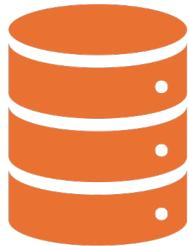


Memory, TempDB, CPU, I/O & Cloud Performance

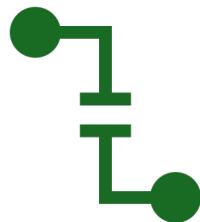
Surendra Panpaliya



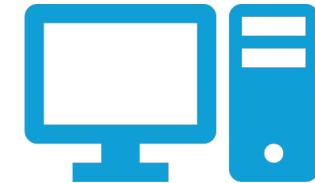
DAY 3 – Memory, TempDB, CPU, I/O & Cloud Performance



**Module 5: TempDB &
Memory Tuning**



**Module 6: Parallelism,
CPU & I/O Tuning**



**Module 7: Query Store, HA
& Azure SQL Performance**

Module 5: TempDB & Memory Tuning

PART 1:

WHAT IS TempDB?



TempDB



**Shared system
database**

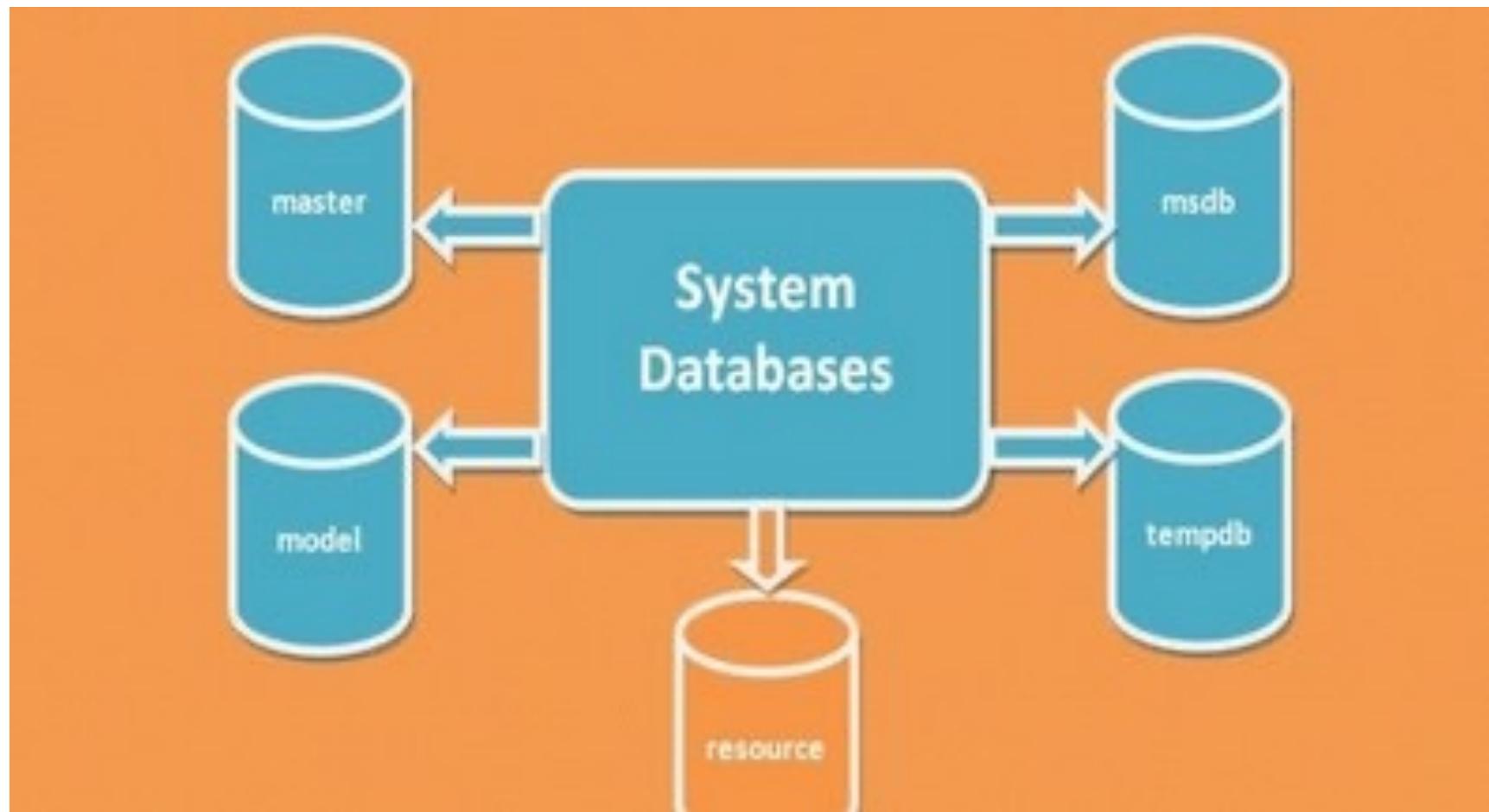


used by *every* database



on the Instance.

TempDB



TempDB use for

Sorts (ORDER BY)

Hash joins / hash aggregates

Spills (when memory is insufficient)

TempDB use for

Temporary objects (#temp tables, table variables)

Version store (Snapshot / RCSI)

Online index operations

Internal structures

TempDB uses special allocation pages:

PFS – Page Free Space

GAM – Global Allocation Map

SGAM – Shared Global Allocation Map

Internal structures



When many
sessions:



create temp tables



spill to TempDB



allocate/deallocate
pages

Real-Life Example



Think
of **TempDB** as:

Common
meeting room
in office

Everyone uses
it

If crowded →
delays

If poorly
managed →
chaos

How TempDB Is Used Internally?

Sorts (ORDER BY, GROUP BY, DISTINCT)

What happens

When SQL Server must **order rows**,

it uses a **Sort operator**.

Why TempDB is used?

SQL Server tries
to sort **in**
memory

If rows don't fit
in memory

→ **spill to**
TempDB

Execution plan signs

Sort operator

Yellow warning: “*Spill occurred*”

Waits: IO_COMPLETION, WRITELOG

Example

```
SELECT *  
FROM dbo.ComplianceFilings  
ORDER BY due_date;
```

Example

If:

No index on due_date

Large rowset

Example

SQL Server:

Reads rows

Allocates memory

If insufficient → writes intermediate rows to **TempDB**

DBA impact

TempDB writes increase

Query slows down

Disk latency matters

DBA fixes



CREATE INDEX
MATCHING ORDER BY



REDUCE RESULT SET



INCREASE MEMORY /
TUNE MEMORY GRANTS

2. Hash Joins / Hash Aggregates

What happens

Hash joins & aggregates need a **hash table** in memory.

Why TempDB is used

If the hash table grows beyond memory grant → **hash spill to TempDB**

Execution plan signs

Hash Match (Join / Aggregate)

Warning: “*Hash spill*”

Waits: RESOURCE_SEMAPHORE, IO_COMPLETION

Example

```
SELECT e.country_code, COUNT(*)  
FROM dbo.Entities e  
JOIN dbo.ComplianceFilings cf ON e.entity_id = cf.entity_id  
GROUP BY e.country_code;
```

Example

If:

Bad cardinality estimate

Missing index

Hash table too big → TempDB used for overflow

DBA fixes

Correct

Correct statistics

Add

Add indexes

Rewrite

Rewrite query (EXISTS instead of GROUP BY if possible)

3. Spills (Memory Insufficient)

What is a spill?

A **spill** happens when:

Memory grant < actual memory needed

Why TempDB is used

TempDB becomes the **overflow area**.

Common spill types

Operator	Spill Type
Sort	Sort spill
Hash Match	Hash spill
Window functions	Worktable spill

How to detect

Execution plan warnings

`sys.dm_exec_query_memory_grants`

TempDB IO spikes

DBA fixes

Fix	Fix bad estimates (stats)
Reduce	Reduce row width
Avoid	Avoid SELECT *
Tune	Tune memory grants

4. Temporary Objects (#temp tables, table variables)

What happens

#Temp tables → physically stored in TempDB

Table variables → metadata + data stored in TempDB
(modern versions)

Example

```
CREATE TABLE #Overdue
(
    entity_id INT,
    due_date DATE
);
```

Example

```
INSERT INTO #Overdue  
SELECT entity_id, due_date  
FROM dbo.ComplianceFilings  
WHERE filing_status = 'OVERDUE';
```

Example

→ All data pages live in **TempDB**

Table variable myth (important!)

✗ “Table variables don’t use TempDB”

✓ They DO, but with different statistics behavior.

DBA tips

Use

Use #temp for large data sets

Index

Index #temp tables

Avoid

Avoid unnecessary temp objects

1) Find the real tempdb object for #Overdue

Run this in the **same session** where you created #Overdue:

```
SELECT
    t.name AS tempdb_table_name,
    t.object_id
FROM tempdb.sys.tables t
WHERE t.name LIKE '#Overdue%';
```

2) See how many pages/extent allocations it is using (fast)

SELECT

```
OBJECT_NAME(p.object_id, DB_ID('tempdb')) AS object_name,  
p.index_id,  
au.type_desc AS allocation_type,  
au.total_pages,  
au.used_pages,  
au.data_pages  
FROM tempdb.sys.partitions p  
JOIN tempdb.sys.allocation_units au  
ON au.container_id = p.hobt_id  
WHERE p.object_id = OBJECT_ID('tempdb..#Overdue');
```

What this tells a DBA?

- `data_pages` = pages containing row data (8 KB each)
- `used_pages` = data + overhead
- `total_pages` = reserved

3) Get the actual page IDs (which pages in tempdb) — DBCC IND

- This shows the list of pages allocated.
- -- Get allocated pages for the object
- DBCC IND('tempdb', '#Overdue', -1);

Output includes:

- PagePID (page number)
- PageFID (file number)
- PageType (data / IAM / etc.)
- This is the classic way DBAs “see pages.”

CSC Example



Morning compliance dashboards



Heavy GROUP BY, ORDER BY



Parallel queries



Snapshot isolation enabled



Hundreds of sessions hit TempDB at the same time

CSC Example

DBAs see waits like:

PAGELATCH_UP

PAGELATCH_EX

Database name = tempdb

How DBAs detect TempDB latch contention

```
SELECT wait_type, waiting_tasks_count, wait_time_ms  
FROM sys.dm_os_wait_stats  
WHERE wait_type LIKE 'PAGELATCH_%'  
ORDER BY wait_time_ms DESC;
```

If **TempDB** is the hot spot → fix file configuration.

TempDB File Configuration

Why multiple TempDB data files?

Because:

Each data file has its own allocation pages

More files = less contention

CSC-recommended baseline

CPU Cores	TempDB Data Files
4	4
8	8
16	8-16
32+	Start with 8, monitor

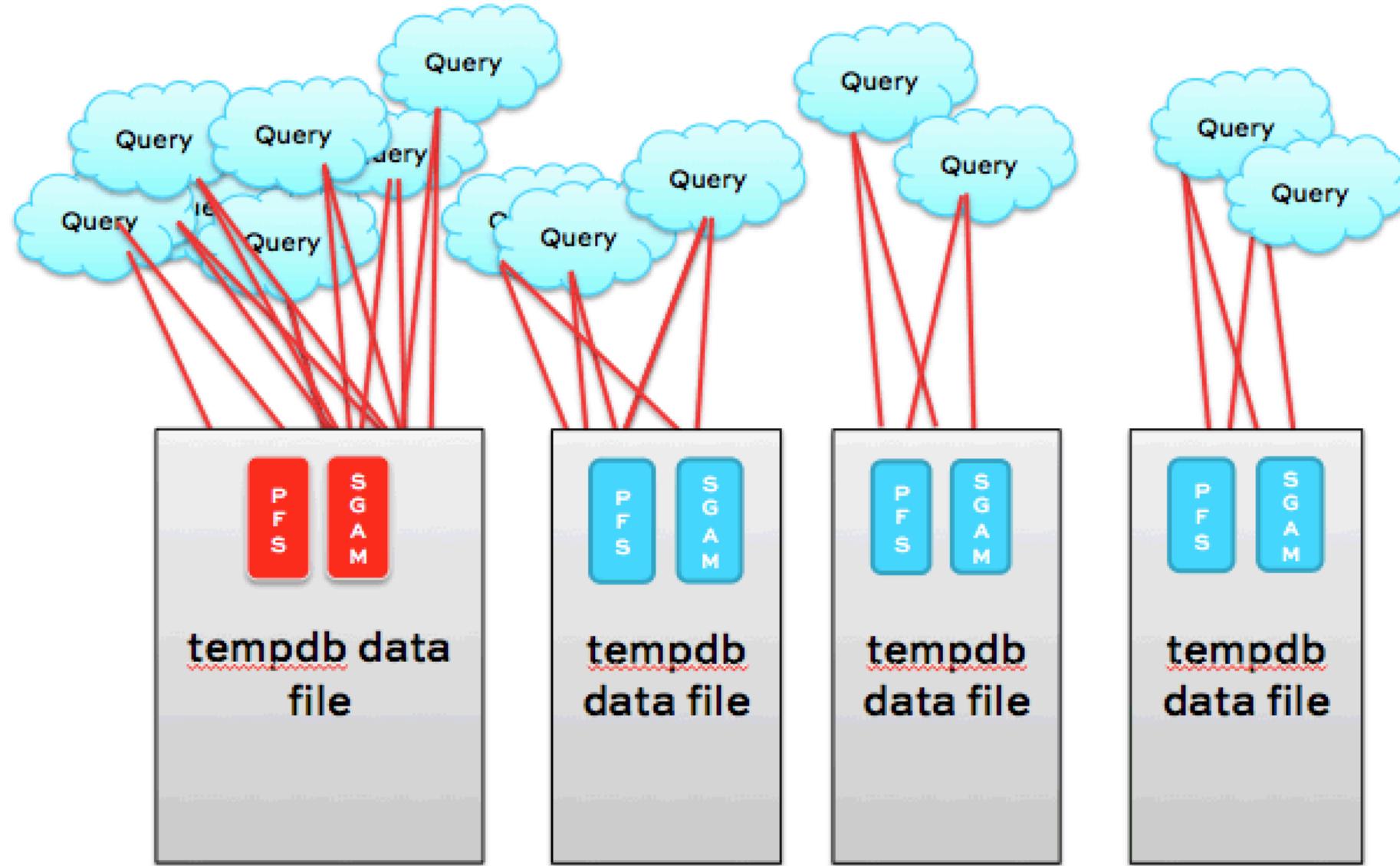
CSC-recommended baseline

Rules

All TempDB data files **same size**

Same autogrowth

Pre-size (avoid autogrowth storms)



Correct TempDB layout (example)

tempdev → 20 GB

tempdev2 → 20 GB

tempdev3 → 20 GB

tempdev4 → 20 GB

templog → 20 GB

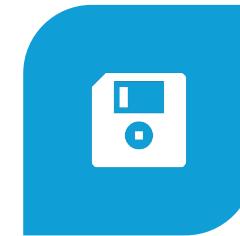
SQL Server Memory Components



A) BUFFER POOL
(DATA CACHE)



WHAT IT DOES



CACHES DATA
PAGES (8 KB)



REDUCES
PHYSICAL IO



USED BY ALMOST
EVERY QUERY

CSC Example

Compliance master tables

Reference tables

Hot indexes



If buffer pool is healthy → fast reads

CSC Example

Problem sign

High PAGEIOLATCH_SH

Lots of disk reads

B) Memory Grants (Query Execution Memory)

Used for:

Hash joins

Hash aggregates

Sorts

Allocated **before** query runs.

C) Workspace Memory

Subset of
memory grants

Used during
query execution

If insufficient →
**spill to
TempDB**

How DBAs see memory grants?

```
SELECT *
```

```
FROM sys.dm_exec_query_memory_grants
```

```
ORDER BY requested_memory_kb DESC;
```

Spill to TempDB (Major Performance Killer)

What is a spill?

When SQL Server:

estimates it needs X memory

actually needs more

runs out

→ It writes intermediate data to **TempDB**

Execution plan warning

Look for:

 “Spill
occurred during
execution”

Hash Match /
Sort operators

CSC Example

Compliance query:

```
SELECT country_code, COUNT(*)
```

```
FROM ComplianceFilings
```

```
GROUP BY country_code
```

```
ORDER BY COUNT(*) DESC;
```

CSC Example

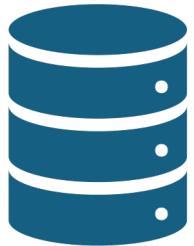
If:

Bad estimates

Stale stats

No covering index

CSC Example



Hash Aggregate spills
to TempDB



TempDB IO spikes



Query slows
dramatically

DBA Fixes

Update	Update statistics
Create	Create covering index
Reduce	Reduce columns
Rewrite	Rewrite query
Increase	Increase memory (carefully)

PART 2: TempDB ARCHITECTURE

ARCHITECTURE



TempDB contains:



Data files (tempdev)



Log file (templog)



Allocation pages (PFS, GAM, SGAM)

TempDB Latch Contention



What is a Latch?



Latch = short-term lock on **memory structure**



TempDB Latch Contention means:



Many queries trying to access same TempDB pages at the same time

Real-Life Example

Single washroom  in big office

→ long queues

→ everyone waits

LAB 1 – DETECT TempDB CONTENTION

Tool:

sys.dm_os_wait_stats

SELECT

wait_type,

wait_time_ms

FROM sys.dm_os_wait_stats

WHERE wait_type LIKE 'PAGELATCH%';

What CSC DBAs Look For

- PAGELOCK_UP on TempDB
- High wait times = contention

PART 3: TempDB FILE CONFIGURATION BEST PRACTICES

Simple Rules (CSC Standard)

Setting	Best Practice
Data files	Multiple (equal size)
Log file	Single, large
Auto-growth	Fixed MB, not %
Disk	Fast storage

Why Multiple Data Files?

Real-Life Example

- One juice counter → crowd
- Multiple counters → faster service

PART 4:

MEMORY COMPONENTS



1

Buffer Pool (Main Memory Area)



What is Buffer Pool?



Where SQL Server keeps **data pages**



Avoids repeated disk reads

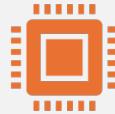
Real-Life Example



- Books kept on desk instead of store room.

2

Memory Grants (Query Reservation)



What is Memory Grant?



Before running a query, SQL Server asks:



“How much memory do I need to sort or join?”



That reserved memory is a **memory grant**.

Real-Life Example



Conference room:

- Chairs reserved before meeting
- Too many chairs → others wait
- Too few chairs → people stand (spill)

3

Workspace Memory

Memory used for:

- Hash joins
- Sort operations
- Part of memory grants.

PART 5: SPILL TO TempDB (COMMON CSC ISSUE)

What is a Spill?



A **spill** happens when:



Query needs more memory



Granted memory is insufficient



SQL Server uses TempDB instead

Real-Life Example



Suitcase full → clothes kept in extra bag

Impact of Spill

- Heavy TempDB usage
- Slow query
- Disk I/O increases

LAB 2 – DETECT SPILLS

Tool: Execution Plan

- Run any query with ORDER BY on large data.

Look for:

- **Sort Warnings**
- **Hash Warnings**

DMV:

sys.dm_exec_query_memory_grants

SELECT

```
    session_id,  
    requested_memory_kb,  
    granted_memory_kb,  
    used_memory_kb,  
    wait_time_ms
```

FROM sys.dm_exec_query_memory_grants;

How CSC Reads This

- requested > granted → memory pressure
- used > granted → spill risk

PART 6: TUNE MEMORY GRANTS (SAFE APPROACH)

Why Memory Grants Become Problem

- Bad statistics
- Parameter sniffing
- Large result sets

Fix Options (CSC Friendly)

- ✓ Update statistics
- ✓ Rewrite query
- ✓ Reduce columns
- ✓ Query Store plan forcing
- ✗ Avoid memory hints initially

LAB 3 – FIX MEMORY GRANT ISSUE



Step 1: Update Statistics



UPDATE STATISTICS Orders;



Step 2: Rerun Query



Observe:



Reduced spills



Lower TempDB usage

PART 7:

INTERNAL vs EXTERNAL

MEMORY PRESSURE



Internal Memory Pressure



SQL Server itself needs more memory:



Buffer pool eviction



Memory grants waiting

External Memory Pressure



Other processes consume memory:



Antivirus



Other databases



OS pressure

LAB 4 – CHECK MEMORY PRESSURE

Tool:

sys.dm_os_process_memory

SELECT

```
physical_memory_in_use_kb,  
large_page_allocations_kb,  
process_physical_memory_low,  
process_virtual_memory_low
```

FROM sys.dm_os_process_memory;

Interpretation

- process_physical_memory_low = 1 → external pressure

PART 8: MEMORY CLERKS (WHO USES MEMORY?)

What is a Memory Clerk?

Memory clerk = category of memory usage.

Examples:

- Buffer pool
- Query execution
- Cache

Tool

sys.dm_os_memory_clerks

SELECT

type,

pages_kb

FROM sys.dm_os_memory_clerks

ORDER BY pages_kb DESC;

CSC DBA Use



IDENTIFY MEMORY HOGS



VERIFY EXPECTED
USAGE

PART 9: CSC INCIDENT – END-TO- END

Problem



Slow reports



TempDB growth

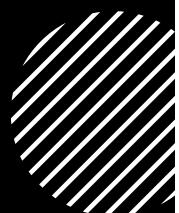


Query spills

Investigation

- ✓ Checked wait stats
- ✓ Found TempDB latch waits
- ✓ Found memory spills

Fix



✓ Added TempDB data files

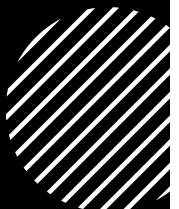


✓ Updated statistics



✓ Reduced memory grants

Result



✓ Stable system



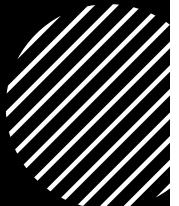
✓ Faster reports



✓ No restart needed



Summary



Term	Simple Meaning
TempDB	Shared work area
Latch	Short memory lock
Buffer Pool	Cached data
Memory Grant	Reserved RAM
Spill	Memory overflow
Internal Pressure	SQL Server needs memory
External Pressure	OS steals memory
Memory Clerk	Memory category



**Thank you for
your support and
patience**

Surendra Panpaliya
Founder and CEO
GKTC Innovations
<https://www.gktcs.com>