**We are going to discuess about below concept:**

**Improve the execution plan of a query with a materialized view**

1. Run the query with the EXPLAIN directive (note the WITH\_RECOMMENDATIONS option as well):

EXPLAIN WITH\_RECOMMENDATIONS

SELECT

T.TransactionItemsCountBucket

,count(\*) as CustomersCount

FROM

(

SELECT

CustomerId,

(

COUNT(\*) -

(

SELECT

MIN(TransactionItemsCount)

FROM

(

SELECT

COUNT(\*) as TransactionItemsCount

FROM

[wwi\_perf].[Sale\_Hash]

GROUP BY

CustomerId

) X

)

) / 100 as TransactionItemsCountBucket

FROM

[wwi\_perf].[Sale\_Hash]

GROUP BY

CustomerId

) T

GROUP BY

T.TransactionItemsCountBucket

ORDER BY

T.TransactionItemsCountBucket

1. Analyze the resulting execution plan. Take a close look to the <materialized\_view\_candidates> section which suggests possible materialized views you can create to improve the performance of the query.

<?xml version="1.0" encoding="utf-8"?>

<dsql\_query number\_nodes="5" number\_distributions="60" number\_distributions\_per\_node="12">

<sql>SELECT

T.TransactionItemsCountBucket

,count(\*) as CustomersCount

FROM

(

SELECT

CustomerId,

(

COUNT(\*) -

(

SELECT

MIN(TransactionItemsCount)

FROM

(

SELECT

COUNT(\*) as TransactionItemsCount

FROM

[wwi\_perf].[Sale\_Hash]

GROUP BY

CustomerId

) X

)

) / 100 as TransactionItemsCountBucket

FROM

[wwi\_perf].[Sale\_Hash]

GROUP BY

CustomerId

) T

GROUP BY

T.TransactionItemsCountBucket

ORDER BY

T.TransactionItemsCountBucket</sql>

<materialized\_view\_candidates>

<materialized\_view\_candidates with\_constants="False">CREATE MATERIALIZED VIEW View1 WITH (DISTRIBUTION = HASH([Expr0])) AS

SELECT [SQLPool01].[wwi\_perf].[Sale\_Hash].[CustomerId] AS [Expr0],

COUNT(\*) AS [Expr1]

FROM [wwi\_perf].[Sale\_Hash]

GROUP BY [SQLPool01].[wwi\_perf].[Sale\_Hash].[CustomerId]</materialized\_view\_candidates>

</materialized\_view\_candidates>

<dsql\_operations total\_cost="0.0242811172881356" total\_number\_operations="9">

<dsql\_operation operation\_type="RND\_ID">

<identifier>TEMP\_ID\_99</identifier>

</dsql\_operation>

<dsql\_operation operation\_type="ON">

<location permanent="false" distribution="AllComputeNodes" />

<sql\_operations>

<sql\_operation type="statement">CREATE TABLE [qtabledb].[dbo].[TEMP\_ID\_99] ([col] INT ) WITH(DISTRIBUTED\_MOVE\_FILE='');</sql\_operation>

</sql\_operations>

</dsql\_operation>

<dsql\_operation operation\_type="BROADCAST\_MOVE">

<operation\_cost cost="0.00096" accumulative\_cost="0.00096" average\_rowsize="4" output\_rows="1" GroupNumber="69" />

<source\_statement>SELECT [T1\_1].[col] AS [col] FROM (SELECT MIN([T2\_1].[col]) AS [col] FROM (SELECT COUNT(CAST ((0) AS INT)) AS [col], 0 AS [col1] FROM [SQLPool01].[wwi\_perf].[Sale\_Hash] AS T3\_1 GROUP BY [T3\_1].[CustomerId]) AS T2\_1 GROUP BY [T2\_1].[col1]) AS T1\_1

OPTION (MAXDOP 6, MIN\_GRANT\_PERCENT = [MIN\_GRANT], DISTRIBUTED\_MOVE(N''))</source\_statement>

<destination\_table>[TEMP\_ID\_99]</destination\_table>

</dsql\_operation>

<dsql\_operation operation\_type="RND\_ID">

<identifier>TEMP\_ID\_100</identifier>

</dsql\_operation>

<dsql\_operation operation\_type="ON">

<location permanent="false" distribution="AllDistributions" />

<sql\_operations>

<sql\_operation type="statement">CREATE TABLE [qtabledb].[dbo].[TEMP\_ID\_100] ([col] INT, [col1] BIGINT ) WITH(DISTRIBUTED\_MOVE\_FILE='');</sql\_operation>

</sql\_operations>

</dsql\_operation>

<dsql\_operation operation\_type="SHUFFLE\_MOVE">

<operation\_cost cost="0.0233211172881356" accumulative\_cost="0.0242811172881356" average\_rowsize="12" output\_rows="95.5518" GroupNumber="75" />

<source\_statement>SELECT [T1\_1].[col1] AS [col], [T1\_1].[col] AS [col1] FROM (SELECT COUNT\_BIG(CAST ((0) AS INT)) AS [col], [T2\_1].[col] AS [col1] FROM (SELECT (([T3\_2].[col] - [T3\_1].[col]) / CAST ((100) AS INT)) AS [col] FROM (SELECT MIN([T4\_1].[col]) AS [col] FROM [qtabledb].[dbo].[TEMP\_ID\_99] AS T4\_1) AS T3\_1 INNER JOIN

(SELECT COUNT(CAST ((0) AS INT)) AS [col] FROM [SQLPool01].[wwi\_perf].[Sale\_Hash] AS T4\_1 GROUP BY [T4\_1].[CustomerId]) AS T3\_2

ON (0 = 0)) AS T2\_1 GROUP BY [T2\_1].[col]) AS T1\_1

OPTION (MAXDOP 6, MIN\_GRANT\_PERCENT = [MIN\_GRANT], DISTRIBUTED\_MOVE(N''))</source\_statement>

<destination\_table>[TEMP\_ID\_100]</destination\_table>

<shuffle\_columns>col;</shuffle\_columns>

</dsql\_operation>

<dsql\_operation operation\_type="RETURN">

<location distribution="AllDistributions" />

<select>SELECT [T1\_1].[col1] AS [col], [T1\_1].[col] AS [col1] FROM (SELECT CONVERT (INT, [T2\_1].[col], 0) AS [col], [T2\_1].[col1] AS [col1] FROM (SELECT ISNULL([T3\_1].[col], CONVERT (BIGINT, 0, 0)) AS [col], [T3\_1].[col1] AS [col1] FROM (SELECT SUM([T4\_1].[col1]) AS [col], [T4\_1].[col] AS [col1] FROM [qtabledb].[dbo].[TEMP\_ID\_100] AS T4\_1 GROUP BY [T4\_1].[col]) AS T3\_1) AS T2\_1) AS T1\_1 ORDER BY [T1\_1].[col1] ASC

OPTION (MAXDOP 6, MIN\_GRANT\_PERCENT = [MIN\_GRANT])</select>

</dsql\_operation>

<dsql\_operation operation\_type="ON">

<location permanent="false" distribution="AllDistributions" />

<sql\_operations>

<sql\_operation type="statement">DROP TABLE [qtabledb].[dbo].[TEMP\_ID\_100]</sql\_operation>

</sql\_operations>

</dsql\_operation>

<dsql\_operation operation\_type="ON">

<location permanent="false" distribution="AllComputeNodes" />

<sql\_operations>

<sql\_operation type="statement">DROP TABLE [qtabledb].[dbo].[TEMP\_ID\_99]</sql\_operation>

</sql\_operations>

</dsql\_operation>

</dsql\_operations>

</dsql\_query>

1. Create the suggested materialized view:

CREATE MATERIALIZED VIEW

mvTransactionItemsCounts

WITH

(

DISTRIBUTION = HASH([CustomerId])

)

AS

SELECT

CustomerId

,COUNT(\*) AS ItemsCount

FROM

[wwi\_perf].[Sale\_Hash]

GROUP BY

CustomerId

1. Check the execution plan again:

EXPLAIN WITH\_RECOMMENDATIONS

SELECT

T.TransactionItemsCountBucket

,count(\*) as CustomersCount

FROM

(

SELECT

CustomerId,

(

COUNT(\*) -

(

SELECT

MIN(TransactionItemsCount)

FROM

(

SELECT

COUNT(\*) as TransactionItemsCount

FROM

[wwi\_perf].[Sale\_Hash]

GROUP BY

CustomerId

) X

)

) / 100 as TransactionItemsCountBucket

FROM

[wwi\_perf].[Sale\_Hash]

GROUP BY

CustomerId

) T

GROUP BY

T.TransactionItemsCountBucket

ORDER BY

T.TransactionItemsCountBucket

The resulting execution plan indicates now the use of the mvTransactionItemsCounts (the BROADCAST\_MOVE distributed SQL operation) materialized view which provides improvements to the query execution time:

<?xml version="1.0" encoding="utf-8"?>

<dsql\_query number\_nodes="5" number\_distributions="60" number\_distributions\_per\_node="12">

<sql>SELECT

T.TransactionItemsCountBucket

,count(\*) as CustomersCount

FROM

(

SELECT

CustomerId,

(

COUNT(\*) -

(

SELECT

MIN(TransactionItemsCount)

FROM

(

SELECT

COUNT(\*) as TransactionItemsCount

FROM

[wwi\_perf].[Sale\_Hash]

GROUP BY

CustomerId

) X

)

) / 100 as TransactionItemsCountBucket

FROM

[wwi\_perf].[Sale\_Hash]

GROUP BY

CustomerId

) T

GROUP BY

T.TransactionItemsCountBucket

ORDER BY

T.TransactionItemsCountBucket</sql>

<materialized\_view\_candidates>

<materialized\_view\_candidates with\_constants="False">CREATE MATERIALIZED VIEW View1 WITH (DISTRIBUTION = HASH([Expr0])) AS

SELECT [SQLPool01].[wwi\_perf].[Sale\_Hash].[CustomerId] AS [Expr0],

COUNT(\*) AS [Expr1]

FROM [wwi\_perf].[Sale\_Hash]

GROUP BY [SQLPool01].[wwi\_perf].[Sale\_Hash].[CustomerId]</materialized\_view\_candidates>

</materialized\_view\_candidates>

<dsql\_operations total\_cost="0.0242811172881356" total\_number\_operations="9">

<dsql\_operation operation\_type="RND\_ID">

<identifier>TEMP\_ID\_111</identifier>

</dsql\_operation>

<dsql\_operation operation\_type="ON">

<location permanent="false" distribution="AllComputeNodes" />

<sql\_operations>

<sql\_operation type="statement">CREATE TABLE [qtabledb].[dbo].[TEMP\_ID\_111] ([col] INT ) WITH(DISTRIBUTED\_MOVE\_FILE='');</sql\_operation>

</sql\_operations>

</dsql\_operation>

<dsql\_operation operation\_type="BROADCAST\_MOVE">

<operation\_cost cost="0.00096" accumulative\_cost="0.00096" average\_rowsize="4" output\_rows="1" GroupNumber="134" />

<source\_statement>SELECT [T1\_1].[col] AS [col] FROM (SELECT MIN([T2\_1].[col]) AS [col] FROM (SELECT CONVERT (INT, [T3\_1].[col], 0) AS [col], 0 AS [col1] FROM (SELECT ISNULL([T4\_1].[col], CONVERT (BIGINT, 0, 0)) AS [col] FROM (SELECT SUM([T5\_1].[ItemsCount]) AS [col] FROM (SELECT [T6\_1].[CustomerId] AS [CustomerId], [T6\_1].[ItemsCount] AS [ItemsCount] FROM [SQLPool01].[dbo].[mvTransactionItemsCounts] AS T6\_1) AS T5\_1 GROUP BY [T5\_1].[CustomerId]) AS T4\_1) AS T3\_1 WHERE ([T3\_1].[col] != CAST ((0) AS BIGINT))) AS T2\_1 GROUP BY [T2\_1].[col1]) AS T1\_1

OPTION (MAXDOP 6, MIN\_GRANT\_PERCENT = [MIN\_GRANT], DISTRIBUTED\_MOVE(N''))</source\_statement>

<destination\_table>[TEMP\_ID\_111]</destination\_table>

</dsql\_operation>

<dsql\_operation operation\_type="RND\_ID">

<identifier>TEMP\_ID\_112</identifier>

</dsql\_operation>

<dsql\_operation operation\_type="ON">

<location permanent="false" distribution="AllDistributions" />

<sql\_operations>

<sql\_operation type="statement">CREATE TABLE [qtabledb].[dbo].[TEMP\_ID\_112] ([col] INT, [col1] BIGINT ) WITH(DISTRIBUTED\_MOVE\_FILE='');</sql\_operation>

</sql\_operations>

</dsql\_operation>

<dsql\_operation operation\_type="SHUFFLE\_MOVE">

<operation\_cost cost="0.0233211172881356" accumulative\_cost="0.0242811172881356" average\_rowsize="12" output\_rows="95.5518" GroupNumber="140" />

<source\_statement>SELECT [T1\_1].[col1] AS [col], [T1\_1].[col] AS [col1] FROM (SELECT COUNT\_BIG(CAST ((0) AS INT)) AS [col], [T2\_1].[col] AS [col1] FROM (SELECT (([T3\_2].[col] - [T3\_1].[col]) / CAST ((100) AS INT)) AS [col] FROM (SELECT MIN([T4\_1].[col]) AS [col] FROM [qtabledb].[dbo].[TEMP\_ID\_111] AS T4\_1) AS T3\_1 INNER JOIN

(SELECT CONVERT (INT, [T4\_1].[col], 0) AS [col] FROM (SELECT ISNULL([T5\_1].[col], CONVERT (BIGINT, 0, 0)) AS [col] FROM (SELECT SUM([T6\_1].[ItemsCount]) AS [col] FROM (SELECT [T7\_1].[CustomerId] AS [CustomerId], [T7\_1].[ItemsCount] AS [ItemsCount] FROM [SQLPool01].[dbo].[mvTransactionItemsCounts] AS T7\_1) AS T6\_1 GROUP BY [T6\_1].[CustomerId]) AS T5\_1) AS T4\_1 WHERE ([T4\_1].[col] != CAST ((0) AS BIGINT))) AS T3\_2

ON (0 = 0)) AS T2\_1 GROUP BY [T2\_1].[col]) AS T1\_1

OPTION (MAXDOP 6, MIN\_GRANT\_PERCENT = [MIN\_GRANT], DISTRIBUTED\_MOVE(N''))</source\_statement>

<destination\_table>[TEMP\_ID\_112]</destination\_table>

<shuffle\_columns>col;</shuffle\_columns>

</dsql\_operation>

<dsql\_operation operation\_type="RETURN">

<location distribution="AllDistributions" />

<select>SELECT [T1\_1].[col1] AS [col], [T1\_1].[col] AS [col1] FROM (SELECT CONVERT (INT, [T2\_1].[col], 0) AS [col], [T2\_1].[col1] AS [col1] FROM (SELECT ISNULL([T3\_1].[col], CONVERT (BIGINT, 0, 0)) AS [col], [T3\_1].[col1] AS [col1] FROM (SELECT SUM([T4\_1].[col1]) AS [col], [T4\_1].[col] AS [col1] FROM [qtabledb].[dbo].[TEMP\_ID\_112] AS T4\_1 GROUP BY [T4\_1].[col]) AS T3\_1) AS T2\_1) AS T1\_1 ORDER BY [T1\_1].[col1] ASC

OPTION (MAXDOP 6, MIN\_GRANT\_PERCENT = [MIN\_GRANT])</select>

</dsql\_operation>

<dsql\_operation operation\_type="ON">

<location permanent="false" distribution="AllDistributions" />

<sql\_operations>

<sql\_operation type="statement">DROP TABLE [qtabledb].[dbo].[TEMP\_ID\_112]</sql\_operation>

</sql\_operations>

</dsql\_operation>

<dsql\_operation operation\_type="ON">

<location permanent="false" distribution="AllComputeNodes" />

<sql\_operations>

<sql\_operation type="statement">DROP TABLE [qtabledb].[dbo].[TEMP\_ID\_111]</sql\_operation>

</sql\_operations>

</dsql\_operation>

</dsql\_operations>

</dsql\_query>