

DBMaker/DBMaster FAQ Document

Question:

How to use COPYTABLE to duplicate a new table from old ones? (DBMR: 2136)

Answer:

We can use call stored procedure "CALL COPYTABLE ()" to duplicate a new table based on an existing table. The usage as below:

```
COPYTABLE(VARCHAR(32)  source_schema_name INPUT,  
          VARCHAR(32)  source_table_name INPUT,  
          VARCHAR(32)  destination_schema_name INPUT,  
          VARCHAR(32)  destination_table_name INPUT,  
          VARCHAR(128) tablespace_lock_mode_option_string INPUT,  
          VARCHAR(2048) where_condition_string INPUT,  
          INT          fg_rename_index INPUT,  
          INT          commit_count INPUT)
```

Copy the source table's definition and data into the destination table.

Need to pay attention to the following comments when use this stored produce:

- (1). It must be run when autocommit is on.
- (2). User can set 0 or 1 for rename index according DCI rule flag.
- (3). Set commit count to commit when copying every nth data. User can set 0 to n.

NOTE: When any error occurs, any operation before the error occurred will be committed. The command that did not execute for copy table will be showed in the _spusr.log. User can set DB_SPLOG directory in the dmconfig.ini or find it in the directory where user executes the application.

DBMaker/DBMaster FAQ Document

The 5th argument needs to pay attention in use process. The 5th argument allows customer set 3 options: Tablespace, Lockmode and Fillfactor. In addition, you can set one of the arguments one time and you also can set 3 options in 5th argument at the same time. Please notice if you specify in same tablespace, all setting will be followed source table, if you specified another tablespace, please remember set different options - lock mode or fillfactor for different value, otherwise, it will take default in different tablespace.

For example:

(1) Create a table named t1 and def the t1 schema.

```
dmSQL> def table t1;  
  
create table SYSADM.T1 (  
  
C1 TIMESTAMP default null )  
  
in DEFTABLESPACE lock mode row fillfactor 100 ;
```

(2) Specify t2 in same tablespace with T1, then def t2.

```
dmSQL> call copytable('SYSADM', 'T1', 'SYSADM', 'T2', "", "", 0, 1);  
  
dmSQL> DEF TABLE T2;  
  
create table SYSADM.T2 (  
  
C1 TIMESTAMP default null )  
  
in DEFTABLESPACE lock mode row fillfactor 100 ;
```

(3) Specified another tablespace for t2 which different with t1, then def t2.

```
dmSQL> call copytable('SYSADM', 'T1', 'SYSADM', 'T2', 'IN TS1 LOCK  
MODE PAGE FILLFACTOR 75', "", 0, 1);  
  
//tablespace : ts1
```

DBMaker/DBMaster FAQ Document

```
//lock mode : page
```

```
//fillfactor : 75
```

```
dmSQL> def table t2;
```

```
Create table SYSADM.T2 (
```

```
  C1 TIMESTAMP default null)
```

```
in TS1 lock mode page fillfactor 75 ;
```

General speaking, lock mode must as row is better.

Version: DBMaker 4.3 or Above

Product: Normal / Bundle

Platform: All