DBMaker

Update Statistics User Guide

Version: 01.01

Document No: 54/DBM546-T12102024-01-UPST
Author: Production Team
Syscom Computer Engineering CO.

Print Date: December 10, 2024

凌羣電腦
THE SYSCOM GROUP

Table of Content

1. Introduction

Welcome to the DBMaker Update Statistics User Guide. This guide discusses the information about statistics and guides the user how to execute UPDATE STATISTICS and related commands. This document is for DBMaker 5.4.6 version.

1.1. Statistics

Statistics represent the amount and distribution of data for a tables and indexes. With statistics, query optimizer can find a more efficient execution plan. However, the statistics will be out of date if data in the table is being inserted, deleted, or updated. Keeping statistics information current helps the database to perform queries more efficiently.

In DBMaker, the UPDATE STATISTICS command should be used to update statistical values and find real time statistics to enhance the efficiency of a query. Simply put, if there's amount of data changed in table, then users should do update statistics.

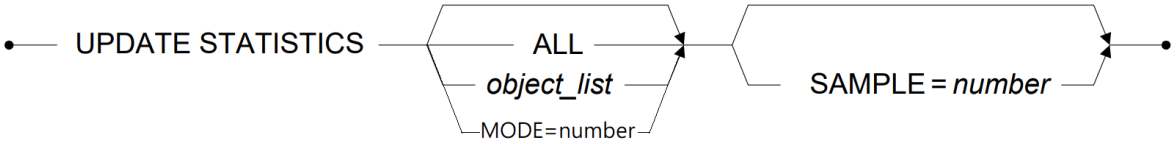
If users need more information about statistics, please refer to *Database Administration User's Guide 19 Query Optimization*.

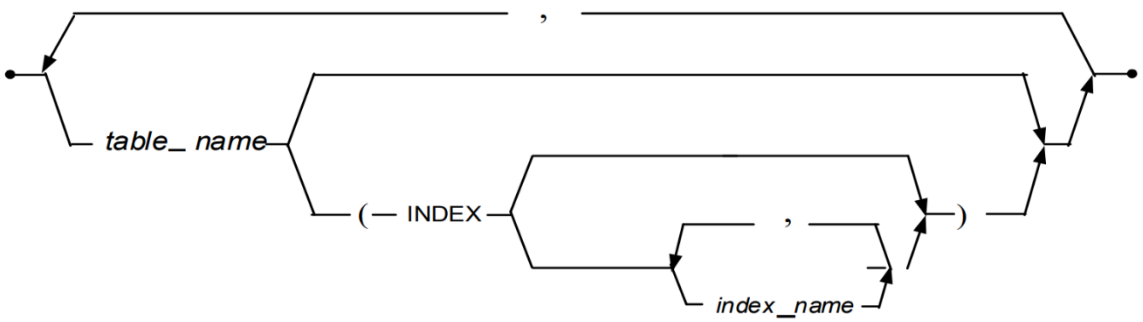
There are three ways to do update statistics in DBMaker database. UPDATE STATISTICS command, Update Statistics Schedule and Update Statistics Daemon. This guide will introduce these ways in the following chapters.

2. Execute Update Statistics

There are different ways to update statistics in database. This chapter will teach users how to execute UPDATE STATISTICS commands. The basic command is UPDATE STATISTICS.

2.1. Update Statistics Syntax





The *object_list* clause

- **ALL**—means forcibly update the statistics values for all schema objects.
- **SAMPLE**—means the sampling rate expressed as a percentage of the whole, an integer between 1 and 100.
- **MODE**—means the mode of sample setting when executing UPDATE STATISTICS command.
  - The default value is 0, means to use the sample rate from database setting (DB_StsSp);
  - set to 1 means to use the sample rate from table setting (UPDATE STATISTICS SET command);
  - set to 2 means UPDATE STATISTICS command will smart decide every table's sample rate.
- **table_name**—means the table to do update statistics
- **index_name**—means the index to do update statistics

➲Example 1:

To update the statistics values for all schema objects, ALL is a default option:

```
dmSQL> UPDATE STATISTICS;
dmSQL> UPDATE STATISTICS ALL;
```

➲Example 2:

To update statistics for all tables including columns, indexes, and system tables with the sampling rate value of 30:

```
dmSQL> UPDATE STATISTICS SAMPLE=30;
```

➲Example 3:

To update statistics for the table t1:

```
dmSQL> UPDATE STATISTICS t1;
```

➲Example 4:

The following update statistics in the database with smart decided sampling rate.

```
dmSQL> UPDATE STATISTICS mode=2;
```

➲Example 5:

The following update statistics in the database with sample ratio defined by table. Users can execute UPDATE STATISTICS SET command to define sample ratio for different tables. Please refer to 4.1.3. UPDATE STATISTICS SET for syntax and examples.

```
dmSQL> UPDATE STATISTICS mode=1;
```

3. Auto Update Statistics Schedule

After version 5.4.6, DBMaker supports using task and schedule to execute UPDATE STATISTICS. This chapter teaches users how to activate Auto Update Statistics Schedule.

3.1. SYSTEM Schedule for Update Statistics

After version 5.4.6, when a database is created, system will automatically create a task UpdStat_Mode2 and a schedule UpdStat_EveryDay. The following are the introduction and how to activate it to do update statistics.

UpdStat_Mode2: a task created by SYSADM, this task means to execute UPDATE STATISTICS mode=2.

```
TASK_CREATE ('UpdStat_Mode2', 'SQL_STATEMENT', 'UPDATE STATISTICS mode=2')
```

UpdStat_EveryDay: a schedule created by SYSADM, this schedule will call execution of task UpdStat_Mode2 every day at 2 a.m.

```
SCHEDULE_CREATE ('UpdStat_EveryDay', 'UpdStat_Mode2', '0 2 * * *', '', '')
```

Enable **UpdStat_EveryDay** to do update statistics:

The default of schedule **UpdStat_EveryDay** is disabled, users need to use stored procedure SCHEDULE_ENABLE to activate it. The following syntax shows users how to enable schedule **UpdStat_EveryDay**.

```
dmSQL> CALL SCHEDULE_ENABLE ('UpdStat_EveryDay');
```

3.2. How to Alter SYSTEM Task and Schedule

A SYSADM can alter UpdStat_Mode2 and UpdStat_EveryDay (ex. changing the mode, schedule timetable…) to define their own update statistics plan. Stored procedure TASK_ALTER and SCHEDULE_ALTER allow users to change task and schedule.

The following syntaxes in this chapter will focus on altering UpdStat_Mode2 and UpdStat_EveryDay.

TASK_ALTER:

```
TASK_ALTER ('UpdStat_Mode2', 'SQL_STATEMENT', ' new update statistics command')
```

SCHEDULE_ALTER:

```
SCHEDULE_ALTER ('UpdStat_EveryDay', 'UpdStat_Mode2', 'new timetable', ' new starttime', ' new endtime')
```

If users need more information and details about TASK_ALTER and SCHEDULE_ALTER, please refer to *SQL Command and Function Reference 5.6 SCHEDULE_ALTER and 5.26 TASK_ALTER.*

➲Example 1:

The following syntax shows users how to alter UpdStat_Mode2 from mode=2 to mode=1

```
dmSQL> CALL TASK_ALTER ('UpdStat_Mode2','SQL_STATEMENT','update statistics mode=1');
```

➲Example 2:

The following syntax shows users how to alter UpdStat_Mode2 to using sample=50

```
dmSQL> CALL TASK_ALTER ('UpdStat_Mode2','SQL_STATEMENT','update statistics sample=50');
```

➲Example 3:

The following syntax shows users how to alter UpdStat_EveryDay from 2 a.m. to 5 a.m.

```
dmSQL> CALL SCHEDULE_ALTER ('UpdStat_EveryDay', 'UpdStat_Mode2', '0 5 * * *', '', '')
```

➲Example 4:

The following syntax shows users how to alter UpdStat_EveryDay from 2 a.m. to 5 a.m. and starts from 2024-07-01 00:00:00 to 2024-07-30 00:00:00

```
dmSQL> CALL SCHEDULE_ALTER ('UpdStat_EveryDay', 'UpdStat_Mode2', '0 5 * * *', '2024-07-01 00:00:00', '2024-07-30 00:00:00')
```

3.3. Define User's Task and Schedule for Update Statistics

This chapter will teach users how to create their own tasks and schedules to do update statistics by examples. Stored procedure about tasks

➲Example 1:

The following syntaxes show users how to do update statistics by task and schedule from creating task. The updstat_task1 executes update statistics, updstat_sch1 sets the execute time at 5 a.m. every day from 2024-01-01 00:00:00 to 2025-01-01 00:00:00

```
dmSQL> CALL TASK_CREATE ('updstat_task1','SQL_STATEMENT', 'update statistics');

//create a task updstat_task1, this task will execute update statistics

dmSQL> CALL SCHEDULE_CREATE ('updstat_sch1', 'updstat_task1', '0 5 * * *', '2024-01-01 00:00:00', '2025-01-01 00:00:00');

//create a schedule updstat_sch1 sets the execute time at 5 a.m. every day from 2024-01-01 00:00:00 to 2025-01-01 00:00:00

dmSQL> CALL SCHEDULE_ENABLE ('updstat_sch1');

//enable updstat_sch1 to start update statistics plan
```

4. Auto Update Statistics Daemon

DBMaker provides a daemon to automatically update statistics for the entire database. Not all statistics on all tables are regenerated, but rather an optimum sample ratio based on how recently statistics were updated for a table and how much the table has been changed since the last time its statistics were updated. Users can select update statistics mode by themselves, set different sample ratio for different table, and change the initial time and interval of update statistics. Users can query system table SYSUSER to get update statistics status. The status is a character string stored in column SQL_CMD. Additional, If the update statistics command is executing, users can abort it with the stored procedure SETSYSTEMOPTION, and the connection is not broken.

Please note, update statistics daemon will not work if update statistics server is not existed or has been terminated. And users cannot set table statistics on temporary table.

Users can enhance the performance of update statistics by the following methods: dmconfig.ini setting, every table setting, SETSYSTEMOPTION, obtain update statistics status and abort update statistics command.

NOTE: After version 5.4.6, task and schedule can execute update statistics command, we'll suggest users replace update statistics daemon with task and schedule.

4.1. Update Statistics Keywords

This chapter tells users keywords about update statistics in dmconfig.ini.

The following keywords are keywords about update statistics, users can set these keywords in dmconfig.ini or by SETSYSTEMOPTION, SETSYSTEMOPTIONW command.

- **DB_StSvr:** This keyword is used to activate the auto update statistics server. A value of 1 indicates that the server is started. A value of 0 indicates that the server is not running. If the auto update statistics server is activated, it will recalculate database statistics according to the value of DB_StsTm and DB_StsTv.
- **DB_StMod:** This keyword specifies the update statistics daemon mode. Setting the value to 0 enables startup update statistics daemon in general mode and the sample ratio will be decided by value of the keyword DB_StsSp in dmconfig.ini; setting the value to 1 enables startup update statistics daemon in every table setting mode, that is to say, the sample ratio will be decided by every table's mode and sample ratio. Users can set every table's mode and sample ratio with the command "UPDATE STATISTICS SET".
- **DB_StsTm:** This keyword specifies the first time of the update statistics daemon will perform auto update statistics. The format for DB_StsTm is yyyy-mm-dd hh:mm:ss.
- **DB_StsTv:** This keyword specifies update statistics daemon time interval. The value like "1-12:30:00" means time interval is every one day, 12 hours and 30 minutes.

- **DB_StsSp:** This keyword specifies update statistics page data sample ratio. A value of -1 indicates the database intelligently obtain sample ratio; A value of 0 indicates the database don't update statistics value. Additional, users can set sample ratio to a number between 1 and 100.

The following database example are update statistics keywords' default value

```
[DBSAMPLE]
; Here omit other keywords
DB_StSvr = 1                        // activate auto update statistics daemon server
DB_StMod = 0                        // use sample ratio set by DB_StsSp
DB_StsTm = 1970-01-01 03:00:00      // the first update statistics time
DB_StsTv = 1-00:00:00               // the time interval is 1 day
DB_StsSp = -1                       // the database will intelligently obtain sample ratio
```

Please refer to *Database Administrator's Guide* for more information.

4.1.1. DMCONFIG.INI SETTING

➲Example 1:

Before starting database, configure some keywords related to update statistics daemon in the dmconfig.ini file:

```
[DB1]
; Here omit other keywords
DB_StSvr = 1                        // activate auto update statistics daemon server
DB_StMod = 1                        // use sample ratio set in table
DB_StsTm = 2010-10-10 10:00:00      // the first update statistics time
DB_StsTv = 12:00:00                 // the time interval is 12 hours
DB_StsSp = 70                       // the sample ratio is 70
```

Now, start database, then update statistics daemon will auto update statistics according to the schedule.

4.1.2. SETSYSTEMOPTION AND SETSYSTEMOPTIONW

The system option of automatic update statistics includes **STSVR**, **STMOD**, **STSTM**, **STSTV** and **STSSP**. DBMaker supports setting these system options at run time by calling the system stored procedure SETSYSTEMOPTION or SETSYSTEMOPTIONW. Additionally, users can obtain these system options info by calling the system stored procedure GETSYSTEMOPTION. For more information on how to use the three system stored procedures SETSYSTEMOPTION, SETSYSTEMOPTIONW and GETSYSTEMOPTION, please refer to the *SQL Command and Function Reference and ODBC Programmer's Guide*.

➲Example 1:

Allow user to set all parameters about update statistics daemon during runtime:

```
dmSQL> CALL SETSYSTEMOPTION('STSVR','1');          //activate automatic update statistic server
dmSQL> CALL SETSYSTEMOPTION('STMOD','1');                    //reset DB_StMod
dmSQL> CALL SETSYSTEMOPTION('STSTM','2009/6/6 20:30:00');    //reset DB_StsTm
dmSQL> CALL SETSYSTEMOPTION('STSTV','7-00:00:00');           //reset DB_StsTv
dmSQL> CALL SETSYSTEMOPTION('STSSP','60');                   //reset DB_StsSp
```

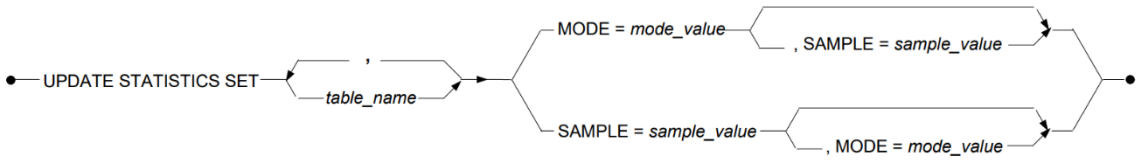
➲Example 2:

Users can execute CALL GETSYSTEMOPTION to select the parameters above:

```
dmSQL> CALL GETSYSTEMOPTION('STSVR', ? );
dmSQL> CALL GETSYSTEMOPTION('STMOD', ? );
dmSQL> CALL GETSYSTEMOPTION('STSTM', ? );
dmSQL> CALL GETSYSTEMOPTION('STSTV', ? );
dmSQL> CALL GETSYSTEMOPTION('STSSP', ? );
```

4.1.3. UPDATE STATISTICS SET

The UPDATE STATISTICS SET command can specify every table's update statistics method and sample ratio for update statistics daemon when it starts in every table setting mode (DB_StMod=1) or executing UPDATE STATISTICS Mode=1. Different from above chapters, UPDATE STATISTICS SET won't alter keywords in dmconfig.ini.



- **table_name**: Name of the table
- **mode_value**: The table update statistics method
  - 0: Sample ratio of table uses value of DB_StsSp in dmconfig.ini. The default value of DB_StsSp is 100.
  - 1: Sample ratio of table uses table update statistics sample ratio which be set in sample_value.
  - 2: Sample ratio of table will be obtained intelligently.
- **sample_value**: The table update statistics sample ratio-1: Intelligently obtain sample ratio0: The database does not need to update statistics value1 ~ 100: Table update statistics sample ratio, the default value is 100

➲Example 1:

Setting the update statistics method and sample ratio for the table jeff.tb_staff, users can select from SYSTABLE to get every table's mode and sample ratio:

```
dmSQL> UPDATE STATISTICS SET jeff.tb_staff MODE = 1，SAMPLE = 80;

dmSQL> SELECT TABLE_NAME, TABLE_OWNER, UPD_STS_MODE,
UPD_STS_SAMPLE FROM SYSTABLE;

TABLE_NAME   TABLE_OWNER   UPD_STS_MODE   UPD_STS_SAMPLE
==========   =============   ==============   ================
TB_STAFF       JEFF                              1                    80

1 rows selected
```

➲Example 2:

Setting the update statistics method and sample ratio for the table jeff.tb_staff and jim.tb_salary :

```
dmSQL> UPDATE STATISTICS SET jeff.tb_staff, jim.tb_salary MODE = 1, SAMPLE = 60;

dmSQL> SELECT table_name, table_owner, upd_sts_mode, upd_sts_sample FROM
SYSTABLE;

TABLE_NAME   TABLE_OWNER   UPD_STS_MODE   UPD_STS_SAMPLE
==========   =============   ==============   ================
TB_STAFF       JEFF                              1                    60
TB_SALARY     JIM                               1                    60

2 rows selected
```

➲Example 3:

This example shows users using UPDATE STATISTICS SET and UPDATE STATISTICS

```
dmSQL> UPDATE STATISTICS SET t2 MODE=1, SAMPLE=10;

dmSQL> UPDATE STATISTICS set t3 MODE=2;

dmSQL> SELECT table_name, upd_sts_mode, upd_sts_sample FROM SYSTABLE;

TABLE_NAME                      UPD_STS_MODE   UPD_STS_SAMPLE

==========================   ==============   ================

T1                                              0                    100

T2                                              1                     10

T3                                              2                    100

dmSQL> UPDATE STATISTICS mode=1, sample=50;

// update statistics command will execute as follows

// t1: mode=0, follow sql command sample=50

// t2: mode=1, follow table setting sample=10

// t3: mode=2, use smart decided sample, in this case sample=100
```

5. Update Statistics Advices

This chapter will give users some advice on doing update statistics.

5.1. Update Statistics Automatically

If the tables in the database do not have a large number of data pages which will take a significant amount of time to do update statistics, users may consider enabling auto update statistics.

Schedule server:

```
call schedule_enable ('UpdStat_EveryDay');

// task and schedule will be created by server. Users only have to enable it. Please refer to chapter 3. Auto Update Statistics Schedule for more information.
```

Update statistics daemon:

```
DB_STSVR=1

DB_STMOD=0

DB_STSSP=-1
```

When sample ratio is smart decided, the following factors will trigger the execution of update statistics. If table matches one of the following factors, sample ratio will be 100 and do update statistics:

1. **Table hasn't done update statistics (statistics = -1)**
2. **Number of data page is small (data page <= 20)**
3. **Difference of number of data page is large (diff page > 2)**
4. **The latest update statistics is old (>10 days)**

5.2. Suggestions for Update Statistics of Different Types of Tables

Usually users should do update statistics regularly. But when table size increases, more time will be used while doing update statistics. Larger tables may delay other tables. Users can consult the following list to decide their own update statistics settings.

1. **Table with heavy SELECT operations and minimal DDL/DML**

   Suggestion: Refer to factor 3 from chapter 5.1. Difference of data page is small, means statistics won't change rapidly.

2. **Tables with high INSERT, UPDATE, and DELETE**

   Suggestion: Highly transmitted table will let statistics be out of date more quickly, users should do update statistics more often.

3. **Historical Tables (no longer subject to change)**

   Suggestion: If data doesn't change, then users don't need to do update statistics.

4. **Very Large Tables (millions or billions of records)**

   Suggestion: Larger tables need more time to do update statistics. If this table doesn't update frequently, update statistics will be unnecessary; If this table is a highly transmitted table, we will suggest users do update statistics respectively.

6.Scenario

This chapter will show users some common situations, these situations are often used in customer's database.

Depends on table size and usage, users may not need to do update statistics on every table every time. The following examples will show users how to ignore specific table(s) or specify certain table while doing update statistics.

NOTE: After version 5.4.6, task and schedule can execute update statistics command, we'll suggest users replace update statistics daemon with task and schedule.

6.1. Update Statistics but Ignore Specific Table

Ignore specific table(s) is often used when update statistics takes a long time and reduces efficiency. The following examples will show users how to ignore table while doing update statistics in different ways.

6.1.1. UPDATE STATISTICS COMMAND/SCHEDULE

Assume there's a table T2 that users don't want to do update statistics, users can refer the following steps:

1. Use UPDATE STATISTICS SET to decide the sample ratio of t2

```
UPDATE STATISTICS SET t2 SAMPLE=0;     //set t2's sample ratio=0
```

2. After t2's sample ratio is set to 0, users can execute UPDATE STATISTICS command

```
UPDATE STATISTICS MODE=1;

//do update statistics using sample ratio from table. In this case t2's sample ratio=0, so t2 will be ignored
```

3. Or use task and schedule to do update statistics:

```
CALL TASK_CREATE ('Updstat_Mode1','SQL_STATEMENT','UPDATE STATISTICS MODE=1');

CALL SCHEDULE_CREATE ('Updstat_Test1','Updstat_Mode1', '@once','','');

CALL SCHEDULE_ENABLE ('Updstat_Test1');

//create task and schedule to do update statistics, this schedule will execute immediately when it is enabled
```

6.1.2. UPDATE STATISTICS DAEMON

Update Statistics Daemon will do update statistics automatically. Base on keywords DB_STMOD and DB_STSSP in dmconfig.ini, users can define the statistics mode and

sample ratio while updating statistics. In this example, because we want to do update statistics base on table setting

1. dmconfig.ini setting:

```
DB_STSVR=1      //activate auto update statistics server
DB_STMOD=1      //the update statistics execution will be decided by every table's mode and sample ratio
```

2. Start database and execute the following commands:

```
UPDATE STATISTICS SET t2 SAMPLE=0;     //set t2's sample ratio=0
```

6.2. Update Statistics on Specific Table

The following examples will show users how to do update statistics on specific table. This situation often happens on a highly transmitted table.

6.2.1. UPDATE STATISTICS COMMAND/SCHEDULE

Assume t2 is a highly transmitted table, and users wants to do update statistics on this table, please execute the following command:

```
UPDATE STATISTICS t2 SAMPLE=100;        //update statistics on t2 with sample ratio=100
```

A highly transmitted table usually needs more often update statistics, the following example will show users how to update statistics on t2 by task and schedule at 0 a.m. and 12 p.m.:

```
CALL TASK_CREATE ('Updstat_t2','SQL_STATEMENT','UPDATE STATISTICS t2 SAMPLE=100');

CALL SCHEDULE_CREATE ('Updstat_Test2','Updstat_t2', '0 0,12 * * *','','');

CALL SCHEDULE_ENABLE ('Updstat_Test2');
```

6.2.2. UPDATE STATISTICS DAEMON

Update Statistics Daemon will do update statistics evenly. Different from update statistics command and schedule, users have to execute UPDATE STATISTICS SET on t2. If users want to do update statistics only on table t2, please refer the following steps:

1. dmconfig.ini setting

```
DB_STSVR=1      //activate update statistics daemon
DB_STMOD=1      //the update statistics execution will be decided by every table's mode and sample ratio
DB_STSSP=0      //the sample ratio of other tables will be 0
```

2. Execute UPDATE STATISTICS SET command to set t2's mode=1 and sample ratio=100.

```
UPDATE STATISTICS SET t2 MODE=1 SAMPLE=100;  //set t2's sample ratio=100, other table's mode=0 means their sample ratio=DB_STSSP=0, so they won't do update statistics
```

6.3. Update Statistics Takes a Long Time and Reduce Database Efficiency

This is a common situation: Users set DB_STSVR=1 activate update statistics daemon to do update statistics but their database have a very large table, the update statistics on this table will take a long time and reduce database efficiency. This situation sometimes causes database error or crash.

Solution 1:

If don't want to set mode and sample ratio individually, set DB_STSSP=-1 to smart decide every tables' sample ratio (do update statistics or not).

```
DB_STSVR=1      //activate update statistics daemon
DB_STMOD=0      //the daemon will do update statistics base on dmconfig.ini setting
DB_STSSP=-1     //the sample ratio of tables will be smart decided, mostly 0 and 100 (update statistics or not)
```

Solution 2:

Find the table causes slow operation, usually the biggest table, and ignore it while doing update statistics.

```
DB_STSVR=1      //activate update statistics daemon
DB_STMOD=1      //the daemon will do update statistics base on table setting
DB_STSSP=100    //only the biggest table is ignored, other tables still use sample ratio set in dmconfig.ini
```

Find the biggest table and execute UPDATE STATISTICS SET to ignore it

```
SELECT TABLE_NAME, NUM_PAGE FROM SYSTABLE ORDER BY NUM_PAGE;
//find the table has the biggest pages(size)
UPDATE STATISTICS SET table_name MODE=1 SAMPLE=0;
//set the biggest table's mode=1 sample ratio=0 to ignore it
```

6.4. Update Statistics Doesn't Finish Properly

If update statistics doesn't end properly (core dump or process is killed), this may cause the following commands failed. Users can execute the following command to solve this problem:

```
UPDATE STATISTICS ALL SAMPLE=0;
```