
#FABCONSQLCON2026
FABCON
Microsoft Fabric
COMMUNITY CONFERENCE
SQLCON
Microsoft SQL
COMMUNITY CONFERENCE
ATLANTA MARCH 16 - 20, 2026

SQL Server Sleuthing: Unmasking SQL Server Performance Issues

Monica Morehouse (Rathbun)
Consultant, Denny Cherry and Associates Consulting

Presentation Rules

Always Ask Questions

Interrupt me

This is a two-way conversation let’s learn from each other’s experiences

THE CRIME SCENE

DON’T GUESS INVESTIGATE!!!!

Symptoms are not root causes
Collect evidence before making changes
Performance tuning is data-driven

THE DETECTIVE MINSET

Users say: The Database is slow?

Code Release

Time Outs

CPU SPIKES

Blocking

WHEN PERFROMANCE GOES WRONG

Query Store – historical evidence
Execution Plans – how SQL thinks
Wait Stats – what SQL is waiting on
DMVs – real-time insight
Extended Events – deep diagnostics

Your Detective Toolkit

DISCOVERING THE CLUES

Glenn Berry's Diagnostic Scripts –

A treasure trove of checks across performance, memory, I/O, and more.

DISCOVERING THE CLUES

sp_whoisactive (Adam Machanic) – Like Scooby's nose, it tells you exactly what's running *right now* and where the slowdown is.

DISCOVERING THE CLUES

Paul Randal's Wait Stats Library –

Velma's encyclopedia of villains.

This library helps you decode what each wait type really means, so you know whether it's a red herring or the real culprit.

DISCOVERING THE CLUES

Query Store –

A built-in case file of past queries, letting you compare before-and-after plans.

WHICH TOOL DO I USE? .....IT DEPENDS

What problem are you trying to solve?

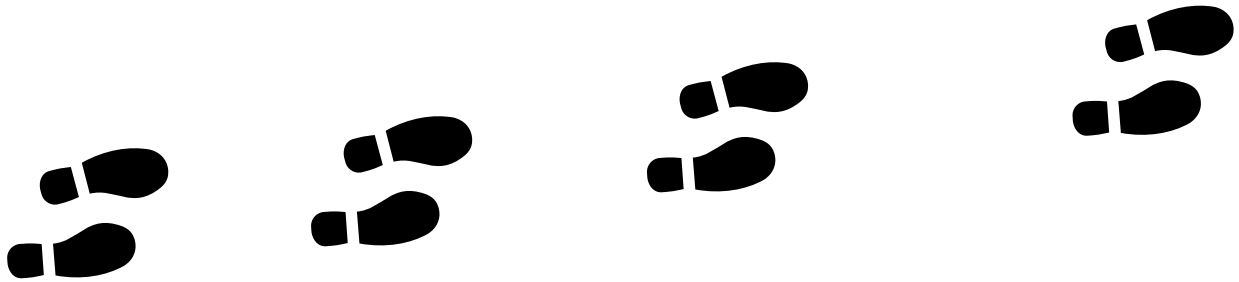
Best metrics, at the least cost

Time needed to

analyze the data

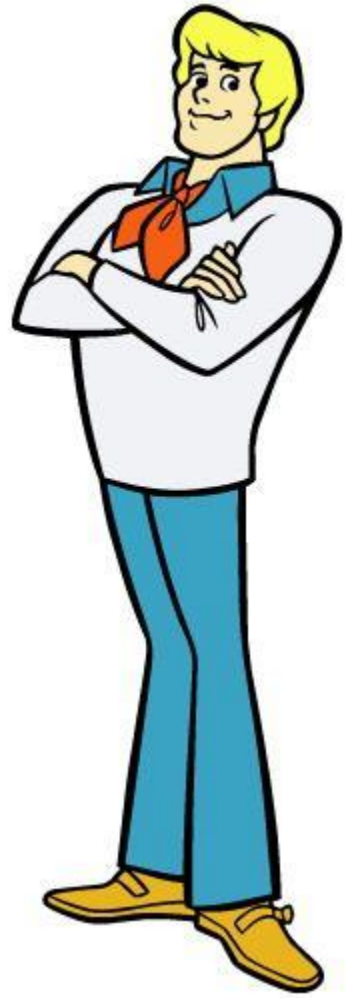
FIND CLUES-

DON’T FORGET THE ERROR LOGS!



Example:
Error 9002: *“The transaction log for database is full due to LOG_BACKUP.”*

Sounds simple, right? Just back up the log and you’re done. But here’s the trick—sometimes the real villain is missing backups at all. It could be:



COMMON PERFORMANCE CULPRITS

DON’T OVER COMPLICATE

Parameter sniffing

Missing or ineffective indexes

Implicit conversions

Outdated statistics

TempDB contention

Blocking & long transactions

Compliation Timeouts

Memory Grants Pending

TURNING CHAOS INTO CLARITY

Query Store: Your Security Camera Footage

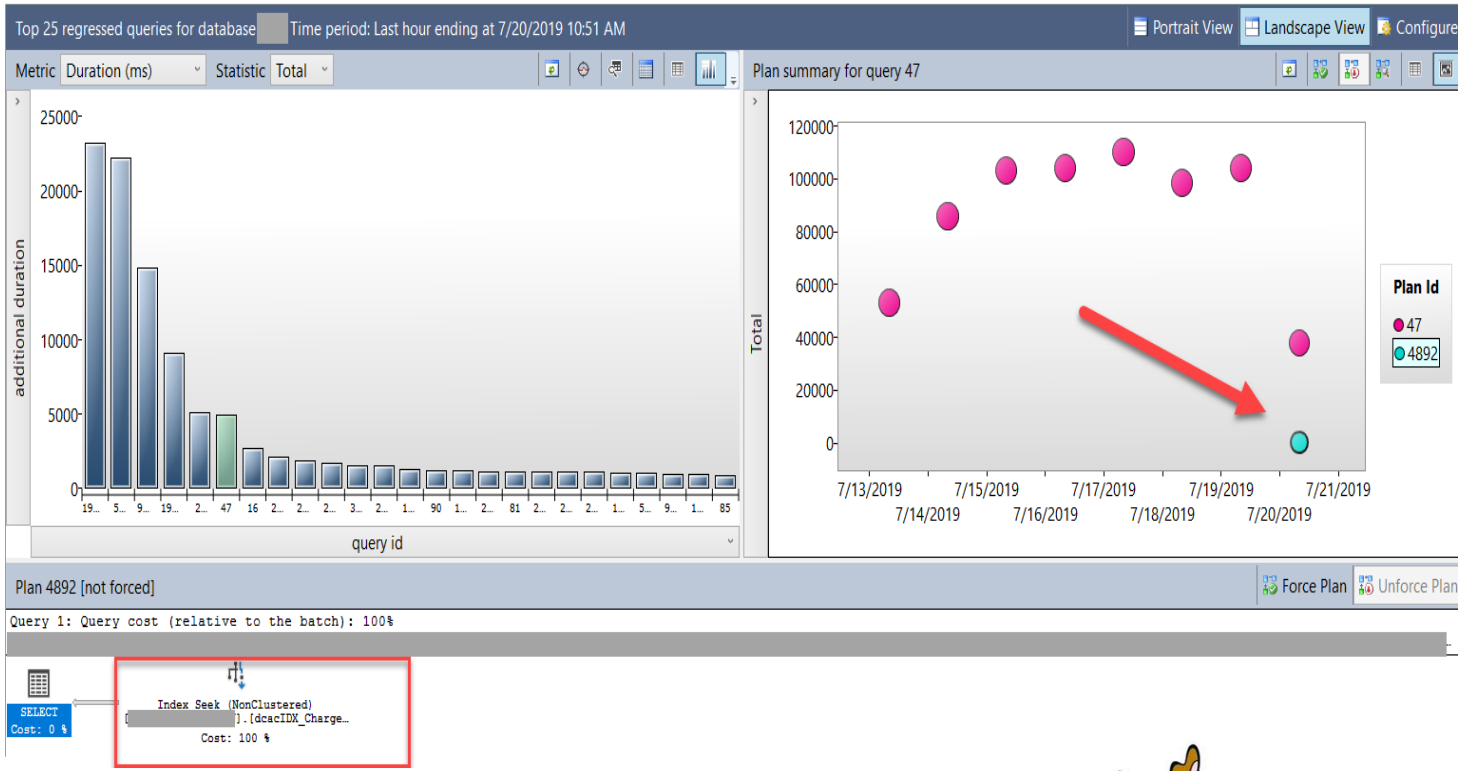
Top resource-consuming queries

Plan regressions

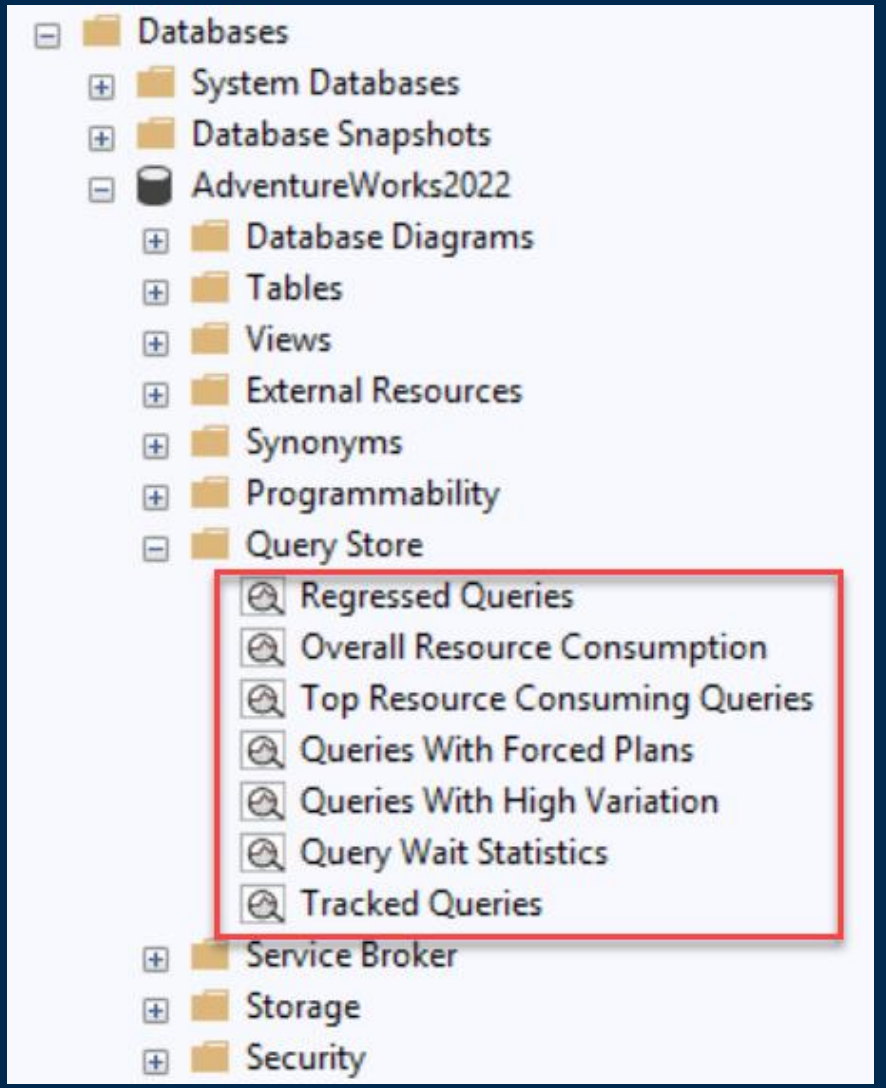
Runtime statistics over time

Compare before and after changes

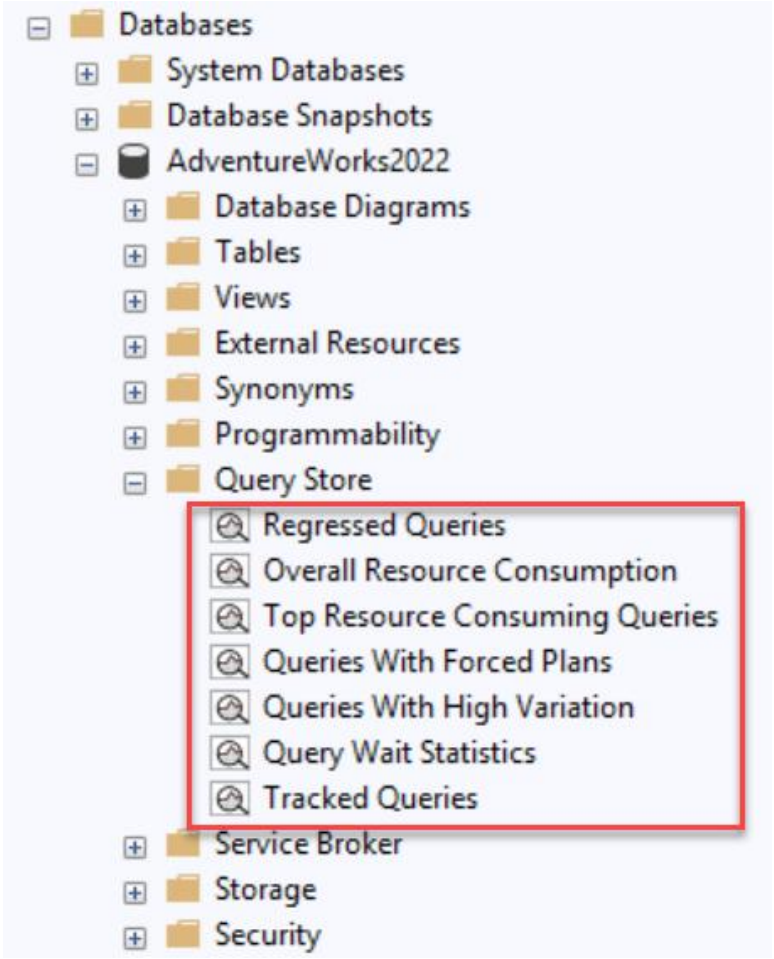
Plan forcing (when appropriate)

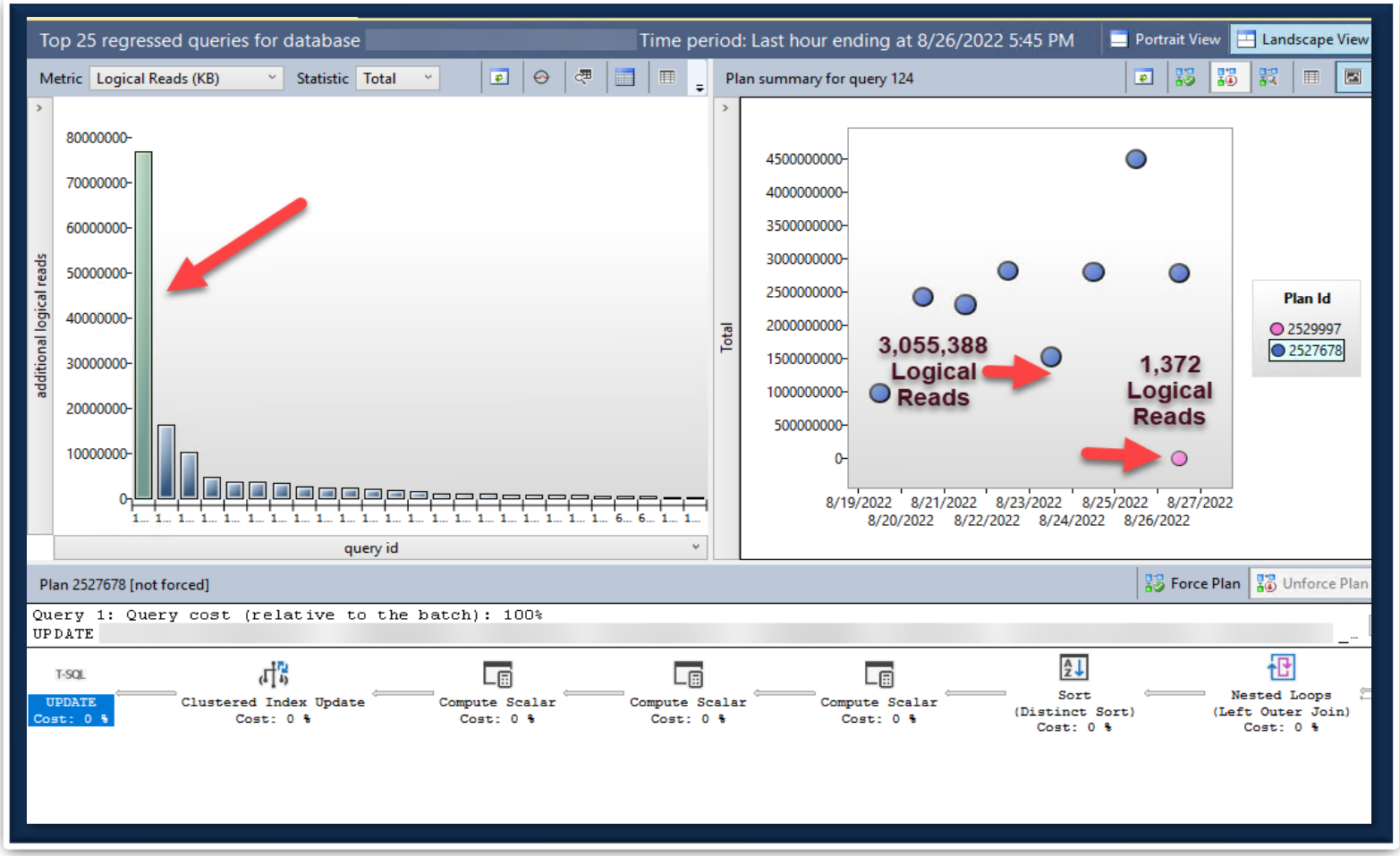


QUERY STORE FOR TUNING

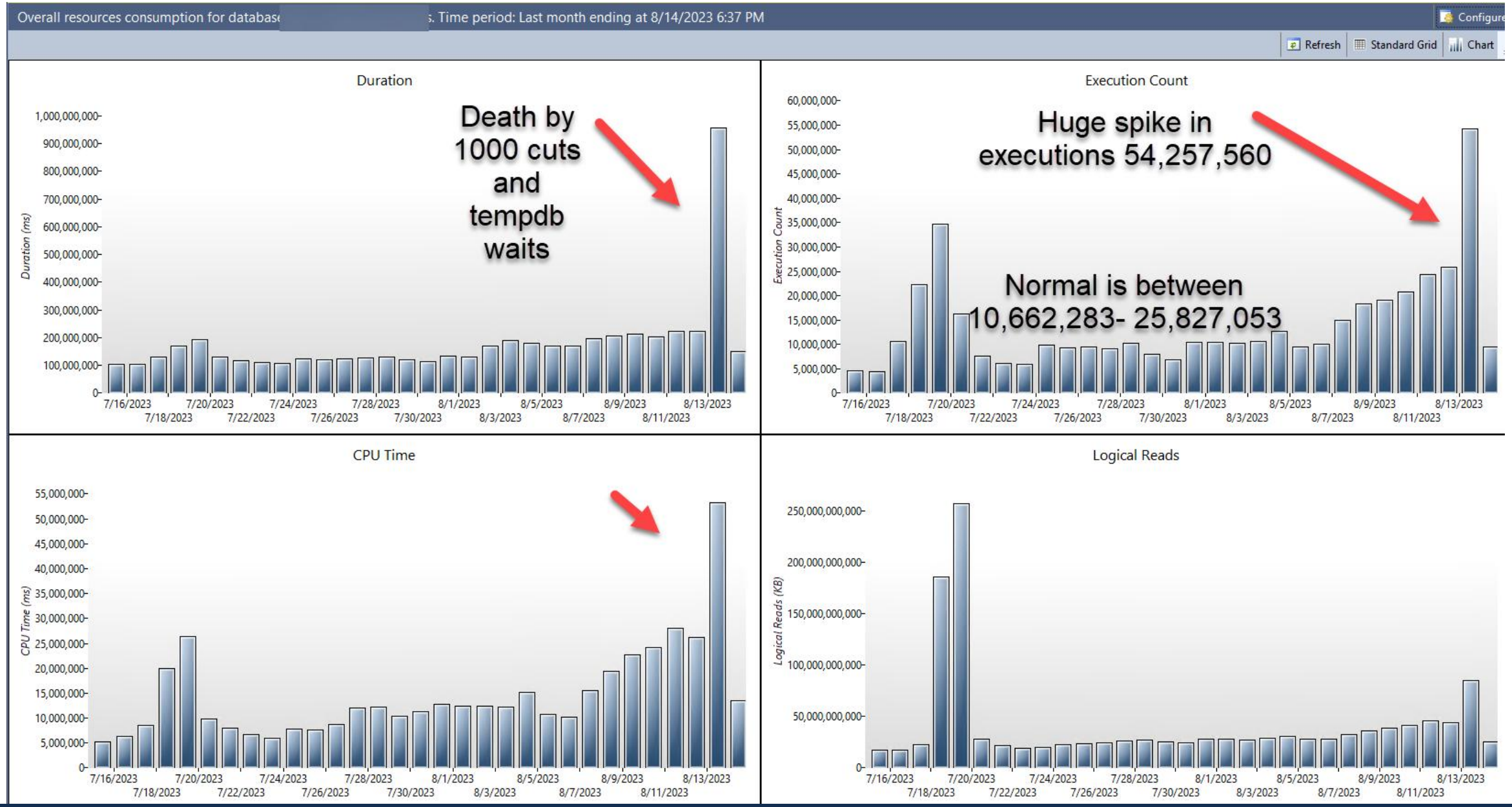


REGRESSED QUERIES





Plans that have gotten WORSE over a time period

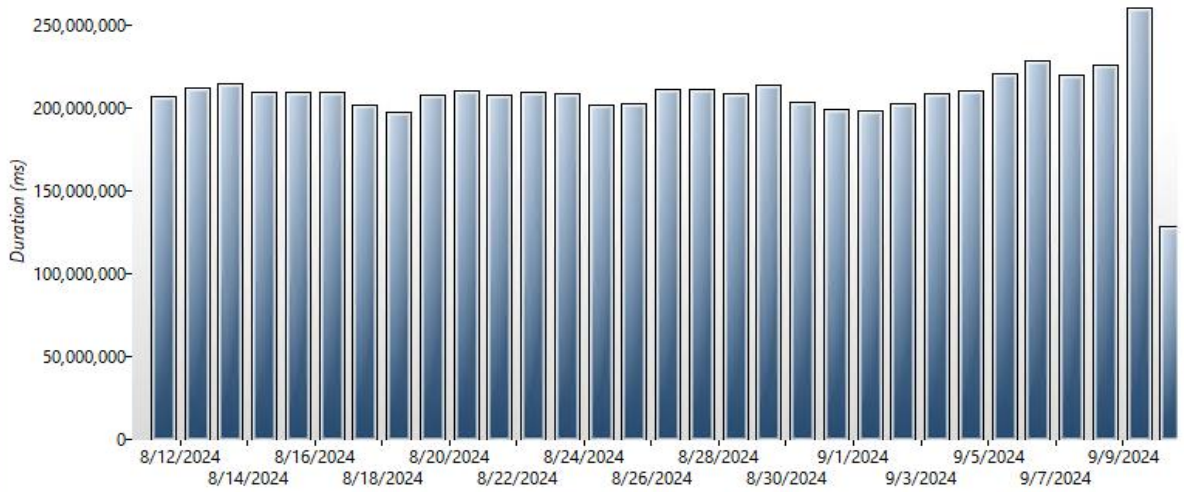


OVERALL RESOURCE CONSUMPTION

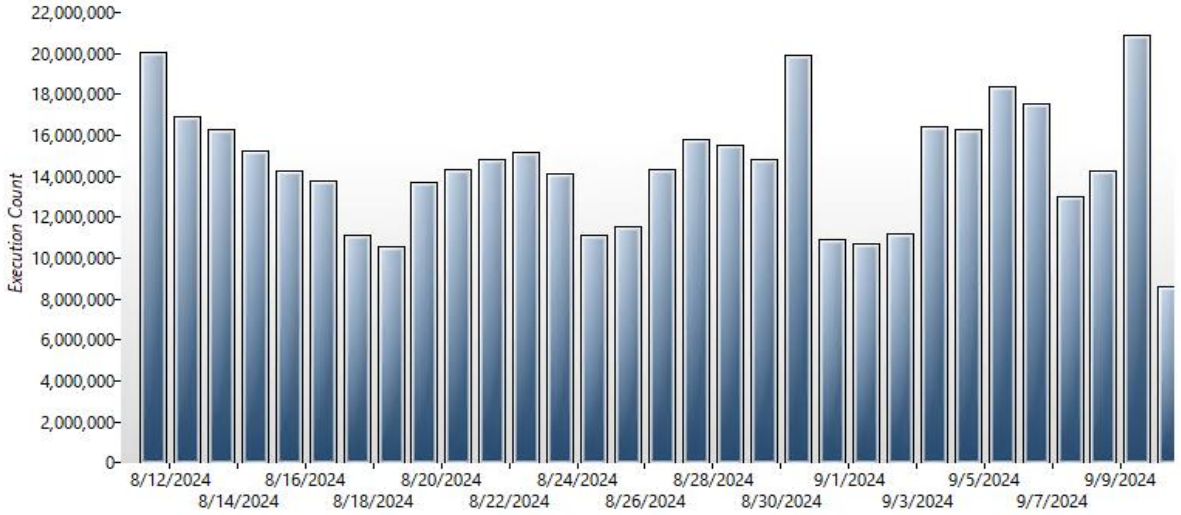
Overall resources consumption for databas | Time period: Last month ending at 9/10/2024 1:26 PM

Refresh | Standard Grid | Chart

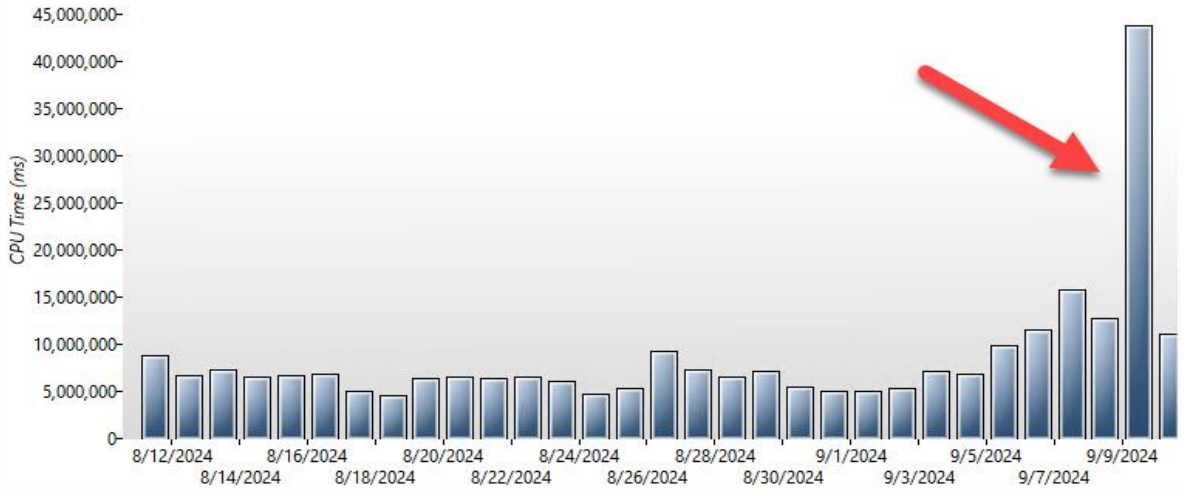
Duration



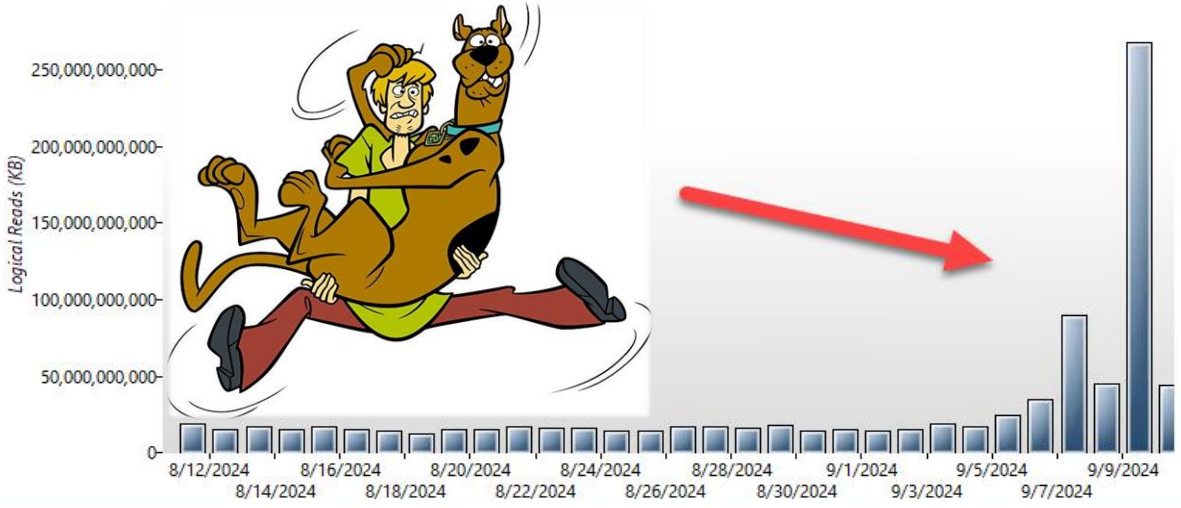
Execution Count



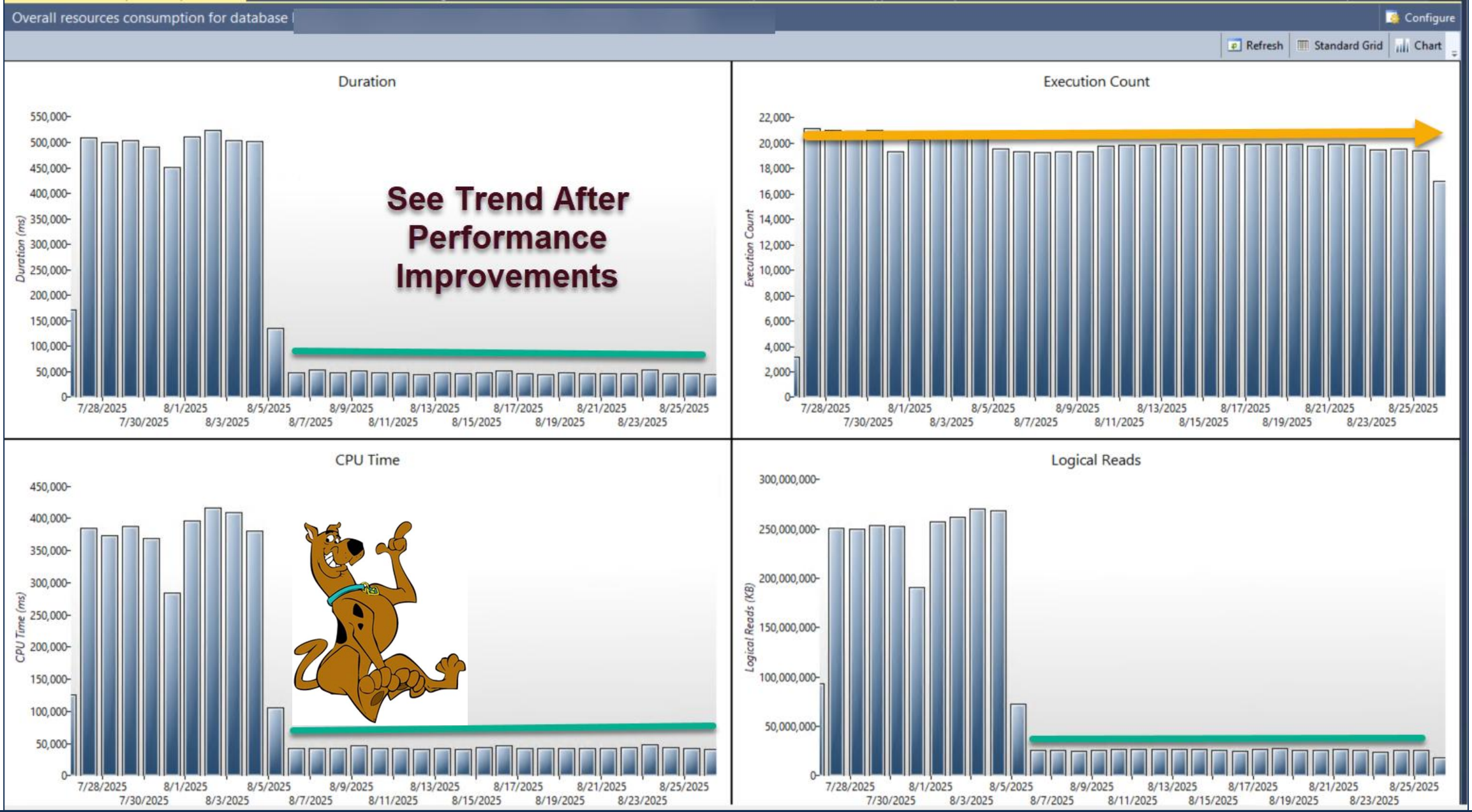
CPU Time



Logical Reads



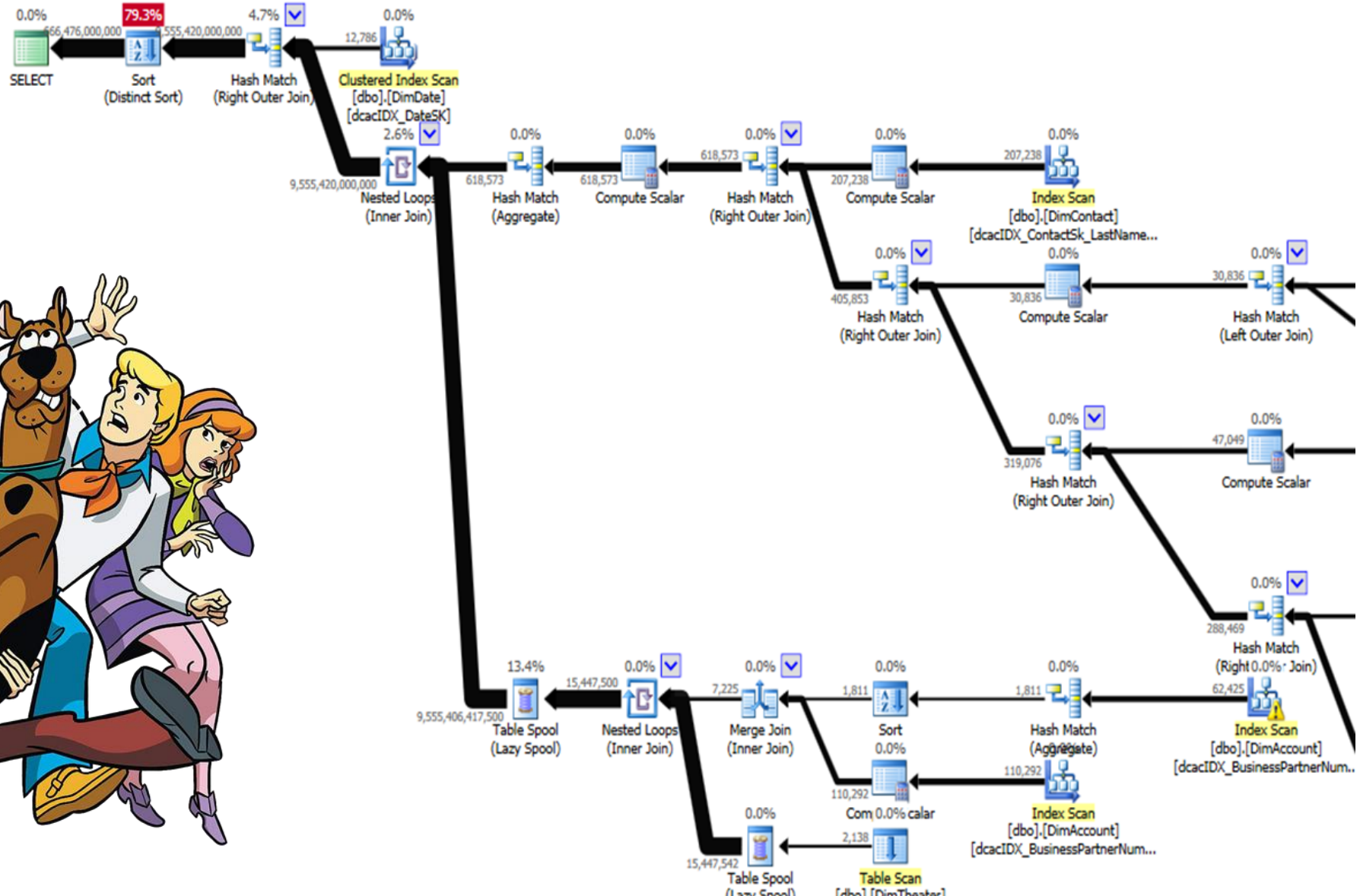
OVERALL RESOURCE CONSUMPTION



OVERALL RESOURCE CONSUMPTION

EXECUTION PLANS

Execution Plans: Reading the Evidence

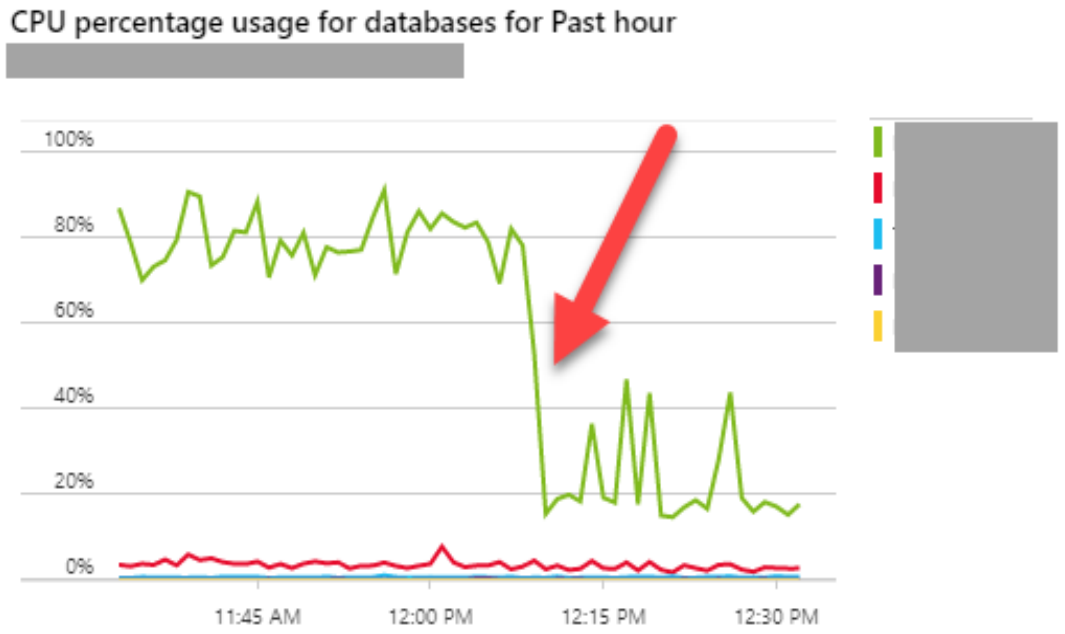


Execution Plans: Reading the Evidence

- Estimated vs Actual plans
- Large memory grants
- Key lookups & scans
- Sorts & hash matches
- Spills and warnings

MISSING INDEXES

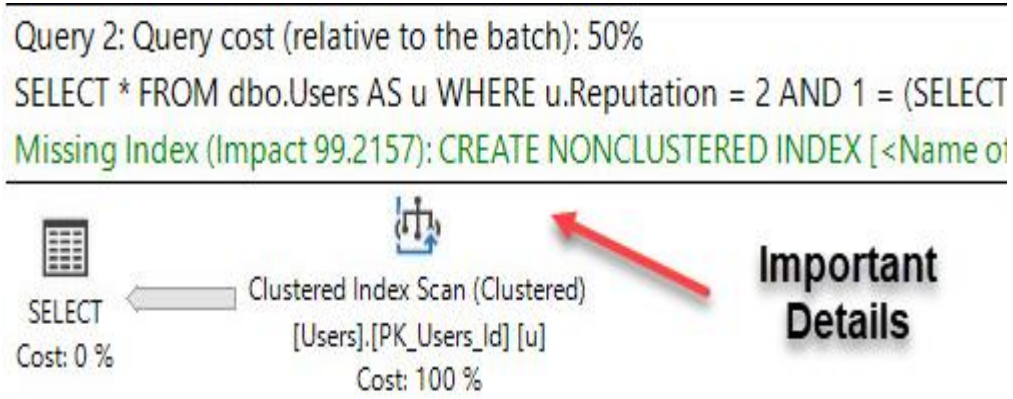
LAST SEEN: NEVER ….IMPACT COULD BE



HAVE YOU SEEN ME?

Impact: 99.2157
Table: Users
Operator Cost: 100%

Name: Names Matter



CAUTION! CONSIDER THEM ARMED & DANGEROUS – DON'T JUST ADD THEM

WAIT STATISTICS: Understanding the Motive

WHO'S HOGGING THE HALLWAY?

Wait Statistics: Understanding the Motive

CPU pressure (SOS_SCHEDULER_YIELD)

IO pressure (PAGEIOLATCH_*)

Locking (LCK_M_*)

Memory pressure (RESOURCE_SEMAPHORE)

TempDB contention (PAGELATCH_*)

HOW LONG HAVE THEY BEEN STUCK?

WAIT QUERIES

TOP 10 WAITS

```
/*Top waiting tasks*/
select top 10 wait_type, waiting_tasks_count, wait_time_ms,
max_wait_time_ms, signal_wait_time_ms
from sys.dm_os_wait_stats
order by waiting_tasks_count desc
go
/*Highest MAX wait time*/
select top 10 wait_type, waiting_tasks_count, wait_time_ms,
max_wait_time_ms, signal_wait_time_ms
from sys.dm_os_wait_stats
order by max_wait_time_ms desc
go
/*Highest total wait time*/
select top 10 wait_type, waiting_tasks_count, wait_time_ms,
max_wait_time_ms, signal_wait_time_ms
from sys.dm_os_wait_stats
order by wait_time_ms desc
```

GLENN BERRY QUERY #38

```
WITH [Waits]
AS (SELECT wait_type, wait_time_ms/ 1000.0 AS [WaitS],
          (wait_time_ms - signal_wait_time_ms) / 1000.0 AS [ResourceS],
           signal_wait_time_ms / 1000.0 AS [SignalS],
           waiting_tasks_count AS [WaitCount],
           100.0 *  wait_time_ms / SUM (wait_time_ms) OVER() AS [Percentage],
           ROW_NUMBER() OVER(ORDER BY wait_time_ms DESC) AS [RowNum]
    FROM sys.dm_os_wait_stats WITH (NOLOCK)
    WHERE waiting_tasks_count > 0)
SELECT
    MAX (W1.wait_type) AS [WaitType],
    CAST (MAX (W1.Percentage) AS DECIMAL (5,2)) AS [Wait Percentage],
    CAST ((MAX (W1.WaitS) / MAX (W1.WaitCount)) AS DECIMAL (16,4)) AS [AvgWait_Sec],
    CAST ((MAX (W1.ResourceS) / MAX (W1.WaitCount)) AS DECIMAL (16,4)) AS [AvgRes_Sec],
    CAST ((MAX (W1.SignalS) / MAX (W1.WaitCount)) AS DECIMAL (16,4)) AS [AvgSig_Sec],
    CAST (MAX (W1.WaitS) AS DECIMAL (16,2)) AS [Wait_Sec],
    CAST (MAX (W1.ResourceS) AS DECIMAL (16,2)) AS [Resource_Sec],
    CAST (MAX (W1.SignalS) AS DECIMAL (16,2)) AS [Signal_Sec],
    MAX (W1.WaitCount) AS [Wait Count],
    CAST (N'https://www.sqlskills.com/help/waits/' + W1.wait_type AS XML) AS [Help/Info URL]
FROM Waits AS W1
INNER JOIN Waits AS W2
ON W2.RowNum <= W1.RowNum
GROUP BY W1.RowNum, W1.wait_type
HAVING SUM (W2.Percentage) - MAX (W1.Percentage) < 99 -- percentage threshold
OPTION (RECOMPILE);
```

BLOCKING

HEY!

WHO'S HOLDING UP THE LINE?

BLOCKING

Lock Concurrency

One SPID holds a lock on resources while another SPID attempts to acquire the same resources

Identify Blocking

sp_WhoIsActive @find_block_leaders = 1;

sp_who2

Sys.dm_exec_requests

Sys.dm_os_waiting_task

SSMS Activity Monitor*

SSMS Reports

SQL Profiler

xEvents

	dd hh:mm:ss.mss	session_id	sql_text	login_name	wait_info	CPU	tempdb_allocations	tempdb_current	blocking_session_id	blocked_session_count
1	00 00:03:02.747	59	<?query -- SELECT SCHEMA_NAME(udf.schema_id) ...		(157312ms)LCK_M_S	220	0	0	58	0
2	00 00:02:59.683	58	<?query -- BEGIN TRAN CREATE TABLE AM2 (foo b...		NULL	0	0	0	NULL	4
3	00 00:02:46.427	60	<?query -- select * from Am2 --?>		(166394ms)LCK_M_SCH_S	24	0	0	58	2
4	00 00:02:05.857	61	<?query -- begin tran CREATE TABLE am3 (foo bit) ...		(125840ms)LCK_M_SCH_S	13	0	0	60	1
5	00 00:01:58.693	62	<?query -- select * from am3 --?>		(118687ms)LCK_M_SCH_S	3	0	0	61	0

KILL SPID 58

Great Reference by Andy Mallon

https://am2.co/2017/10/finding-leader-blocker/

WHAT NOT TO DO

What NOT To Do

Rebuild all indexes blindly

Update all stats daily without reason

Change MAXDOP without evidence

Throw hardware at bad query design

SQL UNMASKED

Building a Drama-Free SQL Environment

Enable and monitor Query Store

Baseline performance

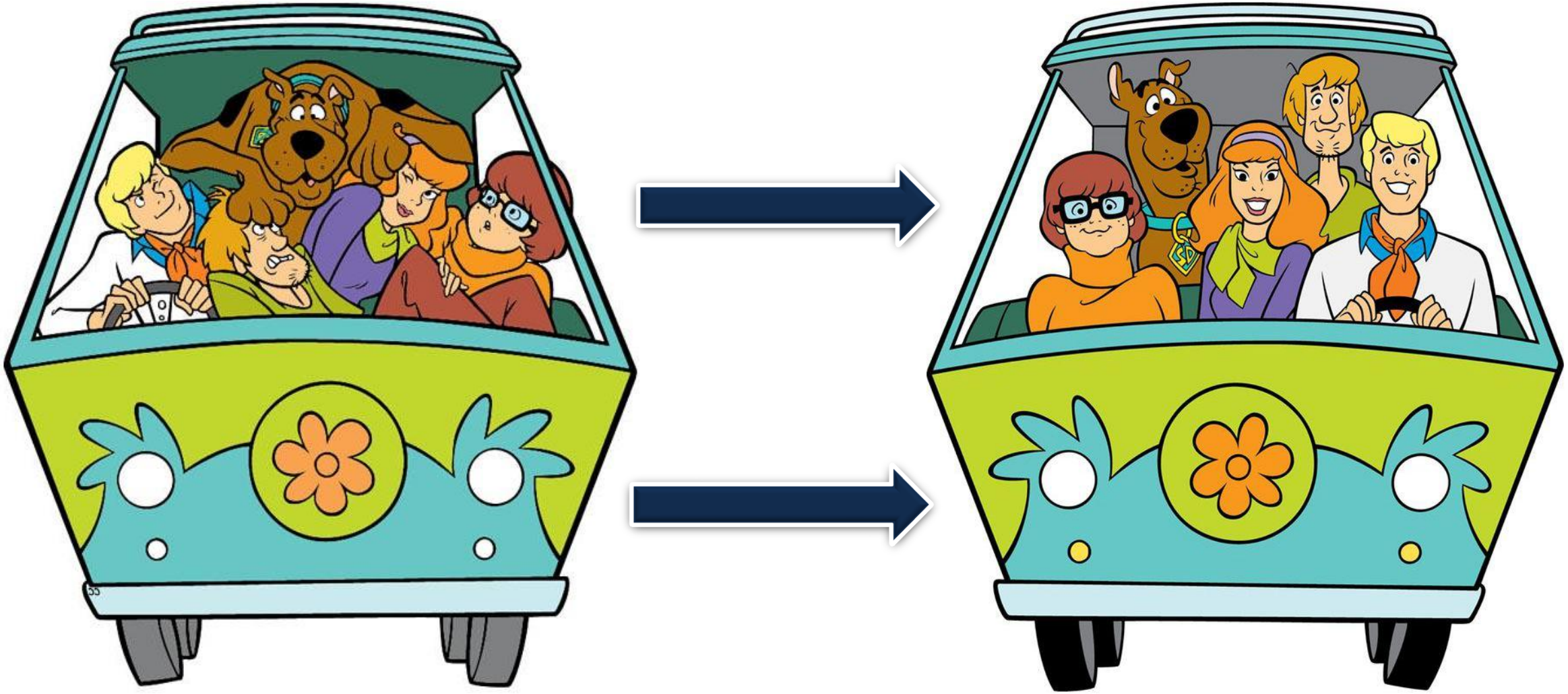
Monitor trends, not single incidents

Current Compatibility level

Automate health checks where possible

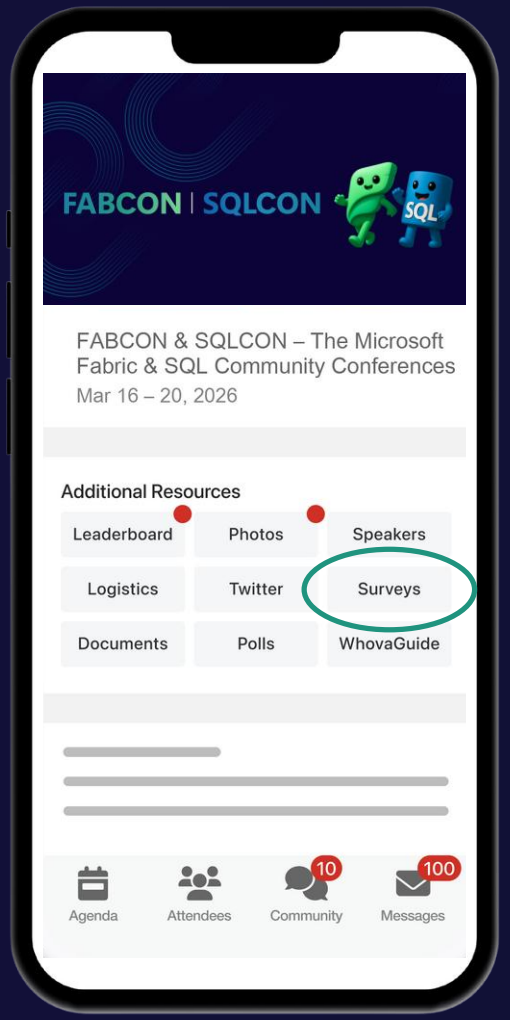
Performance isn’t magic
It’s measurable
It’s diagnosable
It’s solvable

Turning Chaos into Clarity



How was the session?

Complete Session Surveys in Whova for your chance to WIN PRIZES!



Get Two Fabric Certifications for FREE

Attendees of FABCON can take the Fabric Analytics Engineer or Fabric Data Engineer exam for free. Be part of the 2 fastest growing role-based certifications in Microsoft history.

Request your voucher by March 23, 2026.

https://aka.ms/fabcon/cert100

Monica Rathbun

MRathbun@sqlespresso.com

@SQLEspresso

sqlespresso.com

/in/sqlespresso

DCAC

Denny Cherry & Associates Consulting

Your Data, Our Expertise

www.dcac.com

SQLEspresso