1) Verify Active Connections

```
SELECT * FROM sys.dm_pdw_exec_sessions where status <> 'Closed' and session_id <> session_id();
```

2) Monitor active queries

```
SELECT *1
FROM sys.dm_pdw_exec_requests
WHERE status not in ('Completed','Failed','Cancelled')
  AND session_id <> session_id()
ORDER BY submit_time DESC;
```

3) Find top 10 queries longest running queries

```
SELECT TOP 10 *
FROM sys.dm_pdw_exec_requests
ORDER BY total_elapsed_time DESC;
```

4) Monitor rollback

```
--If your queries are failing or taking a long time to proceed, you can check and monitor if you have any transactions rolling back.

SELECT
    SUM(CASE WHEN t.database_transaction_next_undo_lsn IS NOT NULL THEN 1 ELSE 0 END),
    t.pdw_node_id,
    nod.[type]
FROM sys.dm_pdw_nodes_tran_database_transactions t
JOIN sys.dm_pdw_nodes nod ON t.pdw_node_id = nod.pdw_node_id
GROUP BY t.pdw_node_id, nod.[type]
```

5) Monitor tempdb

```
SELECT
    sr.request_id,
    ssu.session_id,
    ssu.pdw_node_id,
    sr.command,
    sr.total_elapsed_time,
    exs.login_name AS 'LoginName',
    DB_NAME(ssu.database_id) AS 'DatabaseName',
    (es.memory_usage * 8) AS 'MemoryUsage (in KB)',
    (ssu.user_objects_alloc_page_count * 8) AS 'Space Allocated For User Objects (in KB)',
    (ssu.user_objects_dealloc_page_count * 8) AS 'Space Deallocated For User Objects (in KB)',
    (ssu.internal_objects_alloc_page_count * 8) AS 'Space Allocated For Internal Objects (in KB)',
    (ssu.internal_objects_dealloc_page_count * 8) AS 'Space Deallocated For Internal Objects (in KB)',
    CASE es.is_user_process
    WHEN 1 THEN 'User Session'
    WHEN 0 THEN 'System Session'
    END AS 'SessionType',
    es.row_count AS 'RowCount'
FROM sys.dm_pdw_nodes_db_session_space_usage AS ssu
    INNER JOIN sys.dm_pdw_nodes_exec_sessions AS es ON ssu.session_id = es.session_id AND ssu.pdw_node_id = es.pdw_node_id
    INNER JOIN sys.dm_pdw_nodes_exec_connections AS er ON ssu.session_id = er.session_id AND ssu.pdw_node_id = er.pdw_node_id
    INNER JOIN microsoft.vw_sql_requests AS sr ON ssu.session_id = sr.spid AND ssu.pdw_node_id = sr.pdw_node_id
    LEFT JOIN sys.dm_pdw_exec_requests exr on exr.request_id = sr.request_id
    LEFT JOIN sys.dm_pdw_exec_sessions exs on exr.session_id = exs.session_id
WHERE DB_NAME(ssu.database_id) = 'tempdb'
    AND es.session_id <> @@SPID
    AND es.login_name <> 'sa'
ORDER BY sr.request_id;
```

6) Transaction log size

```
--The following query returns the transaction log size on each distribution.
--If one of the log files is reaching 160 GB, you should consider scaling up your instance or limiting your transaction size.

SELECT
  instance_name as distribution_db,
  cntr_value*1.0/1048576 as log_file_size_used_GB,
  pdw_node_id
FROM sys.dm_pdw_nodes_os_performance_counters
WHERE
instance_name like 'Distribution_%'
AND counter_name = 'Log File(s) Used Size (KB)'


-- To track bytes and files
SELECT
    r.command,
    s.request_id,
    r.status,
    count(distinct input_name) as nbr_files,
    sum(s.bytes_processed)/1024/1024/1024 as gb_processed
FROM
    sys.dm_pdw_exec_requests r
    inner join sys.dm_pdw_dms_external_work s
        on r.request_id = s.request_id
GROUP BY
    r.command,
    s.request_id,
    r.status
ORDER BY
    nbr_files desc,
    gb_processed desc;
```

7) Memory consumption

—The following query returns SQL Server memory usage and memory pressure per node

```
 SELECT
  pc1.cntr_value as Curr_Mem_KB,
  pc1.cntr_value/1024.0 as Curr_Mem_MB,
  (pc1.cntr_value/1048576.0) as Curr_Mem_GB,
  pc2.cntr_value as Max_Mem_KB,
  pc2.cntr_value/1024.0 as Max_Mem_MB,
  (pc2.cntr_value/1048576.0) as Max_Mem_GB,
  pc1.cntr_value * 100.0/pc2.cntr_value AS Memory_Utilization_Percentage,
  pc1.pdw_node_id
FROM
-- pc1: current memory
sys.dm_pdw_nodes_os_performance_counters AS pc1
-- pc2: total memory allowed for this SQL instance
JOIN sys.dm_pdw_nodes_os_performance_counters AS pc2
ON pc1.object_name = pc2.object_name AND pc1.pdw_node_id = pc2.pdw_node_id
WHERE
pc1.counter_name = 'Total Server Memory (KB)'
AND pc2.counter_name = 'Target Server Memory (KB)'
```

8) Table size queries

-- Dynamic management views (DMVs) show more detail than DBCC commands. Start by creating this view:

```
CREATE VIEW dbo.vTableSizes
AS
WITH base
AS
(
SELECT
 GETDATE()                                                     AS  [execution_time]
, DB_NAME()                                                    AS  [database_name]
, s.name                                                       AS  [schema_name]
, t.name                                                       AS  [table_name]
, QUOTENAME(s.name)+'.'+QUOTENAME(t.name)                      AS  [two_part_name]
, nt.[name]                                                    AS  [node_table_name]
, ROW_NUMBER() OVER(PARTITION BY nt.[name] ORDER BY (SELECT NULL))     AS  [node_table_name_seq]
, tp.[distribution_policy_desc]                                AS  [distribution_policy_name]
, c.[name]                                                     AS  [distribution_column]
, nt.[distribution_id]                                         AS  [distribution_id]
, i.[type]                                                     AS  [index_type]
, i.[type_desc]                                                AS  [index_type_desc]
, nt.[pdw_node_id]                                             AS  [pdw_node_id]
, pn.[type]                                                    AS  [pdw_node_type]
, pn.[name]                                                    AS  [pdw_node_name]
, di.name                                                      AS  [dist_name]
, di.position                                                  AS  [dist_position]
, nps.[partition_number]                                       AS  [partition_nmbr]
, nps.[reserved_page_count]                                    AS  [reserved_space_page_count]
, nps.[reserved_page_count] - nps.[used_page_count]            AS  [unused_space_page_count]
, nps.[in_row_data_page_count]
    + nps.[row_overflow_used_page_count]
    + nps.[lob_used_page_count]                                AS  [data_space_page_count]
, nps.[reserved_page_count]
 - (nps.[reserved_page_count] - nps.[used_page_count])
 - ([in_row_data_page_count]
        + [row_overflow_used_page_count]+[lob_used_page_count])       AS  [index_space_page_count]
, nps.[row_count]                                              AS  [row_count]
from
```

```
    sys.schemas s
INNER JOIN sys.tables t
    ON s.[schema_id] = t.[schema_id]
INNER JOIN sys.indexes i
    ON  t.[object_id] = i.[object_id]
    AND i.[index_id] <= 1
INNER JOIN sys.pdw_table_distribution_properties tp
    ON t.[object_id] = tp.[object_id]
INNER JOIN sys.pdw_table_mappings tm
    ON t.[object_id] = tm.[object_id]
INNER JOIN sys.pdw_nodes_tables nt
    ON tm.[physical_name] = nt.[name]
INNER JOIN sys.dm_pdw_nodes pn
    ON  nt.[pdw_node_id] = pn.[pdw_node_id]
INNER JOIN sys.pdw_distributions di
    ON  nt.[distribution_id] = di.[distribution_id]
INNER JOIN sys.dm_pdw_nodes_db_partition_stats nps
    ON nt.[object_id] = nps.[object_id]
    AND nt.[pdw_node_id] = nps.[pdw_node_id]
    AND nt.[distribution_id] = nps.[distribution_id]
LEFT OUTER JOIN (select * from sys.pdw_column_distribution_properties where distribution_ordinal = 1) cdp
    ON t.[object_id] = cdp.[object_id]
LEFT OUTER JOIN sys.columns c
    ON cdp.[object_id] = c.[object_id]
    AND cdp.[column_id] = c.[column_id]
WHERE pn.[type] = 'COMPUTE'
)
, size
AS
(
SELECT
   [execution_time]
,  [database_name]
,  [schema_name]
,  [table_name]
,  [two_part_name]
,  [node_table_name]
,  [node_table_name_seq]
,  [distribution_policy_name]
,  [distribution_column]
```

```
,  [distribution_id]
,  [index_type]
,  [index_type_desc]
,  [pdw_node_id]
,  [pdw_node_type]
,  [pdw_node_name]
,  [dist_name]
,  [dist_position]
,  [partition_nmbr]
,  [reserved_space_page_count]
,  [unused_space_page_count]
,  [data_space_page_count]
,  [index_space_page_count]
,  [row_count]
,  ([reserved_space_page_count] * 8.0)                    AS [reserved_space_KB]
,  ([reserved_space_page_count] * 8.0)/1000               AS [reserved_space_MB]
,  ([reserved_space_page_count] * 8.0)/1000000            AS [reserved_space_GB]
,  ([reserved_space_page_count] * 8.0)/1000000000         AS [reserved_space_TB]
,  ([unused_space_page_count]   * 8.0)                    AS [unused_space_KB]
,  ([unused_space_page_count]   * 8.0)/1000               AS [unused_space_MB]
,  ([unused_space_page_count]   * 8.0)/1000000            AS [unused_space_GB]
,  ([unused_space_page_count]   * 8.0)/1000000000         AS [unused_space_TB]
,  ([data_space_page_count]     * 8.0)                    AS [data_space_KB]
,  ([data_space_page_count]     * 8.0)/1000               AS [data_space_MB]
,  ([data_space_page_count]     * 8.0)/1000000            AS [data_space_GB]
,  ([data_space_page_count]     * 8.0)/1000000000         AS [data_space_TB]
,  ([index_space_page_count]  * 8.0)                      AS [index_space_KB]
,  ([index_space_page_count]  * 8.0)/1000                 AS [index_space_MB]
,  ([index_space_page_count]  * 8.0)/1000000              AS [index_space_GB]
,  ([index_space_page_count]  * 8.0)/1000000000           AS [index_space_TB]
FROM base
)
SELECT *
FROM size
;



--  `Table space summary`
--  * This query returns the rows and space by table.
```

```
--  * It allows you to see which tables are your largest tables and whether they're `round-robin`, `replicated`, or `hash -distributed`.
--  * For hash-distributed tables, the query shows the distribution column


SELECT
     database_name
,    schema_name
,    table_name
,    distribution_policy_name
,      distribution_column
,    index_type_desc
,    COUNT(distinct partition_nmbr) as nbr_partitions
,    SUM(row_count)                 as table_row_count
,    SUM(reserved_space_GB)         as table_reserved_space_GB
,    SUM(data_space_GB)             as table_data_space_GB
,    SUM(index_space_GB)            as table_index_space_GB
,    SUM(unused_space_GB)           as table_unused_space_GB
FROM
    dbo.vTableSizes
GROUP BY
     database_name
,    schema_name
,    table_name
,    distribution_policy_name
,      distribution_column
,    index_type_desc
ORDER BY
    table_reserved_space_GB desc
;
```

9) Table space by distribution type

```
SELECT
    distribution_policy_name
,   SUM(row_count)              as table_type_row_count
,   SUM(reserved_space_GB)      as table_type_reserved_space_GB
,   SUM(data_space_GB)          as table_type_data_space_GB
,   SUM(index_space_GB)         as table_type_index_space_GB
,   SUM(unused_space_GB)        as table_type_unused_space_GB
FROM dbo.vTableSizes
GROUP BY distribution_policy_name
;
```

10) Table space by index type

```
SELECT
    index_type_desc
,   SUM(row_count)              as table_type_row_count
,   SUM(reserved_space_GB)      as table_type_reserved_space_GB
,   SUM(data_space_GB)          as table_type_data_space_GB
,   SUM(index_space_GB)         as table_type_index_space_GB
,   SUM(unused_space_GB)        as table_type_unused_space_GB
FROM dbo.vTableSizes
GROUP BY index_type_desc
;
```

11) Distribution space summary

```
SELECT
    distribution_id
,    SUM(row_count)              as total_node_distribution_row_count
,    SUM(reserved_space_MB)      as total_node_distribution_reserved_space_MB
,    SUM(data_space_MB)          as total_node_distribution_data_space_MB
,    SUM(index_space_MB)         as total_node_distribution_index_space_MB
,    SUM(unused_space_MB)        as total_node_distribution_unused_space_MB
FROM dbo.vTableSizes
GROUP BY    distribution_id
ORDER BY    distribution_id
;
```

12) Creating ref. temp table (CTAS) to hold mapping info.

```
CREATE TABLE #ref
WITH (DISTRIBUTION = ROUND_ROBIN)
AS
WITH
-- Creating concurrency slots mapping for various DWUs.
alloc
AS
(
SELECT 'DW100c' AS DWU,4 AS max_queries,4 AS max_slots,1 AS slots_used_smallrc,1 AS slots_used_mediumrc,2 AS slots_used_largerc,4 AS
slots_used_xlargerc,1 AS slots_used_staticrc10,2 AS slots_used_staticrc20,4 AS slots_used_staticrc30,4 AS slots_used_staticrc40,4 AS
slots_used_staticrc50,4 AS slots_used_staticrc60,4 AS slots_used_staticrc70,4 AS slots_used_staticrc80
  UNION ALL
   SELECT 'DW200c',8,8,1,2,4,8,1,2,4,8,8,8,8,8
  UNION ALL
   SELECT 'DW300c',12,12,1,2,4,8,1,2,4,8,8,8,8,8
  UNION ALL
   SELECT 'DW400c',16,16,1,4,8,16,1,2,4,8,16,16,16,16
  UNION ALL
   SELECT 'DW500c',20,20,1,4,8,16,1,2,4,8,16,16,16,16
  UNION ALL
   SELECT 'DW1000c',32,40,1,4,8,28,1,2,4,8,16,32,32,32
```

```
  UNION ALL
   SELECT 'DW1500c',32,60,1,6,13,42,1,2,4,8,16,32,32,32
  UNION ALL
   SELECT 'DW2000c',48,80,2,8,17,56,1,2,4,8,16,32,64,64
  UNION ALL
   SELECT 'DW2500c',48,100,3,10,22,70,1,2,4,8,16,32,64,64
  UNION ALL
   SELECT 'DW3000c',64,120,3,12,26,84,1,2,4,8,16,32,64,64
  UNION ALL
   SELECT 'DW5000c',64,200,6,20,44,140,1,2,4,8,16,32,64,128
  UNION ALL
   SELECT 'DW6000c',128,240,7,24,52,168,1,2,4,8,16,32,64,128
  UNION ALL
   SELECT 'DW7500c',128,300,9,30,66,210,1,2,4,8,16,32,64,128
  UNION ALL
   SELECT 'DW10000c',128,400,12,40,88,280,1,2,4,8,16,32,64,128
  UNION ALL
   SELECT 'DW15000c',128,600,18,60,132,420,1,2,4,8,16,32,64,128
  UNION ALL
   SELECT 'DW30000c',128,1200,36,120,264,840,1,2,4,8,16,32,64,128
)
-- Creating workload mapping to their corresponding slot consumption and default memory grant.
,map
AS
(
  SELECT CONVERT(varchar(20), 'SloDWGroupSmall') AS wg_name, slots_used_smallrc AS slots_used FROM alloc WHERE DWU = @DWU
UNION ALL
  SELECT CONVERT(varchar(20), 'SloDWGroupMedium') AS wg_name, slots_used_mediumrc AS slots_used FROM alloc WHERE DWU = @DWU
UNION ALL
  SELECT CONVERT(varchar(20), 'SloDWGroupLarge') AS wg_name, slots_used_largerc AS slots_used FROM alloc WHERE DWU = @DWU
UNION ALL
  SELECT CONVERT(varchar(20), 'SloDWGroupXLarge') AS wg_name, slots_used_xlargerc AS slots_used FROM alloc WHERE DWU = @DWU
  UNION ALL
  SELECT 'SloDWGroupC00',1
  UNION ALL
    SELECT 'SloDWGroupC01',2
  UNION ALL
    SELECT 'SloDWGroupC02',4
  UNION ALL
    SELECT 'SloDWGroupC03',8
```

```
  UNION ALL
    SELECT 'SloDWGroupC04',16
  UNION ALL
    SELECT 'SloDWGroupC05',32
  UNION ALL
    SELECT 'SloDWGroupC06',64
  UNION ALL
    SELECT 'SloDWGroupC07',128
)

-- Creating ref based on current / asked DWU.
, ref
AS
(
  SELECT  a1.*
  ,       m1.wg_name          AS wg_name_smallrc
  ,       m1.slots_used * 250 AS tgt_mem_grant_MB_smallrc
  ,       m2.wg_name          AS wg_name_mediumrc
  ,       m2.slots_used * 250 AS tgt_mem_grant_MB_mediumrc
  ,       m3.wg_name          AS wg_name_largerc
  ,       m3.slots_used * 250 AS tgt_mem_grant_MB_largerc
  ,       m4.wg_name          AS wg_name_xlargerc
  ,       m4.slots_used * 250 AS tgt_mem_grant_MB_xlargerc
  ,       m5.wg_name          AS wg_name_staticrc10
  ,       m5.slots_used * 250 AS tgt_mem_grant_MB_staticrc10
  ,       m6.wg_name          AS wg_name_staticrc20
  ,       m6.slots_used * 250 AS tgt_mem_grant_MB_staticrc20
  ,       m7.wg_name          AS wg_name_staticrc30
  ,       m7.slots_used * 250 AS tgt_mem_grant_MB_staticrc30
  ,       m8.wg_name          AS wg_name_staticrc40
  ,       m8.slots_used * 250 AS tgt_mem_grant_MB_staticrc40
  ,       m9.wg_name          AS wg_name_staticrc50
  ,       m9.slots_used * 250 AS tgt_mem_grant_MB_staticrc50
  ,       m10.wg_name          AS wg_name_staticrc60
  ,       m10.slots_used * 250 AS tgt_mem_grant_MB_staticrc60
  ,       m11.wg_name          AS wg_name_staticrc70
  ,       m11.slots_used * 250 AS tgt_mem_grant_MB_staticrc70
  ,       m12.wg_name          AS wg_name_staticrc80
  ,       m12.slots_used * 250 AS tgt_mem_grant_MB_staticrc80
  FROM alloc a1
```

```
  JOIN map   m1  ON a1.slots_used_smallrc     = m1.slots_used and m1.wg_name = 'SloDWGroupSmall'
  JOIN map   m2  ON a1.slots_used_mediumrc    = m2.slots_used and m2.wg_name = 'SloDWGroupMedium'
  JOIN map   m3  ON a1.slots_used_largerc     = m3.slots_used and m3.wg_name = 'SloDWGroupLarge'
  JOIN map   m4  ON a1.slots_used_xlargerc    = m4.slots_used and m4.wg_name = 'SloDWGroupXLarge'
  JOIN map   m5  ON a1.slots_used_staticrc10    = m5.slots_used and m5.wg_name NOT IN
('SloDWGroupSmall','SloDWGroupMedium','SloDWGroupLarge','SloDWGroupXLarge')
  JOIN map   m6  ON a1.slots_used_staticrc20    = m6.slots_used and m6.wg_name NOT IN
('SloDWGroupSmall','SloDWGroupMedium','SloDWGroupLarge','SloDWGroupXLarge')
  JOIN map   m7  ON a1.slots_used_staticrc30    = m7.slots_used and m7.wg_name NOT IN
('SloDWGroupSmall','SloDWGroupMedium','SloDWGroupLarge','SloDWGroupXLarge')
  JOIN map   m8  ON a1.slots_used_staticrc40    = m8.slots_used and m8.wg_name NOT IN
('SloDWGroupSmall','SloDWGroupMedium','SloDWGroupLarge','SloDWGroupXLarge')
  JOIN map   m9  ON a1.slots_used_staticrc50    = m9.slots_used and m9.wg_name NOT IN
('SloDWGroupSmall','SloDWGroupMedium','SloDWGroupLarge','SloDWGroupXLarge')
  JOIN map   m10  ON a1.slots_used_staticrc60    = m10.slots_used and m10.wg_name NOT IN
('SloDWGroupSmall','SloDWGroupMedium','SloDWGroupLarge','SloDWGroupXLarge')
  JOIN map   m11  ON a1.slots_used_staticrc70    = m11.slots_used and m11.wg_name NOT IN
('SloDWGroupSmall','SloDWGroupMedium','SloDWGroupLarge','SloDWGroupXLarge')
  JOIN map   m12  ON a1.slots_used_staticrc80    = m12.slots_used and m12.wg_name NOT IN
('SloDWGroupSmall','SloDWGroupMedium','SloDWGroupLarge','SloDWGroupXLarge')
  WHERE   a1.DWU = @DWU
)
SELECT  DWU
,       max_queries
,       max_slots
,       slots_used
,       wg_name
,       tgt_mem_grant_MB
,       up1 as rc
,       (ROW_NUMBER() OVER(PARTITION BY DWU ORDER BY DWU)) as rc_id
FROM
(
    SELECT  DWU
    ,       max_queries
    ,       max_slots
    ,       slots_used
    ,       wg_name
    ,       tgt_mem_grant_MB
    ,       REVERSE(SUBSTRING(REVERSE(wg_names),1,CHARINDEX('_',REVERSE(wg_names),1)-1)) as up1
    ,       REVERSE(SUBSTRING(REVERSE(tgt_mem_grant_MBs),1,CHARINDEX('_',REVERSE(tgt_mem_grant_MBs),1)-1)) as up2
```

```
    ,       REVERSE(SUBSTRING(REVERSE(slots_used_all),1,CHARINDEX('_',REVERSE(slots_used_all),1)-1)) as up3
    FROM    ref AS r1
    UNPIVOT
    (
        wg_name FOR wg_names IN (wg_name_smallrc,wg_name_mediumrc,wg_name_largerc,wg_name_xlargerc,
        wg_name_staticrc10, wg_name_staticrc20, wg_name_staticrc30, wg_name_staticrc40, wg_name_staticrc50,
        wg_name_staticrc60, wg_name_staticrc70, wg_name_staticrc80)
    ) AS r2
    UNPIVOT
    (
        tgt_mem_grant_MB FOR tgt_mem_grant_MBs IN (tgt_mem_grant_MB_smallrc,tgt_mem_grant_MB_mediumrc,
        tgt_mem_grant_MB_largerc,tgt_mem_grant_MB_xlargerc, tgt_mem_grant_MB_staticrc10, tgt_mem_grant_MB_staticrc20,
        tgt_mem_grant_MB_staticrc30, tgt_mem_grant_MB_staticrc40, tgt_mem_grant_MB_staticrc50,
        tgt_mem_grant_MB_staticrc60, tgt_mem_grant_MB_staticrc70, tgt_mem_grant_MB_staticrc80)
    ) AS r3
    UNPIVOT
    (
        slots_used FOR slots_used_all IN (slots_used_smallrc,slots_used_mediumrc,slots_used_largerc,
        slots_used_xlargerc, slots_used_staticrc10, slots_used_staticrc20, slots_used_staticrc30,
        slots_used_staticrc40, slots_used_staticrc50, slots_used_staticrc60, slots_used_staticrc70,
        slots_used_staticrc80)
    ) AS r4
) a
WHERE   up1 = up2
AND     up1 = up3
;

-- Getting current info about workload groups.
WITH
dmv
AS
(
  SELECT
        rp.name                                 AS rp_name
  ,     rp.max_memory_kb*1.0/1048576                    AS rp_max_mem_GB
  ,     (rp.max_memory_kb*1.0/1024)
        *(request_max_memory_grant_percent/100)         AS max_memory_grant_MB
  ,     (rp.max_memory_kb*1.0/1048576)
        *(request_max_memory_grant_percent/100)         AS max_memory_grant_GB
  ,     wg.name                                 AS wg_name
```

```
  ,       wg.importance                           AS importance
  ,       wg.request_max_memory_grant_percent             AS request_max_memory_grant_percent
  FROM    sys.dm_pdw_nodes_resource_governor_workload_groups wg
  JOIN    sys.dm_pdw_nodes_resource_governor_resource_pools rp    ON  wg.pdw_node_id  = rp.pdw_node_id
                                          AND wg.pool_id      = rp.pool_id
  WHERE   rp.name = 'SloDWPool'
  GROUP BY
          rp.name
  ,       rp.max_memory_kb
  ,       wg.name
  ,       wg.importance
  ,       wg.request_max_memory_grant_percent
)
-- Creating resource class name mapping.
,names
AS
(
  SELECT 'smallrc' as resource_class, 1 as rc_id
  UNION ALL
    SELECT 'mediumrc', 2
  UNION ALL
    SELECT 'largerc', 3
  UNION ALL
    SELECT 'xlargerc', 4
  UNION ALL
    SELECT 'staticrc10', 5
  UNION ALL
    SELECT 'staticrc20', 6
  UNION ALL
    SELECT 'staticrc30', 7
  UNION ALL
    SELECT 'staticrc40', 8
  UNION ALL
    SELECT 'staticrc50', 9
  UNION ALL
    SELECT 'staticrc60', 10
  UNION ALL
    SELECT 'staticrc70', 11
  UNION ALL
    SELECT 'staticrc80', 12
```

```
)
,base AS
(   SELECT  schema_name
    ,       table_name
    ,       SUM(column_count)                   AS column_count
    ,       ISNULL(SUM(short_string_column_count),0)   AS short_string_column_count
    ,       ISNULL(SUM(long_string_column_count),0)    AS long_string_column_count
    FROM    (   SELECT  sm.name                                    AS schema_name
                ,       tb.name                                    AS table_name
                ,       COUNT(co.column_id)                        AS column_count
                        ,       CASE    WHEN co.system_type_id IN (36,43,106,108,165,167,173,175,231,239)
                                AND  co.max_length <= 32
                                THEN COUNT(co.column_id)
                                END                                AS short_string_column_count
                ,       CASE    WHEN co.system_type_id IN (165,167,173,175,231,239)
                                AND  co.max_length > 32 and co.max_length <=8000
                                THEN COUNT(co.column_id)
                                END                                AS long_string_column_count
                FROM    sys.schemas AS sm
                JOIN    sys.tables  AS tb   on sm.[schema_id] = tb.[schema_id]
                JOIN    sys.columns AS co   ON tb.[object_id] = co.[object_id]
                        WHERE tb.name = @TABLE_NAME AND sm.name = @SCHEMA_NAME
                GROUP BY sm.name
                ,       tb.name
                ,       co.system_type_id
                ,       co.max_length           ) a
GROUP BY schema_name
,       table_name
)
, size AS
(
SELECT  schema_name
,       table_name
,       75497472                                AS table_overhead

,       column_count*1048576*8                  AS column_size
,       short_string_column_count*1048576*32            AS short_string_size,       (long_string_column_count*16777216) AS long_string_size
FROM    base
UNION
SELECT CASE WHEN COUNT(*) = 0 THEN 'EMPTY' END as schema_name
```

```
        ,CASE WHEN COUNT(*) = 0 THEN 'EMPTY' END as table_name
        ,CASE WHEN COUNT(*) = 0 THEN 0 END as table_overhead
        ,CASE WHEN COUNT(*) = 0 THEN 0 END as column_size
        ,CASE WHEN COUNT(*) = 0 THEN 0 END as short_string_size

,CASE WHEN COUNT(*) = 0 THEN 0 END as long_string_size
FROM   base
)
, load_multiplier as
(
SELECT  CASE
         WHEN FLOOR(8 * (CAST (CAST(REPLACE(REPLACE(@DWU,'DW',''),'c','') AS INT) AS FLOAT)/6000)) > 0
          AND CHARINDEX(@DWU,'c')=0
         THEN FLOOR(8 * (CAST (CAST(REPLACE(REPLACE(@DWU,'DW',''),'c','') AS INT) AS FLOAT)/6000))
         ELSE 1
       END AS multiplication_factor
)
      SELECT  r1.DWU
      , schema_name
      , table_name
      , rc.resource_class as closest_rc_in_increasing_order
      , max_queries_at_this_rc = CASE
          WHEN (r1.max_slots / r1.slots_used > r1.max_queries)
              THEN r1.max_queries
          ELSE r1.max_slots / r1.slots_used
              END
      , r1.max_slots as max_concurrency_slots
      , r1.slots_used as required_slots_for_the_rc
      , r1.tgt_mem_grant_MB  as rc_mem_grant_MB
      , CAST((table_overhead*1.0+column_size+short_string_size+long_string_size)*multiplication_factor/1048576    AS DECIMAL(18,2)) AS
est_mem_grant_required_for_cci_operation_MB
      FROM    size
      , load_multiplier
      , #ref r1, names  rc
      WHERE r1.rc_id=rc.rc_id
                AND CAST((table_overhead*1.0+column_size+short_string_size+long_string_size)*multiplication_factor/1048576    AS DECIMAL(18,2))
< r1.tgt_mem_grant_MB
      ORDER BY ABS(CAST((table_overhead*1.0+column_size+short_string_size+long_string_size)*multiplication_factor/1048576    AS
DECIMAL(18,2)) - r1.tgt_mem_grant_MB)
```