Contenido

[1. Introducción 2](#_Toc518063491)

[3. Optimizer Statistics Preferences 4](#_Toc518063492)

[4. Gathering Optimizer Statistics Manually 6](#_Toc518063493)

[5. Referencias 7](#_Toc518063494)

# Introducción

"You don't necessarily need up to date statistics. You need statistics that are representative of your data." Graham Wood.

Meaning, the age of the statistics in your system is not a problem as long as they are still representative of your data. So just looking at the LAST\_ANALYZED column of the DBA\_TABLES view is not an indication of valid stats on your system.

"Do you want the optimizer to give you the best performance, or consistent performance?" Anjo Kolk

Meaning, regularly changing your stats potentially introduces change. Change is not always a good thing.

Neither of these experts are suggesting you never update your stats, just pointing out that in doing so you are altering information the optimizer uses to determine which execution plan is the most efficient. In altering that information it is not unlikely the optimizer may make a different decision. Hopefully it will be the correct decision, but maybe it wont. If you gather statistics for all tables every night, your system will potentially act differently every day. This is the fundamental paradox of gathering statistics.

So what should our statistics strategy be? Here are some suggestions.

Automatic Optimizer Statistics Collection: From 10g onward the database automatically gathers statistics on a daily basis. The default statistics job has come under a lot of criticism over the years, but its value depends on the type of systems you are managing. Most of that criticism has come from people discussing edge cases, like large data warehouses. If you are managing lots of small databases that have relatively modest performance requirements, you can pretty much let Oracle do its own thing where stats are concerned. If you have any specific problems, deal with them on a case by case basis.

Mixed Approach: You rely on the automatic job for the majority of stats collection, but you have specific tables or schemas that have very specific stats requirements. In these cases you can either set the preferences for the objects in question, or lock the stats for the specific tables/schemas to prevent the job from changing them, then devise a custom solution for those tables/schemas.

Manual: You disable the automatic stats collection completely and devise a custom solution for the whole of the database.

Which one of these approaches you take should be decided on a casebycase basis. Whichever route you take, you will be using the DBMS\_STATS package to manage your stats.

Regardless of the approach you take, you need to consider system and fixed object statistics for every database, as these are not gathered by the automatic job.

1. Configuring Automatic Optimizer Statistics Collection

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| |  | | --- | | **To Enable Automatic Job** | | begin  dbms\_auto\_task\_admin.enable (  client\_name => 'auto optimizer stats collection'  ,operation => null  ,window\_name => null  );  end;  / | | |  | | --- | | **To Disable Automatic Job** | | begin  dbms\_auto\_task\_admin.disable (  client\_name => 'auto optimizer stats collection'  ,operation => null  ,window\_name => null  );  end;  / | |
| |  | | --- | | **To Confirm the Change** | | COL client\_name FORMAT a31  SELECT client\_name, status  FROM dba\_autotask\_client  WHERE client\_name = 'auto optimizer stats collection'  ; | | |  | | --- | | **To Change Window Attributes** | | begin  dbms\_scheduler.set\_attribute (  name => 'MONDAY\_WINDOW'  ,attribute => 'repeat\_interval'  ,value => 'freq=daily;byday=mon;byhour=05;byminute=0;bysecond=0'  );  end;  / | |
| |  | | --- | | **To Configure a Specific Window for Statistics** | | begin  Window creation  dbms\_scheduler.create\_window (  window\_name => 'early\_morning\_window'  ,duration => numtodsinterval(1, 'hour')  ,resource\_plan => 'default\_maintenance\_plan'  ,repeat\_interval => 'freq=daily; byhour=5; byminute=0; bysecond=0'  ) ;  To add the new Window to maintenance group  dbms\_scheduler.add\_group\_member (  group\_name => 'maintenance\_window\_group'  ,member => 'early\_morning\_window'  );  end ;  / | | |  | | --- | | **To Remove a Window for Maintenance Group** | | begin  dbms\_scheduler.remove\_group\_member (  group\_name => 'maintenance\_window\_group'  ,member => 'early\_morning\_window'  );  The scheduler window still exists but it wont execute any maintenance task  end;  / | |

# Optimizer Statistics Preferences

SQL> SELECT DBMS\_STATS.GET\_PREFS

('incremental', 'sh','costs')

AS "STAT\_PREFS" FROM DUAL;

STAT\_PREFS

TRUE

SQL> EXEC DBMS\_STATS.SET\_TABLE\_PREFS

('sh', 'costs', 'incremental', 'false');

PL/SQL procedure successfully completed.

SQL> SELECT DBMS\_STATS.GET\_PREFS

('incremental', 'sh', 'costs')

AS "STAT\_PREFS" FROM DUAL;

STAT\_PREFS

FALSE

SQL> EXEC DBMS\_STATS.SET\_SCHEMA\_PREFS

('sh', 'incremental', 'true');

PL/SQL procedure successfully completed.

SQL> SELECT DBMS\_STATS.GET\_PREFS

('incremental', 'sh', 'costs')

AS "STAT\_PREFS" FROM DUAL;

STAT\_PREFS

TRUE

SQL> EXEC DBMS\_STATS.SET\_DATABASE\_PREFS

('incremental', 'false');

PL/SQL procedure successfully completed.

SQL> SELECT DBMS\_STATS.GET\_PREFS

('incremental', 'sh', 'costs')

AS "STAT\_PREFS" FROM DUAL;

STAT\_PREFS

FALSE

SELECT DBMS\_STATS.GET\_PREFS('STALE\_PERCENT', 'SH', 'SALES')

FROM DUAL;

EXEC DBMS\_STATS.SET\_TABLE\_PREFS('SH', 'SALES', 'STALE\_PERCENT', '13');

COL OWNER FORMAT a5

COL TABLE\_NAME FORMAT a15

COL PREFERENCE\_NAME FORMAT a20

COL PREFERENCE\_VALUE FORMAT a30

SELECT \* FROM DBA\_TAB\_STAT\_PREFS;

# Gathering Optimizer Statistics Manually

|  |  |  |  |
| --- | --- | --- | --- |
|  | |  | |
| |  | | --- | | **Dictionary** | | begin  dbms\_stats.gather\_dictionary\_stats;  end;  / | | **System Level** | | begin  To initiate the ystem statistics collection  dbms\_stats.gather\_system\_stats(  gathering\_mode => 'start'  );  dbms\_stats.gather\_system\_stats(  gathering\_mode => 'stop'  );  end;  /  begin  dbms\_stats.gather\_system\_stats(  gathering\_mode => 'interval'  ,interval => <minutes>  ) ;  end;  /  SELECT pname, pval1  FROM sys.aux\_stats$  WHERE sname = 'SYSSTATS\_MAIN'; | | | |  | | --- | | **Fixed Objects (X$ tables)** | | begin  DBMS\_STATS.gather\_fixed\_objects\_stats;  end;  / | | **Database Level** | | begin  dbms\_stats.gather\_database\_stats(  options => 'gather auto'  );  end;  / | | **Schema Level** | | begin  dbms\_stats.gather\_schema\_stats (  options => 'gather auto'  ,ownname => '<schema\_name>'  ,block\_sample => true  );  end;  / | | |
| |  | | --- | | **Table Level** | | begin  dbms\_stats.gather\_table\_stats (  ownname => '<schema\_name>'  ,tabname => '<table\_name>'  ,block\_sample => true  );  end;  / | | | |  | | --- | | **Table Partition Level** | | begin  dbms\_stats.gather\_table\_stats (  ownname => '<schema\_name>'  ,tabname => '<table\_name>'  ,partname => '<partition\_name>'  ,block\_sample => true  );  end;  / | | |
| |  | | --- | | **Index Level** | | begin  dbms\_stats.gather\_index\_stats (  ownname => '<schema\_name>'  ,indname => '<index\_name>'  ) ;  end;  / | | | |  | | --- | | **Index Partition Level** | | begin  dbms\_stats.gather\_index\_stats (  ownname => '<schame\_name>'  ,indname => '<index\_name>'  ,partname => '<partition\_name>'  ) ;  end;  / | | |
| **options**  GATHER\*, GATHER AUTO, GATHER EMPTY, GATHER STALE, LIST STALE, LIST EMPTY |  | |  |
| **estimate\_percent**  DBMS\_STATS.AUTO\_SAMPLE\_SIZE\*, The valid range is [0.000001,100] | **degree**  DBMS\_STATS.AUTO\_DEGREE\*, FROM 1 TO NUMBER OF CPUs | | **granularity** (For partitioned tables)  AUTO\*, ALL, GLOBAL, GLOBAL AND PARTITION, PARTITION, SUBPARTITION |
| **cascade** (To include index statistics)  TRUE\*, FALSE | **method\_opt**  FOR ALL COLUMNS SIZE AUTO\*, | | **no\_invalidate** (For cursors invalidation)  DBMS\_STATS.AUTO\_INVALIDATE\*, TRUE, FALSE |

# Referencias

https://oraclebase.com/articles/misc/costbasedoptimizeranddatabasestatistics