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

Surendra Panpaliya

Day 3

Module 5:

TempDB & Memory Tuning

Module 6:

Parallelism, CPU & I/O Tuning

Module 7:

Query Store, HA & Azure SQL Performance

Module 7:
Query Store, HA & Azure SQL Performance

What is Query Store?

Query Store is SQL Server's **built-in performance recorder**.

It stores:

- Query text
- Execution plans
- Runtime statistics (CPU, duration, reads)
- History **over time**

👉 Think of Query Store as **CCTV for queries**

What Query Store really does (internally)

Query Store is **not a single table.**

It is a **set of internal tables** stored

inside the user database,

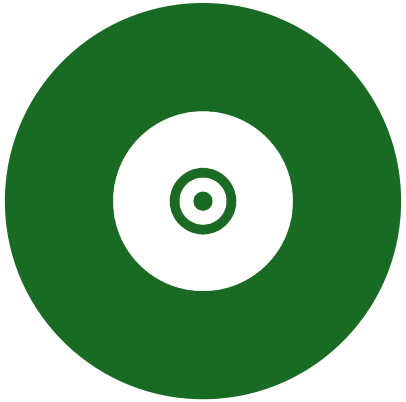
capturing:

What Query Store really does (internally)

Layer	What it stores	Why it matters to DBAs
Query Text	Normalized query text	Identifies logical queries
Query	One logical query	Parent object
Plan	Multiple execution plans	Plan variability
Runtime Stats	Execution metrics per interval	Performance over time
Wait Stats (2017+)	Query-level waits	Root cause (CPU, IO, locking)

Internally backed by:

MEMORY BUFFERS



BACKGROUND FLUSH TO DISK

CLEANUP BASED ON **RETENTION + SIZE**

Real-Life Example 📹

Mall CCTV:

- Shows who entered
- When problem started
- What changed
- Query Store does the same for queries

CSC DBA Example – OLTP Compliance System

Scenario

ComplianceFilings table

Same query runs thousands of times/day

Suddenly response time jumps from **50 ms → 5 seconds**

CSC DBA Example – OLTP Compliance System

WHAT QUERY STORE CAPTURES

SAME QUERY_ID

TWO DIFFERENT PLAN_ID

RUNTIME STATS CLEARLY SHOW **PLAN DIVERGENCE**

Key DMVs you must know

sys.query_store_query

sys.query_store_plan

sys.query_store_runtime_stats

sys.query_store_wait_stats

Internal risks DBAs must manage

Risk	Impact
Query Store = READ_ONLY	Stops capturing (disk full, size limit)
High write overhead	On very busy OLTP
Long retention	Bloated Query Store
Disabled cleanup	Performance regression

CSC Best Practices

Capture mode: **AUTO**

Size limit: **1–5 GB (based on DB size)**

Flush interval: Default OK

Cleanup policy: Keep 30–60 days max

HANDS-ON LAB 1 – CHECK QUERY STORE STATUS

```
SELECT actual_state_desc
FROM sys.database_query_store_options;
```

If OFF:

```
ALTER DATABASE YourDB
SET QUERY_STORE = ON;
```

PART 2: PLAN REGRESSION

What is Plan Regression?

Plan regression means:

Same query suddenly runs slower because SQL Server chose a different plan

What is Plan Regression?

- Same query + new plan = **slower performance**
- Reasons:
- Parameter sniffing
- Statistics update
- Index change
- Cardinality estimation changes
- AG failover

Common reasons:

Statistics update

Parameter sniffing

New index

Upgrade / deployment

Real-Life Example 🚗

Yesterday

Highway route → fast

Today

Same GPS → internal roads → slow

CSC Example – Month End Report

- **Before**
- Nested Loop
- Index Seek
- Duration: 200 ms

CSC Example – Month End Report

- **After**
- Hash Join
- Table Scan
- Duration: 30 seconds

How Query Store detects regression

- SQL Server:
- Compares **historical plan performance**
- Flags **worse plan**
- Can **recommend forced plan**

HANDS-ON LAB 2 – DETECT PLAN REGRESSION

Using Query Store

```
SELECT
    q.query_id,
    rs.avg_duration,
    rs.avg_cpu_time
FROM sys.query_store_runtime_stats rs
JOIN sys.query_store_query q
ON rs.query_id = q.query_id
ORDER BY rs.avg_duration DESC;
```

CSC Interpretation

- Same query_id
- Different performance over time
- Indicates regression

DBA Decision Matrix (CSC)

Situation	Action
Temporary spike	Observe
Consistent regression	Force plan
Parameter driven	Rewrite query
Stats driven	Update stats
Index related	Rebuild / redesign

PART 3: FORCED PLANS vs PLAN GUIDES

What is Plan Forcing?

You tell SQL Server:

“Use this known good plan for this query”

- ✓ Easy to apply
- ✓ Easy to revert
- ✓ Recommended by CSC

Forced Plan (Query Store)

What it does

Forces a specific plan **by plan_id**

Works at **engine level**

Survives restarts

```
EXEC sys.sp_query_store_force_plan
    @query_id = 123,
    @plan_id  = 456;
```

Forced Plan (Query Store)

- **Pros**
- Easy
- Safe
- Reversible
- Visible in Query Store UI

- **Cons**
- Plan may become invalid after schema changes

Plan Guides (Legacy)

- **What it does**
- Injects hints into query text
- Depends on **exact query match**
- EXEC sp_create_plan_guide
- @name = 'Guide1',
- @stmt = N'SELECT ...',
- @hints = N'OPTION (HASH JOIN)';

Problems

- Breaks with whitespace changes
- Hard to maintain
- Not cloud-friendly
- Deprecated mindset

CSC Recommendation

Use Case	Choose
SQL Server 2016+	Query Store Forced Plan
Vendor app (no control)	Query Store
Legacy SQL 2008	Plan Guide (only option)
Azure SQL / MI	Query Store ONLY

HANDS-ON LAB 3 – FORCE A PLAN

1. Identify good plan in Query Store
2. Force it
3. Monitor performance

```
EXEC sys.sp_query_store_force_plan
   @query_id = 10,
   @plan_id  = 25;
```

Plan Guides (OLD & RISKY)

👉 Query Store > Plan Guides

FEATURE	PLAN GUIDES
Introduced	Pre-Query Store
Flexibility	Low
Risk	High
CSC Usage	❌ Avoid

PART 4: ALWAYS ON AVAILABILITY GROUP (AG) PERFORMANCE

What is Always On AG?

Always On AG provides:

High Availability

Disaster Recovery

Readable replicas

Performance Impact (Simple Truth)

- AG adds overhead:
- Log records sent to secondary
- Redo happens on secondary

Key Metrics to Watch

Metric	Meaning
Log Send Queue	Logs waiting to be sent
Redo Queue	Logs waiting to be applied

LAB 4 – CHECK AG HEALTH

```
SELECT
   replica_server_name,
   log_send_queue_size,
   redo_queue_size
FROM sys.dm_hadr_database_replica_states;
```

CSC Interpretation

Observation	Meaning
High log_send_queue	Network / disk issue
High redo_queue	Secondary CPU / I/O slow

PART 5: READABLE SECONDARIES (SMART USAGE)

What Are Readable Secondaries?

Read-only copies of database

Used for reporting

Reduce load on primary

CSC Best Practice

✔️ Run heavy reports on secondary

❌ Do NOT run OLTP queries on secondary

Real-Life Example 🏢

- Cash counter → sales
- Information desk → queries

PART 6: AZURE SQL PERFORMANCE

Azure SQL is NOT On-Prem SQL

Key differences:

Resource limits

Throttling

Platform-managed tuning

Azure SQL Throttling

Throttling means:

Azure temporarily limits CPU, IO, or memory

Real-Life Example 🚦

Traffic police slows vehicles during congestion.

Azure SQL Auto-Tuning

Azure can:

- ✓ Create indexes automatically
- ✓ Drop unused indexes
- ✓ Force last good plan

Query Performance Insight (Azure Tool)

Shows:

- Top CPU queries
- Top duration queries
- Trends over time

LAB 5 – AZURE vs ON-PREM COMPARISON

Area	On-Prem	Azure SQL
MAXDOP	Manual	Platform managed
Memory	Fixed	Elastic
Tuning	DBA driven	Auto-tuning
Monitoring	DMVs	Portal + Query Store

PART 7: CSC INCIDENT – END-TO-END WALKTHROUGH

Problem

- Post-deployment slowdown
- No code change
- Same queries

Investigation

- ✓ Query Store showed plan change
- ✓ New plan had higher CPU
- ✓ AG redo queue normal

Fix

- ✓ Forced old stable plan
- ✓ Enabled Azure auto-tuning
- ✓ Moved reports to readable secondary

Result

- ✓ Performance restored in minutes
- ✓ No rollback
- ✓ Happy business users

Summary

Term	Simple Meaning
Query Store	Query history recorder
Plan Regression	Query got slower
Forced Plan	Lock good plan

Surendra Panpaliya

Founder and CEO

GKTCS Innovations

https://www.gktcs.com