Removing Data Permanently from Vertica and Advanced Projection Design

In Vertica, you have a couple different options for removing data from your database. You can:

* Perform a DELETE and wait for the Tuple Mover(TM) to perform a MERGEOUT
* Manually purge the data
* Drop partitions
* Truncate information from a table

Epoch : A 64-bit number that represents a logical time stamp for the data in Vertica. Every row has an implicitly stored column that records the committed epoch.

**CURRENT EPOCH**:The open epoch that becomes the latest epoch (LE) after a COMMIT operation. The current\_epoch at the time of the COMMIT is the epoch for that DML.

**AHM**:Marks the epoch prior to which data marked for deletion can be purged from the system. The data marked for deletion prior to the AHM will be permanently deleted (purged) during mergeout.

**LAST GOOD AEPOCH** :LGE is the epoch at which all data has been written from WOS to ROS

**Checkpoint Epoch** :Every time the Tuple Mover performs a move-out operation for a given projection, it records a Checkpoint Epoch for that projection, representing an epoch up to which the projection has no data in the WOS. The minimum checkpoint epoch across all projections on that node is called the node’s **Checkpoint Epoch**. It represents a point in time up to which all the data was moved out to disk. If a K-safe=1 database experiences a single-node failure, the node's recovery process attempts to rebuild the data beyond the Checkpoint Epochs from other nodes. If all nodes fail, such as during a power outage, HP Vertica recovers the database back to the minimum Checkpoint Epochs across all the nodes, known as the Last Good Epoch (LGE).

**Refer: attached pdf file :BP\_understanding\_Epochs.pdf**

**Delete process:**

https://my.vertica.com/get-started-vertica/removing-data/

**delete life cycle : Describe the process used to mark records for deletion**

\

­­<https://my.vertica.com/kb/Understanding-Vertica-Epochs/Content/BestPractices/Understanding-Vertica-Epochs.htm?Highlight=epoch>

https://my.vertica.com/get-started-vertica/removing-data/

Merge join and hash join difference

|  |  |
| --- | --- |
| **Merge Join** | **Hash join** |
| Optimiser chooses merge join when projections are sorted on join colums | Optimiser chooses merge join when projections are sorted on join colums |
| merge join uses sorter requirement so it uses less memory | A hash join has no sort requirement, but it consumes more memory because a hash table is built with the values in the inner table.The cost of performing a hash join is low if the entire hash table can fit in memory, but the cost rises when the hash table is written to disk. The optimizer chooses a hash join when projections are not sorted on the join columns |

**Identify the advantages and disadvantages of using delete vectors to identify records marked for deletion**

**Advantage**: When you run a DELETE command, Vertica locates the records you want to remove, and marks them for deletion using a delete vector. A delete vector is a set of tuples that record where (position) and when (epoch number) a row is deleted.

When the Tuple Mover(TM) performs its next MERGEOUT operation (default is every ten minutes), it automatically purge eligible data. The purging process removes deleted data from physical storage so you can reuse the disk space. Data is eligible for purging as long as the:

* Epoch of deletion is BEFORE the AHM at time of purging.
* Data belongs to an active partition

**Disadvantages**: When we have huge delete (more than 10% of table ) then delete vector consumes lot of disk space .

**9.Determine methods available to optimize a projection for delete processing**

Refer below link

<https://my.vertica.com/docs/8.0.x/HTML/index.htm#Authoring/AnalyzingData/Optimizations/OptimizingDELETEsAndUPDATEsForPerformance.htm>

**Distinguish between deleting partitions and deleting records**

https://my.vertica.com/docs/7.0.x/HTML/index.htm#Authoring/AdministratorsGuide/Partitions/DroppingPartitions.htm

**Identify the advantages of a group by pipe versus a group by hash**

<https://my.vertica.com/docs/7.1.x/HTML/index.htm#Authoring/AnalyzingData/Optimizations/AvoidingGROUPBYHASHWithProjectionDesign.htm%3FTocPath%3DAnalyzing%2520Data|Optimizing%2520Query%2520Performance|Optimizing%2520GROUP%2520BY%2520Queries|_____1>