106
DBMS_INMEMORY

The `DBMS_INMEMORY` package provides an interface for In-Memory Column Store (IM column store) functionality.

This chapter contains the following topics:

- DBMS_INMEMORY Overview
- DBMS_INMEMORY Security Model
- Summary of DBMS_INMEMORY Subprograms

DBMS_INMEMORY Overview

This package contains procedures for populating and repopulating the IM column store, and for dropping IM expressions from a specified table.

IM Population and Repopulation

In-Memory population (population) occurs when the database reads existing row-format data from disk, transforms it into columnar format, and then stores it in the IM column store. Only objects with the `INMEMORY` attribute are eligible for population.

Population, which transforms existing data on disk into columnar format, is different from repopulation, which loads new data into the IM column store. Repopulation occurs automatically after their columnar data undergo significant DML activity.

When an object has the `INMEMORY` attribute and a priority other than `NONE`, the database gradually populates the object in the IM column store according to an internal priority queue. Objects with priority of `NONE` are populated only when they undergo a full scan.

The `DBMS_INMEMORY.POPULATE` procedure forces immediate population of an object. The `DBMS_INMEMORY.REPOPULATE` procedure forces immediate repopulation of an object.

> **See Also:**
>
> *Oracle Database In-Memory Guide* to learn more about IM population

IM Expressions

IM expressions populate frequently evaluated query expressions in the IM column store for subsequent reuse. An IM expression is materialized as a hidden virtual column, prefixed with the string `SYS_IME`, and is accessed in the same way as a non-virtual column.

When you use `DBMS_INMEMORY_ADMIN.IME_CAPTURE_EXPRESSIONS`, the database adds the 20 hottest expressions to their respective tables as `SYS_IME` columns and applies the default `INMEMORY` column compression clause. If any `SYS_IME` columns that were added during a previous invocation are no longer in the latest expression list, then the database changes their attribute to `NO INMEMORY`.

The maximum number of SYS_IME columns for a table, regardless of whether the attribute is INMEMORY or NO INMEMORY, is 50. After the 50 expression limit is reached for a table, the database will not add new SYS_IME columns. To make space for new expressions, you must manually drop SYS_IME columns with the DBMS_INMEMORY.IME_DROP_EXPRESSIONS or DBMS_INMEMORY_ADMIN.IME_DROP_ALL_EXPRESSIONS procedures.

> **See Also:**
>
> *Oracle Database In-Memory Guide* to learn more about IM expressions

DBMS_INMEMORY Security Model

The DBMS_INMEMORY package subprograms execute with invoker's rights.

The POPULATE and REPOPULATE procedures require the invoking user to have SELECT privileges on the specified object. For IME_DROP_EXPRESSIONS, the invoking user must have ALTER TABLE privileges on the specified table.

Summary of DBMS_INMEMORY Subprograms

This table lists and briefly describes the DBMS_INMEMORY package subprograms.

Table 106-1 DBMS_INMEMORY Package Subprograms

Subprogram	Description
IME_DROP_EXPRESSIONS Procedure	Drops a specified set of SYS_IME virtual columns from a table
POPULATE Procedure	Forces population of the specified table
REPOPULATE Procedure	Forces repopulation of the specified table
SEGMENT_DEALLOCATE_VERSIONS Procedure	Deallocates non-current IMCUs in the IM column store

IME_DROP_EXPRESSIONS Procedure

This procedure drops a specified set of SYS_IME virtual columns from a table.

Syntax

```
DBMS_INMEMORY.IME_DROP_EXPRESSIONS(
   schema_name     IN    VARCHAR2,
   table_name      IN    VARCHAR2,
   column_name     IN    VARCHAR2 DEFAULT NULL);
```

Parameters

Table 106-2 IME_DROP_EXPRESSIONS Procedure Parameters

Parameter	Description
schema_name	The name of the schema that contains the In-Memory table
table_name	The name of the In-Memory table that contains the SYS_IME columns
column_name	The name of the SYS_IME column. By default this value is null, which specifies all SYS_IME columns in this table.

Usage Notes

Typical reasons for dropping SYS_IME columns are space and performance. The maximum number of SYS_IME columns for a table, regardless of whether the attribute is INMEMORY or NO INMEMORY, is 50. After the 50-expression limit is reached for a table, the database will not add new SYS_IME columns. To make space for new expressions, you must manually drop SYS_IME columns with the DBMS_INMEMORY.IME_DROP_EXPRESSIONS or DBMS_INMEMORY_ADMIN.IME_DROP_ALL_EXPRESSIONS procedures.

To drop a specified SYS_IME column or all SYS_IME columns in the requested table, use DBMS_INMEMORY.IME_DROP_EXPRESSIONS . To populate these segments again, either invoke the DBMS_INMEMORY.POPULATE procedure, or perform a full table scan.

POPULATE Procedure

This procedure forces population of the specified table, partition, or subpartition into the IM column store.

Syntax

```
DBMS_INMEMORY.POPULATE(
   schema_name      IN    VARCHAR2,
   table_name       IN    VARCHAR2,
   subobject_name   IN    VARCHAR2 DEFAULT NULL);
```

Parameters

Table 106-3 *POPULATE Procedure Parameters*

Parameter	Description
schema_name	Name of schema
table_name	Name of table
subobject_name	Partition or subpartition

REPOPULATE Procedure

This procedure forces repopulation of a table, partition, or subpartition that is currently populated in the IM column store.

Syntax

```
DBMS_INMEMORY.REPOPULATE(
   schema_name      IN    VARCHAR2,
   table_name       IN    VARCHAR2,
   subobject_name   IN    VARCHAR2 DEFAULT NULL,
   force            IN    BOOLEAN DEFAULT FALSE);
```

Parameters

Table 106-4 REPOPULATE Procedure Parameters

Parameter	Description
schema_name	Name of the schema that owns the object.
table_name	Name of the table requiring repopulation.
subobject_name	Name of the partition or subpartition. If null, then repopulate the entire table.
force	Whether to repopulate all IMCUs in the segment, just as in initial population. The following values are possible for the force parameter: • FALSE — The database repopulates only IMCUs containing modified rows. This is the default. • TRUE — The database drops the segment, and then rebuilds it. The database increments the statistics and performs all other tasks related to initial population. For example, IMCU 1 contains rows 1 to 500,000, and IMCU 2 contains rows 500,001 to 1,000,000. A statement modifies row 600,000. When force is FALSE, the database only repopulates IMCU 2. When force is TRUE, the database repopulates both IMCUs. Consider further that the INMEMORY_VIRTUAL_COLUMNS initialization parameter is set to ENABLE, and an application creates a new virtual column. When force is FALSE, the database only repopulates IMCU 2 with the new column. When force is TRUE, the database repopulates both IMCUs with the new column.

SEGMENT_DEALLOCATE_VERSIONS Procedure

This procedure deallocates non-current IMCUs in the IM column store.

Syntax

```
DBMS_INMEMORY.SEGMENT_DEALLOCATE_VERSIONS(
   SCHEMA_NAME      IN  VARCHAR2,
   TABLE_NAME       IN  VARCHAR2,
   PARTITION_NAME   IN  VARCHAR2 DEFAULT NULL,
   SPCPRESSURE      IN  BOOLEAN DEFAULT FALSE);
```

Parameters

Table 106-5 SEGMENT_DEALLOCATE_VERSIONS Procedure Parameters

Parameter	Description
schema_name	Name of the schema that owns the object.
table_name	Name of the table requiring repopulation.
partition_name	Name of the partition or subpartition. If null, then repopulate the entire table.
spcpressure	Whether to force deallocation of non-current IMCUs (TRUE), or wait for the database to deallocate them automatically. By default, the database deallocates non-current IMCUs every two minutes.

Usage Notes

During repopulation, the IM column store maintains both the current IMCU and non-current IMCU. This mechanism, which is called *double buffering*, ensures that queries do not decrease performance because an IMCU is unavailable during repopulation. After repopulation completes, the IM column store retains the non-current IMCU for a short time (2 minutes by default) to optimize queries with older SCNs. Typically, the default behavior is sufficient. However, you can force deallocation of non-current IMCUs by using the SEGMENT_DEALLOCATE_VERSIONS procedure.

Example

The following program forces deallocation of non-current IMCUs for the products table:

```
BEGIN
  DBMS_INMEMORY.SEGMENT_DEALLOCATE_VERSIONS(
    schema_name => 'SH'
  , table_name  => 'PRODUCTS'
  , spcpressure => TRUE );
END;
```

> **See Also:**
>
> *Oracle Database In-Memory Guide* to learn more about double buffering in the IM column store