

Altibase 6.5.1.10.0 Patch Notes

Table of Contents

- [New Features](#)
 - [BUG-50493 한글 검색 가능한 정규 표현식 지원](#)
 - [BUG-50608 Private synonym을 생성할 때, synonym과 객체의 이름을 비교할 때 대소문자를 구분하도록 수정합니다.](#)
- [Fixed Bugs](#)
 - [BUG-50530 비활성화\(disable\)된 인덱스에 대한 DDL 수행시, 서버가 비정상 종료할 수 있습니다.](#)
 - [BUG-50541 Query validate과정에서 query_stack을 초과해서 사용하는 문제를 수정합니다.](#)
 - [BUG-50554 AUDIT OUTPUT METHOD가 1일때 audit 기록에서 누락된 에러 코드가 있습니다.](#)
 - [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 메모리 증가하는 문제가 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 Conflict가 발생하여 충돌 방지 정책에 따라 Replace가 연속으로 수행될 경우, SYNC Fail이 발생할 수 있습니다.](#)
 - [BUG-50621 함수 기반 인덱스를 생성하는 수식이 지나치게 긴 경우, 서버 비정상 종료가 발생합니다.](#)
 - [BUG-50659 ORDER BY 가 SELECT 절의 외부 참조가 있는 SUBQUERY를 alias로 참조 시 결과 오류가 발생합니다.](#)
 - [BUG-50697 JDBC에서 PreparedStatement를 이용하여 ping 쿼리 사용시 메모리 누수가 발생합니다.](#)
- [Changes](#)
 - [Version Info](#)
 - [호환성](#)
 - [프로퍼티](#)
 - [성능 뷰](#)

New Features

BUG-50493 한글 검색 가능한 정규 표현식 지원

- **module** : qp
- **Category** : Other
- **재현 빈도** : Always
- **설명** : 한글 검색을 지원하기 위해 PCRE2(펄 호환 정규 표현식, Perl Compatible Regular Expressions) 라이브러리를 사용한 정규 표현식을 지원합니다. 기존 정규 표현식 처리 방식에서 지원하지 않았던 한글 검색이 가능하며, 아래와 같은 검색 기능이 추가되었습니다.
 - 역참조(Backreference)
 - 전방 탐색(Lookahead)
 - 후방 탐색(Lookbehind)
 - 조건부 정규 표현식(Conditional pattern)
 - 문자 속성 정규 표현식(Character with the xx property)

이 기능을 사용하려면 REGEXP_MODE 프로퍼티의 값을 1로 설정해야 하며, Altibase 서버 캐릭터셋이 US7ASCII 또는 UTF-8일 때만 사용할 수 있습니다. 아래의 쿼리로 프로퍼티의 값을 변경할 수 있으며, 변경된 설정을 적용하려면 세션 재접속이 필요합니다.

```
ALTER SYSTEM SET REGEXP_MODE=1;
```

Altibase 에서 사용한 PCRE2 라이브러리의 버전은 10.40 입니다. PCRE2 라이브러리에 대한 자세한 내용은 [PCRE2 홈페이지](#)를 참조하세요.

- **재현 방법**
 - 재현 절차
 - 수행 결과
 - 예상 결과
- **Workaround**
- **변경사항**
 - Performance view
 - Property
 - 추가
 - **REGEXP_MODE**
 - 기본값 : 0
 - 속성 : 변경가능, 단일값
 - 값의 범위: [0,1]
 - 설명 : 정규 표현식 모드를 설정하는 프로퍼티로 설정값의 의미는 아래와 같다.
 - 0: Altibase 정규 표현식 모드. POSIX Basic Regular Expression (BRE)과 Extended Regular Expression(ERE)을 일부 지원한다.
 - 1: PCRE2 호환 모드. 펄 호환 정규 표현식 (Perl Compatible Regular Expressions, PCRE2) 라이브러리의 정규 표현식 문법을 지원한다.

이 모드는 Altibase 서버 캐릭터셋이 US7ASCII 또는 UTF-8일 때 사용할 수 있다.

- Compile Option
- Error Code

BUG-50608 Private synonym을 생성할 때, 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-50530 비활성화(disable)된 인덱스에 대한 DDL 수행시, 서버가 비정상 종료할 수 있습니다.

- **module** : sm
- **Category** : Fatal
- **재현 빈도** : Always
- **설명** : 비활성화(disable)된 인덱스에 대한 DDL 수행시, 서버가 비정상 종료되는 문제를 수정하였습니다. 또한, 비활성화된 인덱스에 대한 통계 정보 수집 프로시저에서 내부적으로 비활성화된 인덱스를 무시하고 성공으로 처리하는 버그도 함께 수정되었습니다. 이번 패치 적용 후 부터 비활성화된 인덱스에 대한 통계정보 수집 프로시저를 수행하면, [ERR-11084 : The Index is disabled.] 에러메시지가 출력됩니다.
- **재현 방법**
 - **재현 절차**

```
CREATE TABLE T1( I1 CHAR(12), I2 NUMBER(9,3), I3 CHAR DEFAULT 'C', I4
NUMBER(9,3)) TABLESPACE SYS_TBS_DISK_DATA;
ALTER TABLE T1 ALL INDEX DISABLE;
CREATE INDEX T1_IDX3 ON T1(I3, I2);
EXEC GATHER_INDEX_STATS('SYS', 'T1_IDX3');
```

- **수행 결과**

```
Execute success.
```

- **예상 결과**

```
[ERR-11084 : The Index is disabled.]
```

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

- 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 재시작 시, 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 메모리 증가하는 문제가 있습니다.

- **module** : mm-plancache
- **Category** : Memory Error
- **재현 빈도** : Always
- **설명** : 동일한 NO_PLAN_CACHE 힌트를 사용하는 쿼리를 여러 번 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-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(20),
    primary key (I1, I3)
);

insert into t1 values (21,0,1,2);
insert into t1 values (23,2,1,2);
insert into t2 values (21,0,2,'data_ing_1');
insert into t2 values (22,1,2,'modified data_ing_1');
insert into t2 values (23,2,2,'modified data_ing_1');

select /*+no_plan_cache */
    a1.i4,
    a2.data
from t1 a1,
     t2 a2
where exists (
    SELECT /*+ no_unnest */ '0' FROM t2 "AAA2" WHERE AAA2.i1 <= 23 AND a2.ID
    = AAA2.ID HAVING a2.i1 = MAX(AAA2.i1));
```

- 수행 결과


```

I4          DATA
-----
2          data_ing_1
2          data_ing_1
2          modified data_ing_1
2          modified data_ing_1
2          modified data_ing_1
2          modified data_ing_1
6 rows selected.

```

- 예상 결과

```

I4          DATA
-----
2          modified data_ing_1
2          modified data_ing_1
2 rows selected.

```

- Workaround

```

/*+ no_unnest */ 힌트 제거
select /*+no_plan_cache */
      a1.i4,
      a2.data
from   t1 a1,
      t2 a2
where  exists (
SELECT '0' FROM t2 "AAA2" WHERE AAA2.i1 <= 23 AND a2.ID = AAA2.ID HAVING
a2.i1 = MAX(AAA2.i1));

```

- 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

```

drop table t1;
drop table t2;
drop table t3;
create table t1
(
  REV                integer not null,
  REVTYPE            smallint,
  tm_id integer not null,
  map_KEY            integer not null,
  map_id             integer not null
);
alter table t1 add constraint t1_pk primary key (REV, tm_id, map_KEY,
map_id);

insert into t1 (REVTYPE, REV, tm_id, map_id, map_KEY) values (0, 15, 1,
1, 1);
insert into t1 (REVTYPE, REV, tm_id, map_id, map_KEY) values (0, 15, 1,
2, 2);
insert into t1 (REVTYPE, REV, tm_id, map_id, map_KEY) values (2, 17, 1,
2, 2);
insert into t1 (REVTYPE, REV, tm_id, map_id, map_KEY) values (2, 17, 1,
1, 1);

```

```

create table t2
(
    REV      integer not null,
    REVTYPE  smallint,
    id       integer not null,
    str      varchar(255)
);
alter table t2 add constraint t2_pk primary key (REV, id);

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');

create table t3
(
    NUMERIC_VALUE integer,
    REV            integer not null,
    REVTYPE        smallint,
    id             integer not null
);
alter table t3 add constraint t3_pk primary key (REV, id);

insert into t3(NUMERIC_VALUE, REV, REVTYPE, ID) values (1, 15, 0, 1);
insert into t3(NUMERIC_VALUE, REV, REVTYPE, ID) values (2, 15, 0, 2);
insert into t3(NUMERIC_VALUE, REV, REVTYPE, ID) values (3, 16, 1, 2);

select /*+ no_plan_cache use_full_store_n1(AA BB ) */ AA.NUMERIC_VALUE
from t3 AA,
     t2 BB
where
    (( EXISTS (SELECT '0' FROM t2 "BB2" WHERE BB.ID = BB2.ID HAVING BB.REV
= MAX(BB2.REV)))
    and
    (EXISTS (SELECT '0' FROM t3 "AA2" WHERE AA.ID = AA2.ID HAVING AA.REV =
MAX(AA2.REV)))
    )
    or
    ( BB.REV = 17 and AA.REV=1)
;

```

- 수행 결과

```

NUMERIC_VALUE
-----
1
1
1
2
2
2
3
3
3
9 rows selected.

```

- 예상 결과

```

NUMERIC_VALUE
-----
1
3
2 rows selected.

```

- **Workaround**

- 변경사항

- Performance view
- Property
- Compile Option
- Error Code

BUG-50617 이중화 SYNC 수행 중 연속적으로 Conflict가 발생하여 충돌 방지 정책에 따라 Replace가 연속으로 수행될 경우, SYNC Fail이 발생할 수 있습니다.

- **module** : rp-receiver
- **Category** : Functional Error
- **재현 빈도** : Always
- **설명** : 이중화 SYNC 수행 중 연속적으로 Conflict가 발생하여 충돌 방지 정책에 따라 Replace가 연속으로 수행될 경우, SYNC Fail이 발생하는 문제를 수정합니다.
- **재현 방법**
 - 재현 절차
 - 수행 결과
 - 예상 결과
- **Workaround**
- **변경사항**
 - Performance view
 - Property
 - Compile Option

- Error Code

BUG-50621 함수 기반 인덱스를 생성하는 수식이 지나치게 긴 경우, 서버 비정상 종료가 발생합니다.

- **module** : qp-ddl-dcl-pvo
- **Category** : Fatal
- **재현 빈도** : Always
- **설명** : 함수 기반 인덱스를 생성하는 수식이 지나치게 긴 경우, 서버가 비정상 종료되는 문제를 수정합니다.
- **재현 방법**
 - 재현 절차
 - 수행 결과
 - 예상 결과
- **Workaround**
- **변경사항**
 - Performance view
 - Property
 - Compile Option
 - Error Code

BUG-50659 ORDER BY 가 SELECT 절의 외부 참조가 있는 SUBQUERY를 alias로 참조 시 결과 오류가 발생합니다.

- **module** : qp
- **Category** : Reliability
- **재현 빈도** : Always
- **설명** : ORDER BY 절에서 SELECT 절의 외부 참조가 있는 서브쿼리를 alias로 참조할 때 결과 오류가 발생하는 문제를 수정합니다.

이 버그는 아래의 4가지 조건이 모두 만족한 경우에만 발생합니다.

1. 디스크 테이블에서 ORDER BY시 디스크 템프를 사용한 경우
2. 서브쿼리의 alias를 ORDER BY에 사용한 경우
3. SELECT의 target에 서브쿼리가 2개 이상 사용되고, 그 중에 ORDER BY 절에 사용된 alias 의 서브쿼리보다 앞에 있는 서브쿼리에서 외부참조 컬럼이 사용된 경우
4. 서브쿼리에 사용된 외부 참조 컬럼이 FROM 절의 인라인 뷰(Inline View)나 일반 뷰에 사용되면서, 뷰머지(View Merge)로 동작한 경우

- **재현 방법**
 - 재현 절차

```
drop table tb1;
drop table tb2;
drop table tb3;
```

```

CREATE TABLE TB1 ( TB1_C1 VARCHAR(10) PRIMARY KEY,
                    TB1_C2 VARCHAR(10) NOT NULL,
                    TB1_C3 VARCHAR(10) NOT NULL )
tablespace sys_tbs_disk_data
;
insert into tb1 values ('aaa','aaa','aaa');
insert into tb1 values ('bbb','aaa','bbb');
insert into tb1 values ('ccc','aaa','ccc');
insert into tb1 values ('ddd','ccc','ddd');
insert into tb1 values ('eee','eee','eee');
CREATE TABLE TB2 ( TB2_C1 VARCHAR(10) PRIMARY KEY,
                    TB2_C2 VARCHAR(10) NOT NULL,
                    TB2_C3 VARCHAR(10) NOT NULL,
                    TB2_C4 NUMERIC(5,2),
                    TB2_C5 DATE DEFAULT SYSDATE )
tablespace sys_tbs_disk_data;
insert into tb2 values('aaa','aaa','aaa',1,sysdate);
insert into tb2 values('bbb','aaa','bbb',1,sysdate);
insert into tb2 values('ccc','bbb','ccc',1,sysdate);
insert into tb2 values('eee','eee','eee',1,sysdate);
insert into tb2 values('fff','eee','fff',1,sysdate);
CREATE TABLE Tb3 ( TB3_C1 VARCHAR(10) PRIMARY KEY,
                    TB3_C2 VARCHAR(10) NOT NULL )
tablespace sys_tbs_disk_data;
insert into tb3 values ( 'aaa','aaa' );
insert into tb3 values ( 'bbb','bbb' );
insert into tb3 values ( 'ccc','ccc' );
insert into tb3 values ( 'fff','fff' );
SELECT
  (SELECT X.TB1_C3 FROM TB1 X WHERE X.TB1_C1 = A.TB2_C1)
,(SELECT X.TB1_C2 FROM TB1 X WHERE X.TB1_C1 = A.TB2_C1) AS AAAA
FROM TB2 A
WHERE EXISTS (SELECT 1 FROM TB3 X WHERE X.TB3_C2 = A.TB2_C1 )
ORDER BY AAAA ASC;

```

○ 수행 결과

```
(SELECTX.TB1_C3FROMTB1XWHEREX.TB1_C1=A.TB2 AAAA
```

```
-----
aaa
ccc
bbb
```

```
4 rows selected.
```

○ 예상 결과

```
(SELECTX.TB1_C3FROMTB1XWHEREX.TB1_C1=A.TB2 AAAA
```

```
-----
aaa      aaa
ccc      aaa
bbb      aaa
```

```
4 rows selected.
```

• Workaround

```

SELECT
  (SELECT X.TB1_C3 FROM TB1 X WHERE X.TB1_C1 = A.TB2_C1)
, (SELECT X.TB1_C2 FROM TB1 X WHERE X.TB1_C1 = A.TB2_C1) AS AAAA
FROM TB2 A
WHERE EXISTS (SELECT /*+ no_unnest */ 1 FROM TB3 X WHERE X.TB3_C2 = A.TB2_C1
)
ORDER BY AAAA ASC;

```

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

BUG-50697 JDBC에서 PreparedStatement를 이용하여 ping 쿼리 사용시 메모리 누수가 발생합니다.

- **module** : mm-jdbc
- **Category** : Functional Error
- **재현 빈도** : Always
- **설명** : JDBC에서 PreparedStatement를 이용하여 ping 쿼리 사용시 메모리 누수가 발생하는 문제를 수정하였습니다. 이 버그를 적용하려면 JDBC 드라이버를 패치해야 합니다.
- **재현 방법**
 - 재현 절차

```

Connection sConn = getConnection("20300");
PreparedStatement sStmt = sConn.prepareStatement("/* PING */ SELECT 1
");
ResultSet sRs = sStmt.executeQuery();
if (sRs.next())
{
    sRs.close();
}
sStmt.executeQuery();
sStmt.executeQuery();
sStmt.executeQuery();
sStmt.close();
sConn.close();

```

- 수행 결과

메모리 사용량이 계속 증가함

- 예상 결과

메모리 사용량이 계속 증가하지 않음

- **Workaround**

ping 쿼리 대신 `select 1 from dual` 사용

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

Changes

Version Info

altibase version	database binary version	meta version	cm protocol version	replication protocol version
6.5.1.10.0	6.3.1	8.1.1	7.1.3	7.4.5

Altibase 6.5.1 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.

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