189

DBMS_SQL_MONITOR

The DBMS_SQL_MONITOR package provides information about Real-Time SQL Monitoring and Real-Time Database Operation Monitoring.

This chapter contains the following topics:

- DBMS_SQL_MONITOR Overview
- DBMS_SQL_MONITOR Security Model
- DBMS_SQL_MONITOR Constants
- Summary of DBMS_SQL_MONITOR Subprograms

> **See Also:**
>
> DBMS_SQLTUNE

DBMS_SQL_MONITOR Overview

The DBMS_SQL_MONITOR package provides information about Real-Time SQL Monitoring and Real-Time Database Operation Monitoring.

These features provide automatic monitoring of SQL statements, PL/SQL blocks, or composite database operations that are considered high-cost. A simple database operation is a single SQL statement or PL/SQL procedure or function. A composite database operation is activity between two defined points in time in a database session. The monitored data is collected in the V$SQL_MONITOR and V$SQL_PLAN_MONITOR views.

The following subprograms begin and end monitoring of a composite database operation:

- BEGIN_OPERATION Function
- END_OPERATION Procedure

The following subprograms report on monitoring data collected in V$SQL_MONITOR and V$SQL_PLAN_MONITOR:

- REPORT_SQL_MONITOR Function
- REPORT_SQL_MONITOR_XML Function
- REPORT_SQL_MONITOR_LIST Function
- REPORT_SQL_MONITOR_LIST_XML Function

DBMS_SQL_MONITOR Security Model

This package is available to PUBLIC and executes with invoker's rights privileges. The reporting functions require privileges to select data from the catalog as provided by the role SELECT_CATALOG_ROLE.

DBMS_SQL_MONITOR Constants

The DBMS_SQL_MONITOR package uses the constants shown in the following table.

Table 189-1 DBMS_SQL_MONITOR Constants

Constant	Type	Value	Description
FORCE_TRACKING	VARCHAR2(30)	'Y'	Force track the composite database operation when the operation starts
NO_FORCE_TRACKING	VARCHAR2(30)	'N'	Do not force track the composite database operation when the operation starts. It is only tracked when it has consumed 5 seconds of CPU or I/O time.

Summary of DBMS_SQL_MONITOR Subprograms

This table lists and describes the DBMS_SQL_MONITOR package subprograms.

Table 189-2 DBMS_SQL_MONITOR Package Subprograms

Subprogram	Description
BEGIN_OPERATION Function	This function starts a database operation in the current session.
END_OPERATION Procedure	This function ends a database operation in the current session. If the specified database operation does not exist, then this function has no effect.
REPORT_SQL_MONITOR Function	This function builds a detailed report with monitoring information for a SQL statement, PL/SQL block, or database operation.
REPORT_SQL_MONITOR_XML Function	This function is identical to the REPORT_SQL_MONITOR function, except that the return type is XMLType.
REPORT_SQL_MONITOR_LIST Function	This function builds a report for all or a subset of database operations that have been monitored by Oracle Database.
REPORT_SQL_MONITOR_LIST_XML Function	This function is identical to the REPORT_SQL_MONITOR_LIST function, except that it returns XMLType.

BEGIN_OPERATION Function

This function starts a database operation in the current session.

Syntax

```
DBMS_SQL_MONITOR.BEGIN_OPERATION (
   dbop_name       IN VARCHAR2,
```

```
   dbop_eid        IN NUMBER   := NULL,
   forced_tracking IN VARCHAR2 := NO_FORCE_TRACKING,
   attribute_list  IN VARCHAR2 := NULL,
   session_id      IN NUMBER   := NULL,
   session_serial  IN NUMBER   := NULL)
  iRETURN NUMBER;
```

Parameters

Table 189-3 *BEGIN_OPERATION Procedure Parameters*

Parameter	Description
dbop_name	Name for the composite database operation.
dbop_eid	Unique identifier for the current execution of the composite database operation.
forced_tracking	Whether tracking is forced. Possible values are: • FORCE_TRACKING - forces the composite database operation to be tracked when the operation starts. You can also use the string variable Y. • NO_FORCE_TRACKING - tracks the operation only when it has consumed at least 5 seconds of CPU or I/O time. You can also use the string variable N. See "DBMS_SQL_MONITOR Constants".
attribute_list	List of user-created attributes. It is a comma-separated list of name-value pairs (for example, 'table_name=emp, operation=load').
session_id	Session ID of the session to be monitored. If omitted (or null), then the database monitors the current session.
session_serial	Serial number of the session to be monitored. If omitted (or null), then the database uses only the session ID to determine the session.

Return Values

This function returns the database operation execution ID. If the value is null for dbop_eid, then the database generates a unique value.

END_OPERATION Procedure

This procedure ends a database operation in the current session. If the specified database operation does not exist, then this function has no effect.

Syntax

```
DBMS_SQL_MONITOR.END_OPERATION(
   dbop_name       IN VARCHAR2,
   dbop_eid        IN NUMBER);
```

Parameters

Table 189-4 END_OPERATION Procedure Parameters

Parameter	Description
dbop_name	Name of a composite database operation
dbop_eid	Unique identifier for the current execution of the composite database operation

REPORT_SQL_MONITOR Function

This function builds a detailed report with monitoring information for a SQL statement, PL/SQL block, or database operation.

For each operation, it gives key information and associated global statistics. Use this function to get detailed monitoring information for a database operation.

The target database operation for this report can be:

- The last database operation monitored by Oracle Database (default, no parameter).
- The last database operation executed in the specified session and monitored by Oracle Database. The session is identified by its session ID and optionally its serial number (`-1` is current session).
- The last execution of a specific database operation identified by its `sql_id`.
- A specific execution of a database operation identified by the combination `sql_id`, `sql_exec_start`, and `sql_exec_id`.
- The last execution of a specific database operation identified by `dbop_name`.
- The specific execution of a database operation identified by the combination `dbop_name`, `dbop_exec_id`.

Syntax

```
DBMS_SQL_MONITOR.REPORT_SQL_MONITOR (
   sql_id                    IN VARCHAR2 DEFAULT  NULL,
   dbop_name                 IN VARCHAR2 DEFAULT  NULL,
   dbop_exec_id              IN NUMBER   DEFAULT  NULL,
   session_id                IN NUMBER   DEFAULT  NULL,
   session_serial            IN NUMBER   DEFAULT  NULL,
   sql_exec_start            IN DATE     DEFAULT  NULL,
   sql_exec_id               IN NUMBER   DEFAULT  NULL,
   inst_id                   IN NUMBER   DEFAULT  NULL,
   start_time_filter         IN DATE     DEFAULT  NULL,
   end_time_filter           IN DATE     DEFAULT  NULL,
   instance_id_filter        IN NUMBER   DEFAULT  NULL,
   parallel_filter           IN VARCHAR2 DEFAULT  NULL,
   plan_line_filter          IN NUMBER   DEFAULT  NULL,
   event_detail              IN VARCHAR2 DEFAULT  'YES',
   bucket_max_count          IN NUMBER   DEFAULT  128,
   bucket_interval           IN NUMBER   DEFAULT  NULL,
   base_path                 IN VARCHAR2 DEFAULT  NULL,
   last_refresh_time         IN DATE     DEFAULT  NULL,
   report_level              IN VARCHAR2 DEFAULT 'TYPICAL',
   type                      IN VARCHAR2 DEFAULT 'TEXT',
   sql_plan_hash_value       IN NUMBER   DEFAULT  NULL,
```

```
  con_name                    IN VARCHAR2 DEFAULT  NULL)
 RETURN CLOB;
```

Parameters

Table 189-5 *REPORT_SQL_MONITOR Procedure Parameters*

Parameter	Description
sql_id	SQL_ID of the simple database operation for which monitoring information should be displayed. Use NULL (default) to display monitoring information for the last simple database operation monitored by Oracle.
dbop_name	DBOP_NAME for which monitoring information of the composite database operation is displayed
dbop_exec_id	Execution ID for the composite database operation for which monitoring information is displayed
session_id	Targets only the subset of statements executed and monitored on behalf of the specified session. Default is NULL. Use -1 or SYS_CONTEXT('SID') for the current session.
session_serial	In addition to session_id, you can specify the session serial number to ensure the desired session incarnation is targeted. This is ignored when session_id is NULL.
sql_exec_start	Time at which execution of the monitored SQL was started. Only applicable when sql_id is specified. Used to display monitoring information for a particular execution of sql_id. When NULL (default), the last execution of sql_id is shown.
sql_exec_id	A numeric ID generated internally by SQL monitor to identify different executions of the same SQL statement. Thus each execution will have the same sql_id but a different sql_exec_id. Only applicable when sql_id is specified and is used to display monitoring information for a particular execution of sql_id. When NULL (default), the last execution of sql_id is shown.
inst_id	Looks only at queries started on the specified instance. Use -1 to target the current instance. The default, NULL will target all instances.
start_time_filter	If not NULL, the report shows activity from V$ACTIVE_SESSION_HISTORY started after this date. If NULL, the reported activity starts once the targeted database operation has started.
end_time_filter	If not NULL, the report shows activity from V$ACTIVE_SESSION_HISTORY started before this date. If NULL, the reported activity ends when the targeted database operation has ended or SYSDATE if the operation is still executing.
instance_id_filter	Only looks at activity for the specified instance. Use NULL (the default) to target all instances. Only relevant if the query runs in parallel.
parallel_filter	Parallel filter applies only to parallel execution and allows you to select only a subset of the processes involved in the parallel execution. The string parallel_filter can be: • NULL - target all parallel execution servers as wells as the query coordinator • ['qc'][servers(<svr_grp>[,] <svr_set>[,] <srv_num>)] where any NULL value is interpreted as ALL

Table 189-5 (Cont.) *REPORT_SQL_MONITOR Procedure Parameters*

Parameter	Description
`plan_line_filter`	Selects activity and execution statistics for the specified line number in the plan of a SQL.
`event_detail`	When set to `NO`, the activity is aggregated by `wait_class` only. Use `YES` (default) to aggregate by `wait_class`, `event_name`.
`bucket_max_count`	Specifies the maximum number of buckets to create in the report
`bucket_interval`	Represents the exact time interval, in seconds, of all histogram buckets. If specified, `bucket_max_count` is ignored.
`base_path`	URL path for flex HTML resources since flex HTML format requires access to external files (Java scripts and the flash `swf` file).
`last_refresh_time`	If not `NULL` (default), the time when the report was last retrieved (`SYSDATE` attribute of the report tag). Use this option when you want to display the report of an running query and when that report is refreshed on a regular basis. This optimizes the size of the report since only the new changed information will be returned. In particular, the following will be optimized: • SQL text will not be returned when this option is specified • Activity histogram will start at the bucket that intersects that time. The entire content of the bucket is returned, even if `last_refresh_time` is after the start of that bucket

Table 189-5 (Cont.) *REPORT_SQL_MONITOR Procedure Parameters*

Parameter	Description
report_level	Level of detail for the report. Of the following, only one can be specified: • NONE: Minimum possible • BASIC: This is equivalent to sql_text-plan-xplan-sessions-instance-activity_histogram-plan_histogram-metrics where the token "-" implies that report section will not be included in the report. • TYPICAL: Everything but plan_histogram • ALL: Everything In addition, individual report sections can also be enabled or disabled by using a ±*section_name*. Several sections are defined: • XPLAN: Shows explain plan. ON by default. • PLAN: Shows plan monitoring statistics. ON by default. • SESSIONS: Show session details. Applies only to parallel queries. ON by default. • INSTANCE: Shows instance details. Applies only to parallel and cross instance queries. ON by default. • PARALLEL: An umbrella parameter for specifying sessions as well as instance details • ACTIVITY: Shows activity summary at global level, plan line level and session • INSTANCE LEVEL: (If applicable). ON by default. • BINDS: Shows bind information when available. ON by default. • METRICS: Shows metric data (such as CPU and IOs) over time. ON by default • ACTIVITY_HISTOGRAM: Shows a histogram of the overall query activity. ON by default. • PLAN_HISTOGRAM: Shows activity histogram at plan line level. OFF by default. • OTHER: Other information. ON by default. In addition, SQL text can be specified at different levels: • -SQL_TEXT: No SQL text in report • +SQL_TEXT: Alright with partial SQL text, that is, up to the first 2000 chars as stored in GV$SQL_MONITOR • SQL_FULLTEXT: No full SQL text, that is, +sql_text • +SQL_FULLTEXT: Show full SQL text (default)
type	Report type: • TEXT: text report (default) • HTML: simple HTML report • ACTIVE: database active report. Some information (explain plan, activity_histogram, metrics and plan_histogram) is only shown when this type is selected • XML: raw data for the report
sql_plan_hash_value	Targets only those with the specified plan hash value. Default is NULL.
con_name	Container name in a multitenant database.

Return Values

SQL monitor report, an XML document.

Usage Notes

The user invoking this function must have privilege to access the following fixed views:

- GV$SQL_MONITOR
- GV$SQL_PLAN_MONITOR
- GV$ACTIVE_SESSION_HISTORY
- GV$SESSION_LONGOPS
- GV$SQL if SQL full text is requested and its length is greater than 2 KB

REPORT_SQL_MONITOR_XML Function

This function is identical to the REPORT_SQL_MONITOR function, except that the return type is XMLType.

Related Topics

- REPORT_SQL_MONITOR Function
  This function builds a detailed report with monitoring information for a SQL statement, PL/SQL block, or database operation.

REPORT_SQL_MONITOR_LIST Function

This function builds a report for all or a subset of database operations that have been monitored by Oracle Database.

For each database operation, it gives key information and associated global statistics.

Syntax

```
DBMS_SQL_MONITOR.REPORT_SQL_MONITOR_LIST (
   sql_id                    IN VARCHAR2 DEFAULT  NULL,
   dbop_name                 IN VARCHAR2 DEFAULT  NULL,
   monitor_type              IN NUMBER   DEFAULT  MONITOR_TYPE_ALL,
   session_id                IN NUMBER   DEFAULT  NULL,
   session_serial            IN NUMBER   DEFAULT  NULL,
   inst_id                   IN NUMBER   DEFAULT  NULL,
   active_since_date         IN DATE     DEFAULT  NULL,
   active_since_sec          IN NUMBER   DEFAULT  NULL,
   last_refresh_time         IN DATE     DEFAULT  NULL,
   report_level              IN VARCHAR2 DEFAULT 'TYPICAL',
   auto_refresh              IN NUMBER   DEFAULT  NULL,
   base_path                 IN VARCHAR2 DEFAULT  NULL,
   type                      IN VARCHAR2 DEFAULT 'TEXT',
   con_name                  IN VARCHAR2 DEFAULT  NULL)
  RETURN CLOB;
```

Parameters

Table 189-6 *REPORT_SQL_MONITOR_LIST Procedure Parameters*

Parameter	Description
sql_id	SQL_ID of the simple database operation for which monitoring information should be displayed. Use NULL (default) to display monitoring information for the last operation monitored by Oracle Database.
dbop_name	DBOP_NAME for which monitoring information of the composite database operation is displayed.
monitor_type	Monitor type: • MONITOR_TYPE_SQL returns only simple database operations • MONITOR_TYPE_DBOP returns composite database operations • MONITOR_TYPE_ALL returns all types
session_id	Targets only the subset of database operations executed and monitored on behalf of the specified session. Default is NULL. Use -1 or SYS_CONTEXT('SID') for the current session.
session_serial	In addition tosession_id, you can specify the session serial number to ensure the desired session incarnation is targeted. This is ignored when session_id is NULL.
inst_id	Looks only at monitored database operations originating from the specified instance. Use -1 to target the instance where the report executed. To target all instances, use NULL (default).
active_since_date	If not NULL (default), returns monitored database operations that have been active since the specified time. This includes all operations that are executing, as well as all operations that have completed their execution after the specified start time.
active_since_sec	If not NULL (default), returns monitored database operations that have been active since the specified time. This includes all operations that are executing, as well as all operations that have completed their execution after the specified date and time. In this case, the start time is specified relative to the current SYSDATE minus a specified number of seconds. For example, use 3600 to limit the report to all operations that have been active in the past 1 hour.
last_refresh_time	If not NULL (default), the time when the list report was last retrieved. This optimizes the case where an application shows the list and refreshes the report on a regular basis (such as once every 5 seconds). In this case, the report will show details about the execution of monitored queries that have been active since the specified last_refresh_time. For other queries, the report returns the execution key (sql_id, sql_exec_start, and sql_exec_id). Also, for queries that have their first refresh time after the specified date, only the SQL execution key and statistics are returned.
report_level	Level of detail for the report. The level can be BASIC (SQL text up to 200 character), TYPICAL (which include full SQL text assuming that cursor has not aged out, in which case the SQL text is included up to 2000 characters), or ALL which is the same as TYPICAL.
auto_refresh	Specifies the duration in seconds after which report data will be automatically refreshed while the monitored SQL or database operation is still executing. This applies to active report types.

Table 189-6 (Cont.) *REPORT_SQL_MONITOR_LIST Procedure Parameters*

Parameter	Description
`base_path`	URL path for flex HTML resources since flex HTML format requires access to external files (java scripts and the flash `swf` file).
`type`	Report type: • `TEXT`: text report (default) • `HTML`: simple HTML report • `ACTIVE`: database active report. Some information (explain plan, `activity_histogram`, metrics, and `plan_histogram`) is only shown when this type is selected. • `XML`: raw data for the report
`con_name`	Container name in a multitenant database.

Return Values

A report in text, XML, or HTML format that contains the list of the database operations monitored.

Usage Notes

- Use the REPORT_SQL_MONITOR Function to get detailed monitoring information for a single database operation.
- The user invoking this function needs to have the privilege to access the fixed views `GV$SQL_MONITOR` and `GV$SQL`.

REPORT_SQL_MONITOR_LIST_XML Function

This function is identical to the `REPORT_SQL_MONITOR_LIST` function, except that it returns `XMLType`.

Related Topics

- REPORT_SQL_MONITOR_LIST Function
  This function builds a report for all or a subset of database operations that have been monitored by Oracle Database.