



# INTRODUCTION TO QUERY STORE

Carlos Robles

MSc, MCP, MCTS, MCSA, MCSE, ITIL v3

Multi-platform DBA



[@dbamastery](https://twitter.com/dbamastery)



[dbamastery@gmail.com](mailto:dbamastery@gmail.com)



[www.dbamastery.com](http://www.dbamastery.com)

[www.sqlguatemala.com](http://www.sqlguatemala.com)

# AGENDA

- Introduction
- What is Query Store
- Architecture Overview
- Configuration
- How to use Query Store
- Demo
- Best Practices
- Considerations

# REAL WORLD SCENARIOS ...

- After upgrading a SQL Server instance to latest version, had issues with plan changes slowing down the front end application
- Had a performance problem with a database and was unable to determine the root cause because someone decided to reboot the server
- Had a third party application experiencing performance problems but no changes to the database schema are not allowed

- Had an application down / slow because of the database is not performing well, upper management is expecting you to push the go faster button
- Had an application that only uses AD-HOC and dynamic SQL queries
- **Guess what?** Query store can help you with all these situations

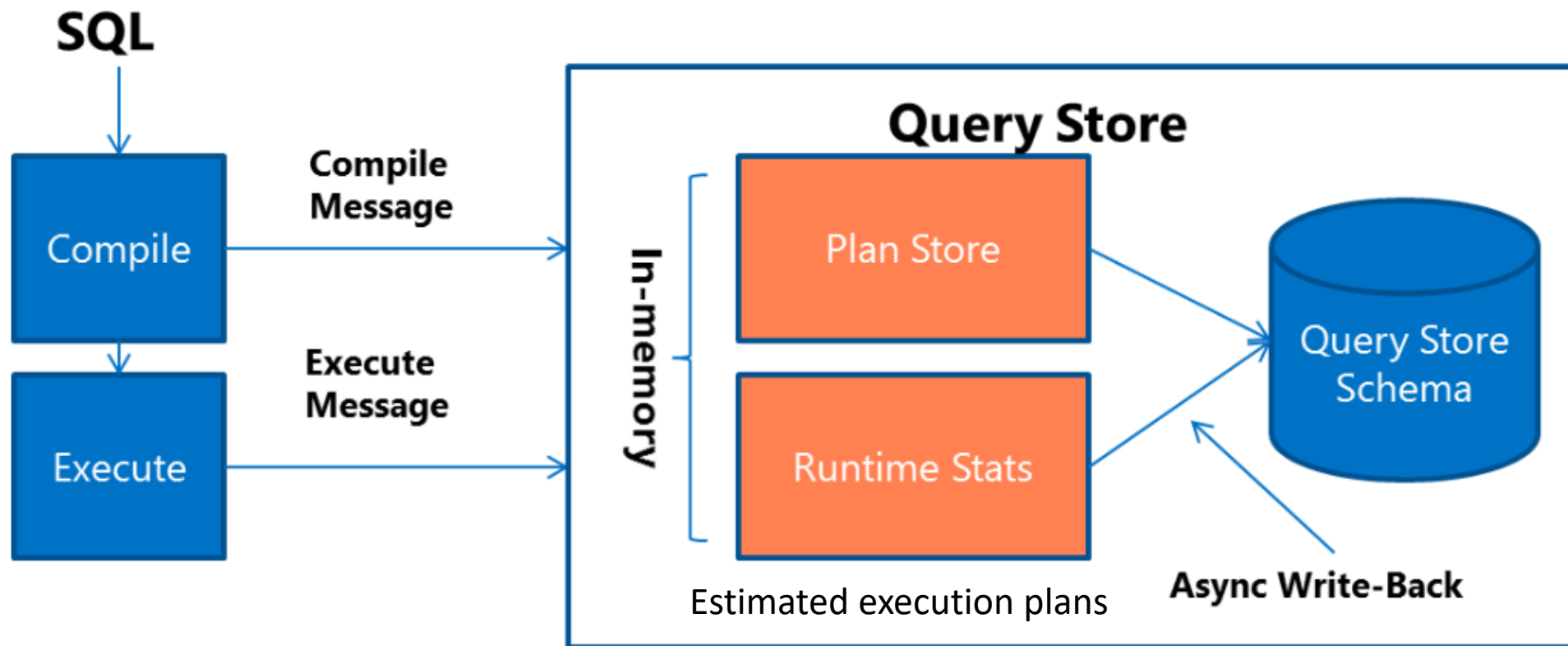
# WHAT IS QUERY STORE

- SQL Server flight data recorder
- Captures query plan, runtime and wait statistics
- It simplifies troubleshooting by helping to find performance differences caused by query plan changes
- It aggregates data by time windows so database usage patterns can be identified
- Available for all editions - SQL Server 2016 and Azure SQL DB

# WHAT MAKES QUERY STORE SO GOOD

- What is the added value?
- Makes query tuning less complex
- The previous solutions were not part of SQL Server engine
- Tracks information independent of what is cached
- Stores every SQL statement from a batch
- Works at database level
- Easy plan forcing (it depends ...)

# ARCHITECTURE OVERVIEW



Aggregation based on Interval length minutes

Aggregated data flush → Async checkpoint mechanism

# CONFIGURATION

- It is enabled at database level (master, TempDB or model)
- Starting from SQL Server 2016 and Azure SQL Database v12
  - All SQL Server editions
- Permissions
  - VIEW DATABASE STATE
  - DB\_OWNER
- T-SQL

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



General	
Enable	True
Operation Mode (Actual)	Read Write
Operation Mode (Requested)	Read Write
Monitoring	
Data Flush Interval (Minutes)	15
Statistics Collection Interval	1 Hour
Query Store Retention	
Max Size (MB)	100
Query Store Capture Mode	All
Size Based Cleanup Mode	Off
Stale Query Threshold (Days)	367

**Statistics Collection Interval**  
The granularity at which runtime statistics are aggregated.

Current Disk Usage

DBOne	250.0 GB	Query Store Available	100.0 MB
Query Store Used	0.0 MB	Query Store Used	0.0 MB

Purge Query Data

```
SELECT * FROM  
sys.database_query_store_options;
```

# OPTIONS RELATED TO DATA COLLECTION

- Operation mode
  - Default Read-Write
  - Read Only
- Query capture mode
  - Default ALL for SQL Server 2016+
  - Default AUTO for Azure SQL Server DB
  - NONE
- Max plans per query
  - Default 200

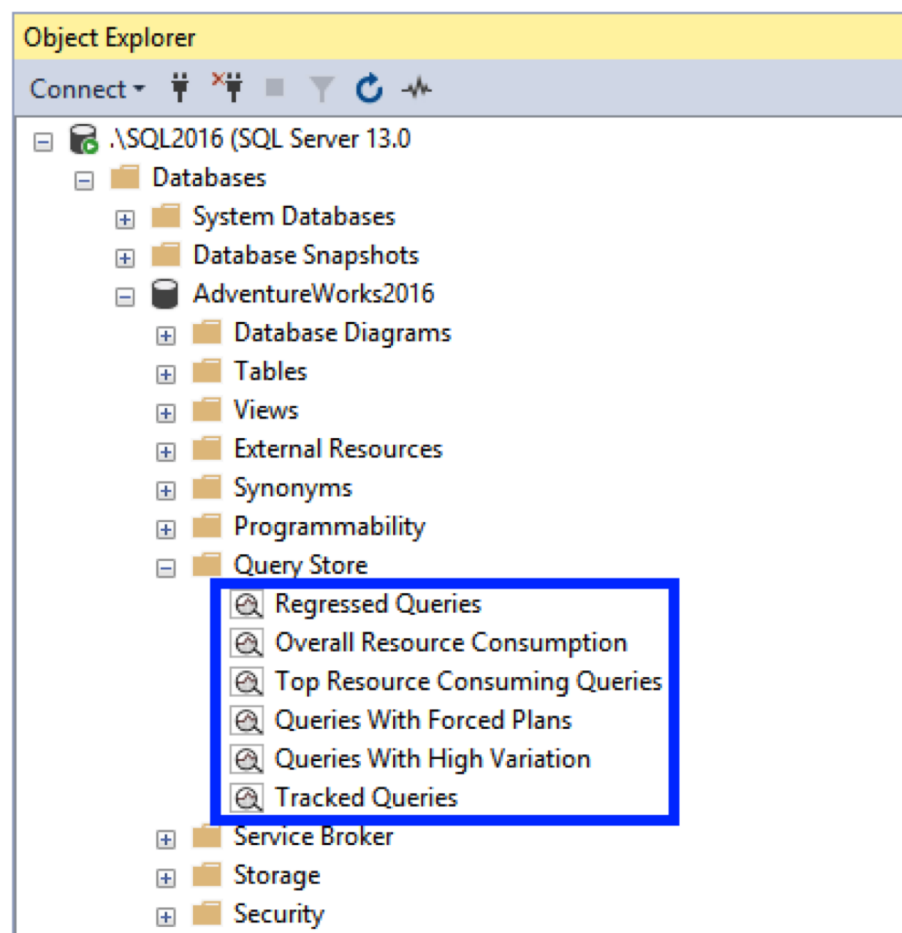
# OPTIONS RELATED TO DATA STORE

- Max storage size MB
  - Default 200 MBs
- Cleanup policy
  - Stale query threshold in days
  - Default 30 days
- Size based cleanup mode
  - AUTO
  - OFF

# OPTIONS RELATED TO DATA AGGREGATION

- Interval length minutes
  - Default 60 minutes
  - 1, 5, 10, 15, 30, 60, 1,440 (24 hours)
  - Lower values, more granularity
- Data flush interval
  - Default 15 minutes
  - Flushing runtime and wait stats data from memory to disk
  - In case of memory pressure data will be flushed to disk immediately

# HOW TO USE QUERY STORE

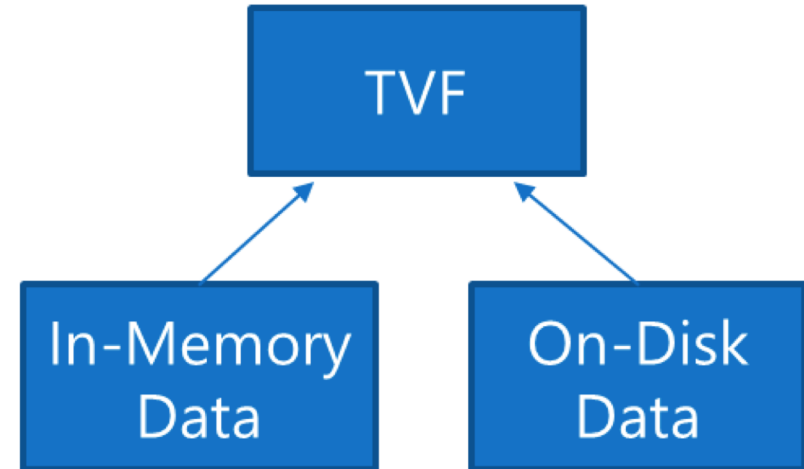
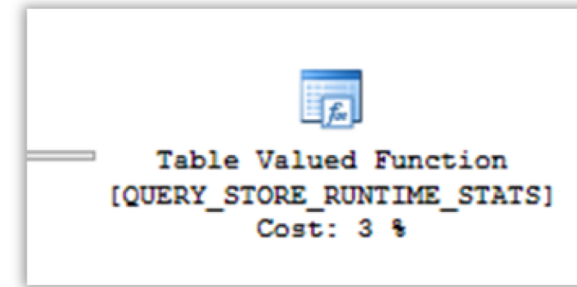


SSMS view	Scenario
Regressed Queries	Pinpoint queries for which execution metrics have recently regressed (i.e. changed to worse). Use this view to correlate observed performance problems in your application with the actual queries that needs to be fixed or improved.
Overall Resource Consumption	Analyze the total resource consumption for the database for any of the execution metrics. Use this view to identify resource patterns (daily vs. nightly workloads) and optimize overall consumption for your database.
Top Resource Consuming Queries	Choose an execution metric of interest and identify queries that had the most extreme values for a provided time interval. Use this view to focus your attention on the most relevant queries which have the biggest impact to database resource consumption.
Queries With Forced Plans	Lists previously forced plans using Query Store. Use this view to quickly access all currently forced plans.
Queries With High Variation	Analyze queries with high execution variation as it relates to any of the available dimensions, such as Duration, CPU time, IO, and Memory usage in the desired time interval. Use this view to identify queries with widely variant performance that can be impacting user experience across your applications.
Tracked Queries	Track the execution of the most important queries in real-time. Typically, you use this view when you have queries with forced plans and you want to make sure that query performance is stable.

## Query Store Catalog Views

sys.query\_store\_query  
sys.query\_store\_plan  
sys.query\_store\_query\_text  
sys.query\_context\_settings

sys.query\_store\_runtime\_stats  
sys.query\_store\_wait\_stats  
sys.query\_store\_runtime\_stats\_interval



# QUERY STORE CATALOG VIEWS

- Plan store
  - Query plan & query text
  - Query context settings
- Runtime stats store
  - Compilation time, duration & last execution
  - CPU & DOP
  - Logical reads & physical reads
  - Writes
  - Wait statistics (SQL Server 2017)

# PLAN GUIDES VS QUERY STORE

- Plan guide
  - Forces ad-hoc queries or stored procedures
    - Object, SQL and Template
  - Add hints without code changes
  - Complex
  - Plan verification (XML)
- Query Store
  - Forces ad-hoc queries or stored procedures
  - No hints, changes in query text changes query id
  - No plan verification (XML)



**DEMO**

# RECAP

- Pinpoint and fix queries with plan choice regressions
- Identify and tune top resource consuming queries
- A/B testing, application changes
- Keep performance stability during the upgrade to newer SQL Server

# BEST PRACTICES

- SSMS latest version
  - Update file in case SSMS is already installed
- XE monitoring
- Trace flags
  - -T7745: Forces Query Store to not flush data to disk on DB shutdown
  - -T7752: Enables asynchronous loads of Query Store
- [Configuration recommended by Microsoft](#)

# CONSIDERATIONS

- Aggregated data (interval)
- CPU additional load ~3-5%
- PerfMon
  - Query Store CPU usage
  - Query Store logical reads, writes
  - Query Store physical reads (ASYNC writer)
- Database datafile distribution
  - PRIMARY (For internal objects and Query Store) MAIN (User objects)
- Persisted after backup \ restore

- For more information

[Best practices according to Microsoft](#)

[How Query Store Collects data](#)

[Monitoring performance by using Query store](#)

[Query Store Usage Scenarios](#)

# RESOURCES

- DBA Master - Query Store checker - Coming soon ...
- [Study your data - SSMS Instance and DB level SSMS reports](#)

**THANK YOU!!**