https://www.sqldbachamps.com
https://github.com/PMSQLDBA/PraveenMadupu
Telegram Group→SQLDBACloud-DevOps https://t.me/+r2OYsT2qRZJkYWVl

Praveen Madupu
Mb: +91 98661 30093
Database Administrator

Native (inbuilt) SQL Server monitoring tools available in **SQL Server 2019, 2022, and likely 2025**, with detailed **features, real-time scenarios, and practical DBA use cases.**

🧭 Overview: Why Inbuilt Monitoring Tools Matter

Microsoft SQL Server comes with several **built-in tools** to help DBAs **monitor, diagnose, and troubleshoot** performance issues **without needing 3rd-party tools**.

These are:

- Free and included with SQL Server installation
- Integrated with SQL Server Management Studio (SSMS) or Azure Data Studio
- Extendable through Dynamic Management Views (DMVs), Extended Events, and PowerShell automation

⚙️ Top Inbuilt SQL Server Monitoring Tools (2019–2025)

#	Tool	Type	Availability	Key Purpose
1	**SQL Server Management Studio (SSMS) Activity Monitor**	GUI	SSMS	Live performance snapshot
2	**Dynamic Management Views (DMVs) & Functions (DMFs)**	T-SQL	All editions	Deep diagnostic queries
3	**SQL Server Profiler**	GUI	All editions (deprecated after 2022)	Real-time query trace
4	**Extended Events (XEvents)**	GUI & T-SQL	All editions	Lightweight tracing framework
5	**Performance Monitor (PerfMon)**	OS-level	Windows	Host system + SQL counters
6	**Query Store**	Database feature	2016+	Query performance tracking
7	**SQL Server Error Logs & Agent Logs**	Log files	All editions	Historical event auditing
8	**Azure Data Studio Performance Insights**	GUI	2019+ (Azure + on-prem)	Visual performance trends
9	**Database Mail Alerts & Operators**	SQL Agent	All editions	Automated alert notifications
10	**Resource Governor**	Engine-level	Enterprise	Workload prioritization and monitoring

🧩 Detailed Explanation, Real-time Scenarios & Use Cases

1️⃣ SQL Server Activity Monitor (SSMS)

Where: Right-click server in SSMS → *Activity Monitor*
Features:

- Real-time CPU, I/O, and memory graphs
- Top expensive queries
- Active sessions and blocking chains
- Resource waits visualization

Real-time Scenario:
📍 Users report "SQL Server is slow."
✅ You open Activity Monitor → see a query consuming 95% CPU.
Query plan shows missing index → DBA creates an index and adds a query hint → performance stabilizes.
Use Case:

- Quick troubleshooting of spikes or locks.
- Identify which sessions are blocking others.

2️⃣ Dynamic Management Views (DMVs) and Functions (DMFs)

Where: sys.dm_exec_*, sys.dm_os_*, sys.dm_db_*, etc.

Features:

- Provide internal performance metrics.
- Examples:
  - sys.dm_exec_requests → current queries
  - sys.dm_os_wait_stats → wait bottlenecks
  - sys.dm_io_virtual_file_stats → disk latency
  - sys.dm_exec_query_stats → top CPU-consuming queries

Real-time Scenario:

📍 CPU utilization > 90%.

✅ Query sys.dm_exec_query_stats ordered by total_worker_time → find top CPU consumers.

Tune or rewrite the query; add missing indexes.

Use Case:

- Core component of any health check script.
- Used by DBAs to automate daily checks and create custom dashboards.

3️⃣ SQL Server Profiler (Trace)

Where: SSMS → Tools → SQL Server Profiler

Features:

- Captures real-time SQL activity.
- Events like SQL:BatchStarting, RPC:Completed, Deadlocks.
- Can save traces to a file or table for analysis.

Real-time Scenario:

📍 You need to audit what queries a third-party app sends to SQL Server.

✅ Use Profiler to trace all statements by that app's login.

Later analyze for optimization or compliance.

Use Case:

- Debugging query-level issues.
- Reverse engineering app interactions with DB.

⚠️ **Note:** Deprecated after SQL Server 2022 → replaced by **Extended Events**.

4️⃣ Extended Events (XEvents)

Where: SSMS → Management → Extended Events → Sessions

Features:

- Lightweight tracing mechanism (replaces Profiler).
- Captures deadlocks, waits, errors, performance events.
- Integrates with SSMS UI (Live Data Viewer).

Real-time Scenario: 📍 Application experiences intermittent timeouts.

✅ You create an Extended Event session capturing "rpc_completed" and "sql_batch_completed" events filtered by duration > 5 seconds.

Find slow queries, review execution plans, fix root cause.

Use Case:

- Long-running query capture.
- Deadlock and wait troubleshooting.
- Real-time and historical performance diagnostics.

5️⃣ Windows Performance Monitor (PerfMon)

Where: perfmon.exe → Add counters → SQLServer:* objects

Key Counters:

- SQLServer:Buffer Manager → Page life expectancy
- SQLServer:Locks → Deadlocks/sec
- SQLServer:SQL Statistics → Batch Requests/sec
- LogicalDisk, Processor, Memory counters

Real-time Scenario:📍 Disk I/O seems slow.

✅ You monitor Avg. Disk sec/Read in PerfMon → find latency > 20ms.
Move database files to faster storage → confirm latency drop.

Use Case:

- OS + SQL integrated monitoring.
- Baseline performance metrics collection.

6️⃣ Query Store

Where: Database → Properties → Query Store

Features:

- Tracks query plans and runtime stats over time.
- Automatically detects performance regressions.
- Allows forcing “known good” plans.

Real-time Scenario:

📍 After an upgrade, reports run slower.
✅ Query Store shows new plan with higher cost → DBA forces previous stable plan → query speed returns to normal.

Use Case:

- Plan regression detection.
- Trend analysis (query duration, CPU, I/O).
- Ideal for 2019+ and Azure SQL DB.

7️⃣ Error Logs & Agent Logs

Where:

- SSMS → Management → SQL Server Logs
- SQLAgent → Error Logs
- Default path: C:\Program Files\Microsoft SQL Server\MSSQL*\MSSQL\Log\

Features:

- Records startup, backup, restore, failed logins, deadlocks, etc.
- Helps troubleshoot crashes and job failures.

Real-time Scenario:

📍 SQL Agent job failed last night.
✅ Review Agent log → find “Timeout expired” error.
Query logs show missing linked server → fix connection.

Use Case:

- Root cause analysis for job or server failures.
- Tracking unauthorized login attempts.

8️⃣ Azure Data Studio Performance Insights

Where: Azure Data Studio → “Insights” dashboard

Features:

- Lightweight, cross-platform tool.
- Visual performance charts (CPU, waits, IO).
- Integration with Notebooks for automation.

Real-time Scenario:

📍 Developer requests quick performance overview for dev server (Linux).
✅ Use Azure Data Studio → CPU usage + top queries displayed graphically.
Share findings via a notebook.

Use Case:

- Developer-friendly monitoring.
- Linux-based SQL Server monitoring (2019+).

9️⃣ SQL Agent Alerts, Operators, and Database Mail

Where: SQL Server Agent → Alerts → Operators

Features:

- Sends email/SMS notifications on errors or thresholds.
- Can trigger on severity levels (e.g., error 823, 824).

Real-time Scenario: 📍 A critical backup job fails at midnight.

✅ DBA gets auto-email from SQL Agent Alert "Job Failed: DB_BACKUP".
DBA checks issue remotely and reruns the job.

Use Case:

- Proactive monitoring and alerting system.
- Key for 24×7 production systems.

🔟 Resource Governor

Where: SSMS → Management → Resource Governor

Features:

- Controls CPU and memory usage per workload group.
- Monitors and prevents resource starvation.

Real-time Scenario:

📍 One analytics query hogs all CPU.
✅ DBA configures Resource Governor to limit analytics workload to 40% CPU.
OLTP workload stays responsive.

Use Case:

- Multi-tenant or shared environment workload control.
- Prevent runaway queries from hurting production.

🧠 Bonus: Other Useful Native Tools (for Advanced DBAs)

Tool	**Description**	**Use Case**
Database Tuning Advisor (DTA)	Recommends indexes/statistics based on workload	Post-query tuning
SQL Server Agent	Job scheduling and alerting	Automate maintenance, monitor job health
System Health Extended Event Session	Always-on lightweight session capturing errors, waits, deadlocks	Continuous background monitoring
TempDB Usage DMV (sys.dm_db_file_space_usage)	Tracks tempdb space allocation	Detect tempdb contention
sp_WhoIsActive (community proc)	Not native, but often used with DMVs	Real-time active sessions & blocking

🧰 Real-Time Monitoring Workflows (2019–2025)

Scenario	**Native Tool Used**	**DBA Workflow**
CPU spike	Activity Monitor / DMVs	Identify top queries → check plans → optimize
Long-running queries	Extended Events / Query Store	Capture and analyze performance over time
Blocking sessions	sp_WhoIsActive / DMVs	Identify blocking chains → resolve root cause
Disk bottleneck	PerfMon + DMV sys.dm_io_virtual_file_stats	Compare disk latency and throughput
Deadlocks	Extended Events / Error Log	Capture and visualize deadlock graph
Query regression post-upgrade	Query Store	Force previous execution plan
Memory pressure	Resource Governor / DMV	Review buffer usage, adjust workload limits

📜 Summary Table: Tool Comparison

Tool	**Real-time?**	**Historical Data?**	**Cloud/Hybrid Support**	**Performance Overhead**	**Skill Level**
Activity Monitor	✅	❌	✅	Low	Beginner
DMVs	✅	✅ (custom scripts)	✅	Low	Intermediate
Profiler	✅	❌	⚠️ Deprecated	High	Intermediate
Extended Events	✅	✅	✅	Low	Intermediate–Advanced
PerfMon	✅	✅	✅	Low	Intermediate
Query Store	❌	✅	✅	Low	Beginner
Error Logs	❌	✅	✅	None	Beginner
Resource Governor	✅	✅	✅	Low	Advanced
Azure Data Studio Insights	✅	✅	✅	Low	Beginner
SQL Agent Alerts	❌	✅	✅	Low	Intermediate

🧠 **Interview-Ready Summary Answer**

“In SQL Server 2019–2025, I primarily use **Activity Monitor** and **DMVs** for real-time analysis, **Query Store** and **Extended Events** for deeper diagnostics, and **SQL Agent Alerts** for proactive monitoring.

For example, once we faced sudden CPU spikes. Using sys.dm_exec_query_stats, I found a query consuming most CPU time — tuning that query reduced CPU from 95% to 40%.

I also leverage **Query Store** to compare plan regressions after upgrades, and **PerfMon** to correlate OS-level bottlenecks with SQL metrics.”