

DBMaker/DBMaster FAQ Document

Question:

Why does the error 8349 occur? But the length of the data does not over the 1024 byte, although the index including a Varchar (2000) type column? And how to avoid this occur? (From DBMR1811)

Answer:

(1). First, when creating an index which including varchar type column, DBMaker does not check the length of this index.

For example:

```
dmSQL> create table dbmr1811(id SERIAL(1),application_name VARCHAR(2000)
default null );

dmSQL> create index idx_dbmr1811 on dbmr1811 (application_name);
```

The above idx_dbmr1811 includes a varchar(2000) column whose length is over 1024, but you can create the index successfully, Because DBMaker will not check the length until using the index.

(2). When using the above index, DBMaker will check length of part indices depend on different execution plan. When one execution plan uses nested join, it does not know the length of data, so it uses the definition length (2000 Bytes in this case) of varchar type as data length to allocate buffer. When checking length, if finding the length of checked index over 1024 Bytes, the error 8394 will occur.

For example:

```
dmSQL> create table c1(a1 VARCHAR(2000) default null );

dmSQL> create index idx_c1 on c1 ( a1 );

dmSQL> create table d1(a1 VARCHAR(2000) default null );

dmSQL> create index idx_d1 on d1 ( a1 );

dmSQL> select * from c1 JOIN d1 on c1.a1 = d1.a1 where 1=1;

ERROR (8349): [DBMaker] total length of index key exceeds maximum length (1024 Bytes)
```

DBMaker/DBMaster FAQ Document

(3).There are two methods to avoid it.

One is executing the “update statistics”. After executing "update statistics", execution plan uses merge join and the length of varchar is actual length but not definition length.

The other is using forced Merge join between tables. DBMaker 4.3 provides force optimizer syntax; you can use “MERGE JOIN...” syntax to let query run, then it use the MERGE JOIN but not the Nested JOIN.

For example:

```
dmSQL> select * from c1 MERGE JOIN d1 on c1.a1 = d1.a1 where 1=1;
```

(4).Please pay attention, although successfully creating this type index (include varchar type column), data for inserting can not over 1024 bytes.

(5).You’d better don’t create ordinary index on long column, this type index will occupy many index pages and may not be used when query. So we suggest user creating text index on long column.