## Working With Automatic Materialized Views

Starting with Oracle Database Release 21c, materialized views can be created and maintained automatically.

The Oracle Database can automatically create and manage materialized views in order to optimize query performance. With very little or no interaction with the DBA, background tasks monitor and analyze workload characteristics and identifies where materialized views will improve SQL performance. The performance benefit of candidate materialized views is measured in the background (using workload queries) before they are made visible to the workload.

#### Note:

Automatic materialized views support partitioned and non-partitioned base tables. Incremental materialized view refresh is supported. In addition, for partitioned tables, there is support for Partition Change Tracking (PCT) view refresh. To be eligible for PCT-based refresh, partitioned base tables must use either range, list, or composite partitioning. If there is performance advantage, the automatic materialized view recommendations will include a partitioned automatic materialized view based on the partitioning of the base table of the materialized view. The partitioning type supported is auto-list partitioning, which will mirror the partitioning of the fact table.

The automatic materialized view maintenance module decides the type of refresh that is the most beneficial at the time of refresh, and will decide during run time whether to switch from incremental refresh to full refresh.

### Overview of Automatic Materialized Views

The database automatically collects workload information, workload queries and query execution statistics. It also maintains and purges the history of the workload. This eliminates a time-consuming DBA task.

Although automatic materialized views can run with minimal DBA interaction, their behavior can be easily adjusted.

This is a summary of automatic materialized view functionality:

- Automatically detects and collects workload query execution statistics. These include buffer-gets, database time, estimated cost, and other statistics.
- Creates candidate materialized views hidden from the database workload and verifies that
  they will deliver the projected performance benefit. It does this by test executing a sample
  of workload queries in the background.
- Provides reports detailing performance test results and which materialized views have been implemented.
- Provides automatic materialized view refresh.

The database implements only automatic materialized views whose benefits far outweigh the cost of maintaining them. It does not implement those that provide marginal benefit.

# Workload Information Provided by the Object Activity Tracking System

Automatic materialized views use workload information provided by the Object Activity Tracking System (OATS) as part of the automated decision-making processes.

Starting in Oracle Database 21c, the Object Activity Tracking System (OATS) tracks various activities associated with database objects. The automatic materialized view feature is one of the clients of this system. In the case of automatic materialized views, the usage data provided by OATS is one of the inputs into the analysis of cost versus benefit for creating or refreshing a materialized view, as well as in determining the best type of refresh and optimal refresh schedule.

OATS takes periodic snapshots of activity within any number of selected tables. The snapshot for each table captures the number of scans, loads, inserts/updates/deletes, truncations, and partition-related activity within the table from the beginning to the end of the snapshot interval. The DBA can use the DBMS\_ACTIVITY PL/SQL package to set the OATS capture interval, snapshot retention period, and space limits.

For example, the DBA\_ACTIVITY\_TABLE view shows the usage data captured within each snaphot.

## Data Dictionary Views That Provide Information About Automatic Materialized Views and OATS

As of Oracle Database 21c, the database includes data dictionary views that display information about automatic materialized views as well as OATS (Object Activity Tracking System).

#### **Views for Monitoring Automatic Materialized Views**

Use the following data dictionary views to check the automatic materialized view configuration and to examine various aspects of automatic materialized views activity:

DBA AUTO MV ANALYSIS ACTIONS

Displays information about analysis and tuning tasks, including actions, commands, advisor-specific flags, and command parameters.

- DBA AUTO MV ANALYSIS EXECUTIONS
  - Displays information about analysis and tuning executions, including concurrency, degree of parallelism (DOP) requested by the user and actual DOP upon execution finish, status, associated advisor, and informational or error message.
- DBA\_AUTO\_MV\_ANALYSIS\_RECOMMENDATIONS
   Displays recommendations associated with automatic materialized views.
- DBA\_AUTO\_MV\_ANALYSIS\_REPORT
   Reports on analyses and recommendations, including task and execution names, sequence number of the journal entry, and message entry in the journal.
- DBA AUTO MV ANALYSIS TASK



Displays analysis details associated with automatic materialized views, including task identifiers and task description, creation and last modification dates, execution data, parent task, status, and other information.

• DBA AUTO MV CONFIG

Displays the current automatic materialized view configuration.



The configuration parameters displayed in this view can be updated with CONFIGURE procedure of the DBMS AUTO MV package.

- DBA\_AUTO\_MV\_MAINT\_REPORT
   Displays the date, time, and message associated with automatic materialized view maintenance actions.
- DBA\_AUTO\_MV\_REFRESH\_HISTORY
   Displays the owner name, view name, date, start and end time, elapsed time, status, and error number (if an error occurred) for each automatic materialized view refresh.
- DBA\_AUTO\_MV\_VERIFICATION\_REPORT
   Displays the task name, execution name, and message associated with verifications.
- DBA\_AUTO\_MV\_VERIFICATION\_STATUS
   Displays the owner, start/end timestamps of verifications, SQL tuning sets used, and SQL Performance Analyzer tasks used in each verification.

#### **Views for Monitoring OATS**

DBA\_ACTIVITY\_CONFIG
 Displays the current value of the configuration parameters that control OATS.



The configuration parameters displayed in this view can be updated with CONFIGURE procedure of the DBMS ACTIVITY package.

- DBMS ACTIVITY TABLE
  - Describes table activity snapshots that were recently taken by OATS.
- DBA\_ACTIVITY\_SNAPSHOT\_META
   Displays information about activity snapshots taken by OATS.
- DBA\_ACTIVITY\_MVIEW
   Describes materialized view activity snapshots that were recently taken by OATS.



Oracle Database Reference



## The DBMS\_AUTO\_MV Package

This package contains procedures for controlling automatic materialized views.

#### DBMS\_AUTO\_MV.CONFIGURE

The DBA can use the <code>CONFIGURE</code> procedure of <code>DBMS\_AUTO\_MV</code> to start, stop, and configure automatic materialized views.

**Table 20-1 Configure Procedure Parameters** 

Parameter	Description and Examples
AUTO_MV_MODE	IMPLEMENT: Implements automatic materialized views.
	OFF: Turns off automatic materialized views.
	REPORT ONLY: Report-only mode.
	<pre>exec dbms_auto_mv.configure('AUTO_MV_MODE', 'IMPLEMENT');</pre>
	<pre>exec dbms_auto_mv.configure('AUTO_MV_MODE', 'OFF');</pre>
	<pre>exec dbms_auto_mv.configure('AUTO_MV_MODE', 'REPORT ONLY');</pre>
AUTO_MV_MAINT_TASK	ENABLE: Activates the task performing the maintenance (refreshes,
	validations, and cleanup).
	DISABLE: Deactivates the task performing the maintenance.
	CLEANUP_AND_DISABLE: Drops all automatic materialized views and deactivates the task. If automatic materialized views maintenance is in progress, then maintenance is allowed to finish before the task is deactivated.
	<pre>exec dbms_auto_mv.configure('AUTO_MV_MAINT_TASK', 'ENABLE');</pre>
	<pre>exec dbms_auto_mv.configure('AUTO_MV_MAINT_TASK', 'DISABLE');</pre>
	<pre>exec dbms_auto_mv.configure('AUTO_MV_MAINT_TASK', 'CLEANUP AND DISABLE');</pre>

Table 20-1 (Cont.) Configure Procedure Parameters

#### Parameter

#### **Description and Examples**

AUTO MV SPACE BUDGET

Specifies the percentage of space budgeted for implementing automatic materialized views within the tablespace where those views were created. This is a percentage of the total space used by all automatic materialized views and associated indexes within the tablespace.

A condition on the enforcement of AUTO MV SPACE BUDGET is the value of AUTO MV DEFAULT TABLESPACE:

- If AUTO MV DEFAULT TABLESPACE is not defined (NULL), then automatic materialized views are created on the tablespace of the view's parent object (which is the largest FACT table in the view's definition). In this case, the budget defined by AUTO MV SPACE BUDGET is enforced within that tablespace.
- If AUTO MV DEFAULT TABLESPACE is defined, then automatic materialized views are created in the designated default tablespace. In this case, the budget set by AUTO MV SPACE BUDGET is ignored.

If the budget is exceeded (possibly because of the growth of automatic materialized views), then the least-used automatic materialized view is dropped.

The value is an integer from 1 to 100. The default is 67 (67% of the total volume of the tablespace).

```
exec dbms auto mv.configure('AUTO MV SPACE BUDGET',
15);
```

AUTO MV DEFAULT TABLESPAC Specifies the default tablespace for the creation of automatic materialized views. Possible values are the name of a valid temporary tablespace or NULL (the default). In the case of NULL, new automatic materialized view is created in the default tablespace of the owner of the parent object. If the view has more than one parent object, such as materialized views defined on multiple base tables, then the default tablespace of the owner of largest base table is selected. If the value is changed dynamically, the change takes effect the next time automatic materialized view recommendations are implemented.

```
exec
dbms auto mv.configure('AUTO MV DEFAULT TABLESPACE',
'MYTABLESPACE');
exec
dbms auto mv.configure('AUTO MV DEFAULT TABLESPACE')
```

Table 20-1 (Cont.) Configure Procedure Parameters

Parameter	Description and Examples
AUTO_MV_TEMP_TABLESPACE	Specifies the temporary tablespace used for creation or refresh of automatic materialized views. Possible values are the name of a valid temporary tablespace or NULL. In the case of NULL, the tablepace is assigned to the owner of the largest parent object of the automatic materialized views. The default is NULL.
	<pre>exec dbms_auto_mv.configure('AUTO_MV_TEMP_TABLESPACE','TE MP2'); exec dbms_auto_mv.configure('AUTO_MV_TEMP_TABLESPACE');</pre>
AUTO_MV_RETENTION	Specifies the number of days automatic materialized views can continue to exist without being queried. If an automatic materialized view remains unused beyond this retention time, it is automatically dropped.  Possible values are any integer between 1 and 373. The default is 33
	<pre>days. exec dbms_auto_mv.configure('AUTO_MV_RETENTION', 365);</pre>
AUTO_MV_ANALYZE_REPORT_RE TENTION	AUTO_MV_ANALYZE_REPORT_RETENTION Specifies the maximum number of days to retain analysis and recommendation history. Possible values are any integer from 0 to 90. A value of 0 means no history is maintained. The default is 31 days.
	<pre>exec dbms_auto_mv.configure('AUTO_MV_ANALYZE_REPORT_RETEN TION', 60);</pre>
AUTO_MV_VERIFY_REPORT_RET	Specifies the maximum number of days to retain verification history. Possible values are any integer from 0 to 90. The value 0 specifies that no verification history will be maintained. The default is 31 days.
	<pre>exec dbms_auto_mv.configure('AUTO_MV_VERIFY_REPORT_RETENT ION', 7);</pre>
AUTO_MV_MAINT_REPORT_RETE NTION	Specifies the maximum number of days to retain history of automatic materialized view maintenance (refreshes) in the DBA_AUTO_MV_REFRESH_* dictionary tables. Possible values are any integer from 0 to 90. The value 0 specifies that no refresh history will be maintained. The default is 31 days.
	<pre>exec dbms_auto_mv.configure('AUTO_MV_MAINT_REPORT_RETENTI ON', 14);</pre>

Table 20-1 (Cont.) Configure Procedure Parameters

#### Parameter

#### **Description and Examples**

AUTO\_MV\_ANALYZE\_WORKLOAD\_WINDOW

Specifies the maximum number of hours to investigate queries from the latest snapshots and make recommendations. Possible values are any integer between from 1 to 8760. The default is 24 hours.

exec
dbms\_auto\_mv.configure('AUTO\_MV\_ANALYZE\_WORKLOAD\_WIN
DOW', 48);

AUTO\_MV\_ANALYZE\_WORKLOAD\_MIN TIME

Specifies the minimum time in seconds for a query to be considered for automatic materialized views recommendation. Queries below this threshold are not considered for recommendations. Possible values are any integer from 0 to 3600. The default is 120 seconds.

exec
dbms\_auto\_mv.configure('AUTO\_MV\_ANALYZE\_WORKLOAD\_MIN
\_TIME', 1800);

AUTO MV SCHEMA

Specifies a schema to be either included or excluded during the creation of automatic materialized views. The schema is added to the *inclusion list* or the *exclusion list* in the configuration. Initially, both lists are empty and automatic materialized views can be created in all the schemas in a database where automatic materialized views are enabled. You can build the inclusion and exclusion lists by calling AUTO\_MV\_SCHEMA multiple times.

The boolean ALLOW determines if the schema is added to the inclusion list (TRUE) or to the exclusion list (FALSE). The default is TRUE. During workload processing, any query that does not contain a reference at least one table in a schema on the inclusion list is not analyzed and not auto tuned. It is not factored into recommendations and verifications. Likewise, if a query references a table in a schema on the exclusion list, that query is excluded from processing.

exec
dbms\_auto\_mv.configure('AUTO\_MV\_SCHEMA', 'SCHEMA\_A');
exec
dbms\_auto\_mv.configure('AUTO\_MV\_SCHEMA', 'SCHEMA\_B',
FALSE);

To enable or disable processing of all schemas, you can specific the schema as NULL. This either enables or disables all of them, depending on the value of ALLOW.

exec dbms\_auto\_mv.configure('AUTO\_MV\_SCHEMA','',
TRUE);

Table 20-1 (Cont.) Configure Procedure Parameters

Parameter	Description and Examples
AUTO_MV_APP_MODULE	Specifies application modules to include or exclude from the creation of automatic materialized views.
	<pre>exec dbms_auto_mv.configure('AUTO_MV_APP_MODULE', 'MODULE1', TRUE); exec dbms_auto_mv.configure('AUTO_MV_APP_MODULE', 'MODULE1', FALSE); exec dbms_auto_mv.configure('AUTO_MV_APP_MODULE', 'MODULE1');</pre>

#### DBMS\_AUTO\_MV.DROP\_AUTO\_MVS

This procedure drops an automatic materialized view. It can be executed only by users who have the DBA role.

Parameter	Description
OWNER	The name of the owner of the automatic materialized view.
MV_NAME	The name of the automatic materialized view.
ALLOW_RECREATE	Allow the materialized view to be recreated if necessary. Optional.

Note that if OWNER is specified and MV\_NAME is set to NULL, then all automatic materialized views owned by OWNER are dropped.

```
exec dbms_auto_mv.drop_auto_mvs('SH', 'AUTO_MV$$_G2MKPB9SA1FB7', TRUE);
exec dbms_auto_mv.drop_auto_mvs('SH', 'AUTO_MV$$_G2MKPB9SA1FB7');
exec dbms_auto_mv.drop_auto_mvs('SH', '');
exec dbms_auto_mv.drop_auto_mvs('SH', '', TRUE);
```

#### DBMS\_AUTO\_MV.RECOMMEND

DBMS\_AUTO\_MV.RECOMMEND generates automatic materialized recommendations based on a given SQL tuning set. This API enables you to manually run automatic materialized view analysis and verification from a command line (instead of through an Automatic SQL Tuning task). You set the workload start and end time and determine whether this execution results in a report only, or an actual implementation. There is no default time limit for the workload window.

Execution of this API requires the DBA role.

#### **Note:**

Automatic materialized view configuration parameters can influence the results of <code>DBMS\_AUTO\_MV.RECOMMEND</code>. For example, the analysis and recommendations of this API are restricted to the schemas specified by the configuration parameter <code>AUTO\_MV\_SCHEMA</code>.

Parameter	Description
STS_OWNER	The name of the owner of the SQL tuning set. Default: SYS.
STS_NAME	The name of the SQL tuning set.  Default: SYS_AUTO_STS.
WORKLOAD_START_TIME	Start time for the workload window.  Default: WORKLOAD_END_TIME minus 24 hours.
WORKLOAD_END_TIME	End time for the workload window. Default: The current timestamp.
AUTO_MV_MODE	REPORT ONLY (recommendations only) or IMPLEMENT.  Default: REPORT ONLY

#### **Example:**

Generate and report recommendations using SYS\_AUTO\_STS for the past 24 hours. Note that the default behavior is REPORT\_ONLY, which means that no automatic materialized view will be implemented.

```
var exec_name varchar2(200);
begin
    :exec_name := dbms_auto_mv.recommend();
end;
SELECT * FROM DBA_AUTO_MV_ANALYSIS_RECOMMENDATIONS
    WHERE exec_name = :exec_name;
```

#### DBMS\_AUTO\_MV.REFRESH

The DBMS\_AUTO\_MV.RECOMMEND API enables you force a refresh of all stale automatic materialized views. The stale automatic materialized views are unconditionally refreshed in descending order, based on their verified query rewrite benefit values. There are no parameters. This routine can be executed only by users with the DBA role.

```
exec dbms_auto_mv.dbms_auto_refresh();
```

#### DBMS\_AUTO\_MV.REPORT\_ACTIVITY

The DBMS\_AUTO\_MV.REPORT\_ACTIVITY This API generates a report on automatic materialized view activities and usage within a specified time window. The report is returned as a CLOB.

Parameter	Description
ACTIVITY_START	The start of the time window.  Default: SYSTIMESTAMP -1.
ACTIVITY_END	The end of the time window.  Default: SYSTIMESTAMP.
TYPE	The format of the report. 'TEXT', 'HTML', and 'XML' are supported.  Default: 'TEXT'.



Parameter	Description
SECTION	The section or sections covered by the report. The value can be any combination of: SUMMARY, MV_DETAILS, QUERY_DETAILS, VERIFICATION_DETAILS or ALL. Default: 'ALL'.
	Vse the "+" or "- " operator to concatenate a single string that includes or excludes sections of the report. This is shown in one of the examples below.
LEVEL	The level of detail in the report: BASIC, TYPICAL or ALL.

#### **Examples:**

Generate a report on all automatic materialized view activities. Output the report in HTML format:

Default: 'TYPICAL'.

select dbms\_auto\_mv.report\_activity(type => 'HTML') from dual;

Generate a report on all automatic materilalized view activities. Exclude the verification details. Output the report in XML format.

select dbms\_auto\_mv.report\_activity(type => 'XML', section => 'ALL-VERIFICATION DETAILS') from dual;

#### DBMS\_AUTO\_MV.REPORT\_LAST\_ACTIVITY

The DBMS\_AUTO\_MV.REPORT\_LAST\_ACTIVITY API generates a report on the most recent automatic materialized view activities and usage.

Parameter	Description
TYPE	The format of the report. 'TEXT', 'HTML', and 'XML' are supported.
	Default: 'TEXT'.



Parameter	Description
SECTION	The section or sections covered by the report. The value can be any combination of: SUMMARY, MV_DETAILS, QUERY_DETAILS, VERIFICATION_DETAILS or ALL. Default: 'ALL'.
	Use the "+" or "- " operator to concatenate a single string that includes or excludes sections of the report. See the examples below.
LEVEL	The level of detail in the report: BASIC, TYPICAL or ALL.  Default: 'TYPICAL'.

#### **Examples:**

Generate a comprehensive report of the most recent activity, at the typical level of detail. Output the report in text format (the default). Note that both of these statements return the same results.

```
select dbms_auto_mv.report_last_activity('TEXT', 'ALL', 'TYPICAL') from dual;
select dbms_auto_mv.report_last_activity() from dual;
```

Generate a report of the most recent activity that includes only the summary and the details of the materialized view. Report at the maximum level of detail. Output in XML format:

```
select dbms_auto_mv.report_last_activity('XML', 'SUMMARY+MV_DETAILS', 'ALL')
from dual;
```

Generate a report of the most recent activity at the basic level of detail. Exclude the verification details. Output in HTML format.

```
select dbms_auto_mv.report_last_activity('XML', 'ALL-VERIFICATION_DETAIL',
'BASIC') from dual;
```

#### For More Information



The Oracle Database PL/SQL Packages and Types Reference.

### The DBMS\_ACTIVITY Package

The DBMS\_ACTIVITY PL/SQL package contains functions and procedures for configuring Object Activity Tracking System (OATS) information collection and management. Data collected by OATS is used in analyses performed by automatic materialized views.

DBAs can use the <code>DBMS\_ACTIVITY.CONFIGURE</code> procedure to control three OATS parameters within a specific database.

ACTIVITY INTERVAL

The interval between snapshots.

```
exec dbms activity.configure('ACTIVITY INTERVAL MINUTES', 30)
```

• ACTIVITY\_RETENTION\_DAYS

How long snapshots are saved.

```
exec dbms activity.configure('ACTIVITY RETENTION DAYS',60)
```

ACTIVITY SPACE PERCENT

How much of available space is reserved for snapshots.

```
exec dbms activity.configure('ACTIVITY SPACE PERCENT', 10)
```

Note:

OATS is intended to be self-managing and the default configuration is recommended, particularly if the automatic materialized views feature is used.

See Also:

The PL/SQL Packages and Types Reference.