

ORACLE 11G DATABASE STATISTICS – STATISTICS PREFERENCES

Inderpal S. Johal, Data Softech Inc.

STATISTICS PREFERENCES

Oracle 11g Automated Maintenance Task is executed, it will look into the default value recommended by Oracle and shown below

Gather Optimizer Statistics Default Options

Estimate Percentage Auto (Oracle recommended) 100% Percentage

Degree of Parallelism Table default Auto System default Degree

Granularity ▼

Cursor Invalidation Auto (Oracle recommended) Immediate None

Cascade Auto (Oracle recommended) True False

Target Object Class (Auto Job) Auto (Oracle recommended) All Oracle

Stale Percentage

Incremental True False

Publish True False

Histograms

The default and recommended values are not always good and you might need to make some changes as per your environment. The change can be for single Database object or group of objects or specific schema. In prior releases, you can lock the Status for such objects so that automated job will skip them and then had to create a new job(s) to create the Statistics for all of such objects based on your requirements. With Oracle 11g, it is no longer required to lock the statistics as we can now set preferences at Schema or Table level which will instructs Auto Statistics Gathering Job to use the preferences. This way we don't need to create any extra Job. If you still don't want to collect statistics for any volatile Database object, you can continue to use LOCK Statistics feature so that Statistics is no longer collected for that object.

When Oracle is collecting the Optimizer Statistics, it will look into the following argument specified with DBMS_STATS package

- | | |
|--------------------|--|
| ▪ CASCADE | - Gather statistics on the indexes as well |
| ▪ DEGREE | - Degree of parallelism |
| ▪ ESTIMATE_PERCENT | - Percentage of rows to estimate (NULL means compute): |
| ▪ METHOD_OPT | |
| ▪ NO_INVALIDATE | - Does not invalidate the dependent cursors if set to TRUE |
| ▪ GRANULARITY | - Granularity of statistics to collect (for partitioned tables). |
| ▪ PUBLISH | - Collect Statistics info directly in DD or in Private area |
| ▪ INCREMENTAL | - Pertaining to Partition tables global Statistics information |
| ▪ STALE_PERCENT | - When the Statistics is considered outdated, default to 10 |

Please Note that the arguments in BLUE above are new in 11g.

The figure below shows the various procedure available for Database Statistics collection

11^g DBMS_STATS

Gathering Optimizer Statistics

- GATHER_DATABASE_STATS Procedures
- GATHER_DICTIONARY_STATS Procedure
- GATHER_FIXED_OBJECTS_STATS Procedure
- GATHER_INDEX_STATS Procedure
- GATHER_SCHEMA_STATS Procedures
- GATHER_SYSTEM_STATS Procedure
- GATHER_TABLE_STATS Procedure



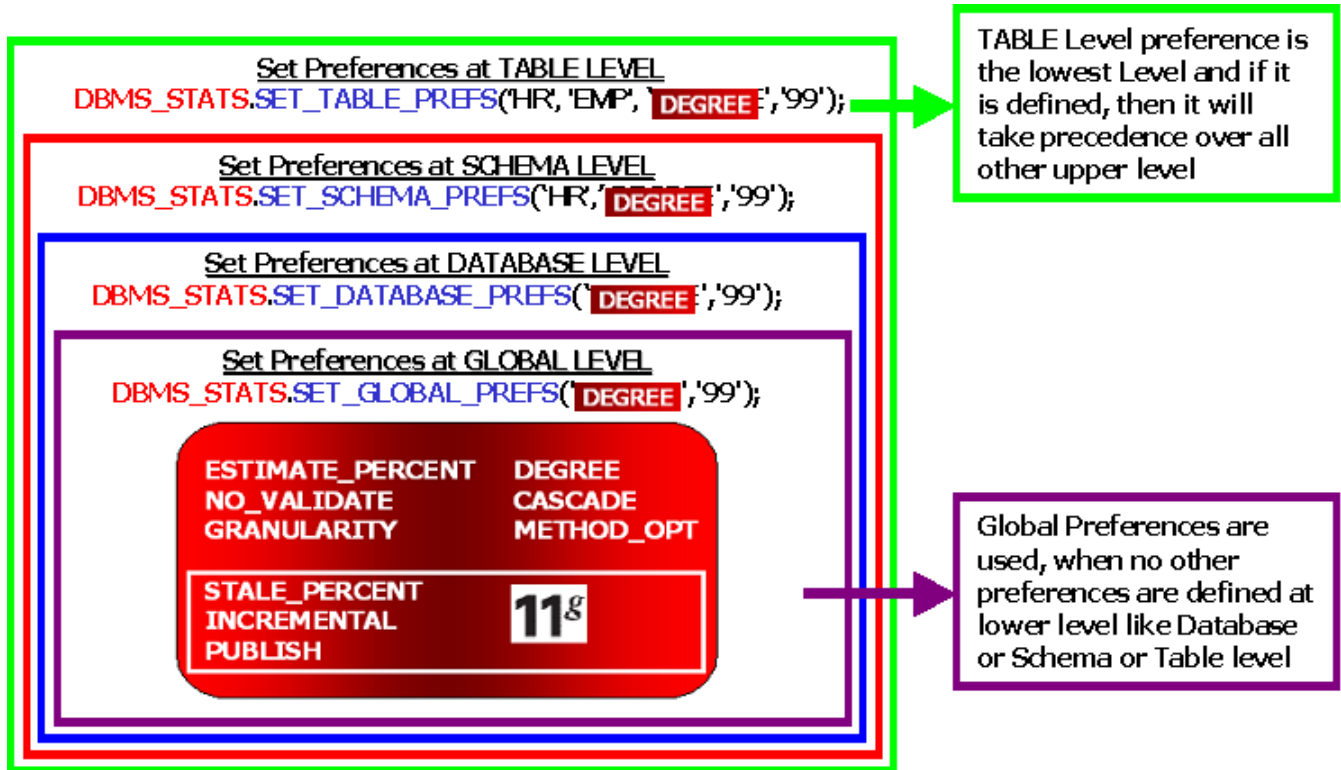
ESTIMATE_PERCENT	DEGREE
NO_VALIDATE	CASCADE
GRANULARITY	METHOD_OPT
STALE_PERCENT	11 ^g
INCREMENTAL	
PUBLISH	

PREFERENCES SETTING PROCESS

We can alter the default option used by Auto Statistics Gathering Job using the following procedure of DBMS_STATS package

1. SET_GLOBAL_PREFS
2. SET_DATABASE_PREFS
3. SET_SCHEMA_PREFS
4. SET_TABLE_PREFS

Below are some of the examples showing the usage of each procedure



Please Note that if you set any Preferences at Database level, then it will be applied to all objects in the database immediately and so next Maintenance Job will take the new Preference setting. While setting preferences at Global Level is only applied to new Object or if there is no other preference available for the table to be analyzed.

You need SYDBA privileges to define Preferences at Global or Database Level.

If you specify the Preference Value as NULL then it will set the Oracle default value.

PREFERENCES RETRIEVAL PROCESS

You can retrieve these preferences setting using any of the following two method

1. GET_PREFS function of DBMS_STATS package
2. DBA_TAB_STAT_PREFS view

GET_PREFS functions usages

```
SQL> SELECT DBMS_STATS.GET_PREFS(' DEGREE' , 'HR', 'EMP')
FROM DUAL;
```

Please Note that if Owner and Tablename is not provided with GET_PREFS function, then the function returns the Global Preferences (if defined) or the default value.

DBA_TAB_STAT_PREFS view

```
SQL> desc dba_tab_stat_prefs
```

Name	Null?	Type
OWNER	NOT NULL	VARCHAR2 (30)
TABLE_NAME	NOT NULL	VARCHAR2 (30)
PREFERENCE_NAME		VARCHAR2 (30)
PREFERENCE_VALUE		VARCHAR2 (1000)

PREFERENCES DELETE PROCESS

You can delete these preferences setting using the following procedure of DBMS_STATS package

1. DELETE_DATABASE_PREFS
2. DELETE_SCHEMA_PREFS
3. DELETE_TABLE_PREFS

Examples

```
DBMS_STATS.DELETE_DATABASE_PREFS ('DEGREE', TRUE);  
DBMS_STATS.DELETE_SCHEMA_PREFS ('SCOTT', 'DEGREE');  
DBMS_STATS.DELETE_TABLE_PREFS ('SCOTT', 'EMP', 'DEGREE');
```

PREFERENCES EXPORT/IMPORT PROCESS

You can Export/Import these preferences so as to move from one server to another. Oracle does not support export or import of statistics across databases of different character sets.

EXPORT_[DATABASE | SCHEMA | TABLE]_PREFS

IMPORT_[DATABASE | SCHEMA | TABLE]_PREFS

Examples

```
DBMS_STATS.EXPORT_DATABASE_PREFS('STATTAB', statown=>'SCOTT');  
DBMS_STATS.IMPORT_DATABASE_PREFS('STATTAB', statown=>'SCOTT');  
DBMS_STATS.IMPORT_TABLE_PREFS('SCOTT', 'EMP', 'STATTAB');
```

CHANGING GLOBAL AND TABLE STATISTIC PREFERENCES

Let's take an example as how we will use the statistics preference in our environment. I will use the parameter STALE_PERCENT in my example. This parameter determines the percentage of rows in a table that have to change before its statistics are considered stale and available to be analyzed again during the next Automatic Maintenance task defined by Oracle.

By Default STALE_PERCENT is defined as 10

Check the setting for STALE_PERCENT for SCOTT.EMP and SCOTT.SALARY

```
SQL> select dbms_stats.get_prefs('STALE_PERCENT', 'SCOTT', 'EMP') stale_percent
      from dual;
```

```
STALE_PERCENT
-----
```

```
10
```

```
SQL> select dbms_stats.get_prefs('STALE_PERCENT', 'SCOTT', 'SALARY') stale_percent
      from dual;
```

```
STALE_PERCENT
-----
```

```
10
```

Check the Global setting for STALE_PERCENT

```
SQL> select dbms_stats.get_prefs('STALE_PERCENT') stale_percent
      from dual;
```

```
STALE_PERCENT
-----
```

```
10
```

Change the STALE_PERCENT to 20 for SCOTT.EMP using SET_TABLE_PREFS procedure

```
SQL> exec dbms_stats.set_table_prefs('SCOTT','EMP','STALE_PERCENT','20');
```

PL/SQL procedure successfully completed.

```
SQL> select dbms_stats.get_prefs('STALE_PERCENT', 'SCOTT', 'EMP') stale_percent
      from dual;
```

```
STALE_PERCENT
-----
```

```
20
```

Now we have changed the Table preference for SCOTT.EMP to 20 while SCOTT.SALARY remain the default value means 10. Let's Change the Global preferences for STALE_PERCENT to 25

```
SQL> execute dbms_stats.set_global_prefs('STALE_PERCENT', '25');
```

PL/SQL procedure successfully completed.

Check the STALE_PERCENT value for SCOTT.SALARY and found that it is changed from 10 to 25 as there is no Table level preferences defined for SCOTT.SALARY

```
SQL> select dbms_stats.get_prefs('STALE_PERCENT', 'SCOTT', 'SALARY') stale_percent
      from dual;
```

```
STALE_PERCENT
-----
```

```
25
```

Now Let's define the Database Level Preferences and change the STALE_PERCENT to 30. This will immediately affect all objects in the database that currently have some preference defined at Schema or object level. If there is no Preference defined, then it will remain to have Global preference setting

```
SQL> exec dbms_stats.set_database_prefs('STALE_PERCENT','30');
```

PL/SQL procedure successfully completed.

```
SQL> select dbms_stats.get_prefs('STALE_PERCENT', 'SCOTT', 'EMP') stale_percent
      from dual;
```

```
STALE_PERCENT
-----
```

```
30
```

```
SQL> select dbms_stats.get_prefs('STALE_PERCENT', 'SH', 'SALARY') stale_percent
      from dual;
```

```
STALE_PERCENT
-----
```

```
25
```

```
SQL> select dbms_stats.get_prefs('STALE_PERCENT') stale_percent
      from dual;
```

```
STALE_PERCENT
-----
```

```
25
```

Now create the new table and see what kind of STALE_PERCENT is attached to it. It will get the global Preferences.

```
SQL> create table SH.test(name varchar2(10));
```

Table created.

```
SQL> select dbms_stats.get_prefs('STALE_PERCENT', 'SH', 'TEST') stale_percent
      from dual;
```

```
STALE_PERCENT
-----
```

```
25
```

Now delete the preference for SCOTT.EMP to null and then check the value again. It will be changed to default Database value for STALE_PERCENT of 10.

```
SQL> select dbms_stats.delete_table_prefs('SH', 'TEST') stale_percent from dual;
```

PL/SQL procedure successfully completed.


```
SQL> select dbms_stats.get_prefs('STALE_PERCENT', 'SH', 'TEST') stale_percent from dual;
STALE_PERCENT
```

```
-----
10
```

Reset the Global Preference Setting for STALE_PERCENT to default value like 10

```
SQL> execute dbms_stats.set_global_prefs('STALE_PERCENT', null);
PL/SQL procedure successfully completed.
```

```
SQL> select dbms_stats.get_prefs('STALE_PERCENT') stale_percent from dual;
STALE_PERCENT
```

```
-----
10
```

Now check the effect of GLOBAL vs DATABASE level Preferences

```
SQL> execute dbms_stats.set_global_prefs('STALE_PERCENT', '15');
PL/SQL procedure successfully completed.
```

```
SQL> exec dbms_stats.set_database_prefs('STALE_PERCENT','33');
PL/SQL procedure successfully completed.
```

```
SQL> select dbms_stats.get_prefs('STALE_PERCENT', 'SCOTT', 'SALARY') stale_percent
       from dual;
STALE_PERCENT
```

```
-----
15      →Because there is no preference set for this table
```

```
SQL> select dbms_stats.get_prefs('STALE_PERCENT', 'SH', 'TEST') stale_percent
       from dual;
STALE_PERCENT
```

```
-----
33      →Because there is preference set for this table and so is replaced Database setting
```

```
SQL> create table sh.test1 (name varchar2(10));
Table created.
```

```
SQL> select dbms_stats.get_prefs('STALE_PERCENT', 'SH', 'TEST1') stale_percent
       from dual;
STALE_PERCENT
```

```
-----
15      →Because there is no preference set for this table
```

```
SQL> exec dbms_stats.set_table_prefs('SH','TEST1','STALE_PERCENT',null);
PL/SQL procedure successfully completed.
```

```
SQL> select dbms_stats.get_prefs('STALE_PERCENT', 'SH', 'TEST1') stale_percent from dual;
STALE_PERCENT
```

```
-----
10      →Because it will take the Factory set Default value
```