# Full-Indexing Management Scripts

-- Get information about FullText catalogs

SELECT fulltext\_catalog\_id, [name], is\_default, is\_accent\_sensitivity\_on, principal\_id, is\_importing

FROM sys.fulltext\_catalogs;

-- This is deprecated

EXEC sp\_help\_fulltext\_catalogs 'ftCatalog';

-- Set Change Tracking to Manual

ALTER FULLTEXT INDEX ON Person.Contact SET CHANGE\_TRACKING MANUAL;

-- Start a Manual Update of the FullText Catalog

ALTER FULLTEXT INDEX ON Person.Contact START UPDATE POPULATION;

-- Resume population in case of an error during manual population

ALTER FULLTEXT INDEX ON Person.Contact RESUME POPULATION;

-- Find out how many changes are pending

SELECT OBJECTPROPERTY(OBJECT\_ID('Person.Contact'), 'TableFulltextPendingChanges') AS 'Full Text Pending Changes';

-- Check Full Text Fragments (lower number of rows is better, closed fragments are bad)

-- Status Codes

-- 0 = Newly created and not yet used

-- 1 = Being used for insert

-- 4 = Closed. Ready for query

-- 6 = Being used for merge input and ready for query

-- 8 = Marked for deletion. Will not be used for query and merge source.

SELECT table\_id, fragment\_id, fragment\_object\_id, [timestamp], [status], data\_size, row\_count

FROM sys.fulltext\_index\_fragments WITH (NOLOCK);

-- Start a Manual Merge (when fragment count is high)

ALTER FULLTEXT CATALOG ftCatalog REORGANIZE;

-- Completely rebuild the FT Index (this may take quite some time)

ALTER FULLTEXT CATALOG ftCatalog

REBUILD WITH ACCENT\_SENSITIVITY=OFF;

-- Check Master Merge Status (1 = in Progress)

SELECT FULLTEXTCATALOGPROPERTY('ftCatalog', 'MergeStatus') AS 'Master Merge Status';

-- Check Populate Status (1 = in Progress)

-- 0 = Idle

-- 1 = Full population in progress

-- 2 = Paused

-- 3 = Throttled

-- 4 = Recovering

-- 5 = Shutdown

-- 6 = Incremental population in progress

-- 7 = Building index

-- 8 = Disk is full. Paused.

-- 9 = Change tracking

SELECT FULLTEXTCATALOGPROPERTY('ftCatalog', 'PopulateStatus') AS 'Populate Status';

-- Check Accent sensitivity of the FT Catalog

SELECT FULLTEXTCATALOGPROPERTY('ftCatalog', 'AccentSensitivity') AS 'Accent Sensitivity';

-- Number of full-text indexed items currently in the full-text catalog

SELECT FULLTEXTCATALOGPROPERTY('ftCatalog', 'ItemCount')AS 'Item Count';

-- Size of the full-text catalog in megabytes

SELECT FULLTEXTCATALOGPROPERTY('ftCatalog', 'IndexSize')AS 'Size in MB';

-- Active FTS Catalogs

SELECT database\_id,catalog\_id,memory\_address,name,is\_paused,[status],status\_description,

previous\_status,previous\_status\_description,worker\_count,active\_fts\_index\_count,auto\_population\_count,

manual\_population\_count,full\_incremental\_population\_count,row\_count\_in\_thousands,is\_importing

FROM sys.dm\_fts\_active\_catalogs;

-- Outstanding FTS batches

SELECT database\_id,catalog\_id,table\_id,batch\_id,memory\_address,crawl\_memory\_address,memregion\_memory\_address,

hr\_batch,is\_retry\_batch,retry\_hints,retry\_hints\_description,doc\_failed,batch\_timestamp

FROM sys.dm\_fts\_outstanding\_batches;

-- FTS Index Population

SELECT database\_id,catalog\_id,table\_id,memory\_address,population\_type,population\_type\_description,

is\_clustered\_index\_scan,range\_count,completed\_range\_count,outstanding\_batch\_count,[status],

status\_description,completion\_type,completion\_type\_description,worker\_count,

queued\_population\_type, queued\_population\_type\_description,start\_time,incremental\_timestamp

FROM sys.dm\_fts\_index\_population;

-- Check transaction log usage

SELECT instance\_name, cntr\_value AS 'Log Percent Used'

FROM sys.dm\_os\_performance\_counters WITH (NOLOCK)

WHERE counter\_name = 'Percent Log Used'

AND instance\_name = 'AdventureWorks'

-- Clear Wait Stats

DBCC SQLPERF('sys.dm\_os\_wait\_stats', CLEAR);

-- Get Top Waits

WITH Waits AS

(

SELECT

wait\_type,

wait\_time\_ms / 1000. AS wait\_time\_s,

100. \* wait\_time\_ms / SUM(wait\_time\_ms) OVER() AS pct,

ROW\_NUMBER() OVER(ORDER BY wait\_time\_ms DESC) AS rn

FROM sys.dm\_os\_wait\_stats

WHERE wait\_type NOT LIKE '%SLEEP%'

-- filter out additional irrelevant waits

)

SELECT

W1.wait\_type,

CAST(W1.wait\_time\_s AS DECIMAL(12, 2)) AS wait\_time\_s,

CAST(W1.pct AS DECIMAL(12, 2)) AS pct,

CAST(SUM(W2.pct) AS DECIMAL(12, 2)) AS running\_pct

FROM Waits AS W1

INNER JOIN Waits AS W2

ON W2.rn <= W1.rn

GROUP BY W1.rn, W1.wait\_type, W1.wait\_time\_s, W1.pct

HAVING SUM(W2.pct) - W1.pct < 90 -- percentage threshold

ORDER BY W1.rn;

-- Detect blocking

SELECT blocked\_query.session\_id AS blocked\_session\_id,

blocking\_query.session\_id AS blocking\_session\_id,

sql\_text.text AS blocked\_text, sql\_btext.text AS blocking\_text, waits.wait\_type AS blocking\_resource

FROM sys.dm\_exec\_requests AS blocked\_query

INNER JOIN sys.dm\_exec\_requests AS blocking\_query

ON blocked\_query.blocking\_session\_id = blocking\_query.session\_id

CROSS APPLY

(SELECT \* FROM sys.dm\_exec\_sql\_text(blocking\_query.sql\_handle)

) sql\_btext

CROSS APPLY

(SELECT \* FROM sys.dm\_exec\_sql\_text(blocked\_query.sql\_handle)

) sql\_text

INNER JOIN sys.dm\_os\_waiting\_tasks AS waits

ON waits.session\_id = blocking\_query.session\_id

-- Index Contention

SELECT dbid=database\_id, objectname=object\_name(s.object\_id)

, indexname=i.name, i.index\_id

, row\_lock\_count, row\_lock\_wait\_count

, [block %]=CAST (100.0 \* row\_lock\_wait\_count / (1 + row\_lock\_count) AS NUMERIC(15,2))

, row\_lock\_wait\_in\_ms

, [avg row lock waits in ms]=CAST (1.0 \* row\_lock\_wait\_in\_ms / (1 + row\_lock\_wait\_count) AS NUMERIC(15,2))

FROM sys.dm\_db\_index\_operational\_stats (db\_id(), NULL, NULL, NULL) AS s

INNER JOIN sys.indexes AS i

ON i.object\_id = s.object\_id

WHERE objectproperty(s.object\_id,'IsUserTable') = 1

AND i.index\_id = s.index\_id

ORDER BY row\_lock\_wait\_count DESC

-- Get clustered index fragmentation

SELECT avg\_fragmentation\_in\_percent

FROM sys.dm\_db\_index\_physical\_stats(DB\_ID(N'AdventureWorks'), OBJECT\_ID(N'AdventureWorks.Person.Contact'), 1, NULL , 'LIMITED');

-- Get non-clustered index fragmentation

SELECT avg\_fragmentation\_in\_percent

FROM sys.dm\_db\_index\_physical\_stats(DB\_ID(N'AdventureWorks'), OBJECT\_ID(N'AdventureWorks.Person.Contact'), 2, NULL , 'LIMITED');