1. **Create a table in SQL Server with 10 000 000 log entries (date + text). Search in the table by date range. Check the speed (without caching).**

CREATE TABLE Logs

(

LogID INT NOT NULL IDENTITY PRIMARY KEY,

LogDate DateTime NOT NULL,

LogText TEXT NOT NULL,

)

//With a few loops and changing date, table is filled. After 2001/01/01 dates are various.

DECLARE @counter INT = 8000000;

WHILE (@counter > 1000000)

BEGIN

UPDATE DateBasePerfomance.dbo.Logs

SET LogDate = '11/22/1999'

WHERE LogID = @counter;

SET @counter = @counter - 1;

END;

//Used query

SELECT l.LogDate, l.LogText

FROM [DateBasePerfomance].[dbo].Logs l

WHERE l.LogDate > '2000/01/01' AND l.LogDate < '2016/01/01'

//Metric without clear cache ~00:02:22

1. **Add an index to speed-up the search by date. Test the search speed (after cleaning the cache).**

// Clearing cache CHECKPOINT; DBCC DROPCLEANBUFFERS;

//Creating Non-Cluster Index

DateBasePerfomance -> Logs -> Indexes -> Non-Cluster Index -> Add -> DateLog

//Metric after clear cache ~00:01:23 Around 50% better search performance!

1. **Add a full text index for the text column. Try to search with and without the full-text index and compare the speed.**

Without text index

SELECT LogID, LogDate, LogText

FROM [DateBasePerfomance].[dbo].Logs

WHERE LogText LIKE 'Text is just like number: 9999999'

With text index: 00:00:00

SELECT LogID, LogText, LogDate

FROM [DateBasePerfomance].[dbo].Logs

WHERE CONTAINS(LogText,'"Text is just like number: 9999999"'