

DBMS_INMEMORY_ADVISE

This package enables you to determine if a workload can benefit from Oracle's Database In-Memory feature.

DBMS_INMEMORY_ADVISE Overview

DBMS_INMEMORY_ADVISE provides tools to determine the benefit of the Database In-Memory feature to the workflow on a database.

DBMS_INMEMORY_ADVISE perform its analysis on the workload of the current database. You can do the same analysis on another database workload by importing and examining AWR snapshots.

Summary of DBMS_INMEMORY_ADVISE Subprograms

In Oracle Database 19c, DBMS_INMEMORY_ADVISE includes a single subprogram: IS_INMEMORY_ELIGIBLE.

Table 108-1 DBMS_INMEMORY_ADVISE Subprograms

Subprogram	Description
IS_INMEMORY_ELIGIBLE Procedure	Examines AWR snapshots to determine if a given workload will benefit or not benefit from the Database In-Memory feature.

IS_INMEMORY_ELIGIBLE Procedure

This procedure examines Automatic Workload Repository (AWR) snapshots from a database to determine the workload's eligibility for Database In-Memory technology.

Many workloads benefit from Database In-Memory, however some may not.

IS_INMEMORY_ELIGIBLE determines whether or not a workload would benefit from Database In-Memory and should be considered eligible or ineligible. Eligibility is gauged by the percentage of analytical activity in the workload. If you are planning to implement Database In-Memory, you can use this tool to quickly identify and filter out databases that are ineligible; those where analytic activity is low and where you would see no substantive gain from the use of Database In-Memory. You can then focus your Database In-Memory deployment on databases whose workload includes more analytic activity and could therefore benefit substantially.

Criterion Used by IS_INMEMORY_ELIGIBLE

The determination that a workflow is eligible or ineligible for Database In-Memory is based on a system-defined threshold. If the percentage of the workload involved analytic activity is below that threshold, then Database In-Memory will not noticeably improve performance. The greater the level of activity above that target analytic workload percentage threshold, the greater is the potential benefit that may be seen from use of Database In-Memory.

IS_INMEMORY_ELIGIBLE Procedure Syntax

IS_INMEMORY_ELIGIBLE has three overloads that let you choose how to define the scope of the eligibility test. The output parameters are the same in all three. The optional input parameter SNAP_DBID is also common.

The input parameters are mutually exclusive to each overload as shown in the syntax below. You cannot combine them in a single procedure call.

Syntax 1: Input as Most Recent *n* Days, Including Today

```
DBMS_INMEMORY_ADVISE.IS_INMEMORY_ELIGIBLE (
    TIME_WINDOW_DAYS    IN NUMBER,
    INMEM_ELIGIBLE       OUT BOOLEAN,
    ANALYSIS_SUMMARY     OUT VARCHAR2,
    SNAP_DBID            IN NUMBER DEFAULT NULL
);
```

Syntax 2: Input as a Snapshot Range

```
DBMS_INMEMORY_ADVISE.IS_INMEMORY_ELIGIBLE (
    START_SNAP_ID        IN NUMBER,
    END_SNAP_ID           IN NUMBER,
    INMEM_ELIGIBLE       OUT BOOLEAN,
    ANALYSIS_SUMMARY     OUT VARCHAR2,
    SNAP_DBID            IN NUMBER DEFAULT NULL
);
```

Syntax 3: Input as Time Interval (Start Time and End Time)

```
DBMS_INMEMORY_ADVISE.IS_INMEMORY_ELIGIBLE (
    START_TIME           IN TIMESTAMP,
    END_TIME             IN TIMESTAMP,
    INMEM_ELIGIBLE       OUT BOOLEAN,
    ANALYSIS_SUMMARY     OUT VARCHAR2,
    SNAP_DBID            IN NUMBER DEFAULT NULL
);
```

Parameters

Table 108-2 IS_INMEMORY_ELIGIBLE Procedure Parameters

Parameter	Description
TIME_WINDOW_DAYS	All snapshots occurring within the last <i>n</i> number of days, including today.
START_TIME	Specifies the start of a timespan for snapshot analysis. All AWR snapshots created after START_TIME, up to and included END_TIME are examined.
END_TIME	Specifies the end of a timespan for snapshot analysis. All AWR snapshots created prior to END_TIME, back to and including START_TIME are examined.

Table 108-2 (Cont.) IS_INMEMORY_ELIGIBLE Procedure Parameters

Parameter	Description
START_SNAP_ID	The first snapshot in a sequence of snapshots. This is the starting point of the workload to be analyzed.
END_SNAP_ID	The end of a sequence of snapshots.
INMEM_ELIGIBLE	The determination of whether or not the workflow is eligible for (will benefit from) Database In-Memory. Possible values: True, False.
ANALYSIS_SUMMARY	The analysis that explains the INMEM_ELIGIBLE evaluation. Example: Observed Analytic Workload Percentage is 3% is less than target Analytic Workload Percentage 20% Workloads with a percentage of analytical activity less than target are considered ineligible.
SNAP_DBID	Database ID of a database from where AWR data has been exported. This enables you to analyze the workload on another database. If you are testing for eligibility on the current database, exclude this parameter.

Usage Notes

Within a given workload, the level of analytical activity may vary over a series of snapshots. Depending on which subset of all available snapshots are selected for testing, the analytic workload percentage calculated by `IS_INMEMORY_ELIGIBLE` may correspondingly vary. So you may want to do several evaluations with different input parameters. Using the `TIME_WINDOW_DAYS` parameter gives you a quick summary of the workload based on that time window. Then you can drill down into specific time periods, using the parameters based on snapshot IDs to get a perspective on patterns in the workload.

If you are considering whether or not to enable Database In-Memory on your databases, do the following:

1. Run the In-Memory Eligibility Test on candidate databases to find out which databases can or cannot effectively use Database In-Memory.
2. Run the In-Memory Advisor on any Oracle 19c database, except those that the In-Memory Eligibility Test has determined are ineligible.

Examples

Example 1: Testing for eligibility within the time window of the last *n* days as input.

```
set serverout on;
set lines 200;

DECLARE
    inmem_eligible BOOLEAN;
    analysis_summary VARCHAR2(200);
BEGIN

    dbms_inmemory_advise.is_inmemory_eligible(1, inmem_eligible,
analysis_summary);
```

```
        DBMS_OUTPUT.PUT_LINE(
            CASE WHEN inmem_eligible = TRUE
                THEN 'Eligibile for In Memory'
                ELSE 'Not Eligible for In Memory'
            END
        );

        DBMS_OUTPUT.PUT_LINE(analysis_summary);

END;
/
```

Example 2: Testing for eligibility with a range of snapshot IDs as input.

```
set serverout on;
set lines 200;
DECLARE
    inmem_eligible BOOLEAN;
    analysis_summary VARCHAR2(200);
BEGIN

    dbms_inmemory_advise.is_inmemory_eligible(355, 356, inmem_eligible,
analysis_summary);

    DBMS_OUTPUT.PUT_LINE(
        CASE WHEN inmem_eligible = TRUE
            THEN 'Eligibile for In Memory'
            ELSE 'Not Eligible for In Memory'
        END
    );

    DBMS_OUTPUT.PUT_LINE(analysis_summary);

END;
/
```

Example 3: Testing for eligibility with a time interval as input. All snapshots beginning and ending within the time interval are included.

```
set serverout on;
set lines 200;
DECLARE
    inmem_eligible BOOLEAN;
    analysis_summary VARCHAR2(120);
BEGIN

    dbms_inmemory_advise.is_inmemory_eligible(TO_TIMESTAMP('21-JUN-23
05.22.27.262 PM', 'DD-Mon-RR HH:MI:SS.FF AM'),
                                                    TO_TIMESTAMP('21-JUN-23
05.26.04.446 PM', 'DD-Mon-RR HH:MI:SS.FF AM'),
inmem_eligible,
analysis_summary
    );
```

```
DBMS_OUTPUT.PUT_LINE(  
    CASE WHEN inmem_eligible = TRUE  
        THEN 'Eligibile for In Memory'  
        ELSE 'Not Eligible for In Memory'  
    END  
);  
  
DBMS_OUTPUT.PUT_LINE(analysis_summary);  
  
END;  
/
```