

Altibase 7.1.0.5.6 Patch Notes

Fixed Bugs

BUG-48888 JDBC API Specification 4.2를 지원하는 JDBC 드라이버(Altibase42.jar)가 추가되었습니다.

BUG-48902 CONVERT 함수 사용 시 source_char_set 이 dest_char_set 을 초과하는 경우 Altibase 서버가 비정상 종료합니다.

BUG-48905 분석함수 ROW_NUMBER 를 LIMIT k와 동일한 의미로 사용한 쿼리 성능을 개선하였습니다.

BUG-48920 디스크 인덱스 키 삽입 과정에서 인덱스 노드 공간 활용을 위해 인덱스 구조 변경이 발생할 때 Altibase 서버가 비정상 종료할 수 있습니다.

BUG-48944 중첩된 LEFT OUTER JOIN 수행 방식을 최적화합니다.

BUG-48969 TRCLOG_EXPLAIN_GRAPH = 1 에서 SELECT 절에 사용된 서브쿼리 정보를 출력하도록 개선합니다.

BUG-48971 레코드가 없는 테이블을 통계정보 수집 시 잘못된 NDV (Number of Distinct Value) 설정으로 FULL SCAN 가능성이 높아지는 현상을 개선합니다.

BUG-48995 Adapter for Oracle, Adapter for JDBC 에 오프라인 이중화 메타 복제(Offline Option) 기능이 추가되었습니다.

Changes

Version Info

호환성

Database binary version

Meta Version

CM protocol Version

Replication protocol Version

프로퍼티

추가된 프로퍼티

변경된 프로퍼티

삭제된 프로퍼티

성능 뷰

추가된 성능 뷰

변경된 성능 뷰

삭제된 성능 뷰

Altibase 7.1.0.5.6 Patch Notes

Fixed Bugs

BUG-48888 JDBC API Specification 4.2를 지원하는 JDBC 드라이버 (Altibase42.jar)가 추가되었습니다.

- **module** : mm-jdbc
- **Category** : Enhancement
- **재현 빈도** : Always
- **설명** : JDBC API Specification 4.2를 지원하는 Altibase JDBC 드라이버(Altibase42.jar)가 추가되었습니다. Altibase42.jar 를 사용하기 위해서 JRE 1.8 이상이 필요합니다.

보다 자세한 설명은 JDBC User's Manual 을 참고하세요.

- [JDBC User's Manual - 1.JDBC 시작하기 - JDBC 드라이버 설치](#)
- [JDBC User's Manual - 6.JDBC 4.2 API References](#)

- 재현 방법
 - 재현 절차
 - 수행 결과
 - 예상 결과
- Workaround
- 변경사항
 - Performance view
 - Property
 - Compile Option
 - Error Code

BUG-48902 CONVERT 함수 사용 시 source_char_set 이 dest_char_set 을 초과하는 경우 Altibase 서버가 비정상 종료합니다.

- module : qp
- Category : Fatal
- 재현 빈도 : Always
- 설명 : CONVERT 함수 사용 시 source_char_set 이 dest_char_set 을 초과하는 경우 예외 처리 부족으로 Altibase 서버가 비정상 종료하는 현상을 수정합니다.
- 재현 방법
 - 재현 절차

```
DROP TABLE BUG_48902;
CREATE TABLE BUG_48902(I1 DECIMAL);
INSERT INTO BUG_48902 VALUES ( -3 );
SELECT CONVERT(I1, 'UTF8', 'UTF8' ) FROM BUG_48902;
```

- 수행 결과

```
ERR-91015 : Communication failure.
Altibase 서버 비정상 종료
```

- 예상 결과

```
ERR-2101D : Invalid data length
```

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

BUG-48905 분석함수 ROW_NUMBER 를 LIMIT k와 동일한 의미로 사용한 쿼리 성능을 개선하였습니다.

- **module** : qp-dml-execute
- **Category** : Enhancement
- **재현 빈도** : Always
- **설명** : 분석함수 ROW_NUMBER 를 LIMIT k와 동일한 의미로 사용한 쿼리 성능을 개선하였습니다.

LIMIT 의 마지막 값 조건에 바인드 변수를 사용한 경우 쿼리 성능이 향상되었습니다.

- **재현 방법**
 - 재현 절차

```
var a integer
var b integer
exec :a := 5;
exec :b := 0;
DROP TABLE T1;
CREATE TABLE T1 ( I1 INT PRIMARY KEY, I2 INT, I3 INT);
INSERT INTO T1 ( SELECT LEVEL, 100-LEVEL, MOD(LEVEL, 2) FROM DUAL
CONNECT BY LEVEL <= 2000000);

SET LINESIZE 1024
SET COLSIZE 10
SET TIMING ON
ALTER SESSION SET EXPLAIN PLAN = ON;
ALTER SESSION SET TRCLOG_DETAIL_PREDICATE = 1;
PREPARE SELECT I1, I2, I3
FROM ( SELECT T1.*
      , ROW_NUMBER() OVER (ORDER BY I2, I3) AS RN
      FROM T1)
WHERE RN < :A AND RN > :B;
```

- 수행 결과

```
I1          I2          I3
-----
2000000      -1999900      0
1999999      -1999899      1
1999998      -1999898      0
1999997      -1999897      1
4 rows selected.

PROJECT ( COLUMN_COUNT: 3, TUPLE_SIZE: 12, COST: 280.00 )
FILTER
[ FILTER ]
AND
RN > :B
RN < :A
VIEW ( ACCESS: 2000000, COST: 279.56 )
PROJECT ( COLUMN_COUNT: 4, TUPLE_SIZE: 24, COST: 152.24 )
WINDOW SORT ( ITEM_SIZE: 24, ITEM_COUNT: 2000000, ACCESS: 5999999,
SORT_COUNT: 1, COST: 146.97 )
```

```

[ ANALYTIC FUNCTION INFO ]
SORT_KEY[0]: (I2,I3)
ROW_NUMBER() OVER (ORDER BY I2, I3)
SCAN ( TABLE: SYS.T1, FULL SCAN, ACCESS: 2000000, COST: 116.76 )
-----
elapsed time : 1.54

```

○ 예상 결과

```

I1          I2          I3
-----
2000000     -1999900     0
1999999     -1999899     1
1999998     -1999898     0
1999997     -1999897     1
4 rows selected.
-----
PROJECT ( COLUMN_COUNT: 3, TUPLE_SIZE: 12, COST: 283.52 )
FILTER
[ FILTER ]
AND
RN > :B
RN < :A
VIEW ( ACCESS: 5, COST: 282.20 )
PROJECT ( COLUMN_COUNT: 4, TUPLE_SIZE: 24, COST: 154.88 )
WINDOW SORT ( ITEM_SIZE: 24, ITEM_COUNT: 5, ACCESS: 14, SORT_COUNT:
1, COST: 146.97 )
[ ANALYTIC FUNCTION INFO ]
SORT_KEY[0]: (I2,I3)
ROW_NUMBER_LIMIT(:A) OVER (ORDER BY I2, I3)
SCAN ( TABLE: SYS.T1, FULL SCAN, ACCESS: 2000000, COST: 116.76 )
-----
elapsed time : 0.39

```

• Workaround

```

PREPARE SELECT I1, I2, I3
FROM ( SELECT T1.*
      , ROW_NUMBER() OVER (ORDER BY I2, I3) AS RN
      FROM T1)
WHERE RN < 5 AND RN > :B;

```

• 변경사항

- Performance view
- Property
- Compile Option
- Error Code

BUG-48920 디스크 인덱스 키 삽입 과정에서 인덱스 노드 공간 활용을 위해 인덱스 구조 변경이 발생할 때 Altibase 서버가 비정상 종료할 수 있습니다.

- **module** : sm
- **Category** : Assert
- **재현 빈도** : Rare
- **설명** : 디스크 인덱스 키 삽입 과정에서 인덱스 노드 공간 활용을 위해 인덱스 구조 변경이 발생할 경우 Altibase 서버가 비정상 종료하는 현상을 수정합니다.
- **재현 방법**
 - 재현 절차
 - 수행 결과
 - 예상 결과
- **Workaround**
- **변경사항**
 - Performance view
 - Property
 - Compile Option
 - Error Code

BUG-48944 중첩된 LEFT OUTER JOIN 수행 방식을 최적화합니다.

- **module** : qp-dml-execute
- **Category** : Efficiency
- **재현 빈도** : Always
- **설명** : 중첩된 LEFT OUTER JOIN 에서 불필요한 테이블 읽기를 제거합니다.

본 버그에서 최적화한 기능을 사용하려면
__LEFT_OUTER_SKIP_RIGHT_ENABLE = 1 을 설정해야 합니다.
- **재현 방법**
 - 재현 절차

```
DROP TABLE A;
DROP TABLE B;
DROP TABLE C;
DROP TABLE D;
CREATE TABLE A(i1 INTEGER, i2 INTEGER);
CREATE TABLE B(i1 INTEGER, i2 INTEGER);
CREATE TABLE C(i1 INTEGER, i2 INTEGER);
CREATE TABLE D(i1 INTEGER, i2 INTEGER);
INSERT INTO A SELECT LEVEL, LEVEL FROM DUAL CONNECT BY LEVEL <= 100;
INSERT INTO B SELECT LEVEL, LEVEL FROM DUAL CONNECT BY LEVEL <= 100;
INSERT INTO C SELECT LEVEL, LEVEL FROM DUAL CONNECT BY LEVEL <= 100;
INSERT INTO D SELECT LEVEL, LEVEL FROM DUAL CONNECT BY LEVEL <= 100;
ALTER TABLE A ADD CONSTRAINT IX_A_PK PRIMARY KEY( i1 );
ALTER TABLE B ADD CONSTRAINT IX_B_PK PRIMARY KEY( i1 );
ALTER TABLE C ADD CONSTRAINT IX_C_PK PRIMARY KEY( i1 );
ALTER TABLE D ADD CONSTRAINT IX_D_PK PRIMARY KEY( i1 );
```

```

set colsize 20;
set linesize 200;
ALTER SESSION SET EXPLAIN PLAN = ON;
ALTER SESSION SET TRCLOG_DETAIL_INFORMATION = 1;
SELECT /*+ USE_NL(A B C D) */
      SUM(1)
      , SUM(NVL2(A.i2,1,0))
      , SUM(NVL2(B.i2,1,0))
      , SUM(NVL2(C.i2,1,0))
      , SUM(NVL2(D.i2,1,0))
FROM A
      LEFT OUTER JOIN B ON B.i1 = A.i1 AND B.i2 = -1
      LEFT OUTER JOIN C ON C.i1 = B.i1
      LEFT OUTER JOIN D ON D.i1 = C.i1
WHERE 1=1 AND A.i1 BETWEEN 10 AND 20;

```

○ 수행 결과

```

-----
PROJECT ( COLUMN_COUNT: 5, TUPLE_SIZE: 40, COST: 672716084.89 )
  GROUP-AGGREGATION ( ITEM_SIZE: 96, GROUP_COUNT: 1, BUCKET_COUNT: 1,
ACCESS: 1, COST: 672716084.89 )
    LEFT-OUTER-JOIN ( METHOD: INDEX_NL, COST: 9558146.46 )
      LEFT-OUTER-JOIN ( METHOD: INDEX_NL, COST: 9459.99 )
        LEFT-OUTER-JOIN ( METHOD: INDEX_NL, COST: 136.01 )
          SCAN ( TABLE: SYS.A, INDEX: SYS.IX_A_PK, RANGE SCAN, ACCESS: 11,
COST: 0.01 )
            [ FIXED KEY ]
            AND
            OR
            A.I1 BETWEEN 10 AND 20
          SCAN ( TABLE: SYS.B, INDEX: SYS.IX_B_PK, RANGE SCAN, ACCESS: 22,
COST: 120.72 )
            [ VARIABLE KEY ]
            OR
            AND
            B.I1 = A.I1
          [ FILTER ]
          B.I2 = -1
        SCAN ( TABLE: SYS.C, INDEX: SYS.IX_C_PK, RANGE SCAN, ACCESS: 11,
COST: 116.76 )
          [ VARIABLE KEY ]
          OR
          AND
          C.I1 = B.I1
        SCAN ( TABLE: SYS.D, INDEX: SYS.IX_D_PK, RANGE SCAN, ACCESS: 11,
COST: 116.76 )
          [ VARIABLE KEY ]
          OR
          AND
          D.I1 = C.I1
-----

```

○ 예상 결과

```


```

LEFT-OUTER-JOIN 노드의 SKIP RIGHT COUNT 정보로 최적화 여부를 확인할 수 있습니다.

```
-----
PROJECT ( COLUMN_COUNT: 5, TUPLE_SIZE: 40, COST: 672716084.89 )
  GROUP-AGGREGATION ( ITEM_SIZE: 96, GROUP_COUNT: 1, BUCKET_COUNT: 1,
ACCESS: 1, COST: 672716084.89 )
    LEFT-OUTER-JOIN ( METHOD: INDEX_NL, SKIP RIGHT COUNT: 11, COST:
9558146.46 )
      LEFT-OUTER-JOIN ( METHOD: INDEX_NL, SKIP RIGHT COUNT: 11, COST:
9459.99 )
        LEFT-OUTER-JOIN ( METHOD: INDEX_NL, COST: 136.01 )
          SCAN ( TABLE: SYS.A, INDEX: SYS.IX_A_PK, RANGE SCAN, ACCESS: 11,
COST: 0.01 )
            [ FIXED KEY ]
            AND
            OR
            A.I1 BETWEEN 10 AND 20
          SCAN ( TABLE: SYS.B, INDEX: SYS.IX_B_PK, RANGE SCAN, ACCESS: 22,
COST: 120.72 )
            [ VARIABLE KEY ]
            OR
            AND
            B.I1 = A.I1
          [ FILTER ]
          B.I2 = -1
        SCAN ( TABLE: SYS.C, INDEX: SYS.IX_C_PK, RANGE SCAN, ACCESS: 11,
COST: 116.76 )
          [ VARIABLE KEY ]
          OR
          AND
          C.I1 = B.I1
        SCAN ( TABLE: SYS.D, INDEX: SYS.IX_D_PK, RANGE SCAN, ACCESS: 11,
COST: 116.76 )
          [ VARIABLE KEY ]
          OR
          AND
          D.I1 = C.I1
-----
```

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

BUG-48969 TRCLOG_EXPLAIN_GRAPH = 1 에서 SELECT 절에 사용된 서브쿼리 정보를 출력하도록 개선합니다.

- **module** : qp-dml-pvo
- **Category** : Enhancement
- **재현 빈도** : Always
- **설명** : TRCLOG_EXPLAIN_GRAPH = 1 에서 SELECT 절에 사용된 서브쿼리 정보를 출력하도록 개선합니다.

- 재현 방법

- 재현 절차

```
@ ?/sample/APRE/schema/schema
ALTER SYSTEM SET TRCLOG_EXPLAIN_GRAPH=1;
CONNECT SYS/MANAGER
ALTER SESSION SET EXPLAIN PLAN = ONLY;
SELECT ENO, E_LASTNAME, DNO,
       ( SELECT DNAME FROM DEPARTMENTS D
         WHERE E.DNO = D.DNO) DNAME
FROM EMPLOYEES E;
```

- 수행 결과

```
||-----
||-----
||[[ PROJECTION GRAPH ]]
||-----
||== Cost Information ==
||INPUT_RECORD_COUNT : 10240
||OUTPUT_RECORD_COUNT: 10240
||RECORD_SIZE       : 28
||SELECTIVITY       : 1
...중략...
| |INDEX SCAN[EMP_IDX1]
| |  ACCESS_COST : 10241
| |  DISK_COST   : 0
| |  TOTAL_COST  : 10241
||-----
```

- 예상 결과

```
SUB-QUERY 그래프가 추가됨
||-----
||-----
||[[ PROJECTION GRAPH ]]
||-----
||== Cost Information ==
||INPUT_RECORD_COUNT : 10240
||OUTPUT_RECORD_COUNT: 10240
||RECORD_SIZE       : 28
||SELECTIVITY       : 1
||GRAPH_ACCESS_COST  : 591.24759
||GRAPH_DISK_COST    : 0
||GRAPH_TOTAL_COST   : 591.24759
||TOTAL_ACCESS_COST  : 10831.24759
||TOTAL_DISK_COST    : 0
||TOTAL_ALL_COST     : 10831.24759
||
||:::SUB-QUERY BEGIN
||
| |-----
| |[[ PROJECTION GRAPH ]]
| |-----
| |== Cost Information ==
```



```

| |INPUT_RECORD_COUNT : 1024
| |OUTPUT_RECORD_COUNT: 1024
| |RECORD_SIZE       : 32
...중략...
| |INDEX SCAN[DEP_IDX1]
| |ACCESS_COST : 348.136
| |DISK_COST   : 0
| |TOTAL_COST  : 348.136
||
||:::SUB-QUERY END
||
...중략...
| |INDEX SCAN[EMP_IDX1]
| |ACCESS_COST : 10241
| |DISK_COST   : 0
| |TOTAL_COST  : 10241
||-----

```

- **Workaround**

- **변경사항**

- Performance view
- Property
- Compile Option
- Error Code

BUG-48971 레코드가 없는 테이블을 통계정보 수집 시 잘못된 NDV (Number of Distinct Value) 설정으로 FULL SCAN 가능성이 높아지는 현상을 개선합니다.

- **module** : qp-select-pvo

- **Category** : Functional Error

- **재현 빈도** : Frequence

- **설명** : 레코드가 없는 테이블을 통계정보 수집 시 NDV (Number of Distinct Value) 가 잘못 산정되는 현상을 개선합니다.

NDV 가 잘못 산정된 경우 INDEX SCAN 보다 FULL SCAN 을 선택하여 쿼리 성능에 영향을 줄 수 있습니다.

- **재현 방법**

- 재현 절차
- 수행 결과
- 예상 결과

- **Workaround**

- **변경사항**

- Performance view
- Property
- Compile Option
- Error Code

BUG-48995 Adapter for Oracle, Adapter for JDBC 에 오프라인 이중화 메타 복제(Offline Option) 기능이 추가되었습니다.

- **module** : dm
- **Category** : Other
- **재현 빈도** : Always
- **설명** : Adapter for Oracle, Adapter for JDBC 운영 중 Altibase 서버 장애 발생 시 동기화하지 못한 데이터를 오프라인 옵션(Offline Option) 기능으로 동기화할 수 있습니다.

이 기능을 사용하기 위해서 아래의 조건을 만족해야 합니다.

- 이중화 객체 생성 시 META_LOGGING 옵션 사용
- Active-Standby 로 구성된 Altibase 이중화 환경 구성

보다 자세한 설명은 매뉴얼을 참고하세요.

- [Adapter for Oracle User's Manual - 3.사용법 - 오프라인 옵션\(Offline Option\)](#)
- [Adapter for JDBC User's Manual - 3.사용법 - 오프라인 옵션\(Offline Option\)](#)

- **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
7.1.0.5.6	6.5.1	8.9.1	7.1.7	7.4.6

Altibase 7.1 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