

# Altibase 7.3.0.0.2 Patch Notes

- [New Features](#)

- [BUG-50557 Windows 환경에서의 BIGINT 배열 저장프로시저에 대한 long long 타입 지원](#)
- [BUG-50604 PSM 내 구문에서 대소문자를 구분하도록 지원합니다.](#)
- [BUG-50605 Windows 용 클라이언트를 제공합니다.](#)
- [BUG-50606 Maven Central Repository에 Altibase 7.3.0.0.2 JDBC 드라이버를 업로드 합니다.](#)
- [BUG-50608 Private synonym을 생성하면서 synonym과 객체의 이름을 비교할 때 대소문자를 구분하도록 수정합니다.](#)

- [Fixed Bugs](#)

- [BUG-49699 Offline Adapter 수행 로그에 SPLIT, MERGE, DROP PARTITION의 DDL이 포함되어 있는 경우, BUILD OFFLINE META 명령이 실패합니다.](#)
- [BUG-50365 하이브리드 파티션드 테이블에서 컬럼 제약을 체크하는 로직에 서버가 비정상 종료할 수 있는 버그가 있어서 수정합니다.](#)
- [BUG-50413 DDL과 alter replication RECEIVE ONLY, DDL REPLICATE 구문 수행시 lock timeout이 발생할수 있습니다.](#)
- [BUG-50435 다중 서브 쿼리의 최적화 과정 중 뷰 머지\(view merge\) 단계에서 예상치 못한 오류가 발생합니다.](#)
- [BUG-50442 파티션드 테이블의 simple fast execution시 비정상 종료](#)
- [BUG-50554 AUDIT OUTPUT METHOD가 1일때 audit 기록에서 누락된 에러 코드가 있습니다.](#)
- [BUG-50576 이중화 파티션이 1개인 경우 해당 파티션을 삭제할 수 없도록 수정합니다.](#)
- [BUG-50581 TIMESTAMPADD 함수를 소문자로 사용하는 경우 \[ERR-31129: Procedure or function not found\] 오류가 발생합니다.](#)
- [BUG-50589 트레이스 로그가 altibase boot.log-2까지 기록된 후 서버 재시작 시, altibase boot.log-0부터 다시 기록되는 문제가 있습니다.](#)
- [BUG-50595 동일한 NO PLAN CACHE 힌트를 사용하는 쿼리를 여러번 수행할 때 메모리가 증가하는 문제가 있습니다.](#)
- [BUG-50597 jdbc에서 date 컬럼을 fetch할때 타임존에 따라 의도한 값이 리턴되지 않을 수 있습니다.](#)
- [BUG-50601 집계 함수를 포함하는 SubQuery Unnest 동작에 결과 오류가 발생합니다.](#)
- [BUG-50607 PSM에서 LABEL로 GOTO할 때 SQL cancel이 동작하지 않을 수 있습니다.](#)
- [BUG-50615 WHERE 절에 여러 개의 서브쿼리와 OR 조건을 사용한 경우, SubQuery Unnest 동작에서 결과 오류가 발생할 수 있습니다.](#)
- [BUG-50617 이중화 SYNC 수행 중 연속적으로 충돌이 발생시, 충돌 방지 정책에 따라 Replace가 연속으로 수행되면 SYNC Fail이 발생할 수 있습니다.](#)
- [BUG-50642 Linux kernel 5에서 altiMon 실행 시, com.altibase.picl.LibLoader.OsNotSupportedException 에러가 발생합니다.](#)

- [Changes](#)

- [Version Info](#)

- [호환성](#)
- [프로퍼티](#)
- [성능 분](#)

# New Features

---

## BUG-50557 Windows 환경에서의 BIGINT 배열 저장프로시저에 대한 long long 타입 지원

- **module** : mm-apre
- **Category** : Portability
- **재현 빈도** : Always
- **설명** : Windows와 32비트 linux 환경에서 BIGINT 배열을 사용하는 저장 프로시저의 경우, 응용프로그램에서 배열 호스트 변수의 데이터 타입으로 long long 을 사용해야 합니다. 그렇지 않고 long을 사용하는 경우, [ERR-31471 Index out of range for host language array.] 오류가 발생합니다.
- **재현 방법**
  - 재현 절차
  - 수행 결과
  - 예상 결과
- **Workaround**
- **변경사항**
  - Performance view
  - Property
  - Compile Option
  - Error Code

## BUG-50604 PSM 내 구문에서 대소문자를 구분하도록 지원합니다.

- **module** : qp-psm-trigger-pvo
- **Category** : Enhancement
- **재현 빈도** : Always
- **설명** : PSM 내 구문에서 %ROWTYPE 또는 RECORD 타입의 컬럼을 참조할 때, 컬럼 이름의 대소문자를 구분하도록 PSM\_CASE\_SENSITIVE\_MODE 프로퍼티가 추가되었습니다. 이 프로퍼티의 값을 1로 설정하면 대소문자를 구분하도록 동작합니다. 이 설정은 LABEL문을 사용할때에도 적용됩니다. 또한, PSM에서 ""(double qoutation)을 사용해서 %ROWTYPE 또는 RECORD 타입 컬럼의 이름을 참조할 때, 쿼리 변환이 잘못되는 버그도 수정되었습니다.
- **재현 방법**
  - 재현 절차

```
drop table "tT";
CREATE TABLE "tT" (
  "t" int,
  "T" int
);
CREATE OR REPLACE PROCEDURE proc1
AS
  r1 "tT"%ROWTYPE;
BEGIN
  r1."t" := 10;
END;
/
```

◦ 수행 결과

```
[ERR-31182 : Duplicate column name
In procedure SYS.PROC1
0005 :      R1."t" := 10;
          ^  ^
]
```

◦ 예상 결과

```
Create success.
```

• Workaround

• 변경사항

◦ Performance view

◦ Property

■ 추가

■ PSM\_CASE\_SENSITIVE\_MODE - 기본값 : 0

- 값의 범위 : [0,1]

- 설명: PSM 내부의 %ROWTYPE 및 RECORD 타입의 column 이름, LABEL 이름 등을 참조할 때 대소문자를 구분할지 여부를 설정하는 프로퍼티. 0으로 설정하면 대소문자를 구분하지 않고, 1로 설정하면 대소문자를 구분합니다.

◦ Compile Option

• Error Code

## BUG-50605 Windows 용 클라이언트를 제공합니다.

• **module** : ul-odbc

• **Category** : Portability

• **재현 빈도** : Always

• **설명** : Windows 용 클라이언트를 제공합니다.

• **재현 방법**

◦ 재현 절차

◦ 수행 결과

◦ 예상 결과

- **Workaround**
- **변경사항**
  - Performance view
  - Property
  - Compile Option
  - Error Code

## BUG-50606 Maven Central Repository에 Altibase 7.3.0.0.2 JDBC 드라이버를 업로드 합니다.

- **module** : mm-jdbc
- **Category** : Enhancement
- **재현 빈도** : Always
- **설명** : JDBC 4.2 API를 부분 지원하는 Altibase42.jar 드라이버를 Maven Central Repository에 업로드하는 기능이 추가되었습니다. 이제 Maven Central Repository에서 'altibase-jdbc'를 검색하여 7.3.0.0.2 버전의 JDBC 드라이버를 이용할 수 있습니다.
- **재현 방법**
  - 재현 절차
  - 수행 결과
  - 예상 결과
- **Workaround**
- **변경사항**
  - Performance view
  - Property
  - Compile Option
  - Error Code

## BUG-50608 Private synonym을 생성하면서 synonym과 객체의 이름을 비교할 때 대소문자를 구분하도록 수정합니다.

- **module** : qp-ddl-dcl-pvo
- **Category** : Enhancement
- **재현 빈도** : Always
- **설명** : Private synonym을 생성할 때 대소문자를 구분할 수 있도록 수정되었습니다. 이전에는 이름의 대소문자를 구별하지 않아 이름이 동일한 synonym을 생성할 수 없었으나, 이 패치의 적용으로 가능해졌습니다.
- **재현 방법**
  - 재현 절차

```
create or replace procedure "proc4" as
begin
    println('proc4');
end;
/
create synonym proc4 for sys."proc4";
```

- 수행 결과

```
[ERR-31208 : Unable to create a synonym with the same name as the
object]
```

- 예상 결과

```
Create success.
```

- Workaround

- 변경사항

- Performance view
- Property
- Compile Option
- Error Code

# Fixed Bugs

## BUG-49699 Offline Adapter 수행 로그에 SPLIT, MERGE, DROP PARTITION의 DDL이 포함되어 있는 경우, BUILD OFFLINE META 명령이 실패합니다.

- **module** : rp-jdbcAdapter
- **Category** : Functional Error
- **재현 빈도** : Always
- **설명** : Offline Adapter에서 BUILD OFFLINE META 명령을 수행하면 메타 파일의 이중화 정보와 offline adapter의 이중화 정보를 비교하는데, 이중화 테이블 이름과 사용자 이름의 갯수가 다르면 명령이 실패하는 문제가 있었습니다.  
갯수가 다르더라도 메타파일의 이중화 테이블 이름과 사용자 이름이 offline adapter의 이중화 정보에 포함되면 명령이 실패하지 않도록 수정합니다.
- **재현 방법**
  - 재현 절차
  - 수행 결과
  - 예상 결과
- **Workaround**
- **변경사항**
  - Performance view
  - Property
  - Compile Option
  - Error Code
    - 오프라인 이중화에서 이중화 테이블 이름과 사용자 이름이 메타파일의 정보와 다를때 발생하는 에러메시지가 추가되었습니다.

```
ERR-611D3 The table name and user name for offline replication do
not match the meta file.(table name: <0%>, user name: <1%>)
```

```
*Cause: The offline replication meta information configuration fails
due to a mismatch between the table name and user name compared to
the information in the meta file.
```

```
*Action: Check the table name and user name for offline replication
and configure the replication that matches the meta file.
```

## BUG-50365 하이브리드 파티션드 테이블에서 컬럼 제약을 체크하는 로직에 서버가 비정상 종료할 수 있는 버그가 있어서 수정합니다.

- **module** : qp
- **Category** : Fatal
- **재현 빈도** : Always

- **설명** : 하이브리드 파티션드 테이블에서 컬럼 제약을 체크하는 로직에 서버가 비정상 종료할 수 있는 버그가 있어서 수정합니다.
- **재현 방법**
  - 재현 절차
  - 수행 결과
  - 예상 결과
- **Workaround**
- **변경사항**
  - Performance view
  - Property
  - Compile Option
  - Error Code

## BUG-50413 이중화 대상 테이블에 DDL 수행 및 DDL 복제, alter replication ... RECEIVE\_ONLY 구문 수행시, lock timeout이 발생합니다.

- **module** : rp
- **Category** : Fatal
- **재현 빈도** : Always
- **설명** : 이중화 대상 테이블에 DDL 수행 및 DDL 복제, alter replication ... RECEIVE\_ONLY 구문 수행시, lock timeout이 발생하는 문제를 수정합니다.
- **재현 방법**
  - 재현 절차
  - 수행 결과
  - 예상 결과
- **Workaround**
- **변경사항**
  - Performance view
  - Property
  - Compile Option
  - Error Code

## BUG-50435 다중 서브 쿼리의 최적화 과정 중 뷰 머지(view merge) 단계에서 예상치 못한 오류가 발생합니다.

- **module** : qp
- **Category** : Functional Error
- **재현 빈도** : Always



- **설명** : 뷰를 포함하는 다중 서브 쿼리의 최적화 과정 중 뷰 머지(view merge) 단계에서 [ERR-31455 : Failed to work because an internal exception occurred from an OS.[Contact Altibase's Support Center]] 오류가 발생하는 문제를 수정하였습니다.
- **재현 방법**
  - 재현 절차

```
DROP TABLE T1 CASCADE;
CREATE TABLE T1( I1 INTEGER, I2 INTEGER );
ALTER TABLE T1 ADD CONSTRAINT T1_PK PRIMARY KEY( I1 );
SELECT I1 FROM T1 WHERE I1 IN ( SELECT I1 FROM ( SELECT DISTINCT I1 FROM
T1 ) WHERE I1 = 1 OR I2 = 1 );
```

- 수행 결과

```
[ERR-31455 : Failed to work because an internal exception occurred from
an OS.[Contact Altibase's Support Center]]
```

- 예상 결과

```
No rows selected.
```

- **Workaround**

```
SELECT I1 FROM T1 WHERE I1 IN ( SELECT /*+ no_unnest */ I1 FROM ( SELECT
DISTINCT I1 FROM T1 ) WHERE I1 = 1 OR I2 = 1 );
또는
SELECT I1 FROM T1 WHERE I1 IN ( SELECT I1 FROM ( SELECT /*+ no_merge */
DISTINCT I1 FROM T1 ) WHERE I1 = 1 OR I2 = 1 );
```

- **변경사항**
  - Performance view
  - Property
  - Compile Option
  - Error Code

## BUG-50442 파티션드 테이블의 simple fast execution시 비정상 종료

- **module** : qp
- **Category** : Fatal
- **재현 빈도** : Always
- **설명** : 파티션드 테이블에 단순한 DML 구문을 수행할 때, SIMPL\_QUERY 최적화를 사용한 경우 서버가 비정상 종료되는 버그를 수정합니다.
- **재현 방법**
  - 재현 절차



- 변경사항
  - Performance view
  - Property
  - Compile Option
  - Error Code

## BUG-50576 이중화 파티션이 1개인 경우 해당 파티션을 삭제할 수 없도록 수정합니다.

- **module** : rp
- **Category** : Functional Error
- **재현 빈도** : Always
- **설명** : 이중화 파티션이 1개인 경우, 해당 파티션을 삭제한 후 이중화를 시작하면 이중화 메타 정보가 유효하지 않아 이중화 시작이 실패하는 문제가 있었습니다. 이에 이중화 파티션이 1개인 경우 해당 파티션을 삭제할 수 없도록 수정하였습니다.

이중화 파티션이 1개인 경우에 파티션 삭제 명령을 시도하면 아래와 같은 에러메시지가 출력됩니다:

ERR-611D4 : Unable to drop the partition because it is the only one replication partition.

해당 파티션을 삭제하려면, 이중화 객체를 삭제하거나 다른 이중화 파티션을 추가한 다음 삭제할 수 있습니다.

- **재현 방법**
  - 재현 절차
  - 수행 결과
  - 예상 결과
- **Workaround**
- **변경사항**
  - Performance view
  - Property
  - Compile Option
  - Error Code
    - 이중화 파티션이 1개인 경우에 파티션 삭제 명령을 시도할 때 발생하는 에러메시지가 추가되었습니다.

ERR-611D4 : Unable to drop the partition because it is the only one replication partition.

\*Cause: The partition cannot be dropped when there is only one replication partition remaining.

\*Action: - Check the count of replication partition.  
 - To drop the last replication partition, drop the replication first.  
 - To drop the partition without dropping the replication object, add another replication partition first and then try again.

## BUG-50581 TIMESTAMPADD 함수를 소문자로 사용하는 경우 [ERR-31129 : Procedure or function not found] 오류가 발생합니다.

- **module** : qp
- **Category** : Functional Error
- **재현 빈도** : Always
- **설명** : TIMESTAMPADD 함수를 소문자로 사용하는 경우, [ERR-31129 : Procedure or function not found] 오류가 발생하는 문제를 수정합니다.
- **재현 방법**
  - 재현 절차

```
ALTER SESSION SET DEFAULT_DATE_FORMAT = 'YYYY-MM-DD HH:MI:SS:SSSSSS';
SELECT /*+ NO_PLAN_CACHE */ timestampadd( SQL_TSI_MICROSECOND, 1, '2020-01-10' ) FROM dual;
```

- 수행 결과

```
[ERR-31129 : Procedure or function not found :
0001 : SELECT /*+ NO_PLAN_CACHE */ timestampadd( SQL_TSI_MICROSECOND, 1,
'2020-01-10' ) FROM DUAL
                                ^           ^
.]
```

- 예상 결과

```
TIMESTAMPADD(SQL_TSI_MICROSECOND,1,'2020-0
-----
2020-01-10 00:00:00:000001
1 row selected.
```

- **Workaround**

```
SELECT /*+ NO_PLAN_CACHE */ TIMESTAMPADD( SQL_TSI_MICROSECOND, 1, '2020-01-10' ) FROM dual ;
```

- **변경사항**

- Performance view
- Property
- Compile Option
- Error Code

**BUG-50589 트레이스 로그가 altibase\_boot.log-2까지 기록된 후 서버 재시작 시, altibase\_boot.log-0부터 다시 기록되는 문제가 있습니다.**

- **module** : id
- **Category** : Functional Error
- **재현 빈도** : Rare
- **설명** : \$ALTIBASE\_HOME/trc의 트레이스 로그가 서버 재 시작시, 이전에 기록했던 순서를 무시하고 다시 0번부터 기록되는 문제가 있습니다. 서버가 재시작해도 트레이스 로그의 순서대로 기록하도록 수정합니다.
- **재현 방법**
  - 재현 절차
  - 수행 결과
  - 예상 결과
- **Workaround**
- **변경사항**
  - Performance view
  - Property
  - Compile Option
  - Error Code

**BUG-50595 동일한 NO\_PLAN\_CACHE 힌트를 사용하는 쿼리를 여러 번 수행할 때 메모리가 증가하는 문제가 있습니다.**

- **module** : mm-plancache
- **Category** : Memory Error
- **재현 빈도** : Always
- **설명** : 동일한 NO\_PLAN\_CACHE 힌트를 사용하는 쿼리를 여러번 수행할 때마다 내부적으로 childPCO의 수가 증가하는 문제가 있고, 이로 인해 메모리 사용량이 증가하는 문제가 있어 수정하였습니다.
- **재현 방법**
  - 재현 절차
  - 수행 결과
  - 예상 결과
- **Workaround**
- **변경사항**
  - Performance view

- Property
- Compile Option
- Error Code

## BUG-50597 jdbc에서 date 컬럼을 fetch할때 타임존에 따라 의도한 값이 리턴되지 않을 수 있습니다.

- **module** : mm-jdbc
- **Category** : Functional Error
- **재현 빈도** : Always
- **설명** : 타임존에 따라 올바르지 않은 date 컬럼 값이 리턴되는 문제를 수정하였습니다.
- **재현 방법**
  - **재현 절차**

```
create table theentity
(
    theid    integer not null,
    thevalue date,
    primary key (theid)
);

Connection sConn = getConnection("21165");
PreparedStatement sPstmt = sConn.prepareStatement("INSERT INTO theentity
(theid, thevalue) VALUES ( ? , ? )");
Timestamp sTimestamp = Timestamp.from(OffsetDateTime.of(2018, 3, 25, 2,
0, 0, 0, ZoneOffset.UTC ).toInstant());
sPstmt.setInt(1, 1);
sPstmt.setTimestamp(2, sTimestamp,
Calendar.getInstance(TimeZone.getTimeZone("UTC")));
sPstmt.execute();
sPstmt.close();

sPstmt = sConn.prepareStatement("select ewil_0.theid, ewil_0.thevalue
\n" +
                                "from theentity ewil_0 \n" +
                                "where ewil_0.theid = ?");

sPstmt.setInt(1, 1);
ResultSet sRs = sPstmt.executeQuery();
if (sRs.next())
{
    Calendar aCal = Calendar.getInstance( TimeZone.getTimeZone( "UTC" )
);
    Timestamp sCol = sRs.getTimestamp(2, aCal);
    Instant sInstant = sCol.toInstant();
    System.out.println("Instant====>" + sInstant);
}

// java 실행시 timezone을 Europe/Paris 로 설정
java -Duser.timezone="Europe/Paris" InstantTest
```

- **수행 결과**

```
Instant====>2018-03-25T03:00:00Z
```

- 예상 결과

```
Instant====>2018-03-25T02:00:00Z
```

- Workaround

```
timezone을 생략하거나 Asia/Seoul로 설정
```

- 변경사항

- Performance view
- Property
- Compile Option
- Error Code

## BUG-50601 집계 함수를 포함하는 SubQuery Unnest 동작에 결과 오류가 발생합니다.

- **module** : qp
- **Category** : Functional Error
- **재현 빈도** : Always
- **설명** : 집계함수에 사용되는 컬럼이 인덱스를 가지고, 이 집계 함수를 포함하는 서브 쿼리에서 SubQuery Unnest 동작시 결과 오류가 발생하는 문제를 수정합니다.
- **재현 방법**
  - 재현 절차

```
drop table t1;
drop table t2;
CREATE TABLE T1
(
    I1          integer not null,
    I2          smallint,
    I3 integer not null,
    I4 integer not null,
    primary key (I1, I3, I4)
);

CREATE TABLE T2
(
    I1 integer not null,
    I2 smallint,
    I3 integer not null,
    I4 varchar(8),
    primary key (I1, I3)
);

INSERT INTO T1(I2, I4, I1, I3) values (0, 2, 21, 1);
INSERT INTO T1(I2, I4, I1, I3) values (2, 2, 23, 1);
```

```

INSERT INTO T2(I2, I4, I1, I3) values (0, 'aaaaa', 21, 2);
INSERT INTO T2(I2, I4, I1, I3) values (1, 'bbbbbb', 22, 2);
INSERT INTO T2(I2, I4, I1, I3) values (2, 'bbbbbb', 23, 2);

SELECT
    T1.I1 as T1_I1,
    T2.I4 as T2_I4
FROM T1,
     T2
WHERE T1.I4 = T2.I3
    and T1.I3 = 1
    and (
        (
            T2.I1 = (SELECT /*+no_unnest */ max(b.I1)
                    from T2 b
                    where b.I1 <= 23
                      and T2.I3 = b.I3)
            and T1.I1 = (SELECT max(a.I1)
                        from T1 a
                        where a.I1 < 23
                          and T1.I3 = a.I3
                          and T1.I4 = a.I4)

            and T1.I2 != 2
            and T2.I2 != 2
        )
        or (
            T1.I1 = 23
            and T2.I1 = 23
            and T1.I2 = 2
            and T2.I2 = 2
        )
    );

```

○ 수행 결과

```

T1_I1      T2_I4
-----
21          aaaaa
21          bbbbb
23          bbbbb
3 rows selected.

```

○ 예상 결과

```

T1_I1      T2_I4
-----
23          bbbbb
1 row selected.

```

• Workaround

```

/*+ no_unnest */ 힌트 사용
select
    T1.I1 as T1_I1,
    T2.I4 as T2_I4

```



```

from T1,
     T2
where T1.I4 = T2.I3
     and T1.I3 = 1
     and (
         (
             T2.I1 = (select /*+no_unnest */ max(b.I1)
                        from T2 b
                        where b.I1 <= 23
                        and T2.I3 = b.I3)
             and T1.I1 = (select /*+no_unnest */ max(a.I1)
                            from T1 a
                            where a.I1 < 23
                            and T1.I3 = a.I3
                            and T1.I4 = a.I4)

             and T1.I2 != 2
             and T2.I2 != 2
         )
         or (
             T1.I1 = 23
             and T2.I1 = 23
             and T1.I2 = 2
             and T2.I2 = 2
         )
     );

```

- 변경사항
  - Performance view
  - Property
  - Compile Option
  - Error Code

## BUG-50607 PSM에서 LABEL로 GOTO할 때 SQL cancel이 동작하지 않을 수 있습니다.

- **module** : qp-psm-trigger-execute
- **Category** : Functional Error
- **재현 빈도** : Always
- **설명** : PSM에서 상위 BLOCK으로 GOTO를 사용하여 LOOP 처럼 동작할 때, SQL cancel 기능이 동작하지 않는 문제를 수정했습니다.
- **재현 방법**
  - 재현 절차

```

BEGIN
<>
BEGIN
    GOTO L1;
END;
END;
/

```

- 수행 결과
- 예상 결과
- **Workaround**
- **변경사항**
  - Performance view
  - Property
  - Compile Option
  - Error Code

## BUG-50615 WHERE 절에 여러 개의 서브쿼리와 OR 조건을 사용한 경우, SubQuery Unnest 동작에서 결과 오류가 발생할 수 있습니다.

- **module** : qp
- **Category** : Functional Error
- **재현 빈도** : Always
- **설명** : WHERE 절에 여러 개의 서브쿼리와 OR 조건을 사용한 경우, SubQuery Unnest 동작에서 결과 오류가 발생하는 문제를 수정합니다.
- **재현 방법**
  - 재현 절차

```
create table T1
(
    REV                integer not null,
    REVTYPE            smallint,
    T_id integer not null,
    map_KEY            integer not null,
    map_id             integer not null,
    primary key (REV, T_id, map_KEY, map_id)
);
create table T2
(
    REV    integer not null,
    REVTYPE smallint,
    id     integer not null,
    str    varchar(255),
    primary key (REV, id)
);
create table T3
(
    T3_I1 integer,
    REV    integer not null,
    REVTYPE smallint,
    id     integer not null,
    primary key (REV, id)
);

insert into T1(REVTYPE, REV, T_id, map_id, map_KEY) values (0, 15, 1, 1, 1);
```

```

insert into T1(REVTYPE, REV, T_id, map_id, map_KEY) values (0, 15, 1, 2,
2);
insert into T1(REVTYPE, REV, T_id, map_id, map_KEY) values (2, 17, 1, 2,
2);
insert into T1(REVTYPE, REV, T_id, map_id, map_KEY) values (2, 17, 1, 1,
1);
insert into T2(REV, REVTYPE, ID, STR) values (15, 0, 1, 'value 1');
insert into T2(REV, REVTYPE, ID, STR) values (15, 0, 2, 'value 2');
insert into T2(REV, REVTYPE, ID, STR) values (16, 1, 2, 'value 3');
insert into T3(T3_I1, REV, REVTYPE, ID) values (1, 15, 0, 1);
insert into T3(T3_I1, REV, REVTYPE, ID) values (2, 15, 0, 2);
insert into T3(T3_I1, REV, REVTYPE, ID) values (3, 16, 1, 2);

select a1.REV,
       c1.T3_I1
from T1 a1,
     T2 b1,
     T3 c1
where a1.map_id = b1.id
     and a1.map_KEY = c1.id
     and a1.T_id = 1
     and (
         (
             b1.REV = (select max(b2.REV)
                       from T2 b2
                       where b2.REV <= 17
                           and b1.id = b2.id)
             and c1.REV = (select max(c2.REV)
                           from T3 c2
                           where c2.REV <= 17
                               and c1.id = c2.id)
             and a1.REV = (select max(a2.REV)
                           from T1 a2
                           where a2.REV < 17
                               and a1.T_id = a2.T_id
                               and a1.map_id = a2.map_id
                               and a1.map_KEY = a2.map_KEY)

             and a1.REVTYPE!=2
             and b1.REVTYPE!=2
             and c1.REVTYPE!=2
         )
         or (
             a1.REV = 17
             and b1.REV = 17
             and c1.REV = 17
             and a1.REVTYPE = 2
             and b1.REVTYPE = 2
             and c1.REVTYPE = 2
         )
     );

```

○ 수행 결과

REV	T3_I1
15	1
15	2
15	3
15	2
15	3

```

15      1
15      2
15      3
15      2
15      3
5 rows selected.

```

- 예상 결과

REV	T3_I1
15	1
15	3

```

15      1
15      3
2 rows selected.

```

- Workaround

```
/*+ NO_UNNEST */ 힌트 사용
```

- 변경사항

- Performance view
- Property
- Compile Option
- Error Code

**BUG-50617 이중화 SYNC 수행 중 연속적으로 충돌이 발생시, 충돌 방지 정책에 따라 Replace가 연속으로 수행되면 SYNC Fail이 발생할 수 있습니다.**

- **module** : rp-receiver
- **Category** : Functional Error
- **재현 빈도** : Always
- **설명** : 이중화 SYNC 수행 중 연속된 충돌이 발생할 때, 충돌 방지 정책에 따라 Replace가 연속으로 수행되는 경우 SYNC Fail이 발생하는 버그를 수정합니다.
- **재현 방법**
  - 재현 절차
  - 수행 결과
  - 예상 결과
- **Workaround**
- **변경사항**
  - Performance view
  - Property
  - Compile Option
  - Error Code

## BUG-50642 Linux kernel 5에서 altiMon 실행 시, com.altibase.picl.LibLoader.OsNotSupportedException 에러가 발생합니다.

- **module** : ux-altiMon
- **Category** : Usability
- **재현 빈도** : Always
- **설명** : 기존에는 Linux kernel 5를 지원하지 않았지만, 이번 패치부터 linux kernel 5 이상에서 altiMon을 실행할 수 있도록 수정되었습니다.
- **재현 방법**
  - 재현 절차
  - 수행 결과
  - 예상 결과
- **Workaround**
- **변경사항**
  - Performance view
  - Property
  - Compile Option
  - Error Code

# Changes

## Version Info

altibase version	database binary version	meta version	cm protocol version	replication protocol version
7.3.0.0.2	6.5.1	8.11.1	7.1.7	7.4.7

Altibase 7.3 패치 버전별 히스토리는 [Version Histories](#) 에서 확인할 수 있다.

## 호환성

### Database binary version

데이터베이스 바이너리 버전은 변경되지 않았다.

데이터베이스 바이너리 버전은 데이터베이스 이미지 파일과 로그파일의 호환성을 나타낸다. 이 버전이 다른 경우의 패치(업그레이드 포함)는 데이터베이스를 재구성해야 한다.

### Meta Version

메타 버전은 변경되지 않았다.

패치를 롤백하려는 경우, [메타다운그레이드](#)를 참고한다.

### CM protocol Version

통신 프로토콜 버전은 변경되지 않았다.

### Replication protocol Version

Replication 프로토콜 버전은 변경되지 않았다.

## 프로퍼티

### 추가된 프로퍼티

### 변경된 프로퍼티

### 삭제된 프로퍼티

## 성능 뷰

### 추가된 성능 뷰

### 변경된 성능 뷰

### 삭제된 성능 뷰

