TABLES

STATOPTIONS

This table contains the options that will be used when gathering new statistics.
Defaults for these options can be set on different levels.

When new statistics need to be gathered for an object, the analyzethis procedure will try to find a matching record in this table and if no exact match can be found then it will look for a record that matches the object as close as possible.
For a table/index subpartition it will use the following order:

1. Same owner, object_type, object_name, partition_name and subpartition name.
2. Same owner, object_type, object_name and partition_name fields and the subpartition field is '*'
3. Same owner, object_type, object_name and partition_name fields and where the subpartition field is null.
4. Same owner, object_type and object_name fields and where the partition field is '*' and the subpartition field is null
5. Same owner, object_type and object_name fields and where the partition and subpartition fields are null.
6. Same owner and object_type fields and where the object_name, partition and subpartition fields are null
7. Same object_type field and the owner, object_name, partition and subpartition fields are null.
8. Use buildin defaults

For a table/index partition it will use the following order:

1. Same owner, object_type, object_name and partition_name fields and subpartition name field is null
2. Same owner, object_type and object_name fields and where the partition field is '*' and the subpartition field is null
3. Same owner, object_type and object_name fields and where the partition and subpartition fields are null.
4. Same owner and object_type fields and where the object_name, partition and subpartition fields are null
5. Same object_type field and the owner, object_name, partition and subpartition fields are null.
6. Use buildin defaults

For a table / index it will use the following order

1. Same owner, object_type and object_name fields and the partition_name and subpartition name fields are null
2. Same owner and object_type fields and where the object_name, partition and subpartition fields are null

3. Same object_type field and the owner, object_name, partition and subpartition fields are null.
4. Use buildin defaults

Column	Datatype	Null	Default	Description
owner	varchar2(30)			owner of the object
object_name	varchar2(30)			object name
part_name	varchar2(30)			partition name use “*” to set a default for all the partitions of the above table or index
subpart_name	varchar2(30)			subpartition name use “*” to set a default for all the subpartitions of the above partition.
object_type	varchar2(6)	not null		type of the object (TABLE | INDEX)
locked	varchar2(5)		‘FALSE’	exclude this object from new statistics gathering (TRUE | FALSE)
auto_sample_size	varchar2(5)		‘TRUE’	when set to TRUE, oracle will determine the appropriate sample size for good statistics. (TRUE | FALSE
estimate_percent	number(9,6)		NULL	percentage of rows to estimate, NULL means compute when auto_sample_size is set to true, it overrules estimate_percent
block_sample	varchar2(5)		‘FALSE’	Whether or not to use random block sampling instead of random row sampling. (TRUE | FALSE)
method_opt	varchar2(250)		‘FOR ALL COLUMNS SIZE AUTO’	Accepts: • FOR ALL [INDEXED | HIDDEN] COLUMNS [size_clause] • FOR COLUMNS [size clause] column|attribute [size_clause] [,column|attribute [size_clause]...]

			size_clause is defined as size_clause := SIZE {integer | REPEAT | AUTO | SKEWONLY} - integer : Number of histogram buckets. Must be in the range [1,254]. - REPEAT : Collects histograms only on the columns that already have histograms. - AUTO : Oracle determines the columns to collect histograms based on data distribution and the workload of the columns. - SKEWONLY : Oracle determines the columns to collect histograms based on the data distribution of the columns.
default_degree	varchar2(5)	‘AUTO’	Let oracle determine which degree level is to be used. DEFAULT: base the degree level on the initialization parameters and the number of cpu’s AUTO: let oracle decide (based on the size of the object) if to use serial execution (degree of 1) or use DEFAULT_DEGREE
degree	number(2)	NULL	Degree of parallelism Null means to use use the table default value specified by the degree clause in the create/alter table/index statement. When default degree is set to ‘AUTO’ or ‘DEFAULT’ then the degree field is ignored.
granularity	varchar2(12)	‘AUTO	Granularity of statistics to collect (only pertinent if the table is partitioned). 'ALL' - gathers all (subpartition, partition, and global) statistics 'AUTO'- determines the granularity based on the partitioning type. This is the default value. 'DEFAULT' - gathers global and partition-level statistics. This option is obsolete, and while currently supported, it is included in the documentation for legacy reasons only. You should use the 'GLOBAL AND PARTITION' for this functionality. Note that the default value is now 'AUTO'.

			'GLOBAL' - gathers global statistics 'GLOBAL AND PARTITION' - gathers the global and partition level statistics. No subpartition level statistics are gathered even if it is a composite partitioned object. 'PARTITION '- gathers partition-level statistics 'SUBPARTITION' - gathers subpartition-level statistics.
cascade	varchar2(5)	‘AUTO’	Gather statistics on the index(es) on this table as well. When set to AUTO, oracle will decide if it will gather statistics on the indexes or not (TRUE | FALSE | CASCADE)
no_invalidate	varchar2(5)	‘AUTO’	Whether or not to invalidate the dependent cursors. TRUE: do not invalidate the dependent cursors FALSE: invalidate the dependent cursors immediately AUTO: let oracle decide when to invalidate dependent cursors

RUN_HISTORY

Displays analyzethis runs.

Column	Datatype	Null	Default	Description
statid	varchar2(11)	not null		unique id
starttime	date			start time of the operation
endtime	date			end time of the operation
operation	varchar2(250)			Description
calcfailures	number(10,0))	not null		number of objects for which the statistics calculation failed only applicable for operations involving new statistics gathering.
expfailures	number(10,0)		‘FALSE’	number of objects for which the new statistics could not be exported. Is also set to 1 if a backup schema operation failes only applicable for operations involving new statistics gathering or schema backups.
protected	varchar2(5)		‘TRUE’	protect this record against deletion. When set to true this record and the related records in the run_history_details and the statistics_history table will not be deleted when using the analyzethis.purgehistory procedure (unless the p_force parameter is set to true)

RUN_HISTORY_DETAILS

Displays details of analyzethis runs

Column	Datatype	Null	Default	Description
statid	varchar2(11)			id of the statistics calculation run
starttime	date			Start of the statistics calculation run
stoptime	date			End time of the statistics calculation run
calc_status	varchar2(250)			Result status of the statistics calculation
exp_status	varchar2(250)			Result status of the export of the newly calculated statistics
ownname	varchar2(30)	not null		Owner of the object for which new statistics where calculated
objtype	varchar2(6)	not null		Type of the object for which new statistics where calculated
objname	varchar2(30)	not null		Name of the object for which new statistics where calculated
partname	varchar2(30)			Partition name of the object for which new statistics where calculated
subpartname	varchar2(30)			Subpartition name of the object for which new statistics where calculated
sto_ownname	varchar2(30)			owner field for the record in the statoptions table that was used as source for the statistics gathering options.
sto_objtype	varchar2(6)			object_type field for the record in the statoptions table that was used as source for the statistics gathering options.
sto_objname	varchar2(30)			object_name field for the record in the statoptions table that was used as source for the statistics gathering options.
sto_partname	varchar2(30)			part_name field for the record in the statoptions table that was used as source for the statistics gathering options.
sto_subpartname	varchar2(30)			subpart_name field for the record in the statoptions table that was used as source for the statistics gathering options.
sto_locked	varchar2(5)			locked option value that was used
sto_auto_sample_size	varchar2(5)			auto_sample_size value that was used

sto_estimate_percent	number(9,6)	estimate_percent value that was used
sto_block_sample	varchar2(5)	block sample value that was ued
sto_method_opt	varchar2(250)	method_opt value that was used
sto_default_degree	varchar2(5)	default_degree value that was used
sto_degree	number(2)	degree value that was used
sto_granularity	varchar2(12)	granularity value that was used
sto_cascade	varchar2(5)	cascade option that was used
sto_no_invalidate	varchar2(11)	no_invalidate option that was used

DEBUGTAB

This table is used for configuring debugging.
Use the procedures init, status and clear in the debug package to manage the data in this table.

Column	Datatype	Null	Default	Description
userid	varchar2(30)			
modules	varchar2(4000)			
locat	varchar2(4000)			
filename	varchar2(4000)			

PACKAGES

GETLASTANALYZEDDATE

Gets the last analyzed date for a given table or index and is used by the AnalyzeThis package.
It is separated from the AnalyzeThis package because it needed the select privilege on the dba_indexes, dba_ind_partitions, dba_ind_subpartitions, dba_tables, dba_tab_partitions and the dba_tab_subpartitions dictionary views.
If required it can be created in a different schema then the main AnalyzeThis package. In this case the owner of the main AnalyzeThis package must be granted execute privilege on the GetLastAnalyzedDate package and a public or private synonym must be created on the GetLastAnalyzedDate package.

Subprogram	Description
GETDATE Function	gets the last_analyzed date for the passed object

GETDATE Function

Syntax

```
FUNCTION getDate
    ( p_owner           IN      varchar2,
      p_object_type     IN      varchar2,
      p_object_name     IN      varchar2,
      p_part_name       IN      varchar2,
      p_subpart_name    IN      varchar2
    )
    RETURN date;
```

Parameters

Parameter	Description
p_owner	Owner of the objecte
p_object_type	Type of the object (TABLE | INDEX)
p_object_name	Name of the object.
p_part_name	Parttition name of the object
p_subpart_name	Subpartition name of the object

GETINDEXTABLE

Contains a procedure to get the table name and owner for a given index and is used by the AnalyzeThis package.
Is is separated from the AnalyzeThis package because it needed the select privilege on the dba_part_indexes, dba_indexes, dba_ind_partitions, dba_ind_subpartitions, dba_tables, dba_tab_partitions and the dba_tab_subpartitions dictionary views.

If required it can be created in a different schema then the main AnalyzeThis package. In this case the owner of the main AnalyzeThis package must be granted execute privilege on the GetIndexTable package and a public or private synonym must be created on the GetIndexTable package.

If the index is a (sub)-partition it checks if it is a global or local partitioned index.
In case of a global partitioned index it returns the table name, in case of a local partitioned index it return the table partition on which the index is active.

Subprogram	Description
GETTABLEINFO Procedure	gets the table_owner and table_name for a passed index

GETTABLEINFO Procedure

Syntax

```
PROCEDURE getTableInfo
    ( p_ind_owner           IN      varchar2,
      p_ind_name            IN      varchar2,
      p_ind_part_name       IN      varchar2    default null,
      p_ind_subpart_name    IN      varchar2    default null,
      p_tab_owner           OUT     varchar2,
      p_tab_name            OUT     varchar2,
      p_tab_part_name       OUT     varchar2,
      p_tab_subpart_name    OUT     varchar2
    );
```

Parameters

Parameter	Description
p_ind_owner	Owner of the index
p_ind_name	Index name
p_in_part_name	Index partition name default null
p_ind_subpart_name	Index subpartition name default null
p_tab_owner	Owner of the retrieved table
p_tab_name	Name of the retrieved table
p_tab_part_name	Partition name of the retrieved table
p_tab_subpart_name	Subpartition name of the retrieved table

ANALYZETHIS

This is the main package.
It contains the procedures to backup statistics, gather new statistics, list the options and maintain the history tables
Depends on the GetlastAnalyzedDate and GetIndexTable packages

Subprogram	Description
GATHERSCHEMASTATS Procedure	Gather statistics for a given schema.
GATHEROBJSTATS Procedure	Gather statistics for a given object.
GATHERSTATS Procedure	Calculates new statistics for all objects given in the passed reference cursor.
BACKUPSCHEMASTATS Procedure	Backup the existing statistics of a given schema
LISTOBJOPTIONS Function	Reports which options would be used when gathering statistics for a given object
NEEDSTATSLIST Function	Pipelined function that returns a list of objects that need statistics.

PURGEHISTORY Procedure	Deletes the records from the run_history_details, run_history and statistics_history tables
ABOUT Procedure	Displays an about message

GATHERSCHEMASTATS Procedure

Gather statistics for a given schema.
The options used in the internal calls to dbms_stats are taken from the statoptions table.
The startime, endtime and number of errors are stored in the run_history table and in run_history_details the starttime, endtime, status and used statistics options are stored. The new generated statistics are also put in statistics_history table

Syntax

```
PROCEDURE GatherSchemaStats
    ( p_schema      in      varchar2,
      p_backup      in      boolean     default TRUE,
      p_option      in      varchar2    default 'AUTO'
    );
```

Parameters

Parameter	Description
p_schema	Name of the schema to analyze
p_backup	Whether or not to backup the current statistics to the statistics_history table before calculating new ones The default is TRUE When new statistics are gathered for an object, these new statistics are always stored in the statistics_history table, independent of the p_backup setting
p_option	Further specification of which objects to gather statistics for The default is “AUTO”. Accepts • AUTO: Oracle implicitly determines which objects need new statistics • STALE: Gathers statistics on stale objects as determined by looking at the *_tab_modification views. • EMPTY: Gathers statistics on objects that currently have no statistics

When using STALE or EMPTY, the following parameters are used in the internal calls to dbms_gather_schema_stats when creating the list of objects for which new statistics will be gathered (note: these parameters are not used when actually gathering the statistics.

```
granularity   =>  'ALL',
cascade       =>  true,
force         =>  true,
```

When using the AUTO option, these parameters are ignored by Oracle.

GatherObjStats Procedure

Gather statistics for a given table or index.
The options used in the internal calls to dbms_stats are taken from the statoptions table.
The startime, endtime and number of errors are stored in the run_history table and in run_history_details the starttime, endtime, status and used statistics options are stored. The new generated statistics are also put in statistics_history table

Syntax

```
PROCEDURE GatherObjStats
    ( p_object_type     in      varchar2,
      p_owner           in      varchar2,
      p_object_name     in      varchar2,
      p_part_name       in      varchar2    default NULL,
      p_subpart_name    in      varchar2    default NULL,
      p_backup          in      boolean     default TRUE
    );
```

Parameters

Parameter	Description
p_object_type	Type of the object to analyze
p_owner	Owner of the object to analyze
p_object_name	Name of the object to analyze
p_part_name	partition name of the object to analyze defaults to NULL

p_subpart_name	sub-partition name of the object to analyze defaults to NULL
p_backup	Whether or not to backup the current statistics to the statistics_history table before calculating new ones. This backup is always on schema level, so when true, all objects in the same schema as the object passed to the procedure will be backupped The default is TRUE When new statistics are gathered for an object, these new statistics are always stored in the statistics_history table, independent of the p_backup setting

---

GatherObjStats Procedure

Calculates new statistics for all objects given in the passed reference cursor.
This function can be used to create statistics on custom object lists, when the GatherSchemaStats or the GatherObjStats procedures are not sufficient.

The passed cursor must be of the type analyzethis.rfc_statsinput.

Because some statistics options set for an object have an impact on other objects then the ones they are set for in the statoptions table (eg cascade or granularity) it could happen that an object (such as an index or a table (sub) partition) is analyzed multiple times. To avoid this, it is important that the objects to be analyzed are ordened in a specific way.

The GatherSchemaStats procedure itself is using the following order:

order by ownname, objtype desc, objname asc, partname asc nulls first, subpartname asc nulls first;

Syntax

```
PROCEDURE GatherStats
    ( p_obj_cursor   in out     rfc_statsinput,
      p_log_message  in         varchar2
    );
```

Parameter	Description
p_obj_cursor	Reference cursor containing the list of objects for which new statistics needs to be generated. The calling procedure is responsible for constructing the ref cursor and to close it afterwards
p_log_message	text that will be used as value for the operation field in the run_history table.

BackupSchemaStats Procedure

Backups the current statistics of a schema.

Syntax

```
PROCEDURE BackupSchemaStats
    ( p_schema          in      varchar2
    );
```

Parameter	Description
p_schema	name of the schema for which the current statistics must backupped

ListObjOptions Function

Retrieves the statistics options from the statoptions table that would be used when new statistics for this object are calculated.
This function can be used to verify the configuration in the statoptoins table, without actually gathering statistics.

Returns a clob containing the options in a report format

Syntax

```
FUNCTION ListObjOptions
    ( p_object_type     in      varchar2,
      p_owner           in      varchar2,
      p_object_name     in      varchar2,
      p_part_name       in      varchar2    default null,
      p_subpart_name    in      varchar2    default null
    )
    RETURN CLOB;
```

Parameters

Parameter	Description
p_object_type	Type of the object to generate the report for
p_owner	Owner of the object to generate the report for
p_object_name	Name of the object to generate the report for
p_part_name	partition name of the object to generate the report for defaults to NULL
p_subpart_name	sub-partition name of the object to generate the report for defaults to NULL

NeedStatsList Function

Pipelined function that returns a list of objects that need statistics.
It uses the dbms_stats.gather_schema_stats procedure, with the options parameter set to ‘LIST AUTO’, ‘LIST STALE’ or ‘LIST EMPTY’, to generate this list.
The other used parameters in the call to the dbms_stats.gather_schema_stats procedure are:

granularity: ALL
cascade: TRUE
force; TRUE

Note that when the option LIST AUTO is used, oracle ignores these passed values and decides itself which values to use.

Syntax

```
FUNCTION NeedStatsList
    ( p_schema      in      varchar2    default user,
      p_option      in      varchar2    default 'AUTO'
    )
    RETURN dbms_stats.objecttab PIPELINED;
```

Parameters

Parameter	Description
p_schema	Name of the schema for which the list needs to be generated
p_option	‘AUTO’, ‘STALE’ or ‘EMPTY’ default AUTO AUTO: Returns the objects having stale or empty statistics. Corresponds with using the ‘LIST AUTO’ value for the options parameter with dbms_stats.gather_schema_stats STALE: Returns the objects having stale statistics Corresponds with using the ‘LIST STALE’ value for the options parameter with dbms_stats.gather_schema_stats EMPTY: Returns the objects without statistics Corresponds with using the ‘LIST EMPTY’ value for the options parameter with dbms_stats.gather_schema_stats

PurgeHistory Procedure

This procedure deletes the records from the run_history_details, run_history and statistics_history table.

Syntax

```
PROCEDURE PurgeHistory
    ( p_days            in      number      default 30,
      p_with_operation  in      varchar2    default null,
      p_rh              in      boolean     default true,
      p_force           in      boolean     default false
    );
```

Parameters

Parameter	Description
p_days	records of which the run endtime is older then p_days will deleted default 30 days

p_with_operation	Puts an additional filter on the "operation" field (checks on the exact string, case sensitive) NULL means no filter default NULL
p_rh	FALSE: delete only the records from the run_history_details table TRUE: delete the records from the run_history, the run_history_details and the statistics history table default: true
p_force	FALSE: delete only the records for which the protected field is set to 'N' TRUE: do not check the value of the field "protected" default: false

ANALYZEDB

This package extends the functionality of the AnalyzeThis package with a procedure to gather statistics on the entire database.
It is separated from the AnalyzeThis package because it needed the select privilege on the dba_segments dictionary view.
If required, it can be removed or put in a different schema then the main AnalyzeThis package. In the second case this new owner must be granted the execute privilege on the AnalyzeThis package and a public or private synonym for the AnalyzeThis package must be created.

Subprogram	Description
GATHERDBSTATS Procedure	Gather statistics for the entire database, excluding sys, system and the schema's listed in dba_registry.

GATHERDBSTATS Procedure

Gather statistics for the intire database.
It gets a list of the schema's existing in the database, excluding sys, system and the other schema's listed in dba_registry, and then loops through this list calling analyzethis.gatherschemastats.

Syntax

```
PROCEDURE GatherDBStats
    ( p_backup      in      boolean     default TRUE,
      p_option      in      varchar2    default 'AUTO'
    );
```

Parameters

Parameter	Description
p_backup	Whether or not to backup the current statistics to the statistics_history table before calculating new ones The default is TRUE When new statistics are gathered for an object, these new statistics are always stored in the statistics_history table, independent of the p_backup setting

p_option	Further specification of which objects to gather statistics for
	The default is “AUTO”.
	
	Accepts
	
	• AUTO: Oracle implicitly determines which objects need new statistics
	• STALE: Gathers statistics on stale objects as determined by looking at the *_tab_modification views.
	• EMPTY: Gathers statistics on objects that currently have no statistics
	
	When using STALE or EMPTY, the following parameters are used in the internal calls to dbms_gather_schema_stats when creating the list of objects for which new statistics will be gathered (note: these parameters are not used when actually gathering the statistics.
	
	`granularity => 'ALL',`
	`cascade => true,`
	`force => true,`
	
	When using the AUTO option, these parameters are ignored by Oracle.