Exercise - View column store storage details

3 minutes

To view column store storage details, perform the following steps

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

```
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;
```

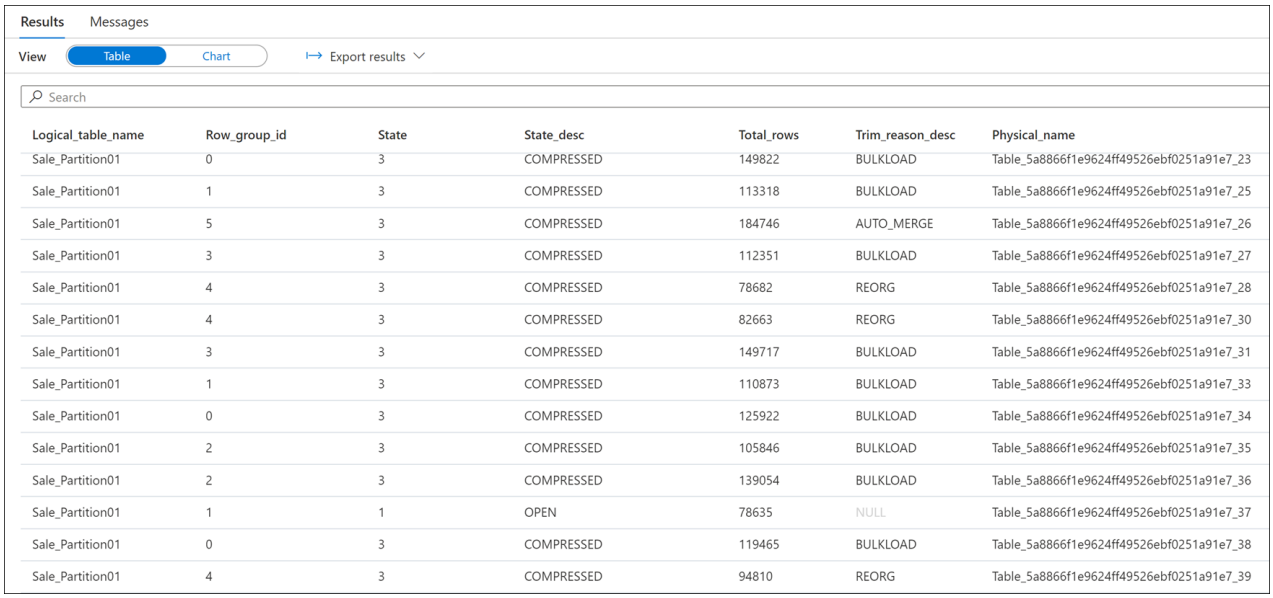
In this query we are using the `sys.dm_pdw_nodes_db_column_store_row_group_physical_stats` DMV which provides current rowgroup-level information about all of the columnstore indexes in the current database.

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

```
SQL

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

3. Explore the results of the query:



Results | Messages

View: Table | Chart | Export results

Search

Logical_table_name	Row_group_id	State	State_desc	Total_rows	Trim_reason_desc	Physical_name
Sale_Partition01	0	3	COMPRESSED	149822	BULKLOAD	Table_5a8866f1e9624ff49526ebf0251a91e7_23
Sale_Partition01	1	3	COMPRESSED	113318	BULKLOAD	Table_5a8866f1e9624ff49526ebf0251a91e7_25
Sale_Partition01	5	3	COMPRESSED	184746	AUTO_MERGE	Table_5a8866f1e9624ff49526ebf0251a91e7_26
Sale_Partition01	3	3	COMPRESSED	112351	BULKLOAD	Table_5a8866f1e9624ff49526ebf0251a91e7_27
Sale_Partition01	4	3	COMPRESSED	78682	REORG	Table_5a8866f1e9624ff49526ebf0251a91e7_28
Sale_Partition01	4	3	COMPRESSED	82663	REORG	Table_5a8866f1e9624ff49526ebf0251a91e7_30
Sale_Partition01	3	3	COMPRESSED	149717	BULKLOAD	Table_5a8866f1e9624ff49526ebf0251a91e7_31
Sale_Partition01	1	3	COMPRESSED	110873	BULKLOAD	Table_5a8866f1e9624ff49526ebf0251a91e7_33
Sale_Partition01	0	3	COMPRESSED	125922	BULKLOAD	Table_5a8866f1e9624ff49526ebf0251a91e7_34
Sale_Partition01	2	3	COMPRESSED	105846	BULKLOAD	Table_5a8866f1e9624ff49526ebf0251a91e7_35
Sale_Partition01	2	3	COMPRESSED	139054	BULKLOAD	Table_5a8866f1e9624ff49526ebf0251a91e7_36
Sale_Partition01	1	1	OPEN	78635	NULL	Table_5a8866f1e9624ff49526ebf0251a91e7_37
Sale_Partition01	0	3	COMPRESSED	119465	BULKLOAD	Table_5a8866f1e9624ff49526ebf0251a91e7_38
Sale_Partition01	4	3	COMPRESSED	94810	REORG	Table_5a8866f1e9624ff49526ebf0251a91e7_39

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:

```
SQL

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

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