Altibase FAQ | SQL & Schema Objects

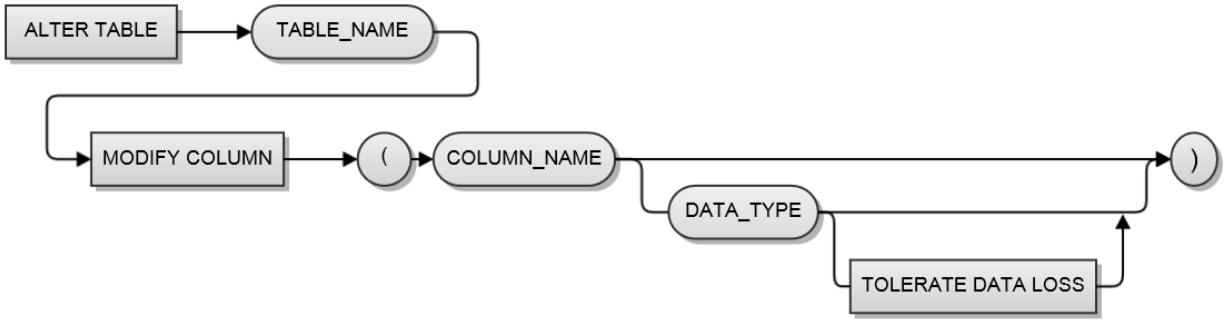
How to modify column

Version

Starting from Altibase version 5.3.3 or later, the column type and length of a table can be modified by using the ALTER TABLE ~ MODIFY COLUMN ~ statements.

Statement

ALTER TABLE table_name MODIFY COLUMN (column_name column_type(length))



TOLERATE DATA LOSS Option

If the table data is not NULL, data loss may occur depending on the conversion type. In order to change the data type at the expense of this data loss, TOLERATE DATA LOSS option can be used.

Conversion of DATE type

When the DATE type is changed, the column data is converted according to the DEFAULT_DATE_FORMATE property.

Precaution

If used incorrectly, the column modify command may cause a load on the DB depending on data loss and the amount of data in the target table, so it should be used with precaution.

1. **Cannot reduce the column size below the original size**

© Altibase Corp.

2. **If the data type of a column is changed, data loss may occur depending on the data type. If the user wants to change the data type at the expense of this data loss, the TOLERATE DATA LOSS option can be used.**
3. **When operating with the target table for replication, it must follow the DDL operation procedure in a replication environment. Please refer to the DDL procedure for the Altibase replication target table.**
4. **If there are many rows in the target table, there may be a delay in operating time and an increase in usage of the logs area.**

> **Info**
>
> When modifying columns, if the table to be changed is a memory table and the current ALTIBASE version is 5.3.3, it is created in the memory table as a copy table for restoration.
>
> In order words, there must be room for memory tablespaces. (For reference, in version 6.1.1, the copy table is saved as a disk tablespace, so if there are only a tablespace and a disk space, it is operatable.)
>
> As recommended by ALTIBASE, if the target table is a memory table, backup is performed with an iloader operation, then a new target table is created and data is imported.

Example

Change the isbn column of the table book to CHAR(20) type and the edition column to BIGINT type.

```
iSQL> ALTER TABLE book MODIFY COLUMN (isbn CHAR(20), edition BIGINT);
Alter success.
```

Below is an example of executing the above query.

```
iSQL> create table t1(c1 integer);
Create success.
iSQL> desc t1;
[ TABLESPACE : SYS_TBS_MEM_DATA ]
[ ATTRIBUTE ]
------------------------------------------------------------------------------
NAME TYPE IS NULL
------------------------------------------------------------------------------
C1 INTEGER FIXED
T1 has no index
T1 has no primary key
iSQL> insert into t1 values(1111111111);
1 row inserted.
iSQL> select * from t1;
C1
--------------
1111111111
1 rows selected.
iSQL> alter table t1 modify(c1 numeric(10));
[ERR-312EE : Invalid length for the data type
0001 : alter table T1 modify(C1 NUMERIC(10))
```

© Altibase Corp.

```
  ^ ^
  ]
  iSQL> alter table t1 modify(c1 numeric(10) tolerate data loss);
  Alter success.
  iSQL> desc t1;
  [ TABLESPACE : SYS_TBS_MEM_DATA ]
  [ ATTRIBUTE ]
  ------------------------------------------------------------------------------
  NAME TYPE IS NULL
  ------------------------------------------------------------------------------
  C1 NUMERIC(10) FIXED
  T1 has no index
  T1 has no primary key ¦
```

Reference

For more detailed information on how to use it, refer to how to use modify columns in the SQL Reference Manual.

© Altibase Corp.