**HANA memory management**

SAP hana st04 Memory and CPU Physical Memory Used/Available – all memory red

From <http://hanadba.blogspot.com/p/tables-in-sap-hana.html>

The SAP HANA database loads columnar tables into memory column by column only upon use. This is sometimes called **lazy loading**. This means that columns that are never used are not loaded which avoids memory waste.

**M\_CS\_TABLES and M\_CS\_COLUMNS views**

The **M\_CS\_TABLES** and **M\_CS\_COLUMNS** views contain a lot of additional information - cardinality, main-storage versus delta storage and more.

**M\_CS\_TABLES** System View - Provides runtime data for column tables - <https://help.sap.com/viewer/4fe29514fd584807ac9f2a04f6754767/2.0.03/en-US/20ad60f77519101498ccb610c33c3ca6.html>

select host, round(sum(MEMORY\_SIZE\_IN\_TOTAL)/1024/1024) as "Total Column Tables MB Used"

from M\_CS\_TABLES

group by host

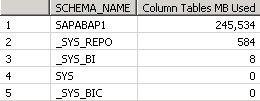


select schema\_name, round(sum(MEMORY\_SIZE\_IN\_TOTAL)/1024/1024) as "Column Tables MB Used"

from M\_CS\_TABLES

group by schema\_name

order by "Column Tables MB Used" desc



select

TABLE\_NAME as "Table",

round(MEMORY\_SIZE\_IN\_TOTAL/1024/1024) as "Total Memory Used MB",

Record\_Count,

round(MEMORY\_SIZE\_IN\_MAIN/1024/1024) as "Main Memory Used MB",

--round(MEMORY\_SIZE\_IN\_HISTORY\_MAIN/1024/1024) as "History Memory Used MB",

round(MEMORY\_SIZE\_IN\_DELTA/1024/1024) as "Delta Memory Used MB",

MERGE\_COUNT,

*--IS\_DELTA\_LOADED,*

PART\_ID as "Partition Number",

*--IS\_LOG\_DELTA,*

*--PERSISTENT\_MERGE*

*--LAST\_CONSISTENCY\_CHECK\_TIME,*

*--LAST\_CONSISTENCY\_CHECK\_ERROR\_COUNT*

LOADED as "Loaded in RAM Memory"

*--IS\_REPLICA*

*--UNUSED\_RETENTION\_PERIOD*

from M\_CS\_TABLES

where SCHEMA\_NAME = 'SAPABAP1' and

TABLE\_NAME like '/BIC/ABDPAYDOC%' and

round(MEMORY\_SIZE\_IN\_TOTAL/1024/1024) >0

order by TABLE\_NAME, PART\_ID

где

***IS\_LOG\_DELTA*** - Indicates whether that currently the redo log *is currently being written* (был ли записан).

***PERSISTENT\_MERGE*** - Indicates whether the new main part will be written to the disk during a table delta merge - unless requested differently.

***PART\_ID*** –

* For partitioned tables the part ID is equal to the sequential number of the partition starting at 1.
* For replicated tables the part ID is 1 for the original table and subsequent part IDs are assigned to replica tables.
* The part ID is 0 for tables that are not partitioned.

**M\_DELTA\_MERGE\_STATISTICS** - lists the table delta merge events since the indexserver was last restarted. Table delta merges - optimize compression runs and application merge hints are listed separatly.

select top 1000 \*

from

M\_DELTA\_MERGE\_STATISTICS

where SCHEMA\_NAME = 'SAPABAP1'

*Пустой результат – не настроена статистика?*

**M\_GARBAGE\_COLLECTION\_STATISTICS** -

*Пустой результат – не настроена статистика?*

**M\_CONTEXT\_MEMORY\_RESET** - memory allocator statistics since last reset. This view contains values accumulated since the last reset of the main view M\_CONTEXT\_MEMORY. Please refer to M\_CONTEXT\_MEMORY for information about the structure and use of this view.

*Пустой результат – не настроена статистика?*

**M\_RS\_TABLES view**

You can use the **M\_RS\_TABLES** view to examine the memory consumption of row tables.

select

round(sum(USED\_FIXED\_PART\_SIZE + USED\_VARIABLE\_PART\_SIZE)/1024/1024) as "Row Tables MB Used" from M\_RS\_TABLES

SELECT

SCHEMA\_NAME,

TABLE\_NAME,

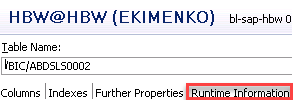
round((USED\_FIXED\_PART\_SIZE +USED\_VARIABLE\_PART\_SIZE)/1024/1024, 2) AS "MB Used"

FROM M\_RS\_TABLES WHERE schema\_name = 'SYS' ORDER BY "MB Used" DESC, TABLE\_NAME

# 6 Tips to avoid HANA Out of Memory (OOM) Errors

<https://blogs.sap.com/2014/01/05/6-tips-to-avoid-hana-out-of-memory-oom-errors/>

1. You should avoid using Calculation Views until you have mastered the Analytic View – always try to build an AV to solve a problem first. This is because AV don’t materialize large datasets unless you materialize the whole Analytic View.
2. HANA keeps a ***compressed main store*** of column tables and an ***uncompressed delta store*** for new items. Periodically a process called mergedog combines the two in a DELTA MERGE process. This is a set of design *tradeoffs* (компромисс) that ensures fast read and fast writes - and good compression.

If you double click on a table and click the *Runtime Information tab* -  -

you will see two memory consumption numbers - Main and Delta. Delta should always be much less than main during correct operation. *You can right click a table any time and select* *Perform Delta Merge*.

**SAP HANA System Tables and Monitoring Views Reference**

<https://anibalg.files.wordpress.com/2012/08/hana_monitor_views_en.pdf>

### **2057046 – FAQ - SAP HANA Delta Merges**

<http://sapkbs.blogspot.com/2016/11/2057046-faq-sap-hana-delta-merges.html>

**mergedog** is the system process that periodically checks column tables to determine whether or not a delta merge operation needs to be executed.

Automatic merges are performed by the mergedog which is configured with the following parameters -

|  |  |
| --- | --- |
| **Parameter** | **Value** |
| indexserver.ini -> [mergedog] -> ***active*** | true - activates mergedog - default false - deactivates mergedog |
| indexserver.ini -> [mergedog] -> ***check\_interval*** | <frequency\_ms> - defines the interval in ms of mergedog invocations - default 60000 - i.e. 1 minute. |

**Delta merge вручную**

**Hard merges** are unconditioned merges controlled by the application or administrator. They are either executed immediately or - in case of high concurrent merge activity - at a later point in time.

MERGE DELTA OF "<table\_name>" [FORCE REBUILD]

где

*FORCE REBUILD* option - indicate that the delta log on disk is cleaned immediately.

или

в SAP HANA Studio - <system> -> Catalog -> <schema> -> <table> -> Perform Delta Merge...

A **forced merge** is a variant of a hard merge which is generally executed immediately –

MERGE DELTA OF "<table\_name>" *WITH PARAMETERS* ('FORCED\_MERGE' = 'ON')

### **How can the history of temporal tables be merged manually?**

MERGE HISTORY DELTA OF "<table\_name>" ...

### **How can a manual delta merge be triggered for individual partitions?**

MERGE DELTA OF "<table\_name>" PART <part\_id> ...

**Мониторинг Delta merge**

Delta merge activities are tracked in table **M\_DELTA\_MERGE\_STATISTICS** and its history in **HOST\_DELTA\_MERGE\_STATISTICS**.

The amount of records in the monitoring view M\_DELTA\_MERGE\_STATISTICS can be controlled with the following parameter

indexserver.ini -> [mergedog] -> delta\_merge\_statistics\_record\_limit.

Default – 100000.

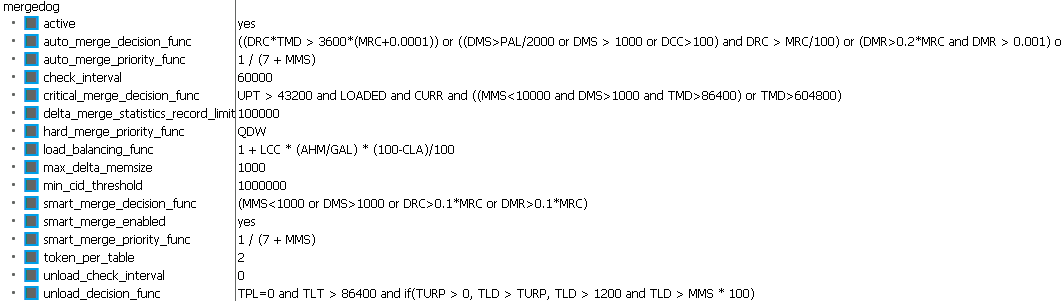
<https://www.stechies.com/sap-hana-monitoring-views-control-amount-records/>

You can use SQL HANA\_Tables\_ColumnStore\_Merges to retrieve information.

The current utilization of the delta storage can be analyzed using SQL HANA\_Tables\_ColumnStore\_DeltaStorage - SAP Note [1969700](https://launchpad.support.sap.com/#/notes/1969700).

**How can peaks of high load due to concurrent or parallelized delta merges be prevented?**

All merge operations apart from forced merges can only start if a so-called **merge token** is assigned. If no merge token is available the merges have to wait. ***The total number of available merge tokens*** is calculated via the following parameter



indexserver.ini -> [mergedog] -> load\_balancing\_func

The value calculation includes information like the number of available CPUs - the CPU utilization and the heap memory situation.

*When a merge is triggered SAP HANA calculates the number of required merge tokens.* If this calculation is not possible - the value defined by the following parameter is used. Default – 2.

indexserver.ini -> [mergedog] -> token\_per\_table

Furthermore there are **priority parameters** that are used to calculate priorities of merge requests so that they can be processed in an optimized order -

indexserver.ini -> [mergedog] -> auto\_merge\_priority\_func

indexserver.ini -> [mergedog] -> hard\_merge\_priority\_func

indexserver.ini -> [mergedog] -> smart\_merge\_priority\_func

**DBTABLOG** is a standard SAP Table which is used to store Log Records of Table Changes data.

# [Housekeeping for BW Requests](https://wiki.scn.sap.com/wiki/display/BI/Housekeeping+for+BW+Requests)

<https://wiki.scn.sap.com/wiki/display/BI/Housekeeping+for+BW+Requests>

One of the pre-condition is that only requests up to the oldest delta requests that were loaded to this data target can be reduced. In order to analyse and to find out the inbound or outbound delta upload transaction **RSREQREDUCE** is very useful.

# Health checks of HANA system

<https://blogs.sap.com/2017/09/04/health-checks-of-hana-system/>

SQL scripts provided by SAP in OSS note [1969700 – SQL Statement Collection for SAP HANA](https://launchpad.support.sap.com/#/notes/1969700). This OSS note provides a collection of very useful SQL scripts with which you can check a wide range of areas that are critical to a HANA system.

## **HANA\_Tables\_ColumnStore\_AutoMergeDisabled**

**Description**

All non-BW tables should have auto merge enabled. This will ensure that the delta store does not grow too much as it has performance implications. ***The delta store contains uncompressed data to speed up insert/update queries.*** The main store is read optimized. So data from delta store needs to be merged regularly with main store to improve the read performance of the queries and also to reduce memory consumption as tables/table partitions are compressed automatically after a merge operation.

However note that ***BW tables use smart merge option which is controlled by the SAP application***. There are certain parameters in the mergedog section under indexserver.ini which controls this. ***Auto merge should not be enabled for BW tables*** otherwise they may interfere with the smart merge decisions and cause **adverse** ('ædvɜːs неблагоприятный) impact.

You can use this SQL script to find out the non-BW tables - by default the modification section of the script excludes BW tables that have auto merge disabled.

Настройки на стороне сервера *приложений*.

Файл indexserver.ini - секция mergedog.

**Smart merges** are conditioned merges controlled by the application. This approach can be useful during specific application scenarios like BW data loads. It makes sure that auto merge operations don't interfer negatively with the business activities.

Smart merges can be controlled by setting the following parameter to 'true' (enabled, default) or 'false' (disabled):

indexserver.ini -> [mergedog] -> smart\_merge\_enabled

**When a smart merge is triggered by the application** the following parameter controls if a merge is actually executed

indexserver.ini -> [mergedog] -> smart\_merge\_decision\_func

Smart merges are triggered with the following command

MERGE DELTA OF "<table\_name>" WITH PARAMETERS ('SMART\_MERGE' = 'ON')

In SAP ABAP environments you can use the function module TREX\_EXT\_MERGE\_DELTA\_INDEX for triggering smart merges from application side.

remove change log data Для этого типа ADSO необходимо сначала распространить запрос журнала изм.

№ сообщения RSDSO\_MANAGE053

Diagnosis

For this type of aDSO *the change log request must first be propagated to all delta-connected target InfoProviders* before the change log request can be removed.

This can be either a bug or a expected behavior, depending on your scenario. Please check these SAP Notes and SAP KBA and implement/follow the ones that matches your scenario: [2222122](https://launchpad.support.sap.com/#/notes/2222122), [2359262](https://launchpad.support.sap.com/#/notes/2359262), [2413256](https://launchpad.support.sap.com/#/notes/2413256) and [2660845](https://launchpad.support.sap.com/#/notes/2660845).

**Sap bw remove change log data Change log request must be propagated first with this type of aDSO**

# BW on HANA - Not able to delete change log entries of ADSO

We have an ADSO with type Delta Calculated. We need to now delete old change log from this ADSO. We have tried below options but of no use.

1. Deleting Via PC variant ***Cleanup old requests from ADSO*** - though job gets completed entries in table remaining unchanged

2. Deleting via program ***RSDSO\_REMOVE\_REQS\_PC*** and API ***RSDSO\_REMOVE\_REQUESTS\_API*** also did not work

3. ***Manage ADSO --> Environment --> Remove Change log*** - only bottom most request is getting deleted, while for other requests it is not allowing to and says Change log request must be propagated first with this type of ADSO.

Please advice how we can delete these old change log as the table heaping close to 2 Billion.

# sab bw hana *pool/joinevaluator/jerequestedattributes/result*

# See - *Note 2088349* - memory leak when querying calculation views which are executed in