35 DBMS_AUTO_SQLTUNE

The DBMS_AUTO_SQLTUNE package is the interface for managing the Automatic SQL Tuning task. Unlike DBMS_SQLTUNE, the DBMS_AUTO_SQLTUNE package requires the DBA role.

The chapter contains the following topics:

DBMS_AUTO_SQLTUNE Overview

The DBMS_AUTO_SQLTUNE package is the interface to SQL Tuning Advisor (DBMS_SQLTUNE) when run within the AutoTask framework.

The database creates the automated system task SYS_AUTO_SQL_TUNING_TASK as part of the catalog scripts. This task automatically chooses a set of high-load SQL from AWR and runs SQL Tuning Advisor on this SQL. The automated task performs the same comprehensive analysis as any other SQL Tuning task.
The automated task tests any SQL profiles it finds by executing both the old and new query plans. Automatic SQL Tuning differs from manual SQL tuning in one important way. If automatic implementation of SQL profiles is enabled (the default is disabled), then the database implements any SQL profiles that promise a great performance benefit. The implementation occurs at tuning time so that the database can immediately benefit from the new plan. You can enable or disable automatic implementation by using the SET_AUTO_TUNING_TASK_PARAMETER API to set the ACCEPT_SQL_PROFILES parameter.

In each maintenance window, the automated tuning task stores its results as a new execution. Each execution result has the same task name but a different execution name. Query the DBA_ADVISOR_EXECUTIONS view for information about task executions. To view reports that span multiple executions, use the REPORT_AUTO_TUNING_TASK Function.

DBMS_AUTO_SQLTUNE Security Model

This package is available to users with the DBA role. For other users, you must grant the EXECUTE privilege on the package explicitly. Note that the EXECUTE_AUTO_TUNING_TASK procedure is an exception: only SYS can invoke it.

Users can call APIs in this package to control how the automatic tuning task behaves when it runs, such as enabling automatic SQL profile creation and configuring the total and per-SQL time limits under which the task runs. Because these settings affect the overall performance of the database, it may not be appropriate for all users with the ADVISOR privilege to have access to this package.

Summary of DBMS_AUTO_SQLTUNE Subprograms

The DBMS_AUTO_SQLTUNE package contains EXECUTE, REPORT, and SET subprograms.

Table 35-1 DBMS_AUTO_SQLTUNE Package Subprograms

Subprogram	Description
EXECUTE_AUTO_TUNING_TASK Function and Procedure	Executes the Automatic SQL Tuning task immediately (SYS only)
REPORT_AUTO_TUNING_TASK Function	Displays a text report of the automatic tuning task's history
SET_AUTO_TUNING_TASK_PARAMETER Procedures	Changes a task parameter value for the daily automatic runs

EXECUTE_AUTO_TUNING_TASK Function and Procedure

This function and procedure executes the Automatic SQL Tuning task (SYS_AUTO_SQL_TUNING_TASK).

Both the function and the procedure run in the context of a new task execution. The difference is that the function returns the name of the new execution.

Syntax

```
DBMS_AUTO_SQLTUNE.EXECUTE_AUTO_TUNING_TASK(
   execution_name     IN VARCHAR2                  := NULL,
   execution_params   IN DBMS_ADVISOR.argList      := NULL,
   execution_desc     IN VARCHAR2                  := NULL)
 RETURN VARCHAR2;

DBMS_AUTO_SQLTUNE.EXECUTE_AUTO_TUNING_TASK(
   execution_name     IN VARCHAR2                  := NULL,
   execution_params   IN DBMS_ADVISOR.argList      := NULL,
   execution_desc     IN VARCHAR2                  := NULL);
```

Parameters

Table 35-2 EXECUTE_AUTO_TUNING_TASK Function and Procedure Parameters

Parameter	Description
execution_name	A name to qualify and identify an execution. If not specified, it is generated by the advisor and returned by function.
execution_params	List of parameters (name, value) for the specified execution. The execution parameters have effect only on the execution for which they are specified. They override the values for the parameters stored in the task (set through the SET_AUTO_TUNING_TASK_PARAMETER Procedures).
execution_desc	A 256-length string describing the execution

Usage Notes

Only SYS can invoke this subprogram. A tuning task can be executed multiple times without having to reset it.

Examples

```
EXEC DBMS_AUTO_SQLTUNE.EXECUTE_AUTO_TUNING_TASK('SYS_AUTO_SQL_TUNING_TASK');
```

REPORT_AUTO_TUNING_TASK Function

This procedure displays the results of an Automatic SQL Tuning task.

Syntax

```
DBMS_AUTO_SQLTUNE.REPORT_AUTO_TUNING_TASK(
   begin_exec       IN    VARCHAR2    := NULL,
   end_exec         IN    VARCHAR2    := NULL,
   type             IN    VARCHAR2    := 'TEXT',
   level            IN    VARCHAR2    := 'TYPICAL',
   section          IN    VARCHAR2    := ALL,
   object_id        IN    NUMBER      := NULL,
   result_limit     IN    NUMBER      := NULL)
RETURN CLOB;
```

Parameters

Table 35-3 REPORT_AUTO_TUNING_TASK Function Parameters

Parameter	Description
begin_exec	Name of the beginning task execution to use. If NULL, the report is generated for the most recent task execution.
end_exec	Name of the ending task execution to use. If NULL, the report is generated for the most recent task execution.
type	Type of the report to produce. Possible values are TEXT which produces a text report.
level	Level of detail in the report: • BASIC: simple version of the report. Just show info about the actions taken by the advisor. • TYPICAL: show information about every statement analyzed, including requests not implemented. • ALL: highly detailed report level, also provides annotations about statements skipped over.
section	Section of the report to include: • SUMMARY: summary information • FINDINGS: tuning findings • PLAN: explain plans • INFORMATION: general information • ERROR: statements with errors • ALL: all sections
object_id	Advisor framework object id that represents a single statement to restrict reporting to. NULL for all statements. Only valid for reports that target a single execution.
result_limit	Maximum number of SQL statements to show in the report

Return Values

A CLOB containing the desired report.

Examples

```
-- Get the whole report for the most recent execution
SELECT DBMS_AUTO_SQLTUNE.REPORT_AUTO_TUNING_TASK
FROM   DUAL;

-- Show the summary for a range of executions
SELECT DBMS_AUTO_SQLTUNE.REPORT_AUTO_TUNING_TASK(:begin_exec, :end_exec,
'TEXT',
       'TYPICAL', 'SUMMARY')
FROM   DUAL;

-- Show the findings for the statement of interest
SELECT DBMS_AUTO_SQLTUNE.REPORT_AUTO_TUNING_TASK(:exec, :exec, 'TEXT',
       'TYPICAL', 'FINDINGS', 5)
FROM   DUAL;
```

SET_AUTO_TUNING_TASK_PARAMETER Procedures

This procedure updates the value of a SQL tuning parameter of type VARCHAR2 or NUMBER for SYS_AUTO_SQL_TUNING_TASK.

Syntax

```
DBMS_AUTO_SQLTUNE.SET_AUTO_TUNING_TASK_PARAMETER(
   parameter     IN   VARCHAR2,
   value         IN   VARCHAR2);

DBMS_AUTO_SQLTUNE.SET_AUTO_TUNING_TASK_PARAMETER(
   parameter     IN   VARCHAR2,
   value         IN   NUMBER);
```

Parameters

Table 35-4 SET_AUTO_TUNING_TASK_PARAMETER Procedure Parameters

Parameter	Description
parameter	Name of the parameter to set. The possible tuning parameters that can be set by this procedure using the parameter in the form VARCHAR2: • MODE: tuning scope (comprehensive, limited) • USERNAME: user name under which the statement is parsed • DAYS_TO_EXPIRE: number of days until the task is deleted • EXECUTION_DAYS_TO_EXPIRE: number of days until the task execution is deleted • DEFAULT_EXECUTION_TYPE: default execution when none is specified by the EXECUTE_AUTO_TUNING_TASK Function and Procedure • TIME_LIMIT: global timeout in seconds • LOCAL_TIME_LIMIT: per-statement timeout in seconds • TEST_EXECUTE: – FULL - test-execute for as much time as necessary, up to the local time limit for the SQL (or the global task time limit if no SQL time limit is set) – AUTO - test-execute for an automatically chosen time proportional to the tuning time – OFF - do not test-execute • BASIC_FILTER: basic filter for SQL tuning set • OBJECT_FILTER: object filter for SQL tuning set • PLAN_FILTER: plan filter for SQL tuning set (see SELECT_SQLSET for possible values) • RANK_MEASURE1: first ranking measure for SQL tuning set • RANK_MEASURE2: second ranking measure for SQL tuning set • RANK_MEASURE3: third ranking measure for SQL tuning set • RESUME_FILTER: extra filter for SQL tuning sets besides BASIC_FILTER • SQL_LIMIT: maximum number of SQL statements to tune • SQL_PERCENTAGE: percentage filter of SQL tuning set statements The following parameters are supported for the automatic tuning task only: • ACCEPT_SQL_PROFILES: whether the task should accept SQL profiles automatically (TRUE or FALSE) • MAX_AUTO_SQL_PROFILES: maximum number of automatic SQL profiles allowed on the system, in sum • MAX_SQL_PROFILES_PER_EXEC: maximum number of SQL profiles that can be automatically implemented per execution of the task.
value	New value of the specified parameter