



Introduction to the Query Store



What does this session cover?

Introduction to the Query Store

Query Store Settings

Troubleshooting with Query Store Reports

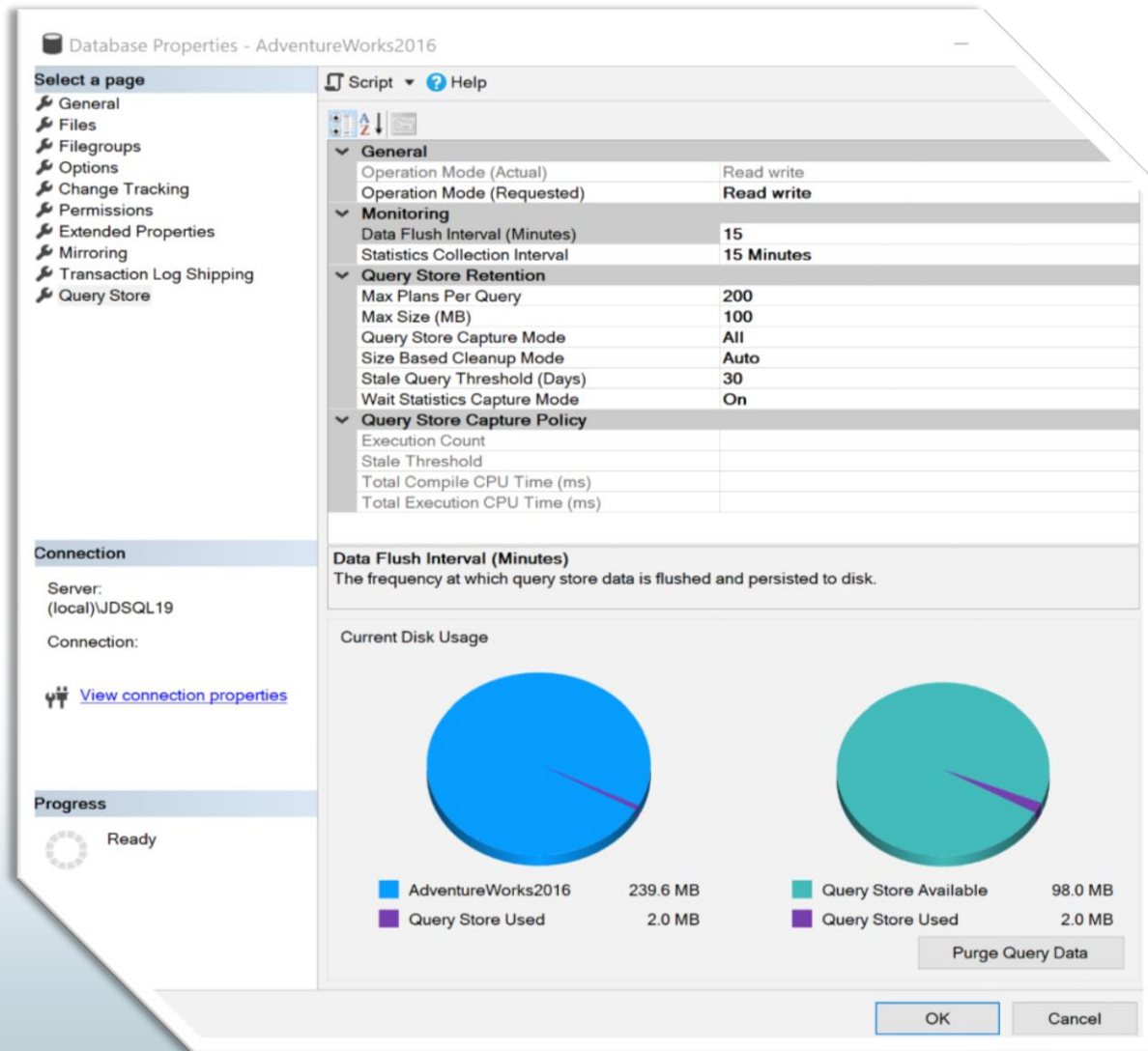
Query Store Catalog Views



Introduction to the Query Store



Introducing the Query Store



Query Store is set at the database level

Cannot be used for Master or TempDB system databases but can be enabled for the Model and MSDB system databases.

The user database stores the data in internal tables that can be accessed by using built-in Query Store views.

SQL Server retains this data until the space allocated to Query Store is full or manually purged.

Why use Query Store?

Before Query Store

- Requires manual proactive monitoring to identify execution plan problems.
- Only the latest plan was stored in the procedure cache
- Restart caused data to be lost
- Frequent recompiles of procedures or use of DBCC FREEPROCACHE
- No history or aggregated gathering of data available.

With Query Store

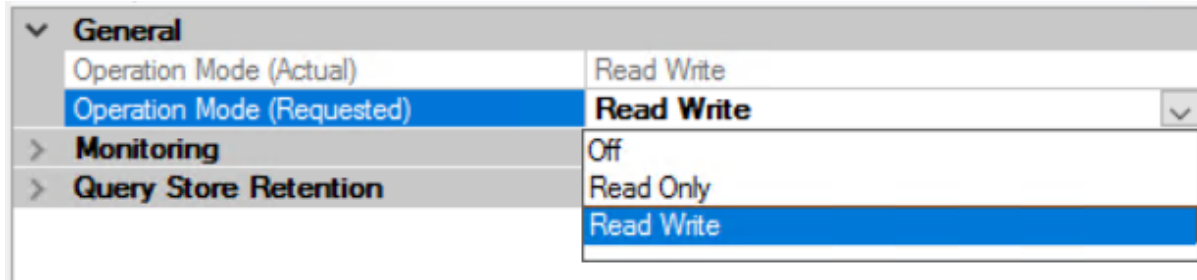
- It stores the history of the execution plans for each query
- It establishes a performance baseline for each plan over time
- It identifies queries that may have regressed
- It is possible to force plans quickly and easily
- It works across server restarts, upgrades, and query recompilation

Query Store Settings



Query Store Operation Modes

Operation Mode can be set under database properties



Operation Mode can be enabled two ways using T-SQL. If only using the ON option, the Mode defaults to **Read_Write**

```
ALTER DATABASE [AdventureWorks2016] SET QUERY_STORE = ON;
```

```
ALTER DATABASE [AdventureWorks2016] SET QUERY_STORE  
(OPERATION_MODE = READ_WRITE);
```

Query Store Monitoring Settings

Data Flush Interval determines the frequency at which data written to the query store is persisted to disk.
(Default is **15 Minutes**).

Monitoring	
Data Flush Interval (Minutes)	15
Statistics Collection Interval	1 Hour

```
ALTER DATABASE [AdventureWorks2016] SET QUERY_STORE  
(INTERVAL_LENGTH_MINUTES = 1,  
DATA_FLUSH_INTERVAL_SECONDS = 60)
```


Query Store Monitoring Settings

Statistics Collection Interval determines the time interval at which runtime execution statistics data is aggregated into the query store. Only the values of 1, 5, 10, 15, 60, and 1440 minutes is allowed. (Default is **60**).

Monitoring	
Data Flush Interval (Minutes)	15
Statistics Collection Interval	1 Hour

```
ALTER DATABASE [AdventureWorks2016] SET QUERY_STORE  
(INTERVAL_LENGTH_MINUTES = 1,  
DATA_FLUSH_INTERVAL_SECONDS = 60)
```

Query Store Retention Settings

Max Plans Per Query is a new retention setting introduced in SQL Server 2017 and is an integer representing the maximum number of plans maintained for each query. (Default is **200**).

▼ Query Store Retention	
Max Plans Per Query	200
Max Size (MB)	100
Query Store Capture Mode	Custom
Size Based Cleanup Mode	Auto
Stale Query Threshold (Days)	30
Wait Statistics Capture Mode	On

```
ALTER DATABASE AdventureWorks2016 SET QUERY_STORE  
(MAX_PLANS_PER_QUERY = 200,  
MAX_STORAGE_SIZE_MB = 100,  
QUERY_CAPTURE_MODE = AUTO,  
SIZE_BASED_CLEANUP_MODE = AUTO,  
CLEANUP_POLICY = (STALE_QUERY_THRESHOLD_DAYS = 367),  
WAIT_STATS_CAPTURE_MODE = ON);  
GO
```

Query Store Retention Settings

Max Size (MB) configures the maximum storage size for the query store. (Default is **100MB**) When the query store limit is reached, query store changes the state from read-write to read-only.

▼ Query Store Retention	
Max Plans Per Query	200
Max Size (MB)	100
Query Store Capture Mode	Custom
Size Based Cleanup Mode	Auto
Stale Query Threshold (Days)	30
Wait Statistics Capture Mode	On

```
ALTER DATABASE AdventureWorks2016 SET QUERY_STORE
(MAX_PLANS_PER_QUERY = 200,
MAX_STORAGE_SIZE_MB = 100,
QUERY_CAPTURE_MODE = AUTO,
SIZE_BASED_CLEANUP_MODE = AUTO,
CLEANUP_POLICY = (STALE_QUERY_THRESHOLD_DAYS = 367),
WAIT_STATS_CAPTURE_MODE = ON);
GO
```

Query Store Retention Settings

Query Store Capture Mode determines to capture all the queries (Default is **ALL**), or relevant queries based on execution count and resource consumption (**AUTO**) or stop capturing queries (**NONE**). SQL Server 2019 introduces an additional (**CUSTOM**) setting.

▼ Query Store Retention	
Max Plans Per Query	200
Max Size (MB)	100
Query Store Capture Mode	Custom
Size Based Cleanup Mode	Auto
Stale Query Threshold (Days)	30
Wait Statistics Capture Mode	On

```
ALTER DATABASE AdventureWorks2016 SET QUERY_STORE
(MAX_PLANS_PER_QUERY = 200,
MAX_STORAGE_SIZE_MB = 100,
QUERY_CAPTURE_MODE = AUTO,
SIZE_BASED_CLEANUP_MODE = AUTO,
CLEANUP_POLICY = (STALE_QUERY_THRESHOLD_DAYS = 367),
WAIT_STATS_CAPTURE_MODE = ON);
GO
```

Query Store Retention Settings

Size Based Cleanup Mode determines whether the cleanup process will be automatically activated when the total amount of data gets close to the maximum size. (Default is **Auto**).

▼ Query Store Retention	
Max Plans Per Query	200
Max Size (MB)	100
Query Store Capture Mode	Custom
Size Based Cleanup Mode	Auto
Stale Query Threshold (Days)	30
Wait Statistics Capture Mode	On

```
ALTER DATABASE AdventureWorks2016 SET QUERY_STORE
(MAX_PLANS_PER_QUERY = 200,
MAX_STORAGE_SIZE_MB = 100,
QUERY_CAPTURE_MODE = AUTO,
SIZE_BASED_CLEANUP_MODE = AUTO,
CLEANUP_POLICY = (STALE_QUERY_THRESHOLD_DAYS = 367),
WAIT_STATS_CAPTURE_MODE = ON);
GO
```

Query Store Retention Settings

Stale Query Threshold (Days) determines the number of days to retain data in the query store. (Default is **30 days** and Maximum is **367 days**).

▼ Query Store Retention	
Max Plans Per Query	200
Max Size (MB)	100
Query Store Capture Mode	Custom
Size Based Cleanup Mode	Auto
Stale Query Threshold (Days)	30
Wait Statistics Capture Mode	On

```
ALTER DATABASE AdventureWorks2016 SET QUERY_STORE  
(MAX_PLANS_PER_QUERY = 200,  
MAX_STORAGE_SIZE_MB = 100,  
QUERY_CAPTURE_MODE = AUTO,  
SIZE_BASED_CLEANUP_MODE = AUTO,  
CLEANUP_POLICY = (STALE_QUERY_THRESHOLD_DAYS = 367),  
WAIT_STATS_CAPTURE_MODE = ON);  
GO
```

Query Store Retention Settings

Wait Statistics Capture Mode is a new retention setting introduced in SQL Server 2017 that controls if Query Store captures wait statistics information.
(Default = **ON**).

▼ Query Store Retention	
Max Plans Per Query	200
Max Size (MB)	100
Query Store Capture Mode	Custom
Size Based Cleanup Mode	Auto
Stale Query Threshold (Days)	30
Wait Statistics Capture Mode	On

```
ALTER DATABASE AdventureWorks2016 SET QUERY_STORE  
(MAX_PLANS_PER_QUERY = 200,  
MAX_STORAGE_SIZE_MB = 100,  
QUERY_CAPTURE_MODE = AUTO,  
SIZE_BASED_CLEANUP_MODE = AUTO,  
CLEANUP_POLICY = (STALE_QUERY_THRESHOLD_DAYS = 367),  
WAIT_STATS_CAPTURE_MODE = ON);  
GO
```

Query Store Capture Policy Settings

Introduced in SQL Server 2019 and available if the Query Store Capture Mode setting has been set to **CUSTOM**.

The value for the **EXECUTION COUNT** is the value a query must exceed within the Stale Threshold time period to be captured by the Query Store.

▼ Query Store Capture Policy	
Execution Count	30
Stale Threshold	1 Hour
Total Compile CPU Time (ms)	1000
Total Execution CPU Time (ms)	100

```
ALTER DATABASE AdventureWorks2016 SET QUERY_STORE
(QUERY_CAPTURE_POLICY =
  (EXECUTION_COUNT = 100,
   STALE_CAPTURE_POLICY_THRESHOLD = 24 Hours,
   TOTAL_COMPILE_CPU_TIME_MS = 2000,
   TOTAL_EXECUTION_CPU_TIME_MS = 1000));
GO
```


Query Store Capture Policy Settings

Introduced in SQL Server 2019 and available if the Query Store Capture Mode setting has been set to **CUSTOM**.

The value for the **Stale Threshold** can be from 1 hour up to 7 days. This setting specifies the time given to exceed the values of the three other settings for a query to be captured.

▼ Query Store Capture Policy	
Execution Count	30
Stale Threshold	1 Hour
Total Compile CPU Time (ms)	1000
Total Execution CPU Time (ms)	100

```
ALTER DATABASE AdventureWorks2016 SET QUERY_STORE
(QUERY_CAPTURE_POLICY =
(EXECUTION_COUNT = 100,
STALE_CAPTURE_POLICY_THRESHOLD = 24 Hours,
TOTAL_COMPILE_CPU_TIME_MS = 2000,
TOTAL_EXECUTION_CPU_TIME_MS = 1000));
GO
```

Query Store Capture Policy Settings

Introduced in SQL Server 2019 and available if the Query Store Capture Mode setting has been set to **CUSTOM**.

The value for the **Total Compile CPU Time (ms)** is the value in milliseconds that a query must exceed within the **Stale Threshold** time period to be captured by the Query Store.

▼ Query Store Capture Policy	
Execution Count	30
Stale Threshold	1 Hour
Total Compile CPU Time (ms)	1000
Total Execution CPU Time (ms)	100

```
ALTER DATABASE AdventureWorks2016 SET QUERY_STORE
(QUERY_CAPTURE_POLICY =
(EXECUTION_COUNT = 100,
STALE_CAPTURE_POLICY_THRESHOLD = 24 Hours,
TOTAL_COMPILE_CPU_TIME_MS = 2000,
TOTAL_EXECUTION_CPU_TIME_MS = 1000));
GO
```

Query Store Capture Policy Settings

Introduced in SQL Server 2019 and available if the Query Store Capture Mode setting has been set to **CUSTOM**.

The value for the **Total Execution CPU Time (ms)** is the value in milliseconds that a query must exceed within the **Stale Threshold** time period to be captured by the Query Store.

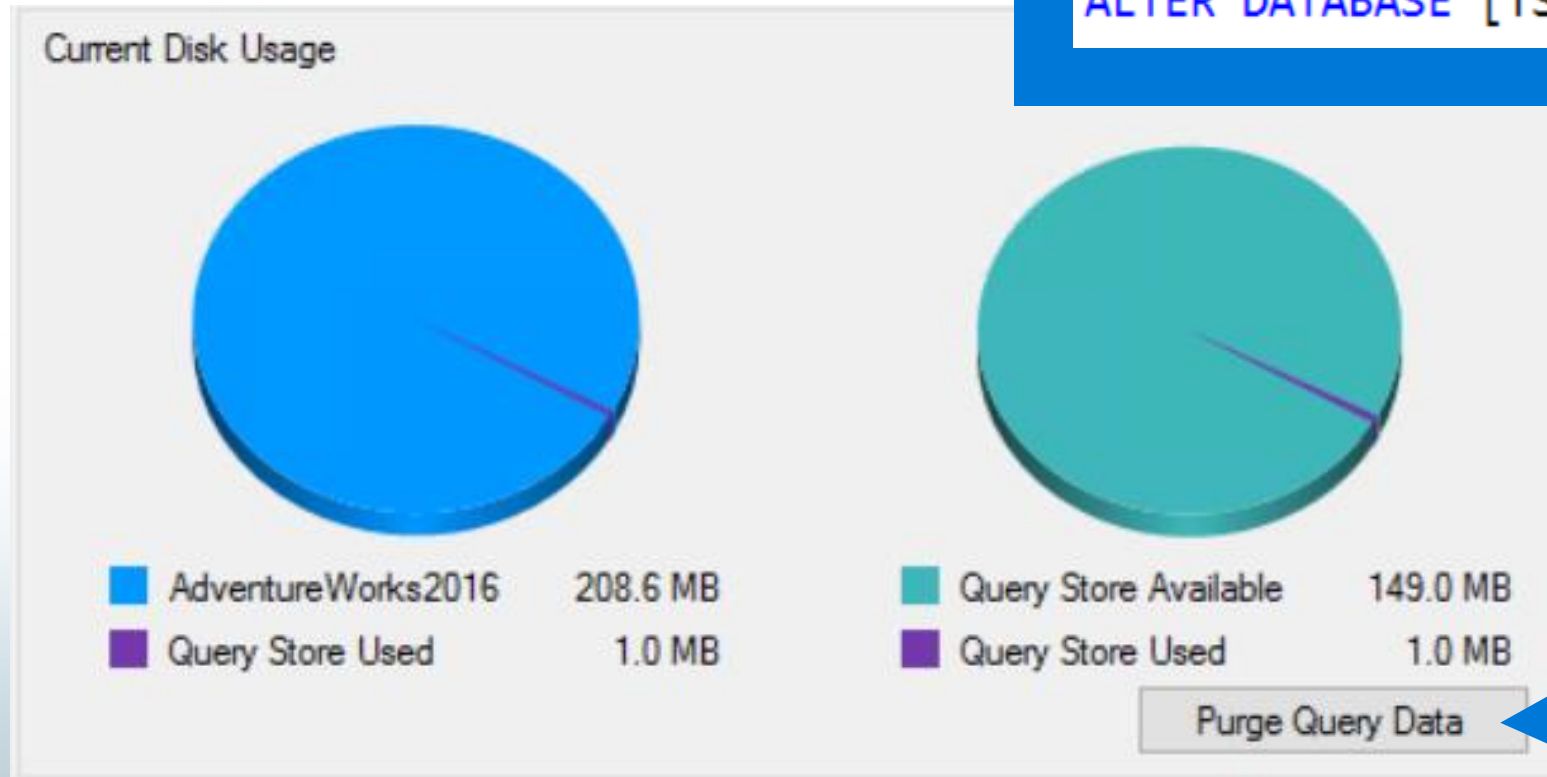
▼ Query Store Capture Policy	
Execution Count	30
Stale Threshold	1 Hour
Total Compile CPU Time (ms)	1000
Total Execution CPU Time (ms)	100

```
ALTER DATABASE AdventureWorks2016 SET QUERY_STORE
(QUERY_CAPTURE_POLICY =
(EXECUTION_COUNT = 100,
STALE_CAPTURE_POLICY_THRESHOLD = 24 Hours,
TOTAL_COMPILE_CPU_TIME_MS = 2000,
TOTAL_EXECUTION_CPU_TIME_MS = 1000));
GO
```

Purge Query Data

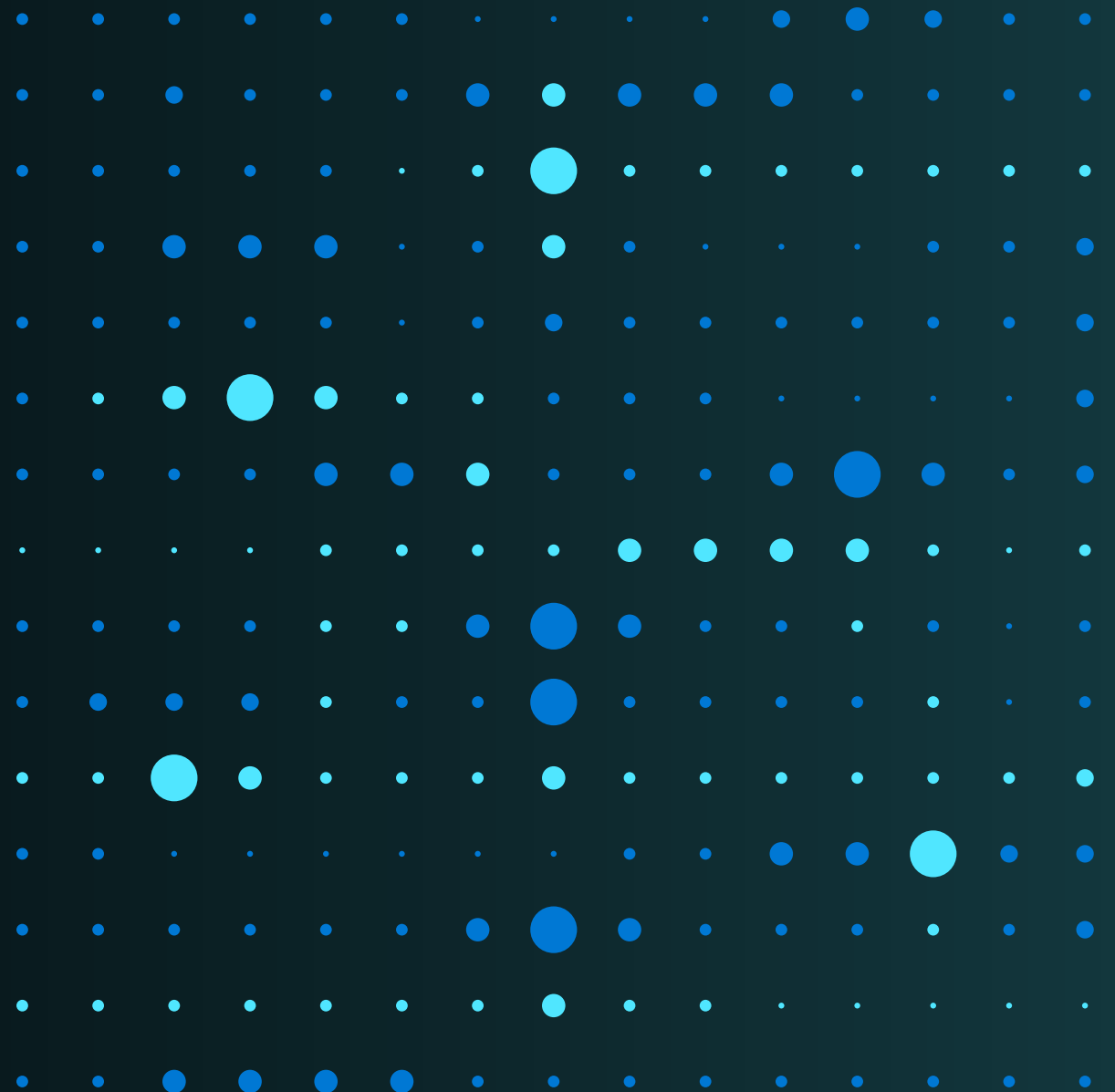
Data can be manually purged from the Query Store.

```
ALTER DATABASE [TSQL] SET QUERY_STORE CLEAR;
```



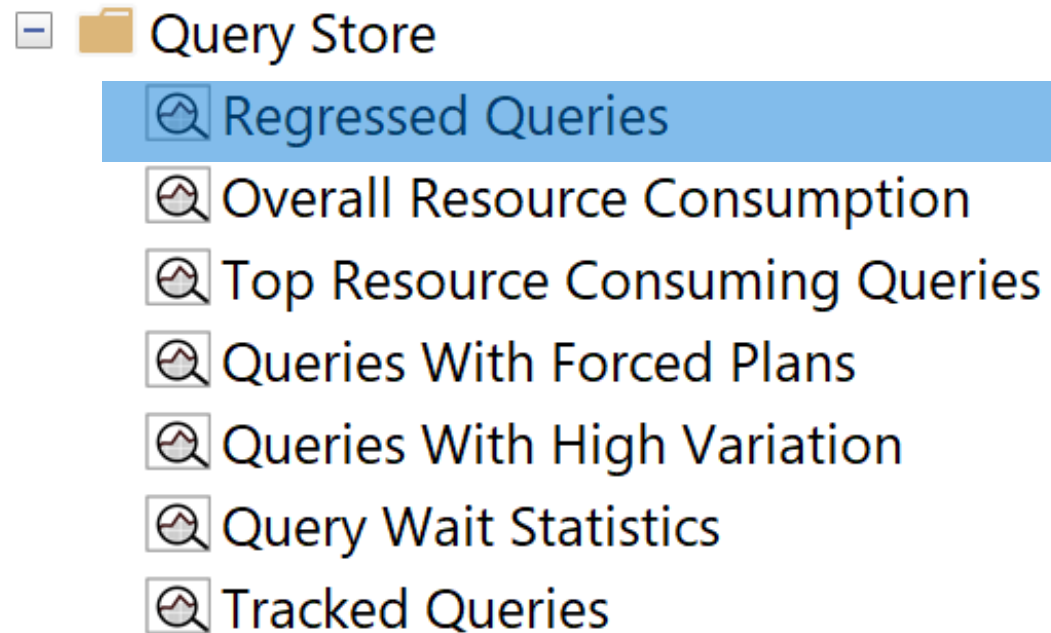
Demonstration

Troubleshooting with Query Store Reports



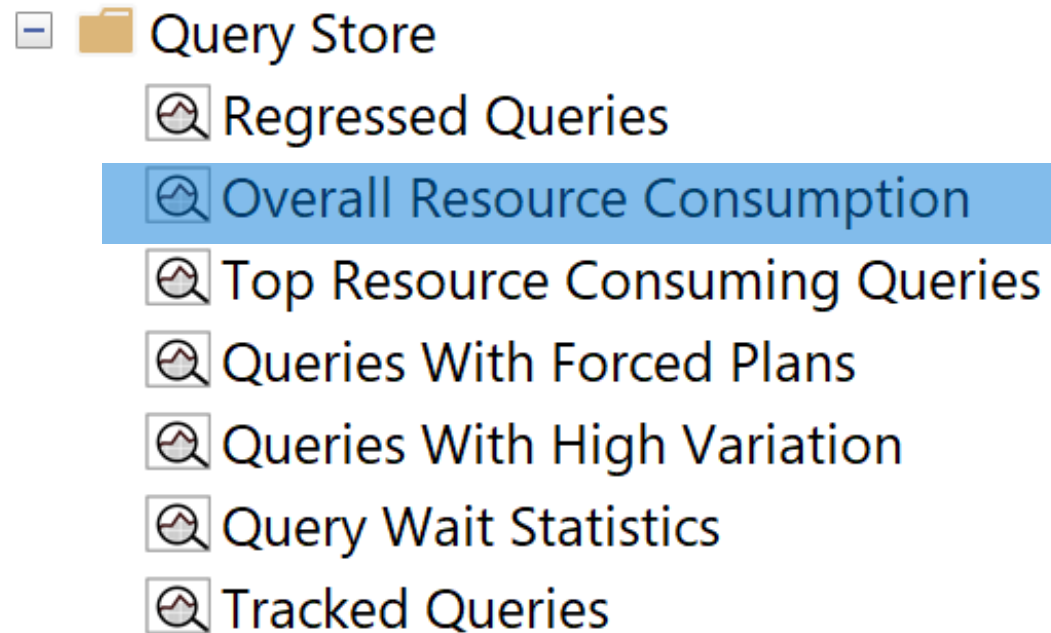
Query Store Reports

Regressed Queries: Use this dashboard to review queries that might have regressed because of execution plan changes



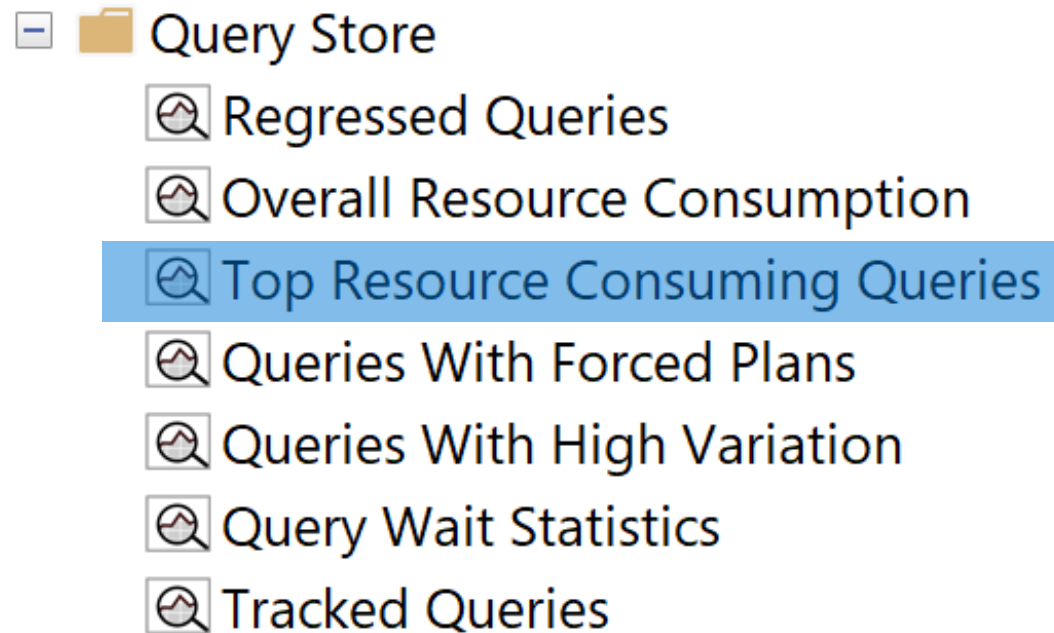
Query Store Reports

Overall Resource Consumption: Use this dashboard to visualize overall resource consumption during the last month in four charts: duration, execution count, CPU time, and logical reads



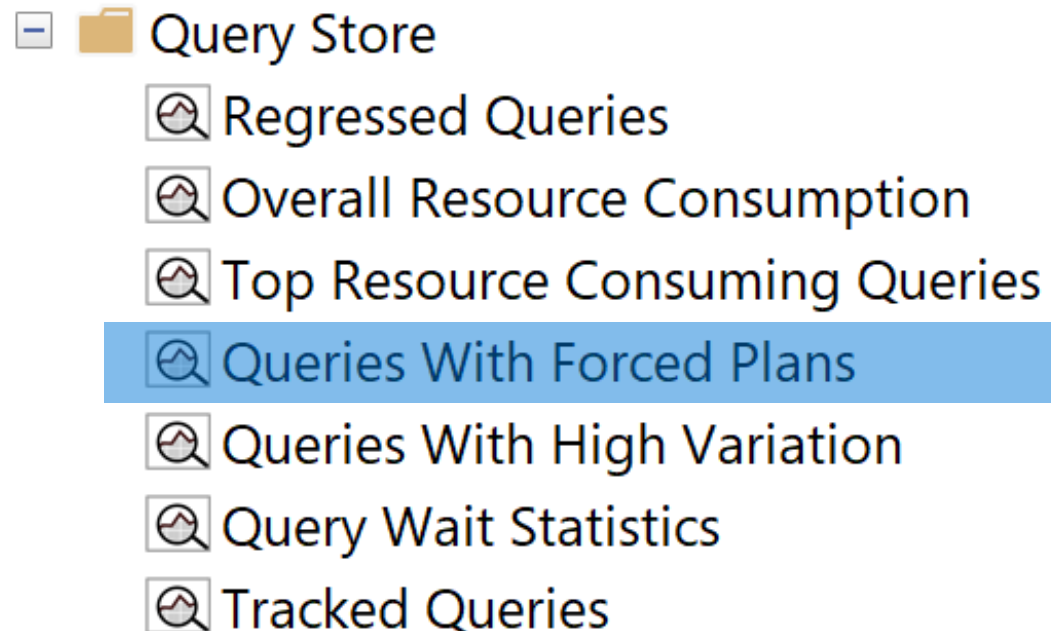
Query Store Reports

Top Resource Consuming Queries: Use this dashboard to review queries in the set of top 25 resource consumers during the last hour



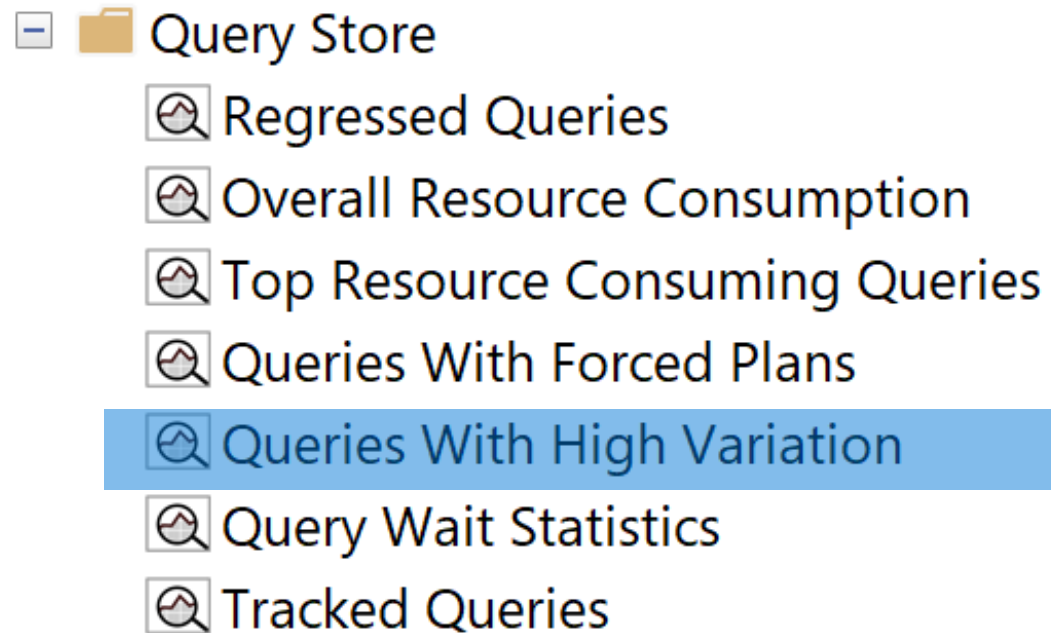
Query Store Reports

Queries With Forced Plans: Used to isolate queries that have been given a forced plan. Requires SQL Server 2016 SP1 or later.



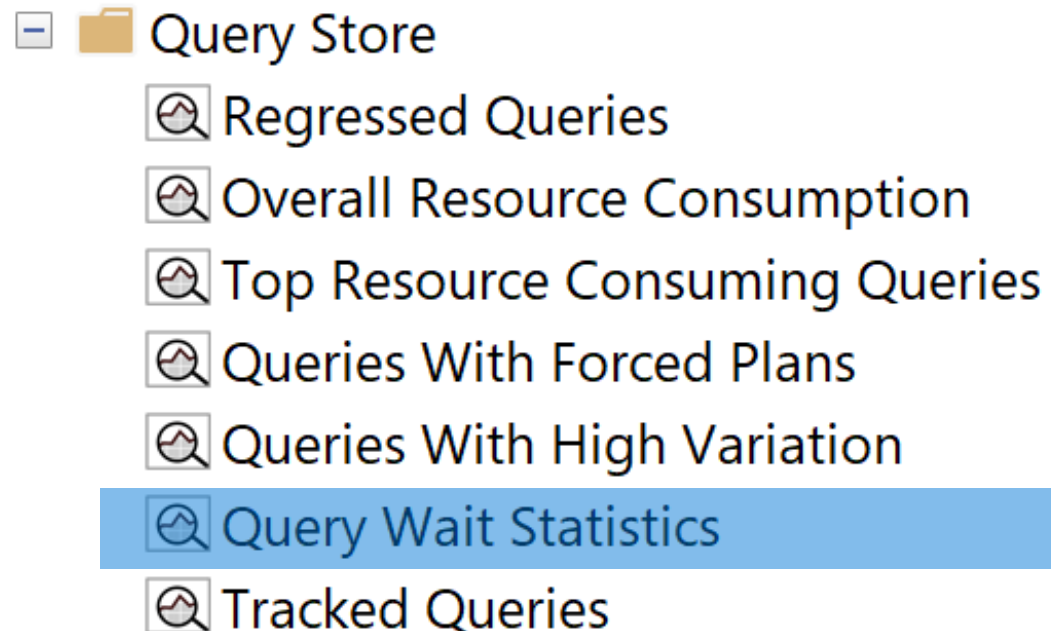
Query Store Reports

Queries With High Variation: Used to locate queries with high variation in query execution. Useful to locate queries with parameterization problems. Requires SQL Server 2016 SP1 or later.



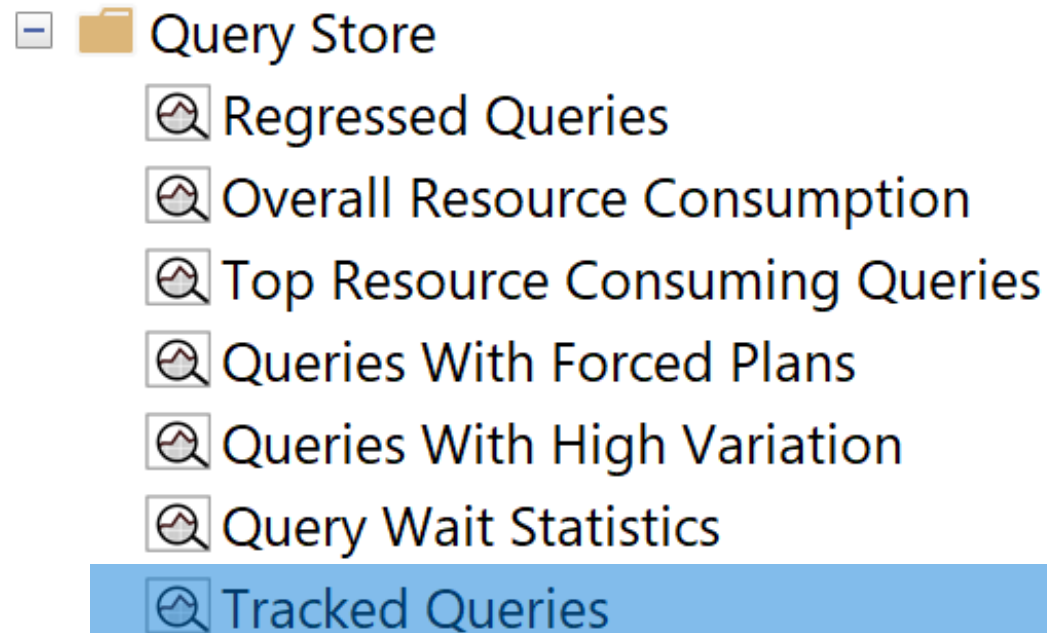
Query Store Reports

Query Wait Statistics shows a bar chart containing the top wait categories in the Query Store. Use the drop down at the top to select an aggregate criteria for the wait time: avg, max, min, std dev, and **total** (default). Requires SQL Server 2017.

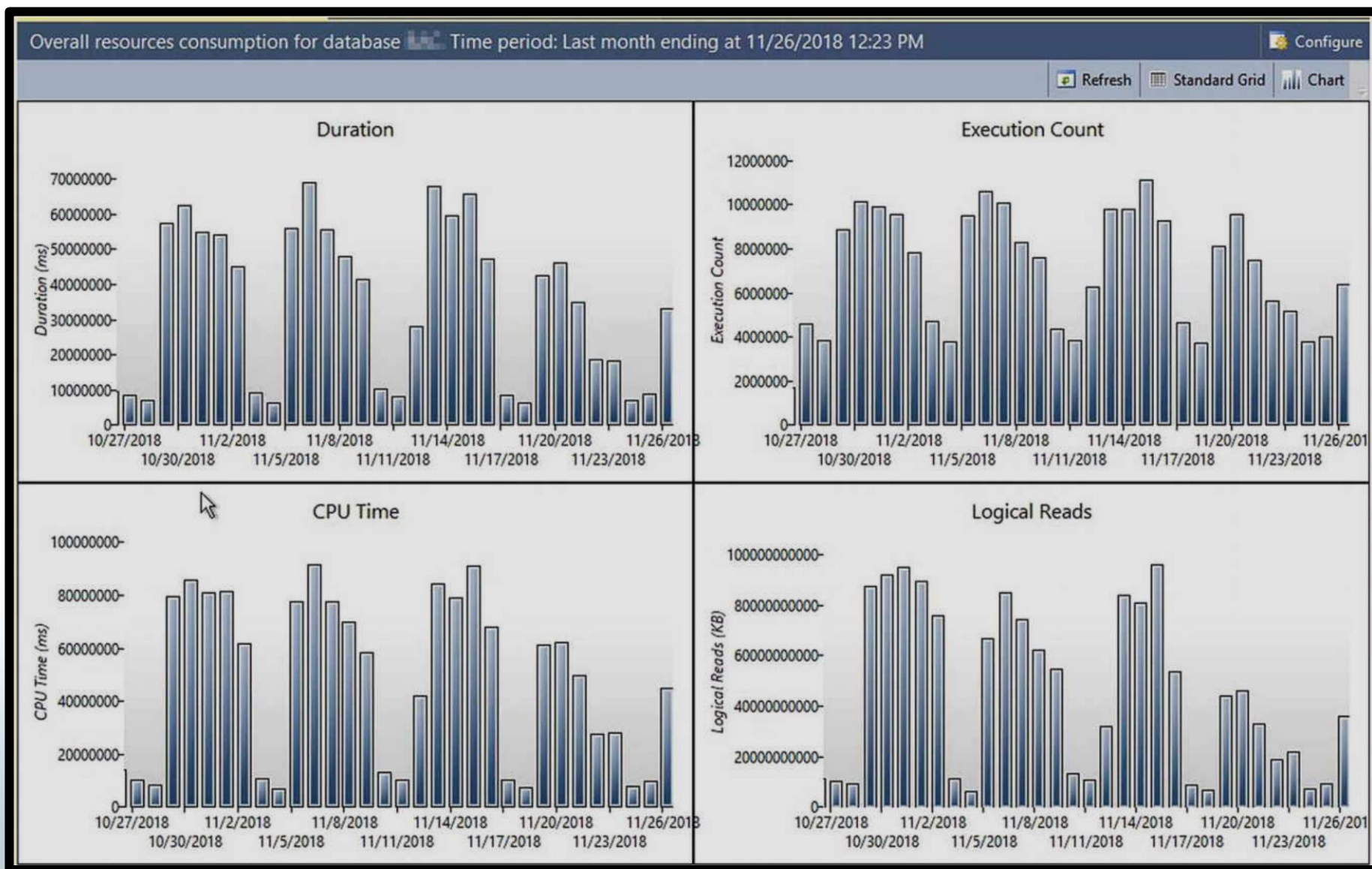


Query Store Reports

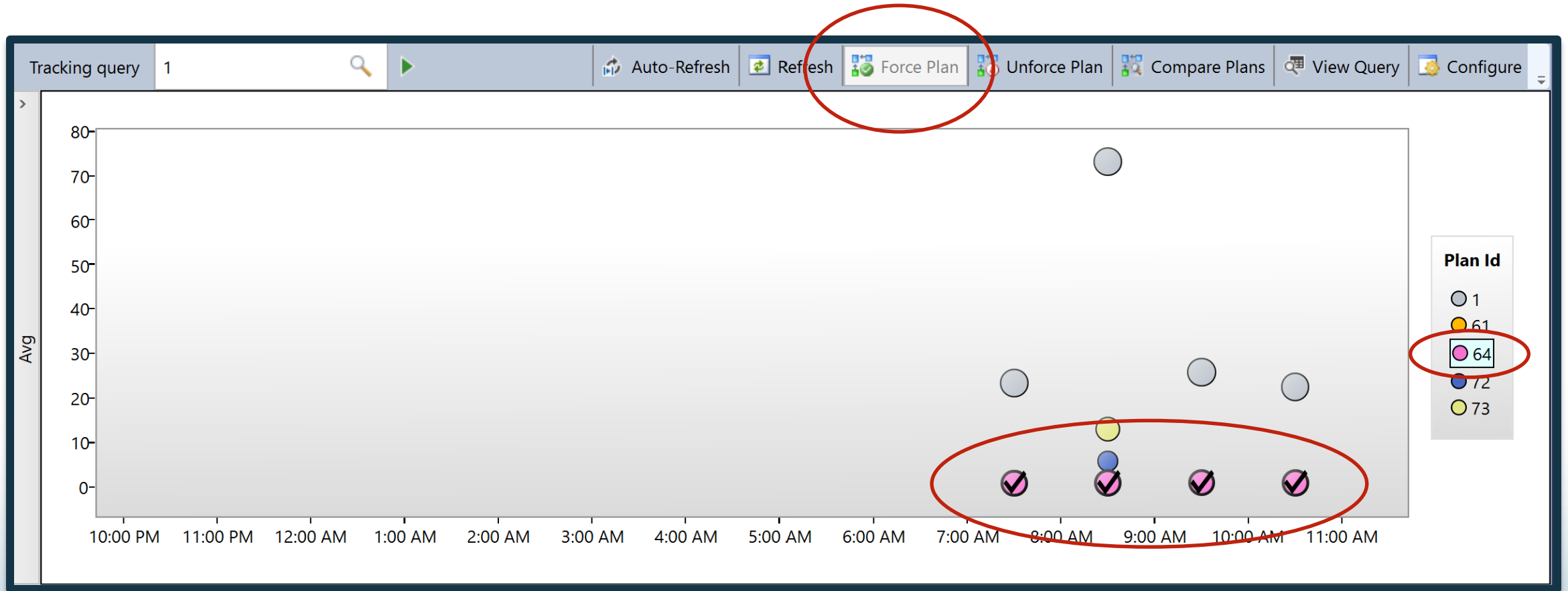
Tracked Queries: Use this dashboard to monitor the execution plans and regression of a specify query



Establishing a Baseline



Force Plan



Plan Compare

Showplan Comparison [AdventureWorks2016]

Plan 64
SELECT ProductID, OrderQty, UnitPrice FROM Sales.SalesOrderData...

Index Seek (NonClustered)
[SalesOrderDet...]
Cost: 100 %

Plan 73
SELECT ProductID, OrderQty, UnitPrice FROM Sales.SalesOrderData...

Nested... (Inner...)
Cost: 0 %

Index Seek (NonClustered)
[SalesOrderDet...]
Cost: 0 %

Key Lookup (Clustered)
[SalesOrderDet...]
Cost: 99 %

Properties

Top Plan
Index Seek (NonClustered)

Bottom Plan
Key Lookup (Clustered)

Property	Top Plan	Bottom Plan
Defined Values	[AdventureWorks2016]	[AdventureWorks2016]
Description	Scan a particular range	Uses a supplied cluster
Estimated CPU Cost	0.0053138	0.0001581
Estimated Execution Row		
Estimated I/O Cost	0.0144923	0.003125
Estimated Number of Rows	1	242
Estimated Number of Rows	4688	1
Estimated Number of Rows	4688	
Estimated Operator Cost	0.0198061 (100%)	0.722865 (99%)
Estimated Rebinds	0	241
Estimated Rewinds	0	0
Estimated Row Size	21 B	17 B
Estimated Subtree Cost	0.0198061	0.722865
Forced Index	False	False
ForceScan	False	False
ForceSeek	False	False
Logical Operation	Index Seek	Key Lookup
Lookup	True	True
Node ID	0	4
NoExpandHint	False	False
Object	[AdventureWorks2016]	[AdventureWorks2016]
Ordered	True	True
Output List	[AdventureWorks2016]	[AdventureWorks2016]
Parallel	False	False
Physical Operation	Index Seek	Key Lookup
Scan Direction	FORWARD	FORWARD
Seek Predicates	Seek Keys[1]: Prefix [Seek Keys[1]: Prefix [
Storage	RowStore	RowStore
TableCardinality	121317	121317

Questions?

Query Store Catalog Views



Query Store contains three stores

Plan Store

- Persists execution plan information
- Stores Query Text and Query Plan information
- You can limit the number of unique plans by using the `max_plans_per_query` configuration.

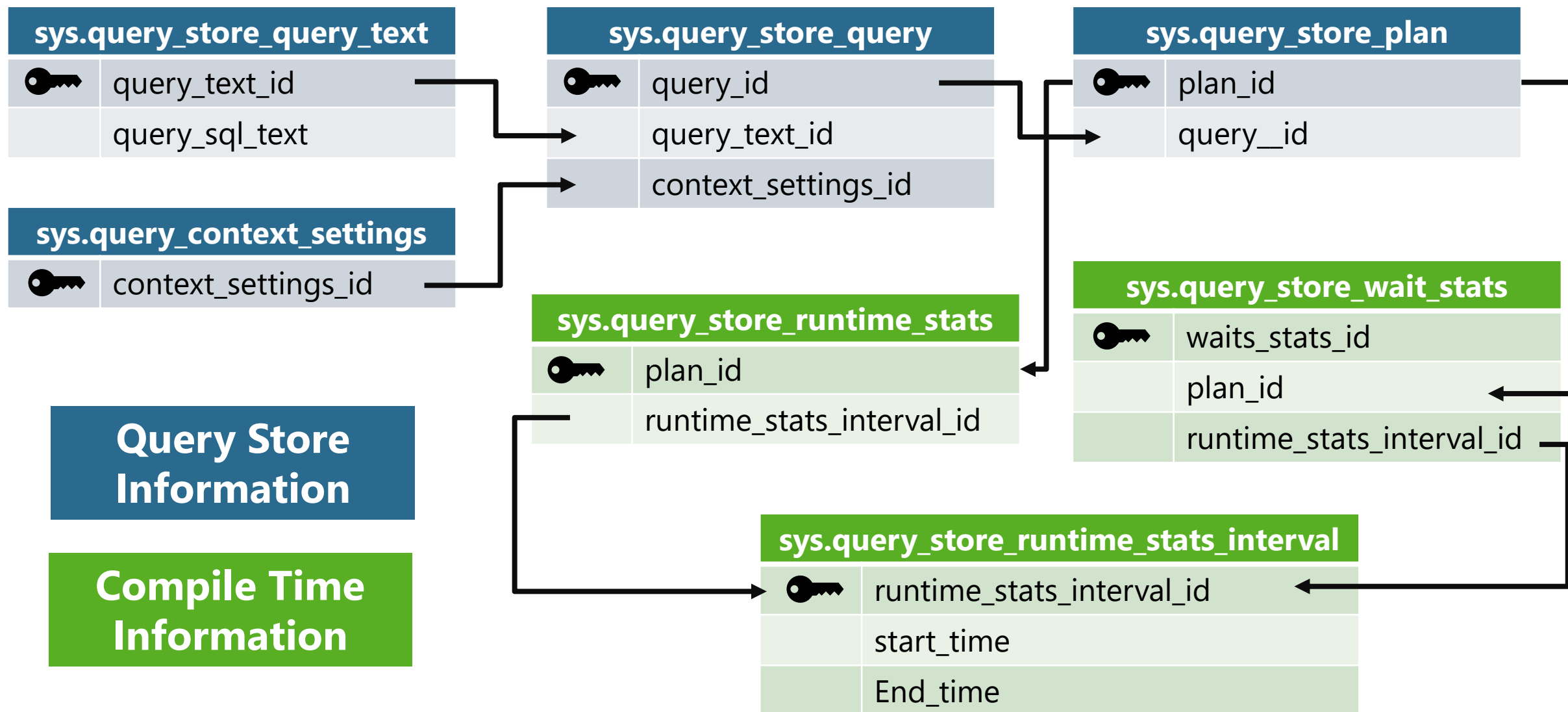
Runtime Stats Store

- Persists execution statistics information
- Captures Compile Time, Duration, CPU usage, Writes, and both Logical and Physical Reads
- To minimize space usage, the runtime stats are aggregated over a fixed time window.

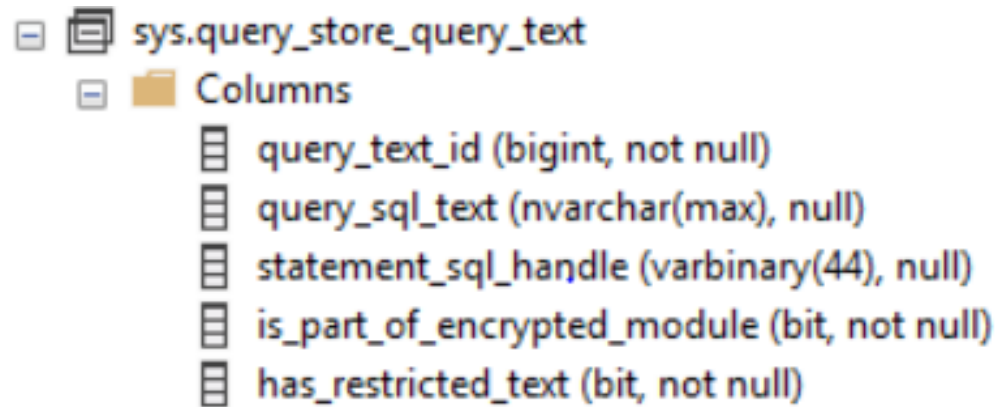
Wait Stat Store

- Persists wait statistics information
- Introduced in SQL Server 2017 and Azure SQL Database
- Organizes wait stats into different wait categories to simplify troubleshooting similar wait types.

Query Store Catalog Views



sys.query_store_query_text



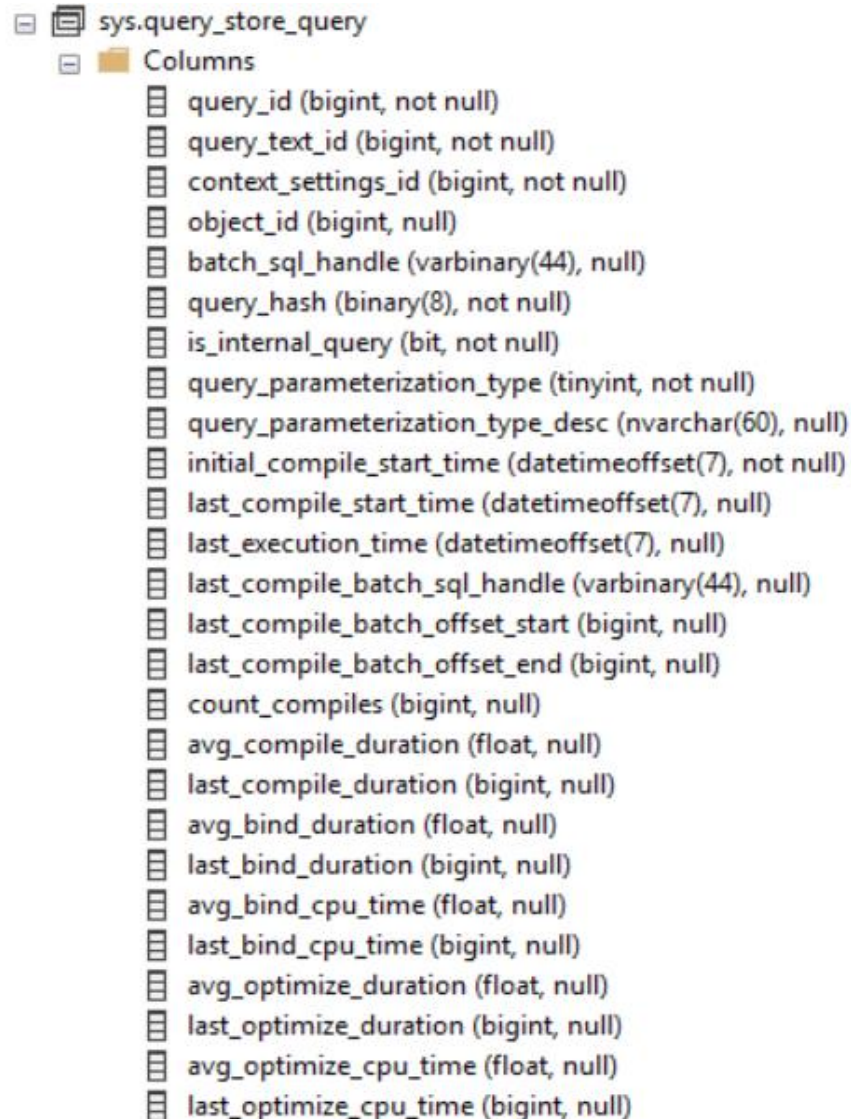
The screenshot shows the SQL Server Enterprise Manager interface. Under the 'sys' folder, the 'query_store_query_text' table is selected. The 'Columns' folder is expanded, showing the following columns:

Column Name	Data Type	Nullable
query_text_id	bigint	not null
query_sql_text	nvarchar(max)	null
statement_sql_handle	varbinary(44)	null
is_part_of_encrypted_module	bit	not null
has_restricted_text	bit	not null

Presents the unique query text that is executed against the database.

Every statement in a batch generates a separate query text entry.

sys.query_store_query



The screenshot shows the SQL Server Enterprise Manager interface. On the left, a tree view displays the 'sys.query_store_query' table under the 'Columns' folder. The main pane on the right lists the columns of this table, each with its data type and nullability. The columns are: query_id (bigint, not null), query_text_id (bigint, not null), context_settings_id (bigint, not null), object_id (bigint, null), batch_sql_handle (varbinary(44), null), query_hash (binary(8), not null), is_internal_query (bit, not null), query_parameterization_type (tinyint, not null), query_parameterization_type_desc (nvarchar(60), null), initial_compile_start_time (datetimeoffset(7), not null), last_compile_start_time (datetimeoffset(7), null), last_execution_time (datetimeoffset(7), null), last_compile_batch_sql_handle (varbinary(44), null), last_compile_batch_offset_start (bigint, null), last_compile_batch_offset_end (bigint, null), count_compiles (bigint, null), avg_compile_duration (float, null), last_compile_duration (bigint, null), avg_bind_duration (float, null), last_bind_duration (bigint, null), avg_bind_cpu_time (float, null), last_bind_cpu_time (bigint, null), avg_optimize_duration (float, null), last_optimize_duration (bigint, null), avg_optimize_cpu_time (float, null), and last_optimize_cpu_time (bigint, null).

Column Name	Data Type	Nullability
query_id	bigint	not null
query_text_id	bigint	not null
context_settings_id	bigint	not null
object_id	bigint	null
batch_sql_handle	varbinary(44)	null
query_hash	binary(8)	not null
is_internal_query	bit	not null
query_parameterization_type	tinyint	not null
query_parameterization_type_desc	nvarchar(60)	null
initial_compile_start_time	datetimeoffset(7)	not null
last_compile_start_time	datetimeoffset(7)	null
last_execution_time	datetimeoffset(7)	null
last_compile_batch_sql_handle	varbinary(44)	null
last_compile_batch_offset_start	bigint	null
last_compile_batch_offset_end	bigint	null
count_compiles	bigint	null
avg_compile_duration	float	null
last_compile_duration	bigint	null
avg_bind_duration	float	null
last_bind_duration	bigint	null
avg_bind_cpu_time	float	null
last_bind_cpu_time	bigint	null
avg_optimize_duration	float	null
last_optimize_duration	bigint	null
avg_optimize_cpu_time	float	null
last_optimize_cpu_time	bigint	null

Presents the query entries that are tracked and forced separately in the Query Store.

A single query text can produce multiple query entries that are executed under different context settings.

sys.query_store_plan

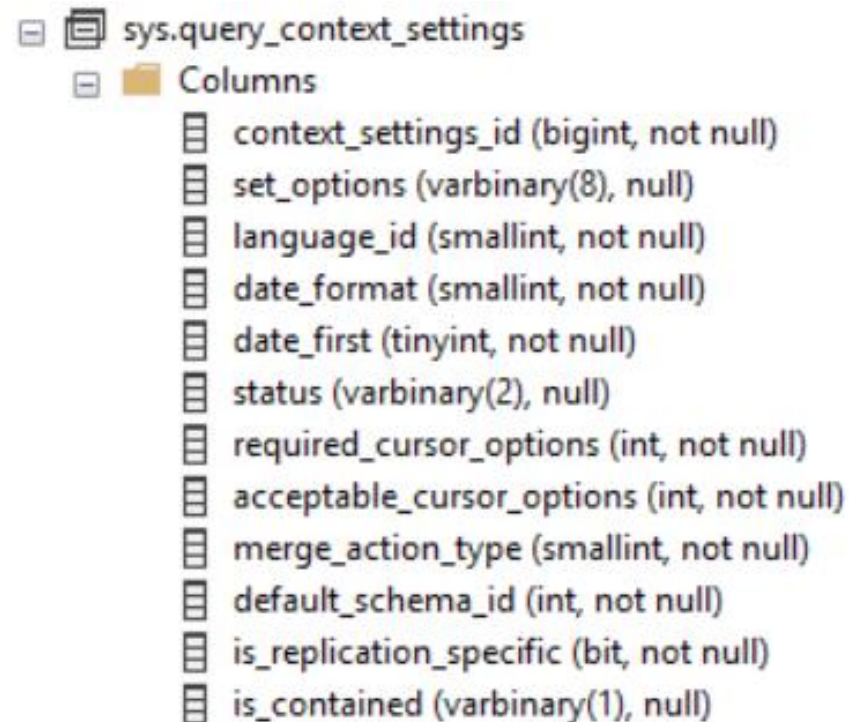


The screenshot shows the SQL Server Enterprise Manager interface. Under the 'Columns' folder, the columns of the sys.query_store_plan table are listed with their data types and nullability.

Column Name	Data Type	Nullability
plan_id	bigint	not null
query_id	bigint	not null
plan_group_id	bigint	null
engine_version	nvarchar(32)	null
compatibility_level	smallint	not null
query_plan_hash	binary(8)	not null
query_plan	nvarchar(max)	null
is_online_index_plan	bit	not null
is_trivial_plan	bit	not null
is_parallel_plan	bit	not null
is_forced_plan	bit	not null
is_natively_compiled	bit	not null
force_failure_count	bigint	not null
last_force_failure_reason	int	not null
last_force_failure_reason_desc	nvarchar(128)	null
count_compiles	bigint	null
initial_compile_start_time	datetimeoffset(7)	not null
last_compile_start_time	datetimeoffset(7)	null
last_execution_time	datetimeoffset(7)	null
avg_compile_duration	float	null
last_compile_duration	bigint	null
plan_forcing_type	int	not null
plan_forcing_type_desc	nvarchar(60)	null

Presents the estimated plan for the query with the compile time statistics.

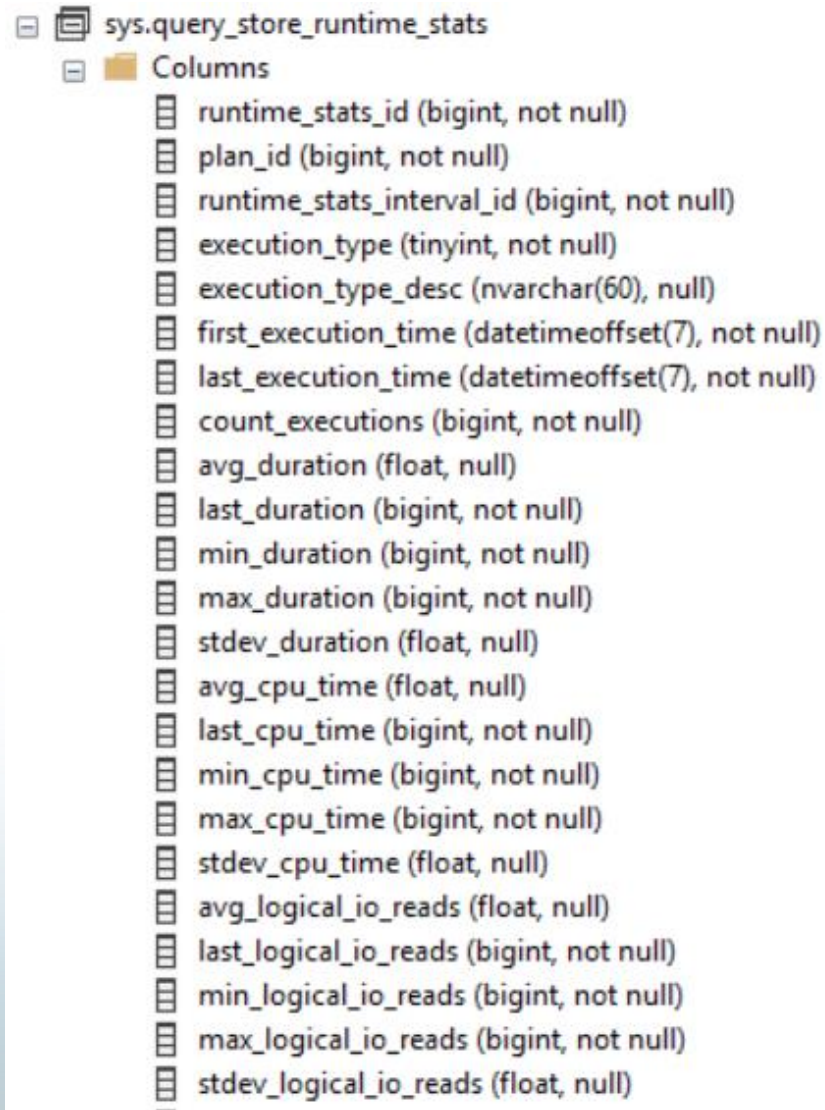
sys.query_context_settings

A screenshot of a database interface showing the structure of the sys.query_context_settings table. The table is expanded to show its columns. The columns are listed with their names and data types, including nullability.

sys.query_context_settings
Columns
context_settings_id (bigint, not null)
set_options (varbinary(8), null)
language_id (smallint, not null)
date_format (smallint, not null)
date_first (tinyint, not null)
status (varbinary(2), null)
required_cursor_options (int, not null)
acceptable_cursor_options (int, not null)
merge_action_type (smallint, not null)
default_schema_id (int, not null)
is_replication_specific (bit, not null)
is_contained (varbinary(1), null)

Presents the unique combinations of settings that affect a plan under which queries are executed.

sys.query_store_runtime_stats



The screenshot shows the 'sys.query_store_runtime_stats' table in SQL Server Enterprise Manager. The 'Columns' section is expanded, listing 22 columns with their data types and nullability. The columns are: runtime_stats_id (bigint, not null), plan_id (bigint, not null), runtime_stats_interval_id (bigint, not null), execution_type (tinyint, not null), execution_type_desc (nvarchar(60), null), first_execution_time (datetimeoffset(7), not null), last_execution_time (datetimeoffset(7), not null), count_executions (bigint, not null), avg_duration (float, null), last_duration (bigint, not null), min_duration (bigint, not null), max_duration (bigint, not null), stdev_duration (float, null), avg_cpu_time (float, null), last_cpu_time (bigint, not null), min_cpu_time (bigint, not null), max_cpu_time (bigint, not null), stdev_cpu_time (float, null), avg_logical_io_reads (float, null), last_logical_io_reads (bigint, not null), min_logical_io_reads (bigint, not null), max_logical_io_reads (bigint, not null), and stdev_logical_io_reads (float, null).

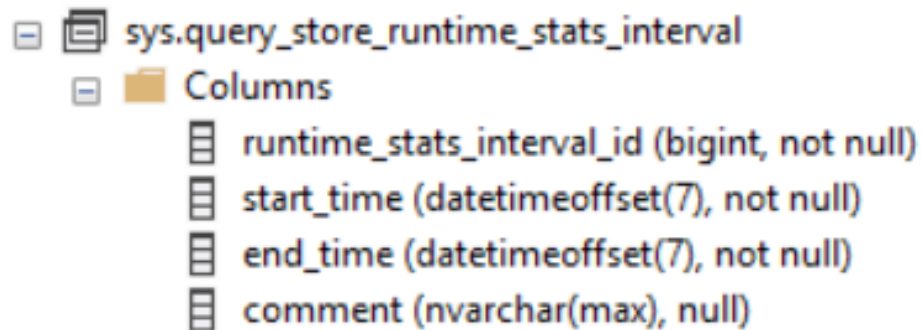
Column Name	Data Type	Nullability
runtime_stats_id	bigint	not null
plan_id	bigint	not null
runtime_stats_interval_id	bigint	not null
execution_type	tinyint	not null
execution_type_desc	nvarchar(60)	null
first_execution_time	datetimeoffset(7)	not null
last_execution_time	datetimeoffset(7)	not null
count_executions	bigint	not null
avg_duration	float	null
last_duration	bigint	not null
min_duration	bigint	not null
max_duration	bigint	not null
stdev_duration	float	null
avg_cpu_time	float	null
last_cpu_time	bigint	not null
min_cpu_time	bigint	not null
max_cpu_time	bigint	not null
stdev_cpu_time	float	null
avg_logical_io_reads	float	null
last_logical_io_reads	bigint	not null
min_logical_io_reads	bigint	not null
max_logical_io_reads	bigint	not null
stdev_logical_io_reads	float	null

Aggregated runtime statistics for executed plans.

Metrics are expressed in the form of Four statistic functions. (Average, Minimum, Maximum, and Standard Deviation).

sys.query_store_runtime_stats_interval

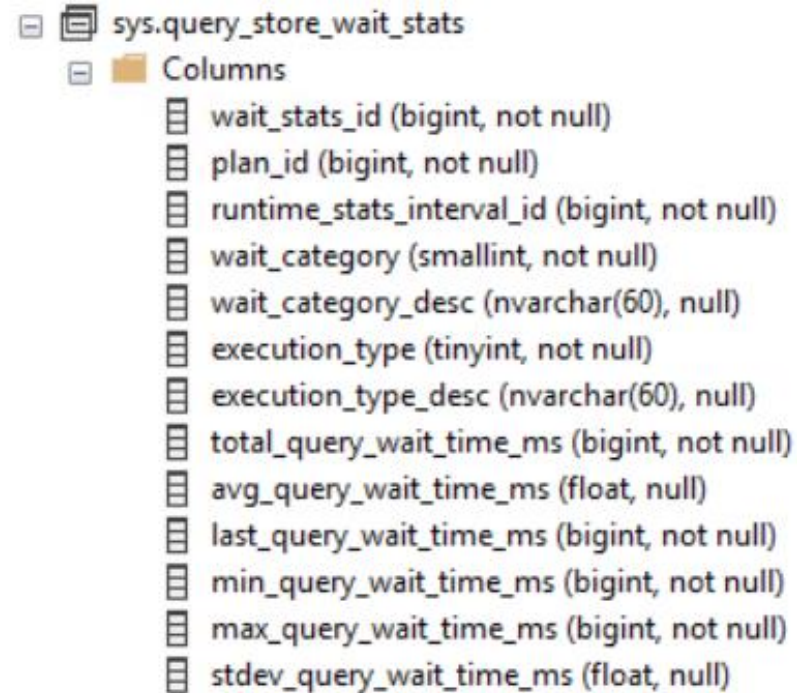
Query Store divides time into automatically generated time windows (intervals) and stores aggregated statistics on that interval for every executed plan.



The screenshot shows the SQL Server Enterprise Manager interface. On the left, a tree view displays the 'sys.query_store_runtime_stats_interval' table under the 'Query Store' folder. The table is expanded, showing its columns. The columns are: 'runtime_stats_interval_id' (bigint, not null), 'start_time' (datetimeoffset(7), not null), 'end_time' (datetimeoffset(7), not null), and 'comment' (nvarchar(max), null).

Column Name	Data Type	Nullability
runtime_stats_interval_id	bigint	not null
start_time	datetimeoffset(7)	not null
end_time	datetimeoffset(7)	not null
comment	nvarchar(max)	null

sys.query_store_plan_wait_stats



The screenshot shows the SQL Server Enterprise Manager interface. On the left, a tree view displays the 'sys.query_store_wait_stats' table under the 'Columns' folder. The main pane on the right lists the columns of this table with their data types and nullability.

Column Name	Data Type	Nullability
wait_stats_id	bigint	not null
plan_id	bigint	not null
runtime_stats_interval_id	bigint	not null
wait_category	smallint	not null
wait_category_desc	nvarchar(60)	null
execution_type	tinyint	not null
execution_type_desc	nvarchar(60)	null
total_query_wait_time_ms	bigint	not null
avg_query_wait_time_ms	float	null
last_query_wait_time_ms	bigint	not null
min_query_wait_time_ms	bigint	not null
max_query_wait_time_ms	bigint	not null
stdev_query_wait_time_ms	float	null

Presents wait category information for past runtime statistics interval.

Wait types are categorized and then wait time is aggregated across those wait categories.

Using Query Store Catalog Views

Finding the TOP 10 most frequently executed SQL Server Queries in the Query Store.

```
SELECT TOP 10 t.query_sql_text, q.query_id
FROM sys.query_store_query_text as t
JOIN sys.query_store_query as q
ON t.query_text_id = q.query_text_id
JOIN sys.query_store_plan as p
ON q.query_id = p.query_id
JOIN sys.query_store_runtime_stats as rs
ON p.plan_id = rs.plan_id
WHERE rs.count_executions > 1
GROUP BY t.query_sql_text, q.query_id
ORDER BY SUM(rs.count_executions)
```