## USE master

# GO

# CREATE DATABASE PerformanceDB

# GO

# USE PerformanceDB

# GO

# CREATE TABLE Logs(

# LogId int NOT NULL PRIMARY KEY IDENTITY,

# InputDate DATETIME,

# LogText NVARCHAR(300),

# )

# DECLARE @RowCount int = 10000000

# WHILE @RowCount > 0

# BEGIN

# DECLARE @LogText nvarchar(100) = 'Text ' + CONVERT(nvarchar(100), @RowCount) +

# ': ' + CONVERT(nvarchar(100), newid())

# DECLARE @InputDate datetime = DATEADD(month, CONVERT(varbinary, newid()) % (50 \* 12),

# getdate())

# INSERT INTO Logs(InputDate, LogText)

# VALUES(@InputDate, @LogText)

# SET @RowCount = @RowCount - 1

# END

# CHECKPOINT; DBCC DROPCLEANBUFFERS; -- Empty the SQL Server cache

# SELECT InputDate

# FROM Logs AS l

# WHERE l.InputDate > '31-Dec-2005' and l.InputDate < '1-Jan-2010'

# 102 044 rows in 6 sec without cash

# 102 044 rows in 1 sec with cash

## CREATE INDEX IDX\_Logs\_InputDate ON Logs(InputDate)

# SELECT InputDate

# FROM Logs AS l

# WHERE l.InputDate > '31-Dec-2005' and l.InputDate < '1-Jan-2010'

# 102 044 rows in less than 1 sec without cash

# DROP INDEX IDX\_Logs\_InputDate ON Logs

## CREATE FULLTEXT CATALOG LogsFullTextCatalog

# WITH ACCENT\_SENSITIVITY = OFF

# CREATE FULLTEXT INDEX ON Logs(LogText)

# KEY INDEX PK\_\_Logs\_\_5E5486487F60ED59

# ON LogsFullTextCatalog

# WITH CHANGE\_TRACKING AUTO

# CHECKPOINT; DBCC DROPCLEANBUFFERS; -- Empty the SQL Server cache

# SELECT COUNT(\*)

# FROM Logs As l

# WHERE CONTAINS(L.LogText, '999')

# 32 920 rows in 2 sec without cash

# CHECKPOINT; DBCC DROPCLEANBUFFERS; -- Empty the SQL Server cache

# -- This is still slow

# SELECT COUNT(\*)

# FROM Logs AS l

# WHERE l.LogText LIKE '%999%'

# 32 920 rows in 7 sec with or without cash

# DROP FULLTEXT INDEX ON Logs

# DROP FULLTEXT CATALOG LogsFullTextCatalog