2127458 - FAQ: SAP HANA Loads and Unloads

Version	46	**Type**	SAP Knowledge Base Article
Language	English	**Master Language**	English
Release Status	Released to Customer	**Category**	How To
Component	HAN-DB (SAP HANA Database)	**Released On**	23.07.2018

Please find the original document at **https://launchpad.support.sap.com/#/notes/2127458**

Symptom

You are interested in details about unloads in SAP HANA environments.

Environment

SAP HANA

Cause

Resolution

1. What are loads and unloads in SAP HANA environments?

When a column is loaded, it is copied from persistence into SAP HANA column store memory.

Unloads are table column displacements from column store memory.

In this SAP Note we focus on loads and unloads of columns in column store. Tables in row store are typically loaded during startup and remain in memory permanently.

2. Which indications exist for SAP HANA load and unload issues?

While loads usually don't indicate issues, unloads are critical for the following reasons: They are often indicators of memory bottlenecks and they introduce overhead because unloaded column may have to be reloaded after some time.

The following SAP HANA alerts indicate problems in the unload area:

Alert	Name	SAP	Description

		Note	
55	Columnstore unloads	1977207	Determines how many columns in columnstore tables have been unloaded from memory. This can indicate performance issues.

SQL: "HANA_Configuration_MiniChecks" (SAP Notes 1969700, 1999993) returns a potentially critical issue (C = 'X') for one of the following individual checks:

Check ID	Details
430	Number of low memory column unloads (last day)
431	Time since last low memory column unload (days)
435	Number of shrink column unloads (last day)
437	Size of unloaded columns (GB, last day)

3. When do loads happen?

Columns are loaded into memory in the following situations:

Reason	Details
Explicit access	When a table column is accessed and it doesn't reside in memory, yet, it is loaded into memory. Exceptions: Hybrid LOB columns are not loaded into memory when they exceed the configured memory threshold (SAP Note 1994962). Paged attributes are only partially loaded into memory (SAP Note 1871386). Time spent for loading table columns during SQL statement preparation can be identified via column TOTAL_TABLE_LOAD_TIME_DURING_PREPARATION in monitoring view M_SQL_PLAN_CACHE. Time spent for loading table columns during SQL statement execution is not inidividually displayed.
Explicit load	You can use the following options to load tables into memory explicitly: Command Details LOAD "<table>" ALL SAP HANA Studio -> <system> -> Catalog -> <schema> -> <table> -> "Load into memory..." Load all table columns of one table into memory LOAD "<table>" ("<column>") LOAD "<table>" ("column1>", ..., "<columnN>") Load the defined table columns into memory LOAD "<table>" DELTA Load the delta storage of the table into memory LOAD "<table>" HISTORY ... When the key word HISTORY is added after <table>, the load is related to the history part of a temporal table. If you want to load all tables into memory (e.g. as recommended in SAP Note 2066313), you can use the following Python script available as part of the SAP HANA installation: /usr/sap/<sid>/HDB<inst_id>/exe/python_support/loa dAllTables.py This tool should only be used in exceptional situations, because loading all tables into memory can result in memory and CPU bottlenecks.
Reload after startup (explicitly configured tables)	The following commands can be used to define tables that should be loaded directly after startup: Command Details ALTER TABLE "<table>" PRELOAD ALL Set the reload flag for all columns of the table ALTER TABLE "<table>" PRELOAD ("<column>") ALTER TABLE "<table>" PRELOAD ("<column1>", ..., "<columnN>") Set the reload flag for the defined columns of the table ALTER TABLE "<table>" PRELOAD NONE Unsets the reload flag for all columns of the table You can check for tables and / or columns with an activated reload flag in the following ways: Columns IS_PRELOAD and IS_PARTIAL_PRELOAD of TABLES Column PRELOAD of TABLE_COLUMNS SQL: "HANA_Tables_ColumnStore_PreloadActive" (SAP Note 1969700) Loading columns flagged with PRELOAD has precedence over columns marked for pre-warming (see below).

Reload after startup (pre-warming based on columns previously loaded) Automatic load of columns on secondary system of system replication environment Automatic load of columns on standby node during auto host failover	The following SAP HANA parameters control column loads during SAP HANA startup and on the secondary system of a system replication scenario based on columns loaded into memory before the shutdown: Parameter Default Unit Details indexserver.ini -> [sql] -> reload_tables true If set to 'true', SAP HANA loads columns into memory during startup, which were located in memory before shutdown. This can be considered as pre-warming in order to make sure that column loads are not required when the table is accessed the first time explicitly. indexserver.ini -> [parallel] -> tables_preloaded_in_parallel 5 Number of tables loaded in parallel after startup A higher value typically results in quicker reloads, but a higher CPU consumption, so it is a trade-off between load time and resource consumption. If you want to adjust it, you should perform some tests to find an optimal value to fulfill your needs. Starting with SAP HANA 2.00.010 this parameter is also considered for reload on secondary system replication sites (SAP Note 1999880). With earlier release levels it is performed sequentially. global.ini -> [system_replication] -> preload_column_tables true Per default SAP HANA loads the columns into the memory of the secondary system of a system replication scenario during normal uptime. This has the advantage that a reload is not required at failover time. If you want to disable this feature (e.g. because only limited memory is available on the secondary side), you can set the preload_column_tables parameter to 'false'. The effect of this parameter depends on the system where it is set: Primary system: Information about loaded tables is collected and persisted in the system replication related snapshot. Secondary system: The load information from primary is evaluated and the tables are loaded accordingly. You can check for tables currently part of this reload information using the following hdbcons command (SAP Note 2222218): tablepreload c -f
Index load / recreation after optimize compression	An implicit index load / creation can happen when all of the following conditions are met: optimize compression is executed for the underlying table (SAP Note 2112604) Explicit or implicit single column index exists for a column (SAP Note 2160391) Compression type of this column is changed during optimize compression run The related call stack modules typically include: AttributeEngine::BlockIndex::setFromSpDocuments AttributeEngine::SpDocuments::buildIndex AttributeEngine::MemoryAvc2::checkIndexCreation AttributeEngine::MemoryAvc2::lazyLoad1 These loads may cause lock situations (e.g. AttributeValueContainer readLock, AttributeValueContainer writeLock) and aren't recorded in monitoring view M_CS_LOADS.

4. How can reload information for startup, host auto failover and system replication be collected?

The following parameter is used to control the collection of reload information:

Parameter	Default	Unit	Details
global.ini -> [persistence] -> tablepreload_write_interval	3600 (Rev. 69 and below) 86400 (Rev. 70 and above)	s	This parameter defines the frequency of collecting table load information for reloads during startup and on the secondary system replication side. Collection of the data is disabled by setting the value to 0. The database trace (SAP Note 2380176) contains warm_upper related messages like the following when preload information is written: ::writeTablePreloadInfo enter <num_tables> tables are open ::writeTablePreloadInfo exit

In exceptional cases you can manually execute the following hdbcons command (SAP Note 2222218) in order to collect the current load state:


```
tablepreload w
```

See SAP Note 1889081 for more information.

5. When do unloads happen?

The following table lists the main reasons for unloads. You can determine the reason of an unload via column REASON of monitoring view M_CS_UNLOADS.

Reason	Details
LOW MEMORY	SAP HANA automatically performs column unloads as part of resource container shrinks when memory becomes scarce. Typical reasons are: A memory request can't be fulfilled because it would exceed the defined SAP HANA allocation limit. SAP HANA <= 1.00.122.12: The resource container reaches the limit defined with parameter global.ini -> [memoryobjects] -> unload_upper_bound (SAP Note 2301382). SAP HANA <= 1.00.122.12: An explicit "resman shrink" is executed with hdbcons (SAP Note 2222218). This is typically critical for performance and should be avoided whenever possible. See SAP Note 1999997 for more information. Starting with SAP HANA 1.00.122.13 situation 2 and 3 are mapped to unload reason SHRINK, see below.
EXPLICIT	Tables can be unloaded explicitly using the following SQL command: UNLOAD "<table_name>" Alternatively you can use SAP HANA Studio for that purpose: SAP HANA Studio -> <system> -> Catalog -> <schema> -> <table> -> "Unload from memory..." Starting with SAP HANA 2.00 it is possible to unload dedicated partitions of a table: UNLOAD "<table_name>" PARTITION <part_id1> [, ..., <part_idN>] Explicit unloads are performed at the end of table imports with R3load (e.g. in context of a system copy) unless they are explicitly switched off with "-omit U".
UNUSED RESOURCE	The following parameters can be used to trigger unloads whenever a column wasn't used for a specific time: Parameter Default Unit Details global.ini -> [memoryobjects] -> unused_retention_period 0 (disabled) s Number of seconds after which an unused object can be unloaded global.ini -> [memoryobjects] -> unused_retention_period_check_interval 7200 s Check frequency for objects exceeding the retention time On table level you can activate an unused retention period with the following command: ALTER TABLE "<table_name>" WITH PARAMETERS ('UNUSED_RETENTION_PERIOD' = '<unused_retention_period_s>') Deactivation is possible via: ALTER TABLE "<table_name>" WITH PARAMETERS ('UNUSED_RETENTION_PERIOD' = '0') Configuring a retention for unloads typically provides no advantage and increases the risk of unnecessary unloads and loads. Therefore these parameters should only be set in exceptional situations.
MERGE	If a column is loaded and unloaded purely for merge reasons, you will find the unload reason MERGE as of SAP HANA SPS 12. See SAP Note 2057046 for more information related to SAP HANA delta merges.
SHRINK	In order to be able to differentiate between OOM situations and resource container shrinks the new unload reason SHRINK was introduced with SAP HANA 1.00.122.13 that covers the following situations: The resource container reaches the limit defined with parameter global.ini -> [memoryobjects] -> unload_upper_bound (SAP Note 2301382). An explicit "resman shrink" is executed with hdbcons (SAP Note 2222218). If a rather low limit is used, an unnecessary high amount of unloads may be performed. You can check with SQL: "HANA_Memory_ResourceContainerConfiguration" (SAP Note 1969700) if the configured settings can be adjusted.

6. In which sequence are columns unloaded?

Usually unloads happen based on a "least recently used" (LRU) approach, so the columns having not being used for the longest time are unloaded first.

If there are tables that should in general be replaced earlier or later, you can prioritize unloads using the UNLOAD PRIORITY setting:

```
ALTER TABLE "<table_name>" UNLOAD PRIORITY <priority>
```

The priority can vary between 0 and 9. Tables with a higher priority are unloaded earlier than tables with a lower priority. SAP HANA considers both the last access time and the unload priority for the proper sequence of unloads, so both factors are important.

The unload priority of a table can be checked via:

```
SELECT UNLOAD_PRIORITY FROM TABLES WHERE TABLE_NAME = '<table_name>'
```

See SAP Note 1999997 ("In which order are objects unloaded from the resource container?") for further details how unload priorities are actually mapped to disposition classes.

7. What are typical unload priorities for tables in SAP environments?

The following table unload priorities are typically used in SAP environments:

Unload priority	Table type	Details
0	Temporary tables System tables	All temporary tables (TABLES.IS_TEMPORARY = 'TRUE', created with NO LOGGING) must not be unloaded and therefore always have unload priority 0. If you want to define a higher unload priority, you receive the following error: SQL error 257: sql syntax error: invalid unload priority for temporary table, only 0 is allowed
5	Default	Per default tables are delivered with a medium unload priority of 5.
7	BW tables (DSO, PSA)	Some BW DSO (/BIC/A*) and PSA (/BIC/B*) can typically be unloaded earlier than other tables and are therefore configured with unload priority 7. Be aware that it depends significantly on the type of DSO and PSA, so it is normal that there are also DSO and PSA tables with unload priority 5.

You can use *SQL: "HANA_Tables_ColumnStore_UnloadPriorities"* (SAP Note 1969700) in order to evaluate existing unload priorities and check for tables with unload priorities different from the standard.

8. How can unloads be monitored?

You can monitor unloads in the following way:

- Monitoring view M_CS_UNLOADS
- SAP HANA Studio -> Administration -> Performance -> Load -> Column Unloads
- *SQL: "HANA_Tables_ColumnStore_UnloadsAndLoads"* (SAP Note 1969700)
- *SQL: "HANA_LoadHistory_Services"* (SAP Note 1969700, SPS 09 and higher)

9. Where is unload information recorded?

Unload information is recorded in unload trace files on disk level. M_CS_UNLOADS is based on these trace files. Typically up to 10 * 10 MB trace files can be written per host and service. As a consequence unload information is still available, even if SAP HANA is restarted.

See SAP Note 2119087 ("Unload trace") for more information.

10. How granular can loads and unloads be performed?

The fines granularities of loads and unloads are:

Mechanism	Loads	Unloads
Manual	per column	per table SAP HANA >= 2.00: per table and partition
Automatic	per column and partition Paged attribute (SAP Note 1871386): per page	per column and partition Paged attribute (SAP Note 1871386): per page

11. How can I check for errors during column loads?

The monitoring view M_CS_LOADS doesn't contain an error information. Whenever a column load is triggered, it is recorded in M_CS_LOADS, independent if it was successful or if it failed (e.g. due to a lack of memory or an inconsistency).

If you want to monitor failed column loads, you can check the trace files for entries starting with "load failed:", e.g.:

```
[224524]{340032}[291/-1] 2016-02-03 18:37:39.190596 e attributes
AttributeValueContainer.cpp(03563) : load failed: exception 1: no.70000000
(AttributeEngine/AttributeStoreFile.h:339)
ste::Exception type AttributeStoreFile error
'I&#01;SAPSR3&#01;T000&#01;A&#01;A&#01;attribute_203.bin': AttributeEngine: error reading file
message additionalInfo $ADDINFO$
exception throw location:
1: 0x00007f43076400c7 in ste::Exception::Exception(char const*, char const*, char const*, int,
char const*)+0x53 at Exception.cpp:12 (libhdbbasement.so)
2: 0x00007f42fc5c8e94 in AttributeEngine::AttributeStoreReadFile::throwError(int, char const*,
long, bool)+0x130 at AttributeStoreFile.h:142 (libhdbcs.so)
```

You can use *SQL: "HANA_TraceFiles_Content" (TRACE_TEXT = 'load failed:%')* available via SAP Note 1969700 for that purpose.

Be aware that load failures with the following error codes can be a consequence of cancellations of sessions while performing a column load (e.g. when connections to the database are established at a time when the column store isn't completely loaded, yet):

```
6900: Attribute engine failed
6923: Attribute load failed
```

This is usually harmless, but you should better avoid it (e.g. by starting the SAP application servers *after* the column store has completely loaded).

12. What is a typical performance of the column store load after startup?

The actual performance of the column store load after startup depends on factors like the configured parallelism (indexserver.ini -> [parallel] -> tables_preloaded_in_parallel), the I/O performance (SAP Note 1999930) and the available system resources. In optimal scenarios, a load throughput of around 200 GB per minute is possible.

Example: (load of 8 TB column store within 40 minutes, 200 GB / minute)

[16899]{-1}[12/-1] 2016-12-04 01:35:35.302222 i TableReload TRexApiSystem.cpp(00628) : Now reloading 45209 tables (loading up to 14 tables in parallel)
[16899]{-1}[12/-1] 2016-12-04 02:15:47.178840 i TableReload TRexApiSystem.cpp(00659) : Finished table reloading

```
-------------------------
|SNAPSHOT_TIME |USED_GB|
-------------------------
|2016/12/04 02:15| 9156|
|2016/12/04 02:14| 9126|
|2016/12/04 02:13| 9085|
|2016/12/04 02:12| 9014|
|2016/12/04 02:11| 8899|
|2016/12/04 02:10| 8700|
|2016/12/04 02:09| 8462|
...
|2016/12/04 01:41| 1586|
|2016/12/04 01:40| 1370|
|2016/12/04 01:39| 1101|
|2016/12/04 01:38| 854|
|2016/12/04 01:37| 680|
|2016/12/04 01:36| 648|
|2016/12/04 01:35| 657|
|2016/12/04 01:34| 625|
|2016/12/04 01:33| 186|
-------------------------
```

Keywords

Columnstore unloads
PRELOAD
IS_PRELOAD
IS_PARTIAL_PRELOAD
reload_tables
tables_preloaded_in_parallel
preload_column_tables
tablepreload
tablepreload_write_interval
invalid unload priority for temporary table
load failed

Products

SAP HANA, platform edition all versions

This document refers to

SAP Note/KBA	Title

2380176	FAQ: SAP HANA Database Trace
2222218	FAQ: SAP HANA Database Server Management Console (hdbcons)
2160391	FAQ: SAP HANA Indexes
2119087	How-To: Configuring SAP HANA Traces
2112604	FAQ: SAP HANA Compression
2057046	FAQ: SAP HANA Delta Merges
1999997	FAQ: SAP HANA Memory
1999993	How-To: Interpreting SAP HANA Mini Check Results
1999930	FAQ: SAP HANA I/O Analysis
1999880	FAQ: SAP HANA System Replication
1994962	How-To: Activation of Hybrid LOBs in SAP HANA
1977207	How to handle HANA Alert 55: Columnstore unloads
2301382	Increased Used Memory Size due to Pool/PersistenceManager/PersistentSpace/DefaultLPA /Page (Rev. 110 - 122.05)
1969700	SQL Statement Collection for SAP HANA

