View column store storage details

Note:

You are not required to complete the processes, tasks, activities, or steps presented in this example. The various samples provided are for illustrative purposes only and it's likely that if you try this out you will encounter issues in your system.

To view column store storage details, perform the following steps

1. Run the following query to create the **vColumnStoreRowGroupStats**:

```
 1
 2
 3
 4
 5
 6
 7
 8
 9
10
11
12
13
14
15
16
17
18
19
20
21
22
```

```sql
create view [wwi_perf].[vColumnStoreRowGroupStats]
as
with cte
as
(
select    tb.[name]                    AS [logical_table_name]
,         rg.[row_group_id]            AS [row_group_id]
,         rg.[state]                   AS [state]
,         rg.[state_desc]              AS [state_desc]
,         rg.[total_rows]              AS [total_rows]
,         rg.[trim_reason_desc]        AS trim_reason_desc
,         mp.[physical_name]           AS physical_name
FROM      sys.[schemas] sm
```

```
JOIN    sys.[tables] tb                ON   sm.[schema_id]           = tb.
[schema_id]
JOIN    sys.[pdw_table_mappings] mp    ON   tb.[object_id]           = mp.
[object_id]
JOIN    sys.[pdw_nodes_tables] nt      ON   nt.[name]                = mp.
[physical_name]
JOIN    sys.[dm_pdw_nodes_db_column_store_row_group_physical_stats] rg      ON
rg.[object_id]     = nt.[object_id]
                                                                              AND
rg.[pdw_node_id]   = nt.[pdw_node_id]
                                      AND rg.[distribution_id]    = nt.
[distribution_id]
)
select *
from cte;
```
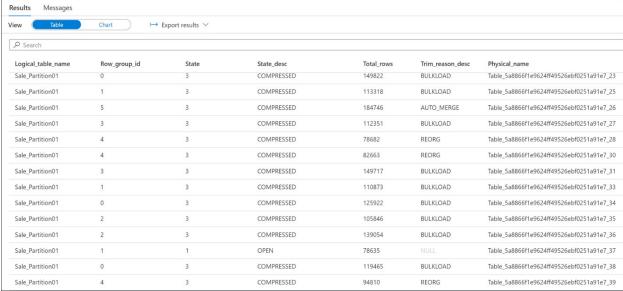
In this query we are using the
sys.dm_pdw_nodes_db_column_store_row_group_physical_stats DMV which
provides current rowgroup-level information about all of the columnstore indexes in the current
database.

2. Explore the statistics of the columnstore for the **Sale_Partition01** table using the
following query:

```
1
2
3
4
5
6
```

```
SELECT
    *
FROM
    [wwi_perf].[vColumnStoreRowGroupStats]
WHERE
    Logical_Table_Name = 'Sale_Partition01'
```

3. Explore the results of the query:

Column store row group statistics for Sale_Partition01

Browse through the results and get an overview of the rowgroup states. Notice the **COMPRESSED** and **OPEN** states of some of the row groups.

4. Explore the statistics of the columnstore for the **Sale_Hash_Ordered** table using the same query:

```
1
2
3
4
5
6
SELECT
    *
FROM
    [wwi_perf].[vColumnStoreRowGroupStats]
WHERE
    Logical_Table_Name = 'Sale_Hash_Ordered'
```

5. Explore the results of the query:

Logical_table_name	Row_group_id	State	State_desc	Total_rows	Trim_reason_desc	Physical_name
Sale_Hash_Ordered	5	3	COMPRESSED	397731	BULKLOAD	Table_426afa17d3444d18b5173cbec7ae54a5_58
Sale_Hash_Ordered	5	3	COMPRESSED	454797	BULKLOAD	Table_426afa17d3444d18b5173cbec7ae54a5_59
Sale_Hash_Ordered	5	3	COMPRESSED	402974	BULKLOAD	Table_426afa17d3444d18b5173cbec7ae54a5_60
Sale_Hash_Ordered	4	3	COMPRESSED	1048576	NO_TRIM	Table_426afa17d3444d18b5173cbec7ae54a5_1
Sale_Hash_Ordered	4	3	COMPRESSED	1048576	NO_TRIM	Table_426afa17d3444d18b5173cbec7ae54a5_2
Sale_Hash_Ordered	4	3	COMPRESSED	1048576	NO_TRIM	Table_426afa17d3444d18b5173cbec7ae54a5_3
Sale_Hash_Ordered	4	3	COMPRESSED	1048576	NO_TRIM	Table_426afa17d3444d18b5173cbec7ae54a5_4
Sale_Hash_Ordered	4	3	COMPRESSED	1048576	NO_TRIM	Table_426afa17d3444d18b5173cbec7ae54a5_5
Sale_Hash_Ordered	4	3	COMPRESSED	1048576	NO_TRIM	Table_426afa17d3444d18b5173cbec7ae54a5_6
Sale_Hash_Ordered	4	3	COMPRESSED	1048576	NO_TRIM	Table_426afa17d3444d18b5173cbec7ae54a5_7
Sale_Hash_Ordered	4	3	COMPRESSED	1048576	NO_TRIM	Table_426afa17d3444d18b5173cbec7ae54a5_8
Sale_Hash_Ordered	4	3	COMPRESSED	1048576	NO_TRIM	Table_426afa17d3444d18b5173cbec7ae54a5_9
Sale_Hash_Ordered	4	3	COMPRESSED	1048576	NO_TRIM	Table_426afa17d3444d18b5173cbec7ae54a5_10

Column store row group statistics for Sale_Hash_Ordered

There is a significant difference in the rowgroup states from the previous one. This highlights one of the potential advantages of ordered CCIs.