SQL Server 2012: Nonclustered Columnstore Indexes

Module 5: Managing Data Modifications

Joe Sack

Joe@SQLskills.com

pluralsight
hardcore developer training

Module Introduction

- Once you’ve created a nonclustered columnstore index on a table, certain data modification operations are no longer allowed

- While this can be a significant limitation, there are techniques that can be used to still keep data up-to-date

- This module will review these data modification techniques

Restricted Data Modifications

- **In SQL Server 2012, adding a nonclustered columnstore index locks down specific data modification capabilities**

- **You cannot perform:**
  - INSERT
  - UPDATE
  - DELETE
  - MERGE
  - Bcp and BULK INSERT

Delta Table

- **Create a “logical” fact table made up of:**
  - A nonclustered columnstore index fact table
  - A “delta” table containing the most recent data, without a columnstore index

- **UNION ALL with a view or common table expression (CTE)**

- **Considerations:**
  - Good performance for a large fact table and a small delta table
  - Requires end-user references to the logical fact table
  - You still need to periodically move delta table rows to the columnstore index fact table
  - UNION ALL can inhibit batch execution mode

Drop, Load, Create

- **Another option is simply to periodically drop the nonclustered columnstore index, load the data, and then recreate the index**

- **This is a viable option *if:***
  - The columnstore index creation time period is acceptable
  - Reporting data doesn’t need to be “real time”
  - You have sufficient resources during the index creation operations

Partition Switching

- **If your relational data warehousing schema is already leveraging native table partitioning, you can perform partition switching**

- **Switch in new data from a staging table and if permitted, switch out data which can be removed**

- **Considerations:**
  - If you're already using table partitioning, you can create staging tables for loading periodically to the fact table covered by the columnstore index
  - The columnstore index has to be partition-aligned
  - The staging table schema and supporting indexes must match the fact table
  - If you're *not* already using table partitioning, you'll need to carefully consider if this is the best option and ensure it makes sense
    - Don't rush into a design without understanding the performance and manageability characteristics