_PARSE_에 여러 개의 레코드로 저장할 때, 이들 레코드 중에서 해당 레코드의 위치를 나타낸다.

PARSE

뷰 구문의 조각난 문자열이다. 한 VIEW_ID 값으로 레코드들을 검색하여 SEQ_NO 순서대로 PARSE 값을 합치면 뷰 전체 구문을 생성할 수 있다.

참조 테이블

SYS_USERS_
SYS_TABLES_

SYS_VIEW_RELATED_

사용자가 정의한 뷰들이 접근하는 객체에 대한 정보를 기록하는 테이블이다.

| Column name | Type | Description |
|---------------------|--------------|---------------------|
| USER_ID | INTEGER | 뷰의 소유자 식별자 |
| VIEW_ID | INTEGER | 뷰 식별자 |
| RELATED_USER_ID | INTEGER | 뷰가 접근하는 객체의 소유자 식별자 |
| RELATED_OBJECT_NAME | VARCHAR(128) | 뷰가 접근하는 객체의 이름 |
| RELATED_OBJECT_TYPE | INTEGER | 뷰가 접근하는 객체의 타입 |

칼럼 정보

USER_ID

뷰 소유자의 사용자 식별자로, SYS_USERS_ 메타 테이블의 한 USER_ID 값과 동일하다.

VIEW_ID

뷰 식별자로, SYS_TABLES_ 메타 테이블의 한 TABLE_ID 값과 동일하다.

RELATED_USER_ID

뷰가 접근하는 객체 소유자의 사용자 식별자로, SYS_USERS_ 메타 테이블의 한 USER_ID 값과 동일하다.

RELATED_OBJECT_NAME

뷰가 접근하는 객체의 이름이다.

RELATED_OBJECT_TYPE

뷰가 접근하는 객체의 타입이다. 뷰는 저장 함수, 테이블, 시퀀스, 다른 뷰, 데이터베이스 링크, 또는 시노님에 접근할 수 있다. 각 객체의 타입 식별자는 다음과 같다.

- 1: 저장 함수
- 2: 테이블, 시퀀스, 뷰
- 4: 데이터베이스 링크
- 5: 시노님

참조 테이블

SYS_USERS_
SYS_TABLES_
SYS_PROCEDURES_

SYS_XA_HEURISTIC_TRANS_

데이터베이스가 가지고 있는 글로벌(Global) 트랜잭션 식별자들과 그 상태를 가지고 있는 메타 테이블이다.

| Column name | Type | Description |
|------------------|--------------|----------------------------------|
| FORMAT_ID | BIGINT | 전역 (Global) 트랜잭션의 형식(Format) 식별자 |
| GLOBAL_TX_ID | VARCHAR(128) | 전역 트랜잭션 식별자 |
| BRANCH_QUALIFIER | VARCHAR(128) | 전역 트랜잭션의 branch qualifier |
| STATUS | INTEGER | 전역 트랜잭션 상태 |
| OCCUR_TIME | DATE | XA 트랜잭션이 발생한 시간 |

칼럼 정보

FORMAT_ID

글로벌 트랜잭션의 형식(Format) 식별자

GLOBAL_TX_ID

글로벌 트랜잭션 식별자

BRANCH_QUALIFIER

글로벌 트랜잭션의 브랜치(Branch) qualifier

STATUS

글로벌 트랜잭션의 상태

SYS_GEOMETRIES_

GEOMETRY 칼럼을 보유한 테이블에 대한 정보를 저장하고 있는 메타 테이블이다.

| Column name | Type | Description |
|-----------------|----------|-----------------------|
| USER_ID | INTERGER | 테이블의 소유자 |
| TABLE_ID | INTERGER | 테이블의 식별자 |
| COLUMN_ID | INTERGER | 칼럼의 식별자 |
| COORD_DIMENSION | INTERGER | GEOMETRY 객체의 차원 |
| SRID | INTERGER | 데이터베이스 내에서의 공간 참조 식별자 |

SYS_GEOMETRY_COLUMNS_

GEOMETRY 칼럼에 공간 참조 식별자(SRID, Spatial Reference ID)를 지정, 관리하기 위해 사용한다.

이 메타 테이블의 synonym은 GEOMETRY_COLUMNS_이다.

| Column name | Type | Description |
|-------------------|--------------|-----------------------|
| F_TABLE_SCHEMA | VARCHAR(128) | 테이블 소유자 이름 |
| F_TABLE_NAME | VARCHAR(128) | 테이블 이름 |
| F_GEOMETRY_COLUMN | VARCHAR(128) | 칼럼의 이름 |
| COORD_DIMENSION | INTERGER | GEOMETRY 객체의 차원 |
| SRID | INTERGER | 데이터베이스 내에서의 공간 참조 식별자 |

USER_SRS_

공간 참조 식별자(SRID, Spatial Reference Identifier)와 이에 대응하는 공간 참조 시스템(SRS, Spatial Reference System)에 관한 정보를 관리하기 위해 사용한다.

이 메타 테이블의 synonym은 SPATIAL_REF_SYS 이다.

SPATIAL_REF_SYS 테이블에 Spatial Reference System 메타 데이터를 등록 및 삭제하기 위해서는 SYS_SPATIAL 패키지의 ADD_SPATIAL_REF_SYS, DELETE_SPATIAL_REF_SYS 프로시저를 사용해야 한다. 메타 데이터를 등록할 때 SRID와 AUTH_SRID를 동일한 값으로 사용하는 것을 권장한다. 자세한 내용은 *Spatial Manual*을 참조한다.

| Column name | Type | Description |
|-------------|---------------|------------------------------------|
| SRID | INTEGER | 데이터베이스 내에서의 공간 참조 식별자 |
| AUTH_NAME | VARCHAR(256) | 표준 이름 |
| AUTH_SRID | INTEGER | 표준 식별자 |
| SRTEXT | VARCHAR(2048) | OGC-WKT 형태로 표현 되는 공간 참조 시스템에 대한 설명 |
| PROJ4TEXT | VARCHAR(2048) | PROJ4에서 사용되는 정보 |

성능 뷰

성능 뷰 (performance view)란 메모리에 존재하는 구조이지만 일반 테이블 형태로 제공되어 시스템 메모리, 프로세스 상태, 세션, 버퍼, 스레드 등에 대한 Altibase 시스템 내부 정보를 사용자가 모니터링 할 수 있다.

사용자가 테이블에 저장된 데이터를 검색하기 위하여 SQL을 사용하는 것처럼, Altibase 운용 시 사용되는 메모리 객체 (예. 세션 정보, 로그 정보)에 관한 정보를 SQL문을 이용하여 성능 뷰로부터 쉽게 검색할 수 있다.

이 절에서는 Altibase가 지원하는 성능 뷰의 종류, 구조 및 기능, 조회 방법, 그리고 각 뷰에서 제공하는 정보에 대해 설명한다.

구조 및 기능

Altibase 내부에는 사용자가 생성한 객체 (테이블 같은)뿐만 아니라 DBMS 자체 운용에 필요한 다수의 정보를 저장하고 있다.

특히 Altibase의 경우 메모리 공간 외에도 디스크 공간에도 테이블 생성 및 조회가 가능한 하이브리드 형태이기 때문에, Altibase 자체에 대한 모니터링 기능이 필수적이라고 할 수 있다.

성능 뷰는 Altibase 운용과정에서 사용되는 대부분의 내부 메모리 구조체를 뷰 형태로 제공한 것이다. 해당 테이블에 대해 조회를 하는 순간에 그 데이터가 실시간으로 생성되기 때문에 언제나 Altibase 프로세스 내부의 최신 정보를 얻을 수 있다.

성능 뷰는 항상 읽기 전용 속성을 가진다. 만일 이 테이블에 대해 변경 연산을 시도한다면, Altibase는 에러를 내고, 해당 트랜잭션에 대한 부분 철회 (rollback)를 수행할 것이다.

성능 뷰의 조회 방법

성능 뷰의 전체 목록은 iSQL에서 다음과 같이 조회할 수 있다.

```
iSQL> SELECT * FROM V$TAB;
```

성능 뷰의 스키마는 일반 테이블과 마찬가지로 iSQL 에서 DESC 명령어를 통해 확인할 수 있고, 데이터는 일반 테이블과 동일하게 SELECT문을 이용하여 검색할 수 있다.

성능 뷰의 종류

성능 뷰의 이름은 V\$로 시작한다. 아래 표는 전체 성능 뷰의 목록이다.

| 이름 | 설명 |
|---------------------------------------|---|
| V\$ACCESS_LIST | 서버에 접근하는 특정 IP 패킷의 접근 허용 및 제한 정보 |
| V\$ALLCOLUMN | 성능 뷰를 구성하는 칼럼 정보 |
| V\$ARCHIVE | 아카이브와 백업 관련 정보 |
| V\$BACKUP_INFO | 현재까지 수행된 증분 백업에 대한 정보 |
| V\$BUFFPAGEINFO | 버퍼 매니저의 버퍼 프레임 통계 정보 |
| V\$BUFFPOOL_STAT | 버퍼 풀 적중 비율 (hit ratio)를 포함한 버퍼 풀 관련 통계 정보 |
| V\$CATALOG | 테이블의 구조 정보 |
| V\$DATABASE | 메모리 데이터베이스의 내부 정보 |
| V\$DATAFILES | 테이블스페이스에서 사용하는 데이터 파일의 정보 |
| V\$DATATYPE | Altibase가 지원하는 데이터 타입의 정보 |
| V\$DBA_2PC_PENDING | in-doubt 상태의 분산 트랜잭션 목록 |
| V\$DBLINK_ALTIlinker_STATUS | 데이터베이스 링크를 위한 Altilinker 프로세스의 상태 정보 |
| V\$DBLINK_DATABASE_LINK_INFO | 데이터베이스에 존재하는 데이터베이스 링크 객체 정보 |
| V\$DBLINK_GLOBAL_TRANSACTION_INFO | 데이터베이스 링크를 사용하는 트랜잭션 정보 |
| V\$DBLINK_LINKER_CONTROL_SESSION_INFO | 링커 제어 세션의 상태 정보 |
| V\$DBLINK_LINKER_DATA_SESSION_INFO | 링커 데이터 세션들의 상태 정보 |
| V\$DBLINK_LINKER_SESSION_INFO | 링커 제어 세션과 링커 데이터 세션의 개수 정보 |
| V\$DBLINK_NOTIFIER_TRANSACTION_INFO | Altilinker가 처리중인 (장애가 발생한) 분산 트랜잭션의 정보 |
| V\$DBLINK_REMOTE_STATEMENT_INFO | 데이터베이스 링크 사용시 원격 서버에서 수행한 구문 (statement) 정보 |
| V\$DBLINK_REMOTE_TRANSACTION_INFO | 데이터베이스 링크 사용시 원격 서버에서 발생한 트랜잭션 정보 |
| V\$DBMS_STATS | 데이터베이스 전체의 통계 정보 |
| V\$DB_FREEPAGELISTS | 사용가능한 페이지 리스트 정보 |

| 이름 | 설명 |
|-----------------------|--|
| V\$DB_PROTOCOL | 서버로 유입되는 데이터베이스 프로토콜의 정보 |
| V\$DIRECT_PATH_INSERT | Direct-path 업로드 관련 통계 정보 |
| V\$DISKTBL_INFO | 디스크 테이블 정보 |
| V\$DISK_BTREE_HEADER | 디스크 BTREE 인덱스들의 헤더 정보 |
| V\$DISK_RTREE_HEADER | 디스크 RTREE 인덱스들의 헤더 정보 |
| V\$DISK_TEMP_INFO | 전체 디스크 임시 테이블의 메모리 사용 정보 |
| V\$DISK_TEMP_STAT | 현재 사용중인 각각의 디스크 임시 테이블 정보 |
| V\$DISK_UNDO_USAGE | 디스크상에서 현재 사용중인 언두 테이블스페이스의 양에 대한 정보 |
| V\$EVENT_NAME | Altibase 서버의 대기 이벤트 정보 |
| V\$EXTPROC_AGENT | 외부 프로시저 실행을 위해 생성된 에이전트 프로세스(agent process)의 정보 |
| V\$FILESTAT | 디스크의 데이터 파일별 I/O 통계 정보 |
| V\$FLUSHER | 버퍼를 플러쉬하는 플러셔에 대한 정보 |
| V\$FLUSHINFO | 버퍼 플러쉬 정보 |
| V\$INDEX | 테이블의 인덱스 정보 |
| V\$INSTANCE | Altibase의 현재 구동 단계 정보 |
| V\$INTERNAL_SESSION | DBMS_CONCURRENT_EXEC 패키지에서 생성된 세션의 정보 |
| V\$LATCH | 버퍼 풀의 버퍼 제어 블록 (BCB) 래치 (latch) 정보와 읽기 또는 쓰기가 시도된 페이지에 대한 read/write latch에 대한 통계 정보 |
| V\$LFG | LFG에 대한 정보와 그룹커밋 관련 통계값 |
| V\$LOCK | 현재 시점에서 데이터베이스의 모든 테이블 레벨 lock 노드 정보 |
| V\$LOCK_STATEMENT | Lock과 statement 에 대한 정보 |
| V\$LOCK_WAIT | 트랜잭션의 락 획득을 위한 대기 상태 정보 |
| V\$LOG | 로그 앵커 정보 |
| V\$MEMGC | 메모리 공간 회수를 위한 garbage collection에 대한 정보 |
| V\$MEMSTAT | Altibase 프로세스가 사용하는 메모리 통계 정보 |

| 이름 | 설명 |
|------------------------------------|---|
| V\$MENTBL_INFO | 메모리 테이블 정보 |
| V\$MEM_BTREE_HEADER | 메모리 BTREE 인덱스의 헤더 정보 |
| V\$MEM_BTREE_NODEPOOL | 메모리 BTREE 인덱스를 위한 노드 풀 정보 |
| V\$MEM_RTREE_HEADER | 메모리 RTREE 인덱스의 헤더 정보 |
| V\$MEM_RTREE_NODEPOOL | 메모리 RTREE 인덱스를 위한 노드 풀 정보 |
| V\$MEM_TABLESPACES | 메모리에 생성된 테이블스페이스 정보 |
| V\$MEM_TABLESPACE_CHECKPOINT_PATHS | 체크포인트 발생시 반영되는 DB 파일의 위치 정보 |
| V\$MEM_TABLESPACE_STATUS_DESC | 메모리 테이블스페이스의 상태 정보 |
| V\$MUTEX | 동시성 제어를 위해서 Altibase 프로세스에서 사용되고 있는 뮤텍스(mutex) 통계 정보 |
| V\$NLS_PARAMETERS | NLS 관련 파라미터 정보 |
| V\$NLS_TERRITORY | 설정 가능한 지역의 이름 정보 |
| V\$OBSOLETE_BACKUP_INFO | 더 이상 유지할 필요가 없는 백업 정보 |
| V\$PKGTEXT | 시스템에서 수행되는 패키지의 문자열 정보 |
| V\$PLANTEXT | SQL의 실행 계획 텍스트 정보 |
| V\$PROCTEXT | 저장 프로시저의 텍스트 정보 |
| V\$PROPERTY | Altibase에 설정된 프로퍼티 정보 |
| V\$QUEUE_DELETE_OFF | DELETE 문을 허용하지 않는 큐 테이블의 객체 식별자(OID) 정보 |
| V\$REPEXEC | 이중화 관리자 정보 |
| V\$REPGAP | 이중화 송신자의 작업 로그 레코드와 현재 생성된 최근 로그 레코드간의 차이 정보 |
| V\$REPGAP_PARALLEL | 병렬 수행중인 이중화 송신 쓰레드의 작업 로그 레코드와 현재 생성된 최근 로그 레코드간의 차이 정보 |
| V\$REPLOGBUFFER | 이중화 전용 로그 버퍼의 정보 |
| V\$REPOFFLINE_STATUS | 오프라인 이중화의 수행 상태 정보 |
| V\$REPRECEIVER | 이중화 수신자 정보 |
| V\$REPRECEIVER_COLUMN | 이중화 수신자의 이중화 대상 칼럼 정보 |
| V\$ REPRECEIVER_PARALLEL | 병렬 수행중인 이중화 수신 쓰레드에 대한 정보 |
| V\$REPRECEIVER_PARALLEL_APPLY | 이중화 적용자 쓰레드에 대한 정보 |

| 이름 | 설명 |
|--------------------------------------|---|
| V\$REPRECEIVER_STATISTICS | 이중화 수신 쓰레드의 작업별 수행시간에 대한 통계 정보 |
| V\$REPRECEIVER_TRANSTBL | 이중화 수신자의 트랜잭션 테이블 정보 |
| V\$REPRECEIVER_TRANSTBL_PARALLEL | 병렬 수행중인 이중화 수신 쓰레드가 사용하는 트랜잭션 테이블 정보 |
| V\$REPRECOVERY | 이중화를 이용한 복구 정보 |
| V\$REPSENDER | 이중화 송신자 정보 |
| V\$REPSENDER_PARALLEL | 병렬 수행중인 이중화 송신 쓰레드에 대한 정보 |
| V\$REPSENDER_SENT_LOG_COUNT | 이중화 송신자가 전송한 로그의 DML 타입별 개수 정보 |
| V\$REPSENDER_SENT_LOG_COUNT_PARALLEL | Eager 모드의 병렬 이중화에서 각 송신 쓰레드가 전송한 로그의 DML 타입별 개수 정보 |
| V\$REPSENDER_STATISTICS | 이중화 송신 쓰레드의 작업 별 수행시간에 대한 통계 정보 |
| V\$REPSENDER_TRANSTBL | 이중화 송신자의 트랜잭션 테이블 정보 |
| V\$REPSENDER_TRANSTBL_PARALLEL | 병렬 수행중인 이중화 송신 쓰레드가 사용하는 트랜잭션 테이블 정보 |
| V\$REPSYNC | 이중화로 동기화 중인 테이블의 정보 |
| V\$SBUFFER_STAT | 보조 버퍼(Secondary Buffer)에 대한 통계 정보 |
| V\$SEGMENT | 테이블과 색인을 구성하는 세그먼트 정보 |
| V\$SEQ | 시퀀스 관련 정보 |
| V\$SERVICE_THREAD | Multiplexing 관련 서비스 쓰레드 정보 |
| V\$SERVICE_THREAD_MGR | 멀티플렉싱 (Multiplexing)과 관련하여 서비스 쓰레드가 생성되거나 삭제된 정보 |
| V\$SESSION | 클라이언트에 대응하는 Altibase 내부에 생성된 세션 정보 |
| V\$SESSION_EVENT | 구동 후부터 현재까지 접속한 세션의 모든 대기 이벤트 통계 정보 |
| V\$SESSION_WAIT | 현재 접속한 상태에 있는 모든 세션의 대기 이벤트 정보 |
| V\$SESSION_WAIT_CLASS | 현재 접속한 상태에 있는 모든 세션에 대해 대기 이벤트, 대기 클래스별로 누적된 대기 통계 정보 |
| V\$SESSIONMGR | Altibase의 세션 통계 정보 |

| 이름 | 설명 |
|---------------------------|---|
| V\$SESSTAT | 현재 접속된 세션의 상태 정보 |
| V\$SFLUSHER | 보조 버퍼(Secondary Buffer)의 페이지를 디스크에 플러시 하는 작업에 대한 정보 |
| V\$SFLUSHINFO | 보조 버퍼(Secondary Buffer)의 플러시 정보 |
| V\$SNAPSHOT | 스냅샷(SNAPSHOT)의 설정 상태와 메모리, 디스크 연두 테이블스페이스의 사용 정보 |
| V\$SQLTEXT | 시스템에서 수행되는 SQL문의 텍스트 정보 |
| V\$SQL_PLAN_CACHE | SQL Plan Cache의 현재 상태 및 통계 정보 |
| V\$SQL_PLAN_CACHE_PCO | SQL Plan Cache에 등록된 Plan Cache 객체에 대한 정보 |
| V\$SQL_PLAN_CACHE_SQLTEXT | SQL Plan Cache에 등록된 SQL 문 정보 |
| V\$STABLE_MEM_DATAFILES | 데이터 파일의 전체 경로 정보 |
| V\$STATEMENT | 현재 Altibase에 생성된 모든 세션의 구문 정보 |
| V\$STATNAME | 시스템 및 세션 상태와 이름 정보 |
| V\$ST_ANGULAR_UNIT | 향후 확장 예정 |
| V\$ST_AREA_UNIT | 향후 확장 예정 |
| V\$ST_LINEAR_UNIT | 향후 확장 예정 |
| V\$SYSSTAT | 시스템 상태 정보 |
| V\$SYSTEM_CONFLICT_PAGE | 페이지 타입 별 래치 경합 정보 |
| V\$SYSTEM_EVENT | 구동부터 현재까지의 대기 이벤트별 누적된 대기 통계 정보 |
| V\$SYSTEM_WAIT_CLASS | 구동부터 현재까지의 대기 클래스별 누적된 대기 통계 정보 |
| V\$TABLE | 모든 성능 뷰의 레코드 및 칼럼 정보 |
| V\$TABLESPACES | 테이블스페이스 정보 |
| V\$TIME_ZONE_NAMES | TIME_ZONE 프로퍼티에 설정할 수 있는 지역 이름과 약어 및 UTC 오프셋 값의 정보 |
| V\$TRACELOG | 트레이스 로깅 정보 |
| V\$TRANSACTION | 트랜잭션 객체 정보 |
| V\$TRANSACTION_MGR | Altibase 트랜잭션 관리자 정보 |
| V\$TSSEGS | 모든 TSS 세그먼트들의 정보 |
| V\$TXSEGS | 바인딩된 트랜잭션 세그먼트들의 정보 |

| 이름 | 설명 |
|--------------------|---|
| V\$UDSEGS | 모든 언두 세그먼트들의 정보 |
| V\$UNDO_BUFF_STAT | Undo 테이블스페이스의 버퍼 풀 관련 통계 정보 |
| V\$USAGE | 데이터베이스에 존재하는 테이블과 인덱스가 사용하는 공간량에 대한 정보 |
| V\$VERSION | Altibase 버전 정보 |
| V\$VOL_TABLESPACES | 휘발성 테이블스페이스에 대한 정보 |
| V\$WAIT_CLASS_NAME | Altibase 서버상의 대기 이벤트들을 클래스로 그룹화하기 위한 정보 |
| V\$XID | DBMS에 현재 존재하는 분산 트랜잭션 브랜치인 XID의 목록 |

V\$ACCESS_LIST

Altibase에 접근하는 특정 IP 패킷의 접근 허용 및 제한 정보를 보여준다.

| Column name | Type | Description |
|-------------|-------------|-------------------------------------|
| ID | INTEGER | ACCESS LIST 식별자 |
| ADDRESS | VARCHAR(40) | IP 주소 |
| OPERATION | VARCHAR(6) | IP 주소 접근 허용 및 제한 여부 |
| MASK | VARCHAR(16) | 서브넷 마스크(IPv4) 또는 prefix 비트 길이(IPv6) |
| LIMIT | INTEGER | 세션 최대 허용 개수 |
| CONNECTED | INTEGER | 현재 세션 접속 개수 |

칼럼 정보

ID

IP 패킷의 접근 허용 및 제한 목록의 식별자를 기술한다.

ADDRESS

IP 패킷 주소를 기술한다

OPERATION

IP 패킷 주소의 접근 허용 및 제한 여부를 보여준다.

- PERMIT : 접근 허용
- DENY: 접근 제한

MASK

IPv4 주소일 경우 서브넷 마스크를 기술하고, IPv6 주소인 경우에는 prefix 비트의 길이를 기술한다. 자세한 내용은 ACCESS_LIST 프로퍼티의 설명을 참조한다

LIMIT

ACCESS_LIST에 명시된 접속 가능한 IP 주소 영역에서 허용되는 최대 접속 세션 개수.

운영 중 RELOAD ACCESS LIST로 ACCESS_LIST를 추가하면, 기존에 연결된 세션은 영향을 받지 않으며, 변경 이후 새로운 연결 요청에 대해서만 ACCESS_LIST 조건이 적용된다. 예를 들어 ACCESS_LIST에 limit 값을 설정 후 RELOAD ACCESS LIST 수행하면, 적용 이후 새로운 연결에 대해서만 limit 값이 적용된다. 이런 경우, V\$ACCESS_LIST 조회시 Limit 값보다 CONNECTED 값이 더 클 수도 있다.

CONNECTED

ACCESS_LIST에 해당하는 현재 접속된 세션 개수

V\$ALLCOLUMN

모든 성능 뷰의 칼럼 정보를 나타낸다.

| Column name | Type | Description |
|-------------|-------------|-------------|
| TABLERNAME | VARCHAR(39) | 성능 뷰 이름 |
| COLNAME | VARCHAR(39) | 성능 뷰의 칼럼 이름 |

칼럼 정보**TABLERNAME**

성능 뷰의 이름을 나타낸다.

COLNAME

성능 뷰의 칼럼 이름을 나타낸다.

V\$ARCHIVE

아카이브와 백업 관련 정보를 보여준다.

| Column name | Type | Description |
|-----------------------|---------------|---|
| LFG_ID | INTEGER | 로그 파일의 그룹 식별자 |
| ARCHIVE_MODE | BIGINT | 아카이브 로그 모드 0: no archive log 모드 1: archive log 모드 |
| ARCHIVE_THR_RUNNING | BIGINT | 아카이브로그 쓰레드 수행 여부 |
| ARCHIVE_DEST | VARCHAR(1024) | 로그를 아카이브 하여 저장하는 디렉터리 |
| NEXTLOGFILE_TO_ARCH | INTEGER | 다음 번에 아카이브 할 로그 파일 번호 |
| OLDEST_ACTIVE_LOGFILE | INTEGER | 온라인로그 파일 중 가장 오래된 로그 파일 번호 |

| Column name | Type | Description |
|-----------------|---------|----------------|
| CURRENT_LOGFILE | INTEGER | 현재 온라인로그 파일 번호 |

칼럼 정보

LFG_ID

이는 0의 값을 가진 로그파일 그룹 고유번호이다.

ARCHIVE_MODE

데이터베이스의 아카이브 로그 모드를 나타낸다.

0: 노(No) 아카이브 로그 모드

1: 아카이브 로그 모드

V\$BACKUP_INFO

현재까지 수행된 모든 증분 백업에 대한 정보를 보여준다.

| Column name | Type | Description |
|--------------------------------|-----------|-----------------------|
| BEGIN_BACKUP_TIME | CHAR(24) | 백업 시작 일시 |
| END_BACKUP_TIME | CHAR(24) | 백업 완료 일시 |
| INCREMENTAL_BACKUP_CHUNK_COUNT | INTEGER | Incremental chunk의 크기 |
| BACKUP_TARGET | INTEGER | 백업 대상 |
| BACKUP_LEVEL | INTEGER | 백업 레벨 |
| BACKUP_TYPE | INTEGER | 백업 유형 |
| TABLESPACE_ID | INTEGER | 백업 대상 테이블스페이스 ID |
| FILE_ID | INTEGER | 백업 대상 데이터파일 ID |
| BACKUP_TAG | CHAR(128) | 백업 태그 이름 |
| BACKUP_FILE | CHAR(512) | 백업 파일 |

칼럼 정보

BEGIN_BACKUP_TIME

백업이 시작된 일시를 나타낸다. 'YYYY-MM-DD HH:MM:SS'의 형식으로 표시된다.

END_BACKUP_TIME

백업이 완료된 일시를 나타낸다. 'YYYY-MM-DD HH:MM:SS'의 형식으로 표시된다.

INCREMENTAL_BACKUP_CHUNK_COUNT

레벨 0 증분 백업의 경우, 항상 0으로 표시된다.

레벨 1 증분 백업의 경우, incremental chunk의 크기를 나타낸다.

incremental chunk에 대해서는 INCREMENTAL_BACKUP_CHUNK_SIZE 프로퍼티의 설명을 참고하라.

BACKUP_TARGET

백업 대상을 나타낸다.

1: 데이터베이스

2: 테이블스페이스

BACKUP_LEVEL

백업 레벨을 나타낸다.

1: 레벨 0

2: 레벨 1

BACKUP_TYPE

백업 유형을 나타낸다.

1: 전체 백업

2: 차등 증분 백업

4: 누적 증분 백업

TABLESPACE_ID

백업된 데이터파일이 속한 테이블스페이스의 ID를 나타낸다.

FILE_ID

백업된 데이터파일의 ID를 나타낸다.

BACKUP_TAG

증분 백업 수행시 사용된 백업 태그 이름을 나타낸다.

BACKUP_FILE

백업 파일 이름을 포함한 전체 경로를 나타낸다.

V\$BUFFPAGEINFO

버퍼 관리자가 관리하는 버퍼 프레임의 페이지 타입별 주요 연산들에 대한 통계치를 보여준다.

| Column name | Type | Description |
|-----------------|-------------|-------------------------|
| PAGE_TYPE | VARCHAR(21) | 페이지 타입 |
| READ_PAGE_COUNT | BIGINT | DISK I/O (READ)를 유발한 횟수 |
| GET_PAGE_COUNT | BIGINT | 버퍼 프레임을 요구한 횟수 |
| FIX_PAGE_COUNT | BIGINT | 버퍼 프레임에 고정(fix)한 횟수 |

| Column name | Type | Description |
|-------------------|--------|------------------------|
| CREATE_PAGE_COUNT | BIGINT | 새로운 버퍼 프레임을 요구한 횟수 |
| HIT_RATIO | DOUBLE | 버퍼 프레임 적중률 (hit ratio) |

칼럼 정보

PAGE_TYPE

버퍼 페이지 타입을 나타내며, 다음과 같은 페이지 타입이 있다.

| PAGE_TYPE | Description |
|-----------------------|---|
| PAGE UNFORMAT | 포맷되지 않은 페이지 |
| PAGE FORMAT | 포맷된 페이지 |
| PAGE INDEX META BTREE | B-트리 인덱스에 대한 메타 정보가 쓰여진 페이지 |
| PAGE INDEX META RTREE | R-트리 인덱스에 대한 메타 정보가 쓰여진 페이지 |
| PAGE INDEX BTREE | B-트리 인덱스 노드가 쓰여진 페이지 |
| PAGE INDEX RTREE | R-트리 인덱스 노드가 쓰여진 페이지 |
| PAGE TABLE | 테이블 레코드가 저장된 페이지 |
| PAGE TEMP TABLE META | 한 임시 테이블에 대한 메타 정보가 저장된 페이지 |
| PAGE TEMP TABLE DATA | 임시 테이블에 저장된 레코드가 쓰여진 페이지 |
| PAGE TSS | 트랜잭션의 상태에 대한 정보가 쓰여진 페이지. 여러 트랜잭션 상태 슬롯 (Transaction Status Slots, TSS)이 한 페이지에 저장될 수 있다. |
| PAGE UNDO | 언두 정보가 저장된 페이지. 한 페이지에 여러 언두 레코드가 저장될 수 있다. |
| PAGE LOB DATA | LOB 타입 데이터가 저장된 페이지. 한 페이지는 한 개의 LOB칼럼만 담을 수 있다. 한 개의 LOB칼럼은 여러 페이지에 걸쳐서 저장될 수 있다. |
| PAGE LOB INODE | 특정 크기를 초과하는 LOB 데이터와 관련된 인덱스 노드가 저장된 페이지 |
| PAGE FMS SEGHDR | 한 개의 FMS 헤더가 저장된 페이지 |
| PAGE FMS EXTDIR | 한 개의 FMS extent directory 가 저장된 페이지 |
| PAGE TMS SEGHDR | 한 개의 TMS 헤더가 저장된 페이지 |

| PAGE_TYPE | Description |
|----------------------|--|
| PAGE TMS LFBMP | 한 개의 TMS 리프 (leaf) 비트맵 노드가 저장된 페이지 |
| PAGE TMS ITBMP | 한 개의 TMS 중간 비트맵 노드가 저장된 페이지 |
| PAGE TMS RTBMP | 한 개의 TMS 루트 비트맵 노드가 저장된 페이지 |
| PAGE TMS EXTDIR | 한 개의 TMS extent directory 가 저장된 페이지 |
| PAGE CMS SEGHDR | 한 개의 CMS 헤더가 저장된 페이지 |
| PAGE CMS EXTDIR | 한 개의 CMS extent directory 가 저장된 페이지 |
| PAGE FEBT FSB | 한 개의 데이터파일 헤더가 저장된 페이지 |
| PAGE FEBT EGH | 데이터파일 내의 확장 그룹 헤더가 저장된 페이지. 한 페이지는 하나의 헤더만 저장할 수 있다. |
| PAGE LOB META | LOB 데이터 칼럼에 대한 메타 정보가 쓰여진 페이지 |
| PAGE HV TEMP NODE | 해쉬 값 기반의 임시 인덱스 노드가 저장된 페이지 |

READ_PAGE_COUNT

서버 구동 이후부터 현재까지 PAGE_TYPE에 해당하는 버퍼 프레임들에 DISK I/O (READ)를 유발시킨 총 횟수를 나타낸다. 0 이상의 값을 갖는다.

GET_PAGE_COUNT

서버 구동 이후부터 현재까지 버퍼 관리자에게 데이터 쓰거나 읽기 목적으로 PAGE_TYPE에 해당하는 버퍼 프레임들을 요구한 총 횟수를 나타낸다. 0 이상의 값을 갖는다.

FIX_PAGE_COUNT

서버 구동 이후부터 현재까지 버퍼 관리자에게 데이터 쓰거나 읽기를 목적으로 PAGE_TYPE에 해당하는 버퍼 프레임들을 고정(Fix)한 총 횟수를 나타낸다. 0 이상의 값을 갖는다.

CREATE_PAGE_COUNT

서버 구동 이후부터 현재까지 버퍼 관리자에게 PAGE_TYPE에 해당하는 새로운 버퍼 프레임들을 요구한 총 횟수를 나타낸다. 0 이상의 값을 갖는다.

HIT_RATIO

서버 구동 이후부터 현재까지 이 버퍼에 대한 적중률 (hit ratio)을 나타낸다. 이 값은 $(GET_PAGE_COUNT + FIX_PAGE_COUNT - READ_PAGE_COUNT) / (GET_PAGE_COUNT + FIX_PAGE_COUNT)$ 로 구해진다.

예제

서버 구동 이후 버퍼에서 관리된 페이지 타입별 주요 연산들의 누적치를 확인한다.

```

iSQL> select * from v$buffpageinfo;
PAGE_TYPE          READ_PAGE_COUNT    GET_PAGE_COUNT
-----
FIX_PAGE_COUNT      CREATE_PAGE_COUNT  HIT_RATIO

```

```

-----
PAGE UNFORMAT          0          0
0                      0          0
PAGE FORMAT            0          0
0                      0          0
PAGE INDEX META BTREE  4          0
4                      0          0
PAGE INDEX META RTREE  0          0
0                      0          0
PAGE INDEX BTREE       12         0
12                     0          0
PAGE INDEX RTREE       0          0
0                      0          0
PAGE TABLE            0          0
0                      0          0
PAGE TEMP TABLE META  0          0
0                      0          0
PAGE TEMP TABLE DATA 0          0
0                      0          0
PAGE TSS                0          0
0                      0          0
PAGE UNDO               0          0
0                      0          0
PAGE LOB DATA          0          0
0                      0          0
PAGE LOB INODE          0          0
0                      0          0
PAGE FMS SEGHDR         0          0
0                      0          0
PAGE FMS EXTDIR         0          0
0                      0          0
PAGE TMS SEGHDR         5          19
4                      0          73.6842105263158
PAGE TMS LFBMP          0          0
0                      0          0
PAGE TMS ITBMP          0          0
0                      0          0
PAGE TMS RTBMP          0          0
0                      0          0
PAGE TMS EXTDIR         0          0
0                      0          0
PAGE CMS SEGHDR         0          1536
0                      512         100
PAGE CMS EXTDIR         0          0
0                      0          0
PAGE FEBT FSB           2          1024
515                     2          99.8046875
PAGE FEBT EGH           0          512
0                      4          100
PAGE LOB META           0          0
0                      0          0
PAGE HV TEMP NODE       0          0
0                      0          0
26 rows selected.

```

V\$BUFFPOOL_STAT

버퍼 풀 적중률과 버퍼 풀 내의 버퍼 제어 블록 (Buffer Control Block, BCB) 개수를 포함하여, 버퍼 풀 관련 통계 정보를 보여준다.

| Column name | Type | Description |
|------------------------|---------|-----------------------------|
| ID | INTEGER | 버퍼 풀 식별자 |
| POOL_SIZE | INTEGER | 버퍼 풀 내의 페이지 개수 |
| PAGE_SIZE | INTEGER | 페이지 크기 (bytes) |
| HASH_BUCKET_COUNT | INTEGER | 해쉬 테이블의 버킷 개수 |
| HASH_CHAIN_LATCH_COUNT | INTEGER | 해쉬 테이블에 사용되는 체인 래치 개수 |
| LRU_LIST_COUNT | INTEGER | LRU 리스트 개수 |
| PREPARE_LIST_COUNT | INTEGER | 버퍼 풀의 Prepare 리스트 개수 |
| FLUSH_LIST_COUNT | INTEGER | 버퍼 풀의 플러시 리스트 개수 |
| CHECKPOINT_LIST_COUNT | INTEGER | 버퍼 풀의 체크포인트 리스트 개수 |
| VICTIM_SEARCH_COUNT | INTEGER | LRU 리스트에서 victim 검색 개수 |
| HASH_PAGES | INTEGER | 현재 해쉬 테이블에 삽입된 페이지 개수 |
| HOT_LIST_PAGES | INTEGER | 현재 LRU hot 리스트에 있는 페이지 개수 |
| COLD_LIST_PAGES | INTEGER | 현재 LRU cold 리스트에 있는 페이지 개수 |
| PREPARE_LIST_PAGES | INTEGER | 현재 Prepare 리스트에 있는 페이지 개수 |
| FLUSH_LIST_PAGES | INTEGER | 현재 플러시 리스트에 있는 페이지 개수 |
| CHECKPOINT_LIST_PAGES | INTEGER | 현재 체크포인트 리스트에 있는 페이지 개수 |
| FIX_PAGES | BIGINT | 래치 없이 페이지 고정을 요청한 누적 횟수 |
| GET_PAGES | BIGINT | 래치를 획득하면서 페이지를 요청한 누적 횟수 |
| READ_PAGES | BIGINT | 페이지 요청시 디스크에서 페이지를 읽은 누적 횟수 |
| CREATE_PAGES | BIGINT | 새로운 페이지를 생성한 누적 횟수 |
| HIT_RATIO | DOUBLE | 시스템 구동 후부터 버퍼 풀에서 누적 적중률 |
| HOT_HITS | BIGINT | LRU hot 리스트에 접근된 누적 횟수 |
| COLD_HITS | BIGINT | LRU cold 리스트에 접근된 누적 횟수 |
| PREPARE_HITS | BIGINT | Prepare 리스트에 접근된 누적 횟수 |
| FLUSH_HITS | BIGINT | 플러시 리스트에 접근된 누적 횟수 |

| Column name | Type | Description |
|-----------------------|---------|---|
| OTHER_HITS | BIGINT | 어떤 리스트에도 속하지 않은 버퍼에 접근된 누적 횟수 |
| PREPARE_VICTIMS | BIGINT | Prepare 리스트에서 교체 대상을 찾은 누적 횟수 |
| LRU_VICTIMS | BIGINT | LRU 리스트에서 교체 대상을 찾은 누적 횟수 |
| VICTIM_FAILS | BIGINT | 교체 대상 검색에 실패한 횟수 |
| PREPARE_AGAIN_VICTIMS | BIGINT | LRU 리스트에서 교체 대상 찾기를 실패한 후, 다시 prepare 리스트에서 교체 대상 버퍼를 찾은 누적 횟수 |
| VICTIM_SEARCH_WARP | BIGINT | Prepare 리스트와 LRU 리스트에서 교체 대상 찾기를 실패한 후 다음 Prepare 리스트로 검색 대상을 옮긴 횟수 |
| LRU_SEARCHES | BIGINT | LRU 리스트에서 검색한 버퍼의 누적 개수 |
| LRU_SEARCHES_AVG | INTEGER | 교체 대상을 검색한 평균 버퍼 수 |
| LRU_TO_HOTS | BIGINT | LRU 리스트에서 hot 영역으로 버퍼 제어 블록 (BCB)을 옮긴 누적 횟수 |
| LRU_TO_COLDS | BIGINT | LRU 리스트에서 cold 영역으로 BCB를 옮긴 누적 횟수 |
| LRU_TO_FLUSHES | BIGINT | LRU 리스트에서 플러시 리스트로 BCB를 옮긴 누적 횟수 |
| HOT_INSERTIONS | BIGINT | LRU hot 리스트에 삽입된 누적 횟수 |
| COLD_INSERTIONS | BIGINT | LRU cold 리스트에 삽입된 누적 횟수 |
| DB_SINGLE_READ_PERF | DOUBLE | 한 개의 데이터 페이지 요청 시, 초당 디스크로부터 읽은 평균 바이트 수 |
| DB_MULTI_READ_PERF | DOUBLE | 여러 데이터 페이지가 동시에 디스크의 데이터파일에서 읽혀질 때, 초당 읽은 평균 바이트 수 |

칼럼 정보

ID

버퍼 풀 고유 번호를 나타낸다. 현재 다중 버퍼 풀을 지원하지 않기 때문에 이 값은 항상 0이다.

POOL_SIZE

버퍼 풀의 페이지 개수이다. $POOL_SIZE * PAGE_SIZE$ 는 프로퍼티 `BUFFER_AREA_SIZE`의 크기와 같다.

PAGE_SIZE

현재 버퍼 풀에서 사용되는 페이지의 크기를 나타낸다. 현재는 다중 버퍼 풀을 지원하지 않기 때문에 8192바이트로 고정되어 있다.

HASH_BUCKET_COUNT

해쉬 테이블의 버킷 개수를 나타낸다. 프로퍼티 BUFFER_HASH_BUCKET_DENSITY에 의해 결정된다. 서버 구동 중에는 변경할 수 없다. 이 값이 클수록 해쉬 버킷 리스트의 탐색 비용이 감소된다.

HASH_CHAIN_LATCH_COUNT

해쉬 테이블에 사용되는 체인 래치의 개수를 나타낸다. 이 값이 클수록 해쉬 탐색시 발생할 수 있는 래치 경쟁이 줄어든다.

LRU_LIST_COUNT

버퍼 풀의 LRU 리스트 개수를 나타낸다.

PREPARE_LIST_COUNT

버퍼 풀의 prepare 리스트 개수를 나타낸다.

FLUSH_LIST_COUNT

버퍼 풀의 플러시 리스트 개수이다. 버퍼에 올라와 있는 페이지 중 수정되어 디스크에 반영해야 할 페이지가 플러시 리스트에 삽입된다.

CHECKPOINT_LIST_COUNT

버퍼 풀의 체크포인트 리스트 개수를 나타낸다.

VICTIM_SEARCH_COUNT

LRU 리스트에서 교체 대상을 검색할 때 몇 개까지 검색할지를 나타낸다. 명시된 값만큼 검색해도 교체 대상을 찾지 못하면 플러시가 prepare 리스트에 clean 버퍼가 삽입될 때까지 대기한다.

HASH_PAGES

해쉬 테이블에 삽입된 버퍼 수를 나타낸다. 이 값은 현재 사용중인 버퍼의 수를 의미한다.

HOT_LIST_PAGES

LRU hot 리스트에 존재하는 버퍼 수를 나타낸다.

COLD_LIST_PAGES

LRU cold 리스트에 존재하는 버퍼 수를 나타낸다.

PREPARE_LIST_PAGES

prepare 리스트에 존재하는 버퍼 수를 나타낸다. 이 값이 0이면 교체 대상을 얻기 위해 LRU 리스트를 조회한다.

FLUSH_LIST_PAGES

플러시 리스트에 존재하는 버퍼 수를 나타낸다. 값이 크면 플러시할 버퍼가 많다는 의미이다.

CHECKPOINT_LIST_PAGES

체크포인트 리스트에 존재하는 버퍼 수를 나타낸다. 이 값은 갱신된 페이지의 수를 의미한다.

FIX_PAGES

래치 획득없이 페이지를 요청한 횟수이다. 시스템 구동 후부터 누적된 횟수이다.

GET_PAGES

페이지 래치 획득과 함께 요청된 횟수를 나타낸다.

READ_PAGES

페이지 요청 시 디스크에서 페이지를 읽은 누적 횟수이다. 버퍼 miss 횟수와 동일한 의미이다.

CREATE_PAGES

새로운 페이지에 데이터를 삽입하기 위해 페이지를 할당한 누적 횟수이다. 페이지 생성은 실제로 디스크 I/O를 수반하지는 않는다.

HIT_RATIO

버퍼 풀의 누적 적중률 (hit ratio)을 나타낸다. 이 값은 $(GET_PAGES + FIX_PAGES - READ_PAGES) / (GET_PAGES + FIX_PAGES)$ 으로 계산할 수 있다. 이 값이 작으면 메모리 버퍼 대신에 디스크로부터 읽기(read page) 횟수가 많다는 것이다. 즉 이 값이 작으면, 시스템이 빠른 질의 처리를 못하고 있다는 것을 보여준다.

HOT_HITS

LRU hot 리스트에서 hit가 발생한 누적 횟수를 나타낸다. Hit란 페이지 요청시 해당 페이지가 이미 버퍼에 있어서 디스크로부터 읽기를 유발시키지 않음을 의미한다.

COLD_HITS

LRU cold 리스트에서 hit가 발생한 누적 횟수를 나타낸다.

PREPARE_HITS

prepare 리스트에서 hit가 발생한 누적 횟수를 나타낸다.

FLUSH_HITS

플러시 리스트에서 hit가 발생한 누적 횟수를 나타낸다.

OTHER_HITS

순간적으로 어떤 리스트에도 속하지 않은 버퍼에 hit 발생한 횟수를 나타낸다. hit가 발생한 버퍼는 항상 어떤 리스트에 존재해야 하는 것은 아니다.

PREPARE_VICTIMS

prepare 리스트에서 교체 대상 버퍼를 찾은 누적 횟수를 나타낸다.

LRU_VICTIMS

LRU 리스트에서 교체 대상 버퍼를 찾은 누적 횟수를 나타낸다.

VICTIM_FAILS

교체 대상 버퍼 찾기에 실패한 누적 횟수를 나타낸다. 이 값은 $PREPARE_AGAIN_VICTIMS$

- $VICTIM_SEARCH_WARP$ 로 계산할 수 있다. $PREPARE_VICTIMS + LRU_VICTIMS + VICTIM_FAILS$ 는 버퍼 풀에서 발생한 총 교체 횟수이다.

PREPARE_AGAIN_VICTIMS

교체 대상 버퍼 찾기에 실패한 후 prepare 리스트에 버퍼가 삽입되기를 대기한다. 이 때 대기 중에 clean 버퍼가 삽입되어 이를 교체 대상으로 선정하게 된 횟수를 나타낸다.

VICTIM_SEARCH_WARP

prepare 리스트에 일정 시간 대기한 후에도 교체 대상 버퍼를 선정하지 못한 경우 다음 prepare 리스트로 넘어가서 교체 대상 버퍼를 찾는 누적 횟수를 나타낸다.

LRU_SEARCHS

LRU 리스트에서 교체 대상 버퍼를 검색한 누적 버퍼 개수를 나타낸다.

LRU_SEARCHS_AVG

교체 대상 검색시 탐색 버퍼의 평균 개수를 나타낸다.

LRU_TO_HOTS

LRU 리스트에서 hot영역으로 옮겨진 버퍼의 누적 개수를 나타낸다.

LRU_TO_COLDS

LRU 리스트에서 cold영역으로 옮겨진 버퍼의 누적 개수를 나타낸다.

LRU_TO_FLUSH

LRU 리스트에서 플러시 리스트로 옮겨진 버퍼의 누적 개수를 나타낸다.

HOT_INSERTIONS

LRU hot 리스트에 삽입된 누적 버퍼 개수를 나타낸다.

COLD_INSERTIONS

LRU cold 리스트에 삽입된 누적 버퍼 개수를 나타낸다.

DB_SINGLE_READ_PERF

디스크 테이블에 대해 FETCH, INSERT, UPDATE 및 DELETE 수행 시, Altibase는 하나의 데이터 페이지를 데이터 파일에서 읽어서 메모리 버퍼에 저장한다. 이 값은 이런 작업 과정 중 초당 디스크에서 읽은 평균 바이트 수이다. (단위: kB/sec)

DB_MULTI_READ_PERF

일명 “full 스캔”이라 불리는 작업 즉, 한 디스크 테이블 전체를 스캔하는 작업 수행 시, Altibase는 여러 데이터 페이지를 동시에 디스크에서 읽어서 메모리 버퍼에 저장한다. 이 값은 이 작업 과정 중 초당 디스크에서 읽은 평균 바이트 수이다. (단위: kB/sec)

V\$CATALOG

데이터베이스에 존재하는 테이블의 구조 정보를 보여준다.

| Column name | Type | Description |
|-------------|---------|-------------|
| TABLE_OID | BIGINT | 테이블의 객체 식별자 |
| COLUMN_CNT | INTEGER | 테이블의 칼럼 개수 |

| Column name | Type | Description |
|---------------------|---------|---------------------------------------|
| COLUMN_VAR_SLOT_CNT | INTEGER | 칼럼 정보를 저장하기 위해 사용된 Variable Slot의 개수 |
| INDEX_CNT | INTEGER | 테이블의 인덱스 개수 |
| INDEX_VAR_SLOT_CNT | INTEGER | 인덱스 정보를 저장하기 위해 사용된 Variable Slot의 개수 |

칼럼 정보

TABLE_OID

테이블의 정보를 가지는 헤더 (Header)의 물리적인 위치를 나타낸다.

COLUMN_CNT

테이블의 정보를 가지는 헤더 (Header)의 물리적인 위치를 나타낸다.

COLUMN_VAR_SLOT_CNT

테이블의 칼럼 정보를 저장하기 위해 사용된 Variable Slot의 개수.

INDEX_CNT

테이블의 인덱스 개수이다.

INDEX_VAR_SLOT_CNT

인덱스에 대한 정보를 저장하기 위해 사용된 Variable Slot의 개수이다.

V\$DATABASE

메모리 데이터베이스에 대한 내부 정보를 보여준다.

| Column name | Type | Description |
|-------------------|--------------|--------------------------------|
| DB_NAME | VARCHAR(128) | 데이터베이스 이름 |
| PRODUCT_SIGNATURE | VARCHAR(512) | 제품 바이너리와 빌드 환경을 나타내는 제품 고유 스트링 |
| DB_SIGNATURE | VARCHAR(512) | 고유한 데이터베이스 식별 스트링 |
| VERSION_ID | INTEGER | 데이터베이스 버전 |
| COMPILE_BIT | INTEGER | 제품이 32 또는 64비트로 컴파일 되었는지 나타냄 |
| ENDIAN | BIGINT | Endian 정보 |
| LOGFILE_SIZE | BIGINT | 로그파일 크기 |
| TX_TBL_SIZE | INTEGER | 트랜잭션 테이블 크기 |
| LAST_SYSTEM_SCN | VARCHAR(29) | 내부 용도 |
| INIT_SYSTEM_SCN | VARCHAR(29) | 내부 용도 |

| Column name | Type | Description |
|----------------------|--------------|------------------------|
| DURABLE_SYSTEM_SCN | VARCHAR(29) | 저장된 시스템 SCN 값 |
| MEM_MAX_DB_SIZE | VARCHAR(256) | 메모리 데이터베이스의 최대 크기 |
| MEM_ALLOC_PAGE_COUNT | BIGINT | 할당된 페이지 총 개수 |
| MEM_FREE_PAGE_COUNT | BIGINT | 사용 가능한 페이지 총 개수 |
| MAX_ACCESS_FILE_SIZ | VARCHAR(12) | 데이터베이스에 생성가능한 최대 파일 크기 |

칼럼 정보

DB_NAME

메모리 데이터베이스의 이름을 나타낸다.

PRODUCT_SIGNATURE

Altibase 제품이 가지는 고유한 제품 정보를 나타낸다.

DB_SIGNATURE

고유한 데이터베이스 식별 스트링이다.

VERSION_ID

Altibase 저장관리자가 유지하는 고유 버전번호를 나타낸다.

COMPILE_BIT

현재 생성된 데이터베이스가 32비트인지 혹은 64비트인지 표현한다.

ENDIAN

현재 생성된 데이터베이스의 Endian을 나타낸다.

- 0: little endian
- 1: big endian

LOGFILE_SIZE

현재 생성된 데이터베이스에서 사용하는 로그 파일의 크기를 바이트 단위로 나타낸다.

TX_TBL_SIZE

트랜잭션 테이블의 크기를 나타낸다.

MEM_MAX_DB_SIZE

메모리 데이터베이스 공간의 확장가능한 최대 크기를 바이트 단위로 나타낸다.

MEM_ALLOC_PAGE_COUNT

현재 메모리 데이터베이스에 할당된 총 페이지 개수를 나타낸다. 이는 확장가능한 최대 크기까지 고려하지 않으며, 현재 메모리 데이터베이스 공간 크기에 대해서만 고려한다. 그러므로, 현재 메모리 데이터베이스 공간의 크기는 MEM_ALLOC_PAGE_COUNT와 MEM_FREE_PAGE_COUNT의 합에 페이지 크기 (메모리 데이터베이스의 페이지 크기는 32KB)를 곱하여 계산할 수 있다.

MEM_FREE_PAGE_COUNT

현재 메모리 데이터베이스 공간에서 할당가능한 페이지 개수를 나타낸다. 현재 할당된 페이지는 포함되지 않는다. 이는 확장가능한 최대 크기까지 고려하지 않으며, 현재 메모리 데이터베이스 공간 크기에 대해서만 고려한다. 그러므로, 현재 메모리 데이터베이스 공간의 크기는 MEM_ALLOC_PAGE_COUNT와 MEM_FREE_PAGE_COUNT의 합에 페이지 크기 (32KB)를 곱하여 표현할 수 있다.

DURABLE_SYSTEM_SCN

데이터베이스 공간에 저장된 시스템 SCN의 값을 나타낸다.

V\$DATAFILES

테이블스페이스에서 사용하는 데이터 파일의 정보를 보여준다.

| Column name | Type | Description |
|-------------------|--------------|---------------------|
| ID | INTEGER | 데이터 파일 식별자 |
| NAME | VARCHAR(256) | 데이터 파일 이름 |
| SPACEID | INTEGER | 테이블스페이스 식별자 |
| OLDEST_LSN_LFGID | INTEGER | 사용하지 않음(0) |
| OLDEST_LSN_FILENO | INTEGER | 아래 참조 |
| OLDEST_LSN_OFFSET | INTEGER | 아래 참조 |
| CREATE_LSN_LFGID | INTEGER | 사용하지 않음(0) |
| CREATE_LSN_FILENO | INTEGER | 아래 참조 |
| CREATE_LSN_OFFSET | INTEGER | 아래 참조 |
| SM_VERSION | INTEGER | 버전 정보 |
| NEXTSIZE | BIGINT | 데이터 파일 확장 시 증가할 크기 |
| MAXSIZE | BIGINT | 최대 크기 |
| INITSIZE | BIGINT | 초기 크기 |
| CURRSIZE | BIGINT | 현재 크기 |
| AUTOEXTEND | INTEGER | 자동 확장 플래그 |
| IOCOUNT | INTEGER | 현재 진행 중인 I/O 작업의 개수 |
| OPENED | INTEGER | 현재 사용 중인지 여부 |
| MODIFIED | INTEGER | 데이터 파일 수정 여부 |
| STATE | INTEGER | 파일의 상태 |
| MAX_OPEN_FD_COUNT | INTEGER | 열 수 있는 최대 FD 개수 |
| CUR_OPEN_FD_COUNT | INTEGER | 열린 FD 개수 |

칼럼 정보

ID

데이터 파일의 식별자를 나타낸다. 아이디는 파일이 생성된 순서대로 순차적으로 부여되어 같은 아이디가 중복되는 일은 없다.

NAME

데이터 파일의 물리적 경로와 이름을 나타낸다.

SPACEID

데이터 파일이 속한 테이블스페이스의 식별자를 나타낸다.

OLDEST_LSN_FILENO

데이터 파일에 페이지를 플러시한 마지막 체크포인트 시점에 버퍼에 올라와 수정되었던 페이지 중 가장 오래된 페이지의 LSN 값의 파일 번호 부분을 나타낸다.

OLDEST_LSN_OFFSET

데이터 파일에 페이지를 플러시한 마지막 체크포인트 시점에 버퍼에 올라와 수정되었던 페이지 중 가장 오래된 페이지의 LSN값의 offset부분을 나타낸다.

CREATE_LSN_FILENO

데이터 파일이 생성된 시점의 LSN 값의 파일 번호 부분을 나타낸다.

CREATE_LSN_OFFSET

데이터 파일이 생성된 시점의 LSN 값의 offset부분을 나타낸다.

SM_VERSION

데이터 파일을 생성한 바이너리의 버전을 나타낸다.

NEXTSIZE

데이터 파일의 autoextend 속성이 on인 경우, 공간 부족 시 데이터 파일은 이 크기만큼 확장된다. 표시되는 값은 페이지 개수이다 (1페이지 = 8kB).

MAXSIZE

데이터 파일의 autoextend 속성이 on인 경우, 공간 부족 시 데이터 파일이 확장될 수 있는 최대 크기를 나타낸다. 표시되는 값은 페이지 개수이다 (1페이지 = 8kB).

INITSIZE

데이터 파일이 최초에 생성된 크기를 나타낸다. 표시되는 값은 페이지 개수이다 (1페이지 = 8kB).

CURRSIZE

데이터 파일의 현재 크기를 나타낸다. 표시되는 값은 페이지 개수이다 (1페이지 = 8kB).

AUTOEXTEND

데이터 파일의 공간이 부족할 때 자동 확장될 지 여부를 나타낸다.

- 0: 자동 확장 안함.
- 1: 자동 확장

IOCOUNT

데이터 파일에 현재 진행 중인 I/O작업의 개수를 나타낸다. 데이터 파일에 I/O가 진행 중이 아니라면, 다음 데이터 파일이 오픈될 수 있다.

OPENED

데이터 파일이 현재 오픈되었는지 나타낸다.

- 0: 닫혀 있음
- 1: 열려 있음

MODIFIED

데이터 파일이 수정되었는지 나타낸다. 데이터 파일에 페이지를 플러시하고 동기화 (synchronization)하지 않으면 이 값이 1이 된다. 플러시 후에 데이터 파일에 동기화를 수행하면 이 값이 0이 된다.

STATE

데이터 파일의 상태를 나타낸다.

- 1: 오프라인 (offline)
- 2: 온라인 (online)
- 6: 백업 중
- 128: 삭제 (dropped)

MAX_OPEN_FD_COUNT

현재 디스크 데이터 파일에서 I/O가 발생할 때 열 수 있는 최대 FD (File Descriptor) 개수

CUR_OPEN_FD_COUNT

현재 디스크 데이터 파일에서 열린 FD (File Descriptor) 개수

V\$DATATYPE

Altibase에서 지원하는 데이터 타입의 정보를 보여준다.¹⁴

[¹⁴] 이 성능 뷰에 저장된 값은 ODBC SQLGetTypeInfo() 함수에서 조회하는 값이다. 자세한 내용은 *ODBC Reference*를 참고한다.

| Column name | Type | Description |
|----------------|-------------|----------------------------------|
| TYPE_NAME | VARCHAR(40) | DBMS에서 지원하는 데이터 타입 이름 |
| DATA_TYPE | SMALLINT | DBMS에서 지원하는 데이터 타입의 내부 정의 값 |
| ODBC_DATA_TYPE | SMALLINT | 데이터 타입에 대응하는 ODBC SQL 데이터 타입 식별자 |
| COLUMN_SIZE | INTEGER | 해당 타입에 대한 최대 칼럼 크기. |
| LITERAL_PREFIX | VARCHAR(4) | 해당 데이터 타입의 리터럴에 대한 접두부로 인식하는 문자 |
| LITERAL_SUFFIX | VARCHAR(4) | 해당 데이터 타입의 리터럴에 대한 접미부로 인식하는 문자. |

| Column name | Type | Description |
|--------------------|-------------|---|
| CREATE_PARAM | VARCHAR(20) | SQL에서 데이터 타입 정의시 괄호로 표현되는 매개 변수 키워드 목록 |
| NULLABLE | SMALLINT | 데이터 타입의 NULL 값 허용 여부 |
| CASE_SENSITIVE | SMALLINT | 대/소문자 구분 여부 |
| SEARCHABLE | SMALLINT | WHERE절에서 데이터 타입 사용 방법 |
| UNSIGNED_ATTRIBUTE | SMALLINT | 데이터 타입의 부호 여부 |
| FIXED_PREC_SCALE | SMALLINT | 데이터 타입이 고정형인지 나타낸다 |
| AUTO_UNIQUE_VALUE | SMALLINT | 향후 확장 예정 |
| LOCAL_TYPE_NAME | VARCHAR(40) | 데이터 타입에 대한 로컬화된 (자국어) 이름 |
| MINIMUM_SCALE | SMALLINT | 허용가능한 최소 소수 자릿수 |
| MAXIMUM_SCALE | SMALLINT | 허용가능한 최대 소수 자릿수 |
| SQL_DATA_TYPE | SMALLINT | SQL_DESC_TYPE에서 지원하는 SQL 데이터 타입 정의 값 |
| SQL_DATETIME_SUB | SMALLINT | datetime 또는 interval 타입의 하위 코드 |
| NUM_PREC_RADIX | INTEGER | 한 칼럼이 보유할수 있는 숫자의 최대 자리수를 계산하기 위해 필요한 비트수 |
| INTERVAL_PRECISION | SMALLINT | DATA_TYPE이 interval인 경우에 해당 데이터 타입에 표현할 수 있는 숫자의 최대 자리수 |

칼럼 정보

ODBC_DATA_TYPE

해당하는 데이터 타입에 대응하는 ODBC SQL 데이터 타입 식별자이다. 이에 대한 자세한 내용은 *ODBC Reference*의 부록 데이터 형을 참고한다.

COLUMN_SIZE

해당 타입에 대한 최대 칼럼 크기이다.

숫자형 타입의 경우 이 값은 타입 정의시에 주어진 Precision 값이다. 문자형 타입의 경우에 이 값은 타입 정의시에 주어진 길이 값이다. 날짜형 타입의 경우 이 값은 문자로 변환될 때 값을 표시하기 위해 필요한 총 문자 수이다.

LITERAL_PREFIX

해당 데이터 타입의 리터럴에 대한 접두부로 인식하는 문자이다. 리터럴 접두부를 적용할 수 없는 데이터 타입인 경우 이 값은 NULL이다.

LITERAL_SUFFIX

해당 데이터 타입의 리터럴에 대한 접미부로 인식하는 문자이다. 리터럴 접두부를 적용할 수 없는 데이터 타입인 경우 이 값은 NULL이다.

CREATE_PARAM

SQL에서 데이터 타입 정의시 괄호내에 표현되는 매개변수 키워드 목록으로 쉼표로 구분된다. 예를 들어 NUMBER(precision, scale) 표현되는 NUMBER의 경우, 괄호 안의 "precision, scale"이 이에 해당된다. 목록에서 키워드는 precision와 scale이다. 매개변수가 필요 없는 데이터 타입의 경우, 이 값은 NULL이다.

NULLABLE

데이터 타입이 NULL 값을 허용하는지를 나타낸다.

- 1: NULL 값을 허용한다.
- 0: NULL 값을 허용하지 않는다.

CASE_SENSITIVE

문자형 데이터 타입의 경우, 이 데이터 타입의 데이터를 정렬할 때 대/소문자를 구분하는지 나타낸다.

- 1: 대/소문자를 구분한다.
- 0: 대/소문자를 구분하지 않는다.

SEARCHABLE

WHERE 절에서 이 데이터 타입을 사용하는 방법을 나타낸다.

- 0: WHERE절에서 사용될 수 없다 (SQL_PRED_NONE).
- 1: WHERE절에서 사용될 수 있으나, LIKE와 함께 사용되어야 한다 (SQL_PRED_CHAR).
- 2: WHERE절에서 LIKE를 제외한 모든 비교 연산자들과 사용될 수 있다 (SQL_PRED_BASIC).
- 3: WHERE절에서 모든 비교 연산자들과 사용될 수 있다 (SQL_SEARCHABLE).

UNSIGNED_ATTRIBUTE

데이터 타입의 부호 여부를 나타낸다.

- 1: 해당 타입이 부호없는 (unsigned) 데이터 타입이다.
- 0: 해당 타입이 부호를 가지는 (signed) 데이터 타입이다.
- NULL: 해당 타입이 숫자형이 아니어서, 이 속성이 적용되지 않는다.

FIXED_PREC_SCALE

데이터 타입이 고정형인지 나타낸다. 해당 데이터 타입이 고정형 숫자 타입이고 항상 같은 정밀도 (precision)와 소수 자릿수 (scale)를 가지면 1 (SQL_TRUE), 그렇지 않은 경우 0 (SQL_FALSE)이다.

LOCAL_TYPE_NAME

데이터 타입에 대한 로컬화된 (자국어) 이름을 나타낸다. 로컬화된 이름이 없는 경우 NULL이다.

MINIMUM_SCALE

숫자형 데이터 타입의 경우, 허용가능한 최소 소수 자릿수이다. 고정 scale 타입일 경우 이 값이 존재하며, scale이 적용되지 않는 타입에 대해서는 이 값이 NULL이다.

MAXIMUM_SCALE

숫자형 데이터 타입의 경우, 허용가능한 최대 소수 자릿수이다. scale이 적용되지 않는 타입의 경우, 이 값은 NULL이다.

SQL_DATA_TYPE

ODBC의 SQL_DESC_TYPE에서 지원하는 SQL 데이터 타입이다. interval, datetime 데이터 타입을 제외한 다른 타입의 경우, ODBC_DATA_TYPE 값과 같다.

SQL_DATETIME_SUB

SQL_DATA_TYPE 값이 SQL_DATETIME 또는 SQL_INTERVAL인 경우 이 값은 datetime 또는 interval의 하위 코드이다. 데이터 타입이 datetime 또는 interval이 아닌 경우 이 값은 NULL이다.

NUM_PREC_RADIX

한 칼럼이 보유할 수 있는 최대 수를 계산하는데 필요한 비트수 또는 자릿수입니다.

INTERVAL_PRECISION

DATA_TYPE이 interval인 경우에 해당 데이터 타입에 표현할 수 있는 숫자의 최대 자릿수이다.

V\$DBA_2PC_PENDING

DBMS에 존재하는 분산 트랜잭션 중에서 현재 in-doubt 상태인 트랜잭션의 XID의 목록을 보여준다. 분산 트랜잭션에서 in-doubt 상태란 커밋할 준비가 된 상태에서 커밋 또는 롤백 명령을 받기 전까지의 트랜잭션 브랜치의 상태를 의미한다.

| Column name | Type | Description |
|---------------|--------------|--|
| LOCAL_TRAN_ID | BIGINT | 글로벌 트랜잭션 아이디 (GLOBAL_TX_ID)와 연계되어 있는 Altibase 내부의 트랜잭션 아이디 |
| GLOBAL_TX_ID | VARCHAR(256) | 글로벌 트랜잭션 아이디 |

칼럼 정보**LOCAL_TRAN_ID**

Altibase 내부의 트랜잭션 아이디로써 글로벌 트랜잭션 아이디와 연계된다.

GLOBAL_TX_ID

트랜잭션 브랜치에 할당한 고유한 트랜잭션 아이디이다. 이 값은 포맷 식별자(format identifier), 글로벌 트랜잭션 식별자(global transaction identifier) 및 브랜치 수식자(branch qualifier)를 포함한 문자열로 표시된다.

V\$DBLINK_ALTLINKER_STATUS

데이터베이스 링크를 위한 AltiLinker 프로세스의 상태 정보를 보여준다.

| Column name | Type | Description |
|--------------------------|--------------|---|
| STATUS | INTEGER | AltiLinker의 상태. 값의 의미는 칼럼 정보 참고. |
| SESSION_COUNT | INTEGER | Altibase와 AltiLinker 프로세스 사이의 세션인 링커 세션의 개수 |
| REMOTE_SESSION_COUNT | INTEGER | AltiLinker 프로세스와 원격 서버들 사이의 세션의 개수 |
| JVM_MEMORY_POOL_MAX_SIZE | INTEGER | JVM 상에서 AltiLinker를 위해 할당하는 메모리 풀의 최대 크기 |
| JVM_MEMORY_USAGE | BIGINT | JVM 상에서 AltiLinker 프로세스의 메모리 사용량 |
| START_TIME | VARCHAR(128) | AltiLinker 프로세스가 시작된 일시 |

칼럼 정보

STATUS

AltiLinker 프로세스의 상태를 나타낸다.

- 0 : AltiLinker 프로세스가 시작되지 않았거나 정상적인 수행이 불가능한 상태이다.
- 1 : AltiLinker 프로세스가 시작된 상태이다.
- 2 : AltiLinker 프로세스와 Altibase 서버 간에 링커 제어 세션(Linker Control Session)이 생성되어 AltiLinker가 정상적으로 수행 중인 상태이다.

V\$DBLINK_DATABASE_LINK_INFO

데이터베이스에 존재하는 데이터베이스 링크 객체에 대한 정보를 보여준다.

| Column name | Type | Description |
|-----------------|---------|---------------------|
| ID | INTEGER | 데이터베이스 링크 객체 식별자 |
| STATUS | INTEGER | 이 데이터베이스 링크 객체의 상태 |
| REFERENCE_COUNT | INTEGER | 이 데이터베이스 링크 객체 참조 수 |

칼럼 정보

STATUS

데이터베이스 링크 객체의 상태를 나타낸다.

- 1(CREATED): 메모리에 데이터베이스 링크 객체 생성이 완료
- 2(META): 메타 테이블에 데이터베이스 링크 객체 정보 등록
- 3(READY): 데이터베이스 링크 객체의 사용이 가능

REFERENCE_COUNT

이 데이터베이스 링크가 현재 참조되고 있는 횟수를 나타낸다.

V\$DBLINK_GLOBAL_TRANSACTION_INFO

현재 데이터베이스 링크를 통해 수행중인 글로벌 트랜잭션에 대한 정보를 나타낸다.

| Column name | Type | Description |
|--------------------------|---------|----------------------------------|
| TRANSACTION_ID | INTEGER | 현재 데이터베이스 링크를 사용하는 글로벌 트랜잭션의 식별자 |
| STATUS | INTEGER | 글로벌 트랜잭션의 현재 상태 |
| SESSION_ID | INTEGER | 글로벌 트랜잭션을 수행하고 있는 링커 데이터 세션의 ID |
| REMOTE_TRANSACTION_COUNT | INTEGER | 글로벌 트랜잭션 내에서 현재 수행중인 원격 트랜잭션의 개수 |
| TRANSACTION_LEVEL | INTEGER | 글로벌 트랜잭션의 실행 레벨 |
| GLOBAL_TRANSACTION_ID | INTEGER | 데이터베이스 링크를 사용하고 있는 글로벌 트랜잭션의 식별자 |

칼럼 정보**STATUS**

글로벌 트랜잭션의 현재 상태를 나타낸다.

- 0(NONE): 트랜잭션이 존재하지 않음
- 1(BEGIN): 트랜잭션이 시작됨
- 2(PREPARE_READY): 트랜잭션이 시작되었으나 현재 수행중인 원격 트랜잭션은 존재하지 않음
- 3(PREPARE_REQUEST): Simple transaction commit level 에서 AltiLinker 프로세스에 prepare를 요청한 상태
- 4(PREPARE_WAIT): Simple transaction commit level에서 모든 원격 트랜잭션에 대해 prepare 완료여부를 기다리는 상태
- 5(PREPARED): 모든 원격 트랜잭션의 prepare 완료
- 6(COMMIT_REQUEST): AltiLinker 프로세스로 commit 을 요청한 상태
- 7(COMMIT_WAIT): AltiLinker 프로세스로부터 commit 에 대한 응답을 기다리는 상태
- 8(COMMITTED): 트랜잭션 commit 완료
- 9(ROLLBACK_REQUEST): AltiLinker 프로세스로 rollback 을 요청한 상태
- 10(ROLLBACK_WAIT): AltiLinker 프로세스로부터 rollback 에 대한 응답을 기다리는 상태
- 11(ROLLBACKED): 트랜잭션 rollback 완료

TRANSACTION_LEVEL

0,1,2로 표시된다. 각 값에 대한 상세한 설명은 DBLINK_GLOBAL_TRANSACTION_LEVEL 프로퍼티의 내용을 참고하도록 한다.

V\$DBLINK_LINKER_CONTROL_SESSION_INFO

Altibase 서버와 AltiLinker 프로세스 사이의 제어 작업을 위해 유일하게 생성되는 링커 제어 세션의 상태 정보를 보여준다.

| Column name | Type | Description |
|-----------------|---------|-------------------------|
| STATUS | INTEGER | 링커 제어 세션의 상태 |
| REFERENCE_COUNT | INTEGER | 링커 제어 세션이 현재 참조되고 있는 횟수 |

칼럼 정보**STATUS**

이 링커 제어 세션의 현재 상태를 나타낸다.

- 0(NONE): 링커 제어 세션이 존재하지 않는 상태
- 1(CREATED): 링커 제어 세션이 생성 완료된 상태
- 2(CONNECTED): AltiLinker 프로세스와 링커 제어 세션이 연결된 상태
- 3(DISCONNECTED): AltiLinker 프로세스와 링커 제어 세션의 연결이 끊어진 상태
- 4(DESTROYED): 링커 제어 세션이 제거된 상태
- 5(LOCKED): 링커 제어 세션이 잠긴 상태
- 6(UNLOCKED): 링커 제어 세션의 잠금이 풀린 상태

V\$DBLINK_LINKER_DATA_SESSION_INFO

Altibase 서버와 AltiLinker 프로세스 사이의 데이터 작업을 수행하기 위해 생성되는 링커 데이터 세션들의 상태 정보를 보여준다.

| Column name | Type | Description |
|-----------------------|---------|-----------------------------|
| ID | INTEGER | 링커 데이터 세션 식별자 |
| STATUS | INTEGER | 링커 데이터 세션의 상태 |
| LOCAL_TRANSACTION_ID | INTEGER | 현재 세션에서 수행 중인 로컬 트랜잭션의 식별자 |
| GLOBAL_TRANSACTION_ID | INTEGER | 현재 세션에서 수행 중인 글로벌 트랜잭션의 식별자 |

칼럼 정보**STATUS**

이 링커 데이터 세션의 현재 상태를 나타낸다.

- 0(NONE): 링커 데이터 세션이 존재하지 않는 상태
- 1(CREATED): 링커 데이터 세션의 생성이 완료된 상태

- 2(CONNECTED): 링커 데이터 세션이 AltiLinker 프로세스와 연결된 상태
- 3(DISCONNECTED): 링커 데이터 세션과 AltiLinker 프로세스와의 연결이 끊어진 상태
- 4(DESTROYED): 링커 데이터 세션이 제거된 상태

V\$DBLINK_LINKER_SESSION_INFO

Altibase 서버와 AltiLinker 프로세스 간에 생성되는 링커 제어 세션(Linker Control Session)과 링커 데이터 세션(Linker Data Session)들이 얼마나 존재하는지 보여준다.

| Column name | Type | Description |
|--------------|------------|-----------------------------|
| SESSION_ID | INTEGER | 링커 세션 식별자 |
| STATUS | INTEGER | 링커 세션의 상태 |
| SESSION_TYPE | VARCHAR(7) | 링커 제어 세션인지 링커 데이터 세션인지를 나타냄 |

칼럼 정보

STATUS

이 링커 세션의 현재 상태를 나타낸다. 상태 값은 V\$DBLINK_LINKER_CONTROL_SESSION_INFO 성능 뷰와 V\$DBLINK_LINKER_DATA_SESSION_INFO 성능 뷰의 STATUS를 참고하도록 한다.

SESSION_TYPE

이 링커 세션이 링커 제어 세션인지 링커 데이터 세션인지를 나타낸다.

- CONTROL: 링커 제어 세션
- DATA: 링커 데이터 세션

V\$DBLINK_NOTIFIER_TRANSACTION_INFO

AltiLinker가 처리 중인 분산 트랜잭션의 정보를 보여준다.

| Column name | Type | Description |
|-----------------------|-------------|--------------------------------------|
| GLOBAL_TRANSACTION_ID | INTEGER | 데이터베이스 링크를 사용하는 트랜잭션의 식별자 |
| TRANSACTION_ID | INTEGER | 로컬 트랜잭션 식별자 |
| XID | VARCHAR(12) | 트랜잭션 브랜치 식별자 |
| TRANSACTION_RESULT | VARCHAR(10) | 해당 트랜잭션을 처리한 결과 (COMMIT/ROLLBACK) |
| TARGET_INFO | VARCHAR(40) | 데이터베이스 링크 객체가 접근할 원격서버의 이름 |

칼럼 정보

GLOBAL_TRANSACTION_ID

데이터베이스 링크를 사용하는 글로벌 트랜잭션의 식별자이다.

TRANSACTION_ID

글로벌 트랜잭션을 처리할 경우에 Altibase가 로컬 트랜잭션을 수행할 때 사용하는 내부 트랜잭션의 식별자이다.

XID

트랜잭션 브랜치에 할당한 고유한 트랜잭션 아이디이다. 이 값은 포맷 식별자(format identifier), 글로벌 트랜잭션 식별자(global transaction identifier), 브랜치 수식자(branch qualifier)를 문자열로 표시한다.

TRANSACTION_RESULT

해당 트랜잭션을 처리한 결과를 나타낸다.

- COMMIT: 트랜잭션을 COMMIT으로 처리한 경우
- ROLLBACK: 트랜잭션을 ROLLBACK으로 처리한 경우

TARGET_INFO

데이터베이스 링크 객체가 접근할 원격서버의 이름을 보여준다.

V\$DBLINK_REMOTE_STATEMENT_INFO

데이터베이스 링크를 사용했을 때, 원격 서버에 파생되어 발생한 질의문 정보를 보여준다

| Column name | Type | Description |
|-----------------------|----------------|----------------------------------|
| TRANSACTION_ID | INTEGER | 데이터베이스 링크를 사용하고 있는 트랜잭션의 식별자 |
| REMOTE_TRANSACTION_ID | INTEGER | 원격 서버에 발생한 트랜잭션 식별자 |
| STATEMENT_ID | BIGINT | 원격 서버에 발생한 구문(statement) 식별자 |
| QUERY | VARCHAR(32000) | 구문 (statement)에서 실행한 질의 내용 |
| GLOBAL_TRANSACTION_ID | INTEGER | 데이터베이스 링크를 사용하고 있는 글로벌 트랜잭션의 식별자 |

칼럼 정보

REMOTE_TRANSACTION_ID

원격 서버에 발생한 트랜잭션 식별자이다. 이 식별자는 실제 원격 서버 상의 트랜잭션 식별자가 아니라, 원격 서버에 트랜잭션을 생성할 때 Altibase가 자체적으로 부여한 식별자이다. 이 식별자는 관리 목적으로 생성된 것이므로, 그 값 자체에 의미를 둘 필요는 없다.

STATEMENT_ID

원격 서버에 발생한 구문 (statement) 식별자이다. 이 식별자는 실제 원격 서버에서 생성된 구문 식별자가 아니라, 원격 서버에 문장을 생성할 때 AltLinker가 자체적으로 부여한 식별자이다. 이 식별자는 관리 목적으로 생성된 것이므로, 그 값 자체에 의미를 둘 필요는 없다.

V\$DBLINK_REMOTE_TRANSACTION_INFO

데이터베이스 링크를 통해 원격 노드에서 수행중인 모든 원격 트랜잭션의 정보를 보여준다.

| Column name | Type | Description |
|-----------------------|-------------|----------------------------------|
| TRANSACTION_ID | INTEGER | 데이터베이스 링크를 사용한 트랜잭션 식별자 |
| REMOTE_TRANSACTION_ID | INTEGER | 원격 서버에 발생한 트랜잭션 식별자 |
| TARGET_INFO | VARCHAR(40) | 데이터베이스 링크 객체가 접근할 원격 서버의 이름 |
| STATUS | INTEGER | 이 글로벌 트랜잭션의 현재 상태 |
| XID | VARCHAR(12) | 트랜잭션 브랜치 식별자 |
| GLOBAL_TRANSACTION_ID | INTEGER | 데이터베이스 링크를 사용하고 있는 글로벌 트랜잭션의 식별자 |

칼럼 정보**REMOTE_TRANSACTION_ID**

원격 서버에 발생한 트랜잭션 식별자이다. 이 식별자는 실제 원격 서버에서 생성된 트랜잭션 식별자가 아니라, 원격 서버에 트랜잭션을 생성할 때 AltLinker가 자체적으로 부여한 식별자이다. 이 식별자는 관리 목적으로 생성된 것이므로, 그 값 자체에 의미를 둘 필요는 없다.

STATUS

이 글로벌 트랜잭션의 현재 상태를 나타낸다.

- 0(NONE): 트랜잭션이 존재하지 않음
- 1(BEGIN): 트랜잭션이 시작됨
- 2(PREPARE_READY): 트랜잭션이 시작되었으나 현재 수행중인 원격 트랜잭션은 존재하지 않음
- 3(PREPARE_WAIT): Simple transaction commit level에서 AltLinker 프로세스로부터 prepare 에 대한 응답을 기다리는 상태
- 4(PREPARED): prepare 완료
- 5(COMMIT_WAIT): AltLinker 프로세스로부터 commit 에 대한 응답을 기다리는 상태
- 6(COMMITTED): 트랜잭션 commit 완료
- 7(ROLLBACK_WAIT): AltLinker 프로세스로부터 rollback 에 대한 응답을 기다리는 상태
- 8(ROLLBACKED): 트랜잭션 rollback 완료

V\$DBMS_STATS

데이터베이스 전체의 통계 정보를 보여준다.

| Column name | Type | Description |
|-------------------|----------|-------------------------------------|
| DATE | CHAR(48) | 마지막으로 통계 정보를 수집한 일시 |
| SAMPLE_SIZE | DOUBLE | 샘플의 크기 |
| NUM_ROW_CHANGE | BIGINT | 마지막 통계 정보 수집 후 행 개수의 변화량 |
| TYPE | CHAR(1) | 통계 대상 유형 S: 시스템 T: 테이블 I: 인덱스 C: 칼럼 |
| SREAD_TIME | DOUBLE | 하나의 페이지를 읽는 데 소요된 시간 |
| MREAD_TIME | DOUBLE | 여러 페이지를 한 번에 읽는 데 소요된 시간 |
| MREAD_PAGE_COUNT | BIGINT | 여러 페이지를 한 번에 읽을 때 읽어 온 페이지 개수 |
| HASH_TIME | DOUBLE | 평균 해쉬 수행 시간 |
| COMPARE_TIME | DOUBLE | 평균 비교 수행 시간 |
| STORE_TIME | DOUBLE | 평균 메모리 임시 테이블 저장 수행 시간 |
| TARGET_ID | BIGINT | 통계 대상 테이블의 OID 또는 인덱스의 ID |
| COLUMN_ID | INTEGER | 통계 대상 칼럼의 ID |
| NUM_ROW | BIGINT | 행의 개수 |
| NUM_PAGE | BIGINT | 페이지의 개수 |
| NUM_DIST | BIGINT | 유일한 행의 개수 |
| NUM_NULL | BIGINT | NULL 개수 |
| AVG_LEN | BIGINT | 행 또는 칼럼 데이터의 평균 길이 |
| ONE_ROW_READ_TIME | DOUBLE | 행 하나를 읽는 평균 시간 |
| AVG_SLOT_COUNT | BIGINT | Leaf 노드 당 슬롯의 평균 개수 |
| INDEX_HEIGHT | BIGINT | 인덱스의 루트에서 leaf 노드까지의 깊이 |
| CLUSTERING_FACTOR | BIGINT | 인덱스에 부합하게 데이터가 정렬되어 있는 정도 |
| MIN | CHAR(48) | 최소 값 |
| MAX | CHAR(48) | 최대 값 |
| META_SPACE | BIGINT | 데이터 관리를 위해 사용된 공간의 크기 |
| USED_SPACE | BIGINT | 데이터를 저장하기 위해 사용된 공간의 크기 |
| AGEABLE_SPACE | BIGINT | 나중에 aging 되어 재활용 가능한 공간의 크기 |
| FREE_SPACE | BIGINT | 사용 가능한 공간의 크기 |

칼럼 정보

DATE

서버가 마지막으로 통계 정보를 수집한 일시를 나타낸다.

SAMPLE_SIZE

통계 정보 수집을 위해 선택된 샘플의 크기를 나타낸다.

NUM_ROW_CHANGE

마지막 통계 정보 수집 이후 행 개수의 변경된 양을 나타낸다.

TYPE

통계 수집 대상의 유형을 나타낸다. 아래의 값들 중 하나가 표시된다.

S: 시스템

T: 테이블

I: 인덱스

C: 칼럼

SREAD_TIME

하나의 페이지를 읽는 데 소요된 평균 시간을 나타낸다.

MREAD_TIME

여러 페이지를 한 번에 읽는 데 소요된 평균 시간을 나타낸다.

MREAD_PAGE_COUNT

여러 페이지를 한 번에 읽을 때 읽어 오도록 설정된 페이지의 개수를 나타낸다.

HASH_TIME

해쉬 수행에 소요된 평균 시간을 나타낸다.

COMPARE_TIME

비교 수행에 소요된 평균 시간을 나타낸다.

STORE_TIME

메모리 임시 테이블에 저장하는 데 소요된 평균 시간을 나타낸다.

TARGET_ID

통계 수집의 대상이 된 테이블의 OID 또는 인덱스의 ID 나타낸다.

COLUMN_ID

통계 수집의 대상이 된 칼럼의 ID를 나타낸다.

NUM_ROW

통계 수집 대상(테이블 또는 인덱스)의 행의 개수를 나타낸다.

NUM_PAGE

통계 수집 대상(테이블 또는 인덱스)의 페이지 개수를 나타낸다.

NUM_DIST

인덱스 또는 칼럼에서 중복되지 않은 유일한 값의 개수를 나타낸다.

NUM_NULL

칼럼에서 NULL의 개수를 나타낸다.

AVG_LEN

행 또는 칼럼의 평균 길이를 나타낸다.

ONE_ROW_READ_TIME

행 하나를 읽는 데 소요된 평균 시간을 나타낸다.

AVG_SLOT_COUNT

Leaf 노드 당 슬롯의 평균 개수를 나타낸다.

INDEX_HEIGHT

인덱스의 루트에서 leaf 노드까지의 깊이를 나타낸다.

CLUSTERING_FACTOR

인덱스에 부합하게 데이터가 정렬되어 있는 정도를 나타낸다.

MIN

인덱스 또는 칼럼의 최소 값을 나타낸다.

MAX

인덱스 또는 칼럼의 최대 값을 나타낸다.

META_SPACE

데이터 관리를 위해 사용된 공간의 크기를 나타낸다.

USED_SPACE

데이터를 저장하기 위해 사용된 공간의 크기를 나타낸다.

AGEABLE_SPACE

나중에 aging 되어 재활용할 수 있는 공간의 크기를 나타낸다.

FREE_SPACE

테이블 또는 인덱스에 할당된 영역 중에서 사용 가능한 공간의 크기를 나타낸다.

V\$DB_FREEPAGELISTS

데이터베이스에서 사용가능한 페이지 리스트 즉, free 페이지들의 정보를 보여준다.

| Column name | Type | Description |
|--------------------|---------|----------------------------|
| SPACE_ID | INTEGER | 사용가능한 페이지들이 속한 테이블스페이스 식별자 |
| RESOURCE_GROUP_ID | INTEGER | 자원 그룹 식별자 |
| FIRST_FREE_PAGE_ID | INTEGER | 리스트 내에서 첫번째 사용가능한 페이지 식별자 |
| FREE_PAGE_COUNT | BIGINT | 리스트 내의 사용가능한 페이지 개수 |

칼럼 정보

RESOURCE_GROUP_ID

다중화된 리스트들을 식별하기 위한 고유 번호이다.

FIRST_FREE_PAGE_ID

해당 리스트의 사용가능한 첫 번째 페이지 식별자이다.

FREE_PAGE_COUNT

해당 리스트 내에서 사용가능한 페이지 개수이다.

V\$DB_PROTOCOL

서버로 유입되는 모든 패킷들의 Altibase 통신 프로토콜 정보를 보여준다.

| Column name | Type | Description |
|-------------|-------------|---------------------|
| OP_NAME | VARCHAR(50) | 프로토콜 이름 |
| OP_ID | INTEGER | 프로토콜의 고유 식별자 |
| COUNT | BIGINT | 이 프로토콜로 유입된 패킷의 누적치 |

V\$DIRECT_PATH_INSERT

Direct-path 업로드 관련 통계 정보를 보여준다.

| Column name | Type | Description |
|-----------------------------|--------|--|
| COMMIT_TX_COUNT | BIGINT | Direct-path 옵션을 사용하여 커밋에 성공한 트랜잭션의 총 개수 |
| ABORT_TX_COUNT | BIGINT | Direct-path 옵션을 사용하여 데이터 업로드 중에 철회한 트랜잭션의 총 개수 |
| INSERT_ROW_COUNT | BIGINT | iLoader에서 direct-path 옵션을 사용하여 삽입한 행의 총 개수 |
| ALLOC_BUFFER_PAGE_TRY_COUNT | BIGINT | 페이지 할당 요청 총 횟수 |

| Column name | Type | Description |
|------------------------------|--------|---------------------|
| ALLOC_BUFFER_PAGE_FAIL_COUNT | BIGINT | 페이지 할당 요청이 실패한 총 횟수 |

칼럼 정보

COMMIT_TX_COUNT

이 값은 iLoader에서 direct-path 옵션을 사용하여 커밋한 트랜잭션의 총 개수로, 누적된다.

ABORT_TX_COUNT

이 값은 direct-path 옵션을 사용하여 데이터 업로드 중에 오류로 인해서 롤백된 트랜잭션의 총 개수로, 누적된다.

INSERT_ROW_COUNT

iLoader에서 direct-path 옵션을 사용하여 삽입한 행의 총 개수로, 누적된다.

ALLOC_BUFFER_PAGE_TRY_COUNT

이 값은 direct-path 옵션을 사용한 데이터 업로드를 위해 페이지 할당이 요청된 총 횟수로, 누적된다.

ALLOC_BUFFER_PAGE_FAIL_COUNT

이 값은 direct-path 옵션을 사용한 데이터 업로드를 위해 페이지 할당이 요청되었으나 메모리 부족 등의 이유로 인해 실패한 총 횟수로, 누적된다.

V\$DISKTBL_INFO

디스크 테이블의 정보를 보여준다.

| Column name | Type | Description |
|---------------------|----------|--------------------------------|
| TABLESPACE_ID | SMALLINT | 테이블스페이스 식별자 |
| TABLE_OID | BIGINT | 테이블 객체 식별자 |
| DISK_TOTAL_PAGE_CNT | BIGINT | 테이블이 가지고 있는 전체 페이지 개수 |
| DISK_PAGE_CNT | BIGINT | 테이블에서 데이터를 갖고 있는 페이지 개수 |
| SEG_PID | INTEGER | 테이블 세그먼트의 페이지 식별자 |
| META_PAGE | INTEGER | Deprecated |
| FST_EXTRID | BIGINT | 테이블의 첫번째 익스텐트의 RID |
| LST_EXTRID | BIGINT | 테이블의 마지막 익스텐트의 RID |
| PCTFREE | SMALLINT | SYS_TABLES_의 설명 참조 |
| PCTUSED | SMALLINT | SYS_TABLES_의 설명 참조 |
| INITRANS | SMALLINT | 한 페이지 내에서 동시 처리 가능한 초기 트랜잭션 개수 |
| MAXTRANS | SMALLINT | 한 페이지 내에서 동시 처리 가능한 최대 트랜잭션 개수 |

| Column name | Type | Description |
|--------------------|---------|---------------------|
| INITEXTENTS | INTEGER | 테이블 생성시 초기 익스텐트 개수 |
| NEXTEXTENTS | INTEGER | 테이블 확장시 할당할 익스텐트 개수 |
| MINEXTENTS | INTEGER | 테이블의 최소 익스텐트 개수 |
| MAXEXTENTS | INTEGER | 테이블의 최대 익스텐트 개수 |
| COMPRESSED_LOGGING | INTEGER | 테이블을 위한 로그 압축 여부 |
| IS_CONSISTENT | INTEGER | 테이블의 일관성 여부 |

테이블 이름을 포함하여 보려면 다음과 같이 메타 테이블과 조인하여 질의를 하여야 한다.

```
SELECT A.TABLE_NAME,
       B.DISK_PAGE_CNT,
       B.PCTFREE,
       B.PCTUSED
FROM SYSTEM_.SYS_TABLES_ A, V$DISKTBL_INFO B
WHERE A.TABLE_OID = B.TABLE_OID;
```

칼럼 정보

PCTFREE

SYS_TABLES_ 설명의 해당하는 칼럼 정보를 참조한다.

PCTUSED

SYS_TABLES_ 설명의 해당하는 칼럼 정보를 참조한다.

INITRANS

하나의 테이블 페이지 내에서 동시에 처리할 수 있는 트랜잭션의 초기 개수를 나타낸다.

MAXTRANS

하나의 테이블 페이지 내에서 동시에 처리할 수 있는 트랜잭션의 최대 개수를 나타낸다.

INITEXTENTS

테이블 세그먼트 생성시 초기 익스텐트 개수를 나타낸다.

NEXTEXTENTS

테이블 세그먼트 확장시 할당할 익스텐트 개수를 나타낸다.

MINEXTENTS

테이블 세그먼트의 최소 익스텐트 개수를 나타낸다.

MAXEXTENTS

테이블 세그먼트의 최대 익스텐트 개수를 나타낸다.

V\$DISK_BTREE_HEADER

디스크 BTREE 인덱스의 헤더 정보를 보여준다.

| Column name | Type | Description |
|----------------------------|-----------|--|
| INDEX_NAME | CHAR(128) | 인덱스 이름 |
| INDEX_ID | INTEGER | 인덱스 식별자 |
| INDEX_TBS_ID | INTEGER | 인덱스가 저장되어 있는 테이블스페이스 식별자 |
| TABLE_TBS_ID | INTEGER | 테이블이 저장되어 있는 테이블스페이스 식별자 |
| IS_UNIQUE | CHAR(1) | 유일 키 인덱스인지 여부 |
| COLLENINFO_LIST | CHAR(64) | 인덱스 값들의 사이즈 목록 |
| IS_CONSISTENT | CHAR(1) | 인덱스의 일관성 여부 |
| IS_CREATED_WITH_LOGGING | CHAR(1) | 인덱스 생성시 LOGGING 옵션 지정 여부 |
| IS_CREATED_WITH_FORCE | CHAR(1) | 인덱스 생성시 NOLOGGING FORCE또는 NOLOGGING NOFORCE 옵션 지정 여부 |
| COMPLETION_LSN_LFG_ID | INTEGER | 사용하지 않음(0) |
| COMPLETION_LSN_FILE_NO | INTEGER | 인덱스 생성 시점의 로그 파일 번호 |
| COMPLETION_LSN_FILE_OFFSET | INTEGER | 인덱스 생성 시점의 로그 파일 오프셋 |
| INIT_TRANS | SMALLINT | 하나의 인덱스 노드에서 동시 처리 가능한 초기 트랜잭션 개수 |
| MAX_TRANS | SMALLINT | 하나의 인덱스 노드에서 동시 처리 가능한 최대 트랜잭션 개수 |
| FREE_NODE_HEAD | INTEGER | 프리 노드의 첫번째 페이지의 ID |
| FREE_NODE_CNT | BIGINT | 프리 노드 리스트 내의 페이지 개수 |
| INITEXTENTS | INTEGER | 인덱스 생성시 초기 익스텐트 개수 |
| NEXTTEXTENTS | INTEGER | 인덱스 사이즈 확장시 할당할 익스텐트 개수 |
| MINEXTENTS | INTEGER | 인덱스 세그먼트의 최소 익스텐트 개수 |
| MAXEXTENTS | INTEGER | 인덱스 세그먼트의 최대 익스텐트 개수 |

칼럼 정보

INDEX_NAME

인덱스의 이름이다.

INDEX_ID

해당 인덱스가 갖는 시스템 내에서 고유한 식별자이다.

INDEX_TBS_ID

인덱스가 저장되어 있는 테이블스페이스 식별자이다.

TABLE_TBS_ID

해당 인덱스가 기반하고 있는 테이블이 저장되어 있는 테이블스페이스 식별자이다.

IS_UNIQUE

유일키 인덱스인지 여부를 나타낸다. 유일키 인덱스는 'Y', 중복키 인덱스의 경우는 'N'이다.

COLLENINFO_LIST

인덱스를 구성하는 값들의 사이즈 리스트이다. 이 리스트는 쉼표로 구분된 스트링으로 표현된다. 가변 길이 칼럼에 해당하는 사이즈는 '?'로 표시된다. 인덱스 키의 크기는 이 리스트에 기반하여 추정 가능하다.

```
SQL> CREATE TABLE D3(I1 SMALLINT, I2 INTEGER, I3 VARCHAR(10), I4 DATE)
TABLESPACE SYS_TBS_DISK_DATA;
Create success.
SQL> CREATE INDEX D3X ON D3(I4,I3,I2,I1);
Create success.
SQL> SELECT COLLENINFO_LIST FROM V$DISK_BTREE_HEADER WHERE INDEX_NAME='D3X';
COLLENINFO_LIST
-----
8,?,4,2
1 row selected.
```

IS_CONSISTENT

인덱스의 일관성 여부를 나타낸다. 일반적인 경우에는 'Y'를 가지며, 인덱스가 비정상적으로 구성되어 있는 경우는 'N'을 갖는다. NOLOGGING이나 NOFORCE를 이용하여 인덱스를 생성한 경우에는 'N'을 가질 수 있다.

IS_CREATED_WITH_LOGGING

인덱스 생성 시 LOGGING 옵션이 지정되었는지 여부를 나타낸다.

IS_CREATED_WITH_FORCE

인덱스 생성 시 강제적 디스크 저장 옵션 (NOLOGGING FORCE 또는 NOLOGGING NOFORCE 옵션) 지정 여부를 나타낸다.

COMPLETION_LSN_FILE_NO

인덱스가 생성된 시점에서의 로그 파일 번호를 나타낸다.

COMPLETION_LSN_FILE_OFFSET

인덱스가 생성된 시점에서의 로그 파일 오프셋 (Offset)을 나타낸다.

INIT_TRANS

삽입, 갱신 또는 삭제를 하기 위해 하나의 인덱스 노드(페이지)에 동시에 접근할 수 있는 트랜잭션의 초기 개수를 나타낸다.

MAX_TRANS

삽입, 갱신 또는 삭제를 하기 위해 하나의 인덱스 노드(페이지)에 동시에 접근할 수 있는 트랜잭션의 최대 개수를 나타낸다.

FREE_NODE_HEAD

FREE_NODE_HEAD는 인덱스 내 FREE NODE들의 첫번째 페이지를 나타낸다. FREE NODE는 노드 내의 모든 키에 삭제 마크가 설정되어 있는 상태의 노드이다.

FREE_NODE_CNT

인덱스 내 FREE NODE의 전체 개수이다.

INITEXTENTS

인덱스 세그먼트 생성시 초기 익스텐트 개수이다.

NEXTTEXTENTS

인덱스 세그먼트 확장시 할당할 익스텐트 개수이다.

MINEXTENTS

인덱스 세그먼트의 최소 익스텐트 개수이다.

MAXEXTENTS

인덱스 세그먼트의 최대 익스텐트 개수이다.

V\$DISK_TEMP_INFO

전체 디스크 임시 테이블이 사용한 메모리의 사용 정보를 보여준다.

| Column name | Type | Description |
|-------------|----------|-------------|
| NAME | CHAR(32) | 메모리의 최솟값 이름 |
| VALUE | CHAR(32) | 메모리 최솟값 |
| UNIT | CHAR(32) | 단위 |

칼럼 정보**VALUE**

서버가 시작한 이후부터 현재까지 동작한 디스크 임시 테이블을 메모리에서 정렬하기 위하여 필요한 메모리의 최솟값을 나타낸다.

V\$DISK_TEMP_STAT

현재 사용중인 각각의 디스크 임시 테이블이 메모리를 사용하는 정보를 보여준다. 이 정보는 [TEMP_STATS_WATCH_TIME](#) 프로퍼티에 설정된 값 이상일 때 통계 정보가 수집된다.

| Column name | Type | Description |
|--------------------|---------|------------------------------|
| TBS_ID | INTEGER | 테이블스페이스 식별자 |
| TRANSACTION_ID | BIGINT | 트랜잭션 식별자 |
| CONSUME_TIME | INTEGER | 디스크 임시 테이블의 수행 시간 |
| READ_COUNT | BIGINT | 데이터를 읽어오는 IO가 발생한 횟수 |
| WRITE_COUNT | BIGINT | 데이터를 저장하는 IO가 발생한 횟수 |
| WRITE_PAGE_COUNT | BIGINT | 페이지가 디스크로 저장된 총 개수 |
| ALLOC_WAIT_COUNT | BIGINT | 메모리 공간 할당을 위해 대기한 총 횟수 |
| WRITE_WAIT_COUNT | BIGINT | 현재 사용되지 않음 |
| QUEUE_WAIT_COUNT | BIGINT | 현재 사용되지 않음 |
| WORK_AREA_SIZE | BIGINT | 디스크 임시 테이블이 사용하는 메모리 크기 |
| MAX_WORK_AREA_SIZE | BIGINT | 디스크 임시 테이블이 사용할수있는 최대 메모리 크기 |
| DISK_USAGE | BIGINT | 디스크에 저장된 데이터 공간의 크기 |
| RUNTIME_MAP_SIZE | BIGINT | map 정보를 위해 사용된 메모리크기 |

칼럼 정보

TBS_ID

디스크 임시 테이블을 사용하는 테이블스페이스 식별자이다.

TRANSACTION_ID

디스크 임시 테이블을 사용하는 트랜잭션의 식별자이다.

CONSUME_TIME

디스크 임시 테이블이 [TEMP_STATS_WATCH_TIME](#) 프로퍼티에 설정된 시간을 초과하여 수행된 경우, 수행되는 시간을 보여준다.

READ_COUNT

디스크 상에 있는 데이터를 읽어오기 위해 READ IO가 발생한 횟수

WRITE_COUNT

디스크 상에 데이터를 저장하기 위해 WRITE IO가 발생한 횟수

WRITE_PAGE_COUNT

디스크 임시 테이블이 디스크로 저장되는 페이지의 총 개수

ALLOC_WAIT_COUNT

해쉬(hash) 정렬을 하기 위해 메모리의 공간 할당을 대기하는 횟수

WRITE_WAIT_COUNT

디스크에 데이터를 저장하기 위해 대기하는 횟수

QUEUE_WAIT_COUNT

디스크 상에 데이터를 저장하기 위해 큐에 입력되기까지 대기하는 횟수

WORK_AREA_SIZE

해쉬(hash) 정렬을 하기 위해 메모리에서 사용된 공간

DISK_USAGE

디스크 임시 테이블이 디스크로 저장된 공간의 크기

V\$DISK_UNDO_USAGE

디스크상에서 현재 사용중인 언두 테이블스페이스의 양에 대한 정보를 보여준다.

| Column name | Type | Description |
|---------------------|--------|-------------------------------|
| TX_EXT_CNT | BIGINT | 트랜잭션 세그먼트용 익스텐트의 개수 |
| USED_EXT_CNT | BIGINT | 언두 세그먼트에서 현재 사용중인 익스텐트의 개수 |
| UNSTEALABLE_EXT_CNT | BIGINT | 다른 언두 세그먼트가 가져갈 수 없는 익스텐트의 개수 |
| REUSABLE_EXT_CNT | BIGINT | 재사용 가능한 익스텐트의 개수 |
| TOTAL_EXT_CNT | BIGINT | 언두 테이블스페이스의 총 익스텐트 개수 |

칼럼 정보**TX_EXT_CNT**

모든 트랜잭션 세그먼트의 익스텐트 개수이다. 이 익스텐트들은 언두 세그먼트용으로는 사용되지 않는다.

USED_EXT_CNT

언두 세그먼트에서 현재 사용중인 익스텐트의 개수이다. 현재 사용중인 익스텐트들은 후속 작업에서 재 사용되지 않는다.

REUSABLE_EXT_CNT

더 이상 필요하지 않은 언두 레코드만 가지고 있어 재사용이 가능한 익스텐트의 개수이다.

V\$DR_CONNECTION_INFO

DR 환경에 현재 참여하고 있는 서버들의 정보를 보여준다.

| Column name | Type | Description |
|-------------|-------------|------------------|
| SERVER_NAME | VARCHAR(40) | 서버 이름 |
| SERVER_IP | VARCHAR(64) | 서버의 IP 주소 |
| SERVER_PORT | INTEGER | 서버 청취자의 청취 포트 번호 |

칼럼 정보

SERVER_NAME

DR 환경을 구성하는 서버에 주어진 이름이다.

SERVER_IP

DR 환경을 구성하는 서버의 IP 주소이다.

SERVER_PORT

서버의 청취자가 청취하는 포트 번호이다.

V\$DR_GAP

DR 환경에 현재 참여하고 있는 서버들 간의 동기화 격차를 보여준다.

| Column name | Type | Description |
|--------------|-------------|--|
| SERVER_NAME | VARCHAR(40) | 서버 이름 |
| CURRENT_SN | BIGINT | Active 서버: 현재 전송중인 로그 레코드의 식별 번호. Standby 서버: 현재 적용중인 로그 레코드의 식별 번호. |
| SYNCED_SN | BIGINT | Active 서버 : 항상 0. 의미 없는 값. Standby 서버에서 마지막으로 전송받은 로그 레코드의 식별 번호 |
| SN_GAP | BIGINT | Active 서버: 항상 0. 의미 없는 값. Standby 서버: 대응하는 Active 서버의 CURRENT_SN과 해당 Standby 서버의 SYNCED_SN의 차이. |
| APPLY_SN_GAP | BIGINT | Active 서버 : 항상 0. 의미 없는 값. Standby 서버 : SYNCED_SN과 CURRENT_SN 의 차이. |

칼럼 정보

SERVER_NAME

DR 환경을 구성하는 서버의 이름이다.

CURRENT_SN

Active 서버에서는 현재 전송중인 로그 레코드의 식별 번호가 표시된다.

Standby 서버에서는 현재 데이터베이스에 적용중인 로그 레코드의 식별 번호가 표시된다.

SYNCED_SN

Active 서버에서는 항상 0이 표시된다.

Standby 서버에서는 마지막으로 전송받은 로그 레코드의 식별 번호가 표시된다.

SN_GAP

Active 서버에서는 항상 0이 표시된다.

Standby 서버에서는 현재 Active 서버의 CURRENT_SN과 해당 Standby 서버의 SYNCED_SN의 차이가 표시된다.

APPLY_SN_GAP

Active 서버에서는 항상 0이 표시된다.

Standby 서버에서는 SYNCED_SN과 CURRENT_SN의 차이가 표시된다.

V\$DR_SERVERS

DR 환경을 구성하는 서버들의 정보를 보여준다.

| Column name | Type | Description |
|-------------|-------------|------------------|
| SERVER_NAME | VARCHAR(40) | 서버 이름 |
| SERVER_IP | VARCHAR(64) | 서버의 IP 주소 |
| SERVER_PORT | INTEGER | 서버 청취자의 청취 포트 번호 |

칼럼 정보**SERVER_NAME**

DR 환경을 구성하는 서버에 주어진 이름이다.

SERVER_IP

DR 환경을 구성하는 서버의 IP 주소이다.

SERVER_PORT

서버의 청취자가 청취하는 포트 번호이다.

V\$DR_STATUS

DR 환경에 참여하고 있는 서버들의 현재 상태를 보여준다.

| Column name | Type | Description |
|--------------|-------------|-------------|
| SERVER_NAME | VARCHAR(40) | 서버 이름 |
| CURRENT_MODE | VARCHAR(7) | 동기화 모드 |

| Column name | Type | Description |
|----------------|------------|------------------|
| SERVER_ROLE | VARCHAR(7) | 서버의 역할 |
| SERVER_MODE | VARCHAR(7) | 사용자가 설정한 동기화 모드 |
| SERVER_STATUS | VARCHAR(8) | 서버 상태 |
| FAILOVER_SN | BIGINT | Fail-Over 시점의 SN |
| FAILOVER_COUNT | BIGINT | Fail-Over 횟수 |

칼럼 정보

SERVER_NAME

DR 환경을 구성하는 서버의 이름이다.

CURRENT_MODE

현재 동작중인 동기화 모드로, async 또는 sync로 표시된다.

모든 Standby 서버들은 Active 서버의 동기화 모드를 그대로 따른다.

SERVER_ROLE

서버의 역할을 나타낸다. active 또는 standby로 표시된다.

SERVER_MODE

사용자가 설정한 동기화 모드로, async 또는 sync로 표시된다.

SERVER_STATUS

현재 서버의 동작 상태를 나타낸다. run, stop, 또는 Failure Server Repair로 표시된다.

FAILOVER_SN

Fail-Over가 발생한 시점의 SN이다.

FAILOVER_COUNT

Fail-Over가 발생한 누적 횟수이다.

V\$EVENT_NAME

Altibase 서버에서 대기하고 있는 다양한 대기 이벤트들의 정보를 보여준다.

| Column name | Type | Description |
|---------------|--------------|-------------|
| EVENT_ID | INTEGER | 대기 이벤트 식별자 |
| NAME | VARCHAR(128) | 대기 이벤트의 이름 |
| WAIT_CLASS_ID | INTEGER | 대기 클래스 식별자 |
| WAIT_CLASS | VARCHAR(128) | 대기 클래스의 이름 |

칼럼 정보

EVENT_ID

대기하고 있는 이벤트의 식별자이다.

NAME

대기하고 있는 이벤트의 이름이다. 다음 표는 식별자, 이름 및 그에 대한 설명을 보여준다.

| EVENT_ID | 이름 | 설명 |
|----------|---|---|
| 0 | latch: buffer busy waits | 다른 세션이 변경하고 있는 블록에 접근하기 위한 대기 |
| 1 | latch: drdb B-tree index SMO | B-tree 인덱스의 SMO (Structure Modification Operation)를 수행하는 세션에 의해 발생하는 대기 |
| 2 | latch: drdb B-tree index SMO by other session | 다른 세션에 의해 수행되는 B-tree 인덱스의 SMO 연산이 완료될 때까지 대기 |
| 3 | latch: drdb R-tree index SMO | R-tree 인덱스의 SMO 연산을 하고 있는 세션에 의해 발생하는 대기 |
| 4 | db file multi page read | 다중 페이지 읽기 요청이 완료되기를 대기하는 세션에 의해 발생 |
| 5 | db file single page read | 단일 페이지 읽기 요청이 완료되기를 대기하는 세션에 의해 발생 |
| 6 | db file single page write | LRU flush를 수행하기 전에 free BCB가 확보될 때까지 대기 |
| 7 | enq: TX - row lock contention, data row | 갱신을 위해 로우(row)에 잠금을 하기 위한 대기 |
| 8 | enq: TX - allocate TXSEG entry | 트랜잭션 세그먼트 엔트리를 할당하기 위한 대기 |
| 9 | latch free: drdb file i/o | 디스크 파일에 read/write I/O를 수행하기 위해서 파일 래치를 획득하기를 대기 |
| 10 | latch free: drdb tbs list | 다른 스레드에 의해 사용되고 있는 테이블스페이스의 해쉬 래치를 얻기 위해 대기 |
| 11 | latch free: drdb tbs creation | 테이블스페이스 생성시 파일을 생성 하려는 세션에 의해 발생하는 대기 |
| 12 | latch free: disk page list entry | 다른 스레드에 의해 사용되고 있는 디스크 페이지 리스트 엔트리의 래치를 획득하기를 대기 |
| 13 | latch free: drdb transaction segment freelist | 트랜잭션 세그먼트 프리 리스트에 대한 대기 |

| EVENT_ID | 이름 | 설명 |
|----------|---|---|
| 14 | latch free: drdb LRU list | 버퍼 풀의 LRU 리스트들에 대한 대기 |
| 15 | latch free: drdb prepare list | 버퍼 풀의 prepare 리스트들에 대한 대기 |
| 16 | latch free: drdb prepare list wait | 버퍼 풀의 prepare 리스트에 BCB가 추가될 때까지 대기 |
| 17 | latch free: drdb flush list | 버퍼 풀의 flush 리스트들에 대한 대기 |
| 18 | latch free: drdb checkpoint list | 버퍼 풀의 checkpoint 리스트들에 대한 대기 |
| 19 | latch free: drdb buffer flusher min recovery LSN | 버퍼 풀 flusher의 Recovery LSN 동시성 제어를 위한 래치에 대기 |
| 20 | latch free: drdb buffer flush manager req job | 버퍼 풀의 플러시 작업의 동시성 제어를 위한 래치에 대기 |
| 21 | latch free: drdb buffer bcb mutex | 버퍼 풀의 BCB 동시성 제어를 위한 래치에 대기 |
| 22 | latch free: drdb buffer bcb read io mutex | 버퍼 풀의 BCB로 페이지를 적재하기 위한 래치에 대기 |
| 23 | latch free: drdb buffer buffer manager expand mutex | 버퍼 풀의 확장에 대한 대기 |
| 24 | latch free: drdb buffer hash mutex | 버퍼 풀의 해쉬에 대한 대기 |
| 25 | latch free: plan cache LRU List mutex | 리스트에 plan을 추가, 이동 또는 제거시, Plan cache내 LRU 리스트의 래치를 획득하기 위한 대기 |
| 26 | latch free: statement list mutex | 리스트에 statement를 추가, 이동 또는 삭제시, Statement 리스트의 래치를 획득하기 위한 대기 |
| 27 | latch free: others | 다른 쓰레드에 의해 사용되고 있는 위에서 언급되지 않은 모든 래치에 대해서 획득하기를 대기 |
| 28 | replication before commit | EAGER 모드에서, COMMIT 이전의 구문들에 대응하는 모든 XLog들을 원격 서버에서 재현할 때까지 로컬 서버가 트랜잭션 커밋을 대기 (<i>Replication Manual</i> 의 EAGER 모드 설명 참조) |
| 29 | replication after commit | EAGER 모드에서, COMMIT 구문에 대응하는 XLog를 원격 서버에 송신할 때까지 로컬 서버가 트랜잭션 커밋을 대기 (<i>Replication Manual</i> 의 EAGER 모드 설명 참조) |
| 30 | no wait event | 대기 이벤트가 존재하지 않음 |

WAIT_CLASS_ID

대기 이벤트의 클래스 식별자이다. 클래스 식별자에 대한 자세한 정보는 V\$WAIT_CLASS_NAME를 참조하기 바란다.

WAIT_CLASS

대기 이벤트는 상위 개념의 대기 클래스로 그룹화된다. 대기 클래스에 대한 자세한 정보는 V\$WAIT_CLASS_NAME를 참조하기 바란다.

V\$EXTPROC_AGENT

외부 프로시저 실행을 위해 생성된 에이전트 프로세스(agent process)의 정보를 보여준다.

| Column name | Type | Description |
|-------------|-------------|--------------------------------|
| SID | INTEGER | 에이전트 프로세스를 생성한 세션의 식별자 |
| PID | INTEGER | 에이전트 프로세스의 pid |
| SOCK_FILE | VARCHAR(64) | 프로세스 간 통신을 위한 소켓의 경로 |
| CREATED | INTEGER | 에이전트 프로세스가 생성된 일시 |
| LAST_SEND | INTEGER | 에이전트 프로세스가 마지막으로 결과를 반환한 일시 |
| LAST_RECV | INTEGER | 에이전트 프로세스가 마지막으로 호출 메시지를 받은 일시 |
| STATE | VARCHAR(11) | 에이전트 프로세스의 상태 |

칼럼 정보**SID**

에이전트 프로세스를 생성한 세션의 식별자를 나타낸다. 에이전트 프로세스는 세션에 종속적이다.

PID

에이전트 프로세스의 프로세스 ID를 나타낸다.

SOCK_FILE

프로세스 간의 통신에 사용되는 소켓의 경로를 나타낸다.

CREATED

에이전트 프로세스가 생성된 일시를 나타낸다.

LAST_SEND

에이전트 프로세스가 외부 프로시저를 호출한 서버 세션으로 가장 최근에 결과를 반환한 일시를 나타낸다.

LAST_RECV

에이전트 프로세스가 서버 세션으로부터 가장 최근에 호출 메시지를 받은 일시를 나타낸다.

LAST_RECV

에이전트 프로세스의 상태를 나타낸다. 아래의 값 중 하나로 표시된다.

- INITIALIZED : 최초로 생성되어 호출을 기다림
- RUNNING : 외부 프로시저(External Procedure) 실행 중
- STOPPED : 외부 프로시저(External Procedure) 실행 완료
- FAILED : 비정상 종료 됨. 에이전트 프로세스가 이미 종료되었을 수도 있음

V\$FILESTAT

Altibase 구동 이후 각 디스크에 있는 데이터 파일별 I/O 통계 정보를 보여준다. 통계 정보를 통해 핫스팟(hotspot) 데이터 파일을 알 수 있다.

| Column name | Type | Description |
|----------------|---------|-------------------------------------|
| SPACEID | INTEGER | 테이블스페이스 식별자 |
| FILEID | INTEGER | 데이터 파일 식별자 |
| PHYRDS | BIGINT | 물리적 Read I/O 발생 횟수 |
| PHYWRTS | BIGINT | 물리적 Write I/O 발생 횟수 |
| PHYBLKRD | BIGINT | 물리적인 읽기로 판독한 페이지 개수 |
| PHYBLKWRT | BIGINT | 물리적인 쓰기로 기록한 페이지 개수 |
| SINGLEBLKRDS | BIGINT | 단일 페이지에 대한 읽기 작업 횟수 |
| READTIM | DOUBLE | Read I/O 작업 시간 (milliseconds) |
| WRITETIM | DOUBLE | Write I/O 작업 시간 (milliseconds) |
| SINGLEBLKRDTIM | DOUBLE | 단일 페이지에 대한 읽기에 걸린 시간 (milliseconds) |
| AVGIOTIM | DOUBLE | 평균 I/O 작업 시간 (milliseconds) |
| LSTIOTIM | DOUBLE | 마지막 I/O 작업 시간 (milliseconds) |
| MINIOTIM | DOUBLE | 최소 I/O 작업 시간 (milliseconds) |
| MAXIORTM | DOUBLE | 최대 Read I/O 작업 시간 (milliseconds) |
| MAXIOWTM | DOUBLE | 최대 Write I/O 작업 시간 (milliseconds) |

칼럼 정보**SPACEID**

테이블스페이스의 식별자이다.

FILEID

데이터 파일의 식별자이다.

PHYRDS

물리적 Read I/O가 발생한 횟수다.

PHYWRTS

물리적 Write I/O가 발생한 횟수다.

PHYBLKRD

물리적인 Read로 판독한 페이지 개수이다.

PHYBLKWRT

물리적인 Write로 기록한 페이지 개수이다.

SINGLEBLKRDS

단일 페이지에 대한 Read 작업 횟수이다.

READTIM

Read I/O 작업에 걸린 시간이다. (단위: milliseconds)

WRITETIM

Write I/O 작업에 걸린 시간이다. (단위: milliseconds)

SINGLEBLKRDTIM

단일 페이지에 대하여 Read 작업에 걸린 시간이다. (단위: milliseconds)

AVGIOTIM

I/O 작업에 걸린 평균 시간이다. (단위: milliseconds)

LSTIOTIM

마지막 I/O 작업에 걸린 시간이다. (단위: milliseconds)

MINIOTIM

I/O 작업에 걸린 최소 시간이다. (단위: milliseconds)

MAXIORTM

Read I/O 작업에 걸린 최대 시간이다. (단위: milliseconds)

MAXIOWTM

Write I/O 작업에 걸린 최대 시간이다. (단위: milliseconds)

V\$FLUSER

플러시 작업에 대한 정보를 보여준다.

| Column name | Type | Description |
|-------------|---------|-------------|
| ID | INTEGER | Flusher 식별자 |

| Column name | Type | Description |
|------------------------|---------|--|
| ALIVE | INTEGER | Flusher가 현재 활동 중인지 여부 |
| CURRENT_JOB | INTEGER | 현재 작업 1: 교체 플러시 중 2: 체크포인트 플러시 중 3: 객체 플러시 중 |
| DOING_IO | INTEGER | Flusher가 디스크 I/O 수행중인지 여부 |
| INIOB_COUNT | INTEGER | 플러시되는 내용을 그 안에 저장하기 위해 내부 버퍼에 직접 접근한 횟수 |
| REPLACE_FLUSH_JOBS | BIGINT | 완료된 교체 플러시 작업의 누적 횟수 |
| REPLACE_FLUSH_PAGES | BIGINT | 교체 플러시로 디스크에 쓰여진 페이지의 누적 개수 |
| REPLACE_SKIP_PAGES | BIGINT | Replacement 플러시 중에 플러시가 취소된 페이지의 누적 개수 |
| CHECKPOINT_FLUSH_JOBS | BIGINT | 완료된 체크포인트 플러시 작업의 누적 횟수 |
| CHECKPOINT_FLUSH_PAGES | BIGINT | 체크포인트 플러시로 디스크에 쓰여진 페이지의 누적 개수 |
| CHECKPOINT_SKIP_PAGES | BIGINT | 체크포인트 플러시 중에 플러시가 취소된 페이지의 누적 개수 |
| OBJECT_FLUSH_JOBS | BIGINT | 객체 플러시가 수행된 누적 횟수 |
| OBJECT_FLUSH_PAGES | BIGINT | 객체 플러시로 디스크에 쓰여진 페이지의 누적 개수 |
| OBJECT_SKIP_PAGES | BIGINT | 객체 플러시 중에 플러시가 취소된 페이지의 누적 개수 |
| LAST_SLEEP_SEC | INTEGER | 작업이 모두 완료된 후 Flusher가 잠들어 있던 시간의 길이 |
| TIMEOUT | BIGINT | 작업 유무를 확인하기 위해서 잠든 Flusher가 깨어난 횟수 |
| SIGNALED | BIGINT | Altibase로부터의 시그널에 의해 Flusher가 깨어난 횟수 |
| TOTAL_SLEEP_SEC | BIGINT | Flusher가 잠들어 있던 시간의 총 길이 |
| TOTAL_FLUSH_PAGES | BIGINT | 플러시된 페이지의 누적 개수 |
| TOTAL_LOG_SYNC_USEC | BIGINT | 버퍼에 있는 리두 로그를 디스크로 쓰는 데 걸린 시간의 누적 양 |
| TOTAL_DW_USEC | BIGINT | DoubleWrite 버퍼의 내용을 디스크로 쓰는데 걸린 시간의 누적 양 |
| TOTAL_WRITE_USEC | BIGINT | 데이터 페이지를 데이터 파일에 쓰는데 걸린 시간의 누적 양 |

| Column name | Type | Description |
|--------------------------|--------|--------------------------------------|
| TOTAL_SYNC_USEC | BIGINT | 데이터 페이지를 디스크로 강제 플러시하는데 걸린 시간의 누적 양 |
| TOTAL_FLUSH_TEMP_PAGES | BIGINT | 플러시된 임시 페이지의 누적 개수 |
| TOTAL_TEMP_WRITE_USEC | BIGINT | 임시 페이지를 임시 파일에 쓰는데 걸린 시간의 누적 양 |
| TOTAL_CALC_CHECKSUM_USEC | BIGINT | 체크섬(checksum) 계산에 걸린 시간의 누적 양 |
| DB_WRITE_PERF | DOUBLE | 데이터 페이지를 데이터 파일에 쓸 때 초당 기록한 평균 바이트 수 |
| TEMP_WRITE_PERF | DOUBLE | 임시 페이지를 임시 파일에 쓸 때 초당 기록한 평균 바이트 수 |

칼럼 정보

ID

Flusher 식별자이다. 식별자는 중복되지 않는다.

ALIVE

Flusher 가 현재 동작 중인지 여부를 나타낸다. 각 Flusher는 DCL구문으로 시작하거나 중지할 수 있다.

CURRENT_JOB

Flusher가 현재 수행중인 작업의 유형을 나타낸다.

- 1: 교체 플러시 수행 중임을 가리킨다. 교체 플러시의 목적은 오랜 시간 접근되지 않은 버퍼를 플러시하여 교체 가능하도록 하는 데 있다.
- 2: 체크포인트 플러시 수행 중임을 가리킨다. 체크포인트 플러시의 목적은 가장 오래 전에 갱신된 버퍼를 플러시하여 체크포인트 시간을 줄이는 데 있다.
- 3: 인덱스, 테이블, 세그먼트 등의 특정 객체를 플러시하고 있음을 가리킨다.

DOING_IO

Flusher가 현재 자신의 업무 수행을 위해서 디스크 I/O 작업 중인지 여부를 나타낸다.

INIOB_COUNT

Flusher는 페이지를 디스크에 기록하기 위해서, 그 내용을 내부 버퍼 (IOB)에 저장한다. 이 값은 그 내부 버퍼에 플러시할 내용을 저장하기 위해 접근한 횟수를 가리킨다.

REPLACE_FLUSH_JOBS

교체 플러시 작업을 수행한 횟수이다.

REPLACE_FLUSH_PAGES

교체 플러시 작업에 의해 디스크에 쓰여진 페이지의 누적 개수이다.

REPLACE_SKIP_PAGES

교체 플러시 중에 정책 또는 효율의 이유로 인해서 플러시 작업이 취소된 페이지의 누적 개수이다.

CHECKPOINT_FLUSH_JOBS

체크포인트 플러시 작업을 수행한 누적 횟수이다.

CHECKPOINT_FLUSH_PAGES

체크포인트 플러시 작업에 의해 디스크에 쓰여진 페이지의 누적 개수이다.

CHECKPOINT_SKIP_PAGES

체크포인트 플러시 중에 정책 또는 효율의 이유로 인해서 플러시가 취소된 페이지의 누적 개수이다.

OBJECT_FLUSH_JOBS

객체 플러시 작업을 수행한 누적 횟수이다.

OBJECT_FLUSH_PAGES

객체 플러시 작업에 의해 디스크에 쓰여진 페이지의 누적 개수이다.

OBJECT_SKIP_PAGES

객체 플러시 중에 정책 또는 효율의 이유로 인해서 플러시가 취소된 페이지의 누적 개수이다.

LAST_SLEEP_SEC

가장 최근에 모든 작업을 완료한 Flusher가 더 이상 작업이 없어서 잠들어 있던 시간의 길이이다.

TIMEOUT

작업이 없어서 잠들어 있던 Flusher가 작업 유무를 확인하기 위해서 일정 간격으로 깨어나야 할 필요가 있다. 이 값은 깨어난 누적 횟수다.

SIGNALED

어떤 작업의 빠른 처리를 위해서 Altibase는 잠든 Flusher에게 시그널을 주어서 깨울 수 있다. 이 값은 그 시그널에 의해 Flusher가 깨어난 횟수이다.

TOTAL_SLEEP_SEC

Flusher가 처리할 작업이 없어서 잠든 상태로 대기하고 있었던 시간의 총 합이다.

TOTAL_FLUSH_PAGES

체크포인트 플러시 또는 교체 플러시 중에 플러시된 페이지의 누적 개수이다

TOTAL_LOG_SYNC_USEC

데이터 페이지가 플러시될 때, WAL (Write Ahead Logging) 기법을 따라서 리두 로그가 먼저 디스크에 기록되어야 한다. 이 값은 리두 로그가 디스크에 기록되는데 소요된 시간의 누적 양이다.

TOTAL_DW_USEC

이 값은 doublewrite 버퍼의 내용을 디스크로 쓰는 데 걸린 시간의 누적 값이다. Doublewrite란 페이지들을 데이터 파일에 쓰기 전에, doublewrite buffer라 불리는 DW 파일에 먼저 기록하는 것을 말한다. Doublewrite buffer에 일단 기록된 후에, 그 페이지들은 데이터 파일의 올바른 위치에 다시 기록된다. 페이지를 데이터 파일에 기록하는 중에 운영 체제가 멈추거나 이들 데이터 파일이 손상된다면, 이 doublewrite 버퍼의 손상되지 않은 페이지를 이용해서 복구가 가능하다.

TOTAL_WRITE_USEC

데이터 페이지를 데이터 파일에 쓰는데 걸린 시간의 누적값이다. 이 값은 디스크에 플러시하는데 걸린 시간은 포함하지 않는다.

TOTAL_SYNC_USEC

데이터 페이지를 데이터 파일에 강제로 플러시 하는데 소요된 시간의 누적값이다.

TOTAL_FLUSH_TEMP_PAGES

플러시된 임시 페이지들의 누적 개수이다. (임시 페이지는 Sort 연산과 hash join을 할 때 사용되는 임시 테이블을 저장하는 데이터 페이지이다.)

TOTAL_TEMP_WRITE_USEC

임시 페이지들을 임시 파일에 기록하는데 걸린 시간의 누적값이다.

TOTAL_CALC_CHECKSUM_USEC

페이지에 오류가 있는지를 판단하기 위해 사용되는Checksum을 계산하는 데 걸린 시간의 누적 값이다.

DB_WRITE_PERF

데이터 페이지를 데이터 파일에 쓸 때 초당 기록된 bytes 수의 평균값으로 단위는 KB/Sec이다.

TEMP_WRITE_PERF

임시 페이지를 임시 파일에 쓸 때 초당 기록된 bytes 수의 평균값으로 단위는 KB/Sec이다.

V\$FLUSHINFO

버퍼 플러시 정보를 보여준다.

| Column name | Type | Description |
|---------------------------|---------|---|
| LOW_FLUSH_LENGTH | INTEGER | 교체 플러시(replacement flush)를 유발시킬 수 있는 최소한의 플러시 리스트 길이 |
| HIGH_FLUSH_LENGTH | INTEGER | 플러셔가 REPLACE_FLUSH_COUNT 값을 무시하고 플러시 리스트의 모든 버퍼를 플러시하는 플러시 리스트 길이 |
| LOW_PREPARE_LENGTH | INTEGER | 교체 플러시를 유발시킬 수 있는 최소한의 prepare 리스트 길이. 이 길이 이하가 되면 교체 플러시가 발생한다. |
| CHECKPOINT_FLUSH_COUNT | BIGINT | 체크포인트 플러시 수행시 플러시 할 버퍼의 개수 |
| FAST_START_IO_TARGET | BIGINT | 체크포인트 플러시 수행시 플러시 하지 않을 데이터 페이지의 개수 |
| FAST_START_LOGFILE_TARGET | INTEGER | 체크포인트 플러시 수행시 플러시 하지 않을 로그 파일의 개수 |
| REQ_JOB_COUNT | INTEGER | 현재 플러시 관리자에 등록된 작업의 개수 |

칼럼 정보

LOW_FLUSH_LENGTH

이는 교체 플러시(replacement flush)를 유발시킬 수 있는 최소한의 플러시 리스트 길이이다.

HIGH_FLUSH_LENGTH

이는 플러셔가 REPLACE_FLUSH_COUNT 값을 무시하고 플러시 리스트의 모든 버퍼를 플러시하는 플러시 리스트 길이이다.

LOW_PREPARE_LENGTH

이는 교체 플러시를 유발시킬 수 있는 최소한의 prepare 리스트 길이이다. 이 길이 이하가 되면 교체 플러시가 발생한다.

CHECKPOINT_FLUSH_COUNT

이는 체크포인트 플러시 수행시 플러시 할 버퍼의 개수이다.

FAST_START_IO_TARGET

이는 체크포인트 플러시 수행시 플러시 하지 않을 더티 페이지의 개수이다.

FAST_START_LOGFILE_TARGET

이는 체크포인트 플러시 수행시 플러시 하지 않을 로그 파일의 개수이다. 이들은 가장 최근에 생성된 로그 파일들이다.

REQ_JOB_COUNT

이는 플러시 관리자에 등록된 작업의 개수이다.

V\$INDEX

현재 데이터베이스에 존재하는 인덱스 정보를 보여준다.

| Column name | Type | Description |
|---------------|------------|---|
| TABLE_OID | BIGINT | 테이블 헤더의 객체 식별자 |
| INDEX_SEG_PID | INTEGER | 디스크 인덱스의 경우 인덱스 세그먼트 헤더 (header)의 페이지 식별자 |
| INDEX_ID | INTEGER | 인덱스 식별자 |
| INDEXTYPE | VARCHAR(7) | 해당 인덱스가 주 키 (primary key)로 사용되는지 일반 인덱스인지 식별하기 위한 구분자 |

칼럼 정보

TABLE_OID

이는 인덱스가 생성된 테이블의 객체 식별자로, 테이블 정보를 갖고 있는 헤더의 물리적인 위치를 저장한다.

INDEXTYPE

이 값은 해당 인덱스가 주 키 (primary key)로서 사용되는지 또는 일반 인덱스인지를 나타낸다.

- PRIMARY: 주 키로 사용되는 인덱스
- NORMAL: 일반 인덱스

V\$INSTANCE

현재 Altibase의 구동 단계, 구동된 시간, 구동 후 경과된 시간에 관한 정보를 보여준다.

| Column name | Type | Description |
|------------------|-------------|---|
| STARTUP_PHASE | VARCHAR(13) | 현재 구동 단계 |
| STARTUP_TIME_SEC | BIGINT | Altibase가 구동된 시각을 시스템 시간으로 나타낸다 (단위: seconds) |
| WORKING_TIME_SEC | BIGINT | 구동하여 지금까지 경과한 시간 |

V\$INTERNAL_SESSION

Altibase의 DBMS_CONCURRENT_EXEC 패키지에서 생성된 세션에 대한 정보를 보여준다. 더 자세한 컬럼에 대한 정보는 V\$SESSION의 컬럼 정보를 참조한다.

| Column name | Type | Description |
|--------------------|--------------|-------------------------------|
| ID | BIGINT | 세션 식별자 |
| TRANS_ID | BIGINT | 세션에서 현재 수행중인 트랜잭션의 식별자 |
| QUERY_TIME_LIMIT | BIGINT | 세션의 쿼리 시간 초과 |
| DDL_TIME_LIMIT | BIGINT | 세션의 DDL문 수행 시간 초과 |
| FETCH_TIME_LIMIT | BIGINT | 현재 세션의 Fetch 시간 초과 |
| UTRANS_TIME_LIMIT | BIGINT | 현재 세션의 갱신(update) 트랜잭션 시간 초과 |
| IDLE_TIME_LIMIT | BIGINT | 현재 세션의 Idle 시간 초과 |
| IDLE_START_TIME | INTEGER | 세션의 Idle상태로 진입한 시각 |
| ACTIVE_FLAG | INTEGER | 트랜잭션 활성 플래그 |
| OPENED_STMT_COUNT | INTEGER | 사용 중인 구문 개수 |
| DB_USERNAME | VARCHAR(128) | 데이터베이스 사용자 이름 |
| DB_USERID | INTEGER | 데이터베이스 사용자 식별자 |
| DEFAULT_TBSID | BIGINT | 사용자의 디폴트 테이블스페이스 식별자 |
| DEFAULT_TEMP_TBSID | BIGINT | 사용자의 디폴트 임시(temp) 테이블스페이스 식별자 |

| Column name | Type | Description |
|------------------------|--------------|--|
| SYSDBA_FLAG | INTEGER | sysdba 로 접속했는지 여부 |
| AUTOCOMMIT_FLAG | INTEGER | Autocommit 플래그 |
| SESSION_STATE | VARCHAR(13) | 세션의 상태 |
| ISOLATION_LEVEL | INTEGER | 세션의 고립 수준(isolation level) |
| REPLICATION_MODE | INTEGER | 이중화 모드 |
| TRANSACTION_MODE | INTEGER | 트랜잭션 모드 |
| COMMIT_WRITE_WAIT_MODE | INTEGER | 아래 참조 |
| OPTIMIZER_MODE | INTEGER | 최적화 모드 |
| HEADER_DISPLAY_MODE | INTEGER | SELECT 질의의 결과 출력시, 칼럼 이름만 출력할 것인지 테이블 이름도 함께 출력할 것인지 여부 0: 칼럼 이름과 함께 테이블 이름도 출력 1: 칼럼 이름만 출력 |
| CURRENT_STMT_ID | INTEGER | 사용 중인 구문 식별자 |
| STACK_SIZE | INTEGER | 질의 처리를 위한 스택의 크기(단위: bytes) |
| DEFAULT_DATE_FORMAT | VARCHAR(64) | 디폴트 날짜 형식 예) DD-MON-RRRR |
| TRX_UPDATE_MAX_LOGSIZE | BIGINT | DML 로그의 최대 크기(단위: bytes) |
| PARALLE_DML_MODE | INTEGER | Deprecated |
| LOGIN_TIME | INTEGER | 클라이언트 접속 시간 |
| FAILOVER_SOURCE | VARCHAR(256) | FailOver가 일어났을 때의 접속 정보 |
| NLS_TERRITORY | VARCHAR(40) | 세션의 지역 이름 |
| NLS_ISO_CURRENCY | VARCHAR(40) | 세션의 ISO 통화 기호 |
| NLS_CURRENCY | VARCHAR(10) | 세션의 지역 통화 기호 |
| NLS_NUMERIC_CHARACTERS | VARCHAR(2) | 세션의 소수점 문자와 그룹 구분자 |
| TIME_ZONE | VARCHAR(40) | 세션에 설정된 타임 존의 지역 이름, 약어 또는 UTC_OFFSET |
| LOB_CACHE_THRESHOLD | INTEGER | LOB_CACHE_THRESHOLD 프로퍼티에 설정된 값 |
| QUERY_REWRITE_ENABLE | VARCHAR(7) | QUERY_REWRITE_ENABLE 프로퍼티에 설정된 값 |

칼럼 정보

TRANS_ID

세션에서 현재 수행하고 있는 트랜잭션 식별자를 나타낸다. 현재 수행중인 트랜잭션이 없으면 이 값은 -1 이 된다.

ACTIVE_FLAG

세션이 어떤 구문을 수행하고 있을 경우 1로 나타난다. 그러나 단지 연결만 되어있거나, 트랜잭션을 커밋 또는 롤백한 후에는 0으로 표시된다.

SYSDBA_FLAG

접속된 세션이 sysdba 모드인지 아닌지를 나타낸다.

- 1: sysdba 모드

AUTOCOMMIT_FLAG

접속된 세션이 autocommit 모드인지를 나타낸다.

- 0: non-autocommit
- 1: autocommit

SESSION_STATE

| STATE | Description |
|------------------|--|
| INIT | 클라이언트로부터 요청이 들어오기를 기다리고 있는 상태 |
| AUTH | 사용자 인증을 마친 상태 |
| SERVICE READY | 서비스 준비상태 (트랜잭션을 만들 수 없는 상태로 XA 세션의 경우에만 이 상태로 올 수 있다.) |
| SERVICE | 서비스 상태 |
| END | 정상 종료 (트랜잭션이 있을 경우 커밋) 하고 있는 상태 |
| ROLLBACK | 비정상 종료 (트랜잭션이 있을 경우 ROLLBACK)하고 있는 상태. 클라이언트가 끊기거나 서버에서 세션을 강제로 끊을 때 발생한다. |
| UNKNOWN | 알 수 없는 상태 |

REPLICATION_MODE

세션의 이중화 모드를 나타낸다.

- 0: DEFAULT
- 16: NONE

TRANSACTION_MODE

트랜잭션 모드를 나타낸다.

- 0: READ/WRITE
- 4: READ ONLY

COMMIT_WRITE_WAIT_MODE

- 0: commit 시, 로그를 디스크에 기록할 때까지 기다리지 않는다.
- 1: commit 시, 로그를 디스크에 기록할 때까지 기다린다.

OPTIMIZER_MODE

해당 세션에 설정된 최적화 모드를 나타낸다.

- 1: 규칙 기반 (rule based)
- 0: 비용 기반 (cost based)

QUERY_REWRITE_ENABLE

세션에서 QUERY_REWRITE_ENABLE 프로퍼티에 설정된 값을 표시한다. QUERY_REWRITE_ENABLE 프로퍼티에 대해서는 2장을 참고하라.

- FALSE: Altibase 서버에서 쿼리 변환 시에 함수 기반 인덱스 미적용(disable)
- TRUE: Altibase 서버에서 쿼리 변환 시에 함수 기반 인덱스 적용(enable)

V\$LATCH

버퍼 풀의 BCB 래치 정보를 보여준다. 래치 정보에는 읽기 혹은 쓰기가 시도된 페이지에 대하여 래치 시도 횟수와 바로 래치를 잡는 횟수, 잡지 못한 횟수 등이 포함된다. 이 통계 정보는 각각 읽기/쓰기 래치로 구분하여 보여준다.

| Column name | Type | Description |
|--------------------|---------|----------------------|
| SPACE_ID | INTEGER | 테이블스페이스 식별자 |
| PAGE_ID | INTEGER | 페이지 식별자 |
| TRY_READ_LATCH | BIGINT | 읽기 래치 시도 횟수 |
| READ_SUCCESS_IMME | BIGINT | 읽기 래치를 바로 성공한 횟수 |
| READ_MISS | BIGINT | 읽기 래치를 바로 잡지 못한 횟수 |
| TRY_WRITE_LATCH | BIGINT | 쓰기 래치 시도 횟수 |
| WRITE_SUCCESS_IMME | BIGINT | 쓰기 래치를 바로 성공한 횟수 |
| WRITE_MISS | BIGINT | 쓰기 래치를 바로 잡지 못한 횟수 |
| SLEEPS_CNT | BIGINT | 래치를 잡기 위하여 sleep한 횟수 |

V\$LIBRARY

C/C++ Internal procedure에서 동적으로 로드한 라이브러리의 정보를 보여준다. 라이브러리 정보를 통해서 원하는 라이브러리를 제대로 로드했는지 확인할 수 있다.

| Column name | Type | Description |
|-------------|------------|-----------------|
| FILE_SPEC | CHAR(4000) | 동적 라이브러리 파일의 경로 |

| Column name | Type | Description |
|-----------------|-------------|---------------------------------------|
| REFERENCE_COUNT | INTEGER | 동적 라이브러리를 참조하는 Internal procedure의 개수 |
| FILE_SIZE | INTEGER | 동적 라이브러리의 파일 크기 (Bytes) |
| CREATE_TIME | VARCHAR(48) | 동적 라이브러리가 생성된 시간 |
| OPEN_TIME | VARCHAR(48) | 동적 라이브러리를 로드한 시간 |

칼럼 정보

FILE_SPEC

라이브러리 객체가 가리키는 동적 라이브러리 파일의 경로를 나타낸다. 라이브러리 파일이 위치하는 기본 경로 (\$ALTIBASE_HOME/lib)에 대한 상대 경로로 표시된다.

REFERENCE_COUNT

동적 라이브러리를 참조하는 Internal 저장 프로시저 또는 저장 함수의 개수를 나타낸다.

FILE_SIZE

동적 라이브러리 파일의 크기를 나타낸다. (단위 : Bytes)

CREATE_TIME

동적 라이브러리를 생성한 일시를 나타낸다. 파일 정보에서 얻어서 저장한다.

OPEN_TIME

동적 라이브러리를 로드한 일시를 나타낸다.

V\$LFG

이 뷰는 데이터베이스 관리자가 그룹 커밋의 동작을 모니터링 할 수 있는 통계 정보를 제공한다. 각 칼럼에 대한 보다 상세한 정보는 이 매뉴얼의 그룹 커밋 부분을 참조한다.

| Column name | Type | Description |
|-----------------------|---------|-------------------------------------|
| LFG_ID | INTEGER | 로그파일그룹 식별자 |
| CUR_WRITE_LF_NO | INTEGER | 기록중인 로그 파일 번호 |
| CUR_WRITE_LF_OFFSET | INTEGER | 기록중인 로그 파일 오프셋 |
| LF_OPEN_COUNT | INTEGER | 열린 로그파일의 개수 |
| LF_PREPARE_COUNT | INTEGER | 미리 생성한 로그파일의 개수 |
| LF_PREPARE_WAIT_COUNT | INTEGER | 새 로그파일로 스위치시 대기 횟수 |
| LST_PREPARE_LF_NO | INTEGER | 가장 최근에 미리 생성한 로그파일의 번호 |
| END_LSN_LFGID | INTEGER | 사용하지 않음(0) |
| END_LSN_FILE_NO | INTEGER | Altibase 재구동시 리두가 시작될 LSN의 파일 번호 부분 |

| Column name | Type | Description |
|-----------------------|---------|---|
| END_LSN_OFFSET | INTEGER | Altibase 재구동시 리두가 시작될 LSN의 파일 오프셋 부분 |
| FIRST_DELETED_LOGFILE | INTEGER | 삭제된 첫 번째 로그파일 |
| LAST_DELETED_LOGFILE | INTEGER | 삭제된 마지막 로그파일 |
| RESET_LSN_LFGID | INTEGER | 사용하지 않음(0) |
| RESET_LSN_FILE_NO | INTEGER | 특정 시점으로 복구 후 새 로그가 기록될 LSN의 파일번호 부분 |
| RESET_LSN_OFFSET | INTEGER | 특정 시점으로 복구 후 새 로그가 기록될 LSN의 오프셋 부분 |
| UPDATE_TX_COUNT | INTEGER | 현재 데이터베이스에 변경을 가하는 트랜잭션의 개수 (그룹커밋에서만 유효하다) |
| GC_WAIT_COUNT | INTEGER | 디스크 I/O를 기다린 횟수 (그룹커밋에서만 유효하다) |
| GC_ALREADY_SYNC_COUNT | INTEGER | 이미 디스크 I/O가 수행된 횟수 (그룹커밋에서만 유효하다) |
| GC_REAL_SYNC_COUNT | INTEGER | 그룹커밋 도중 실제 발생한 디스크 I/O 작업 횟수 (그룹커밋에서만 유효하다) |

칼럼 정보

LFG_ID

이는 0의 값을 가진 로그파일 그룹 고유번호이다.

CUR_WRITE_LF_NO

현재 로그를 기록하기 위해 사용하고 있는 로그 파일의 번호이다.

CUR_WRITE_LF_OFFSET

현재 로그를 기록하기 위해 사용하고 있는 로그 파일의 오프셋이다.

LF_OPEN_COUNT

디스크상에 존재하는 로그 파일 중 Altibase가 사용하기 위해 오픈 (Open)한 로그파일의 개수를 나타낸다.

LF_PREPARE_COUNT

로그파일 생성 스레드가 지금까지 미리 생성한 로그파일의 개수이다.

LF_PREPARE_WAIT_COUNT

Altibase는 기록중이던 로그파일을 다 사용하면 새로운 로그파일로 스위칭한다. 이 값은 사용할 로그파일을 미리 만들어 두지 못해서 로그 파일이 생성되기를 기다린 횟수를 나타낸다.

이 값이 크다면 PREPARE_LOG_FILE_COUNT 프로퍼티의 값을 더 큰 값으로 재설정하여 충분한 개수의 로그파일이 미리 만들어지도록 한다. PREPARE_LOG_FILE_COUNT 프로퍼티에 대한 설명은 2장을 참조한다.

LST_PREPARE_LF_NO

로그파일 생성 스레드가 가장 최근에 미리 생성한 로그파일의 번호이다.

END_LSN_FILE_NO

이 값은 Altibase 재구동 시 리두를 시작할 LSN (Log Sequence Number) 중 로그파일의 번호 부분이다. 최소한 이 LSN 이후의 로그는 반드시 리두 된다는 것을 보장할 수 있다.

END_LSN_OFFSET

이 값은 Altibase 재구동 시 리두를 시작할 LSN (Log Sequence Number) 중 로그파일 안의 오프셋 부분이다. 최소한 이 LSN 이후의 로그는 반드시 리두 된다는 것을 보장할 수 있다.

FIRST_DELETED_LOGFILE

이 값은 체크포인트 중 불필요한 로그파일로 분류되어 삭제된 로그파일 중 첫번째 로그파일의 번호이다. 이 칼럼의 값은 체크포인트 중에 해당 로그파일 번호의 로그파일까지 포함하여 삭제된 상태임을 의미한다.

LAST_DELETED_LOGFILE

이 값은 체크포인트 중 불필요한 로그파일로 분류되어 삭제된 로그파일 중 마지막 로그파일이다. 이 칼럼의 값은 체크포인트 중에 해당 로그파일까지 삭제된 상태임을 의미한다.

RESET_LSN_FILE_NO

RESET_LSN은 시스템 장애나 다른 이유로 인해 특정 시각까지만 데이터베이스를 복구한 후, 발생하는 새로운 작업들의 로그를 기록하는 LSN이다. 이 칼럼은 RESET_LSN 중 로그파일 번호 부분이다.

RESET_LSN_OFFSET

RESET_LSN 중 로그파일 안의 오프셋 부분을 나타낸다.

UPDATE_TX_COUNT

현재 데이터베이스에 변경을 가하는 트랜잭션 중 이 LFG에 속한 트랜잭션 수를 실시간으로 반환한다.

GC_WAIT_COUNT

그룹커밋을 위해 이 LFG에 속한 트랜잭션들이 디스크 I/O를 기다린 횟수를 보여준다.

GC_ALREADY_SYNC_COUNT

그룹커밋 도중 이 LFG에 속한 트랜잭션들을 위한 디스크 I/O가 이미 수행되었다면, 해당 트랜잭션에 대해서는 별도의 디스크 I/O를 수행할 필요가 없어진다. 이 값은 이것이 발생한 누적 횟수이다.

GC_REAL_SYNC_COUNT

그룹커밋 도중 이 LFG에 속한 트랜잭션들이 실제로 디스크 I/O를 수행한 횟수를 나타낸다.

V\$LOCK

현재 시점에서 데이터베이스의 모든 테이블에 대한 잠금(Lock) 노드 정보를 보여준다.

| Column name | Type | Description |
|----------------|-------------|------------------------------------|
| LOCK_ITEM_TYPE | VARCHAR(7) | 잠금 대상 객체의 종류 (Type) |
| TBS_ID | INTEGER | 테이블스페이스 식별자 |
| TABLE_OID | BIGINT | 테이블 객체 식별자 |
| DBF_ID | BIGINT | 데이터베이스 파일 식별자 |
| TRANS_ID | BIGINT | 트랜잭션 식별자 |
| LOCK_DESC | VARCHAR(32) | 잠금 모드를 가리키는 문자열 Ex) IX, IS, X |
| LOCK_CNT | INTEGER | 해당 잠금 노드의 잠금 개수 |
| IS_GRANT | BIGINT | 해당 테이블에 대하여 잠금을 잡고 있는지 대기하고 있는지 여부 |

칼럼 정보

LOCK_ITEM_TYPE

잠금 (Lock) 대상 객체 유형을 나타내며 다음의 값을 가진다.

| Value | Description |
|---------|---------------|
| NONE | 이 값을 가질 수 없음. |
| TBS | 테이블스페이스 |
| TBL | 테이블 |
| DBF | 데이터베이스 파일 |
| UNKNOWN | 객체 유형을 알 수 없음 |

V\$LOCK_STATEMENT

잠금 (Lock)을 잡고 있는 구문 (statement)과 잠금을 획득하기를 대기하고 있는 구문 (statement) 정보를 보여준다.

| Column name | Type | Description |
|-------------|----------------|---------------|
| SESSION_ID | INTEGER | 세션 식별자 |
| ID | INTEGER | statement 식별자 |
| TX_ID | BIGINT | 트랜잭션 식별자 |
| QUERY | VARCHAR(16384) | 질의문 |
| STATE | INTEGER | statement 상태 |

| Column name | Type | Description |
|----------------|-------------|------------------------------------|
| BEGIN_FLAG | INTEGER | statement 시작 여부를 알려주는 플래그 |
| LOCK_ITEM_TYPE | VARCHAR(7) | 잠금 대상 객체의 종류 (Type) |
| TBS_ID | INTEGER | 테이블스페이스 식별자 |
| TABLE_OID | BIGINT | 테이블 객체 식별자 |
| DBF_ID | BIGINT | 데이터베이스 파일 식별자 |
| LOCK_DESC | VARCHAR(32) | 잠금 모드를 가리키는 문자열 예) IX, IS, X |
| LOCK_CNT | INTEGER | 해당 잠금 노드의 잠금 개수 |
| IS_GRANT | BIGINT | 해당 테이블에 대하여 잠금을 잡고 있는지 대기하고 있는지 여부 |

V\$LOG

로그 앵커 정보를 보여준다.

| Column name | Type | Description |
|---------------------------|-------------|---|
| BEGIN_CHKPT_LFGID | INTEGER | 사용하지 않음(0) |
| BEGIN_CHKPT_FILE_NO | INTEGER | 가장 최근 수행된 체크포인트의 체크포인트 시작 로그의 로그 파일 번호 |
| BEGIN_CHKPT_FILE_OFFSET | INTEGER | 가장 최근 수행된 체크포인트의 체크포인트 시작 로그의 로그 오프셋 |
| END_CHKPT_LFGID | INTEGER | 사용하지 않음(0) |
| END_CHKPT_FILE_NO | INTEGER | 가장 최근 수행된 체크포인트의 체크포인트 종료 로그의 로그 파일 번호 |
| END_CHKPT_FILE_OFFSET | INTEGER | 가장 최근 수행된 체크포인트의 체크포인트 종료 로그의 로그 오프셋 |
| SERVER_STATUS | VARCHAR(15) | 서버의 상태를 나타낸다. |
| ARCHIVELOG_MODE | VARCHAR(12) | 데이터베이스의 아카이브 로그 모드 여부 |
| TRANSACTION_SEGMENT_COUNT | INTEGER | 언두 테이블스페이스에 생성할 트랜잭션 세그먼트의 개수 |
| OLDEST_LFGID | INTEGER | 사용하지 않음(0) |
| OLDEST_LOGFILE_NO | INTEGER | 재구동 복구 시에 디스크 관련 리두가 시작되는 로그 파일 번호 |
| OLDEST_LOGFILE_OFFSET | INTEGER | 재구동 복구 시에 디스크 관련 리두가 시작되는 로그 파일 오프셋(offset) |

칼럼 정보

SERVER_STATUS

이 값은 서버의 상태를 나타내는 문자열이다.

- SERVER SHUTDOWN: 종료된 상태
- SERVER STARTED: 동작중

ARCHIVELOG_MODE

데이터베이스의 아카이브 로그 모드 여부를 나타낸다.

- ARCHIVE: 이 모드에서는 미디어 복구 수행에 사용하기 위해 불필요한 로그 파일이 별도의 디렉터리에 저장된다.
- NOARCHIVE: 이 모드에서는 불필요한 로그 파일이 삭제된다.

V\$LOCK_WAIT

시스템에서 수행되는 트랜잭션 간의 대기 정보를 나타낸다.

| Column name | Type | Description |
|-------------------|--------|----------------|
| TRANS_ID | BIGINT | 대기 트랜잭션 식별자 |
| WAIT_FOR_TRANS_ID | BIGINT | 대기 대상 트랜잭션 식별자 |

칼럼 정보

TRANS_ID

현재 대기하고 있는 트랜잭션의 식별자이다.

WAIT_FOR_TRANS_ID

대기하고 있는 TRANS_ID의 트랜잭션이 어떠한 트랜잭션에 대해 대기하고 있는지를 나타내는 식별자이다.

```
SQL> select * from v$lock_wait;
V$LOCK_WAIT.TRANS_ID  V$LOCK_WAIT.WAIT_FOR_TRANS_ID
-----
1216                  2208
5344                  2208
2 rows selected.
```

위에 예제에서, 트랜잭션 2208에 대해서 트랜잭션 1216과 트랜잭션 5344가 현재 대기하고 있다.

V\$MEMGC

메모리 공간 회수 즉, 가비지 콜렉션 (memory garbage collection) 정보를 보여준다.

| Column name | Type | Description |
|-------------------------|--------------|--|
| GC_NAME | VARCHAR(128) | 가비지 콜렉터의 이름 MEM_LOGICAL_AGER: 구버전 인덱스 키 슬롯 해제 쓰레드 MEM_DELTHR: 삭제된 레코드를 해제하고 DROP TABLE 등 지연 (pending) 연산을 하는 쓰레드 |
| CURRSYSTEMVIEWSCN | VARCHAR(29) | 현재 시스템 view SCN |
| MINMEMSCNINTXS | VARCHAR(29) | 메모리 관련 트랜잭션의 view SCN 중 가장 작은 SCN |
| OLDESTTX | INTEGER | 가장 오래된 트랜잭션 식별자 (MINMEMSCNINTXS를 소유한 트랜잭션의 식별자) |
| SCNOFTAIL | VARCHAR(29) | 공간 회수 OID 리스트의 tail의 commit SCN |
| IS_EMPTY_OIDLIST | BIGINT | 공간 회수 OID 리스트가 비어 있는지 여부 0: 비어 있음 1: 비어 있지 않음 |
| ADD_OID_CNT | BIGINT | 공간 회수 처리를 위하여 OID 추가를 발생 시킨 트랜잭션의 개수 |
| GC_OID_CNT | BIGINT | 가비지 콜렉션으로 인해 OID를 회수한 횟 수 |
| AGING_REQUEST_OID_CNT | BIGINT | 공간 회수 처리를 요청한 OID의 개수 |
| AGING_PROCESSED_OID_CNT | BIGINT | 공간 회수 처리된 OID의 개수 |
| THREAD_COUNT | INTEGER | 공간 회수 쓰레드의 개수 |

칼럼 정보

Altibase는 MVCC를 지원하므로 하나의 레코드에 대해 여러 버전이 생길 수 있다. 즉 하나의 레코드는 1개의 최신버전과 다수의 구버전으로 구성된다. MVCC에 대한 자세한 내용은 *Getting Started Guide* 와 *Administrator's Manual*의 다중 버전 동시성 제어 (MVCC, Multi-Version Concurrency Control) 기법 부분을 참조한다.

AGING_REQUEST_OID_CNT

한 트랜잭션이 레코드 10건을 지우고 커밋할 경우, 10건의 구버전 레코드가 생기기 때문에 10건의 공간 회수 대상이 생긴다. 하지만 기존 ADD_OID_CNT는 트랜잭션 단위로 계산하기 때문에 1 증가한다. 이해 반해 AGING_REQUEST_OID_CNT는 OID 단위로 계산하기 때문에 10만큼 증가한다.

AGING_PROCESSED_OID_CNT

가비지 콜렉터(garbage collector 혹은 ager)가 하나의 가비지 콜렉션(garbage collection 혹은 aging) OID 리스트에 존재하는 구버전 레코드 10건을 지울 경우, GC_OID_CNT는 리스트 단위로 계산하기 때문에 1 증가한다. 이해 반해 AGING_PROCESSED_OID_CNT는 OID 단위로 계산하기 때문에 10 증가한다.

THREAD_COUNT

공간 회수(garbage collection, aging) 쓰레드 개수를 나타낸다.

V\$MEMSTAT

Altibase 프로세스가 사용하는 메모리의 통계 정보를 보여준다.

| Column name | Type | Description |
|----------------|----------|-------------------------------------|
| NAME | CHAR(64) | 메모리 모듈 이름 |
| ALLOC_SIZE | BIGINT | 해당 모듈의 메모리 사용량(단위: 바이트) |
| ALLOC_COUNT | BIGINT | 해당 모듈에서 ALLOC_SIZE를 구성하는 단위 메모리의 개수 |
| MAX_TOTAL_SIZE | BIGINT | 해당 모듈이 보유했던 최대 메모리 크기(단위: 바이트) |

칼럼 정보

NAME

Altibase가 사용하는 모듈 이름을 나타낸다. 이 칼럼은 다음의 메모리 모듈을 포함한다.

| 이름 | 설명 |
|---------------------|---|
| Altiwrap | Altiwrap을 위해 사용되는 메모리 |
| Async_IO_Manager | 비동기 I/O 발생시 사용되는 메모리 |
| Audit_Manager | Audit 관리자용 메모리 |
| CatalogCache_Memory | 현재 사용되지 않음 |
| Clock_Manager | 클록 (Clock) 관리자를 위한 메모리. 클록 관리자는 시스템 시간을 확인할 때 CPU 클록을 사용한다. |
| CM_Buffer | 통신(TCP, Unix Domain 소켓, IPC, IPCDA)을 위해 사용된 버퍼 메모리 |
| CM_DataType | 큰 패킷을 송수신하는데 사용되는 메모리 |
| CM_Interface | CM Interface에서 사용되는 메모리 |
| CM_Multiplexing | 통신을 위한 세션 정보 저장을 위해 사용되는 메모리 |
| CM_NetworkInterface | 각 통신 노드에 대한 정보를 저장하기 위해 사용되는 메모리 |
| Condition_Variable | 다중 쓰레드 제어를 위한 condition variables를 관리하는데 사용되는 메모리 |
| DatabaseLink | 데이터베이스 링크에 의해 사용되는 메모리 |
| Disaster_recovery | 재해 복구에서 사용되는 메모리 |

| 이름 | 설명 |
|---------------------------------------|--|
| Disaster_recovery_Control | 재해 복구의 역할 관리자가 사용하는 메모리 |
| Disaster_recovery_Executor | 재해 복구관련 실행시 사용되는 메모리 |
| Disaster_recovery_Storage | 현재 사용되지 않음 |
| Dynamic Module Loader | 공유 라이브러리 로딩시 사용되는 메모리 |
| External_Procedure | 익스터널 프로시저에서 사용되는 메모리 |
| External_Procedure_Agent | 익스터널 프로시저 에이전트에서 사용되는 메모리 |
| Fixed_Table | 고정(Fixed) 테이블에서 사용되는 메모리 |
| GIS_DataType | GIS 데이터를 처리하는데 사용되는 메모리 |
| GIS_Disk_Index | GIS 데이터를 위한 디스크 공간 인덱스를 관리하는데 사용되는 메모리 |
| GIS_Function | 공간 관련 계산에 사용되는 메모리 |
| GIS_TEMP_MEMORY | R-tree 인덱스 생성에 사용되는 메모리 |
| IDU_MEM_OTHER | 기타 용도 |
| Index_Memory | 인덱스 정보를 관리하는데 사용되는 메모리 |
| InMemoryRecovery_Memory | 현재 사용되지 않음 |
| Latch | Latch에서 사용되는 관리자용 메모리 |
| Legacy_Transaction_Manager | Legacy 트랜잭션 정보를 관리하기 위해 사용되는 메모리 |
| LOG_Memory | 현재 사용되지 않음 |
| Main_Module_CDBC_CONDITIONBUF_MEMPOOL | 현재 사용되지 않음 |
| Main_Module_CDBC_CURSORDATA_MEMPOOL | 현재 사용되지 않음 |
| Main_Module_CDBC_MAIN | 현재 사용되지 않음 |
| Main_Module_CDBC_QP | 현재 사용되지 않음 |
| Main_Module_CDBC_STATE_MEMPOOL | 현재 사용되지 않음 |
| Main_Module_Channel | Altibase 메인 모듈에 의해 사용되는 메모리 |
| Main_Module_DirectAttach | 현재 사용되지 않음 |
| Main_Module_Distributed | XA 관리를 위해 사용되는 메모리 |
| Main_Module_Queue | 큐를 위해 사용되는 메모리 |
| Main_Module_Thread | 쓰레드 관리를 위해 사용되는 메모리 |

| 이름 | 설명 |
|------------------------------|--|
| Main_Module_Utility | 현재 사용되지 않음 |
| Mathematics | 다양한 종류의 수학 연산을 위해 사용되는 메모리 |
| MMAP | mmap 시스템 콜로 할당받아 온 메모리 |
| Mutex | Mutex 관리자용 메모리 |
| OS_Independent | 현재 사용되지 않음 |
| Process_ThreadInfo | 현재 사용되지 않음 |
| Profile_Manager | 프로파일 관리자에 의해 사용되는 메모리 |
| Query_Binding | 호스트 변수 바인딩에 사용되는 메모리 |
| Query_Common | 기타 다른 목적으로 사용되는 메모리 |
| Query_Common_Remote_Call | 현재 사용되지 않음 |
| Query_Conversion | 현재 사용되지 않음 |
| Query_DML | DML 구문 실행을 위해 사용되는 메모리 |
| Query_Execute | 쿼리 실행시 사용되는 메모리 |
| Query_Execute_Cache | Deterministic 함수 결과의 캐시를 위해 사용되는 메모리 |
| Query_Result_Cache | Result 결과의 캐시를 위해 사용되는 메모리 |
| Query_Meta | 서버 동작 중에 사용되는 캐시된 메타 정보 관리를 위해 사용되는 메모리 |
| Query_Prepare | 실행을 위해 쿼리를 prepare하는데 사용되는 메모리 |
| Query_PSM_Concurrent_Execute | DBMS_CONCURRENT_EXEC 패키지를 실행하기 위해 사용되는 메모리 |
| Query_PSM_Execute | PSM (Persistent Stored Module) 실행을 위해 사용되는 메모리 |
| Query_PSM_Node | PSM에서 연관 배열을 위해 사용되는 메모리 |
| Query_Sequence | 시퀀스 관리를 위해 사용되는 메모리 |
| Query_Transaction | 트리거 실행을 위해 사용되는 메모리 |
| Remote_Call_Client | 현재 사용되지 않음 |
| Remote_Call_Server | 현재 사용되지 않음 |
| Replication_Common | 현재 사용되지 않음 |

| 이름 | 설명 |
|-------------------------------|---|
| Replication_Control | 이중화 관리자에 의해 사용되는 메모리 |
| Replication_Data | XLog 처리에 사용되는 메모리 |
| Replication_Executor | 현재 사용되지 않음 |
| Replication_Met | 메타 캐시에 의해 사용되는 메모리 |
| Replication_Module_Property | 현재 사용되지 않음 |
| Replication_Network | 이중화를 위한 통신에 사용되는 메모리 |
| Replication_Receiver | 이중화 수신자에 의해 사용되는 메모리 |
| Replication_Recovery | 이중화를 이용한 복구 수행시 사용되는 메모리 |
| Replication_Sender | 이중화 송신자에 의해 사용되는 메모리 |
| Replication_Storage | XLog를 적용하는데 사용되는 메모리 |
| Replication_Sync | 이중화에서 동기화를 위해 사용되는 메모리 |
| RESERVED | TLSF 메모리 관리자 사용시 할당받았으나 아직 분배하지 않은 영역 |
| Socket_Manager | 현재 사용되지 않음 |
| SQL Plan Cache Control | SQL Plan Cache 실행 시 사용되는 메모리 |
| Storage_DataPort | DataPort 실행 시 사용되는 메모리 |
| Storage_Disk_Buffer | 디스크 버퍼 관리자에 의해 사용되는 메모리 |
| Storage_Disk_Collection | 디스크 테이블에 대한 Direct-Path INSERT와 LOB 연산에 사용되는 메모리 |
| Storage_Disk_Datafile | I/O 버퍼와 데이터 파일 노드 생성 같은 데이터 파일 관리 작업에 사용되는 메모리 |
| Storage_Disk_Index | 디스크 인덱스 관리에 사용되는 메모리 |
| Storage_Disk_Page | 디스크 LOB 세그먼트 descriptor와 디스크 테이블 페이지 리스트 뮤텍스 할당에 사용되는 메모리 |
| Storage_Disk_Recovery | 디스크 데이터베이스의 일관성 보장을 위해 사용되는 메모리 |
| Storage_Disk_SecondaryBuffer | 보조 디스크 버퍼 관리자에 의해 사용되는 메모리 |
| Storage_Global_Memory_Manager | 현재 사용되지 않음 |

| 이름 | 설명 |
|--|---|
| Storage_Memory_Ager | 가비지 콜렉터와 데이터베이스 정제 (refining) 쓰레드가 사용하는 메모리 |
| Storage_Memory_Collection | 메모리 테이블의 레코드 관리를 위해 사용되는 메모리 |
| Storage_Memory_Index | 메모리 인덱스 관리를 위해 사용되는 메모리 |
| Storage_Memory_Interface | 스토리지 모듈 인터페이스 레벨에서 사용되는 메모리 |
| Storage_Memory_Locking | 테이블과 테이블스페이스 잠금에 사용되는 메모리 |
| Storage_Memory_Logical_Ager | 현재 사용되지 않음 |
| Storage_Memory_Manager | 메모리 데이터가 실제로 저장되는 메모리 |
| Storage_Memory_Page | 메모리 페이지 관리를 위해 사용되는 메모리 |
| Storage_Memory_Recovery | 복구 수행을 위해 사용되는 메모리 |
| Storage_Memory_Recovery_Archive_Thread | 현재 사용되지 않음 |
| Storage_Memory_Recovery_Chkpt_Thread | 현재 사용되지 않음 |
| Storage_Memory_Recovery_LFG_Thread | 현재 사용되지 않음 |
| Storage_Memory_Transaction | 트랜잭션 정보를 관리하기 위해 사용되는 메모리 |
| Storage_Memory_Utility | 스토리티 관리자 툴이 이용될 때 사용되는 메모리 |
| Storage_Tablespace | 테이블스페이스 노드를 관리하고 할당하는데 사용되는 메모리 |
| SYSTEM | malloc 함수를 이용하여 운영체제에서 직접 할당받은 메모리 |
| Tablespace Free Extent Pool | 테이블스페이스의 free 익스텐트 풀을 관리하기 위해 사용되는 메모리 |
| Temp_Memory | 임시 공간 할당시 사용되는 메모리 |
| Thread_Stack | 쓰레드가 생성될 때 쓰레드 스택용으로 사용하는 메모리 |
| Timer_Manager | 시스템 시간 확인 시 타이머 쓰레드를 사용하는 타이머 관리자를 위한 메모리 |

| 이름 | 설명 |
|-----------------------------------|---|
| Transaction_DiskPage_Touched_List | 트랜잭션에 의해 영향을 받은 디스크 데이터 페이지를 관리하기 위해 사용되는 메모리 |
| Transaction_OID_List | 메모리 데이터베이스의 OID (객체 식별자) 리스트를 만드는 데 사용되는 메모리 |
| Transaction_Private_Buffer | 현재 사용되지 않음 |
| Transaction_Segment_Table | 언두 세그먼트와 TSS (Transaction Status Slots)을 관리하는데 사용되는 메모리 |
| Transaction_Table | 트랜잭션 객체를 할당하는데 사용되는 메모리 |
| Transaction_Table_Info | 트랜잭션에 의해 변경되는 테이블 정보를 관리하는데 사용되는 메모리 |
| Utility_Module | 현재 사용되지 않음 |
| Volatile_Log_Buffer | 휘발성 로그 버퍼 메모리 |
| Volatile_Memory_Manager | 휘발성 메모리 데이터를 저장하는 메모리 |
| Volatile_Memory_Page | 휘발성 메모리 페이지를 관리하는데 사용되는 메모리 |
| WATCHDOG | 현재 사용되지 않음 |

ALLOC_SIZE

해당 모듈에서 사용하고 있는 메모리 사용량을 나타낸다.

ALLOC_COUNT

해당 모듈에서 ALLOC_SIZE를 구성하는 단위 메모리의 개수를 나타낸다.

MAX_TOTAL_SIZE

해당 모듈이 보유했던 최대 메모리 크기를 나타낸다.

V\$MEMTBL_INFO

메모리 테이블의 상태를 보여준다.

| Column name | Type | Description |
|--------------|----------|----------------------------|
| TABSPACE_ID | SMALLINT | 테이블스페이스 식별자 |
| TABLE_OID | BIGINT | 테이블 객체 식별자 |
| MEM_PAGE_CNT | BIGINT | 테이블의 고정 길이 칼럼이 저장되는 페이지 개수 |

| Column name | Type | Description |
|-------------------------|---------|--|
| MEM_VAR_PAGE_CNT | BIGINT | 테이블의 가변 길이 칼럼이 저장되는 페이지 개수 |
| MEM_SLOT_PERPAGE | INTEGER | 고정 길이 칼럼이 저장되는 페이지 하나에 들어갈 수 있는 슬롯(slot)의 개수 |
| MEM_SLOT_SIZE | BIGINT | 테이블 레코드의 고정 영역의 크기 |
| FIXED_ALLOC_MEM | DOUBLE | 테이블에 할당한 고정 영역 메모리 크기 (단위: 바이트) |
| FIXED_USED_MEM | BIGINT | 테이블에서 실제 사용하고 있는 고정 영역 메모리 크기 (단위: 바이트) |
| VAR_ALLOC_MEM | DOUBLE | 테이블에 할당한 가변 영역 메모리 크기 (단위: 바이트) |
| VAR_USED_MEM | BIGINT | 테이블에서 실제 사용하고 있는 가변 영역 메모리 크기 (단위: 바이트) |
| MEM_FIRST_PAGEID | BIGINT | 테이블의 고정 페이지 중 제일 앞에 있는 페이지 번호 |
| STATEMENT_REBUILD_COUNT | BIGINT | statement를 재구성 (rebuild)한 횟수 |
| UNIQUE_VIOLATION_COUNT | BIGINT | 유일 키 제약조건이 위반된 횟수 |
| UPDATE_RETRY_COUNT | BIGINT | 갱신 시 재시도 횟수 |
| DELETE_RETRY_COUNT | BIGINT | 삭제 시 재시도 횟수 |
| COMPRESSED_LOGGING | INTEGER | 로그 압축 여부 |
| IS_CONSISTENT | INTEGER | 테이블의 일관성 여부 |

테이블 이름을 포함하여 보려면 다음과 같이 SYS_TABLES_ 메타 테이블과 조인하여 질의를 하여야 한다.

```
SELECT A.TABLE_NAME,
       B.MEM_PAGE_CNT,
       B.MEM_SLOT_SIZE,
       B.MEM_FIRST_PAGEID
FROM   SYSTEM_.SYS_TABLES_ A, V$MENTBL_INFO B
WHERE  A.TABLE_OID = B.TABLE_OID;
```

칼럼 정보

TABLESPACE_ID

해당 테이블이 저장되어 있는 테이블스페이스의 식별자이다. 다음의 테이블스페이스가 기본으로 생성된다. 사용자가 새로 생성하는 테이블스페이스의 식별자는 4보다 큰 값이다.

- 0: SYS_TBS_MEM_DIC
- 1: SYS_TBS_MEM_DATA
- 2: SYS_TBS_DISK_DATA

- 3: SYS_TBS_DISK_UNDO
- 4: SYS_TBS_DISK_TEMP

TABLE_OID

이는 테이블의 객체 식별자로, 테이블 정보를 갖고 있는 헤더의 물리적인 위치를 가리킨다. 이 값은 시스템에 의해 내부적으로만 사용된다.

STATEMENT_REBUILD_COUNT

Prepare-Execute할 때 한번 Prepare된 statement는 구문분석 (Parsing), 유효성 검사 (Validation), 최적화 (Optimizing) 없이 실행만 한다. 그런데 statement가 Prepare된 후 질의 대상 객체 (테이블스페이스, 테이블, 색인 등)에 대해 DDL이 수행된 경우, 실행시에 statement는 자동으로 재구성 (rebuild)되며 그 때마다 이 값은 증가된다.

UNIQUE_VIOLATION_COUNT

유일 키 제약조건이 위반될 때, 이 값이 증가된다.

UPDATE_RETRY_COUNT

갱신이 재시도될 때 이 값이 증가된다.

DELETE_RETRY_COUNT

삭제가 재시도될 때 이 값이 증가된다

V\$MEM_BTREE_HEADER

메모리 BTREE의 헤더 정보를 보여준다.

| Column name | Type | Description |
|--------------------|--------------|--------------------------|
| INDEX_NAME | VARCHAR(128) | 인덱스 이름 |
| INDEX_ID | INTEGER | 인덱스 식별자 |
| INDEX_TBS_ID | INTEGER | 인덱스가 저장되어 있는 테이블스페이스 식별자 |
| TABLE_TBS_ID | INTEGER | 테이블이 저장되어 있는 테이블스페이스 식별자 |
| IS_UNIQUE | CHAR(1) | 유일 키 인덱스 여부 |
| IS_NOT_NULL | CHAR(1) | 널 (NULL) 허용 여부 |
| USED_NODE_COUNT | INTEGER | 인덱스가 사용중인 노드의 개수 |
| PREPARE_NODE_COUNT | INTEGER | 노드 요구를 대비하여 미리 할당된 노드 개수 |
| BUILT_TYPE | CHAR(1) | 인덱스 생성시 사용된 키 타입 |

칼럼 정보

INDEX_NAME

인덱스의 이름이다.

INDEX_ID

해당 인덱스가 갖는 시스템 내에서 고유한 식별자이다.

INDEX_TBS_ID

인덱스가 저장되어 있는 테이블스페이스 식별자이다.

TABLE_TBS_ID

해당 인덱스가 생성된 테이블이 저장되어 있는 테이블스페이스 식별자이다.

IS_UNIQUE

유일 키 인덱스 여부를 나타낸다. 유일 키 인덱스는 'T'를 갖고, 중복키 인덱스의 경우는 'F'를 갖는다.

IS_NOT_NULL

널(NULL)의 허용 여부를 나타낸다. 주 키 (primary key) 인덱스의 경우는 'F'를 갖고, 나머지 인덱스는 'T'를 갖는다.

USED_NODE_COUNT

현재 인덱스에 달려있는 노드의 총 개수를 의미한다. 이 개수는 노드 분할시에 증가되고, 노드 삭제시에 감소된다.

PREPARE_NODE_COUNT

노드 할당에 따른 시스템 부하를 고려하여 미리 할당받아 둔 노드의 개수를 의미한다.

BUILT_TYPE

인덱스 생성 시 키 값을 사용했는지 레코드 포인터를 사용했는지를 나타낸다. 키 값으로 생성되었을 경우 'V'를 갖고, 레코드 포인터로 생성되었을 경우 'P'를 갖는다.

V\$MEM_BTREE_NODEPOOL

메모리 BTREE 인덱스를 위한 노드 풀 정보를 보여준다. 해당 노드 풀은 모든 메모리 BTREE 인덱스의 노드 할당과 반환을 관리한다.

| Column name | Type | Description |
|------------------|---------|--------------------------|
| TOTAL_PAGE_COUNT | INTEGER | 노드 풀의 전체 페이지 수 |
| TOTAL_NODE_COUNT | INTEGER | 노드 풀의 전체 노드 수 |
| FREE_NODE_COUNT | INTEGER | 노드 풀 내에서 할당되지 않은 노드 수 |
| USED_NODE_COUNT | INTEGER | 인덱스로 할당된 노드 수 |
| NODE_SIZE | INTEGER | 노드의 크기 (바이트) |
| TOTAL_ALLOC_REQ | BIGINT | 노드 풀에 요청된 노드 할당 횟수 (누적값) |

| Column name | Type | Description |
|----------------|---------|--------------------------|
| TOTAL_FREE_REQ | BIGINT | 노드 풀에 요청된 노드 삭제 횟수 (누적값) |
| FREE_REQ_COUNT | INTEGER | 노드 풀에서 삭제 대기중인 노드 수 |

칼럼 정보

TOTAL_PAGE_COUNT

BTREE 인덱스를 위한 노드 풀에 할당된 페이지의 개수를 나타낸다.

TOTAL_NODE_COUNT

BTREE 인덱스를 위한 노드 풀에 할당된 노드의 개수를 나타낸다. TOTAL_PAGE_COUNT와 NODE_SIZE에 의해 결정된다.

FREE_NODE_COUNT

BTREE 인덱스에 할당되지 않고 노드 풀에 남아 있는 노드 수를 나타낸다.

USED_NODE_COUNT

현재 BTREE 인덱스에 할당된 노드의 총 수를 나타낸다.

NODE_SIZE

하나의 BTREE 인덱스 노드 크기를 나타낸다.

TOTAL_ALLOC_REQ

노드 풀에 요청된 노드 할당 횟수를 나타낸다. 시스템이 시작된 후부터 누적된 값을 유지한다.

TOTAL_FREE_REQ

인덱스에서 사용되었던 노드가 삭제되어 노드 풀에 반환 요청된 횟수를 나타낸다. 시스템이 시작된 후부터 누적된 값을 유지한다.

FREE_REQ_COUNT

삭제 대기중인 BTREE 인덱스에 사용되었던 노드 수를 나타낸다.

V\$MEM_RTREE_HEADER

메모리 RTREE 인덱스의 헤더 정보를 보여준다.

| Column name | Type | Description |
|----------------|----------|--------------------------|
| INDEX_NAME | CHAR(40) | 인덱스 이름 |
| INDEX_ID | INTEGER | 인덱스 식별자 |
| TABLE_TBS_ID | INTEGER | 테이블이 저장되어 있는 테이블스페이스 식별자 |
| TREE_MBR_MIN_X | DOUBLE | RTREE 인덱스의 최소 X 값 |
| TREE_MBR_MIN_Y | DOUBLE | RTREE 인덱스의 최소 Y 값 |
| TREE_MBR_MAX_X | DOUBLE | RTREE 인덱스의 최대 X 값 |

| Column name | Type | Description |
|--------------------|---------|--------------------------|
| TREE_MBR_MAX_Y | DOUBLE | RTREE 인덱스의 최대 Y 값 |
| USED_NODE_COUNT | INTEGER | 인덱스가 사용 중인 노드의 개수 |
| PREPARE_NODE_COUNT | INTEGER | 노드 요구를 대비하여 미리 할당된 노드 개수 |

칼럼 정보

INDEX_NAME

인덱스의 이름이다.

INDEX_ID

해당 인덱스가 갖는 시스템 내에서 고유한 식별자이다.

TABLE_TBS_ID

해당 인덱스와 연결되어 있는 테이블의 테이블스페이스 식별자이다.

TREE_MBR_MIN_X

해당 RTREE 인덱스의 최소 경계 사각형들 중 최소 X 값을 나타낸다.

TREE_MBR_MIN_Y

해당 RTREE 인덱스의 최소 경계 사각형들 중 최소 Y 값을 나타낸다.

TREE_MBR_MAX_X

해당 RTREE 인덱스의 최소 경계 사각형들 중 최대 X 값을 나타낸다.

TREE_MBR_MAX_Y

해당 RTREE 인덱스의 최소 경계 사각형들 중 최대 Y 값을 나타낸다.

USED_NODE_COUNT

현재 인덱스에 달려있는 노드의 총 개수를 의미한다. 해당 개수는 노드 분할시에 증가되고, 노드 삭제 시에 감소된다.

PREPARE_NODE_COUNT

노드 할당에 따른 시스템 부하를 고려하여 미리 할당받은 노드의 개수를 의미한다.

V\$MEM_RTREE_NODEPOOL

메모리 RTREE 인덱스를 위한 노드 풀 정보를 보여준다. 해당 노드 풀은 모든 메모리 RTREE 인덱스의 노드 할당과 반환을 관리한다.

| Column name | Type | Description |
|------------------|---------|-----------------------|
| TOTAL_PAGE_COUNT | INTEGER | 노드 풀의 전체 페이지 수 |
| TOTAL_NODE_COUNT | INTEGER | 노드 풀의 전체 노드 수 |
| FREE_NODE_COUNT | INTEGER | 노드 풀 내에서 할당되지 않은 노드 수 |

| Column name | Type | Description |
|-----------------|---------|-------------------------|
| USED_NODE_COUNT | INTEGER | 인덱스로 할당된 노드 수 |
| NODE_SIZE | INTEGER | 노드의 크기 (바이트) |
| TOTAL_ALLOC_REQ | BIGINT | 노드 풀에 요청된 노드 할당 횟수(누적값) |
| TOTAL_FREE_REQ | BIGINT | 노드 풀에 요청된 노드 삭제 횟수(누적값) |
| FREE_REQ_COUNT | INTEGER | 노드 풀에서 삭제 대기중인 노드 수 |

칼럼 정보

TOTAL_PAGE_COUNT

RTREE 인덱스의 노드 풀에 할당된 페이지의 수를 나타낸다.

TOTAL_NODE_COUNT

RTREE 인덱스의 노드 풀에 할당된 노드의 수를 나타낸다. TOTAL_PAGE_COUNT와 NODE_SIZE에 의해 결정된다.

FREE_NODE_COUNT

RTREE 인덱스에 할당되지 않고 노드 풀에 남아 있는 노드 수를 나타낸다.

USED_NODE_COUNT

RTREE 인덱스에 할당된 노드의 총 수를 나타낸다.

NODE_SIZE

하나의 RTREE 인덱스 노드 크기를 나타낸다.

TOTAL_ALLOC_REQ

노드 풀에 요청된 노드 할당 횟수를 나타낸다. 시스템이 시작된 후부터 누적된 값을 유지한다.

TOTAL_FREE_REQ

인덱스에서 사용되었던 노드가 삭제되어 노드 풀에 반환 요청된 횟수를 나타낸다. 시스템이 시작된 후부터 누적된 값을 유지한다.

FREE_REQ_COUNT

RTREE 인덱스에서 사용되었던 노드가 삭제 대기중인 노드 수를 나타낸다.

V\$MEM_TABLESPACES

메모리에 생성된 테이블스페이스 정보를 보여준다.

| Column name | Type | Description |
|--------------|--------------|-------------|
| SPACE_ID | INTEGER | 테이블스페이스 식별자 |
| SPACE_NAME | VARCHAR(512) | 테이블스페이스 이름 |
| SPACE_STATUS | INTEGER | 테이블스페이스 상태 |

| Column name | Type | Description |
|---------------------|-------------|------------------------------|
| SPACE_SHM_KEY | INTEGER | 테이블스페이스의 공유 메모리 키 |
| AUTOEXTEND_MODE | INTEGER | 테이블스페이스의 자동 확장 모드 |
| AUTOEXTEND_NEXTSIZE | BIGINT | 자동 확장시 확장되는 크기 (bytes) |
| MAXSIZE | BIGINT | 테이블스페이스의 최대 크기 (bytes) |
| CURRENT_SIZE | BIGINT | 테이블스페이스의 현재 크기 (bytes) |
| DBFILE_SIZE | DOUBLE | 데이터베이스 이미지 파일의 크기(bytes) |
| DBFILE_COUNT_0 | INTEGER | 파일 그룹이 0번인 데이터베이스 이미지 파일의 개수 |
| DBFILE_COUNT_1 | INTEGER | 파일 그룹이 1번인 데이터베이스 이미지 파일의 개수 |
| TIMESTAMP | VARCHAR(64) | 테이블스페이스 생성 시각 |
| ALLOC_PAGE_COUNT | BIGINT | 테이블스페이스의 전체 페이지 개수 |
| FREE_PAGE_COUNT | BIGINT | 테이블스페이스의 프리(Free) 페이지 개수 |
| RESTORE_TYPE | BIGINT | 메모리에 테이블스페이스를 올리는 방법 |
| CURRENT_DB | INTEGER | 핑퐁 체크포인트 대상 파일 집합 |
| HIGH_LIMIT_PAGE | BIGINT | 테이블스페이스가 가질 수 있는 최대 페이지 개수 |
| PAGE_COUNT_PER_FILE | BIGINT | 데이터베이스 이미지 파일당 페이지 개수 |
| PAGE_COUNT_IN_DISK | INTEGER | 디스크에 존재하는 페이지의 개수 |

칼럼 정보

SPACE_STATUS

테이블스페이스 상태 값이다. 자세한 내용은 V\$MEM_TABLESPACE_STATUS_DESC를 참고한다.

SPACE_SHM_KEY

테이블스페이스가 공유 메모리에 적재되었을 때 사용되는 공유 메모리 키를 나타낸다.

AUTOEXTEND_MODE

자동확장 (Autoextend) 모드 여부를 나타낸다. 1 이면 자동확장으로 설정된 상태이며, 1이 아니면 설정되지 않은 상태이다.

AUTOEXTEND_NEXTSIZE

자동 확장시 확장되는 크기 (bytes)이다.

MAXSIZE

테이블스페이스의 최대 크기 (bytes)이다.

CURRENT_SIZE

현재 테이블스페이스 크기 (bytes)를 나타낸다.

DBFILE_SIZE

테이블스페이스의 데이터베이스 이미지 파일의 크기 (bytes)를 나타낸다.

DBFILE_COUNT_0

Altibase는 핑퐁 체크포인트 방식을 사용하기 때문에 각 데이터베이스 이미지 파일 (database Image file) 별로 두 개씩 유지하는데, 이 중 0번 파일 그룹에 해당하는 파일 개수이다.

DBFILE_COUNT_1

Altibase는 핑퐁 체크포인트 방식을 사용하기 때문에 각 데이터베이스 이미지 파일 (database Image file) 별로 두 개씩 유지하는데, 이 중 1번 파일 그룹에 해당하는 파일 개수이다.

TIMESTAMP

테이블스페이스 생성 시점의 타임스탬프 값을 가진다.

ALLOC_PAGE_COUNT

테이블스페이스가 가지고 있는 페이지의 개수를 나타낸다.

FREE_PAGE_COUNT

테이블스페이스의 빈 (free) 페이지 개수를 나타낸다.

RESTORE_TYPE

테이블스페이스를 메모리에 올리는 방법이다. 다음의 값을 갖는다.

| 적재 방법 | 값 | 설명 |
|-------------------------|---|---|
| RESTORE_TYPE_DYNAMIC | 0 | 동적 메모리에 올린다. |
| RESTORE_TYPE_SHM_CREATE | 1 | 공유 메모리를 생성해서 테이블스페이스를 공유 메모리에 올린다. |
| RESTORE_TYPE_SHM_ATTACH | 2 | 테이블스페이스를 공유 메모리에 Attach한다. 이미 데이터베이스가 공유 메모리에 올라와 있는 상태에서 공유 메모리를 프로세스에 Attach한다. |

CURRENT_DB

체크포인트 시 더티 페이지 (Dirty Page, 변경된 페이지)가 내려가는 데이터베이스 이미지 파일 그룹으로 0 혹은 1 값을 가진다.

HIGH_LIMIT_PAGE

테이블스페이스가 가질 수 있는 최대 페이지 개수를 나타낸다.

PAGE_COUNT_PER_FILE

데이터베이스 이미지 파일 당 페이지의 개수를 나타낸다.

PAGE_COUNT_IN_DISK

디스크에 존재하는 데이터베이스 이미지 파일들의 전체 페이지의 개수이다. Altibase는 데이터베이스 확장 시 디스크에서 파일이 바로 확장되는 것이 아니라 체크포인트 시에 확장되기 때문에 메모리에 존재하는 데이터베이스 페이지 개수와 디스크에 존재하는 페이지 개수가 다를 수 있다.

V\$MEM_TABLESPACE_CHECKPOINT_PATHS

특정 테이블스페이스에 대해서 체크포인트 발생 시 변경된 페이지 (Dirty Page)가 반영되는 데이터베이스 이미지 파일의 위치 즉 디렉터리 경로를 보여준다.

| Column name | Type | Description |
|-----------------|--------------|-----------------------------|
| SPACE_ID | INTEGER | 테이블스페이스 식별자 |
| CHECKPOINT_PATH | VARCHAR(512) | 데이터베이스 이미지 파일들이 위치한 디렉터리 경로 |

V\$MEM_TABLESPACE_STATUS_DESC

메모리 테이블스페이스의 상태를 나타내는 값과 그에 대한 설명을 보여준다. 이 값은 V\$MEM_TABLESPACES 성능 뷰의 SPACE_STATUS 칼럼이 가질 수 있는 값이다.

| Column name | Type | Description |
|-------------|-------------|-------------------|
| STATUS | INTEGER | 메모리 테이블스페이스의 상태 값 |
| STATUS_DESC | VARCHAR(64) | 상태 값에 대한 설명 |

칼럼 정보**STATUS**

메모리 테이블스페이스의 상태 값을 나타낸다.

STATUS_DESC

메모리 테이블스페이스의 상태 값에 대한 설명을 나타낸다.

메모리 테이블스페이스의 상태 값과 설명은 다음과 같다.

| STATUS_DESC | Description |
|-------------|---------------------------|
| OFFLINE | 테이블스페이스가 오프라인 상태이다. |
| ONLINE | 테이블스페이스가 온라인 상태이다. |
| DISCARDED | 테이블스페이스가 폐기 (DISCARD)되었다. |
| DROPPED | 테이블스페이스가 삭제되었다. |
| BACKUP | 테이블스페이스 백업 중이다. |
| CREATING | 테이블스페이스 생성 중이다. |

| STATUS_DESC | Description |
|----------------------|---|
| DROPPING | 테이블스페이스 삭제 요청이 된 상태이다. |
| DROP_PENDING | 테이블스페이스 삭제 중이다. |
| SWITCHING_TO_OFFLINE | 테이블스페이스가 오프라인 상태로 바뀌고 있다. |
| SWITCHING_TO_ONLINE | 테이블스페이스가 온라인 상태로 바뀌고 있다. |
| BLOCK_BACKUP | 테이블스페이스에 대해서 백업할 수 없다. 현재 다른 연산을 수행하는 중이므로 백업은 이 연산이 완료된 후에 할 수 있다. |

V\$MUTEX

Altibase 프로세스에서 사용되고 있는 동시성 제어와 관련된 뮤텝스 통계 정보를 보여준다.

| Column name | Type | Description |
|--------------------|-------------|-------------------------------------|
| NAME | VARCHAR(64) | 뮤텝스 이름 |
| TRY_COUNT | BIGINT | 잠금 (Lock) 시도 횟수 |
| LOCK_COUNT | BIGINT | 잠금 성공 횟수 |
| MISS_COUNT | BIGINT | 잠금을 잡지 못하여 대기한 횟수 |
| SPIN_VALUE | INTEGER | 향후 확장 예정 |
| TOTAL_LOCK_TIME_US | BIGINT | 잠금을 잡고 있던 시간의 총합 (microseconds) |
| MAX_LOCK_TIME_US | BIGINT | 잠금을 잡고 있던 시간 중 최대 시간 (microseconds) |
| THREAD_ID | VARCHAR(64) | 현재 잠금을 잡고 있는 스레드 ID |

V\$NLS_PARAMETERS

서버 및 클라이언트의 NLS (National Language Support) 관련 정보를 세션 단위로 보여준다.

| Column name | Type | Description |
|---------------------------|-------------|--------------------------------|
| SESSION_ID | BIGINT | 세션 식별자 |
| NLS_USE | VARCHAR(40) | 클라이언트의 문자 집합 |
| NLS_CHARACTERSET | VARCHAR(40) | 데이터베이스 문자 집합 |
| NLS_NCHAR_CHARACTERSET | VARCHAR(40) | 국가 문자 집합 |
| NLS_COMP | VARCHAR(7) | 문자 비교 방법 |
| NLS_NCHAR_CONV_EXCP | VARCHAR(7) | 문자 집합 변환시 에러 처리 방법 |
| NLS_NCHAR_LITERAL_REPLACE | VARCHAR(7) | SQL문 내에 NCHAR 리터럴이 존재하는지 검사 여부 |

칼럼 정보

SESSION_ID

세션의 고유 번호를 나타낸다.

NLS_USE

클라이언트의 문자 집합 (Character set)을 나타낸다. 클라이언트에서 문자 데이터를 처리할 때 사용할 기본 문자 집합을 지정한다. 현재 Altibase에서 지원하는 문자 집합과 그에 해당하는 NLS_USE 설정은 아래와 같다.

| 언어 | 문자 집합 | NLS_USE |
|----------|---------------------|--|
| 영어 (기본값) | US7ASCII | US7ASCII, ASCII, ENGLISH |
| 한글 | KSC-5601 완성형 | KSC5601, KO16KSC5601, KOREAN |
| | MS 확장 완성형 | MS949, CP949, WINDOWS949 |
| 일어 | EUC-JP (UNIX) | EUCJP |
| | Shift-JIS (Windows) | SHIFTJIS |
| | MS932 (Windows) | MS932, CP932 |
| 중국어 | 중국 | GB231280, ZHS16CGB231280, CHINESE, MS936 |
| | 대만 | BIG5, ZHT16BIG5, TAIWAN |
| 공통 | 유니코드 (UTF-8) | UTF8, UNICODE |

데이터베이스 문자 집합과 다른 문자 집합의 데이터를 저장할 경우 문자 집합 간의 변환 및 호환성을 고려해야 한다. 다국어 지원에 대한 보다 자세한 내용은 *Getting Started Guide* 를 참조한다.

NLS_CHARACTERSET

서버의 데이터베이스 문자 집합 (database character set)을 나타낸다.

NLS_NCHAR_CHARACTERSET

국가 문자 집합 (national character set)을 나타낸다.

NLS_COMP

데이터베이스 생성시 지정한 문자 집합에 해당하는 언어의 사전에 나오는 문자 순서대로 비교하는 것을 나타낸다. 현재는 한글 (KSC-5601 완성형 또는 MS 확장 완성형)로 설정된 경우에만 지원한다.

NLS_NCHAR_CONV_EXCP

문자 집합 변환시 오류 처리를 어떻게 할 것인지를 보여준다.

NLS_NCHAR_LITERAL_REPLACE

클라이언트가 SQL문 내에 NCHAR 리터럴이 있는지 검사하는 여부를 나타내는 칼럼으로, TRUE 또는 FALSE가 나올 수 있다. TRUE일 경우에는 클라이언트가 SQL문 내에 NCHAR 리터럴이 있는지 매번 검사하여 NCHAR 리터럴을 제외한 부분만 데이터베이스 문자 집합으로 변환하고, FALSE일 경우에는 검사하지 않고 SQL문 전체를 데이터베이스 문자 집합으로 변환한다.

V\$NLS_TERRITORY

데이터베이스 또는 현재 세션에 설정 가능한 지역의 이름이 저장되어 있는 성능 뷰이다.

| Column name | Type | Description |
|-------------|-------------|---------------|
| NAME | VARCHAR(40) | 설정 가능한 지역의 이름 |

V\$OBSOLETE_BACKUP_INFO

더 이상 유지할 필요가 없는 백업에 대한 정보를 보여준다.

이 뷰의 칼럼들은 V\$BACKUP_INFO 성능 뷰의 일부이므로 자세한 내용은 V\$BACKUP_INFO 성능 뷰의 칼럼 정보를 참고하도록 한다.

| Column name | Type | Description |
|--------------------------------|-----------|-----------------------|
| BEGIN_BACKUP_TIME | CHAR(24) | 백업 시작 일시 |
| END_BACKUP_TIME | CHAR(24) | 백업 완료 일시 |
| INCREMENTAL_BACKUP_CHUNK_COUNT | INTEGER | Incremental chunk의 크기 |
| BACKUP_TARGET | INTEGER | 백업 대상 |
| BACKUP_LEVEL | INTEGER | 백업 레벨 |
| BACKUP_TYPE | INTEGER | 백업 유형 |
| TABSPACE_ID | INTEGER | 백업 대상 테이블스페이스 ID |
| FILE_ID | INTEGER | 백업 대상 데이터파일 ID |
| BACKUP_TAG | CHAR(128) | 백업 태그 이름 |
| BACKUP_FILE | CHAR(512) | 백업 파일 |

V\$PKGTEXT

시스템에서 수행되는 패키지의 문자열 정보를 나타낸다.

| Column name | Type | Description |
|-------------|-------------|----------------|
| PACKAGE_OID | BIGINT | 패키지 식별자 |
| PIECE | INTEGER | 문자열 조각의 일련 번호 |
| TEXT | VARCHAR(64) | 패키지 구문의 문자열 조각 |

칼럼 정보

PACKAGE_OID

패키지를 유일하게 가리키는 객체 식별자, 즉 OID이다.

PIECE

패키지의 전체 구문을 64바이트 길이의 문자열로 나누어 저장한다. PIECE는 나뉘어진 64바이트 조각의 일련 번호로 0부터 시작된다.

TEXT

패키지 텍스트의 일부분인 64바이트 텍스트 조각의 내용을 나타낸다.

V\$PLANTEXT

서버에서 수행되는 SQL의 실행계획 (execution plan) 정보를 나타낸다.

| Column name | Type | Description |
|-------------|-------------|--------------------|
| SID | INTEGER | 세션 식별자 |
| STMT_ID | INTEGER | 문장(statement) 식별자 |
| PIECE | INTEGER | 실행계획 문자열 조각의 일련 번호 |
| TEXT | VARCHAR(64) | 실행계획 문자열 조각 |

칼럼 정보

SID

실행계획이 속한 세션의 고유 번호를 나타낸다.

STMT_ID

statement 식별자를 나타낸다.

PIECE

한 문장에 대한 전체 실행계획 텍스트를 64바이트 길이로 나누어 저장한다. PIECE는 나뉘어진 64바이트 문자열의 일련 번호로 0부터 시작된다.

TEXT

실행계획 전체 텍스트의 일부분인 64바이트 텍스트 조각의 내용이다.

V\$PROCINFO

| Column name | Type | Description |
|--------------|------------|--------------------------------------|
| PROC_OID | BIGINT | 저장 프로시저의 객체 식별자 |
| MODIFY_COUNT | INTEGER | 저장 프로시저가 재 생성 또는 재 컴파일 된 횟수 |
| STATUS | VARCHAR(7) | 객체의 상태를 나타낸다. INVALID이면 실행 불가능 상태이다. |

| Column name | Type | Description |
|-------------|-------------|------------------------------------|
| SESSION_ID | INTEGER | 저장 프로시저의 STATUS를 변경한 세션의 ID를 나타낸다. |
| PROC_TYPE | VARCHAR(10) | 저장 프로시저의 타입을 나타낸다. |

칼럼 정보

PROC_OID

저장 프로시저 또는 저장 함수의 식별자로, SYS_PROCEDURES_ 메타 테이블의 한 PROC_OID 값과 동일하다.

MODIFY_COUNT

저장 프로시저 또는 함수가 재 생성 또는 재 컴파일 할 때마다 1씩 증가한다. 초기값은 0이다.

STATUS

저장 프로시저 또는 함수의 실행 가능 여부를 나타내는 값이다. VALID는 실행가능함을 나타낸다. SYS_PROCEDURES_ 메타 테이블의 STATUS 칼럼 설명을 참조한다.

SESSION_ID

저장 프로시저 또는 함수의 상태를 INVALID로 변경한 세션의 ID를 나타낸다. 상태가 변경된 적이 없으면 이 값이 0 또는 -1이다.

PROC_TYPE

저장 프로시저의 타입을 나타낸다. 가능한 값은 다음과 같다.

- NORMAL : 일반 프로시저
- EXTERNAL C : C/C++ External Procedure
- INTERNAL C : C/C++ Internal Procedure
- UNKNOWN : 서버를 구동할 때 저장 프로시저 컴파일에 실패하면 내부 프로시저 타입을 알 수 없어서 UNKNOWN으로 표시한다. 이후 컴파일이 되어 VALID 상태가 되면 정확한 타입이 설정된다.

V\$PROCTEXT

시스템에서 수행되는 저장 프로시저의 문자열 정보를 나타낸다.

| Column name | Type | Description |
|-------------|-------------|--------------------|
| PROC_OID | BIGINT | 저장 프로시저의 객체 식별자 |
| PIECE | INTEGER | 문자열 조각의 일련 번호 |
| TEXT | VARCHAR(64) | 저장 프로시저 구문의 문자열 조각 |

칼럼 정보

PROC_OID

저장 프로시저를 유일하게 가리키는 객체 식별자 즉 OID이다.

PIECE

저장 프로시저의 전체 구문을 64바이트 길이의 문자열로 나누어 저장한다. PIECE는 나뉘어진 64바이트 조각의 일련 번호로 0부터 시작된다.

TEXT

저장 프로시저 텍스트의 일부분인 64바이트 텍스트 조각의 내용을 나타낸다.

V\$PROPERTY

Altibase 내부에 설정된 프로퍼티의 정보를 보여준다.

| Column name | Type | Description |
|-------------|--------------|----------------|
| NAME | VARCHAR(256) | 프로퍼티의 이름 |
| STOREDCOUNT | INTEGER | 설정된 프로퍼티 값의 개수 |
| ATTR | BIGINT | 프로퍼티 속성 |
| MIN | VARCHAR(256) | 최솟값 |
| MAX | VARCHAR(256) | 최댓값 |
| VALUE1 | VARCHAR(256) | 설정된 첫 번째 값 |
| VALUE2 | VARCHAR(256) | 설정된 두 번째 값 |
| VALUE3 | VARCHAR(256) | 설정된 세 번째 값 |
| VALUE4 | VARCHAR(256) | 설정된 네 번째 값 |
| VALUE5 | VARCHAR(256) | 설정된 다섯 번째 값 |
| VALUE6 | VARCHAR(256) | 설정된 여섯 번째 값 |
| VALUE7 | VARCHAR(256) | 설정된 일곱 번째 값 |
| VALUE8 | VARCHAR(256) | 설정된 여덟 번째 값 |

칼럼 정보

NAME

해당 프로퍼티의 이름을 나타낸다.

STOREDCOUNT

해당 프로퍼티에 몇 개의 값이 설정되어 있는지 나타낸다. 8개까지 중복된 값을 가질 수 있다.

ATTR

해당 프로퍼티의 속성을 나타낸다.

MIN

해당 프로퍼티의 최솟값을 나타낸다.

MAX

해당 프로퍼티의 최댓값을 나타낸다.

VALUE1 ~ 8

실제 설정된 프로퍼티의 값을 나타낸다.

V\$QUEUE_DELETE_OFF

DELETE 문을 허용하지 않는 큐 테이블의 객체 식별자(OID) 정보를 가지고 있다. CREATE QUEUE 또는 ALTER QUEUE에서 DELETE OFF 절을 사용한 큐 테이블은 DELETE 문을 허용하지 않는다.

| Column name | Type | Description |
|-------------|--------|-------------|
| TABLE_OID | BIGINT | 테이블 객체 식별자 |

칼럼 정보**TABLE_OID**

테이블 객체 식별자로, SYS_TABLES_메타 테이블에서 하나의 TABLE_OID와 일대일로 대응된다.

V\$REPEXEC

이중화 관리자 정보를 보여준다.

| Column name | Type | Description |
|--------------------|---------|-------------|
| PORT | INTEGER | 사용중인 포트 번호 |
| MAX_SENDER_COUNT | INTEGER | 최대 송신자 개수 |
| MAX_RECEIVER_COUNT | INTEGER | 최대 수신자 개수 |

칼럼 정보**PORT**

지역서버의 이중화 관리자가 원격 서버의 이중화 요청을 받아들이는 포트번호 이다.

MAX_SENDER_COUNT

지역서버에서 생성 가능한 이중화 송신 쓰레드의 최대 개수이다.

MAX_RECEIVER_COUNT

지역서버에서 생성 가능한 이중화 수신 쓰레드의 최대 개수이다.

V\$REPGAP

이중화 송신자의 작업 로그 레코드와 가장 최근 생성된 로그 레코드간의 차이를 보여준다. 단, 이중화 송신 쓰레드가 동작 중일때만 정보를 보여준다.

| Column name | Type | Description |
|--------------|-------------|--|
| REP_NAME | VARCHAR(40) | 이중화 객체의 이름 |
| START_FLAG | BIGINT | 시작 옵션 |
| REP_LAST_SN | BIGINT | 마지막 로그 레코드의 식별 번호 |
| REP_SN | BIGINT | 현재 전송중인 로그 레코드의 식별 번호 |
| REP_GAP | BIGINT | 이중화 갭에 해당하는 로그파일의 실제 사이즈 (단위: 프로퍼티 REPLICATION GAP UNIT 에 설정된 단위) |
| REP_GAP_SIZE | BIGINT | 이중화 갭에 해당하는 로그파일의 실제 사이즈 (bytes) |
| READ_LFG_ID | INTEGER | 현재 읽고 있는 로그 파일 그룹(사용하지 않음, 0) |
| READ_FILE_NO | INTEGER | 현재 읽고 있는 로그 파일 번호 |
| READ_OFFSET | INTEGER | 현재 읽고 있는 위치 |

칼럼 정보**REP_NAME**

지역서버에 생성된 이중화 객체의 이름이다.

START_FLAG

지역서버의 이중화 구동시에 명시한 구동 옵션이다.

- NORMAL: 0
- QUICK: 1
- SYNC: 2
- SYNC_ONLY: 3
- SYNC RUN: 4
- SYNC END: 5
- RECOVERY from Replication: 6
- OFFLINE: 7
- PARALLEL: 8

REP_LAST_SN

지역서버의 트랜잭션에 의해 가장 최근에 로깅된 로그 레코드의 식별 번호이다.

REP_SN

지역서버의 이중화 송신 스레드가 현재 송신하고 있는 로그 레코드의 식별 번호이다.

REP_GAP

이중화 갭의 로그파일 사이즈를 프로퍼티 [REPLICATION_GAP_UNIT](#)에 설정된 단위로 보여준다. 프로퍼티 REPLICATION_GAP_UNIT을 통해 단위를 수정 할 수있으며, 기본값은 메가바이트이다. 즉, REP_GAP_SIZE의 값을 프로퍼티 REPLICATION_GAP_UNIT으로 나눈 값이며, 나머지가 생기면 올림한다.

REP_GAP_SIZE

이중화 갭의 로그파일 사이즈를 의미하며, 바이트 단위로 보여준다.

READ_FILE_NO

이중화 송신자가 현재 읽고 있는 로그 파일 번호이다. 하지만 이중화 송신자가 이중화 로그 버퍼에 있는 로그를 읽고 있을 때에는 갱신되지 않는다. 만약 읽고 있는 로그가 이중화 로그 버퍼에서 읽고 있는 것인지 확인하려면, V\$RELOGBUFFER의 READ_SN값이 BUFFER_MIN_SN과 BUFFER_MAX_SN 사이의 값인지 확인한다.

READ_OFFSET

로그 파일 내에서 현재 읽고 있는 위치를 나타낸다.

V\$REPGAP_PARALLEL

병렬 동작중인 이중화 송신 스레드의 작업 로그 레코드와 가장 최근 생성된 로그 레코드간의 차이를 보여준다. 단, 이 정보는 여러 이중화 송신 스레드가 병렬 동작 중일때만 보여준다.

| Column name | Type | Description |
|--------------|-------------|--|
| REP_NAME | VARCHAR(40) | 이중화 객체의 이름 |
| CURRENT_TYPE | VARCHAR(9) | 이중화 송신 스레드의 유형 |
| REP_LAST_SN | BIGINT | 마지막 로그 레코드의 식별 번호 |
| REP_SN | BIGINT | 현재 전송중인 로그 레코드의 식별 번호 |
| REP_GAP | BIGINT | 이중화 갭에 해당하는 로그파일의 실제 사이즈 (단위: 프로퍼티 REPLICATION_GAP_UNIT 에 설정된 단위) |
| REP_GAP_SIZE | BIGINT | 이중화 갭에 해당하는 로그파일의 실제 사이즈 (bytes) |
| READ_LFG_ID | INTEGER | 현재 읽고 있는 로그 파일 그룹(사용하지 않음, 0) |
| READ_FILE_NO | INTEGER | 현재 읽고 있는 로그 파일 번호 |
| READ_OFFSET | INTEGER | 현재 읽고 있는 위치 |
| PARALLEL_ID | INTEGER | 병렬 동작중인 다중 스레드를 구분하는 식별자 |

칼럼 정보

REP_NAME

지역서버에 생성된 이중화 객체의 이름이다.

CURRENT_TYPE

이는 이중화 송신 쓰레드의 현재 상태를 나타내는 것으로, 다음 중 한 값을 가질 수 있다.

- NORMAL: 이 값은 액티브 서버쪽의 송신 쓰레드가 트랜잭션 로그를 분석하여 XLog로 변환한 후, 대기 서버로 XLog를 전송하는 것을 의미한다.
- QUICK: 이중화를 QUICKSTART 옵션으로 시작하면 이 값이 보여질 수 있는데, 이는 전송 시작 위치가 변경중임을 나타내며, 송신 쓰레드는 예전 로그를 무시하고 가장 최근 로그부터 전송을 시작할 것이다. 시작 위치 변경 후에는, QUICK에서 NORMAL로 바뀔 것이다.
- SYNC: 이 값은 SYNC 옵션으로 이중화를 시작할 때 보여진다. 동기화가 완료된 후, NORMAL (LAZY 모드) 또는 PARALLEL (EAGER 모드)로 바뀌어 보여진다.
- SYNC_ONLY: 이 값은 SYNC ONLY 옵션으로 이중화를 시작할 때 보여진다. 동기화가 완료된 후, 송신 쓰레드는 종료될 것이다.
- RECOVERY: 이 값은 송신 쓰레드가 다른 서버에서 손상된 데이터를 복원하기 위해 실행중임을 나타낸다.
- OFFLINE: 이 값은 액티브 서버가 오프라인이고 대기 서버에 로그를 적용할 때, 송신 쓰레드가 액티브 서버의 로그를 읽기 위해 실행중임을 나타낸다.
- PARALLEL: 이 값은 이중화 대상 테이블과 관련된 XLog를 여러 송신 쓰레드가 병렬로 송신중임을 나타낸다. 이 값은 PARALLEL 옵션과 함께 EAGER 모드로 이중화를 시작할 때 보여질 수 있다. SYNC 또는 SYNC ONLY 옵션과 함께 이중화를 시작할 때 지정할 수 있는 PARALLEL 옵션과는 다르다.

REP_LAST_SN

지역서버의 트랜잭션에 의해 가장 최근에 로깅된 로그 레코드의 식별 번호이다.

REP_SN

지역서버의 이중화 송신 쓰레드가 현재 송신 중인 로그 레코드의 식별 번호이다.

REP_GAP

이중화 갭의 로그파일 사이즈를 프로퍼티 [REPLICATION_GAP_UNIT](#)에 설정된 단위로 보여준다. 프로퍼티 REPLICATION_GAP_UNIT을 통해 단위를 수정 할 수있으며, 기본값은 메가바이트이다. 즉, REP_GAP_SIZE의 값을 프로퍼티 REPLICATION_GAP_UNIT으로 나눈 값이며, 나머지가 생기면 올림한다.

REP_GAP_SIZE

이중화 갭의 로그파일 사이즈를 의미하며, 바이트 단위로 보여준다.

READ_FILE_NO

현재 읽고 있는 로그 파일 번호이다.

READ_OFFSET

로그 파일 내에서 현재 읽고 있는 위치를 나타낸다.

PARALLEL_ID

한 송신자를 위해 병렬 동작중인 여러 스레드 중 하나의 식별자이다.

V\$REPOLOGBUFFER

이중화 송신 스레드가 동작 중일 때 이중화 송신자 전용 로그 버퍼의 상태 정보를 보여준다.

| Column name | Type | Description |
|---------------|-------------|------------------------------------|
| REP_NAME | VARCHAR(40) | 이중화 객체의 이름 |
| BUFFER_MIN_SN | BIGINT | 전용 로그 버퍼의 최소 로그 식별 번호 |
| READ_SN | BIGINT | 이중화 송신 스레드가 다음 읽어야 할 로그 레코드의 식별 번호 |
| BUFFER_MAX_SN | BIGINT | 전용 로그 버퍼의 최대 로그 식별 번호 |

칼럼 정보**REP_NAME**

지역서버에 생성된 이중화 객체의 이름이다.

BUFFER_MIN_SN

이중화 전용 로그 버퍼에 저장된 로그 레코드의 식별 번호중 최솟값이다.

READ_SN

이중화 전용 로그 버퍼 내에서 이중화 송신 스레드가 다음에 읽어야 할 로그 레코드의 식별 번호이다.

BUFFRT_MAX_SN

이중화 전용 로그 버퍼에 저장된 로그 레코드의 식별 번호중 최댓값이다.

V\$REPOFFLINE_STATUS

오프라인 이중화의 수행 상태를 표시한다.

| Column name | Type | Description |
|--------------|-------------|----------------------------|
| REP_NAME | VARCHAR(40) | 이중화 객체의 이름 |
| STATUS | BIGINT | 오프라인 이중화의 수행 상태 |
| SUCCESS_TIME | INTEGER | 오프라인 이중화가 성공적으로 수행된 시점의 시간 |

칼럼 정보

REP_NAME

지역 서버에 생성된 이중화 객체의 이름이다.

STATUS

오프라인 이중화의 수행 상태

- 0: 시작되지 않았음
- 1: 시작됨
- 2: 종료
- 3: 실패

SUCCESS_TIME

가장 최근에 오프라인 이중화가 성공적으로 수행된 시점의 시각을 시스템 시간으로 표시한다. 이중화가 성공적으로 시작되어 종료되었을 경우 종료된 시각이 설정되고, 그 외는 0으로 설정된다.

V\$REPRECEIVER

이중화 수신자의 정보를 보여준다.

| Column name | Type | Description |
|----------------------|-------------|--|
| REP_NAME | VARCHAR(40) | 이중화 객체의 이름 |
| MY_IP | VARCHAR(64) | 지역서버의 IP 주소 |
| MY_PORT | INTEGER | 지역서버의 이중화 포트 번호 |
| PEER_IP | VARCHAR(64) | 원격서버의 IP 주소 |
| PEER_PORT | INTEGER | 원격서버의 이중화 포트 번호 |
| APPLY_XSN | BIGINT | 처리중인 XSN |
| INSERT_SUCCESS_COUNT | BIGINT | 지역 서버에서 수신 스레드가 적용에 성공한 INSERT 로그 레코드의 수 |
| INSERT_FAILURE_COUNT | BIGINT | 지역 서버에서 수신 스레드가 적용에 실패한 INSERT 로그 레코드의 수 |
| UPDATE_SUCCESS_COUNT | BIGINT | 지역 서버에서 수신 스레드가 적용에 성공한 UPDATE 로그 레코드의 수 |
| UPDATE_FAILURE_COUNT | BIGINT | 지역 서버에서 수신 스레드가 적용에 실패한 UPDATE 로그 레코드의 수 |
| DELETE_SUCCESS_COUNT | BIGINT | 지역 서버에서 수신 스레드가 적용에 성공한 DELETE 로그 레코드의 수 |
| DELETE_FAILURE_COUNT | BIGINT | 지역 서버에서 수신 스레드가 적용에 실패한 DELETE 로그 레코드의 수 |
| PARALLEL_ID | INTEGER | 항상 0이 표시됨 |

| Column name | Type | Description |
|---------------------------|---------|---------------------------------|
| SQL_APPLY_TABLE_COUNT | INTEGER | SQL 반영 모드로 동작하는 테이블 개수 |
| APPLIER_INIT_BUFFER_USAGE | BIGINT | 병렬 적용자에 대기중인 큐의 현재 크기 (단위 byte) |

칼럼 정보

REP_NAME

지역서버에 생성된 이중화 객체의 이름이다.

MY_IP

지역서버의 IP 주소값이다.

MY_PORT

지역서버의 수신 쓰레드가 사용하는 포트번호이다.

PEER_IP

원격서버의 IP 주소값이다.

PEER_PORT

원격서버의 송신 쓰레드가 사용하는 포트번호이다.

APPLY_XSN

원격서버에서 송신 쓰레드가 전송하여 지역서버에서 수신 쓰레드가 적용 중인 XLog의 SN을 나타낸다.

INSERT_SUCCESS_COUNT

지역서버에서 수신 쓰레드가 적용에 성공한 INSERT 로그레코드의 수를 나타낸다.

COMMIT 또는 ROLLBACK과 무관하게 계산된다. 즉 ROLLBACK을 수행해도 COUNT가 줄어들지 않는다.

INSERT_FAILURE_COUNT

지역서버에서 수신 쓰레드가 적용에 실패한 INSERT 로그레코드의 수를 나타낸다. (Conflict를 포함)

COMMIT 또는 ROLLBACK과 무관하게 계산된다. 즉 ROLLBACK을 수행해도 COUNT가 줄어들지 않는다.

UPDATE_SUCCESS_COUNT

지역서버에서 수신 쓰레드가 적용에 성공한 UPDATE 로그레코드의 수를 나타낸다.

COMMIT 또는 ROLLBACK과 무관하게 계산된다. 즉 ROLLBACK을 수행해도 COUNT가 줄어들지 않는다.

UPDATE_FAILURE_COUNT

지역서버에서 수신 쓰레드가 적용에 실패한 UPDATE 로그레코드의 수를 나타낸다. (Conflict를 포함)

COMMIT 또는 ROLLBACK과 무관하게 계산된다. 즉 ROLLBACK을 수행해도 COUNT가 줄어들지 않는다.

DELETE_SUCCESS_COUNT

지역서버에서 수신 쓰레드가 적용에 성공한 DELETE 로그레코드의 수를 나타낸다.

COMMIT 또는 ROLLBACK과 무관하게 계산된다. 즉 ROLLBACK을 수행해도 COUNT가 줄어들지 않는다.

DELETE_FAILURE_COUNT

지역서버에서 수신 쓰레드가 적용에 실패한 DELETE 로그레코드의 수를 나타낸다. (Conflict를 포함)

COMMIT 또는 ROLLBACK과 무관하게 계산된다. 즉 ROLLBACK을 수행해도 COUNT가 줄어들지 않는다.

PARALLEL_ID

항상 0이 표시된다.

Eager 모드에서는 V\$REPRECEIVER_PARALLEL의 PARALLEL_ID가 0인 이중화 수신자와 동일한 수신자이며, eager 모드가 아닌 경우 의미 없는 값이다.

SQL_APPLY_TABLE_COUNT

SQL 반영 모드로 동작하는 테이블의 개수이다.

APPLIER_INIT_BUFFER_USAGE

병렬 적용자(Parallel Applier) 옵션으로 이중화를 사용할 때, 적용자 쓰레드에 할당된 XLog의 메모리 총 사용량을 나타낸다. 단위는 byte이다.

V\$REPRECEIVER_COLUMN

이중화 수신자의 이중화 대상 칼럼 정보를 보여준다.

| Column name | Type | Description |
|----------------|--------------|------------------------|
| REP_NAME | VARCHAR(40) | 이중화 이름 |
| USER_NAME | VARCHAR(128) | 사용자 이름 |
| TABLE_NAME | VARCHAR(128) | 테이블 이름 |
| PARTITION_NAME | VARCHAR(128) | 파티션 이름 |
| COLUMN_NAME | VARCHAR(128) | 칼럼 이름 |
| APPLY_MODE | INTEGER | 0: Binary 모드 1: SQL 모드 |

칼럼 정보**REP_NAME**

지역 서버에 생성된 이중화 객체의 이름이다.

USER_NAME

지역 서버의 이중화 대상 테이블 소유자의 사용자 이름이다. SYS_USERS_ 메타 테이블의 한 USER_NAME 값과 일치한다.

TABLE_NAME

지역 서버의 이중화 대상 테이블의 이름으로 SYS_TABLES_ 메타 테이블의 한 TABLE_NAME 값과 일치한다.

PARTITION_NAME

지역 서버의 이중화 대상 파티션 이름이다.

COLUMN_NAME

지역 서버의 이중화 대상 칼럼 이름이다.

APPLY_MODE

테이블에 데이터를 반영하는 모드이다.

- 0: Binary 모드
- 1: SQL 모드

V\$REPRECEIVER_PARALLEL

병렬 동작중인 이중화 수신 쓰레드의 정보를 보여준다.

| Column name | Type | Description |
|----------------------|-------------|---|
| REP_NAME | VARCHAR(40) | 이중화 객체의 이름 |
| MY_IP | VARCHAR(64) | 지역서버의 IP 주소 |
| MY_PORT | INTEGER | 지역서버의 이중화 포트 번호 |
| PEER_IP | VARCHAR(64) | 원격서버의 IP 주소 |
| PEER_PORT | INTEGER | 원격서버의 이중화 포트 번호 |
| APPLY_XSN | BIGINT | 현재 처리중인 XSN |
| INSERT_SUCCESS_COUNT | BIGINT | 지역 서버에서 수신 쓰레드가 적용에 성공한 INSERT 로그레코드의 수 |
| INSERT_FAILURE_COUNT | BIGINT | 지역 서버에서 수신 쓰레드가 적용에 실패한 INSERT 로그레코드의 수 |
| UPDATE_SUCCESS_COUNT | BIGINT | 지역 서버에서 수신 쓰레드가 적용에 성공한 UPDATE 로그레코드의 수 |
| UPDATE_FAILURE_COUNT | BIGINT | 지역 서버에서 수신 쓰레드가 적용에 실패한 UPDATE 로그레코드의 수 |
| DELETE_SUCCESS_COUNT | BIGINT | 지역 서버에서 수신 쓰레드가 적용에 성공한 DELETE 로그레코드의 수 |
| DELETE_FAILURE_COUNT | BIGINT | 지역 서버에서 수신 쓰레드가 적용에 실패한 DELETE 로그레코드의 수 |
| PARALLEL_ID | INTEGER | 병렬 동작중인 여러 이중화 수신 쓰레드 중 하나의 식별자 |

칼럼 정보

REP_NAME

이중화 객체의 이름이다.

MY_IP

지역서버의 IP 주소값이다.

MY_PORT

지역서버의 수신 쓰레드가 사용하는 포트번호이다.

PEER_IP

원격서버의 IP 주소값이다.

PEER_PORT

원격서버의 송신 쓰레드가 사용하는 포트번호이다.

APPLY_XSN

원격서버에서 송신 쓰레드가 전송하여 지역서버에서 수신 쓰레드가 적용 중인 XLog의 SN을 나타낸다.

INSERT_SUCCESS_COUNT

지역서버에서 수신 쓰레드가 적용에 성공한 INSERT 로그레코드의 수를 나타낸다.

COMMIT 또는 ROLLBACK과 무관하게 계산된다. 즉 ROLLBACK을 수행해도 COUNT가 줄어들지 않는다.

INSERT_FAILURE_COUNT

지역서버에서 수신 쓰레드가 적용에 실패한 INSERT 로그레코드의 수를 나타낸다. (Conflict를 포함)

COMMIT 또는 ROLLBACK과 무관하게 계산된다. 즉 ROLLBACK을 수행해도 COUNT가 줄어들지 않는다.

UPDATE_SUCCESS_COUNT

지역서버에서 수신 쓰레드가 적용에 성공한 UPDATE 로그레코드의 수를 나타낸다.

COMMIT 또는 ROLLBACK과 무관하게 계산된다. 즉 ROLLBACK을 수행해도 COUNT가 줄어들지 않는다.

UPDATE_FAILURE_COUNT

지역서버에서 수신 쓰레드가 적용에 실패한 UPDATE 로그레코드의 수를 나타낸다. (Conflict를 포함)

COMMIT 또는 ROLLBACK과 무관하게 계산된다. 즉 ROLLBACK을 수행해도 COUNT가 줄어들지 않는다.

DELETE_SUCCESS_COUNT

지역서버에서 수신 쓰레드가 적용에 성공한 DELETE 로그레코드의 수를 나타낸다.

COMMIT 또는 ROLLBACK과 무관하게 계산된다. 즉 ROLLBACK을 수행해도 COUNT가 줄어들지 않는다.

DELETE_FAILURE_COUNT

지역서버에서 수신 쓰레드가 적용에 실패한 DELETE 로그레코드의 수를 나타낸다. (Conflict를 포함)

COMMIT 또는 ROLLBACK과 무관하게 계산된다. 즉 ROLLBACK을 수행해도 COUNT가 줄어들지 않는다.

PARALLEL_ID

동일 이중화 객체에 해당하는 여러 이중화 수신자 중 하나의 식별자이다.

V\$REPRECEIVER_PARALLEL_APPLY

이중화 수신자의 정보를 보여준다.

| Column name | Type | Description |
|------------------------|-------------|---|
| REP_NAME | VARCHAR(40) | 이중화 객체의 이름 |
| PARALLEL_APPLIER_INDEX | INTEGER | 적용자 번호 |
| APPLY_XSN | BIGINT | 처리중인 XSN |
| INSERT_SUCCESS_COUNT | BIGINT | 지역 서버에서 수신 스레드가 적용에 성공한 INSERT 로그레코드의 수 |
| INSERT_FAILURE_COUNT | BIGINT | 지역 서버에서 수신 스레드가 적용에 실패한 INSERT 로그레코드의 수 |
| UPDATE_SUCCESS_COUNT | BIGINT | 지역 서버에서 수신 스레드가 적용에 성공한 UPDATE 로그레코드의 수 |
| UPDATE_FAILURE_COUNT | BIGINT | 지역 서버에서 수신 스레드가 적용에 실패한 UPDATE 로그레코드의 수 |
| DELETE_SUCCESS_COUNT | BIGINT | 지역 서버에서 수신 스레드가 적용에 성공한 DELETE 로그레코드의 수 |
| DELETE_FAILURE_COUNT | BIGINT | 지역 서버에서 수신 스레드가 적용에 실패한 DELETE 로그레코드의 수 |
| STATUS | VARCHAR(10) | RECEIVER APPLIER 의 현재 동작 상태 |

칼럼 정보**STATUS**

RECEIVER APPLIER 의 현재 동작 상태를 나타낸다

- INITIALIZE : 초기화 중
- WORKING : 데이터 반영 중
- DEQUEUEING : XLog 를 receiver 로 부터 전달 받기를 대기 중
- WAITING : 다른 Applier 들이 transaction 반영을 대기 중
- STOP : 종료

나머지 칼럼 정보에 대한 자세한 내용은 V\$REPRECEIVER를 참조한다.

V\$REPRECEIVER_STATISTICS

이중화 수신 쓰레드의 작업 별 수행시간에 대해 통계 정보를 보여준다. TIMED_STATISTICS 프로퍼티의 값이 1로 설정되어 있을 때만 통계정보가 이 뷰에 수집된다. 통계치 측정 간격과 측정 방식은 TIMER_THREAD_RESOLUTION과 TIMER_RUNNING_LEVEL 프로퍼티 값을 조정하여 정할 수 있다.

| Column name | Type | Description |
|---------------------|-------------|-------------------------------------|
| REP_NAME | VARCHAR(40) | 이중화 객체의 이름 |
| PARALLEL_ID | INTEGER | 병렬 동작중인 이중화 수신 쓰레드들 중 하나의 식별자 |
| RECV_XLOG | BIGINT | XLog 수신에 소요된 전체 시간 |
| CONVERT_ENDIAN | BIGINT | Endian(Byte Order) 변환 작업에 소요된 전체 시간 |
| BEGIN_TRANSACTION | BIGINT | 트랜잭션 시작에 걸린 전체 시간 |
| COMMIT_TRANSACTION | BIGINT | 트랜잭션 커밋에 걸린 전체 시간 |
| ABORT_TRANSACTION | BIGINT | 트랜잭션 롤백에 걸린 전체 시간 |
| OPEN_TABLE_CURSOR | BIGINT | 테이블 커서를 여는데 걸린 전체 시간 |
| CLOSE_TABLE_CURSOR | BIGINT | 테이블 커서를 닫는데 걸린 전체 시간 |
| INSERT_ROW | BIGINT | INSERT의 로그를 재연하는데 소요된 전체 시간 |
| UPDATE_ROW | BIGINT | UPDATE의 로그를 재연하는데 소요된 전체 시간 |
| DELETE_ROW | BIGINT | DELETE의 로그를 재연하는데 소요된 전체 시간 |
| OPEN_LOB_CURSOR | BIGINT | OPEN LOB CURSOR 작업에 걸린 전체 시간 |
| PREPARE_LOB_WRITING | BIGINT | PREPARE LOB CURSOR 작업에 걸린 전체 시간 |
| WRITE_LOB_PIECE | BIGINT | WRITE LOB PIECE 작업에 걸린 전체 시간 |
| FINISH_LOB_WRITE | BIGINT | FINISH LOB WRITE 작업에 걸린 전체 시간 |
| CLOSE_LOB_CURSOR | BIGINT | CLOSE LOB CURSOR 작업에 걸린 전체 시간 |
| COMPARE_IMAGE | BIGINT | 충돌 해결을 위한 데이터 비교 작업에 소요된 전체 시간 |
| SEND_ACK | BIGINT | ACK 송신에 걸린 전체 시간 |

칼럼 정보

REP_NAME

지역서버에 생성된 이중화 객체의 이름이다.

PARALLEL_ID

수신자가 가지는 고유한 ID로 해당 수신 쓰레드가 속한 이중화 내에서 유일한 값을 가진다. 이 ID는 EAGER모드에서 병렬 수신자로 동작할 때, 각 쓰레드를 구별하기 위해 각 수신 쓰레드에 주어진다.

RECV_XLOG

XLog를 수신하는데 걸린 시간의 누적 값이다. 이 값은 새로운 XLog가 오기를 기다리는 시간도 포함한다.

CONVERT_ENDIAN

ENDIAN (byte order) 변환 작업에 소요된 시간의 누적 값이다. 송신 서버와 수신 서버 장비간의 ENDIAN (byte order) 이 다를 때 변환작업이 발생한다.

BEGIN_TRANSACTION

트랜잭션 BEGIN작업에 소요된 시간의 누적 값이다.

COMMIT_TRANSACTION

트랜잭션 COMMIT작업에 소요된 시간의 누적 값이다.

ABORT_TRANSACTION

트랜잭션 ROLL BACK작업에 소요된 시간의 누적 값이다.

OPEN_TABLE_CURSOR

테이블 커서 열기 작업에 소요된 시간의 누적 값이다.

CLOSE_TABLE_CURSOR

테이블 커서 닫기 작업에 소요된 시간의 누적 값이다.

INSERT_ROW

수신 쓰레드가 INSERT 문의 로그를 반영하는데 소요된 시간의 누적 값이다.

UPDATE_ROW

수신 쓰레드가 UPDATE 문의 로그를 반영하는데 소요된 시간의 누적 값이다.

DELETE_ROW

수신 쓰레드가 DELETE 문의 로그를 반영하는데 소요된 시간의 누적 값이다.

OPEN_LOB_CURSOR

LOB 연산 작업 중 OPEN LOB CURSOR작업 시간의 누적 값이다.

PREPARE_LOB_WRITING

LOB 연산 작업 중 PREPARE LOB WRITING작업 시간의 누적 값이다.

WRITE_LOB_PIECE

LOB 연산 작업 중 WRITE LOB PIECE작업 시간의 누적 값이다.

FINISH_LOB_WRITE

LOB 연산 작업 중 FINISH LOB WRITE작업 시간의 누적 값이다.

CLOSE_LOB_CURSOR

LOB 연산 작업 중 FINISH CLOSE LOB CURSOR작업 시간의 누적 값이다.

COMPARE_IMAGE

데이터 충돌을 검사하기 위해서, 양 쪽 서버의 데이터를 비교하는 작업 시간의 누적 값이다.

SEND_ACK

Sender에게 ACK을 보내는 데 걸린 시간의 누적 값이다.

V\$REPRECEIVER_TRANSTBL

이중화 수신자의 트랜잭션 테이블의 정보를 보여준다.

| Column name | Type | Description |
|------------------------|-------------|---|
| REP_NAME | VARCHAR(40) | 이중화 객체의 이름 |
| LOCAL_TID | BIGINT | 지역 트랜잭션 식별자 |
| REMOTE_TID | BIGINT | 원격 트랜잭션 식별자 |
| BEGIN_FLAG | INTEGER | 현재 사용하지 않음 |
| BEGIN_SN | BIGINT | 트랜잭션의 최초 로그 레코드 SN |
| PARALLEL_ID | INTEGER | 동일 이중화 객체에서 병렬 동작중인 여러 이중화 수신 쓰레드 중 하나의 식별자 |
| PARALLEL_APPLIER_INDEX | INTEGER | 트랜잭션을 수행하는 적용자의 번호 |

칼럼 정보**REP_NAME**

지역서버에 생성된 이중화 객체의 이름이다.

LOCAL_TID

지역서버에서 실행중인 트랜잭션의 식별자이다.

REMOTE_TID

원격서버에서 실행중인 트랜잭션의 식별자이다. 이미 실행이 끝났을 수도 있다.

V\$REPRECEIVER_TRANSTBL_PARALLEL

병렬 동작중인 다중 이중화 수신 쓰레드들의 트랜잭션 테이블 정보를 보여준다.

| Column name | Type | Description |
|-------------|-------------|-------------|
| REP_NAME | VARCHAR(40) | 이중화 객체의 이름 |

| Column name | Type | Description |
|-------------|---------|------------------------------------|
| LOCAL_TID | INTEGER | 지역 트랜잭션 식별자 |
| REMOTE_TID | INTEGER | 원격 트랜잭션 식별자 |
| BEGIN_FLAG | INTEGER | 현재 사용하지 않음 |
| BEGIN_SN | BIGINT | 트랜잭션의 최초 로그 레코드 SN |
| PARALLEL_ID | INTEGER | 같은 이중화 이름을 갖는 여러 수신 쓰레드들 중 하나의 식별자 |

칼럼 정보

REP_NAME

지역서버에 생성된 이중화 객체의 이름이다.

LOCAL_TID

지역서버에서 실행중인 트랜잭션의 식별자이다.

REMOTE_TID

원격서버에서 실행중인 트랜잭션의 식별자이다. 이미 실행이 끝났을 수도 있다.

PARALLEL_ID

병렬 동작중인 여러 이중화 수신 쓰레드들 중 하나의 식별자이다.

V\$REPRECOVERY

이중화를 이용한 복구 정보를 보여준다.

| Column name | Type | Description |
|----------------------|-------------|---|
| REP_NAME | VARCHAR(40) | 이중화 객체 이름 |
| STATUS | INTEGER | 복구에 대한 현재 상태 1: 복구 정보 생성 중 2: 복구 요청 대기 중 3: 복구 진행 중 |
| START_XSN | BIGINT | 복구를 위한 전송시작 SN |
| XSN | BIGINT | 복구를 위해 현재 전송중인 로그 SN |
| END_XSN | BIGINT | 복구를 위한 마지막 전송 SN |
| RECOVERY_SENDER_IP | VARCHAR(64) | 지역 서버의 복구를 위한 송신자 IP 주소 |
| PEER_IP | VARCHAR(64) | 원격 서버의 복구를 위한 수신자 IP 주소 |
| RECOVERY_SENDER_PORT | INTEGER | 지역서버의 복구를 위한 송신자 포트 번호 |
| PEER_PORT | INTEGER | 원격서버의 복구를 위한 수신자 포트 번호 |

칼럼 정보

REP_NAME

지역 서버에 생성된 이중화 객체의 이름이다.

STATUS

지역서버의 이중화 송신 쓰레드의 현재 상태를 나타낸다.

- 1: 복구 정보 생성 중
- 2: 복구 요청 대기 중
- 3: 복구 진행 중

START_XSN

지역 서버의 복구를 위해 송신 쓰레드가 전송할 시작 로그 레코드의 SN을 나타낸다.

XSN

지역서버의 복구를 위해 이중화 송신 쓰레드가 현재 송신중인 로그 레코드의 SN을 나타낸다.

END_XSN

지역 서버의 복구를 위해 송신 쓰레드가 전송할 마지막 로그 레코드의 SN을 나타낸다.

RECOVERY_SENDER_IP

지역 서버의 복구를 위한 송신자 IP 주소이다.

PEER_IP

원격 서버의 복구를 위한 IP 주소이다.

RECOVERY_SENDER_PORT

지역 서버의 복구를 위한 송신 쓰레드가 사용하는 포트번호이다.

PEER_PORT

원격 서버의 복구를 위한 수신 쓰레드가 사용하는 포트번호이다.

V\$REPSENDER

이중화 송신자의 정보를 보여준다.

| Column name | Type | Description |
|----------------|-------------|--------------------|
| REP_NAME | VARCHAR(40) | 이중화 객체의 이름 |
| START_FLAG | BIGINT | 시작 옵션 |
| NET_ERROR_FLAG | BIGINT | 에러 상태 플래그 |
| XSN | BIGINT | 현재 송신중인 로그 레코드의 SN |
| COMMIT_XSN | BIGINT | Commit 로그 레코드의 SN |
| STATUS | BIGINT | 현재 상태 |

| Column name | Type | Description |
|----------------|-------------|-----------------|
| SENDER_IP | VARCHAR(64) | 송신자 IP 주소 |
| PEER_IP | VARCHAR(64) | 원격 서버의 IP 주소 |
| SENDER_PORT | INTEGER | 송신 포트 번호 |
| PEER_PORT | INTEGER | 원격 서버의 포트 번호 |
| READ_LOG_COUNT | BIGINT | 읽은 로그의 개수 |
| SEND_LOG_COUNT | BIGINT | 읽어서 송신한 로그의 수 |
| REPL_MODE | VARCHAR(7) | 사용자가 지정한 이중화 모드 |
| ACT_REPL_MODE | VARCHAR(7) | 실제 이중화 모드 |

칼럼 정보

REP_NAME

지역서버에 생성된 이중화 객체의 이름이다.

START_FLAG

지역서버의 이중화 구동시에 명시한 구동 옵션이다. 다음 값들을 가질 수 있다.

- : 0
- QUICK: 1
- SYNC: 2
- SYNC_ONLY: 3
- SYNC RUN : 4
- SYNC END : 5
- RECOVERY from Replication : 6
- OFFLINE: 7
- PARALLEL: 8

NET_ERROR_FLAG

네트워크 오류 발생 여부를 나타낸다. 디폴트는 0이며, 1은 오류가 발생했음을 나타낸다.

XSN

지역서버의 이중화 송신 스레드가 송신중인 로그 레코드의 SN을 나타낸다.

COMMIT_XSN

지역서버에서 가장 최근에 COMMIT한 트랜잭션이 로깅한 COMMIT 로그 레코드의 SN을 나타낸다.

STATUS

지역서버의 이중화 송신 스레드의 현재 상태를 나타낸다.

- 0: STOP
- 1: RUN

- 2: RETRY
- 3: FAILBACK NORMAL
- 4: FAILBACK MASTER
- 5: FAILBACK SLAVE
- 6: SYNC
- 7: FAILBACK EAGER
- 8: FAILBACK FLUSH
- 9: IDLE

SENDER_IP

지역서버의 IP 주소이다.

PEER_IP

원격서버의 IP 주소이다.

SENDER_PORT

지역서버의 이중화 송신 스레드가 사용하는 포트번호이다.

PEER_PORT

원격서버의 이중화 수신 스레드가 사용하는 포트번호이다.

READ_LOG_COUNT

지역서버에서 송신 스레드가 읽은 로그 레코드의 수를 나타낸다.

SEND_LOG_COUNT

지역서버에서 송신 스레드가 읽어서 송신한 로그레코드의 수를 나타낸다.

REPL_MODE

사용자에 의해서 설정된 이중화 모드를 나타낸다. 이중화 모드의 종류는 LAZY 또는 EAGER이다. 이중화 모드에 대한 자세한 설명은 *Replication Manual*을 참조하기 바란다.

ACT_REPL_MODE

실제로 동작 중인 이중화 모드를 나타내며, REPL_MODE와 다를 수도 있다.

이중화 모드를 EAGER 로 설정했을 때, 장애 등으로 인하여 이중화 갭이 있는 경우, 이중화는 LAZY 모드로 동작하게 된다.

이 외의 경우에는 REPL_MODE의 값과 동일하다.

V\$REPSENDER_PARALLEL

병렬 동작중인 이중화 송신 스레드들의 정보를 보여준다.

| Column name | Type | Description |
|--------------|-------------|-------------|
| REP_NAME | VARCHAR(40) | 이중화 객체의 이름 |
| CURRENT_TYPE | VARCHAR(9) | 시작 옵션 |

| Column name | Type | Description |
|----------------|-------------|---|
| NET_ERROR_FLAG | BIGINT | 에러 상태 플래그 |
| XSN | BIGINT | 전송중인 로그 레코드의 SN |
| COMMIT_XSN | BIGINT | Commit 로그 레코드의 SN |
| STATUS | VARCHAR(15) | 현재 상태 |
| SENDER_IP | VARCHAR(64) | 송신자 IP 주소 |
| PEER_IP | VARCHAR(64) | 원격 서버의 IP 주소 |
| SENDER_PORT | INTEGER | 송신 포트 번호 |
| PEER_PORT | INTEGER | 원격 서버의 포트 번호 |
| READ_LOG_COUNT | BIGINT | 읽은 로그의 개수 |
| SEND_LOG_COUNT | BIGINT | 읽어서 송신한 로그의 수 |
| REPL_MODE | VARCHAR(7) | 사용자가 지정한 이중화 모드 |
| PARALLEL_ID | INTEGER | 같은 이중화 이름을 가지는 여러 이중화 송신 쓰레드들 중 하나의 식별자 |

칼럼 정보

REP_NAME

지역서버에 생성된 이중화 객체의 이름이다.

CURRENT_TYPE

V\$REPGAP_PARALLEL 성능 뷰의 CURRENT_TYPE 칼럼 설명을 참조하기 바란다.

NET_ERROR_FLAG

네트워크 오류 발생 여부를 나타낸다. 디폴트는 0이며, 1은 오류가 발생했음을 나타낸다.

XSN

지역서버의 이중화 송신 쓰레드가 송신중인 로그 레코드의 SN을 나타낸다.

COMMIT_XSN

지역서버에서 가장 최근에 COMMIT한 트랜잭션이 로깅한 COMMIT 로그 레코드의 SN을 나타낸다.

STATUS

지역서버의 이중화 송신 쓰레드의 현재 상태를 나타낸다.

- 0: STOP
- 1: RUN
- 2: RETRY
- 3: FAILBACK NORMAL
- 4: FAILBACK MASTER

- 5: FAILBACK SLAVE
- 6: SYNC
- 7: FAILBACK EAGER
- 8: FAILBACK FLUSH
- 9: IDLE

SENDER_IP

지역서버의 IP 주소이다.

PEER_IP

원격서버의 IP 주소이다.

SENDER_PORT

지역서버의 이중화 송신 스레드가 사용하는 포트번호이다.

PEER_PORT

원격서버의 이중화 수신 스레드가 사용하는 포트번호이다.

READ_LOG_COUNT

지역서버에서 송신 스레드가 읽은 로그레코드의 수를 나타낸다.

SEND_LOG_COUNT

지역서버에서 송신 스레드가 읽어서 송신한 로그레코드의 수를 나타낸다.

REPL_MODE

사용자에 의해서 설정된 이중화 모드를 나타낸다. 이중화 모드의 종류는 LAZY 또는 EAGER이다. 이중화 모드에 대한 자세한 설명은 *Replication Manual*을 참조하기 바란다.

PARALLEL_ID

병렬 동작중인 여러 이중화 송신 스레드들 중 하나의 식별자이다.

V\$REPSENDER_SENT_LOG_COUNT

이중화 송신자가 전송한 로그를 DML 타입 별로 분류하여 개수를 보여준다. 이중화 로그는 이중화 송신자가 시작하면 실시간으로 전송되며, 전송할 때마다 이 성능 뷰의 데이터가 갱신된다.

Eager 모드의 병렬 이중화의 경우, Parent Sender에 대한 정보만 이 성능 뷰에 보여주며, 각 Sender 스레드에 대한 정보는 V\$REPSENDER_SENT_LOG_COUNT_PARALLEL 성능 뷰에 보여준다.

| Column name | Type | Description |
|------------------|-------------|-------------------|
| REP_NAME | VARCHAR(40) | 이중화 객체의 이름 |
| CURRENT_TYPE | VARCHAR(9) | 이중화 송신 스레드의 유형 |
| TABLE_OID | BIGINT | 테이블 객체 식별자 |
| INSERT_LOG_COUNT | INTEGER | 전송한 INSERT 로그의 개수 |
| DELETE_LOG_COUNT | INTEGER | 전송한 DELETE 로그의 개수 |

| Column name | Type | Description |
|------------------|---------|-------------------|
| UPDATE_LOG_COUNT | INTEGER | 전송한 UPDATE 로그의 개수 |
| LOB_LOG_COUNT | INTEGER | 전송한 LOB 관련 로그의 개수 |

칼럼 정보

REP_NAME

지역서버에 생성된 이중화 객체의 이름이다.

CURRENT_TYPE

V\$REPGAP_PARALLEL 성능 뷰의 CURRENT_TYPE 칼럼 설명을 참조하기 바란다.

V\$REPSENDER_SENT_LOG_COUNT_PARALLEL

Eager 모드의 병렬 이중화의 각 이중화 송신 쓰레드가 전송한 로그를 DML 타입 별로 분류하여 개수를 보여준다. 이중화 로그는 이중화 송신자가 시작하면 실시간으로 전송되며, 전송할 때마다 이 성능 뷰의 데이터가 갱신된다.

Eager 모드의 병렬 이중화의 경우, Parent Sender에 대한 정보는 V\$REPSENDER_SENT_LOG_COUNT 성능 뷰에 보여주며, 각 Sender 쓰레드에 대해서만 이 성능 뷰에 보여준다.

| Column name | Type | Description |
|------------------|-------------|--------------------------|
| REP_NAME | VARCHAR(40) | 이중화 객체의 이름 |
| CURRENT_TYPE | VARCHAR(9) | 이중화 송신 쓰레드의 유형 |
| PARALLEL_ID | INTEGER | 병렬 동작중인 다중 쓰레드를 구분하는 식별자 |
| TABLE_OID | BIGINT | 테이블 객체 식별자 |
| INSERT_LOG_COUNT | INTEGER | 전송한 INSERT 로그의 개수 |
| DELETE_LOG_COUNT | INTEGER | 전송한 DELETE 로그의 개수 |
| UPDATE_LOG_COUNT | INTEGER | 전송한 UPDATE 로그의 개수 |
| LOB_LOG_COUNT | INTEGER | 전송한 LOB 관련 로그의 개수 |

칼럼 정보

REP_NAME

지역서버에 생성된 이중화 객체의 이름이다.

CURRENT_TYPE

V\$REPGAP_PARALLEL 성능 뷰의 CURRENT_TYPE 칼럼 설명을 참조하기 바란다.

PARALLEL_ID

한 송신자를 위해 병렬 동작중인 여러 쓰레드 중 하나의 식별자이다.

V\$REPSENDER_STATISTICS

이중화 송신 쓰레드의 작업 별 수행시간에 대해 통계 정보를 보여준다. TIMED_STATISTICS 프로퍼티의 값이 1로 설정되어 있을 때만 통계정보가 이 뷰에 수집된다. 통계치 측정 간격과 측정 방식은 TIMER_THREAD_RESOLUTION과 TIMER_RUNNING_LEVEL 프로퍼티 값을 조정하여 정할 수 있다.

| Column name | Type | Description |
|--------------------------|-------------|-----------------------------------|
| REP_NAME | VARCHAR(40) | 이중화 객체의 이름 |
| PARALLEL_ID | INTEGER | 병렬 동작중인 이중화 송신 쓰레드들 중 하나의 식별자 |
| WAIT_NEW_LOG | BIGINT | 수신 쓰레드에게 보낼 새 로그를 대기하는 데 걸린 전체 시간 |
| READ_LOG_FROM_REPLBUFFER | BIGINT | 이중화 로그 버퍼로부터 로그를 읽어오는 데 걸린 전체 시간 |
| READ_LOG_FROM_FILE | BIGINT | 로그 파일로부터 로그를 읽어오는 데 걸린 전체 시간 |
| CHECK_USEFUL_LOG | BIGINT | 이중화 대상 로그인지 판별하는 데 걸린 전체 시간 |
| ANALYZE_LOG | BIGINT | 로그를 분석하고 XLog형태로 변환하는 데 걸린 전체 |
| SEND_XLOG | BIGINT | XLog를 송신하는 데 걸린 전체 시간 |
| RECV_ACK | BIGINT | ACK를 수신하는 데 걸린 전체 시간 |
| SET_ACKEDVALUE | BIGINT | 수신자로부터 받은 ACK값을 분석하는 데 걸린 전체 시간 |

칼럼 정보**REP_NAME**

지역서버에 생성된 이중화 객체의 이름이다.

PARALLEL_ID

송신자가 가지는 고유한 ID로 해당 송신 쓰레드가 속한 이중화 내에서 유일한 값을 가진다. 이 ID는 EAGER모드에서 병렬 송신자로 동작할 때, 각 쓰레드를 구별하기 위해 각 송신 쓰레드에 주어진다.

WAIT_NEW_LOG

수신 쓰레드로 보내기 위해 읽어올 로그가 로그 버퍼 또는 로그 파일에 쓰여지기를 기다리는 데 걸린 시간의 누적 값이다.

READ_LOG_FROM_REPLBUFFER

이중화 로그 버퍼에서 로그를 읽어오는 데 걸린 시간의 누적 값이다. REPLICATION_LOG_BUFFER_SIZE 값이 0보다 큰 값으로 설정되어 있는 경우에만 이 값이 유효하다.

READ_LOG_FROM_FILE

로그 파일에서 로그를 읽어오는 데 걸린 시간의 누적 값이다.

CHECK_USEFUL_LOG

이중화 대상 로그인지 판별하는 데 걸린 시간의 누적 값이다.

ANALYZE_LOG

로그를 분석하여 이중화를 위한 XLog로 변환하는 데 걸린 시간의 누적 값이다.

SEND_XLOG

XLog를 수신 스레드에 전송하는 데 걸린 시간의 누적 값이다.

RECV_ACK

수신 스레드로부터 ACK를 받기 위해 대기한 시간과 수신하는 데 걸린 시간의 누적 값이다.

SET_ACKEDVALUE

수신 스레드로부터 받은 ACK값을 분석하는데 걸린 시간의 누적 값이다.

V\$REPSENDER_TRANSTBL

이중화 송신자의 트랜잭션 테이블의 정보를 보여준다.

| Column name | Type | Description |
|-------------|-------------|--------------------|
| REP_NAME | VARCHAR(40) | 이중화 이름 |
| START_FLAG | BIGINT | 시작 옵션 |
| LOCAL_TID | BIGINT | 지역 트랜잭션 식별자 |
| REMOTE_TID | BIGINT | 원격 트랜잭션 식별자 |
| BEGIN_FLAG | INTEGER | 트랜잭션의 BEGIN 전송 여부 |
| BEGIN_SN | BIGINT | 트랜잭션의 최초 로그 레코드 SN |

칼럼 정보**REP_NAME**

지역서버에 생성된 이중화 객체의 이름이다.

START_FLAG

V\$REPSENDER 성능 뷰의 START_FLAG 칼럼의 설명을 참고한다.

LOCAL_TID

지역서버에서 실행되는 트랜잭션의 식별자이다.

REMOTE_TID

원격서버에서 실행되는 트랜잭션의 식별자이다.

V\$REPSENDER_TRANSTBL_PARALLEL

병렬 동작중인 이중화 송신 쓰레드의 트랜잭션 테이블의 정보를 보여준다.

| Column name | Type | Description |
|--------------|-------------|----------------------------------|
| REP_NAME | VARCHAR(40) | 이중화 이름 |
| CURRENT_TYPE | VARCHAR(9) | 이중화 송신 쓰레드의 유형 |
| LOCAL_TID | BIGINT | 지역 트랜잭션 식별자 |
| REMOTE_TID | BIGINT | 원격 트랜잭션 식별자 |
| BEGIN_FLAG | INTEGER | 트랜잭션의 BEGIN 전송 여부 |
| BEGIN_SN | BIGINT | 트랜잭션의 최초 로그 레코드 SN |
| PARALLEL_ID | INTEGER | 병렬 동작중인 여러 이중화 송신 쓰레드들 중 하나의 식별자 |

칼럼 정보**REP_NAME**

이중화 객체의 이름이다.

CURRENT_TYPE

V\$REPGAP_PARALLEL 성능 뷰의 CURRENT_TYPE 칼럼 설명을 참조하기 바란다.

LOCAL_TID

지역서버에서 실행되는 트랜잭션의 식별자이다.

REMOTE_TID

원격서버에서 실행되는 트랜잭션의 식별자이다.

PARALLEL_ID

병렬 동작중인 여러 이중화 송신 쓰레드들 중 하나의 식별자이다.

V\$REPSYNC

이중화를 사용해서 동기화 중인 테이블의 정보를 보여준다.

| Column name | Type | Description |
|-------------|--------------|---------------|
| REP_NAME | VARCHAR(40) | 이중화 객체의 이름 |
| SYNC_TABLE | VARCHAR(128) | 동기화 대상 테이블 이름 |

| Column name | Type | Description |
|-------------------|--------------|-------------------|
| SYNC_PARTITION | VARCHAR(128) | 동기화 대상 파티션 이름 |
| SYNC_RECORD_COUNT | BIGINT | 원격 서버에 동기화된 레코드 수 |
| SYNC_SN | BIGINT | 현재 사용하지 않음 |

칼럼 정보

REP_NAME

지역서버에 생성된 이중화 객체의 이름이다.

SYNC_TABLE

동기화 대상 테이블 이름이다.

SYNC_PARTITION

동기화 대상 파티션 이름이다.

SYNC_RECORD_COUNT

지역 서버에서 원격 서버로 이중화 테이블들의 데이터를 동기화할 때, REPLICATION_SYNC_TUPLE_COUNT 프로퍼티에 설정한 레코드 개수 단위로 데이터를 읽어서 처리한다.

이 칼럼은 이는 동기화 진행 중에는 동기화 된 레코드의 개수를 보여주며, 동기화가 완료되면 -1을 보여준다.

V\$REPL_REMOTE_META_REPLICATIONS

수신자가 가지고 있는 송신자의 SYS_REPLICATIONS_ 메타 테이블의 정보를 보여준다.

수신 스레드가 수행 중인 서버에서 조회할 수 있다.

| Column name | Type | Description |
|--------------------------|-------------|---|
| REPLICATION_NAME | VARCHAR(40) | 이중화 이름 |
| XSN | BIGINT | 송신자가 XLog 전송을 재개할 재시작 SN(Sequence Number) |
| ITEM_COUNT | INTEGER | 이중화 대상 테이블 개수 |
| CONFLICT_RESOLUTION | INTEGER | 이중화 충돌 해결 방법 |
| REPL_MODE | INTEGER | 기본 이중화 모드 |
| ROLE | INTEGER | 이중화 스레드의 역할 |
| OPTIONS | INTEGER | 부가적인 이중화 기능을 위한 플래그 |
| REMOTE_FAULT_DETECT_TIME | DATE | 원격 서버의 장애 감지 시각 |

칼럼 정보

REPLICATION_NAME

원격 서버의 이중화 이름으로, 이중화 생성 시 사용자가 명시한다.

XSN

원격 서버의 이중화가 시작될 때, 송신 스레드에서 로그 전송을 시작해야 할 SN을 나타낸다.

ITEM_COUNT

원격 서버의 이중화 대상 테이블의 개수이다. 해당 이중화에 대해 원격 서버의 SYS_REPL_ITEMS_ 메타 테이블에 이 수만큼 레코드들이 존재한다.

CONFLICT_RESOLUTION

원격 서버의 이중화 충돌 해결 방법을 기록한다.

- 0: 기본 값
- 1: Master Server로 동작
- 2: Slave Server로 동작

이중화 충돌 해결 방법에 대한 자세한 설명은 *Replication Manual*을 참조한다.

REPL_MODE

원격 서버의 이중화 생성시에 지정한 기본 이중화 모드이다.

- 0: LAZY MODE (기본 값)
- 2: EAGER MODE

기본 이중화 모드는 ALTER SESSION SET REPLICATION 구문으로 세션의 이중화 모드를 설정하지 않았을 때 사용된다.

기본 이중화 모드에 관한 자세한 내용은 *Replication Manual*을 참조하며, ALTER SESSION SET REPLICATION 구문에 관한 내용은 *SQL Reference*을 참조한다.

ROLE

이중화 스레드의 롤(ROLE)을 의미한다.

- 0: 일반 이중화 (롤을 지정하지 않은 경우)
- 1: Log Analyzer 전용 이중화 (FOR ANALYSIS 만 사용한 경우)
- 2: Propagable Logging (FOR PROPAGABLE LOGGING을 사용한 경우)
- 3: Propagation (FOR PROPAGATION 을 사용한 경우)
- 4: Log analyzer 용 이중화에서 Propagation 설정 한 경우(FOR ANALYSIS PROPAGATION을 사용한 경우)

Log Analyzer 전용 이중화에 대한 자세한 내용은 Log Analyzer User's Manual을 참고한다.

OPTIONS

원격 서버의 이중화 부가 기능을 나타내는 플래그이다. 이중화 옵션의 종류는 아래와 같으며, 각 옵션을 설정시 이진수로 제어되며, 십진수로 변환되어 표시된다. 두 개 이상의 옵션을 사용할 경우 각각의 옵션에 해당하는 이진수 합이 십진수로 반환된다.

- 0(000000): 이중화 옵션을 사용하지 않음

- 1(000001): 복구 옵션 사용
- 2(000010): 오프라인 옵션 사용
- 4(000100): 이중화 갭 해소 옵션 사용
- 8(001000): 병렬 적용자 옵션 사용
- 16(010000): 이중화 트랜잭션 그룹 옵션 사용
- 32(100000): 로컬 이중화 옵션 사용

REMOTE_FAULT_DETECT_TIME

원격 서버의 이중화 동작 중에 원격 서버의 장애를 감지한 시점을 기록한다.

V\$REPL_REMOTE_META_ITEMS

수신자가 가지고 있는 송신자의 SYS_REPL_ITEMS_ 메타 테이블 정보를 보여준다.

수신 스레드가 수행 중인 서버에서 조회할 수 있다.

| Column name | Type | Description |
|-----------------------|--------------|----------------------|
| REPLICATION_NAME | VARCHAR(40) | 이중화 이름 |
| TABLE_OID | BIGINT | 테이블 객체 식별자 |
| LOCAL_USER_NAME | VARCHAR(128) | 지역 서버의 대상 테이블 소유자 이름 |
| LOCAL_TABLE_NAME | VARCHAR(128) | 지역 서버의 대상 테이블 이름 |
| LOCAL_PARTITION_NAME | VARCHAR(128) | 지역 서버의 파티션 이름 |
| REMOTE_USER_NAME | VARCHAR(128) | 원격 서버의 대상 테이블 소유자 이름 |
| REMOTE_TABLE_NAME | VARCHAR(128) | 원격 서버의 대상 테이블 이름 |
| REMOTE_PARTITION_NAME | VARCHAR(128) | 원격 서버의 파티션 이름 |
| INVALID_MAX_SN | BIGINT | 건너 뛴 로그의 최대 SN |

칼럼 정보

REPLICATION_NAME

원격 서버의 사용자가 명시한 이중화 이름으로, 원격 서버의 SYS_REPLICATIONS_ 메타 테이블의 한 REPLICATION_NAME 값과 동일하다.

TABLE_OID

원격 서버의 이중화 대상 테이블 또는 파티션의 식별자로, 원격 서버의 SYS_TABLES_ 메타 테이블의 한 TABLE_OID 값 또는 SYS_TABLES_PARTITIONS_의 한 PARTITION_OID 값과 동일하다.

LOCAL_USER_NAME

원격 서버의 이중화 대상 테이블 소유자의 사용자 이름으로, 원격 서버의 SYS_USERS_ 메타 테이블의 한 USER_NAME 값과 동일하다.

LOCAL_TABLE_NAME

원격 서버의 이중화 대상 테이블의 이름으로, 원격 서버의 SYS_TABLES_ 메타 테이블의 한 TABLE_NAME 값과 동일하다.

LOCAL_PARTITION_NAME

원격 서버의 이중화 대상 파티션의 이름이다.

REMOTE_USER_NAME

지역 서버의 이중화 대상 테이블 소유자의 사용자 이름으로, 지역 서버의 SYS_USERS_ 메타 테이블의 한 USER_NAME 값과 동일하다.

REMOTE_TABLE_NAME

지역 서버의 이중화 대상 테이블의 이름으로, 지역 서버의 SYS_TABLES_ 메타 테이블의 한 TABLE_NAME 값과 동일하다.

REMOTE_PARTITION_NAME

지역 서버의 이중화 대상 파티션의 이름이다.

INVALID_MAX_SN

원격 서버의 이중화 대상 테이블에 DDL구문 또는 동기화 작업이 수행되는 시점에서 가장 최근에 기록된 SN이 저장된다. 해당 SN까지의 테이블 로그를 이중화에서 건너뛴다.

V\$REPL_REMOTE_META_COLUMNS

수신자가 가지고 있는 송신자의 SYS_REPL_OLD_COLUMNS_ 메타 테이블 정보를 보여 준다.

수신 스레드가 수행 중인 서버에서 조회할 수 있다.

| Column name | Type | Description |
|----------------------|--------------|-----------------------|
| REPLICATION_NAME | VARCHAR(40) | 이중화 이름 |
| TABLE_OID | BIGINT | 테이블 객체 식별자 |
| COLUMN_NAME | VARCHAR(128) | 칼럼 이름 |
| MT_DATATYPE_ID | INTEGER | 데이터 타입 식별자 |
| MT_LANGUAGE_ID | INTEGER | 언어 식별자 |
| MT_FLAG | INTEGER | 내부 플래그 |
| MT_PRECISION | INTEGER | 정밀도 |
| MT_SCALE | INTEGER | 소수 자릿수 |
| MT_ENCRYPT_PRECISION | INTEGER | 암호화 칼럼 정밀도 |
| MT_POLICY_NAME | VARCHAR(16) | 암호화 칼럼에 사용된 정책의 이름 |
| MT_SRID | INTEGER | GEOMETRY 칼럼에 적용된 SRID |
| SM_ID | INTEGER | 칼럼 식별자 |

| Column name | Type | Description |
|-------------|---------|-------------|
| SM_FLAG | INTEGER | 내부 플래그 |
| SM_OFFSET | INTEGER | 내부 오프셋 |
| SM_SIZE | INTEGER | 내부 크기 |
| QP_FLAG | INTEGER | 내부 플래그 |

칼럼 정보

REPLICATION_NAME

원격 서버의 사용자가 명시한 이중화 이름이다. 메타 테이블의 한 REPLICATION_NAME값과 동일하다.

TABLE_OID

원격 서버의 이중화 송신 스레드가 현재 사용중인 이중화 대상 테이블의 식별자이다. SYS_TABLES_ 메타 테이블의 어떤 TABLE_OID 값과도 일치하지 않을 수 있다.

COLUMN_NAME

원격 서버의 이중화 송신 스레드가 현재 복제중인 이중화 대상 칼럼의 이름이다.

MT_DATATYPE_ID

데이터 타입 식별자로, 내부 값이다.

MT_LANGUAGE_ID

언어 식별자로, 내부 값이다.

MT_FLAG

Altibase 서버가 사용하는 내부 플래그이다.

MT_PRECISION

숫자 타입의 경우, 칼럼의 정밀도 (숫자 자리수)를 나타낸다. 타입의 경우, 문자형 데이터 타입의 길이를 나타낸다.

MT_SCALE

숫자 타입의 경우, 칼럼의 소수점 이하 자릿수를 나타낸다.

MT_ENCRYPT_PRECISION

암호화된 칼럼의 정밀도 (크기)를 나타낸다.

MT_POLICY_NAME

암호화된 칼럼의 경우, 칼럼에 적용된 보안 정책의 이름을 나타낸다.

MT_SRID

GEOMETRY 칼럼의 경우, 칼럼에 적용된 SRID를 나타낸다.

SM_ID

칼럼 식별자이다. 0부터 시작한다.

SM_FLAG

Altibase 서버가 사용하는 내부 플래그이다.

SM_OFFSET

Altibase 서버가 사용하는 내부 오프셋이다.

SM_SIZE

Altibase 서버가 사용하는 내부 크기이다.

QP_FLAG

Altibase 서버가 내부적으로 사용하는 플래그이다.

V\$REPL_REMOTE_META_INDEX_COLUMNS

수신자가 가지고 있는 송신자의 SYS_REPL_OLD_INDEX_COLUMNS_ 메타 테이블 정보를 보여 준다.

수신 쓰레드가 수행 중인 서버에서 조회할 수 있다.

| Column name | Type | Description |
|------------------|-------------|-------------|
| REPLICATION_NAME | VARCHAR(40) | 이중화 이름 |
| TABLE_OID | BIGINT | 테이블 객체 식별자 |
| INDEX_ID | INTEGER | 인덱스 식별자 |
| KEY_COLUMN_ID | INTEGER | 칼럼 식별자 |
| KEY_COLUMN_FLAG | INTEGER | 내부 플래그 |

칼럼 정보**REPLICATION_NAME**

원격 서버의 사용자가 명시한 이중화 이름으로, 원격 서버의 SYS_REPLICATIONS_ 메타 테이블의 한 REPLICATION_NAME 값과 동일하다.

TABLE_OID

원격 서버의 이중화 송신 쓰레드가 현재 복제 중인 이중화 대상 테이블의 식별자이다. 원격 서버의 SYS_TABLES_ 메타 테이블의 어떤 TABLE_OID 값과도 일치하지 않을 수 있다.

INDEX_ID

원격 서버의 이중화 송신 쓰레드가 현재 복제 중인 이중화 대상 인덱스의 식별자이다.

KEY_COLUMN_ID

인덱스를 구성하는 칼럼의 식별자이다.

KEY_COLUMN_FLAG

인덱스를 구성하는 칼럼의 내부 플래그이다.

V\$REPL_REMOTE_META_INDICES

수신자가 가지고 있는 송신자의 SYS_REPL_OLD_INDICES_ 메타 테이블의 정보를 보여 준다.

수신 스레드가 수행 중인 서버에서 조회할 수 있다.

| Column name | Type | Description |
|------------------|--------------|----------------|
| REPLICATION_NAME | VARCHAR(40) | 이중화 이름 |
| TABLE_OID | BIGINT | 테이블 객체 식별자 |
| INDEX_ID | INTEGER | 인덱스 식별자 |
| INDEX_NAME | VARCHAR(128) | 인덱스 이름 |
| TYPE_ID | INTEGER | 인덱스 타입 식별자 |
| IS_UNIQUE | CHAR(1) | 글로벌 유니크 인덱스 여부 |
| IS_RANGE | CHAR(1) | 범위 검색 가능 여부 |

칼럼 정보**REPLICATION_NAME**

원격 서버의 사용자가 명시한 이중화 이름으로, 원격 서버의 SYS_REPLICATIONS_ 메타 테이블의 한 REPLICATION_NAME과 동일하다.

TABLE_OID

원격 서버의 이중화 송신 스레드가 현재 복제 중인 이중화 대상 테이블의 식별자이다. 원격 서버의 SYS_TABLES_ 메타 테이블의 어떤 TABLE_OID 값과도 일치하지 않을 수 있다.

INDEX_ID

원격 서버의 이중화 송신 스레드가 현재 복제 중인 이중화 대상 인덱스의 식별자이다.

INDEX_NAME

원격 서버의 이중화 송신 스레드가 현재 복제 중인 이중화 대상 인덱스의 이름이다.

TYPE_ID

인덱스 유형 식별자로, 내부 값이다.

IS_UNIQUE

글로벌 유니크 인덱스인지 여부를 나타낸다. 'Y'는 글로벌 유니크를 나타내고, 'N'은 글로벌 유니크가 아님을 나타낸다.

IS_RANGE

범위 검색 가능 여부를 나타낸다. 'Y'는 범위 검색이 가능한 인덱스이고, 'N'은 범위 검색이 불가능한 인덱스임을 나타낸다.

V\$REPL_REMOTE_META_CHECKS

수신자가 가지고 있는 송신자의 이중화 테이블의 제약 조건에 관한 정보를 보여 준다.

수신 스레드가 수행 중인 서버에서 조회할 수 있다.

| Column name | Type | Description |
|------------------|---------------|--------------------|
| REPLICATION_NAME | VARCHAR(40) | 이중화 이름 |
| TABLE_OID | BIGINT | 테이블 객체 식별자 |
| CONSTRAINT_ID | INTEGER | 제약조건 식별자 |
| CONSTRAINT_NAME | VARCHAR(128) | 제약조건 이름 |
| COLUMN_CNT | INTEGER | 제약조건에 관련된 칼럼 개수 |
| CHECK_CONDITION | VARCHAR(4000) | CHECK 제약조건의 조건 문자열 |

칼럼 정보**REPLICATION_NAME**

원격 서버의 사용자가 명시한 이중화 이름으로, 원격 서버의 SYS_REPLICATIONS_ 메타 테이블의 한 REPLICATION_NAME과 동일하다.

TABLE_OID

원격 서버의 이중화 송신 스레드가 현재 복제 중인 이중화 대상 테이블의 식별자이다. 원격 서버의 SYS_TABLES_ 메타 테이블의 어떤 TABLE_OID 값과도 일치하지 않을 수 있다.

CONSTRAINT_ID

제약 조건 식별자로 시스템 시퀀스에 의해 자동으로 부여된다.

CONSTRAINT_NAME

제약 조건의 이름을 나타낸다.

COLUMN_CNT

제약 조건에 관련된 칼럼들의 개수를 나타낸다. 예를 들어 UNIQUE (i1, i2, i3) 과 같은 제약 조건을 생성하였다면 이 값은 3일 것이다.

CHECK_CONDITION

사용자가 CHECK 제약조건을 지정할 때 정의한 무결성 규칙(Integrity Rule)을 나타낸다.

V\$RESERVED_WORDS

SQL에서 사용되는 모든 키워드를 보여준다.

| Column name | Type | Description |
|---------------|-------------|-------------|
| KEYWORD | VARCHAR(40) | 키워드의 이름 |
| LENGTH | INTEGER | 키워드의 길이 |
| RESERVED_TYPE | INTEGER | 키워드의 타입 |

칼럼 정보

KEYWORD

SQL에서 사용 되는 키워드의 이름이다.

LENGTH

키워드의 길이이다.

RESERVED_TYPE

키워드의 타입이다.

- 0: 테이블의 칼럼 이름으로 사용할 수 없다.
- 1: 테이블의 칼럼 이름으로 사용할 수 있다.

V\$SBUFFER_STAT

보조 버퍼(Secondary Buffer)에 대한 통계 정보를 보여준다.

| Column name | Type | Description |
|------------------------|---------|----------------------|
| PAGE_COUNT | INTEGER | 보조 버퍼의 크기(페이지 개수) |
| HASH_BUCKET_COUNT | INTEGER | 해쉬 테이블의 버킷 개수 |
| HASH_CHAIN_LATCH_COUNT | INTEGER | 해쉬 테이블에 사용되는 래치 개수 |
| CHECKPOINT_LIST_COUNT | INTEGER | 체크포인트 리스트 개수 |
| HASH_PAGES | INTEGER | 해쉬 테이블에 등록된 페이지 개수 |
| FLUSH_PAGES | INTEGER | 플러시된 페이지 개수 |
| CHECKPOINT_LIST_PAGES | INTEGER | 체크포인트 리스트에 있는 페이지 개수 |
| GET_PAGES | BIGINT | 페이지가 요청된 횟수 |
| READ_PAGES | BIGINT | 페이지를 읽어간 횟수 |
| WRITE_PAGES | BIGINT | 페이지를 쓴 횟수 |
| HIT_RATIO | DOUBLE | 보조 버퍼 적중률 |
| SINGLE_PAGE_READ_USEC | BIGINT | single page Read 시간 |

| Column name | Type | Description |
|------------------------|--------|------------------------------|
| SINGLE_PAGE_WRITE_USEC | BIGINT | single page Write 시간 |
| MPR_READ_USEC | BIGINT | Full scan Read시 Read한 시간 |
| MPR_READ_PAGE_COUNT | BIGINT | Full scan Read시 Read한 페이지 수 |
| SINGLE_READ_PERF | DOUBLE | single 페이지를 read한 양(KB)/ sec |
| MULTI_READ_PERF | DOUBLE | mpr로 페이지를 read한 양(KB)/ sec |

칼럼 정보

PAGE_COUNT

보조 버퍼의 크기가 페이지 개수로 표시된다.

HASH_BUCKET_COUNT

해쉬 테이블의 버킷 개수를 나타낸다.

HASH_CHAIN_LATCH_COUNT

해쉬 테이블에 사용되는 체인 래치의 개수를 나타낸다.

CHECKPOINT_LIST_COUNT

체크포인트 리스트 개수를 나타낸다.

HASH_PAGES

해쉬 테이블에 삽입된 페이지 개수를 나타낸다. 이 값은 현재 사용중인 페이지 수를 의미한다.

FLUSH_PAGES

서버 구동 이후부터 현재까지 보조 버퍼에서 플러시된 페이지의 총 개수를 나타낸다.

CHECKPOINT_LIST_PAGES

체크포인트 리스트에 존재하는 페이지 수를 나타낸다.

GET_PAGES

버퍼 관리자가 서버 구동 이후부터 현재까지 데이터 읽기 목적으로 보조 버퍼의 페이지를 요청한 누적 횟수를 나타낸다.

READ_PAGES

페이지 요청 시 버퍼 관리자가 보조 버퍼에서 메모리 버퍼로 페이지를 읽은 누적 횟수이다.

WRITE_PAGES

보조 버퍼에 페이지를 쓴 누적 횟수이다.

HIT_RATIO

서버 구동 이후부터 현재까지 보조 버퍼에 대한 누적 적중률 (hit ratio)을 나타낸다.

SINGLE_PAGE_READ_USEC

보조 버퍼에서 하나의 페이지를 읽는데 소요된 누적 시간이다. (단위: micro-seconds)

SINGLE_PAGE_WRITE_USEC

하나의 페이지를 보조 버퍼에 쓰는데 소요된 누적 시간이다. (단위: micro-seconds)

MPR_READ_USEC

보조 버퍼에서 여러 페이지를 동시에 읽는데 소요된 누적 시간이다. (단위: micro-seconds)

MPR_READ_PAGE_COUNT

"full 스캔" 수행을 위해 보조 버퍼에서 여러 데이터 페이지를 동시에 읽은 페이지의 누적 개수이다.

SINGLE_READ_PERF

하나의 데이터 페이지를 보조 버퍼에서 읽을 때의 초당 읽은 평균 바이트 수이다. (단위: kB/sec)

MULTI_READ_PERF

"full 스캔" 수행을 위해 보조 버퍼에서 여러 데이터 페이지들을 동시에 읽을 때의 초당 읽은 평균 바이트 수이다. (단위: kB/sec)

V\$SEGMENT

디스크 테이블과 디스크 인덱스를 구성하는 세그먼트의 상태, 종류 및 할당된 익스텐트의 개수를 보여준다.

| Column name | Type | Description |
|--------------------|------------|----------------------|
| SPACE_ID | INTEGER | 테이블스페이스 식별자 |
| TABLE_OID | BIGINT | 테이블 헤더의 객체 식별자 |
| SEGMENT_PID | INTEGER | 세그먼트 페이지의 식별자 |
| SEGMENT_TYPE | VARCHAR(7) | 세그먼트의 종류 |
| SEGMENT_STATE | VARCHAR(7) | 세그먼트의 상태 |
| EXTENT_TOTAL_COUNT | BIGINT | 세그먼트에 할당된 익스텐트의 총 개수 |

칼럼 정보**SEGMENT_PID**

세그먼트 헤더가 저장된 페이지의 식별자이다.

SEGMENT_TYPE

- INDEX: 해당 세그먼트가 인덱스 세그먼트임을 나타낸다.
- LOB: 해당 세그먼트가 LOB 세그먼트임을 나타낸다.
- TABLE: 해당 세그먼트가 테이블 세그먼트임을 나타낸다.
- TSSEG: 해당 세그먼트가 TSS 세그먼트임을 나타낸다.
- UDSEG: 해당 세그먼트가 언두 세그먼트임을 나타낸다.

SEGMENT_STATE

- USED: 해당 세그먼트가 사용 중임을 나타낸다.
- FREE: 해당 세그먼트가 비어 있음을 나타낸다.

EXTENT_TOTAL_COUNT

세그먼트에 할당된 익스텐트의 총 개수이다.

V\$SEQ

시퀀스 관련 정보를 보여준다.

| Column name | Type | Description |
|---------------|------------|--------------|
| SEQ_OID | BIGINT | 시퀀스 객체 식별자 |
| CURRENT_SEQ | BIGINT | 현재 시퀀스 값 |
| START_SEQ | BIGINT | 시퀀스의 시작 값 |
| INCREMENT_SEQ | BIGINT | 시퀀스의 증가 값 |
| CACHE_SIZE | BIGINT | 캐시 크기 |
| MAX_SEQ | BIGINT | 시퀀스 최댓값 |
| MIN_SEQ | BIGINT | 시퀀스 최솟값 |
| IS_CYCLE | VARCHAR(7) | 시퀀스 값의 순환 여부 |

칼럼 정보**SEQ_OID**

고유한 시퀀스 식별자로 이는 시퀀스 생성시 시스템에 의해 할당된다. 이 값은 SYS_TABLES_ 메타 테이블의 TABLE_TYPE 칼럼의 값이 'S' 인 레코드들 중 한 TABLE_OID 칼럼 값과 일치한다.

CURRENT_SEQ

현재 시퀀스의 값을 나타낸다.

START_SEQ

시퀀스 생성시 지정한 시퀀스의 시작 값을 나타낸다.

INCREMENT_SEQ

시퀀스 번호가 증가되는 값을 나타낸다.

MAX_SEQ

시퀀스를 사용해서 생성 가능한 최댓값을 나타낸다.

MIN_SEQ

시퀀스를 사용해서 생성 가능한 최솟값을 나타낸다.

IS_CYCLE

해당 시퀀스가 최댓값에 도달한 경우 순환하여 최솟값부터 다시 시퀀스 값을 생성할 것인지 여부를 나타낸다.

- YES: 순환 한다
- NO: 순환 하지 않는다. 만약 시퀀스가 최댓값에 도달할 경우 다음 시퀀스 값을 요청하면, 에러가 발생한다.

V\$SERVICE_THREAD

서비스 쓰레드 정보를 보여준다.

| Column name | Type | Description |
|------------------|-------------|-------------------------------------|
| ID | INTEGER | 서비스 쓰레드 식별자 |
| TYPE | VARCHAR(20) | 서비스 쓰레드 접속 방법 |
| STATE | VARCHAR(10) | 서비스 쓰레드의 현재 상태 |
| RUN_MODE | VARCHAR(9) | 서비스 쓰레드 운영 모드 |
| SESSION_ID | BIGINT | 서비스 쓰레드가 수행중인 세션의 식별자 |
| STATEMENT_ID | INTEGER | 서비스 쓰레드가 수행중인 Statement 의 식별자 |
| START_TIME | INTEGER | 서비스 쓰레드가 생성된 시각 |
| EXECUTE_TIME | BIGINT | 서비스 쓰레드가 현재 쿼리를 수행하는데 걸린 시간 |
| TASK_COUNT | INTEGER | 서비스 쓰레드가 처리중인 세션의 개수 |
| READY_TASK_COUNT | INTEGER | 서비스 쓰레드가 요청을 처리해 주기를 대기하고 있는 세션의 개수 |
| THREAD_ID | BIGINT | 서비스 쓰레드의 thread id |

서버에서 클라이언트의 요청을 받아 질의를 수행하는 쓰레드를 서비스 쓰레드라 한다. Altibase는 이러한 서비스 쓰레드를 생성하는 아래의 두 가지 모드를 제공한다:

- 전용 쓰레드 모드(Dedicated Thread Mode):
서버에 다수의 클라이언트가 접속하여 질의를 수행하는 경우, 서버는 각 클라이언트 세션별로 하나의 서비스 쓰레드를 생성하여 질의를 수행한다.
- 멀티플렉싱 쓰레드 모드(Multiplexing Thread Mode):
Altibase 서버는 서버에 최적화된 개수의 서비스 쓰레드만 생성하고, 클라이언트 세션들이 이를 공유한다.

Altibase는 필요에 따라 동적으로 서비스 쓰레드를 추가하거나 삭제하여 항상 최적화된 개수의 서비스 쓰레드를 유지하도록 설계되어 있다. 단, DEDICATED_THREAD_INIT_COUNT 또는 MULTIPLEXING_THREAD_COUNT 프로퍼티에서 지정한 최소 개수만큼의 서비스 쓰레드는 유지한다.

칼럼 정보

ID

서비스 쓰레드의 식별자를 나타낸다. System thread ID (Light Weight Process ID등과 같은)가 아니라 Altibase 내부에서 유지하는 ID이다.

TYPE

서비스 쓰레드 접속 방법으로 다음과 같은 값을 가진다.

- SOCKET(MULTIPLEXING): TCP 또는 Unix Domain 방식
- SOCKET(DEDICATED): TCP 또는 Unix Domain 방식
- IPC: IPC 방식
- IPCDA : IPCDA 방식

STATE

서비스 쓰레드의 현재 상태를 나타낸다. 다음과 같은 값을 가진다.

- NONE: 서비스 쓰레드가 초기화된 상태
- POLL: 서비스 쓰레드가 이벤트를 기다리고 있는 상태
- QUEUE-WAIT: 서비스 쓰레드가 Queue를 대기하는 상태
- EXECUTE: 서비스 쓰레드가 Statement를 수행중인 상태
- UNKNOWN: 서비스 쓰레드의 상태를 알 수 없음

RUN_MODE

서비스 쓰레드의 운영 모드를 나타내는 것으로 SHARED 또는 DEDICATED 두 가지 모드가 있다.

- SHARED: 여러 클라이언트 연결들이 하나의 서비스 쓰레드를 공유한다.
- DEDICATED: 하나의 클라이언트 연결(Connection)이 하나의 서비스 쓰레드에 할당되어 해당 서비스 쓰레드를 독점하여 사용한다.

현재 서비스 쓰레드의 운영 모드 전환은 큐 (QUEUE) 관련 작업에만 적용되고 있으며, SHARED 모드에서 DEDICATED 모드로만 전환할 수 있다.

STATEMENT_ID

서비스 쓰레드가 수행중인 SQL statement의 식별자를 나타낸다.

START_TIME

서비스 쓰레드가 생성된 시각을 시스템 시간으로 나타낸다. (단위: 초)

EXECUTE_TIME

서비스 쓰레드가 현재 수행하고 있는 질의 (query)를 수행하는 데 걸린 시간을 나타낸다. (단위: 마이크로 초)

TASK_COUNT

서비스 쓰레드에 할당된 전체 세션의 개수를 나타낸다.

READY_TASK_COUNT

서비스 스레드가 자신의 요청을 처리해 주기를 대기하고 있는 세션의 개수를 나타낸다.

V\$SERVICE_THREAD_MGR

서비스 스레드가 생성되거나 삭제된 횟수를 누적해서 보여준다.

| Column name | Type | Description |
|------------------|---------|-----------------|
| ADD_THR_COUNT | INTEGER | 서비스 스레드가 추가된 횟수 |
| REMOVE_THR_COUNT | INTEGER | 서비스 스레드가 삭제된 횟수 |

Altibase는 필요에 따라 동적으로 서비스 스레드를 추가하거나 삭제하여 항상 최적화된 개수의 서비스 스레드를 유지하는데, 이 성능 뷰는 서비스 스레드가 추가되거나 삭제된 횟수를 누적해서 보여준다.

칼럼 정보

ADD_THR_COUNT

서비스 스레드가 동적으로 추가된 횟수의 누적값이다.

REMOVE_THR_COUNT

서비스 스레드가 동적으로 삭제된 횟수의 누적값이다.

V\$SESSION

Altibase 내부에 생성된 클라이언트 세션에 대한 정보를 보여준다.

| Column name | Type | Description |
|-------------------------|--------------|------------------------|
| ID | BIGINT | 세션 식별자 |
| TRANS_ID | BIGINT | 세션에서 현재 수행중인 트랜잭션의 식별자 |
| TASK_STATE | VARCHAR(11) | 태스크 상태 |
| COMM_NAME | VARCHAR(64) | 접속 정보 |
| XA_SESSION_FLAG | INTEGER | XA 세션 플래그 |
| XA_ASSOCIATE_FLAG | INTEGER | XA associate 플래그 |
| QUERY_TIME_LIMIT | BIGINT | 아래 참조 |
| DDL_TIME_LIMIT | BIGINT | 아래 참조 |
| FETCH_TIME_LIMIT | BIGINT | 아래 참조 |
| UTRANS_TIME_LIMIT | BIGINT | 아래 참조 |
| IDLE_TIME_LIMIT | BIGINT | 아래 참조 |
| IDLE_START_TIME | INTEGER | 아래 참조 |
| ACTIVE_FLAG | INTEGER | 트랜잭션 활성 플래그 |
| OPENED_STMT_COUNT | INTEGER | 사용 중인 구문 개수 |
| CLIENT_PACKAGE_VERSION | VARCHAR(40) | 클라이언트 패키지 버전 |
| CLIENT_PROTOCOL_VERSION | VARCHAR(40) | 클라이언트의 통신 프로토콜 버전 |
| CLIENT_PID | BIGINT | 클라이언트 프로세스 아이디 |
| CLIENT_TYPE | VARCHAR(40) | 접속한 클라이언트의 타입 |
| CLIENT_APP_INFO | VARCHAR(128) | 접속한 애플리케이션의 타입 |

| Column name | Type | Description |
|------------------------------------|--------------|--|
| CLIENT_NLS | VARCHAR(40) | 클라이언트 문자 집합 |
| DB_USERNAME | VARCHAR(128) | 데이터베이스 사용자 이름 |
| DB_USERID | INTEGER | 데이터베이스 사용자 식별자 |
| DEFAULT_TBSID | BIGINT | 사용자의 디폴트 테이블스페이스 식별자 |
| DEFAULT_TEMP_TBSID | BIGINT | 사용자의 디폴트 임시(temp) 테이블스페이스 식별자 |
| SYSDBA_FLAG | INTEGER | Sysdba 로 접속했는지 여부 |
| AUTOCOMMIT_FLAG | INTEGER | Autocommit 플래그 |
| SESSION_STATE | VARCHAR(13) | 세션의 상태 |
| ISOLATION_LEVEL | INTEGER | 고립도 (isolation level) |
| REPLICATION_MODE | INTEGER | 이중화 모드 |
| TRANSACTION_MODE | INTEGER | 트랜잭션 모드 |
| COMMIT_WRITE_WAIT_MODE | INTEGER | 아래 참조 |
| OPTIMIZER_MODE | INTEGER | 최적화 모드 |
| HEADER_DISPLAY_MODE | INTEGER | SELECT 질의의 결과 출력시, 칼럼 이름만 출력할 것인지 테이블 이름도 함께 출력할 것인지 여부. 0: 칼럼 이름과 함께 테이블 이름도 출력 1: 칼럼 이름만 출력 |
| CURRENT_STMT_ID | INTEGER | 사용 중인 statement 식별자 |
| STACK_SIZE | INTEGER | 스택 크기(단위: bytes) |
| DEFAULT_DATE_FORMAT | VARCHAR(64) | 디폴트 날짜 형식 예) DD-MON-RRRR |
| TRX_UPDATE_MAX_LOGSIZE | BIGINT | DML 로그의 최대 크기(단위: bytes) |
| PARALLE_DML_MODE | INTEGER | Deprecated |
| LOGIN_TIME | INTEGER | 클라이언트 접속 시간 |
| FAILOVER_SOURCE | VARCHAR(256) | FailOver가 일어났을 때의 접속 정보 |
| NLS_TERRITORY | VARCHAR(40) | 세션의 지역 이름 |
| NLS_ISO_CURRENCY | VARCHAR(40) | 세션의 ISO 통화 기호 |
| NLS_CURRENCY | VARCHAR(10) | 세션의 지역 통화 기호 |
| NLS_NUMERIC_CHARACTERS | VARCHAR(2) | 세션의 소수점 문자와 그룹 구분자 |
| TIME_ZONE | VARCHAR(40) | 세션에 설정된 타임 존의 지역 이름, 약어 또는 UTC_OFFSET |
| LOB_CACHE_THRESHOLD | INTEGER | LOB_CACHE_THRESHOLD 프로퍼티에 설정된 값 |
| QUERY_REWRITE_ENABLE | VARCHAR(7) | QUERY_REWRITE_ENABLE 프로퍼티에 설정된 값 |
| DBLINK_GLOBAL_TRANSACTION_LEVEL | INTEGER | DBLINK_GLOBAL_TRANSACTION_LEVEL 프로퍼티에 설정된 값 |
| DBLINK_REMOTE_STATEMENT_AUTOCOMMIT | INTEGER | DBLINK_REMOTE_STATEMENT_AUTOCOMMIT 프로퍼티에 설정된 값 |
| MAX_STATEMENTS_PER_SESSION | INTEGER | 세션에 허용된 STATEMENT 최대 개수 |
| SSL_CIPHER | VARCHAR(256) | 현재 사용하는 암호화 알고리즘 |
| SSL_CERTIFICATE_SUBJECT | VARCHAR(256) | 클라이언트 인증서 정보 |
| SSL_CERTIFICATE_ISSUER | VARCHAR(256) | 클라이언트 인증서 발행기관 |
| CLIENT_INFO | VARCHAR(128) | 접속한 애플리케이션 타입 |
| MODULE | VARCHAR(128) | 수행 중인 프로시저 모듈이름 |
| ACTION | VARCHAR(128) | 수행 중인 프로시저 동작상태 |
| REPLICATION_DDL_SYNC | INTEGER | 이중화 중 DDL 복제 여부 |
| REPLICATION_DDL_TIMELIMIT | BIGINT | 아래 참조 |
| MESSAGE_CALLBACK | VARCHAR(7) | 클라이언트 메시지 콜백 등록상태 |

칼럼 정보

ID

현재 연결된 세션의 고유 식별자를 나타낸다.

TRANS_ID

세션에서 현재 수행하고 있는 트랜잭션 식별자를 나타낸다. 현재 수행중인 트랜잭션이 없으면 이 값은 -1 이 된다.

TASK_STATE

현재 태스크의 상태를 아래와 같이 나타낸다.

| STATE | Description |
|-------------|---|
| WAITING | 클라이언트로 부터 요청이 들어오기를 기다리고 있는 상태 |
| READY | 클라이언트로부터 수신된 요청을 처리하기 위한 쓰레드를 할당받기 위해 대기하는 상태 |
| EXECUTING | 쓰레드를 할당받은 후 작업을 수행중인 상태 |
| QUEUE WAIT | QUEUE에 입력되기를 기다리는 상태. 큐에 입력된 후에 dequeue된다. |
| QUEUE READY | QUEUE에 입력된 후, dequeue를 위해 쓰레드 할당을 기다리는 상태 |
| UNKNOWN | 알 수 없는 상태 |

COMM_NAME

클라이언트의 접속 정보를 나타낸다. 통신 타입 (TCP/IP, UNIX domain 소켓, IPC, IPCDA 또는 SSL)에 따라서 보여주는 포맷이 다르다. TCP/IP와 SSL의 경우에는 클라이언트 IP 주소와 연결 포트 번호가 여기에 포함된다.

XA_SESSION_FLAG

현재의 세션이 XA 세션인지 나타낸다.

- 0: XA 세션이 아니다

XA_ASSOCIATE_FLAG

XA 세션과 글로벌 트랜잭션 간의 Association 상태를 나타낸다.

QUERY_TIME_LIMIT

현재 세션의 쿼리 시간 초과(timeout) 값을 나타낸다.

DDL_TIME_LIMIT

현재 세션의 DDL문 수행 시간 초과(timeout) 값을 나타낸다.

FETCH_TIME_LIMIT

현재 세션의 Fetch 시간 초과(timeout) 값을 나타낸다.

UTRANS_TIME_LIMIT

현재 세션의 갱신(update) 트랜잭션 시간 초과(timeout) 값을 나타낸다.

IDLE_TIME_LIMIT

현재 세션의 Idle 시간 초과(timeout) 값을 나타낸다.

IDLE_START_TIME

세션이 Idle상태로 진입한 시각을 표시한다.

ACTIVE_FLAG

세션이 어떤 구문을 수행하고 있을 경우 1로 나타난다. 그러나 단지 연결만 되어있거나, 트랜잭션을 커밋(commit) 또는 롤백(rollback)한 이후라면 0으로 표시된다.

OPENED_STMT_COUNT

해당 세션이 현재 수행중인 구문 (statement)의 개수를 나타낸다.

CLIENT_PACKAGE_VERSION

접속된 클라이언트의 패키지 버전이다.

CLIENT_PROTOCOL_VERSION

접속된 클라이언트가 사용하는 통신 프로토콜의 버전이다.

CLIENT_PID

접속된 클라이언트의 프로세스 아이디를 나타낸다. 자바 응용프로그램일 경우 이 값은 유효하지 않다.

CLIENT_TYPE

접속된 클라이언트의 타입을 표시하는 문자열이다.

아래처럼 구성된다.

```
CLIENT_TYPE ::= app-type hyphen word-size endian
  app-type   ::= CLI | WIN_ODBC | UNIX_ODBC
  hyphen     ::= -
  word-size  ::= 32 | 64
  endian     ::= BE | LE
  BE : Big Endian, LE : Little Endian
```

예)

```
CLI-32LE
UNIX_ODBC-32BE
```

CLIENT_APP_INFO

접속된 클라이언트의 애플리케이션 정보이다. 클라이언트 응용프로그램에 의해 설정되는 값이다.

CLIENT_NLS

접속된 클라이언트의 문자 집합을 나타낸다.

DB_USERNAME

접속된 클라이언트가 사용하는 사용자 이름을 나타낸다.

DB_USERID

사용자명에 대하여 Altibase가 숫자로 식별하는 아이디를 나타낸다.

DEFAULT_TBSID

사용자의 디폴트 테이블스페이스 식별자를 나타낸다.

DEFAULT_TEMP_TBSID

사용자의 디폴트 임시 테이블스페이스 식별자를 나타낸다.

SYSDBA_FLAG

접속된 세션이 sysdba 모드인지 아닌지를 나타낸다.

- 1: sysdba 모드

AUTOCOMMIT_FLAG

접속된 세션이 autocommit 모드인지를 나타낸다.

- 0: non-autocommit
- 1: autocommit

SESSION_STATE

| STATE | Description |
|---------------|--|
| INIT | 클라이언트로부터 요청이 들어오기를 기다리고 있는 상태 |
| AUTH | 사용자 인증을 마친 상태 |
| SERVICE READY | 서비스 준비상태 (트랜잭션을 만들 수 없는 상태로 XA 세션의 경우에만 이 상태로 올 수 있다.) |
| SERVICE | 서비스 상태 |
| END | 정상 종료 (트랜잭션이 있을 경우 커밋) 하고 있는 상태 |
| ROLLBACK | 비정상 종료 (트랜잭션이 있을 경우 ROLLBACK)하고 있는 상태. 클라이언트가 끊기거나 서버에서 세션을 강제로 끊을 때 발생한다. |
| UNKNOWN | 알 수 없는 상태 |

ISOLATION_LEVEL

해당 세션에 설정된 고립 수준 (isolation level)를 나타낸다.

REPLICATION_MODE

세션의 이중화 모드를 나타낸다.

- 0: DEFAULT(이중화)
- 16: NONE

TRANSACTION_MODE

트랜잭션 모드를 나타낸다.

- 0: READ/WRITE
- 4: READ ONLY

COMMIT_WRITE_WAIT_MODE

- 0: commit 시, 로그를 디스크에 기록할 때까지 기다리지 않는다.
- 1: commit 시, 로그를 디스크에 기록할 때까지 기다린다.

OPTIMIZER_MODE

해당 세션에 설정된 최적화 모드를 나타낸다.

- 1: 규칙 기반 (rule based)
- 0: 비용 기반 (cost based)

CURRENT_STMT_ID

현재 수행중인 구문 (statement)의 식별자를 나타낸다.

STACK_SIZE

해당 세션에 설정된 질의 처리기를 위한 스택 크기를 나타낸다.

DEFAULT_DATE_FORMAT

해당 세션에 설정된 기본 날짜 형식을 나타낸다. 1장의 날짜형 데이터 타입을 참조한다.

예)

```
DD-MON-RRRR
```

TRX_UPDATE_MAX_LOGSIZE

하나의 DML에 의해 생성될 수 있는 로그의 최대 크기를 나타낸다.

LOGIN_TIME

클라이언트가 접속한 시간을 나타낸다.

FAILOVER_SOURCE

이 값은 Fail-Over가 일어났을 때, 발생한 Fail-Over의 종류 (CTF 또는 STF)와 접속 서버에 대한 정보를 나타낸다. 여기서 접속 서버 정보란 CTF (Connection Time Failover)일 경우에는 첫 번째로 접속을 시도한 서버의 주소 및 포트 번호이고, STF (Service Time Failover)일 경우에는 연결이 되어 있던 서버의 주소 및 포트 번호이다.

ex) primary 서버가 127.0.0.1:10000이고 alternative 서버가 127.0.0.2:20000일 때:

- 127.0.0.1에 접속을 실패한 후 CTF가 발생하여 127.0.0.2로 접속될 경우, FAILOVER_SOURCE의 값은 다음과 같다: CTF 127.0.0.1:10000
- 127.0.0.2에 접속 중이었으나 오류가 발생하여 127.0.0.1로 STF가 발생한 경우, FAILOVER_SOURCE의 값은 다음과 같다: STF 127.0.0.2:20000

NLS_TERRITORY

현재 연결된 세션의 지역 이름을 표시한다.

NLS_ISO_CURRENCY

현재 연결된 세션의 ISO 통화 기호를 표시한다.

NLS_CURRENCY

현재 연결된 세션의 지역 통화 기호를 표시한다.

NLS_NUMERIC_CHARACTERS

현재 연결된 세션의 소수점 문자와 그룹 구분자를 표시한다.

TIME_ZONE

세션에 설정된 타임 존의 지역이름이나 약어 또는 UTC 오프셋 값이 표시된다.

LOB_CACHE_THRESHOLD

세션에서 LOB_CACHE_THRESHOLD 프로퍼티에 설정된 값을 표시한다. LOB_CACHE_THRESHOLD 프로퍼티에 대해서는 2장을 참고하라.

QUERY_REWRITE_ENABLE

세션에서 QUERY_REWRITE_ENABLE 프로퍼티에 설정된 값을 표시한다. QUERY_REWRITE_ENABLE 프로퍼티에 대해서는 2장을 참고하라.

- FALSE: Altibase 서버에서 쿼리 변환 시에 함수 기반 인덱스 미적용(disable)
- TRUE: Altibase 서버에서 쿼리 변환 시에 함수 기반 인덱스 적용(enable)

DBLINK_GLOBAL_TRANSACTION_LEVEL

세션에서 DBLINK_GLOBAL_TRANSACTION_LEVEL 프로퍼티에 설정된 글로벌 트랜잭션 수행 레벨을 표시한다. DBLINK_GLOBAL_TRANSACTION_LEVEL 프로퍼티에 대해서는 2장을 참고하라.

- 0: remote statement execution level
- 1: simple transaction commit level
- 2: Two-Phase Commit

DBLINK_REMOTE_STATEMENT_AUTOCOMMIT

세션에서 DBLINK_REMOTE_STATEMENT_AUTOCOMMIT 프로퍼티에 설정된 원격 데이터베이스의 AUTOCOMMIT 모드를 표시한다. DBLINK_REMOTE_STATEMENT_AUTOCOMMIT 프로퍼티에 대해서는 2장을 참고하라.

- 0: autocommit-off
- 1: autocommit-on

MAX_STATEMENTS_PER_SESSION

하나의 세션에서 실행할 수 있는 statement의 최대 개수이다. MAX_STATEMENTS_PER_SESSION 프로퍼티의 값을 기본값으로 한다.

SSL_CERTIFICATE_SUBJECT

클라이언트 인증이 설정되지 않은 경우(SSL_CLIENT_AUTHENTICATION 프로퍼티 값이 0)에는 이 정보가 나타나지 않는다.

SSL_CERTIFICATE_ISSUER

클라이언트 인증이 설정되지 않은 경우(SSL_CLIENT_AUTHENTICATION 프로퍼티 값이 0)에는 이 정보가 나타나지 않는다.

CLIENT_INFO

접속된 클라이언트의 애플리케이션 정보이다. 클라이언트 응용프로그램에 의해 설정되는 값이다. SET_CLIENT_INFO() 내장 프로시저를 사용하여 설정할 수 있다.

MODULE

수행중인 프로시저의 모듈이름에 관한 정보이다. SET_MODULE() 내장 프로시저를 사용하여 설정한다.

ACTION

수행중인 프로시저의 모듈이름에 관한 정보이다. SET_MODULE() 내장 프로시저를 사용하여 설정한다.

REPLICATION_DDL_SYNC

이중화 중 DDL 복제 허용 여부를 나타낸다.

- 0: 이중화 중 DDL 복제를 지원하지 않는다.
- 1: 이중화 중 DDL 복제를 지원한다.

REPLICATION_DDL_TIMEOUT

현재 세션의 이중화를 통한 DDL 복제 수행 시간 초과(timeout) 값을 나타낸다.

DDL 복제를 수행하는 지역 서버를 기준으로 초과값 측정된다.

MESSAGE_CALLBACK

접속된 클라이언트의 메시지 콜백 등록 상태를 나타낸다. 메시지 콜백 등록 상태에 따라 서버는 메시지 전송 여부를 결정한다.

- REG
클라이언트는 메시지콜백을 등록하였으며, 서버는 메시지를 클라이언트로 전송한다.
- UNREG
클라이언트는 메시지콜백을 등록하지 않았으며, 서버는 메시지를 클라이언트로 전송하지 않는다.

- UNKNOWN

클라이언트의 메시지콜백 등록 여부를 알 수 없으며, 서버는 메시지를 클라이언트로 전송한다. 해당 기능이 없는 구버전 클라이언트가 접속한 경우 UNKNOWN 상태를 가진다.

V\$SESSION_EVENT

현재 Altibase에 접속중인 세션별로 모든 대기 이벤트들에 대한 통계 정보(누적치)를 보여준다.

| Column name | Type | Description |
|-------------------|--------------|-----------------------------------|
| SID | INTEGER | 세션의 식별자 |
| EVENT | VARCHAR(128) | 대기 이벤트 이름 |
| TOTAL_WAITS | BIGINT | 대기 이벤트에 대한 총 대기 횟수 |
| TOTAL_TIMEOUTS | BIGINT | 지정된 시간 이후에도 요청한 리소스를 획득하는데 실패한 횟수 |
| TIME_WAITED | BIGINT | 대기 이벤트에 대한 총 대기시간 (밀리초) |
| AVERAGE_WAIT | BIGINT | 대기 이벤트에 대한 평균 대기시간 (밀리초) |
| MAX_WAIT | BIGINT | 대기 이벤트에 대한 최대 대기시간 (밀리초) |
| TIME_WAITED_MICRO | BIGINT | 대기 이벤트에 대한 총 대기 시간 (마이크로초) |
| EVENT_ID | INTEGER | 대기 이벤트의 식별자 |
| WAIT_CLASS_ID | INTEGER | 대기 이벤트 클래스의 식별자 |
| WAIT_CLASS | VARCHAR(128) | 대기 이벤트 클래스 이름 |

칼럼 정보

SID

대기하고 있는 세션의 식별자를 나타낸다.

EVENT

대기 이벤트의 이름을 나타낸다.

TOTAL_WAITS

대기 이벤트가 대기하고 있는 총 대기 횟수를 나타낸다.

TOTAL_TIMEOUTS

대기 이벤트가 지정된 시간 이후에도 요청한 리소스를 획득하는데 실패한 횟수를 나타낸다.

TIME_WAITED

대기 이벤트에 대한 총 대기 시간을 나타낸다. (단위: 밀리초)

AVERAGE_WAIT

대기 이벤트에 대한 평균 대기 시간을 나타낸다. (단위: 밀리초)

MAX_WAIT

대기 이벤트에 대한 최대 대기 시간을 나타낸다. (단위: 밀리초)

TIME_WAITED_MICRO

대기 이벤트에 대한 총 대기 시간을 나타낸다. (단위: 마이크로초)

EVENT_ID

대기하고 있는 이벤트의 ID를 나타낸다.

WAIT_CLASS_ID

세션에 대기하고 있는 이벤트의 클래스 ID를 나타낸다.

WAIT_CLASS

세션에 대기하고 있는 이벤트를 그룹화한 클래스의 이름을 나타낸다.

V\$SESSION_WAIT

현재 접속된 모든 세션의 대기 이벤트 정보를 보여준다. 그러나 이전에 접속했던 세션과 관련된 대기 이벤트들의 정보는 제공되지 않는다.

| Column name | Type | Description |
|----------------|--------------|----------------|
| SID | BIGINT | 세션의 ID |
| SEQNUM | INTEGER | 대기 이벤트의 ID |
| EVENT | VARCHAR(128) | 대기 이벤트의 이름 |
| P1 | BIGINT | 대기 이벤트의 파라미터 1 |
| P2 | BIGINT | 대기 이벤트의 파라미터 2 |
| P3 | BIGINT | 대기 이벤트의 파라미터 3 |
| WAIT_CLASS_ID | INTEGER | 대기 클래스의 ID |
| WAIT_CLASS | VARCHAR(128) | 대기 클래스의 이름 |
| WAIT_TIME | BIGINT | 대기시간 (밀리초) |
| SECOND_IN_WAIT | BIGINT | 대기시간 (초) |

칼럼 정보**SID**

현재 접속된 세션의 ID를 나타낸다.

SEQNUM

세션에 대기하고 있는 대기 이벤트의 ID를 나타낸다.

EVENT

세션에 대기하고 있는 이벤트의 이름을 나타낸다.

WAIT_CLASS_ID

대기하고 있는 이벤트의 클래스 ID를 나타낸다.

WAIT_CLASS

대기하고 있는 이벤트를 그룹화한 클래스의 이름을 나타낸다.

WAIT_TIME

해당 이벤트가 대기하고 있는 시간을 나타낸다. (단위: 밀리초)

SECOND_IN_WAIT

해당 이벤트가 대기하고 있는 시간을 나타낸다. (단위: 초)

V\$SESSION_WAIT_CLASS

현재 접속된 모든 세션의 대기 이벤트를 분류하여 대기 정보의 누적된 통계치를 보여준다. 그러나 이전에 접속했던 세션과 관련된 대기 이벤트들의 정보는 제공되지 않는다.

| Column name | Type | Description |
|---------------|--------------|--------------------------------------|
| SID | INTEGER | 세션의 식별자 |
| SERIAL | INTEGER | 대기 이벤트의 ID |
| WAIT_CLASS_ID | INTEGER | 대기 클래스의 ID |
| WAIT_CLASS | VARCHAR(128) | 대기 클래스의 이름 |
| TOTAL_WAITS | BIGINT | 세션에서 이 대기 이벤트를 기다린 총 횟수 |
| TIME_WAITED | DOUBLE | 세션에서 이 대기 이벤트를 기다리는데 소요된 전체 시간 (밀리초) |

칼럼 정보**SID**

세션의 식별자이다.

SERIAL

대기 이벤트의 식별자이다.

WAIT_CLASS_ID

대기하고 있는 이벤트의 클래스 ID를 나타낸다.

WAIT_CLASS

대기하고 있는 이벤트를 그룹화한 클래스의 이름을 나타낸다.

TOTAL_WAITS

세션에서 이 대기 이벤트를 기다린 총 횟수이다.

TIME_WAITED

세션에서 이 대기 이벤트를 기다리는데 소요된 전체 시간 (단위: 밀리초)

예제

<예제1> 다음의 SELECT 쿼리는 각 세션별로 대기 이벤트를 기다린 총 횟수와 대기에 소요된 전체 시간을 세션, 대기 이벤트 및 대기 클래스로 분류하여 출력한다.

```
select sid, serial, wait_class_id, sum(total_waits), sum(time_waited)
from v$session_wait_class
group by sid, serial, wait_class_id
order by total_waits desc;
```

V\$SESSIONMGR

세션 통계 정보를 보여준다.

| Column name | Type | Description |
|-------------------------|---------|-------------|
| TASK_COUNT | INTEGER | 연결된 세션 개수 |
| BASE_TIME | INTEGER | 현재 시간 |
| LOGIN_TIMEOUT_COUNT | INTEGER | 아래 참조 |
| IDLE_TIMEOUT_COUNT | INTEGER | 아래 참조 |
| QUERY_TIMEOUT_COUNT | INTEGER | 아래 참조 |
| DDL_TIMEOUT_COUNT | INTEGER | 아래 참조 |
| FETCH_TIMEOUT_COUNT | INTEGER | 아래 참조 |
| UTRANS_TIMEOUT_COUNT | INTEGER | 아래 참조 |
| SESSION_TERMINATE_COUNT | INTEGER | 아래 참조 |

칼럼 정보**TASK_COUNT**

현재 접속된 세션의 총 개수를 나타낸다.

BASE_TIME

Altibase 서버가 유지하고 있는 현재 시각을 시스템 시간 (초)로 나타낸다.

LOGIN_TIMEOUT_COUNT

Altibase가 구동된 이후에 발생한 로그인 타임 아웃 횟수를 나타낸다.

IDLE_TIMEOUT_COUNT

Altibase가 구동된 이후에 발생한 Idle 시간 초과 횟수를 나타낸다.

DDL_TIMEOUT_COUNT

Altibase가 구동된 이후에 발생한 DDL문 수행 시간 초과 횟수를 나타낸다.

QUERY_TIMEOUT_COUNT

Altibase가 구동된 이후에 발생한 쿼리 시간 초과 횟수를 나타낸다.

FETCH_TIMEOUT_COUNT

Altibase가 구동된 이후에 발생한 Fetch 시간 초과 횟수를 나타낸다.

UTRANS_TIMEOUT_COUNT

Altibase가 구동된 이후에 발생한 갱신(Update) 트랜잭션의 시간 초과 횟수를 나타낸다.

SESSION_TERMINATE_COUNT

Altibase가 구동된 이후에 sysdba에 의해 강제로 연결이 끊긴 세션의 개수를 나타낸다.

V\$SESSTAT

현재 접속된 모든 세션의 통계치를 나타낸다.

| Column name | Type | Description |
|-------------|--------------|-------------|
| SID | INTEGER | 세션 식별자 |
| SEQNUM | INTEGER | 통계 일련 번호 |
| NAME | VARCHAR(128) | 통계 이름 |
| VALUE | BIGINT | 통계 값 |

각 상태에 대한 설명은 V\$STATNAME을 참조한다.

칼럼 정보**SID**

세션의 고유 아이디를 나타낸다.

SEQNUM

통계 식별을 위한 일련 번호이다.

NAME

통계 이름을 나타낸다.

VALUE

통계치로 반환된 값을 64비트 정수로 나타낸다.

V\$SFLUSHER

보조 버퍼(Secondary Buffer)의 페이지를 디스크에 플러시 하는 작업에 대한 정보를 보여준다.

| Column name | Type | Description |
|------------------------|---------|--|
| ID | INTEGER | Flusher 식별자 |
| ALIVE | INTEGER | Flusher가 현재 활동 중인지 여부 |
| CURRENT_JOB | INTEGER | 현재 작업 1: 교체 플러시 중 2: 체크포인트 플러시 중 3: 객체 플러시 중 |
| DOING_IO | INTEGER | Flusher가 디스크 I/O 수행 중인지 여부 |
| INIOB_COUNT | INTEGER | 플러시되는 내용을 그 안에 저장하기 위해 내부 버퍼에 직접 접근한 횟수 |
| REPLACE_FLUSH_JOBS | BIGINT | 완료된 교체 플러시 작업의 누적 횟수 |
| REPLACE_FLUSH_PAGES | BIGINT | 교체 플러시로 디스크에 쓰여진 페이지의 누적 개수 |
| REPLACE_SKIP_PAGES | BIGINT | 교체 플러시 중에 플러시가 취소된 페이지의 누적 개수 |
| CHECKPOINT_FLUSH_JOBS | BIGINT | 완료된 체크포인트 플러시 작업의 누적 횟수 |
| CHECKPOINT_FLUSH_PAGES | BIGINT | 체크포인트 플러시로 디스크에 쓰여진 페이지의 누적 개수 |
| CHECKPOINT_SKIP_PAGES | BIGINT | 체크포인트 플러시 중에 플러시가 취소된 페이지의 누적 개수 |
| OBJECT_FLUSH_JOBS | BIGINT | 객체 플러시가 수행된 누적 횟수 |
| OBJECT_FLUSH_PAGES | BIGINT | 객체 플러시로 디스크에 쓰여진 페이지의 누적 개수 |
| OBJECT_SKIP_PAGES | BIGINT | 객체 플러시 중에 플러시가 취소된 페이지의 누적 개수 |
| LAST_SLEEP_SEC | INTEGER | 작업이 모두 완료된 후 Flusher가 잠들어 있던 시간의 길이 |
| TIMEOUT | BIGINT | 작업 유무를 확인하기 위해서 잠든 Flusher가 깨어난 횟수 |
| SIGNALLED | BIGINT | Altibase 서버로부터의 시그널에 의해 Flusher가 깨어난 횟수 |
| TOTAL_SLEEP_SEC | BIGINT | Flusher가 잠들어 있던 총 시간 |
| TOTAL_FLUSH_PAGES | BIGINT | 플러시된 페이지의 누적 개수 |

| Column name | Type | Description |
|------------------------|--------|---------------------------------------|
| TOTAL_DW_USEC | BIGINT | DoubleWrite 버퍼의 내용을 디스크로 쓰는데 걸린 누적 시간 |
| TOTAL_WRITE_USEC | BIGINT | 데이터 페이지를 데이터 파일에 쓰는데 걸린 누적 시간 |
| TOTAL_SYNC_USEC | BIGINT | 데이터 페이지를 디스크로 강제 플러시하는데 걸린 누적 시간 |
| TOTAL_FLUSH_TEMP_PAGES | BIGINT | 플러시된 임시 페이지의 누적 개수 |
| TOTAL_TEMP_WRITE_USEC | BIGINT | 임시 페이지를 임시 파일에 쓰는데 걸린 누적 시간 |
| DB_WRITE_PERF | DOUBLE | 데이터 페이지를 데이터 파일에 쓸 때 초당 기록한 평균 바이트 수 |
| TEMP_WRITE_PERF | DOUBLE | 임시 페이지를 임시 파일에 쓸 때 초당 기록한 평균 바이트 수 |

칼럼 정보

ID

Flusher 식별자이다. 식별자는 중복되지 않는다.

ALIVE

Flusher 가 현재 동작 중인지 여부를 나타낸다. 각 Flusher는 DCL구문으로 시작하거나 중지할 수 있다.

CURRENT_JOB

Flusher가 현재 수행중인 작업의 유형을 나타낸다.

- 1: 교체 플러시 수행 중임을 가리킨다. 교체 플러시의 목적은 오랜 시간 접근되지 않은 버퍼를 플러시하여 교체 가능하도록 하는 데 있다.
- 2: 체크포인트 플러시 수행 중임을 가리킨다. 체크포인트 플러시의 목적은 가장 오래 전에 갱신된 버퍼를 플러시하여 체크포인트 시간을 줄이는 데 있다.
- 3: 인덱스, 테이블, 세그먼트 등의 특정 객체를 플러시하고 있음을 가리킨다.

DOING_IO

Flusher가 현재 자신의 업무 수행을 위해서 디스크 I/O 작업 중인지 여부를 나타낸다.

INIOB_COUNT

Flusher는 페이지를 디스크에 기록하기 위해서, 그 내용을 내부 버퍼 (IOB)에 저장한다. 이 값은 그 내부 버퍼에 플러시할 내용을 저장하기 위해 접근한 횟수를 가리킨다.

REPLACE_FLUSH_JOBS

교체 플러시 작업을 수행한 누적 횟수이다.

REPLACE_FLUSH_PAGES

교체 플러시 작업에 의해 디스크에 쓰여진 페이지의 누적 개수이다.

REPLACE_SKIP_PAGES

교체 플러시 중에 정책 또는 효율의 이유로 인해서 플러시 작업이 취소된 페이지의 누적 개수이다.

CHECKPOINT_FLUSH_JOBS

체크포인트 플러시 작업을 수행한 누적 횟수이다.

CHECKPOINT_FLUSH_PAGES

체크포인트 플러시 작업에 의해 디스크에 쓰여진 페이지의 누적 개수이다.

CHECKPOINT_SKIP_PAGES

체크포인트 플러시 중에 정책 또는 효율의 이유로 인해서 플러시가 취소된 페이지의 누적 개수이다.

OBJECT_FLUSH_JOBS

객체 플러시 작업을 수행한 누적 횟수이다.

OBJECT_FLUSH_PAGES

객체 플러시 작업에 의해 디스크에 쓰여진 페이지의 누적 개수이다.

OBJECT_SKIP_PAGES

객체 플러시 중에 정책 또는 효율의 이유로 인해서 플러시가 취소된 페이지의 누적 개수이다.

LAST_SLEEP_SEC

가장 최근에 모든 작업을 완료한 Flusher가 더 이상 작업이 없어서 잠들어 있던 시간의 길이이다.

TIMEOUT

작업이 없어서 잠들어 있던 Flusher가 작업 유무를 확인하기 위해서 일정 간격으로 깨어나야 할 필요가 있다. 이 값은 깨어난 누적 횟수다.

SIGNALLED

어떤 작업의 빠른 처리를 위해서 Altibase는 잠든 Flusher에게 시그널을 주어서 깨울 수 있다. 이 값은 그 시그널에 의해 Flusher가 깨어난 횟수이다.

TOTAL_SLEEP_SEC

Flusher가 처리할 작업이 없어서 잠든 상태로 대기하고 있었던 시간의 총 합이다.

TOTAL_FLUSH_PAGES

체크포인트 플러시, 교체 플러시, 또는 객체 플러시에 의해 플러시된 페이지의 누적 개수이다.

TOTAL_DW_USEC

이 값은 doublewrite 버퍼의 내용을 디스크로 쓰는 데 걸린 시간의 누적 값이다. Doublewrite란 페이지들을 데이터 파일에 쓰기 전에, doublewrite buffer라 불리는 DW 파일에 먼저 기록하는 것을 말한다. Doublewrite buffer에 일단 기록된 후에, 그 페이지들은 데이터 파일의 올바른 위치에 다시 기록된다. 페이지를 데이터 파일에 기록하는 중에 운영 체제가 멈추거나 이들 데이터 파일이 손상된다면, 이 doublewrite 버퍼의 손상되지 않은 페이지를 이용해서 복구가 가능하다.

TOTAL_WRITE_USEC

데이터 페이지를 데이터 파일에 쓰는데 걸린 시간의 누적 값이다. 이 값은 디스크에 플러시 하는데 걸린 시간은 포함하지 않는다.

TOTAL_SYNC_USEC

데이터 페이지를 데이터 파일에 강제로 플러시 하는데 소요된 시간의 누적 값이다.

TOTAL_FLUSH_TEMP_PAGES

플러시된 임시 페이지들의 누적 개수이다. (임시 페이지는 Sort 연산과 hash join을 할 때 사용되는 임시 테이블을 저장하는 데이터 페이지이다.)

TOTAL_TEMP_WRITE_USEC

임시 페이지들을 임시 파일에 기록하는데 걸린 시간의 누적 값이다.

DB_WRITE_PERF

데이터 페이지를 데이터 파일에 쓸 때 초당 기록된 bytes 수의 평균값으로 단위는 KB/Sec이다.

TEMP_WRITE_PERF

임시 페이지를 임시 파일에 쓸 때 초당 기록된 bytes 수의 평균값으로 단위는 KB/Sec이다.

V\$SFLUSHINFO

보조 버퍼(Secondary Buffer)의 플러시 정보를 보여준다.

| Column name | Type | Description |
|-----------------------|---------|---|
| FLUSHER_COUNT | INTEGER | 플러시할 페이지의 개수 |
| CHECKPOINT_LIST_COUNT | INTEGER | 체크포인트 리스트 개수 |
| REQ_JOB_COUNT | INTEGER | 현재 플러시 관리자에 등록된 작업의 개수 |
| REPLACE_PAGES | INTEGER | 교체 플러시로 플러시할 페이지의 개수 |
| CHECKPOINT_PAGES | INTEGER | 체크포인트 플러시할 페이지의 개수 |
| MIN_BCB_ID | INTEGER | 체크포인트 대상 페이지 중 가장 빠른 recoveryLSN을 가진 페이지에 대응하는 BCB 식별자 |
| MIN_SPACEID | INTEGER | 체크포인트 대상 페이지 중 가장 빠른 recoveryLSN을 가진 페이지가 속해 있는 테이블스페이스의 ID |
| MIN_PAGEID | INTEGER | 체크포인트 대상 페이지 중 가장 빠른 recoveryLSN을 가진 페이지의 ID |

칼럼 정보**FLUSHER_COUNT**

보조 버퍼에서 디스크로 플러시할 페이지의 개수를 나타낸다.

CHECKPOINT_LIST_COUNT

체크포인트 리스트 개수를 나타낸다.

REQ_JOB_COUNT

이는 플러시 관리자에 등록된 작업의 개수이다.

REPLACE_PAGES

교체 플러시 작업에 의해 보조 버퍼에서 디스크에 플러시될 페이지의 개수를 나타낸다.

CHECKPOINT_PAGES

보조 버퍼에서 디스크로 체크포인트 플러시할 페이지의 개수를 나타낸다.

MIN_BCB_ID

체크포인트 대상 페이지 중 가장 빠른 recovery LSN을 가진 페이지에 대응하는 BCB 식별자를 나타낸다.

MIN_SPACEID

체크포인트 대상 페이지 중 가장 빠른 recovery LSN을 가진 페이지가 속해 있는 테이블스페이스의 ID를 나타낸다.

MIN_PAGEID

체크포인트 대상 페이지 중 가장 빠른 recovery LSN을 가진 페이지의 ID를 나타낸다.

V\$SNAPSHOT

스냅샷(SNAPSHOT)의 설정 상태와 메모리, 디스크 연두 테이블스페이스의 사용량을 보여준다.

| Column name | Type | Description |
|-------------------------|---------|---|
| SCN | BIGINT | 스냅샷에 설정된 SNAPSHOT SCN 값 |
| BEGIN_TIME | BIGINT | 스냅샷 설정 시의 UNIX_TIME |
| BEGIN_MEM_USAGE | INTEGER | 스냅샷 설정시의 메모리 사용 비율 |
| BEGIN_DISK_UNDO_USAGE | INTEGER | 스냅샷 설정 시의 BEGIN 시의 디스크 연두 테이블스페이스 사용 비율 |
| CURRENT_TIME | BIGINT | 현재 시간의 UNIX_TIME |
| CURRENT_MEM_USAGE | INTEGER | 현재 메모리 사용 비율 |
| CURRENT_DISK_UNDO_USAGE | INTEGER | 현재 디스크 연두 테이블스페이스 사용 비율 |

칼럼 정보**SCN**

BEGIN SNAPSHOT 시에 설정된 SCN 값을 나타낸다. iLoader가 이 SCN을 기준으로 데이터를 EXPORT한다.

BEGIN_TIME

BEGIN SNAPSHOT 구문이 실행될 때의 시간을 UNIX_TIME으로 나타낸다.

BEGIN_MEM_USAGE

BEGIN SNAPSHOT 구문이 실행될 때의 메모리 사용량을 백분율로 나타낸다.

BEGIN_DISK_UNDO_USAGE

BEGIN SNAPSHOT 구문이 실행될 때의 디스크 언두 테이블스페이스의 사용량을 백분율로 나타낸다.

CURRENT_TIME

현재 시간을 UNIX_TIME으로 나타낸다

CURRENT_MEM_USAGE

현재 메모리 사용량을 백분율로 나타낸다.

CURRENT_DISK_UNDO_USAGE

현재 디스크 언두 테이블스페이스의 사용량을 백분율로 나타낸다.

V\$SQLTEXT

서버에서 현재 수행되는 SQL 텍스트 정보를 나타낸다.

| Column name | Type | Description |
|-------------|-------------|----------------|
| SID | INTEGER | 세션 식별자 |
| STMT_ID | INTEGER | statement 식별자 |
| PIECE | INTEGER | 텍스트 조각의 일련 번호 |
| TEXT | VARCHAR(64) | SQL 텍스트 문자열 조각 |

칼럼 정보**SID**

SQL 텍스트가 실행된 세션의 고유 번호를 나타낸다.

STMT_ID

세션에서 실행된 SQL 구문 (statement)의 식별자이다.

PIECE

실행되는 전체 SQL 문을 64바이트 단위의 문자열 조각으로 나누어 저장한다. PIECE는 각 조각의 일련 번호로 0부터 시작된다.

TEXT

전체 SQL 문의 일부분인 64바이트 단위의 문자열 조각이다.

V\$SQL_PLAN_CACHE

SQL Plan Cache의 현재 상태 및 통계 정보를 나타낸다.

| Column name | Type | Description |
|--------------------------|---------|--|
| MAX_CACHE_SIZE | BIGINT | SQL Plan Cache의 최대 크기 (bytes) |
| CURRENT_HOT_LRU_SIZE | BIGINT | LRU 리스트에서 현재 HOT 영역의 크기 |
| CURRENT_COLD_LRU_SIZE | BIGINT | LRU 리스트에서 현재 COLD 영역의 크기 |
| CURRENT_CACHE_SIZE | BIGINT | 현재 SQL Plan Cache의 크기 (bytes) |
| CURRENT_CACHE_OBJ_COUNT | INTEGER | 현재 SQL Plan Cache에 등록된 PCO 수 |
| CACHE_HIT_COUNT | BIGINT | SQL Plan Cache에 등록된 PCO의 활용 횟수 |
| CACHE_MISS_COUNT | BIGINT | SQL Plan Cache에서 plan 검색과정에서 PCO를 찾지 못한 횟수 |
| CACHE_IN_FAIL_COUNT | BIGINT | SQL Plan Cache에 새로운 PCO 삽입 시 cache 최대 크기 제약으로 실패한 횟수 |
| CACHE_OUT_COUNT | BIGINT | SQL Plan Cache에서 제거된 PCO의 개수 |
| CACHE_INSERTED_COUNT | BIGINT | SQL Plan Cache에 추가된 PCO의 개수 |
| NONE_CACHE_SQL_TRY_COUNT | BIGINT | DDL과 DCL 등의 Cache 비대상 구문의 시도 횟수 |

칼럼 정보

MAX_CACHE_SIZE

SQL Plan Cache의 최대 크기이다. SQL Plan Cache의 최대 크기를 줄이거나 늘리기 위해서는 'alter system set SQL_PLAN_CACHE_SIZE = ' 구문을 실행한다.

CURRENT_HOT_LRU_SIZE

SQL Plan Cache의 LRU 리스트 중에서 빈번하게 참조되는 PCO는 HOT 영역에서 관리되는데, 그 크기 (byte)를 나타낸다.

CURRENT_COLD_LRU_SIZE

SQL Plan Cache의 LRU 리스트 중 자주 참조되지 않은 PCO는 COLD 영역에서 관리되는데, 그 크기(byte)를 나타낸다.

CURRENT_CACHE_SIZE

SQL Plan Cache에 현재 삽입된 PCO들의 전체 크기(byte)를 나타낸다.

CURRENT_CACHE_OBJ_COUNT

SQL Plan Cache에 삽입된 PCO들의 수를 나타낸다.

CACHE_HIT_COUNT

SQL Plan Cache에 삽입된 PCO들이 사용된 전체 횟수를 나타낸다.

CACHE_MISS_COUNT

SQL Plan Cache에 없는 PCO 참조 시도 횟수를 나타낸다.

CACHE_IN_FAIL_COUNT

Cache의 최대 메모리 크기 제약으로 인해 현재 참조하지 않는 PCO들을 찾아 cache에서 삭제 및 해제 시도를 수행했음에도 불구하고, PCO를 삽입하지 못한 횟수이다.

CACHE_OUT_COUNT

SQL Plan Cache에 추가되었다가 삭제된 PCO의 개수를 의미한다.

CACHE_INSERTED_COUNT

SQL Plan Cache에 추가된 PCO의 개수를 의미한다.

NONE_CACHE_SQL_TRY_COUNT

SQL Plan Cache에 저장되지 않는 구문이 발생한 횟수이다. 그 구문은 DDL과 DCL 구문이다.

V\$SQL_PLAN_CACHE_PCO

SQL Plan Cache에 등록된 PCO에 대한 정보를 나타낸다.

PCO는 SQL 문장, 실행 계획, Plan Environment 정보를 가진 객체로, SQL 문 수행 시 필요한 실행 계획을 세션 간 공유하여 질의 성능을 향상시키는 효과를 가진다.

PCO는 Parent PCO와 Child PCO로 구분된다.

Parent PCO

SQL 문장과 SQL 문장을 비교, 관리하기 위한 정보를 가진 PCO이다. Parent PCO는 서로 다른 SQL 문장마다 하나씩 존재한다.

Child PCO

실행 계획에 영향을 미치는 요소인 Plan Environment를 비교하기 위해 관리하는 PCO이다. 동일한 SQL 문장이라도 사용자, NLS(National Language Support), 통계정보와 같은 Plan Environment에 따라 서로 다른 실행 계획이 생성될 수 있다. Child PCO는 PCO 생성 당시의 Plan Environment와 실행 계획, 실행 계획의 크기 정보를 저장한다. 반드시 Parent PCO를 가지며 하나의 Parent PCO는 여러 Child PCO를 가질 수 있다.

| Column name | Type | Description |
|---------------|-------------|------------------|
| SQL_TEXT_ID | VARCHAR(64) | Parent PCO 식별자 |
| PCO_ID | INTEGER | Child PCO 식별자 |
| CREATE_REASON | VARCHAR(28) | PCO를 생성한 이유 |
| HIT_COUNT | INTEGER | PCO 참조 횟수 |
| REBUILD_COUNT | INTEGER | PCO가 rebuild된 횟수 |
| PLAN_STATE | VARCHAR(17) | PCO의 plan 상태 |

| Column name | Type | Description |
|-----------------|-------------|-------------------------|
| LRU_REGION | VARCHAR(11) | LRU 리스트에서 PCO가 속해 있는 영역 |
| PLAN_SIZE | INTEGER | PCO의 plan 크기 |
| FIX_COUNT | INTEGER | PCO를 참조 중인 Statement 수 |
| PLAN_CACHE_KEEP | VARCHAR(6) | PCO의 Keep 상태 |

칼럼 정보

SQL_TEXT_ID

Parent PCO의 식별자이다.

PCO_ID

Child PCO의 식별자이다.

CREATE_REASON

PCO를 생성한 이유이며 다음과 같은 값이 올 수 있다.

- CREATE_BY_CACHE_MISS
SQL Plan cache에 필요한 PCO가 없어서 생성한 경우
- CREATE_BY_PLAN_INVALIDATION
prepare 과정에서 SQL Plan Cache에서 PCO를 찾았지만, Plan에서 참조한 데이터베이스 객체가 유효 상태가 아니어서 새로 생성한 경우
- CREATE_BY_PLAN_TOO_OLD
execute 과정에서 Plan에서 참조한 객체의 통계 정보의 변경폭이 한계치를 넘었거나, DDL이 발생하여 새로 PCO를 생성한 경우

HIT_COUNT

PCO의 참조 횟수를 나타낸다.

REBUILD_COUNT

PCO의 plan이 다시 컴파일된 횟수를 나타낸다.

PLAN_STATE

PCO의 plan 상태를 나타내며, 다음과 같은 값을 가질 수 있다.

- READY
PCO에 SQL 문장, 실행 계획(Execution Plan) 및 Plan Environment 가 모두 할당되어 있는 상태
- OLD_PLAN
Plan이 유효한 상태가 아니어서 앞으로 사용되지 않는 plan 상태

LRU_REGION

Hot-Cold LRU 리스트는 PCO의 교체 정책을 관리하는 자료 구조이다. SQL Plan Cache는 Altibase 서버 프로퍼티 SQL_PLAN_CACHE_SIZE에 의해 크기가 정해져있어 제한된 수의 PCO가 등록된다. 이 칼럼은 PCO가 Hot-Cold LRU 리스트에서 어느 영역에 속해 있는지를 보여준다.

- HOT_REGION

사용 빈도가 많은 PCO

- COLD_REGION

사용 빈도가 적은 PCO

PLAN_SIZE

PCO의 plan 크기를 나타낸다.

FIX_COUNT

PCO를 참조 중인 statement 수를 나타낸다. FIX_COUNT가 1 이상이면 victim에 선정되지 않는다.

PLAN_CACHE_KEEP

PCO의 keep 상태를 나타내며 다음과 같은 값을 가질 수 있다.

- KEEP
Plan이 keep 되어 있는 상태로 victim에 선정되지 않는다.
- UNKEEP
PLAN이 unkeep 되어 있는 상태로 victim에 선정될 수 있다.

V\$SQL_PLAN_CACHE_SQLTEXT

[Parent PCO](#)에 대한 정보를 보여준다.

| Column name | Type | Description |
|------------------------|----------------|--------------------------------------|
| SQL_TEXT_ID | VARCHAR(64) | Parent PCO 식별자 |
| SQL_TEXT | VARCHAR(16384) | SQL 문장 |
| CHILD_PCO_COUNT | INTEGER | Parent PCO가 현재 가지고 있는 Child PCO의 개수 |
| CHILD_PCO_CREATE_COUNT | INTEGER | Parent PCO 내에 지금까지 생성된 Child PCO의 개수 |
| PLAN_CACHE_KEEP | VARCHAR(6) | SQL_TEXT_ID에 해당하는 PCO의 Keep 상태 |

칼럼 정보

SQL_TEXT_ID

Parent PCO의 식별자이다. 앞의 4자리 숫자는 Parent PCO가 저장된 bucket 의 번호를 나타내며, 나머지 숫자는 그 bucket 내에서 SQL 문장의 일련번호를 나타낸다.

SQL_TEXT

SQL 문장을 나타낸다.

CHILD_PCO_COUNT

Parent PCO가 현재 가지고 있는 Child PCO의 수이다.

CHILD_PCO_CREATE_COUNT

Parent PCO 내에 지금까지 생성된 Child PCO의 개수이다. Parent PCO 내에 Child PCO가 생성되는 경우는 다음의 2가지이다.

- 기존 PCO 중 하나와 SQL문장은 같지만 Plan을 생성한 환경이 맞지 않아서 새로운 Child PCO를 생성한다.
- 기존 PCO가 참조하는 객체의 변경 또는 객체의 통계 정보의 변경 폭이 한계치를 넘는 경우 새로운 Child PCO를 생성한다.

PLAN_CACHE_KEEP

SQL_TEXT_ID에 해당하는 plan cache 객체의 keep 상태를 나타내며 다음과 같은 값을 가질 수 있다.

- KEEP
PLAN이 keep 되어 있는 상태로 victim에 선정되지 않는다.
- UNKEEP
PLAN이 unkeep 되어 있는 상태로 victim에 선정될 수 있다.

V\$STABLE_MEM_DATAFILES

데이터베이스에 존재하는 데이터 파일의 전체 경로를 보여준다.

| Column name | Type | Description |
|---------------|----------------|---------------|
| MEM_DATA_FILE | VARCHAR(4096) | 데이터 파일의 전체 경로 |

칼럼 정보

MEM_DATA_FILE

데이터베이스에 존재하는 데이터 파일의 전체 경로이다.

V\$STATEMENT

현재 연결된 세션 별로 가장 최근 실행된 구문 (statement)에 대한 정보를 보여준다.

| Column name | Type | Description |
|-----------------------|----------------|-------------------|
| ID | INTEGER | 구문 식별자 |
| PARENT_ID | INTEGER | 부모 구문의 식별자 |
| CURSOR_TYPE | INTEGER | 커서의 종류 |
| SESSION_ID | INTEGER | 해당 구문이 속한 세션의 아이디 |
| TX_ID | BIGINT | 트랜잭션 식별자 |
| QUERY | VARCHAR(16384) | 수행된 SQL 스트링 |
| LAST_QUERY_START_TIME | INTEGER | 가장 최근의 쿼리 시작 시간 |
| QUERY_START_TIME | INTEGER | 현재 쿼리 시작 시간 |
| FETCH_START_TIME | INTEGER | 현재 Fetch 시작 시간 |

| Column name | Type | Description |
|-----------------------|-------------|--|
| EXECUTE_STATE | VARCHAR(8) | 현재 Statement의 상태 |
| FETCH_STATE | VARCHAR(12) | 현재 Statement의 FETCH 상태 |
| ARRAY_FLAG | INTEGER | Array 수행 플래그 |
| ROW_NUMBER | INTEGER | 현재 처리중인 행의 번호 |
| EXECUTE_FLAG | INTEGER | 수행 여부 플래그 |
| BEGIN_FLAG | INTEGER | 구문의 시작 여부 |
| TOTAL_TIME | BIGINT | 총 경과 시간 |
| PARSE_TIME | BIGINT | 파싱 소요 시간 |
| VALIDATE_TIME | BIGINT | 정당성 검사 소요 시간 |
| OPTIMIZE_TIME | BIGINT | 최적화 소요 시간 |
| EXECUTE_TIME | BIGINT | 실행 소요 시간 |
| FETCH_TIME | BIGINT | Fetch 소요 시간 |
| SOFT_PREPARE_TIME | BIGINT | Prepare 과정에서 SQL Plan Cache에서 plan 탐색 시간 |
| SQL_CACHE_TEXT_ID | VARCHAR(64) | Parent PCO 식별자 또는 NO_SQL_CACHE_STMT |
| SQL_CACHE_PCO_ID | INTEGER | Child PCO 식별자 |
| OPTIMIZER | BIGINT | 최적화 모드 |
| COST | BIGINT | 최적화 비용 |
| USED_MEMORY | BIGINT | 향후 확장 예정 |
| READ_PAGE | BIGINT | 읽은 디스크 페이지의 개수 |
| WRITE_PAGE | BIGINT | 기록한 디스크 페이지의 개수 |
| GET_PAGE | BIGINT | 접근한 디스크 페이지의 개수 |
| CREATE_PAGE | BIGINT | 생성된 디스크 페이지의 개수 |
| UNDO_READ_PAGE | BIGINT | 읽은 UNDO 디스크 페이지의 개수 |
| UNDO_WRITE_PAGE | BIGINT | 기록한 UNDO 디스크 페이지의 개수 |
| UNDO_GET_PAGE | BIGINT | 접근한 UNDO 디스크 페이지의 개수 |
| UNDO_CREATE_PAGE | BIGINT | 생성한 UNDO 디스크 페이지의 개수 |
| MEM_CURSOR_FULL_SCAN | BIGINT | 인덱스 없이 메모리 테이블을 검색한 횟수 |
| MEM_CURSOR_INDEX_SCAN | BIGINT | 인덱스를 사용해서 메모리 테이블을 검색한 횟수 |

| Column name | Type | Description |
|---------------------------|--------------|---------------------------------------|
| DISK_CURSOR_FULL_SCAN | BIGINT | 인덱스 없이 디스크 테이블을 검색한 횟수 |
| DISK_CURSOR_INDEX_SCAN | BIGINT | 인덱스를 사용해서 디스크 테이블을 검색한 횟수 |
| EXECUTE_SUCCESS | BIGINT | 구문 실행 성공 횟수 |
| EXECUTE_FAILURE | BIGINT | 구문 실행 실패 횟수 |
| FETCH_SUCCESS | BIGINT | Fetch 성공 횟수 |
| FETCH_FAILURE | BIGINT | Fetch 실패 횟수 |
| PROCESS_ROW | BIGINT | 처리된 레코드 개수 |
| MEMORY_TABLE_ACCESS_COUNT | BIGINT | 검색 대상 메모리 테이블에서 이 구문이 검색하는 레코드의 개수 |
| SEQNUM | INTEGER | 대기 이벤트 식별자 |
| EVENT | VARCHAR(128) | 대기 이벤트 이름 |
| P1 | BIGINT | 대기 이벤트 파라미터 1 |
| P2 | BIGINT | 대기 이벤트 파라미터 2 |
| P3 | BIGINT | 대기 이벤트 파라미터 3 |
| WAIT_TIME | BIGINT | 대기 시간 (단위: 밀리초) |
| SECOND_IN_TIME | BIGINT | 대기 시간 (단위: 초) |
| SIMPLE_QUERY | INTEGER | SIMPLE QUERY 여부 |
| MATHEMATICS_TEMP_MEMORY | BIGINT | 분석 함수에서 사용하는 MATHEMATICS TEMP 메모리 사용량 |

칼럼 정보

ID

해당 구문이 가지고 있는 세션 내부에서 구별되는 유일한 식별자이다.

PARENT_ID

해당 구문의 부모 구문 식별자이다.

CURSOR_TYPE

16진수 값 0x02 는 메모리 커서를 가리키고, 16진수 값 0x04 는 디스크 커서를 가리킨다.

SESSION_ID

해당 구문이 속한 세션의 식별자이다.

TX_ID

현재 수행중인 트랜잭션의 식별자이다.

QUERY

현재 구문이 수행하고 있거나 수행했던 쿼리 문자열을 나타낸다.

LAST_QUERY_START_TIME

마지막으로 수행된 쿼리의 시작 시간을 시스템 시간(초)으로 나타낸다.

QUERY_START_TIME

현재 수행중인 구문이 쿼리를 시작한 시간을 시스템 시간(초)으로 나타낸다.

FETCH_START_TIME

현재 구문이 SELECT일 경우 fetch가 시작된 시간을 시스템 시간(초)으로 나타낸다.

EXECUTE_STATE

현재 statement의 상태를 나타내며, 다음과 같은 값을 갖는다.

- ALLOC: 해당 구문이 할당된 상태
- PREPARED: 해당 구문이 PREPARE 된 상태
- EXECUTED: 구문의 EXECUTE가 끝난 상태
- UNKNOWN: 알 수 없는 상태

FETCH_STATE

구문(statement)의 fetch 상태를 나타내며, 다음과 같은 값을 갖는다.

- PROCEED: FETCH 진행 중
- CLOSE: FETCH 종료
- NO_RESULTSET: 결과집합을 생성하지 않는 구문
- INVALIDATED: 유효하지 않은 상태
- UNKNOWN: 알 수 없는 상태

ARRAY_FLAG

현재 statement가 array 또는 batch 모드로 수행중인지 여부를 나타내며, 다음과 같은 값을 갖는다.

- 0: Array나 batch 모드가 아님
- 1: Array나 batch 모드로 수행중임

ROW_NUMBER

Array 또는 batch 모드로 수행시 현재 처리중인 행의 번호를 나타내며, 1번부터 시작한다.

EXECUTE_FLAG

현재 statement가 수행중인지 여부를 나타내며 다음과 같은 값을 갖는다.

- 0: 현재 수행중이 아님
- 1: 현재 수행중임

BEGIN_FLAG

현재 statement가 시작되었는지 여부를 나타낸다.

- 0: 현재 구문이 시작되지 않았거나, 종료되었음
- 1: 현재 구문이 시작됨

TOTAL_TIME

현재 구문의 총 수행시간을 나타내며 단위는 마이크로 초이다.

해당 구문의 종류에 따라 EXECUTE_TIME에 PVO 시간 또는 Fetch 시간이 추가될 수 있다.

PARSE_TIME

쿼리의 구문 검사 시간을 마이크로 초 단위로 나타낸다.

VALIDATE_TIME

쿼리의 의미 검사 시간을 마이크로 초 단위로 나타낸다.

OPTIMIZE_TIME

쿼리의 최적화 수행 시간을 마이크로 초 단위로 나타낸다.

EXECUTE_TIME

쿼리의 순수 실행 시간을 마이크로 초 단위로 나타낸다. Select의 경우에는 첫번째 Fetch가 일어나기 전까지의 수행시간을 나타낸다.

FETCH_TIME

SELECT 쿼리의 경우 fetch 소요 시간을 마이크로 초 단위로 나타낸다.

SOFT_PREPARE_TIME

Prepare 과정에서 SQL 문장과 plan 생성시 필요한 각종 변수들을 이용하여 SQL Plan Cache에서 이에 부합하는 plan 객체를 찾는데 소요된 시간을 나타낸다. (단위: 마이크로 초)

SQL_CACHE_TEXT_ID

[Parent PCO](#) 식별자 또는 NO_SQL_CACHE_STMT 가 올 수 있다.

NO_SQL_CACHE_STMT는 SQL Plan Cache에 등록되지 않은 문장을 의미한다. 다음의 문장들은 SQL Plan Cache에 등록되지 않는다.

- DDL 문장
- DCL 문장
- NO_PLAN_CACHE 힌트를 사용한 문장

SQL_CACHE_PCO_ID

SQL Plan Cache 에 등록된 [Child PCO](#) 식별자를 나타낸다.

OPTIMIZER

최적화 모드를 나타내며 다음과 같은 값을 갖는다.

- 0: 비용(COST) 기반 최적화
- 1: 규칙(RULE) 기반 최적화

COST

질의 최적화하는 비용값을 나타낸다.

USED_MEMORY

향후 확장 예정.

READ_PAGE

질의 수행시 물리적으로 읽은 디스크 데이터 페이지의 개수를 나타낸다.

WRITE_PAGE

질의 수행시 물리적으로 기록한 디스크 데이터 페이지의 개수를 나타낸다.

GET_PAGE

질의 수행시 접근한 디스크 데이터 페이지의 개수를 나타낸다.

CREATE_PAGE

질의 수행시 생성한 디스크 데이터 페이지의 개수를 나타낸다.

UNDO_READ_PAGE

질의 수행시 물리적으로 읽은 디스크 UNDO 페이지의 개수를 나타낸다.

UNDO_WRITE_PAGE

질의 수행시 물리적으로 기록한 디스크 UNDO 페이지의 개수를 나타낸다.

UNDO_CREATE_PAGE

질의 수행시 생성한 디스크 UNDO 페이지의 개수를 나타낸다.

MEM_CURSOR_FULL_SCAN

질의 수행시 메모리 테이블에 대한 검색 중 인덱스 없이 검색한 횟수를 나타낸다.

MEM_CURSOR_INDEX_SCAN

질의 수행시 메모리 테이블에 대한 검색 중 인덱스를 이용한 검색 횟수를 나타낸다.

DISK_CURSOR_FULL_SCAN

질의 수행시 디스크 테이블에 대한 검색 중 인덱스 없이 검색한 횟수를 나타낸다.

DISK_CURSOR_INDEX_SCAN

질의 수행시 디스크 테이블에 대한 검색 중 인덱스를 이용한 검색 횟수를 나타낸다.

EXECUTE_SUCCESS

질의 수행의 성공 횟수를 나타낸다.

EXECUTE_FAILURE

질의 수행의 실패 횟수를 나타낸다.

PROCESS_ROW

질의 수행시 처리한 레코드의 개수를 나타낸다.

MEMORY_TABLE_ACCESS_COUNT

구문 실행 시에 검색 대상이 되는 메모리 테이블들에서 검색되는 레코드 수의 총합이다. 이는 구문 실행 계획에 나타나는 ACCESS 수의 총합과 같다.

SEQNUM

대기 이벤트의 식별자이다.

EVENT

대기 이벤트의 이름이다.

P1

대기 이벤트에 사용되는 파라미터이다.

P2

대기 이벤트에 사용되는 파라미터이다.

P3

대기 이벤트에 사용되는 파라미터이다.

WAIT_TIME

대기 시간 (단위: 밀리초)이다.

SECOND_IN_TIME

대기 시간 (단위: 초)이다.

SIMPLE_QUERY

SIMPLE QUERY 수행 상태를 나타내며, 다음과 같은 값을 갖는다.

0: SIMPLE QUERY 아님

1: SIMPLE QUERY

MATHEMATICS_TEMP_MEMORY

분석 함수에서 사용하는 MATHEMATICS TEMP 메모리 사용량을 나타낸다.

V\$STATNAME

이 테이블은 시스템 전체의 통계 정보를 보여주는 V\$SYSSTAT와 각 세션의 통계 정보를 보여주는 V\$SESSTAT의 통계 정보 일련번호와 이름을 보여준다.

이 테이블은 자체로는 의미가 없으며, 위의 두 가지 성능 뷰와 연결될 때 의미가 있다.

| Column name | Type | Description |
|-------------|--------------|-------------|
| SEQNUM | INTEGER | 통계 일련 번호 |
| NAME | VARCHAR(128) | 통계 이름 |

칼럼 정보

SEQNUM

통계 일련 번호이다.

NAME

통계의 이름을 나타낸다. 각 통계의 일련 번호와 설명은 아래 표와 같다. 각 통계치는 V\$SYSSTATE과 V\$SESSTAT 성능 뷰에서 64비트 정수로 표현된다.

| SEQ | NAME | Description |
|-----|-------------------------|---------------------------------------|
| 0 | logon current | 현재 접속된 사용자 수 |
| 1 | logon cumulative | 접속 사용자 수의 누적합 |
| 2 | data page read | 시스템/세션의 페이지 읽은 횟수 |
| 3 | data page write | 시스템/세션의 페이지 쓴 횟수 |
| 4 | data page gets | 시스템/세션에서 래치를 사용해서 페이지에 접근한 횟수 |
| 5 | data page fix | 시스템/세션에서 래치를 사용하지 않고 페이지에 접근한 횟수 |
| 6 | data page create | 시스템/세션의 페이지 생성 횟수 |
| 7 | undo page read | 시스템/세션의 UNDO 페이지 읽은 횟수 |
| 8 | undo page write | 시스템/세션의 UNDO 페이지 쓴 횟수 |
| 9 | undo page gets | 시스템/세션에서 래치를 사용해서 UNDO 페이지에 접근한 횟수 |
| 10 | undo page fix | 시스템/세션에서 래치를 사용하지 않고 UNDO 페이지에 접근한 횟수 |
| 11 | undo page create | 시스템/세션의 UNDO 페이지 생성 횟수 |
| 12 | base time in second | 시스템이 유지하고 있는 내부 시간(초) |
| 13 | query timeout | 시스템/세션에서 발생한 Query Timeout 횟수 |
| 14 | ddl timeout | 시스템/세션에서 발생한 DDL Timeout 횟수 |
| 15 | idle timeout | 시스템/세션에서 발생한 Idle Timeout 횟수 |
| 16 | fetch timeout | 시스템/세션에서 발생한 Fetch Timeout 횟수 |
| 17 | utrans timeout | 시스템/세션에서 발생한 utrans Timeout 횟수 |
| 18 | session terminated | 시스템/세션에서 발생한 세션 강제 종료 횟수 |
| 19 | ddl sync timeout | 시스템/세션에서 발생한 DDL Sync Timeout 횟수 |
| 20 | statement rebuild count | 시스템/세션에서 statement가 rebuild된 횟수 |

| SEQ | NAME | Description |
|-----|------------------------------------|---|
| 21 | unique violation count | 시스템/세션에서 유일 키 제약 위배 횟수 |
| 22 | update retry count | 시스템/세션에서 갱신 작업 재시도 횟수 |
| 23 | delete retry count | 시스템/세션에서 삭제 작업 재시도 횟수 |
| 24 | lock row retry count | 시스템/세션에서 행 잠금 재시도 횟수 |
| 25 | session commit | 시스템/세션에서 발생한 commit 횟수 |
| 26 | session rollback | 시스템/세션에서 발생한 rollback 횟수 |
| 27 | fetch success count | 시스템/세션에서 fetch 성공 횟수 |
| 28 | fetch failure count | 시스템/세션에서 fetch 실패 횟수 |
| 29 | execute success count | 시스템/세션에서 쿼리가 성공적으로 수행된 횟수 |
| 30 | execute success count : insert | 시스템/세션에서 insert 구문이 성공적으로 수행된 횟수 |
| 31 | execute success count : update | 시스템/세션에서 update 구문이 성공적으로 수행된 횟수 |
| 32 | execute success count : delete | 시스템/세션에서 delete 구문이 성공적으로 수행된 횟수 |
| 33 | execute success count : select | 시스템/세션에서 select 구문이 성공적으로 수행된 횟수 |
| 34 | rep_execute success count : insert | 시스템/세션에서 이중화 대상인 테이블에 insert 구문이 성공적으로 수행된 횟수 |
| 35 | rep_execute success count : update | 시스템/세션에서 이중화 대상인 테이블에 update 구문이 성공적으로 수행된 횟수 |
| 36 | rep_execute success count : delete | 시스템/세션에서 이중화 대상인 테이블에 delete 구문이 성공적으로 수행된 횟수 |
| 37 | execute failure count | 시스템/세션에서 Query의 수행이 실패한 횟수 |
| 38 | prepare success count | 시스템/세션에서 Prepare가 성공한 횟수 |
| 39 | prepare failure count | 시스템/세션에서 Prepare가 실패한 횟수 |
| 40 | rebuild count | 시스템/세션에서 plan cache object 의 rebuild 횟수 |
| 41 | write redo log count | 시스템/세션에서 기록한 로그 레코드의 개수 |
| 42 | write redo log bytes | 시스템/세션에서 기록한 로그의 총 바이트 수 |
| 43 | read socket count | 시스템/세션에서 소켓으로부터 데이터를 읽은 횟수 |
| 44 | write socket count | 시스템/세션에서 소켓에 데이터를 쓴 횟수 |
| 45 | byte received via inet | 시스템/세션에서 INET 소켓을 통해 읽은 데이터 (단위: 바이트) |

| SEQ | NAME | Description |
|-----|--------------------------------------|---|
| 46 | byte sent via inet | 시스템/세션에서 INET 소켓을 통해 쓴 데이터 (단위: 바이트) |
| 47 | byte received via unix domain | 시스템/세션에서 Unix Domain 소켓으로부터 읽은 데이터 (단위: 바이트) |
| 48 | byte sent via unix domain | 시스템/세션에서 Unix Domain 소켓에 쓴 데이터 (단위: 바이트) |
| 49 | semop count for receiving via ipc | 시스템/세션에서 IPC로 읽기 과정에서 수행한 세마퍼 연산 횟수 |
| 50 | semop count for sending via ipc | 시스템/세션에서 IPC로 쓰기 과정에서 수행한 세마퍼 연산 횟수 |
| 51 | memory table cursor full scan count | 시스템/세션에서 수행한 메모리 테이블에 대한 full scan 커서 열기 횟수 (full scan 커서는 한 테이블 전체를 스캔하는 forward-only 커서이다) |
| 52 | memory table cursor index scan count | 시스템/세션에서 수행한 메모리 테이블에 대한 인덱스 스캔 커서 열기 횟수 |
| 53 | memory table cursor GRID scan count | 시스템/세션에서 수행한 메모리 테이블에 대한 GRID 스캔 커서 열기 횟수 |
| 54 | disk table cursor full scan count | 시스템/세션에서 수행한 디스크 테이블에 대한 full scan 커서 열기 횟수 |
| 55 | disk table cursor index scan count | 시스템/세션에서 수행한 디스크 테이블에 대한 인덱스 커서 열기 횟수 |
| 56 | disk table cursor GRID scan count | 시스템/세션에서 수행한 디스크 테이블에 대한 GRID 스캔 커서 열기 횟수 |
| 57 | lock acquired count | 시스템/세션에서 수행한 테이블에 대한 잠금 획득 횟수 (주의: 내부적인 이유로, V\$SYSSTAT의 이 값은 아래 "lock released" 값과 같지 않을 수 있다. 그러나 V\$SESSTAT의 경우에는 두 값이 동일해야 한다.) |
| 58 | lock released count | 시스템/세션에서 수행한 테이블에 대한 잠금 해제 횟수 |
| 59 | service thread created count | 시스템/세션에서 생성된 서비스 스레드 개수 |
| 60 | memory table access count | 시스템/세션에서 메모리 테이블에 접근한 횟수 |
| 61 | missing ppco x-trylatch count | Parent PCO에 x-trylatch의 실패 횟수 |
| 62 | read IB count | 시스템/세션에서 IB로부터 데이터를 읽은 횟수 |
| 63 | write IB count | 시스템/세션에서 IB에 데이터를 쓴 횟수 |
| 64 | byte received via IB | 시스템/세션에서 IB를 이용하여 읽은 데이터(단위: 바이트) |

| SEQ | NAME | Description |
|-----|---|---|
| 65 | byte sent via IB | 시스템/세션에서 IB를 이용하여 쓴 데이터(단위: 바이트) |
| 66 | elapsed time ¹⁵ : query parse | 쿼리 구문 해석에 소요된 누적 시간 |
| 67 | elapsed time: query validate | 쿼리 유효성 검사에 소요된 누적 시간 |
| 68 | elapsed time: query optimize | 쿼리 최적화에 소요된 누적 시간 |
| 69 | elapsed time: query execute | 쿼리 수행에 소요된 누적 시간 |
| 70 | elapsed time: query fetch | 쿼리 결과 fetch에 소요된 누적 시간 |
| 71 | elapsed time: soft prepare | Soft prepare에 소요된 누적 시간 |
| 72 | elapsed time: analyze values in DML(disk) | DML 구문 (INSERT 또는 UPDATE) 실행 시 입력 칼럼 값을 분석하는데 소요된 누적 시간 |
| 73 | elapsed time: record lock validation in DML(disk) | 레코드 갱신이 가능한지 확인하는데 소요된 누적 시간 |
| 74 | elapsed time: allocate data slot in DML(disk) | DML 작업 중 데이터 슬롯을 할당하는데 소요된 누적 시간 |
| 75 | elapsed time: write undo record in DML(disk) | 언두 레코드를 기록하는데 소요된 누적 시간 |
| 76 | elapsed time: allocate tss in DML(disk) | 트랜잭션 슬롯을 할당하는데 소요된 누적 시간 |
| 77 | elapsed time: allocate undopage in DML(disk) | 언두 페이지를 할당하는데 소요된 누적 시간 |
| 78 | elapsed time: index operation in DML(disk) | 인덱스에 키를 추가하는데 소요된 누적 시간 |
| 79 | elapsed time: create page(disk) | 페이지 생성에 소요된 누적 시간 |
| 80 | elapsed time: get page(disk) | 래치를 사용해서 페이지에 접근하는데 소요된 누적 시간 |
| 81 | elapsed time: fix page(disk) | 래치를 사용하지 않고 페이지에 접근하는데 소요된 누적 시간 |
| 82 | elapsed time: logical aging by tx in DML(disk) | 현재 사용되지 않음 |
| 83 | elapsed time: physical aging by tx in DML(disk) | 현재 사용되지 않음 |
| 84 | elapsed time: replace (plan cache) | 리스트내의 한 플랜을 다른 플랜으로 교체하는데 소요된 누적 시간 |

| SEQ | NAME | Description |
|-----|--|---|
| 85 | elapsed time: victim free in replace (plan cache) | 리스트내의 한 플랜을 다른 플랜으로 교체 중에 희생된 플랜을 해제하는데 소요된 누적 시간 |
| 86 | elapsed time: hard rebuild | 플랜 캐시에서 찾아낸 플랜이 유효하지 않아서 rebuild 하는데 소요된 누적 시간 |
| 87 | elapsed time: soft rebuild | 플랜 캐시에서 찾아낸 플랜이 유효하지 않아서 rebuild 하는 것을 다른 트랜잭션이 대기하는데 소요된 누적 시간 |
| 88 | elapsed time: add hard-prepared plan to plan cache | Hard prepare (즉 플랜 강제 생성)된 플랜을 플랜 캐시에 추가하는데 소요된 누적 시간 |
| 89 | elapsed time: add hard-rebuilt plan to plan cache | Hard rebuild (86번 참고)된 플랜을 플랜 캐시에 추가하는데 소요된 누적 시간 |
| 90 | elapsed time: search time for parent PCO | 부모 PCO (SQL 텍스트를 갖는 Plan Cache Object) 를 찾는데 소요된 누적 시간 |
| 91 | elapsed time: creation time for parent PCO | 새로운 부모 PCO를 생성하는데 소요된 누적 시간 |
| 92 | elapsed time: search time for child PCO | 98번과 99번의 합 (즉 $98 + 99$). 이 값은 누적된다. |
| 93 | elapsed time: creation time for child PCO | 새로운 자식 PCO (실행 계획을 갖는 Plan Cache Object)를 생성하는데 소요된 누적 시간 |
| 94 | elapsed time: validation time for child PCO | 자식 PCO의 유효성 검사에 소요된 누적 시간 |
| 95 | elapsed time: creation time for new child PCO by rebuild at execution | 실행 단계에서 플랜을 재구축하는 경우 새로운 자식 PCO를 생성하는데 소요된 누적 시간 |
| 96 | elapsed time: creation time for new child PCO by rebuild at soft prepare | Soft prepare 중 플랜을 재구축하는 경우 새로운 자식 PCO를 생성하는데 소요된 누적 시간 |
| 97 | elapsed time: hard prepare time | 플랜 캐시에 찾으려는 플랜이 없을 때 hard prepare (즉 플랜을 생성)하는데 소요된 누적 시간 |
| 98 | elapsed time: matching time for child PCO | 같은 SQL 텍스트를 갖는 두 개 이상의 자식 PCO가 플랜 캐시에 있는 경우 어떤 플랜이 원하는 것인 것 결정하는데 소요된 누적 시간 |
| 99 | elapsed time: waiting time for hard prepare | 97번과 88번의 합 (즉 $97 + 88$). 이 값은 누적된다. |
| 100 | elapsed time: moving time from cold region to hot region | COLD 영역에서 HOT 영역으로 플랜을 이동하는데 소요된 누적 시간 |

| SEQ | NAME | Description |
|-----|---|---|
| 101 | elapsed time: waiting time for parent PCO when choosing plan cache replacement victim | 교체 대상을 선택할 때, 자식 PCO의 검사를 위해 부모 PCO의 래치 획득에 대기한 누적 시간 |
| 102 | elapsed time: privilege checking time during soft prepare | Soft prepare 중 객체 접근을 위한 권한 검사에 소요된 누적 시간 |
| 103 | elapsed time: copying logs to replication log buffer (sender side) | 로그를 이중화 로그 버퍼에 복사한 누적 시간 (송신자 측) |
| 104 | elapsed time: sender(s) waiting for new logs | 송신자가 수신자에게 보낼 새로운 로그를 대기한 누적 시간 |
| 105 | elapsed time: sender(s) reading logs from replication log buffer | 송신자가 이중화 로그 버퍼로부터 로그를 읽은 누적 시간 |
| 106 | elapsed time: sender(s) reading logs from log file(s) | 송신자가 로그 파일로부터 로그를 읽은 누적 시간 |
| 107 | elapsed time: sender(s) checking whether logs are useful | 송신자가 로그를 이중화 해야 하는 로그인지 체크하는데 소요된 누적 시간 |
| 108 | elapsed time: sender(s) analyzing logs | 송신자가 로그를 분석하고 XLog로 변환한 누적 시간 |
| 109 | elapsed time: sender(s) sending XLogs to receiver(s) | 송신자가 XLog를 수신자에게 보내는 데 걸린 누적 시간 |
| 110 | elapsed time: sender(s) receiving ACK from receiver(s) | 송신자가 수신자로부터 ACK를 받기를 대기하고 수신하는데 걸린 누적 시간 |
| 111 | elapsed time: sender(s) setting ACKed value | 수신자로부터 받은 ACK값을 분석하는데 걸린 누적 시간 |
| 112 | elapsed time: receiver(s) receiving XLogs from sender(s) | 수신자가 송신자로부터 XLog를 받는 데 걸린 누적 시간 |
| 113 | elapsed time: receiver(s) performing endian conversion | 수신자가 byte order를 변환하는데 걸린 누적 시간 |
| 114 | elapsed time: receiver(s) beginning transaction(s) | 수신자가 트랜잭션을 시작하는 데 걸린 누적 시간 |
| 115 | elapsed time: receiver(s) committing transaction(s) | 수신자가 트랜잭션을 커밋하는 데 걸린 누적 시간 |

| SEQ | NAME | Description |
|-----|---|--|
| 116 | elapsed time: receiver(s) aborting transaction(s) | 수신자가 트랜잭션을 롤백하는 데 걸린 누적 시간 |
| 117 | elapsed time: receiver(s) opening table cursor(s) | 수신자가 테이블 커서를 여는 데 걸린 누적 시간 |
| 118 | elapsed time: receiver(s) closing table cursor(s) | 수신자가 테이블 커서를 닫는 데 걸린 누적 시간 |
| 119 | elapsed time: receiver(s) inserting rows | 수신자가 레코드를 입력하는 데 걸린 누적 시간 |
| 120 | elapsed time: receiver(s) updating rows | 수신자가 레코드를 변경하는 데 걸린 누적 시간 |
| 121 | elapsed time: receiver(s) deleting rows | 수신자가 레코드를 삭제하는 데 걸린 누적 시간 |
| 122 | elapsed time: receiver(s) opening lob cursor(s) | 수신자가 LOB cursor를 닫는 데 걸린 누적 시간 |
| 123 | elapsed time: receiver(s) preparing to write LOB(s) | 수신자가 LOB 쓰기를 준비하는 데 걸린 누적 시간 |
| 124 | elapsed time: receiver(s) writing LOB piece(s) | 수신자가 LOB piece(s)를 쓰는 데 걸린 누적 |
| 125 | elapsed time: receiver(s) finish writing LOBs | 수신자가 LOB 쓰기를 마치는 데 걸린 누적 시간 |
| 126 | elapsed time: receiver(s) closing LOB cursor(s) | 수신자가 lob cursor를 닫는 데 걸린 누적 시간 |
| 127 | elapsed time: receiver(s) comparing images to check for conflicts | 수신자가 데이터 충돌을 검사하기 위해서, 양 쪽 서버의 이미지 데이터를 비교하는 데 걸린 누적 시간 |
| 128 | elapsed time: receiver(s) sending ACK | 수신자가 ACK를 보내는 데 걸린 누적 시간 |
| 129 | elapsed time: receiver(s) trim LOB(s) | 수신자가 LOB trim을 마칠 때까지 대기한 누적 시간 |
| 130 | elapsed time: task schedule | task 스케줄링으로 대기하는 총 누적 시간 (단위: Microsecond) |
| 131 | max time: task schedule | task 스케줄링으로 대기한 최대 시간. 대기 시간이 가장 긴 것만 기록 (단위: Microsecond) |

[¹⁵] elapsed time 단위 : microsecond

V\$SYSSTAT

시스템 상태를 보여준다. 그러나 상태값은 모든 세션의 정보에 기반하여 3초마다 갱신되기 때문에, 보여주는 값들은 시간이 지난 값일 수 있다.

| Column name | Type | Description |
|-------------|--------------|-------------|
| SEQNUM | INTEGER | 통계치 일련 번호 |
| NAME | VARCHAR(128) | 통계치 이름 |
| VALUE | BIGINT | 통계치 값 |

각 통계치에 대한 설명은 V\$STATNAME 성능 뷰를 참조한다.

칼럼 정보

SEQNUM

시스템의 통계치를 나타내는 일련 번호를 나타낸다.

NAME

통계치 일련 번호에 해당하는 이름을 나타낸다.

VALUE

통계치 일련 번호에 해당하는 현재 시스템의 값을 64비트 정수로 표현한다.

V\$SYSTEM_CONFLICT_PAGE

디스크 버퍼 공간 상에서 페이지간 래치(Latch) 경합에 의한 병목 구간을 분석할 수 있도록 페이지 타입별로 경합 정보를 보여준다.

TIMED_STATISTICS 프로퍼티가 1로 설정된 경우에만 정보를 수집한다.

| Column name | Type | Description |
|-----------------|-------------|-------------|
| PAGE_TYPE | VARCHAR(21) | 페이지 타입 |
| LATCH_MISS_CNT | BIGINT | 래치 획득 실패 횟수 |
| LATCH_MISS_TIME | BIGINT | 대기 시간 |

칼럼 정보

PAGE_TYPE

페이지 타입을 나타낸다.

LATCH_MISS_CNT

버퍼 페이지의 래치 획득 실패 횟수를 나타낸다.

LATCH_MISS_TIME

버퍼 페이지의 래치 획득 실패로 인한 대기 시간 (단위: 마이크로 초)을 나타낸다.

V\$SYSTEM_EVENT

Altibase 구동 후부터 현재까지 대기 이벤트별로 누적된 대기 통계 정보를 보여준다.

| Column name | Type | Description |
|-------------------|--------------|-----------------------------------|
| EVENT | VARCHAR(128) | 대기 이벤트 이름 |
| TOTAL_WAITS | BIGINT | 대기 이벤트에 대한 총 대기 횟수 |
| TOTAL_TIMEOUTS | BIGINT | 지정된 시간 이후에도 요청한 리소스를 획득하는데 실패한 횟수 |
| TIME_WAITED | BIGINT | 대기 이벤트에 대한 대기시간 (밀리초) |
| AVERAGE_WAIT | BIGINT | 대기 이벤트에 대한 평균 대기시간 (밀리초) |
| TIME_WAITED_MICRO | BIGINT | 대기 이벤트에 대한 대기 시간 (마이크로초) |
| EVENT_ID | INTEGER | 대기 이벤트의 식별자 |
| WAIT_CLASS_ID | INTEGER | 대기 클래스의 식별자 |
| WAIT_CLASS | VARCHAR(128) | 대기 클래스 이름 |

칼럼 정보**EVENT**

대기 이벤트의 이름을 나타낸다.

TOTAL_WAITS

이 대기 이벤트에 대한 전체 대기 횟수를 나타낸다.

TOTAL_TIMEOUTS

이 대기 이벤트에 대해 지정된 시간 이후에도 요청한 리소스를 획득하는데 실패한 횟수를 나타낸다.

TIME_WAITED

이 대기 이벤트에 대한 모든 세션들의 총 대기 시간을 나타낸다. (단위: 밀리초)

AVERAGE_WAIT

이 대기 이벤트에 대한 평균 대기 시간을 나타낸다. (단위: 밀리초)

TIME_WAITED_MICRO

이 대기 이벤트에 대한 모든 세션들의 총 대기 시간을 나타낸다. (단위: 마이크로초)

EVENT_ID

대기 이벤트의 ID를 나타낸다.

WAIT_CLASS_ID

이벤트를 그룹화한 대기 클래스 식별자를 나타낸다.

WAIT_CLASS

이벤트를 그룹화한 대기 클래스의 이름이다.

V\$SYSTEM_WAIT_CLASS

Altibase 구동 후부터 현재까지의 대기 클래스별로 분류해서 누적된 대기 통계 정보를 보여준다.

| Column name | Type | Description |
|---------------|--------------|--------------------------|
| WAIT_CLASS_ID | INTEGER | 대기 이벤트 식별자 |
| WAIT_CLASS | VARCHAR(128) | 대기 클래스 이름 |
| TOTAL_WAITS | BIGINT | 대기 클래스에 대한 총 대기 횟수 |
| TIME_WAITED | DOUBLE | 대기 클래스에 대한 총 대기 시간 (밀리초) |

칼럼 정보**WAIT_CLASS_ID**

대기 클래스 식별자이다.

WAIT_CLASS

대기 클래스 이름이다.

TOTAL_WAITS

이 대기 클래스를 대기한 총 횟수다.

TIME_WAITED

세션에서 이 대기 클래스를 대기한 총 시간이다. (단위: 밀리초)

예제

<예 1> 현재 발생하는 대기 이벤트에 대한 대기 클래스별 대기 횟수와 대기 시간을 보여준다.

```
iSQL> select * from v$system_wait_class order by total_waits desc;
```

<예 2> 가장 오래 대기한 대기 클래스부터 대기 클래스 별로 전체 대비 대기 횟수 비율과 대기 시간 비율을 내림차순으로 출력한다.

```
iSQL> select
    WAIT_CLASS,
    TOTAL_WAITS,
    round(100 * (TOTAL_WAITS / SUM_WAITS), 2) PCT_WAITS,
```

```

        TIME_WAITED,
        round(100 * (TIME_WAITED / SUM_TIME),2) PCT_TIME
from
  (select WAIT_CLASS,
    TOTAL_WAITS,
    TIME_WAITED
  from V$SYSTEM_WAIT_CLASS
  where WAIT_CLASS != 'Idle'),
  (select sum(TOTAL_WAITS) SUM_WAITS,
    sum(TIME_WAITED) SUM_TIME
  from V$SYSTEM_WAIT_CLASS
  where WAIT_CLASS != 'Idle')
order by 5 desc;

```

V\$TABLE

성능 뷰 리스트를 보여준다.

| Column name | Type | Description |
|-------------|-------------|-------------|
| NAME | VARCHAR(39) | 뷰 이름 |
| SLOTSIZE | INTEGER | 레코드의 크기 |
| COLUMNCOUNT | SMALLINT | 칼럼의 개수 |

칼럼 정보

NAME

성능 뷰의 이름이다.

SLOTSIZE

해당 성능 뷰가 가진 한 레코드의 크기이다.

COLUMNCOUNT

해당 성능 뷰가 가진 칼럼의 개수이다.

V\$TABLESPACES

테이블스페이스의 정보를 보여준다.

| Column name | Type | Description |
|--------------|-------------|-------------------|
| ID | INTEGER | 테이블스페이스 식별자 |
| NAME | VARCHAR(40) | 테이블스페이스 이름 |
| NEXT_FILE_ID | INTEGER | 다음 생성될 데이터 파일 식별자 |
| TYPE | INTEGER | 테이블스페이스 타입 |
| STATE | INTEGER | 테이블스페이스의 상태 |

| Column name | Type | Description |
|----------------------|-------------|---|
| EXTENT_MANAGEMENT | VARCHAR(20) | 사용자가 디스크 테이블스페이스를 생성할 때 정한 익스텐트 (extent)를 관리하는 방식 |
| SEGMENT_MANAGEMENT | VARCHAR(20) | 테이블스페이스의 세그먼트 타입 |
| DATAFILE_COUNT | INTEGER | 테이블스페이스의 파일 개수 |
| TOTAL_PAGE_COUNT | BIGINT | 총 페이지 개수 |
| EXTENT_PAGE_COUNT | INTEGER | 해당 테이블스페이스의 익스텐트 크기 (페이지 개수) |
| ALLOCATED_PAGE_COUNT | BIGINT | 해당 테이블스페이스에서 초기화된 페이지 개수 |
| PAGE_SIZE | INTEGER | 테이블스페이스의 페이지 크기(bytes) |
| ATTR_LOG_COMPRESS | INTEGER | 테이블스페이스에 속하는 테이블에 DML 수행 시 로그 압축 여부 |

칼럼 정보

ID

테이블스페이스의 식별자이다. 사용자 테이블스페이스는 식별자 값으로 5부터 부여되며, 계속 증가한다.

NAME

CREATE TABLESPACE 구문에 정의된 테이블스페이스의 이름이다.

NEXT_FILE_ID

테이블스페이스에 데이터 파일이 추가될 경우, 데이터 파일에 부여할 식별자이다. 하나의 데이터 파일이 추가될 때마다 이 값은 1 씩 증가한다.

TYPE

테이블스페이스의 타입을 나타낸다.

- 0: 메모리 시스템 디렉터리 (MEMORY_SYSTEM_DICTIONARY)
- 1: 메모리 시스템 데이터 (MEMORY_SYSTEM_DATA)
- 2: 메모리 사용자 데이터 (MEMORY_USER_DATA)
- 3: 디스크 시스템 데이터 (DISK_SYSTEM_DATA)
- 4: 디스크 사용자 데이터 (DISK_USER_DATA)
- 5: 디스크 시스템 템프 (DISK_SYSTEM_TEMP)
- 6: 디스크 사용자 템프 (DISK_USER_TEMP)
- 7: 디스크 시스템 언두 (DISK_SYSTEM_UNDO)
- 8: 휘발성 사용자 데이터 (VOLATILE_USER_DATA)

STATE

테이블스페이스의 상태를 나타낸다.

- 1: 오프라인 (OFFLINE)
- 2: 온라인 (ONLINE)
- 5: 백업중인 오프라인 테이블스페이스
- 6: 백업중인 온라인 테이블스페이스
- 128: 삭제된 테이블스페이스 (Dropped)
- 1024: 폐기된 테이블스페이스 (Discarded)
- 1028: 백업중인 폐기된 테이블스페이스

EXTENT_MANAGEMENT

사용자가 디스크 테이블스페이스를 생성할 때 결정한 익스텐트를 관리하는 방식이다. 현재는 비트맵 (BITMAP) 방식을 제공한다.

- BITMAP: 테이블스페이스의 모든 익스텐트의 할당 여부를 관리

SEGMENT_MANAGEMENT

테이블스페이스에서 세그먼트를 생성할 때 어떤 타입으로 생성된 것인지를 나타낸다.

- MANUAL: 프리(Free) 페이지 관리를 프리 리스트로 하는 세그먼트 (FMS, Free list Management Segment) 생성
- AUTO: 프리 페이지 관리를 비트맵 인덱스 기반으로 하는 세그먼트 (TMS, bitmap-based Tree Management Segment) 생성

DATAFILE_COUNT

테이블스페이스에 포함된 데이터 파일의 개수를 나타낸다.

TOTAL_PAGE_COUNT

테이블스페이스의 크기를 페이지 개수로 나타낸다. 실제 테이블스페이스의 크기는 이 값과 페이지 크기의 곱 (TOTAL_PAGE_COUNT * PAGE_SIZE)으로 계산할 수 있다. 파일마다 파일 헤더를 위한 한 페이지씩을 제외하고 실제 사용할 수 있는 페이지이다.

EXTENT_PAGE_COUNT

해당 테이블스페이스의 익스텐트 크기를 페이지 개수로 나타낸다. 하나의 익스텐트가 가지는 페이지 개수를 의미하며, 최소 3개 이상의 페이지를 갖는다.

ALLOCATED_PAGE_COUNT

해당 테이블스페이스에서 초기화된 페이지의 개수를 나타낸다.

PAGE_SIZE

테이블스페이스의 각 페이지 크기를 나타낸다. 디스크 테이블스페이스의 페이지는 8KB, 메모리 테이블스페이스의 페이지는 32KB이다.

ATTR_LOG_COMPRESS

테이블스페이스에 속하는 테이블에 DML을 수행할 때, 로그 압축 수행 여부를 나타낸다.

- 0: LOG COMPRESS 수행 안한다.
- 1: LOG COMPRESS 수행한다.

V\$TIME_ZONE_NAMES

TIME_ZONE 프로퍼티에 설정할 수 있는 지역 이름과 약어 및 UTC 오프셋 값의 목록을 보여주는 성능 뷰이다.

| Column name | Type | Description |
|-------------|-------------|-------------|
| NAME | VARCHAR(40) | 지역 이름 또는 약어 |
| UTC_OFFSET | VARCHAR(6) | UTC 오프셋 |

칼럼 정보**NAME**

Asia/Seoul 또는 KST와 같은 타임 존 설정을 위한 지역 이름의 문자열 또는 약어이다.

UTC_OFFSET

타임 존의 UTC(협정 세계시)로부터의 오프셋 값이다. 예를 들어, Asia/Seoul의 경우 UTC 오프셋이 +09:00이다.

V\$TRACELOG

데이터베이스 내부 모듈의 수행 내역을 남기는 메시지 로깅 관련 정보를 보여준다.

| Column name | Type | Description |
|-------------|-------------|---|
| MODULE_NAME | VARCHAR(16) | 모듈명 |
| TRCLEVEL | INTEGER | 로깅 레벨 (1~32) |
| FLAG | VARCHAR(8) | 이 모듈의 로깅 설정 여부 |
| POWLEVEL | BIGINT | 2의 (레벨 - 1) 거듭제곱 ($2^{(TRCLEVEL-1)}$) |
| DESCRIPTION | VARCHAR(64) | 설정된 레벨에 대한 설명 |

칼럼 정보**MODULE_NAME**

Altibase 모듈의 이름을 나타낸다. 현재 Altibase는 SERVER, QP, RP, SM, MM, CM의 모듈로 구성되며, 각 모듈 별로 메시지 로그를 남길 수 있다.

TRCLEVEL

이력을 남기기 위한 메시지 로깅 레벨을 나타낸다. 1에서 32의 값을 가진다.

FLAG

이 모듈의 이력 메시지가 출력되도록 설정되어 있는지 여부와 레벨을 나타낸다.

- X: 출력되지 않는 상태
- O: 출력중인 상태
- SUM: 이 값은 이 레코드의 POWLEVEL 칼럼의 값이 각 모듈에서 FLAG 값이 'O'인 POWLEVEL 칼럼 값들의 합임을 나타낸다.

출력 설정에 대한 자세한 내용은 하단의 사용방법을 참고한다.

POWLEVEL

2의 (TRCLEVEL-1) 제곱, 즉 $2^{(TRCLEVEL-1)}$ 이다. 사용자가 로깅 레벨을 쉽게 설정할 수 있도록, 저장 프로시저 addTrcLevel()와 delTrcLevel()가 제공된다. 해당 저장 프로시저는 패키지에 포함된 tracelog.sql를 실행하여 생성할 수 있다.

DESCRIPTION

레벨에 대응하는 설명을 나타낸다.

예제

현재 서버 모듈에 대해 설정된 트레이스 로깅 레벨을 확인한다.

```
isQL> select module_name, trclevel, flag, powlevel, description from v$tracelog
where module_name like '%SER%';
MODULE_NAME TRCLEVEL FLAG POWLEVEL DESCRIPTION
-----
SERVER 1 1 O 1 [DEFAULT] TimeOut(Query,Fetch,Idle,UTrans) Trace Log
SERVER 2 2 O 2 [DEFAULT] Network Operation Fail Trace Log
SERVER 3 4 O 4 [DEFAULT] Memory Operation Warning Trace Log
SERVER 4 8 X 8 ---
SERVER 5 16 X 16 ---
SERVER 6 32 X 32 ---
SERVER 7 64 X 64 ---
SERVER 8 128 X 128 ---
SERVER 9 256 X 256 ---
SERVER 10 512 X 512 ---
SERVER 11 1024 X 1024 ---
SERVER 12 2048 X 2048 ---
SERVER 13 4096 X 4096 ---
SERVER 14 8192 X 8192 ---
SERVER 15 16384 X 16384 ---
SERVER 16 32768 X 32768 ---
SERVER 17 65536 X 65536 ---
SERVER 18 131072 X 131072 ---
SERVER 19 262144 X 262144 ---
SERVER 20 524288 X 524288 ---
SERVER 21 1048576 X 1048576 ---
SERVER 22 2097152 X 2097152 ---
SERVER 23 4194304 X 4194304 ---
SERVER 24 8388608 X 8388608 ---
```

```

SERVER 25 X 16777216 ---
SERVER 26 X 33554432 ---
SERVER 27 X 67108864 ---
SERVER 28 X 134217728 ---
SERVER 29 X 268435456 ---
SERVER 30 X 536870912 ---
SERVER 31 X 1073741824 ---
SERVER 32 X 2147483648 ---
SERVER 99 SUM 7 Total Sum of Trace Log Values
33 rows selected.

```

사용 방법

Altibase는 8개의 모듈 SERVER, SM, QP, RP, RP_CONFLICT, DR, MM, CM에 대하여 메시지 로깅 프로퍼티가 존재한다.

- SERVER_MSGLOG_FLAG: 통신 및 서버 메시지
- SM_MSGLOG_FLAG: 저장관리자 관련 메시지
- QP_MSGLOG_FLAG: 질의처리기 관련 메시지
- RP_MSGLOG_FLAG: 이중화 관련 메시지
- RP_CONFLICT_MSGLOG_FLAG: 이중화 충돌 관련 메시지
- LB_MSGLOG_FLAG: 서비스 쓰레드 동작 관련 메시지
- MM_MSGLOG_FLAG: 메인 모듈 관련 메시지
- CM_MSGLOG_FLAG: 통신 관련 메시지

각 프로퍼티는 32개의 비트로 설정할 수 있는데, 각 비트에 대한 메시지 종류 및 설명은 V\$TRACELOG를 참조한다.

메시지 로깅 내역의 변경 방법은 다음과 같다.

- 서버의 로깅 메시지가 모두 출력되지 않도록 할 때.

```
alter system set server_msglog_flag=0
```

- 서버의 로깅 메시지 중 첫번째, 두번째, 다섯번째 비트에 해당하는 메시지를 출력하도록 할 때 (1+2+5).

```
alter system set server_msglog_flag=8
```

- 이중화 로깅 메시지 중 충돌 관련 메시지만 출력하고자 할 때.

```
alter system set rp_msglog_flag=2
```

- 질의처리기에서 저장 프로시저의 오류 라인(첫번째 비트)과 DDL의 수행 내역(두번째 비트)을 로깅하고자 할 경우 (1+2)

```
alter system set qp_msglog_flag=3
```

- 이중화 충돌 관련 메시지 중 SQL(세번째 비트)을 출력하고자 할 때.

```
alter system set rp_conflict_msglog_flag=4
```

V\$TRANSACTION

트랜잭션 객체의 정보를 보여준다.

| Column name | Type | Description |
|------------------------------|-------------|-------------|
| ID | BIGINT | 트랜잭션 식별자 |
| SESSION_ID | INTEGER | 아래 참조 |
| MEMORY_VIEW_SCN | VARCHAR(29) | 아래 참조 |
| MIN_MEMORY_LOB_VIEW_SCN | VARCHAR(29) | 아래 참조 |
| DISK_VIEW_SCN | VARCHAR(29) | 아래 참조 |
| MIN_DISK_LOB_VIEW_SCN | VARCHAR(29) | 아래 참조 |
| COMMIT_SCN | VARCHAR(29) | 아래 참조 |
| STATUS | BIGINT | 아래 참조 |
| UPDATE_STATUS | BIGINT | 아래 참조 |
| LOG_TYPE | INTEGER | 아래 참조 |
| XA_COMMIT_STATUS | BIGINT | 아래 참조 |
| XA_PREPARED_TIME | VARCHAR(64) | 아래 참조 |
| FIRST_UNDO_NEXT_LSN_LFGID | INTEGER | 사용하지 않음(0) |
| FIRST_UNDO_NEXT_LSN_FILENO | INTEGER | 아래 참조 |
| FIRST_UNDO_NEXT_LSN_OFFSET | INTEGER | 아래 참조 |
| CURRENT_UNDO_NEXT_SN | BIGINT | 내부 용도 |
| CURRENT_UNDO_NEXT_LSN_LFGID | INTEGER | 사용하지 않음(0) |
| CURRENT_UNDO_NEXT_LSN_FILENO | INTEGER | 내부 용도 |
| CURRENT_UNDO_NEXT_LSN_OFFSET | INTEGER | 내부 용도 |
| LAST_UNDO_NEXT_LSN_LFGID | INTEGER | 사용하지 않음(0) |
| LAST_UNDO_NEXT_LSN_FILENO | INTEGER | 아래 참조 |
| LAST_UNDO_NEXT_LSN_OFFSET | INTEGER | 아래 참조 |
| LAST_UNDO_NEXT_SN | BIGINT | 아래 참조 |
| SLOT_NO | INTEGER | 아래 참조 |
| UPDATE_SIZE | BIGINT | 아래 참조 |
| ENABLE_ROLLBACK | BIGINT | 내부 용도 |

| Column name | Type | Description |
|---------------------------|---------|--------------------|
| FIRST_UPDATE_TIME | INTEGER | 아래 참조 |
| LOG_BUF_SIZE | INTEGER | 내부 용도 |
| LOG_OFFSET | INTEGER | 내부 용도 |
| SKIP_CHECK_FLAG | BIGINT | 내부 용도 |
| SKIP_CHECK_SCN_FLAG | BIGINT | 내부 용도 |
| DDL_FLAG | BIGINT | 아래 참조 |
| TSS_RID | BIGINT | 아래 참조 |
| RESOURCE_GROUP_ID | INTEGER | 로그 파일 그룹(LFG)의 식별자 |
| LEGACY_TRANS_COUNT | INTEGER | 내부 용도 |
| ISOLATION_LEVEL | INTEGER | 아래 참조 |
| PROCESSED_UNDO_TIME | INTEGER | 아래 참조 |
| ESTIMATED_TOTAL_UNDO_TIME | INTEGER | 아래 참조 |
| TOTAL_LOG_COUNT | BIGINT | 아래 참조 |
| TOTAL_UNDO_LOG_COUNT | BIGINT | 아래 참조 |
| PROCESSED_UNDO_LOG_COUNT | BIGINT | 아래 참조 |

칼럼 정보

ID

해당 트랜잭션을 구분할 수 있는 번호로, 0부터 $2^{32} - 1$ 까지의 값을 가진다. 이 값들은 재사용될 수 있다.

SESSION_ID

트랜잭션이 수행되고 있는 세션의 식별자이다. 이 트랜잭션이 어떤 세션과도 연관되어 있지 않다면 -1을 보여주는데, 이는 XA 환경에서 트랜잭션 브랜치가 prepare 된 상태를 나타낸다.

MEMORY_VIEW_SCN

Altibase는 MVCC를 사용하기 때문에 테이블에 대해 각 커서들이 열린 시점을 나타내는 SCN을 가진다. 이 항목은 현재 해당 트랜잭션에서 메모리 테이블에 대해 열려있는 커서의 View SCN 중 가장 작은 값을 나타낸다. 이 값이 263이면 어떤 커서도 열려 있지 않다는 것을 의미한다.

MIN_MEMORY_LOB_VIEW_SCN

현재 해당 트랜잭션에서 열린 메모리 LOB 커서 중 가장 오래된 커서의 SCN을 나타낸다. 이 값이 2^{63} 이면 어떤 커서도 열려있지 않다는 것을 의미한다.

DISK_VIEW_SCN

현재 해당 트랜잭션에서 디스크 테이블에 대해 열려있는 커서의 View SCN 중 가장 작은 값을 나타낸다. 값의 범위는 MEMORY_VIEW_SCN과 동일하다.

MIN_DISK_LOB_VIEW_SCN

현재 해당 트랜잭션에서 열린 디스크 LOB 커서중 가장 오래된 커서의 SCN을 나타낸다. 이 값이 2^{63} 이면 어떤 커서도 열려있지 않다는 것을 의미한다.

COMMIT_SCN

트랜잭션이 커밋한 시점의 시스템 SCN이다. 아직 트랜잭션이 커밋되지 않았다면 2^{63} 을 가진다.

STATUS

현재 트랜잭션의 상태를 나타낸다.

- 0: BEGIN
- 1: PRECOMMIT
- 2: COMMIT_IN_MEMORY
- 3: COMMIT
- 4: ABORT
- 5: BLOCKED
- 6: END

UPDATE_STATUS

해당 트랜잭션이 현재까지 갱신연산을 수행한 트랜잭션인지 read-only 트랜잭션인지를 나타낸다.

- 0: read-only
- 1: updating

LOG_TYPE

해당 트랜잭션이 이중화에 관련된 테이블을 갱신한 적이 있는지를 나타낸다.

- 0: 일반
- 1: 이중화 관련

XA_COMMIT_STATUS

글로벌 트랜잭션에 의한 로컬 트랜잭션의 현재 상태를 표시한다.

- 0: BEGIN
- 1: PREPARED
- 2: COMPLETE

XA_PREPARED_TIME

글로벌 트랜잭션에 의한 로컬 트랜잭션이 PREPARE 명령을 글로벌 트랜잭션 관리자로부터 받은 시점을 나타낸다.

FIRST_UNDO_NEXT_LSN_FILENO

트랜잭션이 처음 기록한 로그의 위치를 나타내는 LSN 중 파일 번호를 나타낸다.

FIRST_UNDO_NEXT_LSN_OFFSET

트랜잭션이 처음 기록한 로그의 위치를 나타내는 LSN 중 파일 내에서의 위치(오프셋)를 나타낸다.

LAST_UNDO_NEXT_LSN_FILENO

트랜잭션이 마지막 기록한 로그의 위치를 나타내는 LSN 중 파일 번호를 나타낸다.

LAST_UNDO_NEXT_LSN_OFFSET

트랜잭션이 마지막 기록한 로그의 위치를 나타내는 LSN 중 파일 내에서의 위치(오프셋)를 나타낸다.

LAST_UNDO_NEXT_SN

트랜잭션이 마지막 기록한 로그의 일련번호이다.

SLOT_NO

트랜잭션 풀 내에서 해당 트랜잭션 객체의 순번을 나타낸다.

UPDATE_SIZE

트랜잭션이 수행한 갱신(Update) 연산에 의해 작성된 로그의 크기를 나타낸다. 이 값은 프로퍼티 중 LOCK_ESCALATION_MEMORY_SIZE 값과 비교되어, 이 값보다 더 커지면 이후로는 테이블에 X락을 잡고 in-place update 방식으로 갱신을 수행하게 된다.

FIRST_UPDATE_TIME

최초로 데이터베이스에 대한 변경이 일어난 시각이 기록된다.

DDL_FLAG

이 트랜잭션이 DDL구문을 수행 중인지 나타낸다.

- 0: non-DDL
- 1: DDL

TSS_RID

디스크 테이블에 대한 갱신 연산 수행을 위해 얻은 TSS (Transaction Status Slot)의 물리적 위치를 나타낸다. 이 값이 0이 아니면 해당 트랜잭션은 디스크 테이블에 대해 갱신연산을 한번이라도 수행했음을 나타낸다.

ISOLATION_LEVEL

트랜잭션의 고립화 수준(isolation level)을 나타낸다.

- 0: READ COMMITTED
- 1: REPEATABLE READ
- 2: SERIALIZABLE

PROCESSED_UNDO_TIME

해당 트랜잭션의 UNDO 시작 시점부터 현재까지 UNDO 진행된 시간 (단위: 초)

ESTIMATED_TOTAL_UNDO_TIME

해당 트랜잭션의 UNDO완료 될 때까지 추정되는 총 소요 시간 (단위: 초)

TOTAL_LOG_COUNT

해당 트랜잭션의 총 로그 개수

TOTAL_UNDO_LOG_COUNT

해당 트랜잭션에서 앞으로 UNDO 해야 할 총 로그 개수

PROCESSED_UNDO_LOG_COUNT

해당 트랜잭션에서 현재까지 언두 완료된 로그 개수

V\$TRANSACTION_MGR

Altibase 트랜잭션 관리자의 정보를 보여준다.

| Column name | Type | Description |
|----------------------|-------------|------------------------|
| TOTAL_COUNT | INTEGER | 트랜잭션 총 개수 |
| FREE_LIST_COUNT | INTEGER | 프리 리스트 개수 |
| BEGIN_ENABLE | BIGINT | 새로운 트랜잭션 시작 가능 여부 |
| ACTIVE_COUNT | INTEGER | 작업중인 트랜잭션의 개수 |
| SYS_MIN_DISK_VIEWSCN | VARCHAR(29) | 트랜잭션 중 가장 작은 디스크 뷰 SCN |

칼럼 정보**TOTAL_COUNT**

Altibase는 시스템 시작시에 프로퍼티에 지정된 개수의 트랜잭션 객체들을 트랜잭션 풀에 미리 생성해 두고 이것을 사용한다. 이 값은 현재 Altibase에서 생성한 트랜잭션 객체의 총 개수를 나타낸다.

FREE_LIST_COUNT

트랜잭션 풀을 분할 관리하는 리스트의 개수를 나타낸다.

BEGIN_ENABLE

새로운 트랜잭션을 시작할 수 있는지를 나타낸다.

- 0: disabled
- 1: enabled

ACTIVE_COUNT

현재 할당되어 작업을 수행중인 트랜잭션 객체의 개수를 나타낸다.

SYS_MIN_DISK_VIEWSCN

트랜잭션 중에서 가장 작은 디스크 뷰 SCN이다.

V\$TSSEGS

언두 테이블스페이스에 존재하는 모든 TSS 세그먼트의 목록을 출력한다.

| Column name | Type | Description |
|----------------------|---------|------------------------------|
| SPACE_ID | INTEGER | 언두 테이블스페이스 식별자 |
| SEG_PID | INTEGER | TSS 세그먼트 페이지 식별자 |
| TXSEG_ENTRY_ID | INTEGER | 트랜잭션 세그먼트 식별자 |
| CUR_ALLOC_EXTENT_RID | BIGINT | TSS 세그먼트에서 현재 사용중인 익스텐트의 RID |
| CUR_ALLOC_PAGE_ID | INTEGER | TSS 세그먼트에서 현재 사용중인 페이지의 식별자 |
| TOTAL_EXTENT_COUNT | BIGINT | TSS 세그먼트의 총 익스텐트 개수 |
| TOTAL_EXTDIR_COUNT | BIGINT | TSS 세그먼트의 총 익스텐트 디렉터리 개수 |
| PAGE_COUNT_IN_EXTENT | INTEGER | 하나의 익스텐트의 총 페이지 개수 |

칼럼 정보**SPACE_ID**

언두 테이블스페이스 식별자이다.

SEG_PID

TSS 세그먼트 페이지의 식별자이다.

TXSEG_ENTRY_ID

트랜잭션 세그먼트의 식별자이다.

CUR_ALLOC_EXTENT_RID

TSS 세그먼트에서 현재 사용중인 익스텐트 RID (Resource Identifier)를 나타낸다.

CUR_ALLOC_PAGE_ID

TSS 세그먼트에서 현재 사용중인 페이지의 식별자이다.

TOTAL_EXTENT_COUNT

TSS 세그먼트의 총 익스텐트의 개수이다.

TOTAL_EXTDIR_COUNT

TSS 세그먼트의 총 익스텐트 디렉터리의 개수이다.

PAGE_COUNT_IN_EXTENT

하나의 익스텐트의 총 페이지의 개수이다.

V\$TXSEGS

트랜잭션에 바인딩되어 온라인 상태로 있는 세그먼트의 목록을 출력한다.

| Column name | Type | Description |
|----------------------|-------------|---------------------------------|
| ID | INTEGER | 트랜잭션 세그먼트의 식별자 |
| TRANS_ID | BIGINT | 세그먼트를 바인딩한 트랜잭션의 식별자 |
| MIN_DISK_VIEW_SCN | VARCHAR(29) | 해당 트랜잭션의 최소 디스크 뷰 SCN |
| COMMIT_SCN | VARCHAR(29) | 해당 트랜잭션의 커밋 SCN |
| FIRST_DISK_VIEW_SCN | VARCHAR(29) | 해당 트랜잭션의 첫번째 디스크 뷰 SCN |
| TSS_RID | BIGINT | 트랜잭션 TSS RID |
| TSSEG_EXTENT_RID | BIGINT | TSS를 할당한 TSS 세그먼트의 익스텐트 RID |
| FST_UDSEG_EXTENT_RID | BIGINT | 트랜잭션이 사용한 언두 세그먼트의 첫번째 익스텐트 RID |
| LST_UDSEG_EXTENT_RID | BIGINT | 트랜잭션이 사용한 언두 세그먼트의 마지막 익스텐트 RID |
| FST_UNDO_PAGEID | INTEGER | 트랜잭션이 기록한 첫번째 언두 레코드의 페이지 식별자 |
| FST_UNDO_SLOTNUM | SMALLINT | 트랜잭션이 기록한 첫번째 언두 레코드의 슬롯 번호 |
| LST_UNDO_PAGEID | INTEGER | 트랜잭션이 기록한 마지막 언두 레코드의 페이지 식별자 |
| LST_UNDO_SLOTNUM | SMALLINT | 트랜잭션이 기록한 마지막 언두 레코드의 슬롯 번호 |

칼럼 정보**ID**

트랜잭션 세그먼트의 식별자이다.

TRANS_ID

세그먼트를 바인딩한 트랜잭션의 식별자이다.

MIN_DISK_VIEW_SCN

트랜잭션의 최소 디스크 뷰 SCN을 나타낸다.

COMMIT_SCN

해당 트랜잭션의 커밋 SCN을 나타낸다.

FIRST_DISK_VIEW_SCN

해당 트랜잭션의 첫번째 디스크 뷰 SCN을 나타낸다.

TSS_RID

해당 트랜잭션이 할당받은 TSS (Transaction Status Slot)의 RID를 나타낸다.

TSSEG_EXTENT_RID

TSS를 할당한 TSS 세그먼트의 익스텐트 RID를 나타낸다.

FST_UDSEG_EXTENT_RID

트랜잭션이 사용한 언두 세그먼트의 첫번째 익스텐트 RID를 나타낸다.

LST_UDSEG_EXTENT_RID

트랜잭션이 사용한 언두 세그먼트의 마지막 익스텐트 RID를 나타낸다.

FST_UNDO_PAGEID

해당 트랜잭션이 갱신때 기록했던 첫번째 언두 레코드의 페이지 식별자를 나타낸다.

FST_UNDO_SLOTNUM

해당 트랜잭션이 갱신때 기록했던 첫번째 언두 레코드의 페이지 내에서의 슬롯 번호를 나타낸다.

LST_UNDO_PAGEID

해당 트랜잭션이 갱신때 기록했던 마지막 언두 레코드의 페이지 식별자를 나타낸다.

LST_UNDO_SLOTNUM

해당 트랜잭션이 갱신때 기록했던 마지막 언두 레코드의 페이지 내에서의 슬롯 번호를 나타낸다.

V\$UDSEGS

언두 테이블스페이스에 존재하는 모든 언두(UNDO) 세그먼트의 목록을 출력한다.

| Column name | Type | Description |
|----------------------|---------|----------------------------|
| SPACE_ID | INTEGER | 언두 테이블스페이스 식별자 |
| SEG_PID | INTEGER | 언두 세그먼트 페이지 식별자 |
| TXSEG_ENTRY_ID | INTEGER | 트랜잭션 세그먼트 식별자 |
| CUR_ALLOC_EXTENT_RID | BIGINT | 언두 세그먼트에서 현재 사용중인 익스텐트 RID |
| CUR_ALLOC_PAGE_ID | INTEGER | 언두 세그먼트에서 현재 사용중인 페이지의 식별자 |
| TOTAL_EXTENT_COUNT | BIGINT | 언두 세그먼트의 총 익스텐트 개수 |
| TOTAL_EXTDIR_COUNT | BIGINT | 언두 세그먼트의 총 익스텐트 디렉터리 개수 |
| PAGE_COUNT_IN_EXTENT | INTEGER | 하나의 익스텐트의 총 페이지 개수 |

칼럼 정보

SPACE_ID

언두 테이블스페이스 식별자이다.

SEG_PID

언두 세그먼트 페이지 식별자이다.

TXSEG_ENTRY_ID

트랜잭션 세그먼트 식별자이다.

CUR_ALLOC_EXTENT_RID

언두 세그먼트에서 현재 사용중인 익스텐트 RID를 나타낸다.

CUR_ALLOC_PAGE_ID

언두 세그먼트에서 현재 사용중인 페이지 식별자이다.

TOTAL_EXTENT_COUNT

언두 세그먼트의 총 익스텐트 개수를 나타낸다.

TOTAL_EXTDIR_COUNT

언두 세그먼트의 총 익스텐트 디렉터리 개수를 나타낸다.

PAGE_COUNT_IN_EXTENT

하나의 익스텐트의 총 페이지 개수를 나타낸다.

V\$UNDO_BUFF_STAT

언두 테이블스페이스의 버퍼 풀 관련 통계 정보를 보여준다.

| Column name | Type | Description |
|-------------------|--------|------------------------|
| READ_PAGE_COUNT | BIGINT | 아래 참조 |
| GET_PAGE_COUNT | BIGINT | 버퍼 매니저에 페이지를 요청한 횟수 |
| FIX_PAGE_COUNT | BIGINT | 버퍼 매니저에 언두 페이지를 요청한 횟수 |
| CREATE_PAGE_COUNT | BIGINT | 아래 참조 |
| HIT_RATIO | DOUBLE | 버퍼 프레임의 히트율 |

칼럼 정보

READ_PAGE_COUNT

버퍼 초기화 이후 디스크로부터 페이지를 읽은 총 횟수를 나타낸다.

GET_PAGE_COUNT

버퍼 초기화 이후 버퍼 매니저에 페이지를 요청한 총 횟수를 나타낸다. 만약 페이지가 버퍼에 있다면 버퍼 매니저는 이 요청에 대해 버퍼의 페이지를 리턴하고, 그렇지 않으면 디스크로부터 페이지를 버퍼에 읽어 온 후 리턴한다.

FIX_PAGE_COUNT

버퍼 초기화 이후 버퍼 매니저에 언두 페이지를 래치 없이 요청한 총 횟수를 나타낸다.

CREATE_PAGE_COUNT

버퍼 초기화 이후 트랜잭션이 버퍼 매니저에 페이지 생성을 요청한 총 횟수를 나타낸다. 이 요청에 대해 버퍼 매니저는 버퍼에서 빈 BCB를 확보한 후 페이지를 초기화 하여 리턴한다. 디스크 I/O는 이 연산에서 발생하지 않는다.

V\$USAGE

이 뷰는 데이터베이스에 존재하는 테이블과 인덱스가 사용하는 공간의 양을 보여준다. 이 뷰로부터 올바른 정보를 읽고 싶다면, 먼저 DBMS Stat 내장 프로시저를 실행해서 통계 정보를 수집해야 한다.

DBMS Stat 내장 프로시저에 대한 자세한 설명은 *Stored Procedures Manual*을 참고하기 바란다.

| Column name | Type | Description |
|---------------|---------|---------------------------|
| TYPE | CHAR(1) | 객체 종류 |
| TARGET_ID | BIGINT | 객체 식별자 |
| META_SPACE | BIGINT | 메타 정보를 저장하는 공간의 크기 |
| USED_SPACE | BIGINT | 실제 데이터 저장 공간의 크기 |
| AGEABLE_SPACE | BIGINT | Aging 대상 데이터가 차지하는 공간의 크기 |
| FREE_SPACE | BIGINT | 빈 공간의 크기 |

칼럼 정보**TYPE**

이는 객체의 종류를 나타낸다. 테이블은 T로, 인덱스는 I로 표시된다.

TARGET_ID

이는 객체의 식별자를 나타낸다. 테이블의 경우 그 테이블의 TABLE_OID, 인덱스의 경우 그 인덱스의 INDEX_ID가 표시된다. 이 칼럼과 SYSTEM.SYS_TABLES_ 메타 테이블의 TABLE_OID 또는 SYSTEM.SYS_INDICES_ 메타 테이블의 INDEX_ID와 조인 조회하여 대상 객체의 이름을 알아 낼 수 있다.

META_SPACE

이는 객체의 메타 정보를 저장하기 위해 사용되는 공간의 크기이다.

USED_SPACE

이는 객체의 실제 데이터를 저장하기 위해 사용되는 공간의 크기이다.

AGEABLE_SPACE

Altibase는 MVCC 기법을 사용하기 때문에, 데이터가 테이블 또는 인덱스로부터 삭제되더라도 예전 버전의 데이터가 잠시 유지된다. 이 칼럼의 값은 이런 데이터가 차지하는 공간의 크기이다.

FREE_SPACE

이는 아직 사용된 적이 없거나, 사용 후 반환되어 재활용 가능한 공간의 크기이다.

예제

```
iSQL> exec gather_database_stats();
SYSTEM_.SYS_TABLES_
SYSTEM_.SYS_COLUMNS_
SYSTEM_.SYS_DATABASE_
SYSTEM_.SYS_USERS_
SYSTEM_.SYS_DN_USERS_
SYSTEM_.SYS_TBS_USERS_
SYSTEM_.SYS_INDICES_
SYSTEM_.SYS_INDEX_COLUMNS_
...
Execute success.
```

```
iSQL> DESC V$USAGE;
[ ATTRIBUTE ]
```

| NAME | TYPE |
|---------------|---------|
| TYPE | CHAR(1) |
| TARGET_ID | BIGINT |
| META_SPACE | BIGINT |
| USED_SPACE | BIGINT |
| AGEABLE_SPACE | BIGINT |
| FREE_SPACE | BIGINT |

```
iSQL> select * from v$usage limit 10;
V$USAGE.TYPE V$USAGE.TARGET_ID V$USAGE.META_SPACE V$USAGE.USED_SPACE
V$USAGE.AGEABLE_SPACE V$USAGE.FREE_SPACE
```

| | | | | |
|---|-------|-----|-------|---|
| T | 65568 | 128 | 12672 | 0 |
| | 19968 | | | |
| I | 5 | 0 | 528 | 0 |
| | 1520 | | | |
| I | 6 | 0 | 528 | 0 |
| | 1520 | | | |
| I | 7 | 0 | 528 | 0 |
| | 1520 | | | |
| I | 8 | 0 | 528 | 0 |
| | 1520 | | | |
| T | 67976 | 464 | 66624 | 0 |
| | 63984 | | | |

| | | | | |
|-------------------|-------|-----|------|---|
| I | 9 | 0 | 3240 | 0 |
| | 856 | | | |
| I | 10 | 0 | 3240 | 0 |
| | 856 | | | |
| I | 11 | 0 | 3240 | 0 |
| | 856 | | | |
| T | 89648 | 848 | 2128 | 0 |
| | 29792 | | | |
| 10 rows selected. | | | | |

V\$VERSION

데이터베이스 버전 관련 정보를 보여준다.

| Column name | Type | Description |
|-------------------------|--------------|-------------------|
| PRODUCT_VERSION | VARCHAR(128) | 제품 버전 Ex) 5.5.1.1 |
| PKG_BUILD_PLATFORM_INFO | VARCHAR(128) | 패키지가 빌드된 플랫폼 |
| PRODUCT_TIME | VARCHAR(128) | 패키지가 빌드된 시간 |
| SM_VERSION | VARCHAR(128) | 저장 관리자 버전 |
| META_VERSION | VARCHAR(128) | 메타 테이블 버전 |
| PROTOCOL_VERSION | VARCHAR(128) | 통신 프로토콜 버전 |
| REPL_PROTOCOL_VERSION | VARCHAR(128) | 이중화 프로토콜 버전 |

칼럼 정보

PRODUCT_VERSION

Altibase 제품의 버전 정보를 나타낸다.

PKG_BUILD_PLATFORM_INFO

패키지가 빌드된 플랫폼의 정보를 나타낸다.

PRODUCT_TIME

패키지가 빌드된 날짜와 시간을 나타낸다.

SM_VERSION

저장 관리자의 버전을 나타낸다. 저장 구조가 변경될 때마다 버전이 변경된다.

META_VERSION

데이터베이스 정보를 관리하는 메타 테이블에 대한 버전을 나타낸다.

PROTOCOL_VERSION

데이터베이스의 통신을 위한 프로토콜 버전을 나타낸다.

REPL_PROTOCOL_VERSION

이중화를 위한 프로토콜 버전을 나타낸다.

V\$VOL_TABLESPACES

메모리에 생성된 휘발성 테이블스페이스 정보를 보여준다.

| Column name | Type | Description |
|------------------|--------------|--------------------------|
| SPACE_ID | INTEGER | 테이블스페이스 식별자 |
| SPACE_NAME | VARCHAR(512) | 테이블스페이스 이름 |
| SPACE_STATUS | INTEGER | 테이블스페이스 상태 |
| INIT_SIZE | BIGINT | 테이블스페이스의 초기 크기 (bytes) |
| AUTOEXTEND_MODE | INTEGER | 테이블스페이스의 자동 확장 모드 |
| NEXT_SIZE | BIGINT | 자동 확장시 확장되는 크기 (bytes) |
| MAX_SIZE | BIGINT | 테이블스페이스의 최대 크기 (bytes) |
| CURRENT_SIZE | BIGINT | 테이블스페이스의 현재 크기 (bytes) |
| ALLOC_PAGE_COUNT | BIGINT | 테이블스페이스의 전체 페이지 개수 |
| FREE_PAGE_COUNT | BIGINT | 테이블스페이스의 프리(Free) 페이지 개수 |

칼럼 정보**SPACE_STATUS**

테이블스페이스 상태 값이다. 자세한 내용은 V\$MEM_TABLESPACE_STATUS_DESC를 참고한다.

AUTOEXTEND_MODE

자동확장 (Autoextend) 모드 여부를 나타낸다. 1 이면 자동확장으로 설정된 상태이며, 1이 아니면 설정되지 않은 상태이다.

NEXTSIZE

자동 확장시 확장되는 크기 (bytes)이다.

MAXSIZE

테이블스페이스의 최대 크기 (bytes)이다.

CURRENT_SIZE

현재 테이블스페이스 크기 (bytes)를 나타낸다.

ALLOC_PAGE_COUNT

테이블스페이스가 가지고 있는 페이지의 개수를 나타낸다.

FREE_PAGE_COUNT

테이블스페이스의 빈 (free) 페이지 개수를 나타낸다.

V\$WAIT_CLASS_NAME

Altibase 서버상의 대기 이벤트들을 그룹화하기 위한 정보를 보여준다. 다양한 대기 이벤트들을 분류하기 위해 상위 개념인 대기 클래스를 사용하며 이 성능 뷰를 통하여 대기 클래스들을 확인할 수 있다.

| Column name | Type | Description |
|---------------|--------------|-------------|
| WAIT_CLASS_ID | INTEGER | 대기 클래스의 식별자 |
| WAIT_CLASS | VARCHAR(128) | 대기 클래스 이름 |

칼럼 정보**WAIT_CLASS_ID**

대기 이벤트의 클래스 식별자이다.

WAIT_CLASS

대기 이벤트 그룹화를 위한 상위 개념인, 대기 클래스를 나타낸다. Altibase는 대기 이벤트를 아래와 같이 8개의 대기 클래스로 분류한다.

| WAIT_CLASS_ID | WAIT_CLASS | Description |
|---------------|----------------|--|
| 0 | Other | 아래 클래스를 제외한 대기 이벤트를 포함한다. |
| 1 | Administrative | SYSDBA 권한의 명령 수행으로 인해 사용자가 대기하게 되는 대기 이벤트를 포함한다. |
| 2 | Configuration | 데이터베이스 자원에 대한 부적절한 설정에 관련된 대기 이벤트를 포함한다. |
| 3 | Concurrency | 데이터베이스 내부 자원과 관련된 대기 이벤트를 포함한다. |
| 4 | Commit | REDO 로그가 로그 파일에 동기화되는 것과 관련된 대기 이벤트를 포함한다. |
| 5 | Idle | 세션의 작업이 요청되기를 기다리며 대기하는 대기 이벤트를 포함한다. |
| 6 | User I/O | 사용자 I/O 관련 대기이벤트를 포함한다. |
| 7 | System I/O | 시스템 I/O 관련 대기 이벤트를 포함한다. |
| 8 | Replication | 이중화에서 사용하는 대기 이벤트를 포함하는 클래스이다. |

V\$XID

DBMS내 분산 트랜잭션의 식별자인 XID의 목록을 보여준다. XA에서 분산 트랜잭션 식별자는 분산 트랜잭션이 시작될 때 TM (Transaction Manager) 내부에서 생성되며, 데이터베이스 노드들인 RM (Resource Manager)에게 전달한다.

| Column name | Type | Description |
|------------------|--------------|--------------------------------|
| XID_VALUE | VARCHAR(256) | XID 값을 문자열로 반환 |
| ASSOC_SESSION_ID | INTEGER | XID 객체와 연계된 세션의 식별자 |
| TRANS_ID | INTEGER | XID 객체에 있는 분산 트랜잭션 식별자 |
| STATE | VARCHAR(24) | XID 객체의 상태 |
| STATE_START_TIME | INTEGER | XID 객체의 상태가 설정된 시간 |
| STATE_DURATION | BIGINT | XID 객체의 상태가 설정된 이후 경과된 시간 |
| TX_BEGIN_FLAG | VARCHAR(9) | 트랜잭션 시작 여부를 가리키는 XID 객체 내의 플래그 |
| REF_COUNT | INTEGER | XID 객체를 현재 참조한 있는 횟수 |

칼럼 정보

XID_VALUE

문자열로 표현한 XID 값이다.

ASSOC_SESSION_ID

XID 객체와 연계된 세션의 식별자로써, 이 세션은 해당 XID를 XA_START 시킨 세션이다.

TRANS_ID

XID 객체 내의 분산 트랜잭션의 식별자이다.

STATE

XID 객체의 수행 상태를 나타낸다. 가능한 값은 다음과 같다.

- IDLE: 해당 XID에 연계된 세션이 없는 상태
- ACTIVE: 해당 XID에 연계된 세션이 있는 상태. 즉 XA_START된 경우
- PREPARED: 2PC (Phase Commit) 과정에서 prepare 명령을 수신한 상태
- HEURISTICALLY_COMMITTED: DBMS가 XID의 트랜잭션 브랜치를 강제로 커밋한 상태
- HEURISTICALLY_ROLLED_BACK: DBMS가 XID의 트랜잭션 브랜치를 강제로 롤백한 상태
- NO_TX: XID가 초기화된 상태이거나 XID의 트랜잭션 브랜치를 커밋 또는 롤백한 상태

STATE_START_TIME

XID 객체의 수행 상태가 설정된 시간을 나타낸다.

STATE_DURATION

XID 객체의 상태가 설정된 이후 경과 시간을 나타낸다.

TX_BEGIN_FLAG

트랜잭션 브랜치가 RM에서 시작되었는지 여부를 나타내는 XID 객체 내의 플래그이다.

- BEGIN: 시작된 상태
- NOT BEGIN: 시작되지 않은 상태

REF_COUNT

해당 XID 객체가 현재 참조된 횟수를 나타낸다.

2.샘플 스키마

이 부록은 Altibase 매뉴얼 내의 예제에서 전반적으로 사용된 스키마에 대한 정보를 제공한다.

예제 테이블 정보

스크립트 파일

스키마 생성파일은 \$ALTIABSE_HOME/sample/APRE/schema/schema.sql 파일로 제공된다. 이 파일은 Altibase 매뉴얼에서 사용된 테이블을 생성하고 예제 데이터를 삽입하는 파일이다. 따라서 매뉴얼에 기술되어 있는 예제를 실행하고자 한다면 먼저 제공된 스크립트 파일을 수행해야 한다.

샘플 스키마

기능: 고객과 주문 관리

테이블: employees, departments, customers, orders, goods

사원(employees) 테이블

기본 키: 사원번호(eno)

| 칼럼명 | 데이터 타입 | 설명 | 기타 |
|-------------|--------------|------|----------------------|
| eno | INTEGER | 사원번호 | PRIMARY KEY |
| e_lastname | CHAR(20) | 사원성 | NOT NULL |
| e_firstname | CHAR(20) | 사원이름 | NOT NULL |
| emp_job | VARCHAR(15) | 직책 | NULL 허용 |
| emp_tel | CHAR(15) | 전화번호 | NULL 허용 |
| dno | SMALLINT | 부서번호 | NULL 허용, INDEX ASC |
| salary | NUMBER(10,2) | 월급 | NULL 허용, DEFAULT 0 |
| sex | CHAR(1) | 성별 | NULL 허용 |
| birth | CHAR(6) | 생일 | NULL 허용 |
| join_date | DATE | 입사날짜 | NULL 허용 |
| status | CHAR(1) | 지위 | NULL 허용, DEFAULT 'H' |

부서(departments) 테이블

기본 키: 부서번호(dno)

| 칼럼명 | 데이터 타입 | 설명 | 기타 |
|--------------|----------|------|-------------|
| dno | SMALLINT | 부서번호 | PRIMARY KEY |
| dname | CHAR(30) | 부서명 | NOT NULL |
| dep_location | CHAR(15) | 부서위치 | NULL 허용 |

| 칼럼명 | 데이터 타입 | 설명 | 기타 |
|--------|---------|-------|--------------------|
| mgr_no | INTEGER | 관리자번호 | NULL 허용, INDEX ASC |

고객(customers) 테이블

기본 키: 주민등록번호(cno)

| 칼럼명 | 데이터 타입 | 설명 | 기타 |
|-------------|-------------|--------|-------------|
| cno | CHAR(14) | 주민등록번호 | PRIMARY KEY |
| c_lastname | CHAR(20) | 고객성 | NOT NULL |
| c_firstname | CHAR(20) | 고객이름 | NOT NULL |
| cus_job | VARCHAR(20) | 직업 | NULL 허용 |
| cus_tel | NIBBLE(15) | 전화번호 | NOT NULL |
| sex | CHAR(1) | 성별 | NOT NULL |
| birth | CHAR(6) | 생일 | NULL 허용 |
| postal_cd | VARCHAR(9) | 우편번호 | NULL 허용 |
| address | VARCHAR(60) | 주소 | NULL 허용 |

주문(orders) 테이블

기본 키: 주문번호와 주문일자 (ono, order_date)

| 칼럼명 | 데이터 타입 | 설명 | 기타 |
|--------------|----------|------------|---|
| ono | BIGINT | 주문번호 | PRIMARY KEY |
| order_date | DATE | 주문일자 | PRIMARY KEY |
| eno | INTEGER | 판매사원 | NOT NULL, INDEX ASC |
| cno | BIGINT | 고객주민 번호 | NOT NULL, INDEX DESC |
| gno | CHAR(10) | 상품번호 | NOT NULL, INDEX ASC |
| qty | INTEGER | 주문수량 | NULL 허용, DEFAULT 1 |
| arrival_date | DATE | 도착예정 일자 | NULL 허용 |
| processing | CHAR(1) | 주문상태 | NULL 허용 , O: ORDER, R: PREPARE, D: DELIVERY, C: COMPLETE, DEFAULT 'O' |

상품(goods) 테이블

기본 키: 상품번호(gno)

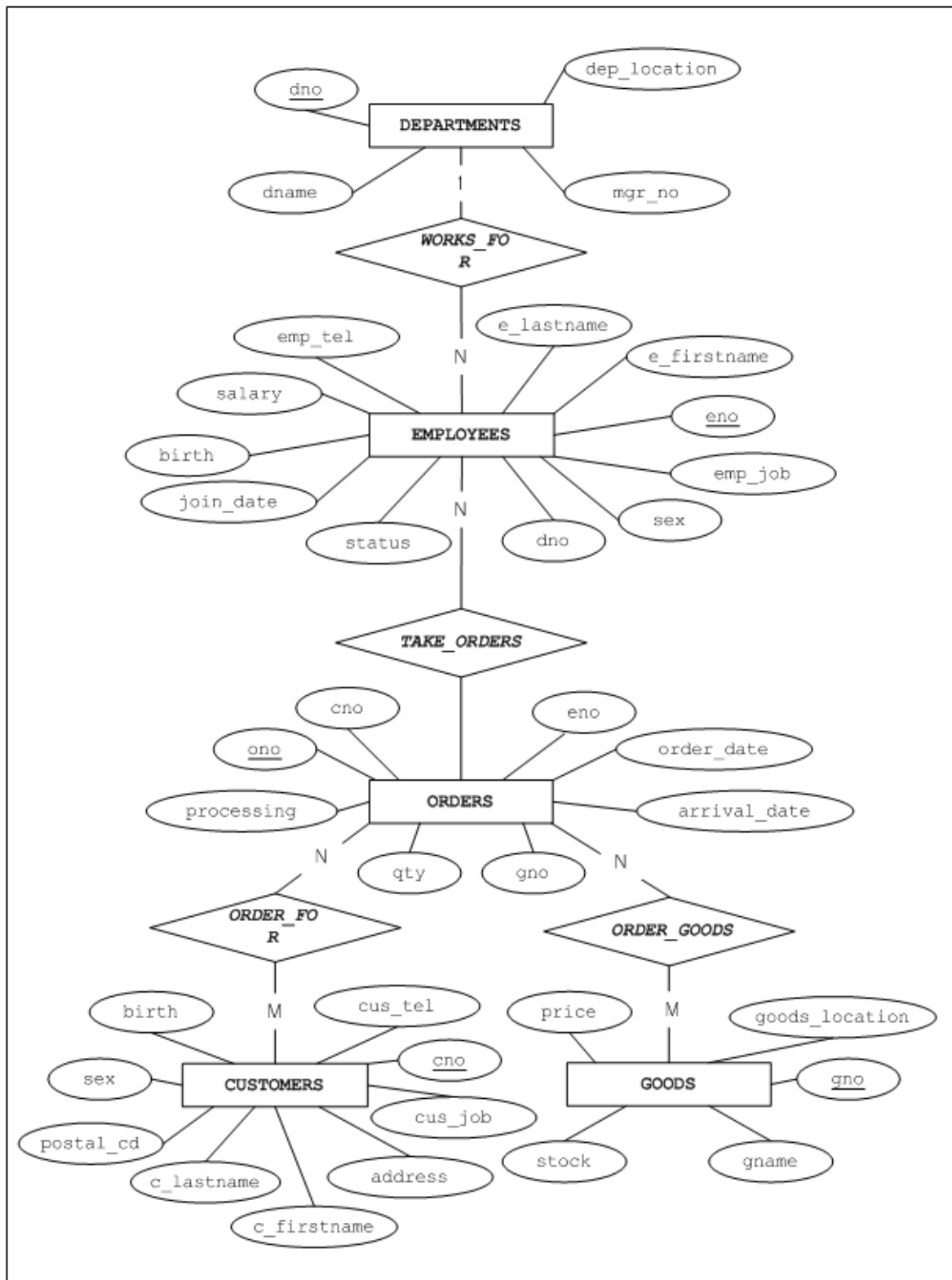
| 칼럼명 | 데이터 타입 | 설명 | 기타 |
|----------------|---------------|------|--------------------|
| gno | CHAR(10) | 상품번호 | PRIMARY KEY |
| gname | CHAR(20) | 상품명 | NOT NULL, UNIQUE |
| goods_location | CHAR(9) | 보관위치 | NULL 허용 |
| stock | INTEGER | 보관수량 | NULL 허용, DEFAULT 0 |
| price | NUMERIC(10,2) | 원가 | NULL 허용 |

dual 테이블

레코드 크기: 1개

| 칼럼명 | 데이터 타입 | 설명 | 기타 |
|-------|---------|----|----|
| DUMMY | CHAR(1) | | |

E-R 다이어그램과 샘플 데이터**E-R 다이어그램**



샘플 데이터

직원 테이블

```
isQL> select * from employees;
```

| ENO | E_LASTNAME | E_FIRSTNAME | EMP_JOB |
|-------------|------------|-------------|----------|
| 1 | Moon | Chan-seung | CEO |
| 01195662365 | 3002 | M | R |
| 2 | Davenport | Susan | designer |

| | | | | | | |
|-------------|-----------|------|-----------|--------|-------------|---|
| 0113654540 | | 1500 | F | 721219 | 18-NOV-2009 | H |
| 3 | Kobain | | Ken | | engineer | |
| 0162581369 | 1001 | 2000 | M | 650226 | 11-JAN-2010 | H |
| 4 | Foster | | Aaron | | PL | |
| 0182563984 | 3001 | 1800 | M | 820730 | | H |
| 5 | Ghorbani | | Farhad | | PL | |
| 01145582310 | 3002 | 2500 | M | | 20-DEC-2009 | H |
| 6 | Momoi | | Ryu | | programmer | |
| 0197853222 | 1002 | 1700 | M | 790822 | 09-SEP-2010 | H |
| 7 | Fleischer | | Gottlieb | | manager | |
| 0175221002 | 4002 | 500 | M | 840417 | 24-JAN-2004 | H |
| 8 | Wang | | Xiong | | manager | |
| 0178829663 | 4001 | | M | 810726 | 29-NOV-2009 | H |
| 9 | Diaz | | Curtis | | planner | |
| 0165293668 | 4001 | 1200 | M | 660102 | 14-JUN-2010 | H |
| 10 | Bae | | Elizabeth | | programmer | |
| 0167452000 | 1003 | 4000 | F | 710213 | 05-JAN-2010 | H |
| 11 | Liu | | Zhen | | webmaster | |
| 0114553206 | 1003 | 2750 | M | | 28-APR-2011 | H |
| 12 | Hammond | | Sandra | | sales rep | |
| 0174562330 | 4002 | 1890 | F | 810211 | 14-DEC-2009 | H |
| 13 | Jones | | Mitch | | PM | |
| 0187636550 | 1002 | 980 | M | 801102 | | H |
| 14 | Miura | | Yuu | | PM | |
| 0197664120 | 1003 | 2003 | M | | | H |
| 15 | Davenport | | Jason | | webmaster | |
| 0119556884 | 1003 | 1000 | M | 901212 | | H |
| 16 | Chen | | wei-wei | | manager | |
| 0195562100 | 1001 | 2300 | F | 780509 | | H |
| 17 | Fubuki | | Takahiro | | PM | |
| 0165293886 | 2001 | 1400 | M | 781026 | 07-MAY-2010 | H |
| 18 | Huxley | | John | | planner | |
| 01755231044 | 4001 | 1900 | M | | 30-OCT-2007 | H |
| 19 | Marquez | | Alvar | | sales rep | |
| 0185698550 | 4002 | 1800 | M | | 18-NOV-2010 | H |
| 20 | Blake | | william | | sales rep | |
| 01154112366 | 4002 | | M | | 18-NOV-2006 | H |

20 rows selected.

부서 테이블

```
isQL> select * from departments;
```

| DNO | DNAME | DEP_LOCATION | MGR_NO |
|------|-----------------------------|--------------|--------|
| 1001 | RESEARCH DEVELOPMENT DEPT 1 | New York | 16 |
| 1002 | RESEARCH DEVELOPMENT DEPT 2 | Sydney | 13 |
| 1003 | SOLUTION DEVELOPMENT DEPT | Osaka | 14 |
| 2001 | QUALITY ASSURANCE DEPT | Seoul | 17 |
| 3001 | CUSTOMERS SUPPORT DEPT | London | 4 |
| 3002 | PRESALES DEPT | Peking | 5 |
| 4001 | MARKETING DEPT | Brasilia | 8 |
| 4002 | BUSINESS DEPT | Palo Alto | 7 |

8 rows selected.

고객 테이블

```

isQL> select * from customers;
CNO                C_LASTNAME          C_FIRSTNAME
-----
CUS_JOB            CUS_TEL              SEX  BIRTH  POSTAL_CD
-----
ADDRESS
-----
1                Sanchez              Estevan
engineer          0514685282          M  720828  90021
2100 Exposition Boulevard Los Angeles USA
2                Martin              Pierre
doctor           023242121           M  821215  V6T 1F2
4712 West 10th Avenue Vancouver BC Canada
3                Morris              Gabriel
designer          023442542           M  811111  75010
D914 Puteaux Ile-de-France France
4                Park                Soo-jung
engineer          022326393           F  840305  609-735
Geumjeong-Gu Busan South Korea
5                Stone               James
webmaster        0233452141          M  821012  6060
142 Francis Street Western Australia AUS
6                Dureault            Phil
WEBPD            025743215           M  810209  H1R-2W1
1000 Rue Rachel Est Montreal Canada
7                Lalani              Yasmin
planner          023143366           F  821225  156772
176 Robinson Road Singapore
8                Kanazawa            Tsubasa
PD               024721114           M  730801  141-0031
2-4-6 Nishi-Gotanda Shinagawa-ku Tokyo JP
9                Yuan                Ai
designer          0512543734          F  690211  200020
10th Floor No. 334 Jiujiang Road Shanghai
10              Nguyen              Anh Dung
                0516232256          M  790815  70000
8A Ton Duc Thang Street District 1 HCMC Vietnam
11              Sato                Naoki
manager          027664545           M  810101  455-8205
3-23 Oye-cho Minato-ku Nagoya Aichi Japan
12              Rodriguez           Aida
banker           023343214           F  810905  76152
3484 Taylor Street Dallas TX USA
13              white               Crystal
engineer         022320119           F  801230  WC2B 4BM
12th Floor Five Kemble Street London UK
14              Kim                 Cheol-soo
banker           024720112           M  660508  135-740
222-55 Samsung-dong Gangnam-gu Seoul Korea
15              Fedorov             Fyodor
manager          0518064398          M  750625  50696
No 6 Leboh Ampang 50100 Kuala Lumpur Malaysia
16              Lefebvre            Daniel

```

```

planner          027544147      M  761225  21004
Chaussee de wavre 114a 1050 Brussels Belgium
17              Yoshida          Daichi
                023543541      M  811001  530-0100
2-7 3-Chome-Kita Tenjinbashi Kita-ku Osaka
18              Zhang            Bao
engineer         024560207      F  840419  100008
2 Chaoyang Men Wai Street Chaoyang Beijing
19              Pahlavi          Saeed
                022371234      M  741231  20037
3300 L Street NW Washington DC USA
20              Dubois           Alisee
webmaster        024560002      F  860405  1357
Chemin de Messidor 7-6 CH-1006 Lausanne Suisse
20 rows selected.

```

주문 테이블

```
isQL> select * from orders;
```

| ONO | | ORDER_DATE | ENO | CNO |
|------------|------|--------------|------------|-----|
| ----- | | | | |
| GNO | QTY | ARRIVAL_DATE | PROCESSING | |
| ----- | | | | |
| 11290007 | | 29-NOV-2011 | 12 | 3 |
| A111100002 | 70 | 02-DEC-2011 | C | |
| 11290011 | | 29-NOV-2011 | 12 | 17 |
| E111100001 | 1000 | 05-DEC-2011 | D | |
| 11290100 | | 29-NOV-2011 | 19 | 11 |
| E111100001 | 500 | 07-DEC-2011 | D | |
| 12100277 | | 10-DEC-2011 | 19 | 5 |
| D111100008 | 2500 | 12-DEC-2011 | C | |
| 12300001 | | 01-DEC-2011 | 19 | 1 |
| D111100004 | 1000 | 02-JAN-2012 | P | |
| 12300002 | | 29-DEC-2011 | 12 | 2 |
| C111100001 | 300 | 02-JAN-2012 | P | |
| 12300003 | | 29-DEC-2011 | 20 | 14 |
| E111100002 | 900 | 02-JAN-2012 | P | |
| 12300004 | | 30-DEC-2011 | 20 | 15 |
| D111100002 | 1000 | 02-JAN-2012 | P | |
| 12300005 | | 30-DEC-2011 | 19 | 4 |
| D111100008 | 4000 | 02-JAN-2012 | P | |
| 12300006 | | 30-DEC-2011 | 20 | 13 |
| A111100002 | 20 | 02-JAN-2012 | P | |
| 12300007 | | 30-DEC-2011 | 12 | 7 |
| D111100002 | 2500 | 02-JAN-2012 | P | |
| 12300008 | | 30-DEC-2011 | 20 | 11 |
| D111100011 | 300 | 02-JAN-2012 | P | |
| 12300009 | | 30-DEC-2011 | 20 | 19 |
| D111100003 | 500 | 02-JAN-2012 | P | |
| 12300010 | | 30-DEC-2011 | 19 | 16 |
| D111100010 | 2000 | 02-JAN-2012 | P | |
| 12300011 | | 30-DEC-2011 | 20 | 15 |
| C111100001 | 1000 | 02-JAN-2012 | P | |
| 12300012 | | 30-DEC-2011 | 12 | 3 |
| E111100012 | 1300 | 02-JAN-2012 | P | |

| | | | | |
|------------|-------|-------------|----|----|
| 12300013 | | 30-DEC-2011 | 20 | 6 |
| C111100001 | 5000 | 02-JAN-2012 | P | |
| 12300014 | | 30-DEC-2011 | 12 | 12 |
| F111100001 | 800 | 02-JAN-2012 | P | |
| 12310001 | | 31-DEC-2011 | 20 | 15 |
| A111100002 | 50 | 09-DEC-2011 | O | |
| 12310002 | | 31-DEC-2011 | 12 | 10 |
| D111100008 | 10000 | 03-JAN-2012 | O | |
| 12310003 | | 31-DEC-2011 | 20 | 18 |
| E111100009 | 1500 | 03-JAN-2012 | O | |
| 12310004 | | 31-DEC-2011 | 19 | 5 |
| E111100010 | 5000 | 08-JAN-2012 | O | |
| 12310005 | | 31-DEC-2011 | 20 | 14 |
| E111100007 | 940 | 03-JAN-2012 | O | |
| 12310006 | | 31-DEC-2011 | 20 | 2 |
| D111100004 | 500 | 03-JAN-2012 | O | |
| 12310007 | | 31-DEC-2011 | 12 | 19 |
| E111100012 | 1400 | 03-JAN-2012 | O | |
| 12310008 | | 31-DEC-2011 | 19 | 1 |
| D111100003 | 100 | 03-JAN-2012 | O | |
| 12310009 | | 31-DEC-2011 | 12 | 5 |
| E111100013 | 500 | 03-JAN-2012 | O | |
| 12310010 | | 31-DEC-2011 | 20 | 6 |
| D111100010 | 1500 | 03-JAN-2012 | O | |
| 12310011 | | 31-DEC-2011 | 19 | 15 |
| E111100012 | 10000 | 03-JAN-2012 | O | |
| 12310012 | | 31-DEC-2011 | 19 | 1 |
| C111100001 | 250 | 03-JAN-2012 | O | |

30 rows selected.

상품 테이블

```

iSQL> SELECT * FROM goods;
GOODS.GNO      GOODS.GNAME      GOODS.GOODS_LOCATION      GOODS.STOCK
-----
GOODS.PRICE
-----
A111100001      IM-300          AC0001                    1000
78000
A111100002      IM-310          DD0001                    100
98000
B111100001      NT-H5000        AC0002                    780
35800
C111100001      IT-U950         FA0001                    35000
7820.55
C111100002      IT-U200         AC0003                    1000
9455.21
D111100001      TM-H5000        AC0004                    7800
12000
D111100002      TM-T88          BF0001                    10000
72000

D111100003      TM-L60          BF0002                    650
45100

```

| | | | |
|-----------------------|---------|--------|-------|
| D111100004 96200 | TM-U950 | DD0002 | 8000 |
| D111100005 23000 | TM-U925 | AC0005 | 9800 |
| D111100006 57400 | TM-U375 | EB0001 | 1200 |
| D111100007 84500 | TM-U325 | EB0002 | 20000 |
| D111100008 10000 | TM-U200 | AC0006 | 61000 |
| D111100009 50000 | TM-U300 | DD0003 | 9000 |
| D111100010 36800 | TM-U590 | DD0004 | 7900 |
| D111100011 45600 | TM-U295 | FA0002 | 1000 |
| E111100001 2290.54 | M-T245 | AC0007 | 900 |
| E111100002 7527.35 | M-150 | FD0001 | 4300 |
| E111100003 2300.55 | M-180 | BF0003 | 1000 |
| E111100004 5638.76 | M-190G | CE0001 | 88000 |
| E111100005 1450.5 | M-U310 | CE0002 | 11200 |
| E111100006 2338.62 | M-T153 | FD0002 | 900 |
| E111100007 966.99 | M-T102 | BF0004 | 7890 |
| E111100008 1000.54 | M-T500 | EB0003 | 5000 |
| E111100009 3099.88 | M-T300 | FA0003 | 7000 |
| E111100010 9200.5 | M-T260 | AC0008 | 4000 |
| E111100011 9832.98 | M-780 | AC0009 | 9800 |
| E111100012 3566.78 | M-U420 | CE0003 | 43200 |
| E111100013 1295.44 | M-U290 | FD0003 | 12000 |
| F111100001 100000 | AU-100 | AC0010 | 10000 |

30 rows selected.

DUAL 테이블

```

iSQL> SELECT * FROM dual;
DUAL.X
-----
X
1 row selected.

```

