Altibase 7.1.0.7.0 Patch Notes

Table of Contents

New Features

BUG-49593 The datetime function TIMESTAMPADD is supported as an SQL function.

- **module** : mt-function
- **Category** : Functionality
- **Reproducibility** : Always
- **Description** : The datetime function TIMESTAMPADD is supported as an SQL function.

  With the addition of the TIMESTAMPADD function, objects (tables, indexes, stored procedures, stored functions, etc.) with the name TIMESTAMPADD cannot be created. Also, if an object with the name TIMESTAMPADD was created in a version prior to Altibase 7.1.0.7.0, it cannot be used from Altibase 7.1.0.7.0. ERR-31001: SQL syntax error occurs when creating or using an object named TIMESTAMPADD in Altibase 7.1.0.7.0 or later.
- **How to reproduce this bug**
  - **Reproduction conditions**

    ```
    SELECT TIMESTAMPADD( SQL_TSI_DAY,1, TO_DATE('2003-01-02', 'YYYY-MM-DD'))
    FROM DUAL;
    ```

  - **Actual Results**

    ```
    ERROR at line 1:ORA-00904: "TIMESTAMPADD": invalid identifier
    ```

  - **Expected Results**

    ```
    '2003-01-03'
    ```
- **Workaround**
- **Changes**
  - Performance view
  - Property
  - Compile Option
  - Error Code

Fixed Bugs

BUG-49488 When setBytes() including LOB data type is processed with executeBatch(), [java.lang.ClassCastException: Altibase.jdbc.driver.datatype.ListBufferHandle] error occurs.

- **module** : mm-jdbc
- **Category** : Functional Error
- **Reproducibility** : Always
- **Description** : [java.lang.ClassCastException: Altibase.jdbc.driver.datatype.ListBufferHandle] error occurs when PreparedStatement.setBytes() is executed with executeBatch() for a BLOB type column. This phenomenon occurs when BLOB data exceeds 65,534 bytes, which is the maximum size that the binary type can handle. Change to process the BLOB type instead of the binary type.

  Add JDBC driver connection properties to keep the existing behavior if you don't want to change it.

  - **JDBC driver connection properties**

    `batch_setbytes_use_lob`

  - **Description**

    When executing PreparedStatement.setBytes() with executeBatch() for a BLOB type column, set whether to process it as binary type or BLOB type. true is treated as a BLOB type and false as a binary type.

  The default value of batch_setbytes_use_lob is true, and if you want to keep the existing behavior, you must add batch_setbytes_use_lob=false and recompile.

  Related descriptions can also be found in the JDBC User's Manual.
- **How to reproduce this bug**
  - **Reproduction conditions**
  - **Actual Results**
  - **Expected Results**
- **Workaround**
- **Changes**
  - Performance view
  - Property
  - Compile Option
  - Error Code

BUG-49545 When using Altibase 7.1.0.6.0 to 7.1.0.6.7 on AIX, the Altibase server may abnormally terminate after performing the query optimizer Fault Tolerance function.

- **module** : id
- **Category** : Fatal
- **Reproducibility** : Always
- **Description** : When Altibase 7.1.0.6.0 to 7.1.0.6.7 version is used on AIX, fixes a bug where the Altibase server abnormally terminates after performing the query optimizer Fault Tolerance function.
- **How to reproduce this bug**
  - **Reproduction conditions**
  - **Actual Results**
  - **Expected Results**
- **Workaround**
- **Changes**
  - Performance view
  - Property
  - Compile Option
  - Error Code

BUG-49557 Improves a phenomenon in which excessive logs are output to altibase_qp.log when [The table structure has been modified.] error is repeated when performing DDL on a replication object.

- **module** : rp-control
- **Category** : Enhancement
- **Reproducibility** : Always
- **Description** : Improves a phenomenon in which excessive logs are output to altibase_qp.log when [The table structure has been modified.] error is repeated when performing DDL on a replication object.

  The [The table structure has been modified.] error is a type of error that is usually resolved by retrying. In case of failure, retries were repeated infinitely, but after 10 retries, [ERR-61069 : Internal server error in replication module (The number of DDL rebuild attempts RPU_REPLICATION_DDL_REBUILD_ERROR_MAX_COUNTT was exceeded.).] Error occurred and changed to handle failure.
- **How to reproduce this bug**
  - **Reproduction conditions**
  - **Actual Results**
  - **Expected Results**
- **Workaround**
- **Changes**
  - Performance view

- Property
- Compile Option
- Error Code

BUG-49611 Altibase server may abnormally terminate due to a memory reference error when executing SQL that joins with the WITH clause.

- **module** : qp
- **Category** : Fatal
- **Reproducibility** : Rare
- **Description** : Fixes a bug where the Altibase server abnormally terminates due to a memory reference error when executing SQL that joins with the WITH clause.
- **How to reproduce this bug**
  - **Reproduction conditions**

```
DROP TABLE COMM_SN_SUBINFO;
CREATE TABLE COMM_SN_SUBINFO(
        SVCNB VARCHAR(6) FIXED ,
        SUBNB VARCHAR(20) FIXED ,
        SUBKIND INTEGER FIXED,
        COM_NAME VARCHAR(50) VARIABLE,
        OFC_CD VARCHAR(10) FIXED,
        OFC_NAME VARCHAR(50) VARIABLE,
        AGNC_NAME VARCHAR(50) VARIABLE,
        END_DATE DATE FIXED,
        BEGIN_DATE DATE VARIABLE,
        CUST_NO VARCHAR(22) VARIABLE,
        SVC_OPT1 VARCHAR(16) VARIABLE,
        SVC_OPT2 VARCHAR(16) VARIABLE,
        PARANB VARCHAR(20) VARIABLE,
        SVCSTATUS INTEGER VARIABLE,
        VXMURLPSTN VARCHAR(128) VARIABLE,
        VXMLURLMOBILE VARCHAR(128) VARIABLE,
        CHAKIND VARCHAR(10) VARIABLE,
        COM_MODE VARCHAR(6) VARIABLE,
        CHANB VARCHAR(20) VARIABLE,
        STORE_OWNER_NO VARCHAR(20) VARIABLE,
        STORE_NAME VARCHAR(50) VARIABLE );
DROP INDEX PK_COMM_SN_SUBINFO;
CREATE UNIQUE INDEX PK_COMM_SN_SUBINFO ON COMM_SN_SUBINFO( SUBNB );
INSERT INTO COMM_SN_SUBINFO ( SUBNB ) SELECT LEVEL FROM DUAL CONNECT BY
LEVEL < 10;
DROP TABLE COMM_FP_SUBINFO;
CREATE TABLE COMM_FP_SUBINFO(
        SVCNB VARCHAR(6) FIXED ,
        SUBNB VARCHAR(20) FIXED ,
        SUBKIND INTEGER FIXED,
        COM_NAME VARCHAR(50) VARIABLE,
        OFC_CD VARCHAR(10) FIXED,
        OFC_NAME VARCHAR(50) VARIABLE,
```

```
        AGNC_NAME VARCHAR(50) VARIABLE,
        END_DATE DATE FIXED,
        BEGIN_DATE DATE VARIABLE,
        CUST_NO VARCHAR(22) VARIABLE,
        SVC_OPT1 VARCHAR(16) VARIABLE,
        SVC_OPT2 VARCHAR(16) VARIABLE,
        PARANB VARCHAR(20) VARIABLE,
        SVCSTATUS INTEGER VARIABLE,
        VXMURLPSTN VARCHAR(128) VARIABLE,
        VXMLURLMOBILE VARCHAR(128) VARIABLE,
        CHAKIND VARCHAR(10) VARIABLE,
        COM_MODE VARCHAR(6) VARIABLE,
        CHANB VARCHAR(20) VARIABLE,
        STORE_OWNER_NO VARCHAR(20) VARIABLE,
        STORE_NAME VARCHAR(50) VARIABLE );
DROP INDEX PK_COMM_FP_SUBINFO;
CREATE UNIQUE INDEX PK_COMM_FP_SUBINFO ON COMM_FP_SUBINFO( SUBNB );
INSERT INTO COMM_FP_SUBINFO ( SUBNB ) SELECT LEVEL FROM DUAL CONNECT BY
LEVEL < 10;
var v1 varchar(10);
exec :v1:= '1';
prepare WITH SEARCH_NB_TBL AS (
       SELECT
       TO_CHAR(regexp_substr(CAST(:v1 AS VARCHAR(10000)),'[^,]+', 1,
level)) AS SEARCH_NB
       FROM DUAL
       CONNECT BY regexp_substr(:v1, '[^,]+', 1, LEVEL) IS NOT NULL
)
SELECT SUBNB
     , CUST_NO
  FROM COMM_SN_SUBINFO A
     , SEARCH_NB_TBL ST
 WHERE A.SUBNB = ST.SEARCH_NB
 UNION
SELECT SUBNB
     , CUST_NO
  FROM COMM_FP_SUBINFO B
     , SEARCH_NB_TBL ST
 WHERE B.SUBNB = ST.SEARCH_NB;
```

    - **Actual Results**
    - **Expected Results**
- **Workaround**
- **Changes**
    - Performance view
    - Property
    - Compile Option
    - Error Code

BUG-49618 If an aggregate function is included in a parallel query, the Altibase server may abnormally terminate due to concurrency issues.

- **module** : qp-select-execute
- **Category** : Fatal
- **Reproducibility** : Rare
- **Description** : Fixes an issue where the Altibase server abnormally terminates due to concurrency issues when an aggregate function is included in a parallel query.
- **How to reproduce this bug**
  - **Reproduction conditions**
  - **Actual Results**
  - **Expected Results**
- **Workaround**
- **Changes**
  - Performance view
  - Property
  - Compile Option
  - Error Code

Changes

Version Info

altibase version	database binary version	meta version	cm protocol version	replication protocol version
7.1.0.7.0	6.5.1	8.10.1	7.1.7	7.4.7

> You can check the module version change history in Version_Histories.

Compatibility

Database binary version

The database binary version has not changed.

Meta Version

The meta version has not changed.

CM protocol Version

The cm protocol version has not changed.

Replication protocol Version

The replication protocol version has not changed.

Altibase Server Properties

Added Properties

Changed Properties

Deleted Properties

Performance Views

Added Performance Views

Changed Performance Views

Deleted Performance Views