

# Altibase 7.3.0.0.7 Patch Notes

---

# Table of Contents

---

- [New Features](#)

- [BUG-47423 APRE에서 RETURNING INTO 절을 지원하도록 개선하였습니다.](#)
- [BUG-50868 AKU 다중 이중화 지원 및 기능 개선](#)
- [BUG-50878 aexport,properties 파일에서 ILOADER FIELD TERM, ILOADER ROW TERM 프로퍼티의 값을 변경합니다.](#)
- [BUG-50912 TCP 소켓 옵션 SO\\_LINGER를 설정하는 연결 속성 socket immediate close를 추가합니다.](#)
- [BUG-50249 PICL 라이브러리에 예외상황 발생시 로그에 남기는 기능을 추가합니다. \(AIX, HP-UX\).](#)
- [BUG-50005 aku 실행 시 옵션을 입력하지 않거나 잘못된 옵션을 입력했을 때, 오류 메시지를 출력하지 않고 help 메시지만 출력하도록 변경하였습니다.](#)

- [Fixed Bugs](#)

- [BUG-50182 테이블의 행\(Row\)을 삭제하는 도중에 ALTER TABLE ... ADD COLUMN... DDL이 실행되면, 잘못된 메모리에 접근하는 오류가 발생할 수 있습니다.](#)
- [BUG-50810 API를 통한 LOB 데이터 작업시, "Internal server error"가 발생할 수 있습니다.](#)
- [BUG-50849 escape syntax에 대문자가 포함된 경우 정상적으로 파싱처리가 되지 않습니다.](#)
- [BUG-50862 송신자로부터 이중화 시작 요청을 수신하는 동시에 수신자가 이중화를 삭제하는 경우, 동시성 문제가 발생할 수 있습니다.](#)
- [BUG-50864 로그 파일의 크기가 비정상적으로 변경된 경우, 잘못된 에러 메시지가 출력됩니다.](#)
- [BUG-50866 온라인 백업된 파일을 이용해서 복구를 시도할 때 MustRedo LSN까지의 로그파일 없는 경우, 복구가 실패했음에도 불구하고 "Database media recovery successful" 메시지를 출력합니다.](#)
- [BUG-50875 Logical Ager가 삭제해야 할 인덱스 키를 찾지 못할 때 Altibase 서버가 비정상 종료합니다.](#)
- [BUG-50881 IPCDA 접속 중에 클라이언트가 종료하면 서버가 비정상 종료합니다.](#)
- [BUG-50891 사용하지 않는 INSPECTION LARGE HEAP THRESHOLD 프로퍼티를 삭제하였습니다.](#)
- [BUG-50905 ODBC 연결 스트링에 DSN 속성을 입력하지 않으면 Invalid attribute value. 에러가 발생합니다.](#)
- [BUG-50907 조인에서 하이브리드 파티션드 테이블이 사용되고 SERIAL EXECUTE MODE 프로퍼티가 1일 때, 결과 값에 오류가 발생할 수 있습니다.](#)
- [BUG-50914 altibase store result, altibase next result를 순차적으로 실행할 때, 두 번째 결과 집합에서 HY010 Function sequence error. 에러가 발생합니다.](#)
- [BUG-50919 altibase stmt next result 함수를 실행할 때, 두 번째 결과 집합에서 "HY010 Function sequence error." 에러가 발생합니다.](#)
- [BUG-50920 데이터 업로드 시 에러가 발생한 경우, 어느 칼럼에서 발생했는지 알 수 없습니다.](#)
- [BUG-50923 SQLPrepare 함수의 SQL 텍스트 문자열 인자가 빈 문자열일 때 에러가 발생하지 않습니다.](#)
- [BUG-50938 AKU 에서 이용하는 문자열 값의 길이 제한이 실제와 다르게 설정되어 있습니다.](#)

- [BUG-50948 서버에 fetch할 데이터가 남아 있는 경우 ResultSet.close\(\)를 해도 커서가 바로 닫히지 않을 수 있습니다.](#)
- [BUG-50949 aku 설정 파일에 주석 입력시, # 이후에 글자를 입력하지 않는 경우 오류가 발생합니다.](#)
- [BUG-50950 aku -p start를 실행할 때 첫 번째 이중화 객체 생성이 실패해도 AKU가 즉시 종료되지 않습니다.](#)
- [BUG-50969 RESET PARAMS가 내부적으로 수행될 때 bakBindParam도 제거되어야 합니다](#)
- [BUG-50975 SIMPLE QUERY 최적화 기능을 활성화하고 JDBC 연결 속성에 remove redundant transmission을 사용할 때, SQL 문 수행 중 메모리 오류가 발생할 수 있습니다.](#)
- [BUG-50997 외부 쿼리를 참조하는 서브 쿼리에서 집계 함수를 사용하는 경우, 특정 조건에서 결과값 오류가 발생합니다.](#)
- [BUG-51004 altibase stmt bind\\_param에 바인딩하는 변수의 타입 또는 포인터가 바뀌는 경우, "Function sequence error"가 발생할 수 있습니다.](#)
- [BUG-51012 SQLColumns함수에서 SchemaName\(스키마 이름\) 인자에 NULL이 전달될 경우, 잘못된 데이터가 반환됩니다.](#)
- [BUG-51013 ODBC 드라이버를 이용하여 다이너셋\(dynaset\) 테스트를 수행할 때, Altibase 데이터베이스에 연결이 성공했으나 다이너셋을 지원하지 않는다는 오류가 발생합니다.](#)
- [Changes](#)
  - [Version Info](#)
  - [호환성](#)
  - [프로퍼티](#)
  - [성능](#)

# New Features

## BUG-47423 APRE에서 RETURNING INTO 절을 지원하도록 개선하였습니다.

- **module** : mm-apre
- **Category** : Functionality
- **재현 빈도** : Always
- **설명** : APRE에서 RETURNING INTO 절을 지원하도록 개선하였습니다.
- **재현 방법**
  - 재현 절차
  - 수행 결과
  - 예상 결과
- **Workaround**
- **변경사항**
  - Performance view
  - Property
  - Compile Option
  - Error Code

## BUG-50868 AKU 다중 이중화 지원 및 기능 개선

- **module** : aku
- **Category** : Enhancement
- **재현 빈도** : Always
- **설명** : AKU에서 이중화는 단일 일중화만 지원했으나, 이제 다중 이중화를 지원하도록 개선되었습니다.
  - aku.conf 파일에 다중 이중화를 설정하려면, 아래와 같이 REPLICATIONS 항목에 쉼표로 구분하여 추가할 수 있습니다. 자세한 내용은 **Utilitis 매뉴얼 - 3.aku** 에서 [이중화 테이블을 설정하는 방법](#)을 참고하세요.
    - 예) REPLICATIONS = (이중화 설정1), (이중화 설정2), (이중화 설정3)..

```
#=====
# Replication Properties
#=====

REPLICATIONS = (
    REPLICATION_NAME_PREFIX = AKU_REP1
    SYNC_PARALLEL_COUNT      = 1
    (
        SYS.T1, SYS.T2, SYS.T3
    )
),
(
    REPLICATION_NAME_PREFIX = AKU_REP2
    SYNC_PARALLEL_COUNT      = 1
```

```

    (
        SYS.T4, SYS.T5, SYS.T6
    )
),
(
    REPLICATION_NAME_PREFIX = AKU_REP3
    SYNC_PARALLEL_COUNT     = 1
    (
        SYS.T7,
        SYS.T8,
        SYS.T9
    )
)
)

```

- 이중화 대상 테이블을 지정하는 방식에 "user\_name.table\_name"의 형식도 지원하도록 개선되었습니다.

- 변경 전(기존)

```

REPLICATIONS = (
    REPLICATION_NAME_PREFIX = "AKU_REP"
    SYNC_PARALLEL_COUNT     = 1
    (
        (
            USER_NAME      = "SYS"
            TABLE_NAME     = "T1"
        ),
        (
            USER_NAME      = "SYS"
            TABLE_NAME     = "T2"
        ),
        (
            USER_NAME      = "SYS"
            TABLE_NAME     = "T3"
        )
    )
)

```

- 변경 후 (기존의 설정방식도 지원하면서, 아래의 설정도 추가로 지원합니다.)

```

REPLICATIONS = (
    REPLICATION_NAME_PREFIX = AKU_REP1
    SYNC_PARALLEL_COUNT     = 1
    (
        SYS.T1, SYS.T2, SYS.T3
    )
)

```

- 그 밖의 개선된 기능은 다음과 같습니다.
  - ALTIBASE\_NLS\_USE 환경 변수를 이용하여 Altibase 서버의 문자집합과 동일한 문자집합을 사용 가능하도록 개선되었습니다.
  - aku.conf 파일 내 주석을 허용하도록 개선되었습니다.
  - AKU 실행 시, 단일 스레드에서 순차적으로 실행되던 작업들이 이제 멀티 스레드를 사용하여 병렬로 수행되므로 전체적인 수행시간이 감소되었습니다.

- AKU에서 이중화 생성할 때 REPLICATION\_MAX\_COUNT를 초과한 경우, 오류를 반환하도록 수정되었습니다.
- 쿼리 수행 실패 시 재시도는 1초 대기 후 1회만 재시도하였으나, AKU\_QUERY\_RETRY\_COUNT 및 AKU\_QUERY\_RETRY\_DELAY\_MSEC 프로퍼티를 이용하여 설정 가능하도록 변경되었습니다.
- 마스터 파드의 장애로 `aku -p start` 명령 수행이 실패한 경우에 대한 조치 방법이 매뉴얼에 추가되었습니다. 자세한 내용은 **Utilitis 매뉴얼 - 3.aku** 에서 [마스터 파드 장애로 aku -p start 명령 수행이 실패했을 때](#)를 참고하세요.
- 재현 방법
  - 재현 절차
  - 수행 결과
  - 예상 결과
- Workaround
- 변경사항
  - Performance view
    - Property
      - AKU\_QUERY\_RETRY\_COUNT 프로퍼티 추가
        - AKU 수행 중 쿼리 수행에 실패했을 때, 재시도 횟수
        - 기본값 : 5
        - 최소값 : 0
        - 최대값 :  $2^{32} - 1$
      - AKU\_QUERY\_RETRY\_DELAY\_MSEC 프로퍼티 추가
        - AKU 수행 중 쿼리 수행에 실패하여 재시도 할 때, 다음 재시도까지 대기 시간 ( msec 단위 )
        - 기본값 : 1000
        - 최소값 : 0
        - 최대값 :  $2^{32} - 1$
    - Compile Option
    - Error Code

## BUG-50878 aexport.properties 파일에서 ILOADER\_FIELD\_TERM, ILOADER\_ROW\_TERM 프로퍼티의 값을 변경합니다.

- **module** : ux-aexport
- **Category** : Usability
- **재현 빈도** : Always
- **설명** : aexport.properties 및 aexport.properties.sample 파일에서 ILOADER\_FIELD\_TERM, ILOADER\_ROW\_TERM 프로퍼티의 값을 기존보다 복잡한 형태로 변경합니다.  
ILOADER\_FIELD\_TERM 프로퍼티의 값은 `^`에서 `^C_C^`으로, ILOADER\_ROW\_TERM 프로퍼티의 값은 `%n`에서 `^R_r^%n`으로 변경되었습니다.
- **재현 방법**

- 재현 절차
- 수행 결과

```
$ cat aexport.properties | grep TERM
#ILOADER_FIELD_TERM = ^#ILOADER_ROW_TERM = %n
```

- 예상 결과

```
$ cat aexport.properties | grep TERM
#ILOADER_FIELD_TERM = ^C_c^#ILOADER_ROW_TERM = ^R_r^%n
```

- **Workaround**

- **변경사항**

- Performance view
- Property
- Compile Option
- Error Code

## BUG-50912 TCP 소켓 옵션 SO\_LINGER를 설정하는 연결 속성 socket\_immediate\_close를 추가합니다.

- **module** : mm-jdbc
- **Category** : Functionality
- **재현 빈도** : Always
- **설명** : TCP 소켓 옵션인 SO\_LINGER의 활성화 여부를 설정하는 연결 속성 socket\_immediate\_close를 추가합니다.
  - true: SO\_LINGER 값을 0으로 설정한다. 소켓을 닫는 즉시 연결이 종료되고 남아있는 데이터는 전송되지 않는다.
  - false: SO\_LINGER를 비활성화한다. 소켓은 바로 닫히지만, 소켓 버퍼에 남아있는 데이터가 있다면 커널은 일정 시간 동안 데이터를 보내려고 시도한다.

이 연결 속성이 추가된 JDBC드라이버를 사용하려면, Altibase JDBC 드라이버를 7.3.0.0.7 이상 버전으로 패치해야 합니다.

- **재현 방법**

- 재현 절차
- 수행 결과
- 예상 결과

- **Workaround**

- **변경사항**

- Performance view
- Property
- Compile Option
- Error Code

## BUG-50249 PICL 라이브러리에 예외상황 발생시 로그에 남기는 기능을 추가합니다. (AIX, HP-UX)

- **module** : ux-altiMon
- **Category** : Functionality
- **재현 빈도** : Rare
- **설명** : altiMon의 모니터링 항목 중 운영체제(OS)의 자원 상태를 수집하는 PICL 라이브러리에서 예외 상황이 발생하면, 이를 altimon.log에 기록하는 기능이 리눅스만 지원했었으나, AIX와 HP-UX환경에서도 지원하도록 개선되었습니다.
  - 예외 상황은 다음 정보를 조회하는 시스템 함수 호출이 실패할 때 발생합니다.
    - Altibase 프로세스 정보 조회 시
    - 파일 시스템 사용량 조회 시
    - 메모리 사용량 조회 시
    - SWAP 메모리 사용량 조회 시
    - 시스템 시간 조회 시
- **재현 방법**
  - 재현 절차
  - 수행 결과
  - 예상 결과
- **Workaround**
- **변경사항**
  - Performance view
  - Property
  - Compile Option
  - Error Code

## BUG-50005 aku 실행 시 옵션을 입력하지 않거나 잘못된 옵션을 입력했을 때, 오류 메시지를 출력하지 않고 help 메시지만 출력하도록 변경하였습니다.

- **module** :
- **Category** : Enhancement
- **재현 빈도** : Always
- **설명** : aku 실행 시 옵션을 입력하지 않거나 잘못된 옵션을 입력했을 때 오류 메시지를 출력하지 않고 help 메시지만 출력하도록 변경하였습니다.
- **재현 방법**
  - 재현 절차

```
% aku
```

- 수행 결과



Error while getting option.

```
=====
=====

      Altibase Kubernetes Utility Help Screen

=====
=====

Usage      : aku      [-h]

                        [--help]

                        [-v ]

                        [--version ]

                        [-i ]

                        [--info ]

                        [-p pod_action ]

                        [--pod pod_action]

-h, --help      : this screen

-v, --version   : version information

-i, --info      : option information

-p, --pod       : [start | stop | clean] specify pod_action

=====
=====
```

○ 예상 결과

```
=====
=====

      Altibase Kubernetes Utility Help Screen

=====
=====

Usage      : aku      [-h]

                        [--help]

                        [-v ]

                        [--version ]

                        [-i ]

                        [--info ]

                        [-p pod_action ]
```

```
 [--pod pod_action]
```

```
 -h, --help      : this screen
```

```
 -v, --version   : version information
```

```
 -i, --info      : option information
```

```
 -p, --pod       : [start | stop | clean] specify pod_action
```

```
=====
```

- **Workaround**

- **변경사항**

- Performance view
- Property
- Compile Option
- Error Code

# Fixed Bugs

**BUG-50182 테이블의 행(Row)을 삭제하는 도중에 ALTER TABLE ... ADD COLUMN... DDL이 실행되면, 잘못된 메모리에 접근하는 오류가 발생할 수 있습니다.**

- **module** : sm
- **Category** : Memory Error
- **재현 빈도** : Always
- **설명** : 테이블의 행(Row)을 삭제하는 도중에 ALTER TABLE ... ADD COLUMN ... 의 DDL이 실행되는 경우, 잘못된 메모리에 접근하는 오류가 발생하지 않도록 개선되었습니다.
- **재현 방법**
  - 재현 절차
  - 수행 결과
  - 예상 결과
- **Workaround**
- **변경사항**
  - Performance view
  - Property
  - Compile Option
  - Error Code

**BUG-50810 API를 통한 LOB 데이터 작업시, "Internal server error"가 발생할 수 있습니다.**

- **module** : sm
- **Category** : Fatal
- **재현 빈도** : Always
- **설명** : API를 통해 LOB 데이터를 조회하거나 쓸 때 내부적으로 쓰기(write) 작업이 실패하는 경우, 트랜잭션이 롤백되지 않아 잘못된 페이지에 접근하는 오류가 발생할 수 있습니다.  
  
이제 LOB 데이터의 쓰기 작업이 실패할 경우, "Failed to complete LOB write operation." 에러메시지가 출력되도록 개선되었습니다. 이 메시지가 나타나면, 사용자는 명시적으로 롤백(Rollback)을 수행하여 잘못된 페이지에 접근하는 오류를 방지할 수 있습니다.
- **재현 방법**
  - 재현 절차
  - 수행 결과
  - 예상 결과
- **Workaround**
- **변경사항**
  - Performance view
  - Property

- Compile Option
- Error Code
  - 에러 메시지 추가
    - **[ERR-111C2 : Failed to complete LOB write operation.(<0%>)]**
    - **Cause:** The previous LOB write operation terminated incompletely.
    - **Action:** Execute ROLLBACK and retry the statement.

## BUG-50849 escape syntax에 대문자가 포함된 경우 정상적으로 파싱처리가 되지 않습니다.

- **module** : mm-cli
- **Category** : Functional Error
- **재현 빈도** : Always
- **설명** : SQLCLI에서 ODBC 이스케이프 시퀀스(escape sequence)에 대문자를 사용하면 "parse error" 에러가 발생하는 현상을 수정하였습니다. 이제 ODBC 이스케이프 시퀀스 대소문자를 구분하지 않습니다. 이전 버전에서는 대문자를 사용하면 "parse error" 에러가 발생하였으나, 이 버그가 반영된 버전부터는 에러가 발생하지 않습니다.

ODBC 이스케이프 시퀀스 항목은 [Microsoft ODBC](#)

[페이지](#)를

참고해 주십시오.

이 버그를 반영하려면 Altibase CLI 드라이버(libodbcli.a)를 패치해야 합니다.

- **재현 방법**
  - **재현 절차**

```
CREATE TABLE T1(i1 int, i2 int);
CREATE OR REPLACE PROCEDURE proc1
AS
  v1 constant INTEGER := 1;
  v2 constant t1.i1%TYPE := 1;
BEGIN
  INSERT INTO t1 VALUES (v1, v2);
END;

rc = SQLExecDirect(stmt, (SQLCHAR *)"{ CALL proc1() }", SQL_NTS);
if (!SQL_SUCCEEDED(rc))
{
  PRINT_DIAGNOSTIC(SQL_HANDLE_STMT, stmt, "SQLExecDirect");
  goto EXIT_STMT;
}
printf("stored procedure call success\n");
```

- **수행 결과**

```

Error : 158 : SQLExecDirect
Diagnostic Record 1
SQLSTATE      : 42000
Message text : SQL syntax error
line 1: parse error
{ CALL proc1() }
^

```

- 예상 결과

```
프로시저 호출 성공
```

- Workaround

- 변경사항

- Performance view
- Property
- Compile Option
- Error Code

**BUG-50862 송신자로부터 이중화 시작 요청을 수신하는 동시에 수신자가 이중화를 삭제하는 경우, 동시성 문제가 발생할 수 있습니다.**

- module : rp-receiver

- Category : Fatal

- 재현 빈도 : Rare

- 설명 : 송신자로부터 이중화 시작 요청을 수신하는 동시에 수신자가 이중화를 삭제하는 경우 발생할 수 있는 동시성 문제를 개선하였습니다.

- 재현 방법

- 재현 절차
- 수행 결과
- 예상 결과

- Workaround

- 변경사항

- Performance view
- Property
- Compile Option
- Error Code

**BUG-50864 로그 파일의 크기가 비정상적으로 변경된 경우, 잘못된 에러 메시지가 출력됩니다.**

- module : sm

- Category : Message Error

- 재현 빈도 : Always

- **설명** : 로그 파일의 크기가 비정상적으로 변경된 경우, 잘못된 에러메시지가 출력되는 문제를 수정합니다.
  - 변경전: [ERR-111AC : OS return Log file size is zero. ( 로그파일 경로 및 이름 ).]
  - 변경후: [ERR-111C1 : The log file size has changed. : (로그파일 경로 및 이름) (current file size: , expected file size: )]
- **재현 방법**
  - 재현 절차
  - 수행 결과
  - 예상 결과
- **Workaround**
- **변경사항**
  - Performance view
  - Property
  - Compile Option
  - Error Code
    - 에러 메시지 추가
      - [ERR-111C1 : The log file size has changed. : <0%s> (current file size: <1%u>, expected file size:<2%u>)]
      - **Cause**: The log file size has changed abnormally, so that the file is invalid.
      - **Action**: Check if the backed-up file for the log file exists and restore it. If it's not available, contact Altibase's Support Center (<http://support.altibase.com>).

**BUG-50866 온라인 백업된 파일을 이용해서 복구를 시도할 때 MustRedo LSN까지의 로그파일이 없는 경우, 복구가 실패했음에도 불구하고 "Database media recovery successful"메시지를 출력합니다.**

- **module** : sm
- **Category** : Fatal
- **재현 빈도** : Always
- **설명** : 온라인 백업된 파일을 이용해서 복구를 시도할 때 MustRedo LSN까지의 로그파일이 없는 경우, [ERR-91015 : Communication failure.] 에러를 출력하도록 수정하였습니다. 이 경우, altibase\_boot.log 에 "Recovery failure, need more logfile ..."의 로그를 확인할 수 있습니다.
- **재현 방법**
  - 재현 절차
  - 수행 결과
  - 예상 결과
- **Workaround**
- **변경사항**
  - Performance view
  - Property

- Compile Option
- Error Code

## BUG-50875 Logical Ager가 삭제해야 할 인덱스 키를 찾지 못할 때 Altibase 서버가 비정상 종료합니다.

- **module** : sm-mem-index
- **Category** : Fatal
- **재현 빈도** : Unknown
- **설명**
  - Logical Ager가 삭제해야 할 인덱스 키를 찾지 못할 때 Altibase 서버가 비정상 종료하는 현상을 수정합니다. Logical Ager가 알 수 없는 이유로 더 이상 참조되지 않는 인덱스 키를 찾는데 실패하면 해당 인덱스의 상태를 ISCONSISTENT로 변경합니다. ISCONSISTENT 상태의 인덱스는 V\$MEM\_BTREE\_HEADER에서 조회할 수 있습니다.
  - ISCONSISTENT 상태의 인덱스가 있는 메모리 테이블에 INSERT 문이 정상적으로 수행되는 현상을 수정합니다. 이 버그가 반영된 버전에서는 ISCONSISTENT 상태의 인덱스가 있는 메모리 테이블에 INSERT 문을 수행하면 [ERR-11110 : The index is inconsistent] 에러가 발생합니다.
- **재현 방법**
  - 재현 절차
  - 수행 결과
  - 예상 결과
- **Workaround**
- **변경사항**
  - Performance view
  - Property
  - Compile Option
  - Error Code

## BUG-50881 IPCDA 접속 중에 클라이언트가 종료하면 서버가 비정상 종료합니다.

- **module** : cm
- **Category** : Fatal
- **재현 빈도** : Always
- **설명** : IPCDA 클라이언트 접속 중 (Handshake 프로토콜 처리 이전)에 클라이언트가 종료되면 서버가 비정상 종료하는 문제를 수정하였습니다.
- **재현 방법**
  - 재현 절차
  - 수행 결과
  - 예상 결과
- **Workaround**
- **변경사항**

- Performance view
- Property
- Compile Option
- Error Code

## BUG-50891 사용하지 않는 INSPECTION\_LARGE\_HEAP\_THRESHOLD 프로퍼티를 삭제하였습니다.

- **module** : sm
- **Category** : Maintainability
- **재현 빈도** : Always
- **설명** : 사용하지 않는 `INSPECTION_LARGE_HEAP_THRESHOLD` 프로퍼티를 삭제하였습니다. 이 버그가 반영된 버전에서는 V\$PROPERTY에서 `INSPECTION_LARGE_HEAP_THRESHOLD` 프로퍼티를 조회할 수 없습니다.
- **재현 방법**
  - 재현 절차
  - 수행 결과
  - 예상 결과
- **Workaround**
- **변경사항**
  - Performance view
  - Property
  - Compile Option
  - Error Code

## BUG-50905 ODBC 연결 스트링에 DSN 속성을 입력하지 않으면 Invalid attribute value. 에러가 발생합니다.

- **module** : ul-odbc
- **Category** : Functional Error
- **재현 빈도** : Always
- **설명** : ODBC 연결 스트링에 DSN 속성을 입력하지 않으면 "Invalid attribute value." 에러가 발생하는 현상을 수정합니다. 이 현상은 'odbc.ini' 파일에 ODBC 데이터 소스 정보가 포함되어 있고, 연결 스트링에 DSN 속성을 입력하지 않을 때 발생합니다. DSN 속성을 사용하지 않아도 연결이 성공하도록 수정하였습니다.

이 버그를 적용하려면 Altibase ODBC 드라이버를 패치해야 합니다.

- **재현 방법**
  - 재현 절차

odbc.ini 파일 예시



```
[ODBC Data Sources]
BUG_50905_ALTIODBC_UL32 = Altibase ODBC 7.3 Driver

[BUG_50905_ALTIODBC_UL32]
Driver      =
/data/eheejung/work/altidev4_r96980/altibase_home/lib/libaltibase_odbc-
64bit-ul32.so
Description = Altibase ODBC
SERVER      = 127.0.0.1
PORT        = 20300
Database    = mydb

[ODBC]
Trace = 1
TraceFile = /tmp/odbc.log

연결 문자열 예시
(SQLCHAR *)"Driver=/usr/altibase/altibase_home/lib/libaltibase_odbc-
64bit-ul32.so;Server=127.0.0.1;User=SYS;Password=MANAGER;Port=20300"

(SQLCHAR
*)"Driver=BUG_50905_ALTIODBC_UL32;Server=127.0.0.1;User=SYS;Password=MAN
AGER;Port=20300"
```

#### ○ 수행 결과

```
Error : 49 : SQLDriverConnect
Diagnostic Record 1
SQLSTATE : HY024
Message text : Invalid attribute value.
Message len : 24
Native error : 0x51015
```

#### ○ 예상 결과

```
연결 성공
```

#### • Workaround

##### • 변경사항

- Performance view
- Property
- Compile Option
- Error Code

**BUG-50907 조인에서 하이브리드 파티션드 테이블이 사용되고 SERIAL\_EXECUTE\_MODE 프로퍼티가 1일 때, 결과 값에 오류가 발생할 수 있습니다.**

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

- **설명** : 조인에서 하이브리드 파티션드 테이블이 사용되고 `SERIAL_EXECUTE_MODE` 프로퍼티가 1일 때, 결과 값에 오류가 발생하는 현상을 수정하였습니다. 이 버그는 조인 방식이 FULL\_NL일 때 발생하며, `USE_HASH` 힌트를 사용하여 이 문제를 회피할 수 있습니다.
- **재현 방법**
  - **재현 절차**

```
CREATE VOLATILE DATA TABLESPACE VOL_TBS SIZE 32M AUTOEXTEND ON;

CREATE TABLE t1( i1 int, c1 clob )
PARTITION BY RANGE(i1)
(
    PARTITION p1 VALUES LESS THAN (1) TABLESPACE SYS_TBS_MEM_DATA,
    PARTITION p2 VALUES LESS THAN (2) TABLESPACE SYS_TBS_DISK_DATA,
    PARTITION p3 VALUES LESS THAN (3) TABLESPACE VOL_TBS,
    PARTITION pd VALUES DEFAULT TABLESPACE SYS_TBS_DISK_DATA
) TABLESPACE SYS_TBS_DISK_DATA UNCOMPRESSED LOGGING;

CREATE TABLE t2( i1 int, c1 clob )
PARTITION BY RANGE(i1)
(
    PARTITION p1 VALUES LESS THAN (2) TABLESPACE SYS_TBS_DISK_DATA,
    PARTITION p2 VALUES LESS THAN (4) TABLESPACE SYS_TBS_MEM_DATA,
    PARTITION p3 VALUES LESS THAN (6) TABLESPACE SYS_TBS_DISK_DATA,
    PARTITION pd VALUES DEFAULT TABLESPACE VOL_TBS
) TABLESPACE SYS_TBS_DISK_DATA UNCOMPRESSED LOGGING;

INSERT INTO t1 VALUES( 0, '0' );
INSERT INTO t1 VALUES( 1, '1' );
INSERT INTO t1 VALUES( 2, '2' );
INSERT INTO t1 VALUES( 3, '3' );
INSERT INTO t1 VALUES( 4, '4' );
INSERT INTO t2 SELECT * FROM t1;

CREATE VIEW v1 AS SELECT t1.i1 i1 FROM t1, t2 WHERE t1.i1 = t2.i1;

SELECT i1 FROM v1;
```

- **수행 결과**

```
SELECT i1 FROM v1;
i1
-----
0
1
2
4
4 rows selected.
```

- **예상 결과**

```

I1
-----
0
1
2
3
4
5 rows selected.

```

- **Workaround**

```
SELECT /*+ USE_HASH( t1, t2 ) */ i1 FROM v1;
```

- **변경사항**

- Performance view
- Property
- Compile Option
- Error Code

## BUG-50914 altibase\_store\_result, altibase\_next\_result를 순차적으로 실행할 때, 두 번째 결과 집합에서 HY010 Function sequence error. 에러가 발생합니다.

- **module** : ux-cdbc
- **Category** : Functional Error
- **재현 빈도** : Always
- **설명** : altibase\_store\_result, altibase\_next\_result를 순차적으로 실행하여 결과 집합을 가져올 때, 두 번째 결과 집합에서 발생하는 "HY010 Function sequence error." 에러를 수정하였습니다.  
이 버그를 반영하려면 ACI 라이브러리(libalticapi.a)를 패치해야 합니다.
- **재현 방법**
  - **재현 절차**

```

-- 스키마
CREATE TABLE RESULT1 (I1 INTEGER, I2 VARCHAR(10));
INSERT INTO RESULT1 VALUES (11, '11_AAA');
INSERT INTO RESULT1 VALUES (12, '12_BBB');
INSERT INTO RESULT1 VALUES (13, '13_CCC');
CREATE TABLE RESULT2 (I1 INTEGER, I2 VARCHAR(10), I3 VARCHAR(10));
INSERT INTO RESULT2 VALUES (21, '21_AAA', '221_AAA');
INSERT INTO RESULT2 VALUES (22, '22_BBB', '222_BBB');
INSERT INTO RESULT2 VALUES (23, '23_CCC', '223_CCC');
CREATE TABLE RESULT3 (I1 INTEGER, I2 VARCHAR(10));
INSERT INTO RESULT3 VALUES (31, '31_AAA');
INSERT INTO RESULT3 VALUES (32, '32_BBB');
INSERT INTO RESULT3 VALUES (33, '33_CCC');
CREATE TABLE RESULT4 (I1 INTEGER, I2 VARCHAR(10));
INSERT INTO RESULT4 VALUES (41, NULL);
INSERT INTO RESULT4 VALUES (NULL, '42_BBB');
INSERT INTO RESULT4 VALUES (NULL, NULL);

```

```

CREATE TYPESET MY_TYPE
AS
    TYPE MY_CUR IS REF CURSOR;
END;
/

CREATE OR REPLACE PROCEDURE PROC_RESULTSET
(
    P1 OUT MY_TYPE.MY_CUR,
    P2 OUT MY_TYPE.MY_CUR,
    P3 OUT MY_TYPE.MY_CUR,
    P4 OUT MY_TYPE.MY_CUR
)
AS
    SSQL_STMT VARCHAR(200);
BEGIN
    SSQL_STMT := 'SELECT * FROM RESULT1';
    OPEN P1 FOR SSQL_STMT;
    SSQL_STMT := 'SELECT * FROM RESULT2';
    OPEN P2 FOR SSQL_STMT;
    SSQL_STMT := 'SELECT * FROM RESULT3';
    OPEN P3 FOR SSQL_STMT;
    SSQL_STMT := 'SELECT * FROM RESULT4';
    OPEN P4 FOR SSQL_STMT;
END;
/

-- ACI 코드 예시
SRC = altibase_query(sAB, "EXEC PROC_RESULTSET");
while (1)
{
    sRes = altibase_store_result(sAB);
    ...중략...
    while ((sRow = altibase_fetch_row(sRes)) != NULL)
    {
        ...중략...
    }
    ...중략...
    SRC = altibase_next_result(sAB);
    ...중략...
}

```

#### 수행 결과

HY010 Function sequence error.

#### 예상 결과

결과 집합을 정상적으로 조회

#### • Workaround

##### • 변경사항

- Performance view
- Property
- Compile Option

- Error Code

**BUG-50919 altibase\_stmt\_next\_result 함수를 실행할 때, 두 번째 결과 집합에서 "HY010 Function sequence error." 에러가 발생합니다.**

- **module** : ux-cdbc
- **Category** : Functional Error
- **재현 빈도** : Always
- **설명** : altibase\_stmt\_next\_result() 함수로 두 번째 결과 집합에 접근할 때 발생하는 "HY010 Function sequence error." 에러를 수정하였습니다.

이 버그를 반영하려면 ACI 라이브러리(libalticapi.a)를 패치해야 합니다.

- **재현 방법**
  - 재현 절차

```
-- 스키마
CREATE TABLE fetch1 (id INTEGER, val VARCHAR(10));
INSERT INTO fetch1 VALUES (1, 'a1');
INSERT INTO fetch1 VALUES (2, 'b2');
CREATE TABLE fetch2 (id INTEGER, val VARCHAR(10), etc VARCHAR(10));
INSERT INTO fetch2 VALUES (4, 'd4', 'asd');
INSERT INTO fetch2 VALUES (5, 'e5', 'qwe');
INSERT INTO fetch2 VALUES (6, 'f6', 'zxc');
CREATE TYPESET fetch_type
AS
    TYPE fetch_cur IS REF CURSOR;
END;
/
CREATE OR REPLACE PROCEDURE fetch_proc
(
    p1 OUT fetch_type.fetch_cur,
    p2 OUT fetch_type.fetch_cur
)
AS
BEGIN
    OPEN p1 FOR 'SELECT * FROM fetch1';
    OPEN p2 FOR 'SELECT * FROM fetch2';
END;
/
-- ACI 코드 예시
SRC = altibase_stmt_prepare(sStmt, "EXEC fetch_proc");
...중략...
SRC = altibase_stmt_execute(sStmt);
...중략...
while (1)
{
    ...중략...
    while ((SRC = altibase_stmt_fetch(sStmt)) != ALTIBASE_NO_DATA)
    {
        ...중략...
    }
}
```

```
...중략...
SRC = altibase_stmt_next_result(sStmt);
...중략...
}
```

- 수행 결과

```
HY010 Function sequence error.
```

- 예상 결과

```
결과 집합을 정상적으로 조회
```

- Workaround

- 변경사항

- Performance view
  - N/A
- Property
- Compile Option
- Error Code

## BUG-50920 데이터 업로드 시 에러가 발생한 경우, 어느 칼럼에서 발생했는지 알 수 없습니다.

- module** : ux-iload
- Category** : Usability
- 재현 빈도** : Always
- 설명** : 데이터 업로드 시 에러가 발생한 경우, 어느 칼럼에서 발생했는지 알 수 없는 문제를 수정하였습니다.

이제 `-verbose` 옵션을 사용하여 데이터 업로드에 실패한 칼럼의 정보를 로그 파일에서 확인할 수 있습니다. `-verbose` 옵션은 `-log` 옵션과 같이 사용해야 합니다. 데이터 업로드 시 `-verbose` 옵션을 사용하면 에러 메시지와 함께 테이블에서 해당 칼럼의 위치를 `-log logfile` 파일에 기록합니다. 단, 반환된 에러에서 칼럼의 정보를 알 수 없으면 칼럼의 위치가 기록되지 않습니다.

- 재현 방법

- 재현 절차

```
-- 테이블 스키마
CREATE TABLE BUG_50920 (c1 INT, c2 int);

-- 테이블에 업로드 할 파일 내용
cat BUG_50920.dat
1,1
2,2147483648
2147483648, 3
2147483648, 2147483648

-- 데이터 업로드 수행
iload -s localhost -u SYS -p manager formout -f BUG_50920.fmt -T
BUG_50920
```

```

iloader -s localhost -u SYS -p manager in -f BUG_50920.fmt -d
BUG_50920.dat -log BUG_50920.log -silent
cat BUG_50920.log
iloader -s localhost -u SYS -p manager in -f BUG_50920.fmt -d
BUG_50920.dat -log BUG_50920.log -silent -verbose
cat BUG_50920.log

```

#### ○ 수행 결과

```

$ iloader -s localhost -u SYS -p manager in -f BUG_50920.fmt -d
BUG_50920.dat -log BUG_50920.log -silent
UPLOAD : 14.3730 msec
      Load Count   : 1(BUG_50920)
      Error Count   : 3

$ cat BUG_50920.log
TableName : BUG_50920
Start Time : Wed Jun 19 19:19:33 2024
Record 2 : 2,2147483648
[ERR-51072 : Numeric value out of range.]
Record 3 : 2147483648,3
[ERR-51072 : Numeric value out of range.]
Record 4 : 2147483648,2147483648
[ERR-51072 : Numeric value out of range.]
End Time : Wed Jun 19 19:19:33 2024
Total Row Count : 4
Load Row Count   : 1
Error Row Count   : 3

$ iloader -s localhost -u SYS -p manager in -f BUG_50920.fmt -d
BUG_50920.dat -log BUG_50920.log -silent -verbose
[ERR-9103B : Invalid option (-verbose)]

```

#### ○ 예상 결과

```

$ iloader -s localhost -u SYS -p manager in -f BUG_50920.fmt -d
BUG_50920.dat -log BUG_50920.log -silent
UPLOAD : 15.8760 msec
      Load Count   : 1(BUG_50920)
      Error Count   : 3

$ cat BUG_50920.log
TableName : BUG_50920
Start Time : Wed Jun 19 19:20:50 2024
Record 2 : 2,2147483648
[ERR-51072 : Numeric value out of range.]
Record 3 : 2147483648,3
[ERR-51072 : Numeric value out of range.]
Record 4 : 2147483648,2147483648
[ERR-51072 : Numeric value out of range.]
End Time : Wed Jun 19 19:20:50 2024
Total Row Count : 4
Load Row Count   : 1
Error Row Count   : 3

```

```

$ iloader -s localhost -u SYS -p manager in -f BUG_50920.fmt -d
BUG_50920.dat -log BUG_50920.log -silent -verbose
UPLOAD : 15.4300 msec
      Load Count   : 1(BUG_50920)
      Error Count   : 3

$ cat BUG_50920.log
TableName : BUG_50920
Start Time : Wed Jun 19 19:21:20 2024
Record 2 : 2,2147483648
[ERR-51072 : Numeric value out of range. at Column [2]]
Record 3 : 2147483648,3
[ERR-51072 : Numeric value out of range. at Column [1]]
Record 4 : 2147483648,2147483648
[ERR-51072 : Numeric value out of range. at Column [1]]
End Time : Wed Jun 19 19:21:20 2024
Total Row Count : 4
Load Row Count   : 1
Error Row Count   : 3

```

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

## BUG-50923 SQLPrepare 함수의 SQL 텍스트 문자열 인자가 빈 문자열일 때 에러가 발생하지 않습니다.

- **module** : mm-cli
- **Category** : Functional Error
- **재현 빈도** : Always
- **설명** : SQLPrepare 함수의 SQL 텍스트 문자열 인자가 빈 문자열일 때 에러가 발생하지 않는 문제를 수정했습니다.

이 버그를 수정하기 전에는 SQLExecute 함수에서 "Failure to find statement" 에러가 발생했으나 이 버그가 수정된 버전에서는 SQLPrepare 함수에서 "SQL statement too short" 에러가 발생합니다.

이 버그를 반영하려면 Altibase CLI 드라이버(libodbcli.a)를 패치해야 합니다.

- **재현 방법**
  - **재현 절차**



```
rc = SQLPrepare(stmt, (SQLCHAR *)"", SQL_NTS);
if (!SQL_SUCCEEDED(rc))
{
    PRINT_DIAGNOSTIC(SQL_HANDLE_STMT, stmt, "SQLPrepare");
    goto EXIT_STMT;
}
rc = SQLExecute(stmt);
if (!SQL_SUCCEEDED(rc))
{
    PRINT_DIAGNOSTIC(SQL_HANDLE_STMT, stmt, "SQLExecute");
    goto EXIT_STMT;
}
```

#### ○ 수행 결과

SQLPrepare에서 에러가 발생하지 않고 SQLExecute에서 에러가 발생함.

```
Error : 236 : SQLExecute
Diagnostic Record 1
    SQLSTATE      : HY000
    Message text   : Failure to find statement
    Message len    : 25
    Native error   : 0x41098
```

#### ○ 예상 결과

SQLPrepare에서 에러가 발생함.

```
Error : 160 : SQLPrepare
Diagnostic Record 1
    SQLSTATE      : HY000
    Message text   : SQL statement too short
    Message len    : 23
    Native error   : 0x41039
```

#### • Workaround

##### • 변경사항

- Performance view
- Property
- Compile Option
- Error Code

## BUG-50938 AKU 에서 이용하는 문자열 값의 길이 제한이 실제와 다르게 설정되어 있습니다.

- **module** : aku
- **Category** : Functional Error
- **재현 빈도** : Always
- **설명** : aku 유틸리티에서 사용하는 문자열 길이의 제한을 올바르게 수정합니다.

`AKU_STS_NAME`, `AKU_SVC_NAME` 프로퍼티의 최대 길이를 쿠버네티스의 제한에 따라 40에서 63으로 변경합니다. 이중화 대상 테이블의 사용자 이름과 이중화 대상 테이블의 이름의 길이를 40에서 128로 변경합니다. 이 제한은 Altibase 객체의 최대 길이를 따릅니다.

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

## BUG-50948 서버에 fetch할 데이터가 남아 있는 경우 ResultSet.close()를 해도 커서가 바로 닫히지 않을 수 있습니다.

- **module** : mm-jdbc
- **Category** : Functional Error
- **재현 빈도** : Always
- **설명** : ResultSet 객체에서 조회할 데이터가 Altibase 서버에 남아 있을 때, ResultSet.close()를 호출해도 커서가 닫히지 않는 문제를 수정하였습니다. 이 문제가 발생하면 Altibase 서버 설정에 의해 FETCH\_TIMEOUT 오류가 발생할 수 있습니다.

이 버그가 수정된 버전을 반영하려면 Altibase JDBC 드라이버를 7.3.0.0.7 이상 버전으로 패치해야 합니다.

- 재현 방법
  - 재현 절차

```
isQL> create table t1 (c1 int);
isQL> insert into t1 select level from dual connect by level <= 1000;
[altibase.properties]
FETCH_TIMEOUT = 5
sStmt.setFetchSize(1);
ResultSet sRs = sStmt.executeQuery("SELECT * FROM t1");
if (sRs.next())
{
    System.out.println("col1 ==>" + sRs.getInt(1));
}
sRs.close();
Thread.sleep(7000);
sRs = sStmt.executeQuery("SELECT 1 FROM dual");
if (sRs.next())
{
    System.out.println(sRs.getString(1));
}
```

- 수행 결과

```
[client]
Exception in thread "main" java.sql.SQLException: Communication link
failure: There was no response from the server, and the channel has
reached end-of-stream.
[server]
[2024/06/03 09:51:01 4EF][PID:26603][Thread-139699788171328][LWP-26592]
[Notify : Fetch Timeout] Session Closed by Server : Session ID = 2
CLIENT_INFO          => TCP 192.168.1.48:54096(PID : 2065530879)
Time Limit            => 5
Running Time          => 6
Transaction ID        => 5152
Caused by Query       => SELECT * FROM t1
```

- 예상 결과

```
1
```

- Workaround

- 변경사항

- Performance view
- Property
- Compile Option
- Error Code

**BUG-50949 aku 설정 파일에 주석 입력시, # 이후에 글자를 입력하지 않는 경우 오류가 발생합니다.**

- **module** : aku
- **Category** : Enhancement
- **재현 빈도** : Always
- **설명** : aku 설정 파일(aku.conf)에 주석 입력시, # 이후에 글자를 입력하지 않을 경우 오류가 발생하는 문제를 수정하였습니다.
- **재현 방법**
  - 재현 절차
  - 수행 결과
  - 예상 결과
- **Workaround**
- **변경사항**
  - Performance view
  - Property
  - Compile Option
  - Error Code

## BUG-50950 aku -p start를 실행할 때 첫 번째 이중화 객체 생성이 실패해도 AKU가 즉시 종료되지 않습니다.

- **module** : aku
- **Category** : Functional Error
- **재현 빈도** : Always
- **설명** : `aku -p start` 수행 중 이중화 객체 생성 단계에서 첫 번째 이중화 객체 생성 중 실패가 발생하면, 바로 종료되지 않고 두 번째 이중화 객체 생성을 시도한 뒤 종료되는 버그를 수정하였습니다. 이중화 객체 생성 중 실패하면 다른 작업을 수행하지 않고 바로 종료되도록 변경하였습니다.
- **재현 방법**
  - 재현 절차
  - 수행 결과
  - 예상 결과
- **Workaround**
- **변경사항**
  - Performance view
  - Property
  - Compile Option
  - Error Code

## BUG-50969 RESET\_PARAMS가 내부적으로 수행될 때 bakBindParam도 제거되어야 합니다

- **module** : ux-cdbc
- **Category** : Functional Error
- **재현 빈도** : Always
- **설명** : `altibase_stmt_bind_param` 함수의 두 번째 인자의 메모리 주소가 변경되면 `Some parameters were not bound.` 에러가 발생하거나 클라이언트가 비정상 종료할 수 있습니다.  
  
이 버그는 `altibase_stmt_bind_param` 함수와 `altibase_stmt_execute` 함수를 반복적으로 수행할 때 발생합니다. `altibase_stmt_bind_param` 함수의 두 번째 인자의 메모리 주소가 변경되면 `Some parameters were not bound.` 에러가 발생하거나 `segment fault`로 클라이언트가 비정상 종료할 수 있습니다.  
  
이 버그를 반영하려면 ACI 라이브러리(`libalticapi.a`)를 패치해야 합니다.
- **재현 방법**
  - 재현 절차

```
src = altibase_stmt_prepare(sStmt, "INSERT INTO bug50969 VALUES(?, ?)");
...중략...
memset(sBind1, 0, sizeof(sBind1));
...중략....
src = altibase_stmt_bind_param(sStmt, sBind1);
```

```

...중략....

sCount++;
sBind1Count = 11;
if (altibase_stmt_execute(sStmt) != ALTIBASE_SUCCESS)
{
    PRINT_STMT_ERROR(sStmt);
    goto end_stmt;
}

altibase_stmt_free_result(sStmt);

memset(sBind2, 0, sizeof(sBind2));
...중략....
SRC = altibase_stmt_bind_param(sStmt, sBind2);
...중략....

sCount++;
sBind2Count = 12;
if (altibase_stmt_execute(sStmt) != ALTIBASE_SUCCESS)
{
    PRINT_STMT_ERROR(sStmt);
    goto end_stmt;
}

```

- 수행 결과

```
[51051] HY000 Some parameters were not bound.
```

- 예상 결과

```

SELECT * FROM bug50969;
C1 C2
-----
11 1
12 2
2 rows selected.

```

- Workaround

- 변경사항

- Performance view
- Property
- Compile Option
- Error Code

## BUG-50975 SIMPLE QUERY 최적화 기능을 활성화하고 JDBC 연결 속성에 remove\_redundant\_transmission을 사용할 때, SQL 문 수행 중 메모리 오류가 발생할 수 있습니다.

- **module** : mm
  - **Category** : Memory Error
  - **재현 빈도** : Always
  - **설명** : "SIMPLE QUERY 최적화" 기능을 활성화하고 JDBC 연결 속성에 remove\_redundant\_transmission을 사용할 때 SQL 문 수행 중 발생하는 메모리 오류를 수정하였습니다.
- 이 버그는 아래 조건을 만족할 때 Altibase 서버가 비정상 종료할 수 있습니다.
- Altibase 서버 프로퍼티 EXECUTOR\_FAST\_SIMPLE\_QUERY = 1 설정
  - JDBC 연결 속성 remove\_redundant\_transmission을 사용
  - SQL 문이 SIMPLE QUERY 최적화 기능이 적용되어 실행
- **재현 방법**
    - **재현 절차**

[Altibase 서버]

```
ALTER SYSTEM SET EXECUTOR_FAST_SIMPLE_QUERY = 1;
```

```
CREATE TABLE BUG
(
  C1 INTEGER NOT NULL,
  C2 INTEGER NOT NULL,
  C3 DECIMAL(12,2),
  C4 DECIMAL(4,4),
  C5 INTEGER,
  C6 VARCHAR(10),
  C7 VARCHAR(20),
  C8 VARCHAR(20),
  C9 VARCHAR(20),
  C10 CHAR(2),
  C11 CHAR(9)
);
```

[Altibase 클라이언트]

```
String sURL = "jdbc:Altibase://127.0.0.1:" + sPort + "/mydb" +
"?remove_redundant_transmission=1";
```

...중략...

```
/* Initialize environment */
try
{
  sCon = DriverManager.getConnection( sURL, sProps );

  sStmt = sCon.createStatement();
```

```

sRS = sStmt.executeQuery( "SELECT C6, C7, C8, C9, C10, C11 FROM
BUG50975");

sRowCount = 0;
/* Fetch all data */
while( sRS.next() )
{
    sRowCount++;
}
System.out.println(sRowCount + " rows selected");

/* Finalize process */
sStmt.close();
sCon.close();

}

```

- 수행 결과

Altibase 서버 비정상 종료

- 예상 결과

SQL 문 정상 수행

- Workaround

- 변경사항

- Performance view
- Property
- Compile Option
- Error Code

**BUG-50997 외부 쿼리를 참조하는 서브 쿼리에서 집계 함수를 사용하는 경우, 특정 조건에서 결과값 오류가 발생합니다.**

- module** : qp
- Category** : Functional Error
- 재현 빈도** : Always
- 설명** : 외부 쿼리를 참조하는 서브 쿼리에서 집계 함수를 사용하는 경우, 특정 조건에서 발생하는 결과값 오류를 수정하였습니다. 이 버그는 다음의 조건을 모두 만족할 때만 발생합니다.
  - AND 연산자의 조건에서 OR 연산자가 포함된 경우.
  - OR 연산자의 조건에 외부 쿼리를 참조하는 서브 쿼리가 존재합니다.
  - 2.의 서브 쿼리가 집계 함수를 사용합니다.
- 재현 방법**
  - 재현 절차

```
drop table T1;
```

```

drop table T2;

create table T1
(
  REV integer,
  REVTYPE smallint,
  id integer
);

insert into T1 values(7,          7,          0);
insert into T1 values(6,          5,          2);
insert into T1 values(5,          6,          1);

create table T2
(
  REV integer ,
  REVTYPE smallint,
  id integer
);

insert into T2 values(15,          3,          2);
insert into T2 values(17,          1,          0);
insert into T2 values(10,          3,          1);

select A.id
from T1 A, T2 B
where
  A.id = B.id
  and
  (
    (B.REV = (select min(BB.REV) from T2 BB where BB.REV < 23 and B.id =
BB.id )
    and
    A.REV = (select (AA.REV) from T1 AA where AA.REV < 23 and A.id =
AA.id)
    )
  or (
    A.REVTYPE = 2
    and B.REVTYPE <= 2
    )
  );

```

○ 수행 결과

```

ID
-----
2
1
2 rows selected.

```

○ 예상 결과



```
ID
-----
0
2
1
3 rows selected.
```

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

**BUG-51004 altibase\_stmt\_bind\_param에 바인딩하는 변수의 타입 또는 포인터가 바뀌는 경우, "Function sequence error"가 발생할 수 있습니다.**

- **module** : ux-cdbc
- **Category** : Functional Error
- **재현 빈도** : Always
- **설명** : altibase\_stmt\_bind\_param에 바인딩하는 변수의 타입 또는 포인터가 바뀌는 경우, "Function sequence error"가 발생하지 않도록 수정했습니다.
- **재현 방법**
  - 재현 절차

```
altibase_stmt_prepare (select ... )
altibase_stmt_bind_param(sStmt, sBind1)
altibase_stmt_execute
altibase_stmt_free_result
altibase_stmt_bind_param(sStmt, sBind2)
altibase_stmt_execute
```

- 수행 결과

```
altibase_stmt_prepare (select ... )
altibase_stmt_bind_param(sStmt, sBind1)
altibase_stmt_execute
altibase_stmt_free_result
altibase_stmt_bind_param(sStmt, sBind2) -> Function sequence error
altibase_stmt_execute
```

- 예상 결과

```
altibase_stmt_prepare (select ... )
altibase_stmt_bind_param(sStmt, sBind1)
altibase_stmt_execute
altibase_stmt_free_result
altibase_stmt_bind_param(sStmt, sBind2) -> Success
altibase_stmt_execute
```

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

## BUG-51012 SQLColumns함수에서 SchemaName(스키마 이름) 인자에 NULL이 전달될 경우, 잘못된 데이터가 반환됩니다.

- **module** : mm-cli
- **Category** : Functional Error
- **재현 빈도** : Always
- **설명** : SQLColumns함수에서 SchemaName(스키마 이름) 인자에 NULL이 전달될 경우, 잘못된 데이터가 반환되는 문제를 수정합니다.
- **재현 방법**
  - 재현 절차
  - 수행 결과
  - 예상 결과
- **Workaround**
- **변경사항**
  - Performance view
  - Property
  - Compile Option
  - Error Code

## BUG-51013 ODBC 드라이버를 이용하여 다이너셋(dynaset) 테스트를 수행할 때, Altibase 데이터베이스에 연결이 성공했으나 다이너셋을 지원하지 않는다는 오류가 발생합니다.

- **module** : mm-cli
- **Category** : Functional Error
- **재현 빈도** : Always
- **설명** : ODBC 드라이버를 이용하여 다이너셋 테스트를 수행할 때, Altibase 데이터베이스에 연결이 성공했으나 다이너셋을 지원하지 않는다는 오류가 발생하는 문제를 해결했습니다.

SQLGetInfo 함수에서 SCROLL\_OPTIONS, SCROLL\_CONCURRENCY 의 매개변수가 전달될 때, 내부 로직에서 누락된 값이 있어 올바르게 동작하도록 수정했습니다.

- 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
7.3.0.0.7	7.3.0	9.3.1	7.1.8	7.4.9

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