Table of Contents

Altibase 7.2.0.0.1 Release Notes

시스템 요구사항

하드웨어 최저 사양

- 1GB RAM (권장: 2GB)
- 1 CPU (권장: 2 CPUs)
- 4GB 하드 디스크 여유 공간 (권장: 12GB)

운영 체제 및 플랫폼

Altibase 7.2.0.0.1 는 아래 표에 나열된 운영체제와 플랫폼 상에서 운영 가능하다.

	Altibase 서버	Altibase 클라이언트	소프트웨어 요구사항
Linux x86-64			
Red Hat Enterprise Linux 6 Red Hat Enterprise Linux 7 Red Hat Enterprise Linux 8	●	●	- GNU glibc 2.12 이상
Linux on Power			
Red Hat Enterprise Linux 6	●	●	- GNU glibc 2.12 이상
Linux on Power (Little Endian)			
Red Hat Enterprise Linux 7	●	●	- GNU glibc 2.17 이상
HP-UX Itanium (IA-64)			
HP-UX 11.31	●	●	
Microsoft Windows (x64)			
Microsoft Windows 2008	-	●	

> Altibase 서버/클라이언트 모두 64-bit 만 지원한다.
> Microsoft Windows 는 Altibase 클라이언트만 지원한다.
>
> Red Hat Enterprise Linux 6, 7, 8 마이너 버전에 대해 호환성을 보장한다.

릴리스 정보

Altibase 7.2 의 새로운 기능

기능 개선 - SQL 확장

CREATE QUEUE 및 ALTER QUEUE 구문에 DELETE 절 추가

큐(QUEUE) 테이블에 DELETE 문 허용 여부를 설정하는 DELETE 절이 추가되었다.

DELETE OFF로 DELETE 문을 허용하지 않으면 DELETE 문을 허용한 경우보다 DEQUEUE 병렬 수행 성능이 향상된다. 구문 사용 방법은 Altibase 7.2 SQL Reference 매뉴얼 을 참고한다. 관련하여 성능 뷰 V$QUEUE_DELETE_OFF 가 추가되었다.

범위 파티션드 객체에 파티션 추가 연산 지원

범위 파티셔드 테이블에 파티션 추가(ADD PARTITION) 구문을 지원한다. 이 기능 추가로 기본 파티션 없는 범위 파티션드 테이블 생성이 가능하다.

Altibase 7.2 에서 범위 파티션드 테이블 생성 시 주의 사항

- Altibase 7.2 에서는 기본 파티션이 없는 범위 파티션드 테이블을 생성할 수 있다.

  참고로 기본 파티션이 없는 범위 파티션드 테이블을 생성하면 SYS_TABLE_PARTITIONS_에서 PARTITION_NAME 이 없는 파티션이 추가로 생성된다.
- 범위 파티션드 객체에서 파티션 추가는 기본 파티션이 없는 범위 파티션드 테이블에서만 사용할 수 있다.
- 기본 파티션이 없는 범위 파티션드 테이블은 기본 파티션 추가 연산을 수행할 수 없다.
- 기본 파티션이 있는 범위 파티션드 테이블은 기본 파티션 삭제 연산을 수행할 수 없다.
- 기본 파티션이 없는 범위 파티션드 테이블은 CONJOIN/DISJOIN 구문을 사용할 수 없다.
- 범위 파티션드 테이블이 이중화 대상 테이블인 경우 파티션 추가 연산을 수행할 수 없다.

기능 개선 - 응용 프로그램 개발 인터페이스

JDBC API Specification 4.2 부분 지원 (PROJ-2707)

Altibase 7.2 에서 JDBC API Specification 4.2를 부분적으로 지원한다.

Altibase 7.2 JDBC 드라이버는 JRE 1.8 이상에서 동작한다. Altibase 7.2 JDBC 드라이버에서 지원하는 JDBC 4.2 API는 Altibase 7.2 JDBC User's Manual 에서 확인할 수 있다. 변경 사항 및 호환성 이슈는 Altibase JDBC 7.2 변경 사항 및 호환성 이슈에서 확인할 수 있다.

- **Auto-loading of JDBC driver class**

  명시적으로 Class.forName() 클래스를 로딩할 필요없이 META-INF/services/java.sql.Driver 파일을 이용한 자동 드라이버 로딩 기능 지원
- **Wrapper Pattern Support**

  프록시에서 구현 객체에 대한 참조를 얻는 JDBC 4.0 표준 인터페이스를 지원한다. 커넥션풀 등에서 생성하는 프록시 객체에서 JDBC 객체를 획득할 수 있다.

```
try (Connection sWrappedCon = dbPool.getConnection()) {
    if (sWrappedCon.isWrapperFor(AltibaseConnection.class)) {
        AltibaseConnection connection = sWrappedCon.unwrap(AltibaseConnection.class);
        ...
        ...
}
```

- **National Character Set Support**

  JDBC 4.0 스펙인 표준 다국어 처리 인터페이스 지원

- **Aborting Connections**

  비동기적으로 데이터베이스와의 물리적 연결을 종료하는 Connection.abort() 인터페이스 지원

- **Standard Socket Network Timeout API Support**

  데이터베이스 서버로부터 소켓 응답 대기 시간을 설정하는 표준 인퍼페이스 Connection.setNetworkTimeout() 지원

- **Connection Management Enhancements**

  Validation Query없이 Connection 객체에서 유효성 검사를 수행하는 Connection.isValid() 지원

- **Large Update Counts Support**

  대용량 레코드 업데이트를 위한 executeLargeUpdate(), executeLargeBatch() 지원

- **Set Client Information Support**

  Connection.setClientInfo()를 이용한 클라이언트 어플리케이션 속성(name) 설정 지원

- **java.sql.SQLType interface Support**

  JDBC 4.2 표준 인터페이스 java.sql.SQLType을 구현한 AltibaseJDBCType 지원

JDK 레벨에서 향상된 기능들은 Altibase JDBC 7.2 에서도 대부분 사용할 수 있다.

- Try-with-resources 구문을 통한 자동 JDBC 리소스 해제

```
try (Statement stmt = con.createStatement()) {
      ResultSet rs = stmt.executeQuery(query);
      while (rs.next()) {
        String coffeeName = rs.getString("aaa");
        int supplierID = rs.getInt("bbb");
      }
    }
}
```

- SQLException에 Enhanced for-each loop 사용

```
catch(SQLException ex) {
     for(Throwable e : ex ) {
        LOG.error("Error occurred: " + e);
     }
}
```

- 커넥션풀 등에서 생성되는 proxy객체에서 실체 JDBC 객체 획득

```
try (Connection sWrappedCon = dbPool.getConnection()) {
    if (sWrappedCon.isWrapperFor(AltibaseConnection.class)) {
        AltibaseConnection connection = sWrappedCon.unwrap(AltibaseConnection.class);
        ...
        ...
}
```

기능 개선 - 유틸리티

altiComp 커밋 카운트 설정 기능 추가

커밋(commit) 카운트를 설정할 수 있는 프로퍼티 COUNT_TO_COMMIT가 추가되었다. 관련 내용은 Altibase 7.2 Utilities Manual 에서 확인할 수 있다.

Altibase 서버 성능 및 안정성 향상

OLTP Scalability 성능 향상(TASK-7073)

- Linux x86-64 CPU 코어 수 24코어 이상에서 조회 트랜잭션 성능 저하 현상 개선
- 메모리 DB 삭제(DELETE) 트랜잭션 성능 향상을 위해 로깅 구조 개선
- 디스크 DB 변경 트랜잭션 성능 향상을 위해 In-place MVCC 동작 방식 개선
- 테이블 잠금(TABLE LOCK) 병목 개선
- INSERT/UPDATE 트랜잭션 처리 시 불필요한 트랜잭션 로그 기록을 제거
- 트랜잭션 로그파일 압축 시 메모리 할당/해제 병목 개선
    - 이와 관련한 영향도 확인
- 휘발성(Volatile) 메모리 DB 트랜잭션 성능 향상
- 커밋 병목 및 가비지 콜렉션 쓰레드 병목 개선
    - 트랜잭션 커밋 후 테이블 정보 업데이트 병목 개선
- 메모리 DB 트랜잭션 성능 향상
    - 디스크 읽기를 유발하는 함수의 병목을 제거
    - Group Commit Log 기능 추가

트랜잭션 로그 기록 성능 향상(TASK-6983)

로그 압축 알고리즘을 압축 속도가 빠른 LZ4 로 변경하였다.

휘발성/비휘발성 메모리 DB 트랜잭션 성능 향상

메모리 테이블 객체 식별자 추적 단계를 간소화하여 휘발성/비휘발성 메모리 DB 트랜잭션 성능이 향상되었다.

언두(undo) 테이블스페이스 재사용 안정성 향상

언두 테이블스페이스와 디스크 인덱스의 불필요한 관계를 제거하여 버그 발생 위험 요소 제거하였다. 디스크 페이지 공간 효율 개선으로 관련 프로퍼티들의 기본값 및 최대값이 변경되었다.

- INDEX_INITTRANS 최대값이 30에서 50으로 변경
- INDEX_MAXTRANS 기본값과 최대값이 30에서 50으로 변경

PARTITIONED TABLE에 대한 LIMIT FOR UPDATE 성능 개선

파티션드 테이블에 대한 LIMIT 연산 시 불필요한 범위 읽기 연산을 제거하여 성능을 개선하였다.

아래 조건을 모두 만족하는 경우에 성능 향상이 있다.

- LIMIT 절의 START VALUE가 1인 경우

```
-- START VALUE가 1 의 예
SELECT * FROM T1 LIMIT 1;      -- 내부적으로 limit start 1, count 1
SELECT * FROM T1 LIMIT 100     -- 내부적으로 limit start 1, count 100
SELECT * FROM T1 LIMIT 1, 30   -- 내부적으로 limit start 1, count 30
```

- PROJECT 하위 노드가 파티션드 테이블의 각각의 파티션에 대한 스캔을 관리하는 노드(PARTITION-COORDINATOR) 이며
- PARTITION-COORDINATOR 노드의 모든 노드가 SCAN 인 경우

조건을 모두 만족하는 실행 계획은 아래와 같은 형태이다.

```
-------------------------------------------------------------
PROJECT ( COLUMN_COUNT: 2, TUPLE_SIZE: 8, COST: 467.05 )
 PARTITION-COORDINATOR ( TABLE: SYS.T1, PARTITION: 4/4, FULL SCAN, ACCESS: 1, COST:
467.05 )
  SCAN ( PARTITION: P4, FULL SCAN, ACCESS: 0, COST: 116.76 )
  SCAN ( PARTITION: P3, FULL SCAN, ACCESS: 0, COST: 116.76 )
  SCAN ( PARTITION: P2, FULL SCAN, ACCESS: 0, COST: 116.76 )
  SCAN ( PARTITION: P1, FULL SCAN, ACCESS: 1, COST: 116.76 )
-------------------------------------------------------------
```

서브쿼리의 인라인 뷰에 ORDER BY절 사용 시 SQL 성능 개선

조건절(WHERE, HAVING 절)에서 사용한 서브쿼리의 인라인뷰에 ORDER BY절이 있는 경우 OBYE(Order By Elimination, 불필요한 ORDER BY 제거) 쿼리 변환을 적용하여 SQL 성능이 향상되었다.

- SQL 사용 예

```
SELECT *
  FROM T1
 WHERE I1 IN (SELECT /*+ NO_UNNEST */I1
                FROM (SELECT *
                        FROM T2
                       ORDER BY I2, I3));
```

> 이 영향을 받는 *SQL*의 실행 계획이 변경된다. *'SUBQUERY FILTER* 안에 *SORT* 플랜 노드 없어진다.

스칼라 서브쿼리(Scalar subquery) 성능 개선

스칼라 서브쿼리 결과가 단일 레코드인지 확인 과정에서 발생하는 오버헤드를 제거하여 성능을 개선하였다.

Altibase 이중화 성능 향상

이중화 Sender 성능 향상

- 압축 로그에서 이중화에 필요한 로그만 압축 해제하는 기능 추가
- xLog 압축 알고리즘을 LZO에서 LZ4로 변경

변경 사항 및 호환성 이슈

DBA와 개발자가 알아야 할 추가, 변경, 및 제거된 기능을 아래에서 설명한다.

Altibase JDBC 4.2 관련 변경 사항 및 호환성 이슈

Altibase JDBC 4.2는 Altibase JDBC 3.0 에 대해 하위 호환성을 보장하지만 일부 인터페이스의 경우 JDBC API Specification 4.2에 따라 동작이 변경되었다.

미지원 기능에 대한 예외 처리 클래스 변경

다음 인터페이스에 대한 예외 처리 클래스가 SQLException에서 SQLFeatureNotSupportedException으로 변경되었다. SQLFeatureNotSupportedException은 SQLException의 하위 클래스이므로 기존 사용자 프로그램은 수정 없이 그대로 동작한다.

- Altibase.jdbc.driver.AltibaseConnection
  - setTypeMap(Map)
- Altibase.jdbc.driver.AltibaseStatement
  - setCursorName(String)
- Altibase.jdbc.driver.AltibasePreparedStatement
  - setArray(int, Array)
  - setRef(int, Ref)
  - setURL(int, URL)
  - setUnicodeStream(int, InputStream, int)
- Altibase.jdbc.driver.Blob
  - position(Blob, long)
  - position(byte[], long)
- Altibase.jdbc.driver.Clob
  - position(Clob, long)
  - position(String, long)
- Altibase.jdbc.driver.CallableStatement
  - getArray(int)
  - getObject(int, Map)
  - getRef(int)
  - getURL(int)
- Altibase.jdbc.driver.AltibaseDatabaseMetaData
  - getColumnPrivileges(String, String, String, String)
  - getUDTs(String, String, String, int[])
- Altibase.jdbc.driver.AltibaseResultSet
  - getCursorName()

- getArray(int)
- getObject(int, Map)
- getRef(int)
- getURL(int)
- getUnicodeStream(int)
- updateArray(int, Array)
- updateRef(int, Ref)

DatabaseMetaData의 일부 인터페이스 결과에 항목 추가

getProcedures(), getProcedureColumns(), getFunctions(), getFunctionColumns() 인터페이스 결과에 SPECIFIC_NAME 컬럼이 추가되었다.

Altibase JDBC 7.2 에서 SPECIFIC_NAME은 다음과 같은 형태로 구현하였다.

```
ProcName(FuncName) + '_' + ouid
```

연결 속성 기본값 변경

- reuse_resultset
  - ResultSet 객체 재사용 여부를 설정한다.
  - Altibase 7.2 기본값은 true로 ResultSet 객체를 재사용하지만 Altibase 7.1 기본값은 false로 재사용하지 않는다.
- lob_null_select
  - LOB 컬럼 값이 NULL일 때 getBlob(), getClob()이 LOB 객체를 리턴하는지 여부
  - Altibase 7.2 기본값은 off로 LOB 객체를 반환하지 않는다. Altibase 7.1 기본값은 on으로 LOB 객체를 반환한다.

Altibase JDBC 4.2만을 위한 JDBC 연결 속성 추가

- getprocedures_return_functions
  - DatabaseMetaData.getProcedures(), getProcedureColumns()의 결과에 function 결과를 포함할지 설정한다. JDBC API Specification 4.2 표준은 function 정보를 제외하지만 Altibase JDBC 4.2는 클라이언트 하위 호환성을 위해 하위 버전과 같게 유지한다.표준에 따라 function정보를 제외하려면 속성값을 false로 설정한다.

CLIENT_TYPE 변경

Altibase 7.2 JDBC 세션의 CLIENT_TYPE은 NEW_JDBC42이다. Altibase 7.2 JDBC Driver 를 이용하여 컴파일 또는 실행한 경우 V$SESSION의 CLIENT_TYPE 값은 NEW_JDBC42 로 조회해야 한다.

Altibase 이중화 호환성

Altibase 7.1 과 Altibase 7.2 양방향 이중화 제약 사항

Altibase 7.1과 Altibase 7.2는 DDL 복제와 오프라인 이중화가 불가하다.

DDL 복제는 이중화 프로토콜 버전(replication protocol version) 세 자리가 모두 일치해야 하는 기능으로, 하위 호환성을 보장하지 않는다.

오프라인 이중화는 바이너리 데이터베이스 버전(binary db version) 세 자리가 모두 일치해야 하는 이중화 부가 기능으로 하위 호환성을 보장하지 않는다.

Altibase 6.5.1 과 Altibase 7.2 양방향 이중화 제약 사항

Altibase 이중화 하위 호환성 보장에 따라 Altibase 6.5.1와 Altibase 7.2 간 단방향 및 양방향 LAZY 모드 이중화는 가능하다. 단, 이중화 대상 테이블에 공간 데이터 타입 컬럼이 있는 경우 Altibase 7.2 에서 Altibase 6.5.1 로 이중화 하는 경우 SRID 값을 가진 데이터를 Altibase 6.5.1 로 동기화할 수 없다.

SQL 결과 및 실행 계획 변화

- 서브쿼리의 인라인 뷰에 ORDER BY절 사용 시 SQL 성능 개선

  이 영향을 받는 SQL의 실행 계획에 변화가 있다. SUBQUERY FILTER 안에 SORT 플랜 노드 없어진다.
- 중첩된 LEFT OUTER JOIN 수행 방식을 최적화

  이 영향을 받는 SQL에서 실행 계획 변경 및 SQL 수행 결과가 달라질 수 있다.
- Subquery Unnesting 관련 기능 변경 및 추가

  이 영향을 받는 SQL에서 실행 계획이 변경될 수 있다.

Altibase 서버 기본 메모리 사용 증가

트랜잭션 로그파일 압축 시 메모리 할당/해제 병목 개선의 영향으로 Altibase 서버의 기본 메모리 사용이 증가한다. V$MEMSTAT의 Storage_Memory_Recovery 항목으로 이전 버전과 증가량을 확인할 수 있다. 메모리 증가량은 TRANSACTION_TABLE_SIZE에 영향을 받는다. TRANSACTION_TABLE_SIZE 기본값 1024 경우 약 32MB 증가, 최대값 16384 경우 약 500M 증가한다.

에러 메시지 변경

큐(QUEUE) 생성 시 디스크 테이블스페이스를 지정한 경우 발생하는 에러 메시지가 ERR-311E5 : The table is not a memory or volatile table. 에서 아래와 같이 변경되었다.

```
iSQL> CREATE QUEUE Q1 ( 7 ) TABLESPACE SYS_TBS_DISK_DATA;
[ERR-314AA : Failed to create queue table in disk tablespace.]
```

데이터베이스 버전

데이터베이스 구성 요소 별 버전

Altibase 버전	데이터베이스 바이너리 버전	메타 버전	통신 프로토콜 버전	이중화 프로토콜 버전
7.1.0.5.8	6.5.1	8.9.1	7.1.7	7.4.6
7.2.0.0.1	7.2.0	8.9.1	7.1.7	7.4.8

호환성

데이터베이스 바이너리 버전

데이터베이스 바이너리 버전은 데이터베이스 이미지 파일과 로그 파일의 호환성을 나타낸다.

로그 파일 로깅 구조 개선으로 데이터베이스 바이너리 버전이 변경되었다. Altibase 7.2 이전 버전 데이터베이스와 호환되지 않으므로 Altibase 버전 업그레이드 시 마이그레이션 작업이 필요하다.

통신 프로토콜 버전

Altibase 서버와 클라이언트 간 통신 규약 호환성을 의미하며 클라이언트 하위 호환성을 알 수 있다.

통신 프로토콜 버전 중 상위 두 자리는 같고 패치 버전이 변경되었다. 메이저 버전과 마이너 버전이 같으면 클라이언트 하위 호환성을 보장한다.

> 클라이언트 하위 호환성은 하위 버전 Altibase 라이브러리로 컴파일한 사용자 응용 프로그램(Altibase 클라이언트)이 상위 버전 Altibase 에서 정상 동작하는 것을 보장한다.

이중화 프로토콜 버전

이중화 프로토콜 버전은 Altibase 이중화 하위 호환성이나 이중화 부가기능 호환 여부를 나타낸다.

메이저 버전과 마이너 버전 변경이 없어 LAZY 모드 이중화는 Altibase 이중화 하위 호환성을 보장하지만 패치 버전 변경으로 이중화 부가기능은 호환되지 않는다.

> **Altibase 이중화 하위 호환성**
>
> Altibase 이중화 하위 호환성이란 이중화 프로토콜 버전이 낮은 버전에서 높은 버전으로 단방향 이중화가 가능함을 의미하며 이중화 프로토콜 버전에서 상위 두 자리(메이저와 마이너 버전)가 같은 경우 보장한다.
> Altibase 이중화 하위 호환성은 LAZY 모드 이중화로 제한한다.
>
> EAGER 모드 이중화는 하위 호환성을 보장하지 않는다.
> DDL 복제는 이중화 프로토콜 버전 세 자리가 모두 일치해야하므로 하위 호환성을 보장하지 않는다.
> 오프라인 이중화를 포함한 이중화 부가기능은 하위 호환성을 보장하지 않는다.

Altibase 서버 프로퍼티

Altibase 7.2.0.0.1 에서 추가, 변경, 삭제된 Altibase 서버 프로퍼티들이다. 각 프로퍼티에 대한 자세한 내용은 General Reference-1.Data Types & Altibase Properties를 참고하기 바란다.

새로운 프로퍼티

- DBLINK_GLOBAL_TRANSACTION_LEVEL

변경된 프로퍼티

- EXECUTE_STMT_MEMORY_MAXIMUM

  기본값이 1073741824에서 2147483648로 변경되었다.

- INDEX_MAXTRANS

  기본값과 MAX값이 30에서 50으로 변경되었다.

- INDEX_INITRANS

  MAX값이 30에서 50으로 변경되었다.

- LOCK_MGR_TYPE

기본값이 0에서 2로 변경되었다.

- PSM_CHAR_DEFAULT_PRECISION

  기본값이 32767에서 32000으로 변경되었다.

- PSM_NCHAR_UTF16_DEFAULT_PRECISION

  기본값이 16383에서 16000으로 변경되었다.

- PSM_NCHAR_UTF8_DEFAULT_PRECISION

  기본값이 10921에서 10666으로 변경되었다.

- PSM_NVARCHAR_UTF16_DEFAULT_PRECISION

  기본값이 16383에서 16000으로 변경되었다.

- PSM_NVARCHAR_UTF8_DEFAULT_PRECISION

  기본값이 10921에서 10666으로 변경되었다.

- PSM_VARCHAR_DEFAULT_PRECISION

  기본값이 32767에서 32000으로 변경되었다.

삭제된 프로퍼티

- GLOBAL_TRANSACTION_LEVEL

메타 테이블

새로운 메타테이블

- SYS_REPL_TABLE_OID_IN_USE_

성능 뷰

아래의 성능 뷰 들이 추가되었다. 각 성능 뷰에 대한 자세한 내용은 General Reference-2.The Data Dictionary를 참고하기 바란다.

새로운 성능 뷰

- V$REPL_REMOTE_META_INDEX_COLUMNS
- V$QUEUE_DELETE_OFF

패키지

OS	CPU	서버/클라이언트	패키지 인스톨러 이름
LINUX	x86-64	Altibase 서버	altibase-server-7.2.0.0.1-LINUX-X86-64bit-release.run
		Altibase 클라이언트	altibase-client-7.2.0.0.1-LINUX-X86-64bit-release.run

다운로드

Package

http://support.altibase.com

Manual

Altibase 7.2 Manuals

설치

Altibase 7.2 Installation Guide