

# Altibase 7.1.0.8.3 Patch Notes

---

# Table of Contents

---

- [New Features](#)
  - [BUG-49600 JDBC 4.0 API의 Connection.createBlob\(\), createClob\(\)을 지원합니다.](#)
  - [BUG-49985 GROUP BY GROUPING SETS 절에 빈 그룹\(Empty Group\)을 사용할 때 메모리 효율성을 개선합니다.](#)
- [Fixed Bugs](#)
  - [BUG-49732 압축 컬럼과 압축 컬럼이 키인 인덱스를 가진 테이블의 테이블스페이스를 변경할 때 Altibase 서버가 비정상 종료합니다.](#)
  - [BUG-49929 로컬 인덱스를 가진 하이브리드 파티션드 테이블을 CREATE TABLE ... SCHEMA ... 구문으로 복제할 때 Altibase 서버가 비정상 종료합니다.](#)
  - [BUG-49965 Multiple update 문의 대상 테이블이 부질의에 MATERIALIZE 힌트를 사용한 뷰일 때 결과 오류가 발생할 수 있습니다.](#)
  - [BUG-50013 Adapter 로그\(jdbcAdapter.trc 또는 oraAdapter.trc\)에 DML 정보를 출력하는 형식을 설정하는 프로퍼티를 추가합니다.](#)
  - [BUG-50025 CREATE DATABASE 수행 후 바로 STARTUP SERVICE를 수행하면 Altibase 서버가 비정상 종료하거나 ERR-11110 : The index is inconsistent 에러가 발생합니다.](#)
  - [BUG-50033 adapter를 시작할 때 ALA 로그 파일을 오픈할 수 없으면 코어 파일이 생성됩니다.](#)
  - [BUG-50057 ROLLUP 함수와 집계 함수 그리고 ORDER BY 절을 함께 사용하면 결과 오류가 발생합니다.](#)
  - [BUG-50058 DROP REPLICATION 수행 시 이중화 수신자 스레드가 종료되지 않을 수 있는 메모리 참조 오류를 수정합니다.](#)
  - [BUG-50059 aku -p end 수행 동작을 변경합니다. 이중화 FLUSH 명령을 수행한 후에 이중화를 중지합니다.](#)
  - [BUG-50061 updatable view에 집합 연산자를 사용했을 때 예외 처리를 변경합니다.](#)
  - [BUG-50071 SET 절의 서브쿼리에서 multiple update의 대상을 참조하면 ERR-31455 에러가 발생하거나 Altibase 서버가 비정상 종료할 수 있습니다.](#)
  - [BUG-50082 비정상적으로 종료된 슬레이브 파드에서 aku -p start 수행 동작을 변경합니다. 이중화 시작 후 FLUSH 명령을 수행합니다.](#)
  - [BUG-50090 Statement 객체의 다음 결과가 ResultSet 객체가 아닐 때 getMoreResults\(\)에서 true를 반환합니다.](#)
  - [BUG-50094 메모리 테이블의 ADD COLUMN 수행 동작을 변경합니다.](#)
  - [BUG-50096 DDL 복제 실패 이후 HeartBeat 스레드의 비정상적인 동작으로 이중화 송신자가 시작되지 않을 수 있습니다.](#)
  - [BUG-50098 delete 또는 Multiple delete 문의 대상 테이블이 부질의에 MATERIALIZE 힌트를 사용한 뷰일 때 결과 오류가 발생할 수 있습니다.](#)
  - [BUG-50106 iSQL에서 desc 명령을 수행하면 ERR-31002 : A single-row subquery has returned more than one row. 에러가 발생합니다.](#)
  - [BUG-50131 조건절 검사 시 Altibase 서버가 비정상 종료하는 현상에 대한 예외처리를 추가합니다.](#)
- [Changes](#)
  - [Version Info](#)

- [호환성](#)
- [프로퍼티](#)
- [성능 분](#)

# New Features

## BUG-49600 JDBC 4.0 API의 Connection.createBlob(), createClob()을 지원합니다.

### module

mm-jdbc

### Category

Enhancement

### 재현 빈도

Always

### 설명

JDBC 4.0 API의 Connection.createBlob(), createClob()을 지원합니다.

관련 설명은 [Altibase 7.1 : JDBC User's Manual - createBlob\(\), createClob\(\)을 이용한 LOB 사용](#)에서 확인할 수 있습니다.

이 버그에서 추가된 기능을 사용하려면 JDBC 4.2를 부분 지원하는 JDBC 드라이버(Altibase42.jar)를 패치해야 합니다.

### 재현 방법

- 재현 절차

```
CREATE TABLE t1 (c1 int, c2 clob, c3 blob);
```

```
Connection sConn = getConnection("20300");
sConn.setAutoCommit(false);
Clob sClob = sConn.createClob();
sClob.setString(1, "dsfjsdkfsdkfjds;fsdkfs");
Blob sBlob = sConn.createBlob();
sBlob.setBytes(1, "dfkjsdlkfjsdfkjsdkfljsdfsd".getBytes());
PreparedStatement sStmt = sConn.prepareStatement("INSERT INTO t1 VALUES (?, ?, ?)");
sStmt.setInt(1, 1);
sStmt.setClob(2, sClob);
sStmt.setBlob(3, sBlob);
sStmt.executeUpdate();
sConn.commit();
```

```
SELECT c1,length(c2), binary_length(c3) FROM t1;
```

- 수행 결과

```
Exception in thread "main" java.sql.SQLException:
    at
    Altibase.jdbc.driver.ex.Error.createSQLException(Error.java:785)
    at
    Altibase.jdbc.driver.AbstractConnection.createClob(AbstractConnection.java:34
    )
    at GetLobTest.doTest(GetLobTest.java:15)
    at GetLobTest.main(GetLobTest.java:8)
```

- 예상 결과

C1	LENGTH(C2)	BINARY_LENGTH(C3)
1	22	26

1 row selected.

## BUG-49985 GROUP BY GROUPING SETS 절에 빈 그룹(Empty Group)을 사용할 때 메모리 효율성을 개선합니다.

### module

qp-select

### Category

Enhancement

### 재현 빈도

Always

### 설명

GROUP BY GROUPING SETS 절에 빈 그룹(Empty Group)을 사용할 때 메모리 할당 시간을 감소하고 메모리 사용 효율을 개선합니다.

### 재현 방법

- 재현 절차

```
ALTER SESSION SET EXPLAIN PLAN = ON;
CREATE TABLE T1 ( I1 INTEGER, I2 INTEGER );
SELECT I1,SUM(I2)
  FROM T1
 GROUP BY GROUPING SETS ( I1, ( ) );
```

- 수행 결과

I1	SUM(I2)
No rows selected.	

```

-----
PROJECT ( COLUMN_COUNT: 2, TUPLE_SIZE: 16, COST: 388.69 )
VIEW ( ACCESS: 0, COST: 386.03 )
BAG-UNION
  PROJECT ( COLUMN_COUNT: 2, TUPLE_SIZE: 16, COST: 130.13 )
    GROUP-AGGREGATION ( ITEM_SIZE: 40, GROUP_COUNT: 0, BUCKET_COUNT: 1024,
ACCESS: 0, COST: 130.06 )
      SCAN ( TABLE: SYS.T1, FULL SCAN, ACCESS: 0, COST: 116.76 )
    PROJECT ( COLUMN_COUNT: 2, TUPLE_SIZE: 16, COST: 135.34 )
      GROUP-AGGREGATION ( ITEM_SIZE: 32, GROUP_COUNT: 0, BUCKET_COUNT: 5120,
ACCESS: 0, COST: 130.06 )
        SCAN ( TABLE: SYS.T1, FULL SCAN, ACCESS: 0, COST: 116.76 )
-----

```

- 예상 결과

```

I1          SUM(I2)
-----
No rows selected.
-----
PROJECT ( COLUMN_COUNT: 2, TUPLE_SIZE: 16, COST: 260.21 )
VIEW ( ACCESS: 1, COST: 260.19 )
BAG-UNION
  PROJECT ( COLUMN_COUNT: 2, TUPLE_SIZE: 16, COST: 130.13 )
    GROUP-AGGREGATION ( ITEM_SIZE: 40, GROUP_COUNT: 0, BUCKET_COUNT: 1024,
ACCESS: 0, COST: 130.06 )
      SCAN ( TABLE: SYS.T1, FULL SCAN, ACCESS: 0, COST: 116.76 )
    PROJECT ( COLUMN_COUNT: 2, TUPLE_SIZE: 16, COST: 130.06 )
      GROUP-AGGREGATION ( ITEM_SIZE: 32, GROUP_COUNT: 1, BUCKET_COUNT: 1,
ACCESS: 1, COST: 130.06 )
        SCAN ( TABLE: SYS.T1, FULL SCAN, ACCESS: 0, COST: 116.76 )
-----

```

# Fixed Bugs

**BUG-49732 압축 컬럼과 압축 컬럼이 키인 인덱스를 가진 테이블의 테이블스페이스를 변경할 때 Altibase 서버가 비정상 종료합니다.**

module

Sm

Category

Fatal

재현 빈도

Always

설명

압축 컬럼과 압축 컬럼이 키인 인덱스를 가진 테이블의 테이블스페이스를 변경할 때 Altibase 서버가 비정상 종료하는 버그를 수정합니다. 또한 인덱스를 가진 테이블의 테이블스페이스를 변경할 때 인덱스 생성 과정을 개선하여 ALTER TABLESPACE 수행 성능을 개선하였습니다.

이 버그는 메모리 테이블에서만 발생합니다.

재현 방법

- 재현 절차

```
CREATE TABLE T1 ( I1 CHAR(10), I2 NUMBER(2) ) COMPRESS (I1);
CREATE MEMORY TABLESPACE USER_MEM_TBS SIZE 1G AUTOEXTEND ON;
CREATE INDEX T1_I1 ON T1(I1);
INSERT INTO T1 VALUES (3,3);
ALTER TABLE T1 ALTER TABLESPACE USER_MEM_TBS;
```

- 수행 결과

```
ERR-91015 : Communication failure.
```

- 예상 결과

```
Alter success.
```

Workaround

없음

변경사항

- Performance view
- Property
- Compile Option

- Error Code

**BUG-49929 로컬 인덱스를 가진 하이브리드 파티션드 테이블을 CREATE TABLE ... SCHEMA ... 구문으로 복제할 때 Altibase 서버가 비정상 종료합니다.**

module

sm

Category

Fatal

재현 빈도

Always

설명

로컬 인덱스를 가진 하이브리드 파티션드 테이블을 CREATE TABLE ... SCHEMA ... 구문으로 복제할 때 Altibase 서버가 비정상 종료하는 버그를 수정합니다.

재현 방법

- 재현 절차

```
CREATE DISK TABLESPACE DISK_TBS_0 DATAFILE 'DISK_TBS_0' SIZE 32M AUTOEXTEND
ON;
CREATE MEMORY TABLESPACE MEM_TBS_0 SIZE 32M AUTOEXTEND ON;
CREATE TABLE T1
( I1 INTEGER,
  I2 VARCHAR(256),
  CONSTRAINT T1CONSTR UNIQUE ( I1 ASC, I2 DESC )
  USING INDEX LOCAL
  ( PARTITION T1P1IDX ON P1,
    PARTITION T1P2IDX ON P2
  )
)
PARTITION BY HASH (I1)
( PARTITION P1 TABLESPACE MEM_TBS_0,
  PARTITION P2 TABLESPACE DISK_TBS_0
)
;
CREATE TABLE T1_COPY FROM TABLE SCHEMA T1 USING PREFIX P;
```

- 수행 결과

ERR-91015 : Communication failure.

- 예상 결과

Create success.



## Workaround

없음

## 변경사항

- Performance view
- Property
- Compile Option
- Error Code

**BUG-49965 Multiple update 문의 대상 테이블이 부질의에 MATERIALIZE 힌트를 사용한 뷰일 때 결과 오류가 발생할 수 있습니다.**

## module

qp-dml-pvo

## Category

Functional Error

## 재현 빈도

Always

## 설명

Multiple update 문의 대상 테이블이 부질의에 MATERIALIZE 힌트를 사용한 뷰일 때 결과 오류가 발생하는 버그를 수정합니다.

## 재현 방법

- 재현 절차

```
DROP VIEW V1;
DROP VIEW V2;
DROP TABLE T1;
CREATE TABLE T1( I1 INTEGER, I2 INTEGER, I3 INTEGER, I4 INTEGER );
CREATE VIEW V1 AS SELECT I1, I2 FROM T1;
CREATE VIEW V2 AS SELECT /*+ MATERIALIZE */ I1, I2 FROM T1;
INSERT INTO T1 SELECT ROWNUM, ROWNUM + 1, ROWNUM, ROWNUM + 1 FROM DUAL
CONNECT BY LEVEL <= 20;
SELECT * FROM T1 WHERE I1 <> I3 OR I2 <> I4;
UPDATE V1 INNER JOIN V1 V3 ON V1.I1 = V3.I1 SET V1.I1 = V1.I1 + 1 WHERE
V1.I1 = 1 OR V3.I1 = 2;
SELECT * FROM T1 WHERE I1 <> I3 OR I2 <> I4;
UPDATE V1 INNER JOIN V2 V3 ON V1.I1 = V3.I1 SET V1.I1 = V1.I1 + 1 WHERE
V1.I1 = 4 OR V3.I1 = 5;
SELECT * FROM T1 WHERE I1 <> I3 OR I2 <> I4;
UPDATE V2 INNER JOIN V1 V3 ON V2.I1 = V3.I1 SET V2.I1 = V2.I1 + 1 WHERE
V2.I1 = 7 OR V3.I1 = 8;
SELECT * FROM T1 WHERE I1 <> I3 OR I2 <> I4;
```

```
UPDATE V2 INNER JOIN V2 V3 ON V2.I1 = V3.I1 SET V2.I1 = V2.I1 + 1 WHERE
V2.I1 = 10 OR V3.I1 = 11;
SELECT * FROM T1 WHERE I1 <> I3 OR I2 <> I4;
```

- 수행 결과

```
isQL> DROP VIEW V1;
Drop success.
isQL> DROP VIEW V2;
Drop success.
isQL> DROP TABLE T1;
Drop success.
isQL> CREATE TABLE T1( I1 INTEGER, I2 INTEGER, I3 INTEGER, I4 INTEGER );
Create success.
isQL> CREATE VIEW V1 AS SELECT I1, I2 FROM T1;
Create success.
isQL> CREATE VIEW V2 AS SELECT /*+ MATERIALIZE */ I1, I2 FROM T1;
Create success.
isQL> INSERT INTO T1 SELECT ROWNUM, ROWNUM + 1, ROWNUM, ROWNUM + 1 FROM DUAL
CONNECT BY LEVEL <= 20;
20 rows inserted.
isQL>
isQL> SELECT * FROM T1 WHERE I1 <> I3 OR I2 <> I4;
T1.I1      T1.I2      T1.I3      T1.I4
-----
No rows selected.
isQL>
isQL> UPDATE V1 INNER JOIN V1 V3 ON V1.I1 = V3.I1 SET V1.I1 = V1.I1 + 1
WHERE V1.I1 = 1 OR V3.I1 = 2;
2 rows updated.
isQL> SELECT * FROM T1 WHERE I1 <> I3 OR I2 <> I4;
T1.I1      T1.I2      T1.I3      T1.I4
-----
2          2          1          2
3          3          2          3
2 rows selected.
isQL>
isQL> UPDATE V1 INNER JOIN V2 V3 ON V1.I1 = V3.I1 SET V1.I1 = V1.I1 + 1
WHERE V1.I1 = 4 OR V3.I1 = 5;
2 rows updated.
isQL> SELECT * FROM T1 WHERE I1 <> I3 OR I2 <> I4;
T1.I1      T1.I2      T1.I3      T1.I4
-----
2          2          1          2
3          3          2          3
5          5          4          5
6          6          5          6
4 rows selected.
isQL>
isQL> UPDATE V2 INNER JOIN V1 V3 ON V2.I1 = V3.I1 SET V2.I1 = V2.I1 + 1
WHERE V2.I1 = 7 OR V3.I1 = 8;
1 row updated.
isQL> SELECT * FROM T1 WHERE I1 <> I3 OR I2 <> I4;
T1.I1      T1.I2      T1.I3      T1.I4
-----
```

```

2          2          1          2
3          3          2          3
5          5          4          5
8          6          5          6
4 rows selected.
isQL>
isQL> UPDATE V2 INNER JOIN V2 V3 ON V2.I1 = V3.I1 SET V2.I1 = V2.I1 + 1
WHERE V2.I1 = 10 OR V3.I1 = 11;
1 row updated.
isQL> SELECT * FROM T1 WHERE I1 <> I3 OR I2 <> I4;
T1.I1      T1.I2      T1.I3      T1.I4
-----
2          2          1          2
3          3          2          3
5          5          4          5
11         6          5          6
4 rows selected.

```

- 예상 결과

```

isQL> DROP VIEW V1;
Drop success.
isQL> DROP VIEW V2;
Drop success.
isQL> DROP TABLE T1;
Drop success.
isQL> CREATE TABLE T1( I1 INTEGER, I2 INTEGER, I3 INTEGER, I4 INTEGER );
Create success.
isQL> CREATE VIEW V1 AS SELECT I1, I2 FROM T1;
Create success.
isQL> CREATE VIEW V2 AS SELECT /*+ MATERIALIZE */ I1, I2 FROM T1;
Create success.
isQL> INSERT INTO T1 SELECT ROWNUM, ROWNUM + 1, ROWNUM, ROWNUM + 1 FROM DUAL
CONNECT BY LEVEL <= 20;
20 rows inserted.
isQL>
isQL> SELECT * FROM T1 WHERE I1 <> I3 OR I2 <> I4;
T1.I1      T1.I2      T1.I3      T1.I4
-----
No rows selected.
isQL>
isQL> UPDATE V1 INNER JOIN V1 V3 ON V1.I1 = V3.I1 SET V1.I1 = V1.I1 + 1
WHERE V1.I1 = 1 OR V3.I1 = 2;
2 rows updated.
isQL> SELECT * FROM T1 WHERE I1 <> I3 OR I2 <> I4;
T1.I1      T1.I2      T1.I3      T1.I4
-----
2          2          1          2
3          3          2          3
2 rows selected.
isQL>
isQL> UPDATE V1 INNER JOIN V2 V3 ON V1.I1 = V3.I1 SET V1.I1 = V1.I1 + 1
WHERE V1.I1 = 4 OR V3.I1 = 5;
2 rows updated.
isQL> SELECT * FROM T1 WHERE I1 <> I3 OR I2 <> I4;

```

```

T1.I1      T1.I2      T1.I3      T1.I4
-----
2          2          1          2
3          3          2          3
5          5          4          5
6          6          5          6
4 rows selected.
isQL>
isQL> UPDATE V2 INNER JOIN V1 V3 ON V2.I1 = V3.I1 SET V2.I1 = V2.I1 + 1
WHERE V2.I1 = 7 OR V3.I1 = 8;
1 row updated.
isQL> SELECT * FROM T1 WHERE I1 <> I3 OR I2 <> I4;
T1.I1      T1.I2      T1.I3      T1.I4
-----
2          2          1          2
3          3          2          3
5          5          4          5
6          6          5          6
8          8          7          8
9          9          8          9
6 rows selected.
isQL>
isQL> UPDATE V2 INNER JOIN V2 V3 ON V2.I1 = V3.I1 SET V2.I1 = V2.I1 + 1
WHERE V2.I1 = 10 OR V3.I1 = 11;
1 row updated.
isQL> SELECT * FROM T1 WHERE I1 <> I3 OR I2 <> I4;
T1.I1      T1.I2      T1.I3      T1.I4
-----
2          2          1          2
3          3          2          3
5          5          4          5
6          6          5          6
8          8          7          8
9          9          8          9
10         10         9          10
11         11         10         11
8 rows selected.

```

## Workaround

없음

## 변경사항

- Performance view
- Property
- Compile Option
- Error Code

## BUG-50013 Adapter 로그(jdbcAdapter.trc 또는 oraAdapter.trc)에 DML 정보를 출력하는 형식을 설정하는 프로퍼티를 추가합니다.

### module

rp-jdbcAdapter

### Category

Enhancement

### 재현 빈도

Always

### 설명

Adapter 로그(jdbcAdapter.trc 또는 oraAdapter.trc)에 DML 정보를 출력하는 형식을 설정하는 프로퍼티를 추가합니다.

- Adapter 프로퍼티 : ADAPTER\_XLOG\_WRITE\_TYPE

DML 정보를 출력하는 형식을 설정한다. 0 또는 1을 설정할 수 있으며 기본값을 0이다.

0은 XLog에 기록된 테이블, 컬럼 정보들을 출력하고 1은 SQL 구문으로 출력한다.

- 출력 예

- ADAPTER\_XLOG\_WRITE\_TYPE = 0

```
[2020-06-17 14:25:13] com.mysql.jdbc.MysqlDataTruncation: Data
truncation: Out of range value for column 'market_disp_seq' at row 1
[2020-06-17 14:25:13] INSERT :
[2020-06-17 14:25:13] Table name tcate
[2020-06-17 14:25:13] Table Id 2
[2020-06-17 14:25:13] Column count 17
[2020-06-17 14:25:13] Column name CATE_C
[2020-06-17 14:25:13] Hidden Column NO
[2020-06-17 14:25:13] Column value '220348'
[2020-06-17 14:25:13] Column name NGROUPSEQ
[2020-06-17 14:25:13] Hidden Column NO
[2020-06-17 14:25:13] Column value '0'
[2020-06-17 14:25:13] Column name PCATE_C
[2020-06-17 14:25:13] Hidden Column NO
[2020-06-17 14:25:13] Column value '0'
[2020-06-17 14:25:13] Column name CATE_N
[2020-06-17 14:25:13] Hidden Column NO
[2020-06-17 14:25:13] Column value
[2020-06-17 14:25:13] -----
[2020-06-17 14:25:13] '취미'
[2020-06-17 14:25:13] -----
```

- ADAPTER\_XLOG\_WRITE\_TYPE = 1

```
INSERT INTO INI.BC_CDR VALUES ( '148946461353114', '0502859198',  
'1', '2001', '2017031413101300', '2017031413101402', '01022223333',  
'01093488256', '11112222', '33334444', '0', TO_DATE('null)null',  
'YYYY-MM-DD HH:MI:SSSSSSS') );
```

## 재현 방법

- 재현 절차
- 수행 결과
- 예상 결과

## Workaround

없음

## 변경사항

- Performance view
- Property
- Compile Option
- Error Code

**BUG-50025 CREATE DATABASE 수행 후 바로 STARTUP SERVICE를 수행하면 Altibase 서버가 비정상 종료하거나 ERR-11110 : The index is inconsistent 에러가 발생합니다.**

## module

Sm

## Category

Assert

## 재현 빈도

Frequency

## 설명

CREATE DATABASE 수행 후 바로 STARTUP SERVICE를 수행하면 Altibase 서버가 비정상 종료하거나 ERR-11110 : The index is inconsistent 에러가 발생하는 버그를 수정합니다.

이 버그를 회피하려면 아래 순서로 STARTUP SERVICE를 수행해야 합니다.

- (1) CREATE DATABASE
- (2) SHUTDOWN ABORT
- (3) STARTUP SERVICE

## 재현 방법

- 재현 절차

```
$ is -sysdba
iSQL(sysdba)> STARTUP PROCESS;
iSQL(sysdba)> CREATE DATABASE MYDB INITSIZE=50M NOARCHIVELOG CHARACTER SET
UTF8 NATIONAL CHARACTER SET UTF16;
iSQL(sysdba)> STARTUP SERVICE;
iSQL(sysdba)> CREATE USER ALTITEST IDENTIFIED BY ALTITEST;
iSQL(sysdba)> DROP USER ALTITEST;
```

- 수행 결과

Altibase 서버 비정상 종료 또는

```
iSQL(sysdba)> DROP USER ALTITEST;
[ERR-11110 : The index is inconsistent]
```

- 예상 결과

```
iSQL(sysdba)> DROP USER ALTITEST;Drop success.
```

## Workaround

데이터베이스를 생성한 후 SHUTDOWN ABORT 명령 수행하고 STARTUP을 수행합니다.

```
$ is -sysdba
iSQL(sysdba)> STARTUP PROCESS;
iSQL(sysdba)> CREATE DATABASE MYDB INITSIZE=50M NOARCHIVELOG CHARACTER SET UTF8
NATIONAL CHARACTER SET UTF16;
iSQL(sysdba)> SHUTDOWN ABORT;
iSQL(sysdba)> STARTUP SERVICE;
iSQL(sysdba)> CREATE USER ALTITEST IDENTIFIED BY ALTITEST;
iSQL(sysdba)> DROP USER ALTITEST;
```

## 변경사항

- Performance view
- Property
- Compile Option
- Error Code

**BUG-50033 adapter를 시작할 때 ALA 로그 파일을 오픈할 수 없으면  
코어 파일이 생성됩니다.**

## module

rp-ala

## Category

Assert

## 재현 빈도

Rare

## 설명

adapter를 시작할 때 ALA 로그 파일을 오픈할 수 없으면 시그널을 받고 adapter 프로세스가 강제 종료되며 코어 파일이 생성됩니다. 예외 처리를 변경하여 adapter 트레이스 로그(예, jdbcAdapter.trc)에 다음과 같은 에러 메시지를 출력하도록 개선하였습니다.

ALA library error: 335873, Failed to open log file

## 재현 방법

- 재현 절차
- 수행 결과
- 예상 결과

## Workaround

없음

## 변경사항

- Performance view
- Property
- Compile Option
- Error Code

**BUG-50057 ROLLUP 함수와 집계 함수 그리고 ORDER BY 절을 함께 사용하면 결과 오류가 발생합니다.**

## module

qp-select

## Category

Functional Error

## 재현 빈도

Frequence

## 설명

ROLLUP 함수와 집계 함수 그리고 ORDER BY 절을 함께 사용할 때 발생하는 결과 오류를 수정합니다. 패치 후 버그 조건에 만족하는 질의문 수행 시 결과가 달라질 수 있습니다.



## 재현 방법

- 재현 절차

```
DROP TABLE T1;
CREATE TABLE T1( I1 VARCHAR(10), I2 VARCHAR(10), I3 VARCHAR(10) );
INSERT INTO T1 VALUES(1,1,1);
INSERT INTO T1 VALUES(1,2,2);
INSERT INTO T1 VALUES(3,2,1);
INSERT INTO T1 VALUES(10,2,1);
SELECT /*+ NO_PLAN_CACHE */ TO_CHAR(COUNT(*))
      FROM T1
      GROUP BY ROLLUP(I1), I2
      ORDER BY 1;
```

- 수행 결과

```
TO_CHAR(COUNT(*))
-----
0
0
0
0
0
0
6 rows selected.
```

- 예상 결과

```
TO_CHAR(COUNT(*))
-----
1
1
1
1
1
3
6 rows selected.
```

## Workaround

없음

## 변경사항

- Performance view
- Property
- Compile Option
- Error Code

**BUG-50058 DROP REPLICATION 수행 시 이중화 수신자 쓰레드가 종료되지 않을 수 있는 메모리 참조 오류를 수정합니다.**

**module**

rp

**Category**

Memory Error

**재현 빈도**

Unknown

**설명**

DROP REPLICATION 수행 시 이중화 수신자 쓰레드가 종료되지 않을 수 있는 메모리 참조 오류를 수정합니다. 이 버그가 발생할 가능성은 매우 낮습니다.

**재현 방법**

- 재현 절차
- 수행 결과
- 예상 결과

**Workaround**

없음

**변경사항**

- Performance view
- Property
- Compile Option
- Error Code

**BUG-50059 aku -p end 수행 동작을 변경합니다. 이중화 FLUSH 명령을 수행한 후에 이중화를 중지합니다.**

**module**

aku

**Category**

Reliability

**재현 빈도**

Always

## 설명

aku -p end 수행 동작을 변경합니다. 데이터 불일치가 발생할 가능성을 방지하기 위해, 이중화를 중지하기 전에 aku -p end를 수행한 해당 파드의 이중화 객체에 ALTER REPLICATION ~ FLUSH ALL 명령을 수행하여 변경 데이터를 모두 다른 파드로 전송합니다.

추가된 FLUSH 동작으로 aku -p end 수행이 본 버그를 반영하기 전보다 오래 걸릴 수 있습니다.

관련하여 aku 프로퍼티 AKU\_FLUSH\_AT\_END가 추가되었습니다.

슬레이브 파드에서 aku -p end 명령을 수행할 때 ALTER REPLICATION ~ FLUSH ALL 명령을 수행하여 이중화 갭을 제거할 것인지 설정합니다. 1 이면 이중화 FLUSH ALL 명령으로 이중화 갭을 제거하고 0이면 제거하지 않습니다. 기본값은 1입니다.

## 재현 방법

- 재현 절차
- 수행 결과
- 예상 결과

## Workaround

없음

## 변경사항

- Performance view
- Property
- Compile Option
- Error Code

**BUG-50061 updatable view에 집합 연산자를 사용했을 때 예외 처리를 변경합니다.**

## module

qp

## Category

Functional Error

## 재현 빈도

Always

## 설명

집합 연산자를 사용한 뷰는 updatable view로 사용할 수 없는 제약 사항에 따라 올바른 에러 메시지를 반환하도록 개선합니다. 이 버그를 반영하기 전/후 에러 메시지는 아래와 같습니다.

- 버그 반영 전

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

- 버그 반영 후

ERR-313A4 : Cannot modify a column for a non key-preserved table

## 재현 방법

- 재현 절차

```
CREATE TABLE D1(I1 INTEGER, I2 INTEGER);  
CREATE OR REPLACE VIEW V1 AS SELECT * FROM D1 UNION SELECT * FROM D1;  
DELETE FROM (SELECT * FROM V1);
```

- 수행 결과

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

- 예상 결과

```
ERR-313A4 : Cannot modify a column for a non key-preserved table
```

## Workaround

없음

## 변경사항

- Performance view
- Property
- Compile Option
- Error Code

**BUG-50071 SET 절의 서브쿼리에서 multiple update의 대상을 참조하면 ERR-31455 에러가 발생하거나 Altibase 서버가 비정상 종료할 수 있습니다.**

## module

qp-dm1-pvo

## Category

Fatal

## 재현 빈도

Always

## 설명

SET 절의 서브쿼리에서 multiple update의 대상을 참조하면 ERR-31455 : Failed to work because an internal exception occurred from an OS. 에러가 발생하거나 Altibase 서버가 비정상 종료하는 버그를 수정합니다. 이 버그는 simple view merging 기능이 활성화되어 있을 때 발생합니다.

## 재현 방법

- 재현 절차
- 수행 결과
- 예상 결과

## Workaround

없음

## 변경사항

- Performance view
- Property
- Compile Option
- Error Code

**BUG-50082 비정상적으로 종료된 슬레이브 파드에서 aku -p start 수행 동작을 변경합니다. 이중화 시작 후 FLUSH 명령을 수행합니다.**

## module

aku

## Category

Reliability

## 재현 빈도

Unknown

## 설명

비정상적으로 종료된 슬레이브 파드에서 aku -p start 를 수행할 때 aku의 동작을 변경합니다.

비정상적으로 종료된 슬레이브 파드는 aku -p end 명령을 수행하지 않았거나 정상적으로 완료하지 않아 Altibase에 이중화 정보가 남아있는 파드를 말합니다. 이전의 이중화 정보가 남아있으면 슬레이브 파드를 다시 시작한 후에 데이터 불일치가 발생할 수 있으므로 이전 데이터를 동기화하는 작업을 추가합니다.

1. aku -p start 명령을 수행한 슬레이드 파드에서 Altibase 서버 프로퍼티 ADMIN\_MODE를 1로 설정하여 데이터베이스 사용자의 접근을 막고, 접속된 세션을 강제 종료한다.
2. aku -p start 명령을 수행한 슬레이드 파드에서 이중화를 시작하고 ALTER REPLICATION ~ FLUSH ALL을 수행한다. 이 명령은 다른 파드로 동기화하지 못한 데이터를 전송한다.
3. 원격 파드에서 aku -p start 명령을 수행한 슬레이드 파드로 ALTER REPLICATION ~ FLUSH WAIT *wait\_time*을 수행한다.

4. aku -p start 명령을 수행한 슬라이드 파드에서 Altibase 서버 프로퍼티 ADMIN\_MODE를 0으로 설정하여 데이터베이스 사용자의 접속을 허용한다.

위와 같이 변경된 동작으로 aku -p start 명령 수행 시간이 버그 반영 전보다 오래 걸릴 수 있습니다.

관련하여 aku 프로퍼티 AKU\_FLUSH\_AT\_START, AKU\_FLUSH\_TIMEOUT\_AT\_START가 추가되었습니다.

aku 프로퍼티 설명은 [Altibase 7.1 Utilities Manual - aku-설정-파일](#)을 참고하시기 바랍니다.

## 재현 방법

- 재현 절차
- 수행 결과
- 예상 결과

## Workaround

없음

## 변경사항

- Performance view
- Property
- Compile Option
- Error Code

## BUG-50090 Statement 객체의 다음 결과가 ResultSet 객체가 아닐 때 getMoreResults()에서 true를 반환합니다.

### module

mm-jdbc

### Category

Functional Error

## 재현 빈도

Always

## 설명

Statement 객체의 다음 결과가 ResultSet 객체가 아닐 때 getMoreResults()에서 true를 반환하는 버그를 수정합니다. JDBC API에 따라 false를 반환하도록 수정합니다.

이 버그를 반영하려면 Altibase JDBC 드라이버를 패치해야 합니다.

이 버그가 반영된 Altibase JDBC 드라이버 7.1.0.8.3 이상에서 getMoreResults() 메소드의 수행 결과가 달라집니다.

- 기존 : 최대 ResultSet 객체 수 만큼 true를 반환
- 변경 : ResultSet 객체 수 - 1 만큼 true를 반환

## 재현 방법

- 재현 절차
- 수행 결과
- 예상 결과

## Workaround

없음

## 변경사항

- Performance view
- Property
- Compile Option
- Error Code

## BUG-50094 메모리 테이블의 ADD COLUMN 수행 동작을 변경합니다.

### module

qp

### Category

Functional Error

### 재현 빈도

Always

### 설명

Altibase 7.1 에는 메모리 테이블의 컬럼을 추가할 때 다른 트랜잭션에 영향을 주지 않는 '실시간 DDL' 기능이 추가되었습니다.

이 기능은 추가할 컬럼의 크기에 상관없이 무조건 VARIABLE로 컬럼을 생성하는 부분에서 아래와 같은 문제점이 발견되어 ADD COLUMN의 내부 동작을 변경합니다.

- (1) DESC table\_name 명령을 수행할 때 CREATE로 생성한 컬럼과 추가된 컬럼의 저장 정보가 다름
- (2) INTEGER와 같은 FIXED 기반의 데이터 타입도 VARIABLE로 저장되며 테이블 크기 증가
- (3) 테이블 크기 증가로 변경 트랜잭션 성능 저하

Altibase 7.1.0.8.3 이상에서 메모리 테이블의 ADD COLUMN 수행 동작은 다음과 같습니다.

'실시간 DDL' 수행 조건이 추가할 컬럼의 크기에 따라 결정됩니다.

- '실시간 DDL' 동작하는 경우  
추가할 컬럼의 크기가 MEMORY\_VARIABLE\_COLUMN\_IN\_ROW\_SIZE에서 지정한 크기보다 크면 VARIABLE 컬럼으로 생성되고 실시간 DDL이 동작합니다.

- '실시간 DDL' 동작하지 않는 경우

추가할 컬럼의 크기가 MEMORY\_VARIABLE\_COLUMN\_IN\_ROW\_SIZE에서 지정한 크기보다 작거나 같으면 FIXED 컬럼으로 생성하고 실시간 DDL이 동작하지 않습니다.

실시간 DDL이 동작하지 않으면 ADD COLUMN 수행은 아래와 같이 처리됩니다.

- 내부적으로 대상 테이블을 재구성합니다.
- 이때, 대상 테이블에 X 잠금을 획득하므로 ADD COLUMN 작업이 완료할 때까지 대상 테이블에 접근할 수 없습니다.
- 대상 데이터 크기에 비례하여 메모리 사용량이 증가합니다.

Altibase 7.1.0.8.3 이전 버전과 같이 동작하게 하려면 Altibase 기술 지원 센터로 문의해주시기 바랍니다.

## 재현 방법

- 재현 절차
- 수행 결과
- 예상 결과

## Workaround

없음

## 변경사항

- Performance view
- Property
- Compile Option
- Error Code

**BUG-50096 DDL 복제 실패 이후 HeartBeat 쓰레드의 비정상적인 동작으로 이중화 송신자가 시작되지 않을 수 있습니다.**

## module

rp

## Category

Functional Error

## 재현 빈도

Always

## 설명

DDL 복제 실패 후 이중화를 종료할 때 HeartBeat 쓰레드가 정상적으로 종료되지 않는 문제로, 이중화 송신자가 시작되지 않는 현상을 수정합니다.



## 재현 방법

- 재현 절차
- 수행 결과
- 예상 결과

## Workaround

없음

## 변경사항

- Performance view
- Property
- Compile Option
- Error Code

**BUG-50098 delete 또는 Multiple delete 문의 대상 테이블이 부질의에 MATERIALIZE 힌트를 사용한 뷰일 때 결과 오류가 발생할 수 있습니다.**

## module

qp-dml-pvo

## Category

Functional Error

## 재현 빈도

Always

## 설명

delete 또는 Multiple delete 문의 대상 테이블이 부질의에 MATERIALIZE 힌트를 사용한 뷰일 때 결과 오류가 발생하는 버그를 수정합니다.

## 재현 방법

- 재현 절차

```
CREATE TABLE T1( I1 INTEGER );
CREATE VIEW V1 AS SELECT I1 I1 FROM T1;
CREATE VIEW V2 AS SELECT /*+ MATERIALIZE */ I1 I1 FROM T1;

TRUNCATE TABLE T1;

INSERT INTO T1 VALUES ( 1 );
INSERT INTO T1 VALUES ( 2 );
DELETE FROM V1 T1 WHERE T1.I1 = ( SELECT * FROM V1 T2 WHERE T2.I1 = 1 );
SELECT * FROM T1;
```

```

TRUNCATE TABLE T1;

INSERT INTO T1 VALUES ( 1 );
INSERT INTO T1 VALUES ( 2 );
DELETE FROM V2 T1 WHERE T1.I1 = ( SELECT * FROM V1 T2 WHERE T2.I1 = 1 );
SELECT * FROM T1;

TRUNCATE TABLE T1;

INSERT INTO T1 VALUES ( 1 );
INSERT INTO T1 VALUES ( 2 );
DELETE FROM T1 USING V2 T1 INNER JOIN V1 T2 ON T1.I1 = T2.I1 WHERE T2.I1 = 1;
SELECT * FROM T1;

```

- 수행 결과

```

isQL> CREATE TABLE T1( I1 INTEGER );
Create success.
isQL> CREATE VIEW V1 AS SELECT I1 I1 FROM T1;
Create success.
isQL> CREATE VIEW V2 AS SELECT /*+ MATERIALIZE */ I1 I1 FROM T1;
Create success.
isQL>
isQL> TRUNCATE TABLE T1;
Truncate success.
isQL>
isQL> INSERT INTO T1 VALUES ( 1 );
1 row inserted.
isQL> INSERT INTO T1 VALUES ( 2 );
1 row inserted.
isQL> DELETE FROM V1 T1 WHERE T1.I1 = ( SELECT * FROM V1 T2 WHERE T2.I1 = 1
);
1 row deleted.
isQL> SELECT * FROM T1;
I1
-----
1
1 row selected.
isQL>
isQL>
isQL> TRUNCATE TABLE T1;
Truncate success.
isQL>
isQL> INSERT INTO T1 VALUES ( 1 );
1 row inserted.
isQL> INSERT INTO T1 VALUES ( 2 );
1 row inserted.
isQL> DELETE FROM V2 T1 WHERE T1.I1 = ( SELECT * FROM V1 T2 WHERE T2.I1 = 1
);
1 row deleted.
isQL> SELECT * FROM T1;
I1
-----
1
1 row selected.

```

```

isQL>
isQL>
isQL> TRUNCATE TABLE T1;
Truncate success.
isQL>
isQL> INSERT INTO T1 VALUES ( 1 );
1 row inserted.
isQL> INSERT INTO T1 VALUES ( 2 );
1 row inserted.
isQL> DELETE FROM T1 USING V2 T1 INNER JOIN V1 T2 ON T1.I1 = T2.I1 WHERE
T2.I1 = 1;
1 row deleted.
isQL> SELECT * FROM T1;
I1
-----
1
1 row selected.

```

- 예상 결과

```

isQL> CREATE TABLE T1( I1 INTEGER );
Create success.
isQL> CREATE VIEW V1 AS SELECT I1 I1 FROM T1;
Create success.
isQL> CREATE VIEW V2 AS SELECT /*+ MATERIALIZE */ I1 I1 FROM T1;
Create success.
isQL>
isQL> TRUNCATE TABLE T1;
Truncate success.
isQL>
isQL> INSERT INTO T1 VALUES ( 1 );
1 row inserted.
isQL> INSERT INTO T1 VALUES ( 2 );
1 row inserted.
isQL> DELETE FROM V1 T1 WHERE T1.I1 = ( SELECT * FROM V1 T2 WHERE T2.I1 = 1
);
1 row deleted.
isQL> SELECT * FROM T1;
I1
-----
2
1 row selected.
isQL>
isQL>
isQL> TRUNCATE TABLE T1;
Truncate success.
isQL>
isQL> INSERT INTO T1 VALUES ( 1 );
1 row inserted.
isQL> INSERT INTO T1 VALUES ( 2 );
1 row inserted.
isQL> DELETE FROM V2 T1 WHERE T1.I1 = ( SELECT * FROM V1 T2 WHERE T2.I1 = 1
);
1 row deleted.
isQL> SELECT * FROM T1;

```

```

I1
-----
2
1 row selected.
isQL>
isQL>
isQL> TRUNCATE TABLE T1;
Truncate success.
isQL>
isQL> INSERT INTO T1 VALUES ( 1 );
1 row inserted.
isQL> INSERT INTO T1 VALUES ( 2 );
1 row inserted.
isQL> DELETE FROM T1 USING V2 T1 INNER JOIN V1 T2 ON T1.I1 = T2.I1 WHERE
T2.I1 = 1;
1 row deleted.
isQL> SELECT * FROM T1;
I1
-----
2
1 row selected.

```

## Workaround

없음

## 변경사항

- Performance view
- Property
- Compile Option
- Error Code

**BUG-50106 isQL에서 desc 명령을 수행하면 ERR-31002 : A single-row subquery has returned more than one row. 에러가 발생합니다.**

## module

ux-isql

## Category

Functional Error

## 재현 빈도

Always

## 설명

iSQL에서 desc 명령을 수행하면 ERR-31002 : A single-row subquery has returned more than one row. 에러가 발생하는 버그를 수정합니다. 이 버그는 아래 조건을 만족할 때 발생합니다.

- 서로 다른 데이터베이스 사용자가 같은 이름의 테이블 A를 생성
- iSQL에서 SET PARTITIONS ON 수행 후
- DESC A 수행

## 재현 방법

- 재현 절차

```
CONNECT SYS/MANAGER
CREATE TABLE T1 (C1 INTEGER, C2 INTEGER);

CREATE USER USER1 IDENTIFIED BY USER1;
CONNECT USER1/USER1
CREATE TABLE T1 (C1 INTEGER, C2 INTEGER);
SET PARTITIONS ON;
DESC T1;
```

- 수행 결과

```
DESC T1;
[ TABLESPACE : SYS_TBS_MEM_DATA ]
[ ATTRIBUTE ]

-----
-
NAME                                TYPE                                IS NULL
-----
-
C1                                INTEGER                            FIXED
C2                                INTEGER                            FIXED
T1 has no index
T1 has no primary key
[ERR-31002 : A single-row subquery has returned more than one row.]
```

- 예상 결과

```
DESC T1;
[ TABLESPACE : SYS_TBS_MEM_DATA ]
[ ATTRIBUTE ]

-----
-
NAME                                TYPE                                IS NULL
-----
-
C1                                INTEGER                            FIXED
C2                                INTEGER                            FIXED
T1 has no index
T1 has no primary key
```

## Workaround

없음

## 변경사항

- Performance view
- Property
- Compile Option
- Error Code

**BUG-50131 조건절 검사 시 Altibase 서버가 비정상 종료하는 현상에 대한 예외처리를 추가합니다.**

## module

qp-dml-execute

## Category

Fatal

## 재현 빈도

Impossible

## 설명

조건절 검사 시 Altibase 서버가 비정상 종료하는 현상에 대한 예외처리를 추가합니다.

## 재현 방법

- 재현 절차
- 수행 결과
- 예상 결과

## Workaround

없음

## 변경사항

- Performance view
- Property
- Compile Option
- Error Code

# Changes

## Version Info

altibase version	database binary version	meta version	cm protocol version	replication protocol version
7.1.0.8.3	6.5.1	8.10.1	7.1.7	7.4.7

Altibase 7.1 패치 버전별 히스토리는 [Version Histories](#) 에서 확인할 수 있다.

## 호환성

### Database binary version

데이터베이스 바이너리 버전은 변경되지 않았다.

### Meta Version

메타 버전은 변경되지 않았다.

### 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