

# SQLintersection

Session: 12/6, 11:30am - 12:30pm

## Gems to Help You Troubleshoot Query Performance

Pedro Lopes  
@SQLPedro



## Speaker: Pedro Lopes



- Senior Program Manager
- SQL Server Tiger team owns in-market and vNext of SQL Server
- Focused on SQL Server Relational Engine (Query Processor, Query Perf)
- 8+ years at Microsoft
- 16+ years with SQL Server

@SQLPedro

[pedro.lopes@microsoft.com](mailto:pedro.lopes@microsoft.com)

<http://aka.ms/sqlserverteam>



© SQLintersection. All rights reserved.  
<http://www.SQLintersection.com>

## Reminder: Intersect with Speakers and Attendees

- Tweet *tips and tricks* that you learn and follow tweets posted by your peers!
  - Follow: #SQLIntersection and/or #DEVIntersection
- Join us – Wednesday Evening – for SQLafterDark
  - Doors open at **7:00 pm**
  - Trivia game starts at **7:30 pm**
    - Winning team receives something fun!*
  - Raffle at the end of the night
    - Lots of great items to win including a seat in a five-day SQLskills Immersion Event!*
  - The first round of drinks is sponsored by SentryOne and SQLskills



© SQLIntersection. All rights reserved.  
<http://www.SQLIntersection.com>

## Overview

- A brief history of SQL Server query performance
- Diagnostics improvements
- Time permits: bonus improvement



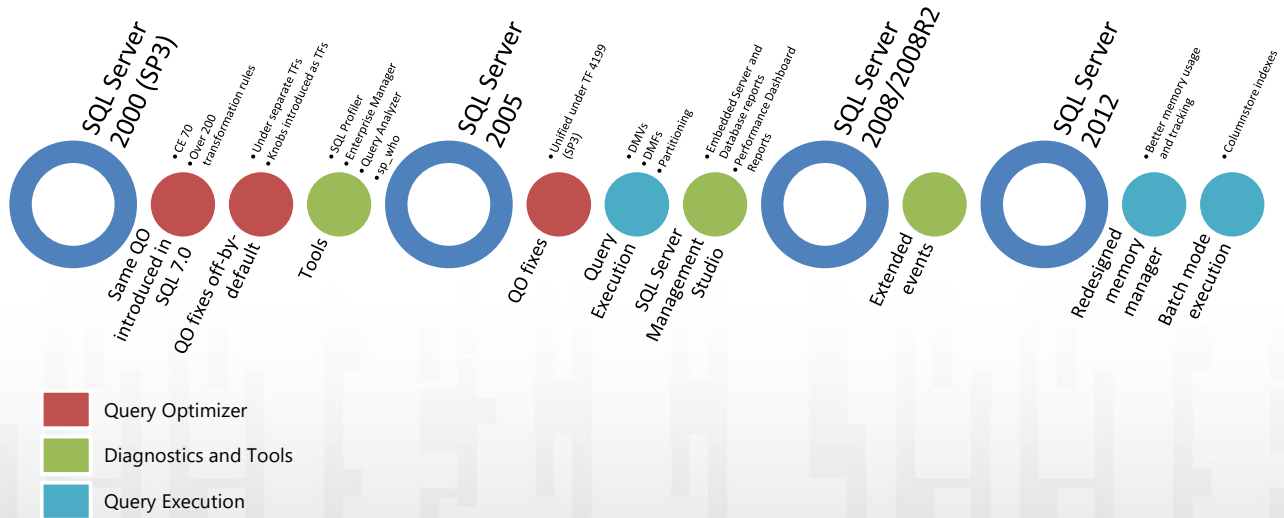
© SQLintersection. All rights reserved.  
<http://www.SQLintersection.com>

**A brief history**  
**SQL Server**  
**Query Performance**



© SQLintersection. All rights reserved.  
<http://www.SQLintersection.com>

# Query Performance Journey



Retired SQL 2000 docs available in PDF at

[https://download.microsoft.com/download/5/4/A/54AFD350-6477-4910-9DF2-4C472C906684/SQL2000\\_release.pdf](https://download.microsoft.com/download/5/4/A/54AFD350-6477-4910-9DF2-4C472C906684/SQL2000_release.pdf)

## Query Performance Journey



Unlocking insights

## Diagnostics Enhancements



© SQLintersection. All rights reserved.  
<http://www.SQLintersection.com>



# Performance Dashboard in SSMS

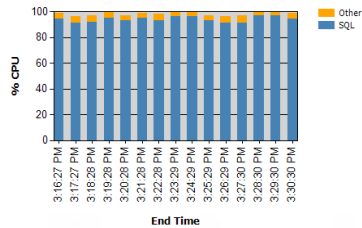
## Microsoft SQL Server Performance Dashboard

Report Local Time: 5/31/2017 3:31:04 PM

13.0.4422.0 - Enterprise Edition (64-bit)

Overall performance may be degraded because the system shows signs of being CPU-bound. This SQL Server instance is consuming the majority of the CPU. Click on any of the SQL data points in the chart below to investigate further.

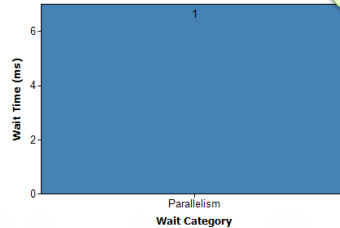
System CPU Utilization



### Current Activity

	User Requests	User Sessions
Count	27	32
Elapsed Time (ms)	4573004	741818
CPU Time (ms)	2043203(44.68%)	101108(13.63%)
Wait Time (ms)	2529801(55.32%)	640710(86.37%)
Cache Hit Ratio	100.000%	98.313%

Current Waiting Requests



Starting with SSMS v17.2

No extra downloads!

No new schema to deploy!

Long standing request by CSS and customers

### Historical Information

[Waits](#) [IO Statistics](#)

[Latches](#)

Expensive Queries

[By CPU](#)

[By Logical Reads](#)

[By Logical Writes](#)

[By Duration](#)

[By Physical Reads](#)

[By CLR Time](#)

### Miscellaneous Information

[Active Traces](#)

[Active Xevent Sessions](#)

[Databases](#)

[Missing Indexes](#)

1

4

16

11

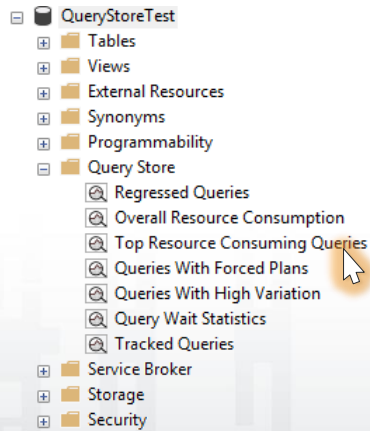
Categorized Wait stats page

New categorized Latches page

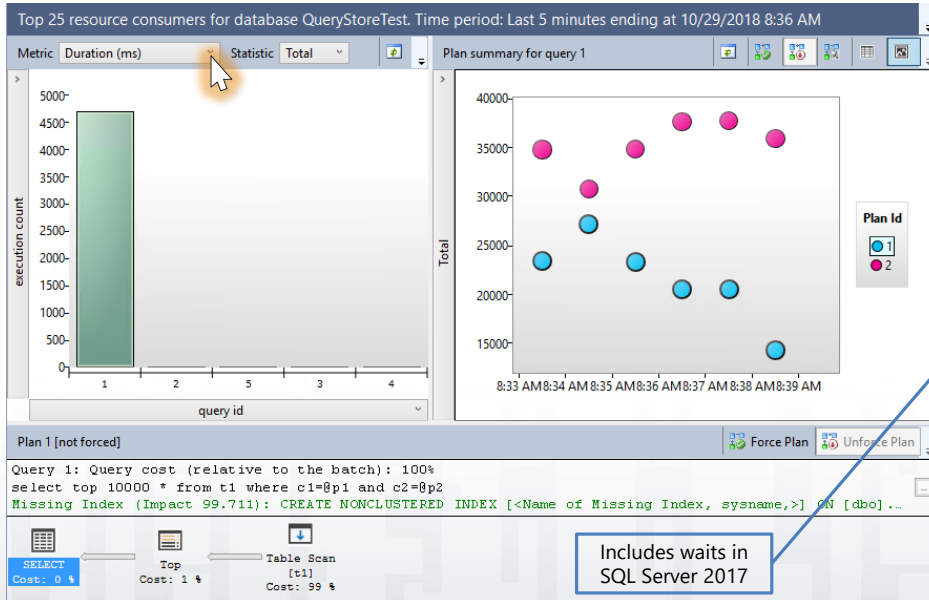
Scoring added to Missing Index Report

# Query Store

**Comprehensive query-performance information when you need it most!**



# Query Store – Top Consumers



Includes waits in SQL Server 2017

Configure Top Resource Consumption...

Resource Consumption Criteria

Check for top consumers of:

- ☐ Execution Count
- ☒ Duration (ms)
- ☐ CPU Time (ms)
- ☐ Logical Reads (KB)
- ☐ Logical Writes (KB)
- ☐ Physical Reads (KB)
- ☐ CLR Time (ms)
- ☐ DOP
- ☐ Memory Consumption (KB)
- ☐ Row Count
- ☐ Log Memory Used (KB)
- ☐ Temp DB Memory Used (KB)
- ☐ Wait Time (ms)

Based on:

- ☐ Avg
- ☐ Max
- ☐ Min
- ☐ Std Dev
- ☒ Total

Time Interval

Last 5 minutes | From: | To: |

Time Format: ☒ Local ☐ UTC

Return

☐ All

☒ Top 25

Filters

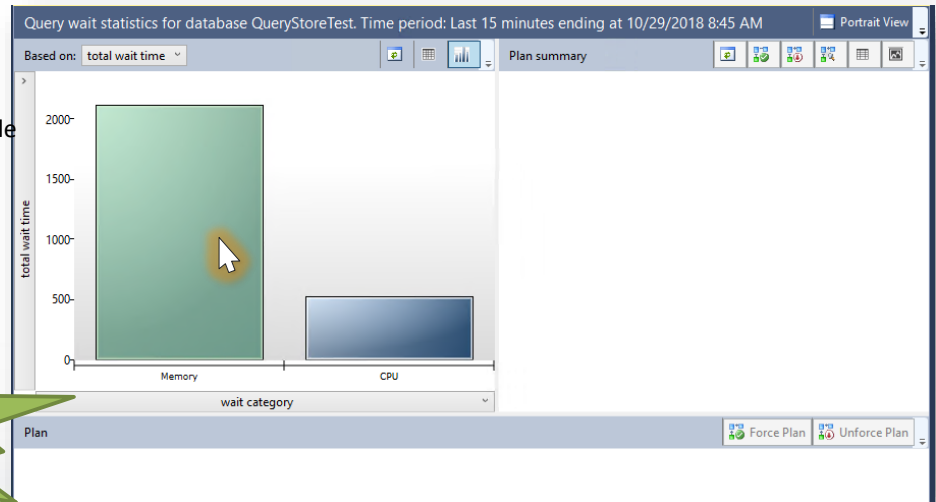
Minimum number of query plans: 1

OK Cancel Apply

## Query Store – Wait Categories

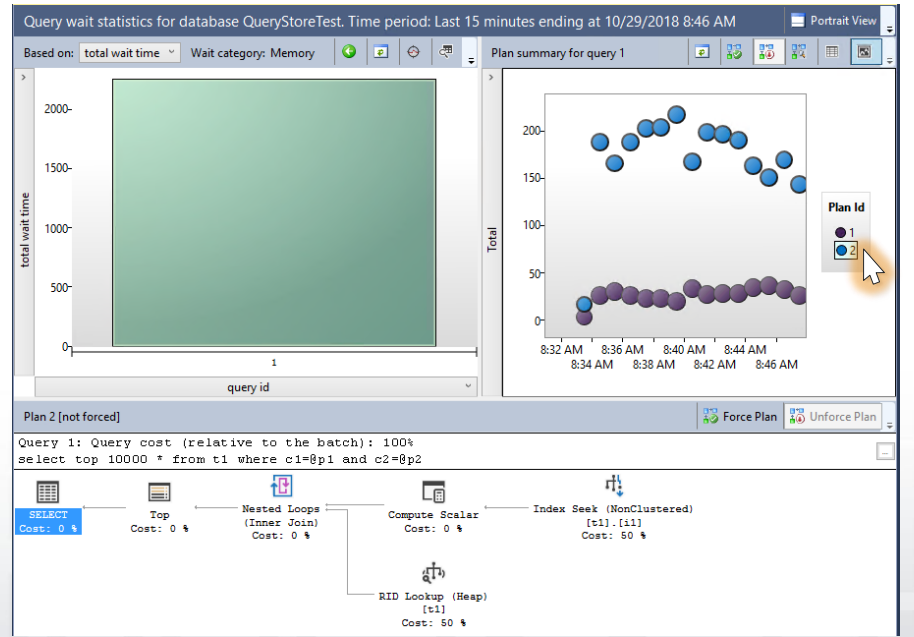
And now waits-based  
troubleshooting is also available  
in UI...

New for  
SSMS 18  
Preview 5



## Query Store – Wait Categories

Then, explore details on selected wait category



## Using Query Store UI

### Demo



© SQLintersection. All rights reserved.  
<http://www.SQLintersection.com>

5m – flow with wait stats, and then to Top resource consuming.

**Query Analysis**

**Diagnostics Enhancements**



© SQLintersection. All rights reserved.  
<http://www.SQLintersection.com>

## Improved diagnostic data for stats blocking

Ever faced long-running queries that wait on synchronous statistics updates? Essentially a blocking scenario.

- New wait type `WAIT_ON_SYNC_STATISTICS_REFRESH` in SQL Server 2019 is surfaced in `sys.dm_os_wait_stats` and `sys.dm_exec_session_wait_stats`.

wait_type	waiting_tasks_count	wait_time_ms	max_wait_time_ms	signal_wait_time_ms
<code>WAIT_ON_SYNC_STATISTICS_REFRESH</code>	1	18781	18781	0

- In `sys.dm_exec_requests` you see the command `SELECT (STATMAN)` if a `SELECT` is waiting for a synchronous stats update.

session_id	request_id	start_time	status	command
59	0	2018-11-13 11:50:36.617	suspended	<code>SELECT (STATMAN)</code>

- Not available in showplan waits – this is a pre-execution wait. Not in query store waits.



# Query plans: fundamental query perf diagnostics

- How data is accessed
- How data is joined
- Sequence of operations
- Use of temporary worktables and sorts
- Estimated rowcounts, iterations, and costs from each step
- Actual rowcounts and iterations
- How data is aggregated
- Use of parallelism
- Query execution warnings
- Query execution stats
- Hardware/Resource stats



© SQLintersection. All rights reserved.  
<http://www.SQLintersection.com>

The query plan consists of physical operators which implement the operation described by logical operators.

The order of these determine how data is accessed. A query plan also determines what support objects must be created, such as worktables or workfiles in tempdb. Remember that these decisions rely heavily on statistics, which are somewhat shown in the execution plan in the form of estimated rows.

Some physical operators may perform other types of operations e.g. the Aggregate operator calculates an expression containing MIN, MAX, SUM, COUNT or AVG, for example.

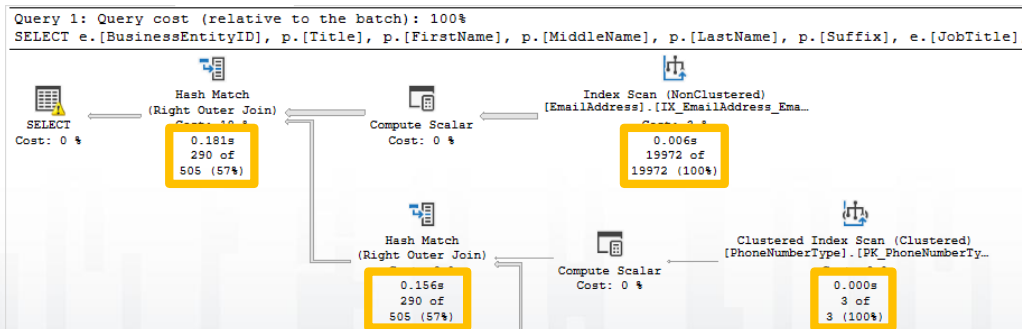
And we might have execution warnings.

More information can be found here:

Showplan Logical and Physical Operators Reference (<https://msdn.microsoft.com/en-us/library/ms191158.aspx>)

## Faster identification of heavy nodes

- SSMS v18 showplan surfaces information on:
  - Elapsed time per operator that consumes data
  - <actual rows> of <estimated rows> (percent of estimate)



**Note:** even on an Actual execution plan, the Cost pct is based on estimations. This is not an accurate measure of true operator cost.

## Getting all context info in Showplan: Trace Flags

- **Shows list of active trace flags:**
  - Query
  - Session
  - Global
- **Useful to understand if active Trace Flags influence execution context**

TraceFlags	
[1]	
IsCompileTime	True
Tracerlag	
[1]	
Scope	Global
Value	2371
[2]	
Scope	Global
Value	7412
[3]	
Scope	Session
Value	9481
[2]	
IsCompileTime	False
Tracerlag	
[1]	
Scope	Global
Value	2371
[2]	
Scope	Global
Value	7412

In SQL 2012 SP4, SQL 2014 SP3, SQL 2016 SP1 and SQL Server 2017

## Getting all context info in Showplan: Param Types

- Easier detection of type conversion issues even when you don't have source code access.

Parameter List	@CustomerID, @State
[1]	@CustomerID
Column	@CustomerID
Parameter Data Type	int
Parameter Runtime Value	(29401)
[2]	@State
Column	@State
Parameter Compiled Value	'WA'
Parameter Data Type	char(2)
Parameter Runtime Value	'WA'

Look for  
conversion  
warnings

```
DECLARE @P1 int;
EXEC sp_prepare @P1 output,
  N'@P1 nvarchar(128)',
  N'SELECT database_id, name
    FROM sys.databases
   WHERE name=@P1';

EXEC sp_execute @P1, N'tempdb';

EXEC sp_unprepare @P1;
```

In SQL 2016 SP1 and SQL 2017

## Getting all context info in Showplan: Statistics

- Identify which statistics were used by the Query Optimizer for a given compilation.
- Gain actionable insight to where estimations came from.



OptimizerStatsUsage	
Database	[AdventureWorks2016CTP3]
LastUpdate	5/12/2017 2:54 AM
ModificationCount	19027
SamplingPercent	100
Schema	[dbo]
Statistics	[IX_CustomersStatus]
Table	[CustomersStatus]

OptimizerStatsUsage	
Database	[AdventureWorks2016CTP3]
LastUpdate	5/12/2017 3:04 AM
ModificationCount	0
SamplingPercent	100
Schema	[dbo]
Statistics	[IX_CustomersStatus]
Table	[CustomersStatus]

In SQL 2017 and SQL 2016 SP2

## Getting all context info in Showplan: Times

- **Persisting information on elapsed and CPU times**

QueryTimeStats	
CpuTime	91903
ElapsedTime	92330

Is all the elapsed time spent on CPU? Look for waits

- **And Scalar UDF elapsed and CPU times**

QueryTimeStats	
CpuTime	628
ElapsedTime	1174
UdfCpuTime	443
UdfElapsedTime	445

How much elapsed time is spent on UDF?

QueryTimeStats: In SQL 2012 SP4, SQL 2014 SP3, SQL 2016 SP1 and SQL Server 2017.

UdfTimeStats: In SQL 2017 CU3, SQL 2016 SP2 and SQL 2014 SP3

Although very useful as a means to create a reusable routine, when using UDFs referenced in a query, you may be facing hidden costs that can be detrimental to performance, in a cursor like execution model.

There are several options to replace UDFs, such as Inline expressions for simple functions or derived tables if possible.

The internal execution of these functions may also be hidden from the query plan of the calling query, as we will see in the upcoming demo.

If you are using simple T-SQL UDFs that do not touch any tables (meaning do not access data), make sure you specify the SCHEMABINDING option during creation of the UDFs. This will make the UDFs schema-bound and ensure that the query optimizer does not generate any unnecessary spool operators for query plans involving these UDFs.

For UDFs that are schema-bound, any attempt to change the underlying objects' schema will raise an error. Using this option ensures that the UDF will not inadvertently break due to changes of an underlying object's schema.

## Getting all context info in Showplan: Waits

- Shows top 10 waits from `sys.dm_exec_session_wait_stats`

Correlate waits  
with overall  
query times

QueryTimeStats	
CpuTime	1045
ElapsedTime	3010

Misc	
Cached plan size	160 KB
CardinalityEstimationModelV	130
CompileCPU	11
CompileMemory	728
CompileTime	136
DatabaseContextSettingsId	3
Degree of Parallelism	12
Estimated Number of Rows	121308
Estimated Operator Cost	0 (0%)
Estimated Subtree Cost	4,48002
Memory Grant	80448
MemoryGrantInfo	
Optimization Level	FULL

WaitStats	
[1]	
WaitCount	98
WaitTimeMs	3
WaitType	LATCH_SH
[2]	
WaitCount	50
WaitTimeMs	761
WaitType	PAGEIOLATCH_SH
[3]	
WaitCount	67
WaitTimeMs	1942
WaitType	LATCH_EX
[4]	
WaitCount	129
WaitTimeMs	2509
WaitType	ASYNC_NETWORK_IO
[5]	
WaitCount	2220
WaitTimeMs	30622
WaitType	CXPACKET

Note: Parallelism waits available in SQL Server 2017 CU3 and 2016 SP2

SQL 2016 SP1 and SQL 2017

## Getting all context info in Showplan: memory

- **Showplan extended to include grant usage per thread and iterator**

Memory Grant	783288
MemoryGrantInfo	
DesiredMemory	28592000
GrantedMemory	783288
GrantWaitTime	0
MaxUsedMemory	0
RequestedMemory	783288
RequiredMemory	4064
SerialDesiredMemory	28588448
SerialRequiredMemory	512

Is the used memory close to granted?

Is the memory above granted?  
Look for grant warnings

- **Also found in sys.dm\_exec\_query\_stats**

SQL 2017 and SQL 2016 SP2

### Memory grant info:

**SerialRequiredMemory:** Required memory in KB if the query runs in serial mode. The query will not start without this memory.

**SerialDesiredMemory:** Memory estimated to fit intermediate results in KB if the query runs in serial mode.

**RequiredMemory:** Required memory in KB for the chosen degree of parallelism. If the query runs in serial mode, this is the same as SerialRequiredMemory.

**DesiredMemory:** Memory estimated to fit intermediate results in KB for the chosen degree of parallelism. If the query runs in serial mode, this is the same as SerialDesiredMemory.

**RequestedMemory:** Memory in KB which the query requests the memory manager to grant. This can be smaller than sum of RequiredMemory and DesiredMemory if it exceeds the maximum allowed for single query.

**GrantWaitTime:** Time in seconds if the query has to wait for successful memory grant.

**MaxUsedMemory:** Maximum memory in KB used by the query.

**MaxQueryMemory:** Maximum memory in KB allowed for single query.



**Notice the warnings**

## **Analyzing query plan properties**



© SQLintersection. All rights reserved.  
<http://www.SQLintersection.com>

## Warning: Memory Grant Wait

- **Occurs when a T-SQL statement or stored procedure waits more than one second for a memory grant or when the initial attempt to get memory fails.**

RESOURCE\_SEMAPHORE waits may indicate excessive number of concurrent queries, or excessive memory request amounts

[illegible]

In SQL 2012 +

Ex. Use this event in combination with events that identify waits to troubleshoot contention issues that impact performance. For excessive mem grant, you can see how to improve estimations (perhaps update stats), or administratively by using `MIN_PERCENT_GRANT` and /or `MAX_PERCENT_GRANT` query hints.

## Warning: Memory Grant

### 3 conditions:

- **Excessive Grant:** when max used memory is too small compared to the granted memory. This scenario can cause blocking and less efficient usage when large grants exist, and a fraction of that memory was used.

The screenshot shows a query plan in SQL Server Enterprise Manager. The plan consists of three steps: a 'SELECT' step (Cost: 0%), a 'Sort' step (Cost: 2%), and a 'Hash Match (Inner Join)' step (Cost: 13%). The 'SELECT' step is highlighted, and its properties window is open. The properties window shows the following details:

SELECT	
Actual Number of Rows	0
Cached plan size	64 KB
Degree of Parallelism	0
Estimated Operator Cost	0 (0%)
Estimated Subtree Cost	0.205452
Memory Grant	67808
Estimated Number of Rows	89.3525

Below the properties, the SQL statement is displayed:

```
SELECT
[fo].[Order Key], [fo].[Description]
FROM [Fact].[Order] AS [fo]
INNER HASH JOIN [Dimension].[Stock
Item] AS [si]
ON [fo].[Stock Item Key] = [si].[Stock Item
Key]
WHERE [fo].[Lineage Key] =
@LineageKey
AND [si].[Lead Time Days] > 0
ORDER BY [fo].[Stock Item Key], [fo].[Order
Date Key] DESC
OPTION (MAXDOP 1)
```

A warning box is visible at the bottom of the properties window, titled 'Warnings'. It contains the following text:

**Warnings**  
The query memory grant detected  
"ExcessiveGrant", which may impact the  
reliability. Grant size: Initial 67808 KB, Final  
67808 KB, Used 1024 KB.

Also in SQL Server 2014 SP2 and SQL Server 2016 SP1.

Ex. For excessive mem grant, you can see how to improve estimations (perhaps update stats), or administratively by using `MIN_PERCENT_GRANT` and /or `MAX_PERCENT_GRANT` query hints.

## Warning: Memory Grant

### 3 conditions:

- ❑ **Excessive Grant:** when max used memory is too small compared to the granted memory. This scenario can cause blocking and less efficient usage when large grants exist, and a fraction of that memory was used.
- ❑ **Grant Increase:** when the dynamic grant starts to increase too much, based on the ratio between the max used memory and initial request memory. This scenario can cause server instability and unpredictable workload performance.
- ❑ **Used More Than Granted:** when the max used memory exceeds the granted memory. This scenario can cause OOM conditions on the server.

The image shows a SQL Server query plan for a SELECT statement. The plan includes a 'SELECT' operator with a 'Cost' of 13. The 'Statement' pane shows the following query:

```
SELECT
[fo].[Order Key], [fo].[Description]
FROM   [Fact].[Order] AS [fo]
INNER HASH JOIN [Dimension].[Stock
Item] AS [si]
ON [fo].[Stock Item Key] = [si].[Stock Item
Key]
WHERE  [fo].[Lineage Key] =
@LineageKey
AND [si].[Lead Time Days] > 0
ORDER BY [fo].[Stock Item Key], [fo].[Order
Date Key] DESC
OPTION (MAXDOP 1)
```

The 'Warnings' pane at the bottom of the plan shows the following message:

```
The query memory grant detected
"GrantIncrease", which may impact the
reliability. Grant size: Initial 2200 KB, Final
5272 KB, Used 4816 KB.
```

SELECT	
Cached plan size	64 KB
Degree of Parallelism	0
Estimated Operator Cost	0 (0%)
Memory Grant	5272
Estimated Subtree Cost	0.205452
Estimated Number of Rows	89.3525

Ex. For excessive mem grant, you can see how to improve estimations (perhaps update stats), or administratively by using MIN\_PERCENT\_GRANT and /or MAX\_PERCENT\_GRANT query hints.

# Warning: Spills

**Sort Operator Details:**

Physical Operation	Sort
Logical Operation	Sort
Actual Execution Mode	Row
Estimated Execution Mode	Row
Actual Number of Rows	121317
Actual Number of Batches	0
Estimated Operator Cost	1.23741 (32%)
Estimated I/O Cost	0.0018769
Estimated CPU Cost	1.23553
Estimated Subtree Cost	2.71983
Estimated Number of Executions	1
Number of Executions	12
Estimated Number of Rows	97454.1
Estimated Row Size	332 B
Actual Rebinds	12
Actual Rewinds	0
Node ID	2

**Warnings:**  
Operator used tempdb to spill data during execution with spill level 1 and 12 spilled thread(s). Sort wrote 4432 pages to and read 4432 pages from tempdb with granted memory 50400KB and used memory 39704KB

**Order By:**  
[AdventureWorks2014].[Production].[Product].Style  
Ascending

**Hash Match Operator Details:**

Physical Operation	Hash Match
Logical Operation	Inner Join
Actual Execution Mode	Row
Estimated Execution Mode	Row
Actual Number of Rows	19620
Actual Number of Batches	0
Estimated I/O Cost	0
Estimated Operator Cost	0.1200468 (20%)
Estimated CPU Cost	0.11053
Estimated Subtree Cost	0.591696
Number of Executions	1
Estimated Number of Executions	1
Estimated Number of Rows	200
Estimated Row Size	11 B
Actual Rebinds	0
Actual Rewinds	0
Node ID	0

**Output List:**  
[AdventureWorks2014].[Sales].[Customer].CustomerID

**Warnings:**  
Operator used tempdb to spill data during execution with spill level 1 and 1 spilled thread(s). Hash wrote 32 pages to and read 32 pages from tempdb with granted memory 1152KB and used memory 992KB

**Hash Keys Probe:**  
[AdventureWorks2014].[Sales].[Customer].CustomerID

**Callout Boxes:**

- Is this spill relevant to go after? Does it consume too many resources?
- And this one? What if this executes dozens of times per minute?

In SQL 2014 SP2; SQL 2016

Sort Spills = sort operations that do not fit into memory

Supported by a Worktable in TempDB

## Spill level 1

Means one pass over the data was enough to complete the sort.

## Spill level 2

Means multiple passes over the data are required to sort the data

Hash Spills = hash recursion or cessation of hashing (hash bailout) has occurred during a hashing operation

Supported by a Workfile in TempDB

## Spill level 1 = Hash recursion

Occurs when the build input does not fit into available memory, resulting in the split of input into multiple partitions that are processed separately.

If any of these partitions still do not fit into available memory, it is split into sub-partitions, which are also processed separately. This splitting process continues until each partition fits into available memory or until the maximum recursion level is reached.

### **Spill level 2 = Hash bailout**

Occurs when a hashing operation reaches its maximum recursion level and shifts to an alternate plan to process the remaining partitioned data.

**Faster insights**



© SQLintersection. All rights reserved.  
<http://www.SQLintersection.com>

## The middle-of-the-night call

- You're on call for supporting the data tier of a mission-critical SQL Server instance.
- Key business processes are being delayed when ETL is running.
- You get a call asking to mitigate the issue and then determine the root cause.





## Defining the problem

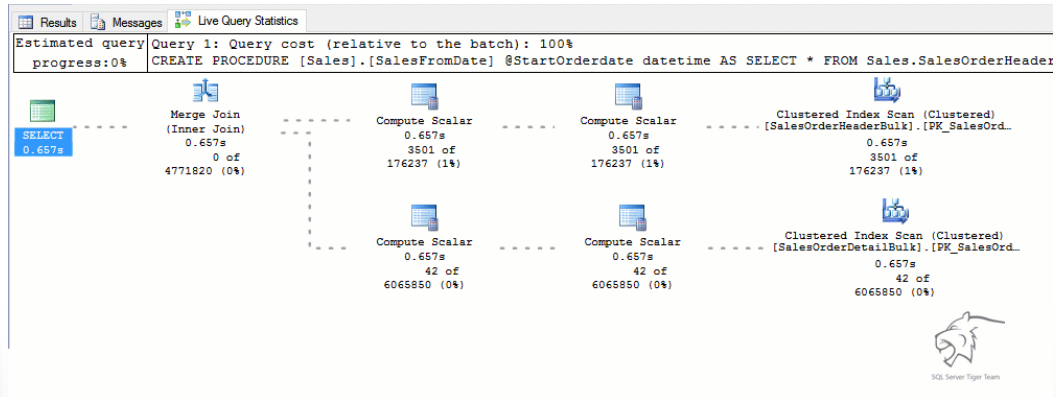
**Reasonable hypothesis: a long running query.**

**Query completion is a prerequisite for the availability of an actual query plan.**

**Actual query plans unsuitable for troubleshooting complex performance issues:**

- **Long running queries**
- **Queries that run indefinitely and never finish execution.**

# What if I could do live query troubleshooting?



- **Overhead goes up to 75% with TPC-C like workload.**
- **It makes bad things worse if running all the time...**

Processes									
S...	U...	Logi...	Dat...	Tas...	Com...	Appl...	Wait Tim...	Wait...	
51	1		master			Microsoft...		0	
53	1		tempdb	RUNNING	SELECT	Microsoft...		0	
54	1		Adventur...	SUSPEN...	SELECT	SQLCMD		25 CXPACK...	
54	1		Adventur...	SUSPEN...	SELECT	SQLCMD		25 CXPACK...	
54	1		Adventur...	SUSPEN...	SELECT	SQLCMD		25 CXPACK...	
54	1		Adventur...	SUSPEN...	SELECT	SQLCMD		25 CXPACK...	
54	1		Adventur...	SUSPEN...	SELECT	SQLCMD		25 CXPACK...	
54	1		Adventur...	SUSPEN...	SELECT	SQLCMD		25 CXPACK...	
54	1		Adventur...	SUSPEN...	SELECT	SQLCMD		25 CXPACK...	
54	1		Adventur...	SUSPEN...	SELECT	SQLCMD		25 CXPACK...	
54	1		Adventur...	SUSPEN...	SELECT	SQLCMD		25 CXPACK...	
54	1		Adventur...	SUSPEN...	SELECT	SQLCMD		25 CXPACK...	
54	1		Adventur...	SUSPEN...	SELECT	SQLCMD		25 CXPACK...	
54	1		Adventur...	SUSPEN...	SELECT	SQLCMD		25 CXPACK...	

Details

Show Live Execution Plan

Kill Process

Trace Process in SQL Server Profiler

### Resource Waits


### Data File I/O

### Recent Expensive Queries

### Active Expensive Queries

## Query progress – anytime, anywhere

- Starting with SQL Server 2016 SP1\* and 2017, the new *lightweight query execution statistics profile infrastructure* allows continuous collection of per-operator query execution statistics
  - Using global TF 7412
  - Enabling *query\_thread\_profile* and *query\_plan\_profile* extended event
  - Using query hint **USE HINT('query\_plan\_profile')** in SQL Server 2017 CU11 and 2016 SP2 CU3 (KB 4458593)



Default in SQL  
Server 2019 and  
no TF needed

\* Also available in SQL 2014 SP2 and 2016 RTM as a less optimized versions than SQL 2016 SP1 and 2017. Becomes default ON for SQL 2019.

## Query progress – anytime, anywhere

- When lightweight profiling is on, *sys.dm\_exec\_query\_profiles* is also populated for all sessions
  - Enables new LQS feature in SSMS (including Activity Monitor)
  - New DMF `sys.dm_exec_query_statistics_xml`
- The following still use regular profiling infra
  - SET STATISTICS XML (or Include Actual Plan)
  - query\_post\_execution\_showplan extended event

S...	U...	Login	Dat...	Tas...	Com...	Appl...	Wait Tim...	Wait...
57	1	Adventur...	RUNNA...	SELECT	SQLCMD		0	
57	1	Adventur...	RUNNA...	SELECT	SQLCMD		0	
57	1	Adventur...	RUNNA...	SELECT	SQLCMD		0	
57	1	Adventur...	RUNNA...	SELECT	SQLCMD		0	
57	1	Adventur...	RUNNA...	SELECT	SQLCMD		328	CXPACK...
57	1	Adventur...	RUNNA...	SELECT	SQLCMD		328	CXPACK...
57	1	Adventur...	RUNNA...	SELECT	SQLCMD		328	CXPACK...
57	1	Adventur...	RUNNA...	SELECT	SQLCMD		328	CXPACK...
57	1	Adventur...	RUNNA...	SELECT	SQLCMD		0	

**Resource Waits**

**Data File I/O**

**Recent Expensive Queries**

**Active Expensive Queries**

\* Also available in SQL 2014 SP2 and 2016 RTM as a less optimized versions than SQL 2016 SP1 and 2017.

See more information in [https://blogs.msdn.microsoft.com/sql\\_server\\_team/query-progress-anytime-anywhere/](https://blogs.msdn.microsoft.com/sql_server_team/query-progress-anytime-anywhere/) and <https://docs.microsoft.com/en-us/sql/relational-databases/performance/query-profiling-infrastructure>

**Bringing it all together with live troubleshooting + new xEvents**

**Demo**



© SQLintersection. All rights reserved.  
<http://www.SQLintersection.com>

3m demo on finding offending workload with LQS and fixing it with TF OFF.

## What is the impact of Lightweight Profiling?

**Query Execution Statistics Profiling Infrastructure tests with TPC-C like workloads**


Infra Type	Overhead percent (up to)	
	no active xEvents	Active xEvent query_post_execution_showplan
Regular	<b>75.5</b>	93.17
Lightweight in SQL Server 2014 SP2/2016	3.5	62.02
Lightweight in SQL Server 2016 SP1 and above	<b>2</b>	14.3

Reference: [https://blogs.msdn.microsoft.com/sql\\_server\\_team/query-progress-anytime-anywhere/](https://blogs.msdn.microsoft.com/sql_server_team/query-progress-anytime-anywhere/)

## The infamous truncation error

### Loading data into a table, and hitting this?

Msg 8152, Level 16, State 30, Line 13  
String or binary data would be truncated.  
The statement has been terminated.



It will be  
default in SQL  
Server 2019

### Starting with SQL Server 2019 CTP 2.1, you get this:

Msg 2628, Level 16, State 1, Line 14  
String or binary data would be truncated in table  
'Adventureworks2016CTP3.Sales.SalesOrderHeaderTest', column  
'CreditCardApprovalCode'. Truncated value: '1231736V18604'.  
The statement has been terminated.

Also in SQL Server 2017 CU12 and upcoming SQL Server 2016 SP2 CUx

Needs TF 460

See more in blog post @ <http://aka.ms/sqlserverteam>

Link to blog post: [https://blogs.msdn.microsoft.com/sql\\_server\\_team/string-or-binary-data-would-be-truncated-replacing-the-infamous-error-8152/](https://blogs.msdn.microsoft.com/sql_server_team/string-or-binary-data-would-be-truncated-replacing-the-infamous-error-8152/)



## Bookmarks



SQL Server Tiger Team

SQL Server Team (Tiger) Blog	<a href="http://aka.ms/sqlserverteam">http://aka.ms/sqlserverteam</a>
Tiger Toolbox GitHub	<a href="http://aka.ms/tigertoolbox">http://aka.ms/tigertoolbox</a>
SQL Server Release Blog	<a href="http://aka.ms/sqlreleases">http://aka.ms/sqlreleases</a>
SQL Server Performance Center	<a href="http://aka.ms/sqlperfcenter">http://aka.ms/sqlperfcenter</a>
BP Check	<a href="http://aka.ms/bpcheck">http://aka.ms/bpcheck</a>
SQL Server Standards Support	<a href="http://aka.ms/sqlstandards">http://aka.ms/sqlstandards</a>
Trace Flags	<a href="http://aka.ms/traceflags">http://aka.ms/traceflags</a>
SQL Server Support lifecycle	<a href="http://aka.ms/sqlifecycle">http://aka.ms/sqlifecycle</a>
SQL Server Updates	<a href="http://aka.ms/sqlupdates">http://aka.ms/sqlupdates</a>
SQL Server Guides	<a href="http://aka.ms/sqlserverguides">http://aka.ms/sqlserverguides</a>
SQL Feedback (New "Connect")	<a href="http://aka.ms/sqlfeedback">http://aka.ms/sqlfeedback</a>
T-SQL Syntax Conventions	<a href="http://aka.ms/sqlconventions">http://aka.ms/sqlconventions</a>
SQL Server Error Codes	<a href="http://aka.ms/sqlerrors">http://aka.ms/sqlerrors</a>
Database Compatibility	<a href="http://aka.ms/dbcompat">http://aka.ms/dbcompat</a>

# Questions?



Don't forget to complete an online evaluation!

**Practical guidance to make your tier-1 SQL Server ROAR!**

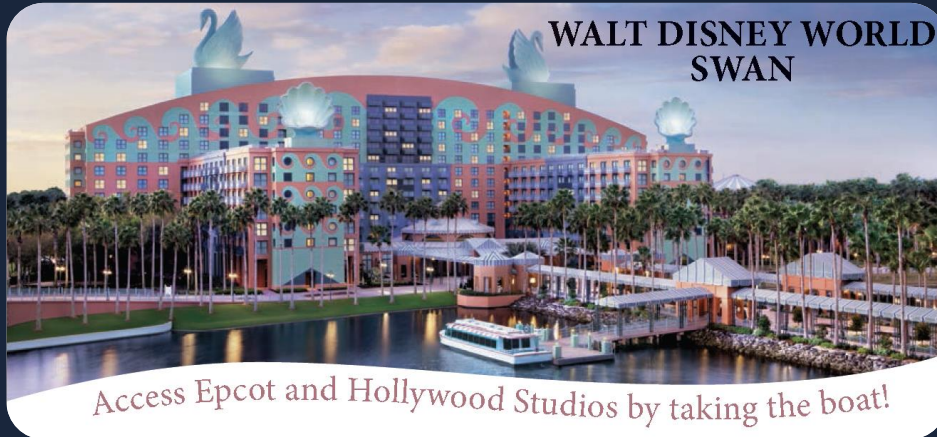
Your evaluation helps organizers build better conferences  
and helps speakers improve their sessions.



**Thank you!**

# Save the Date!

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



## 2019

### June 9-14

*We're back in Orlando!*



*Leave the every day behind and enter a world of wonder and enchantment at the Walt Disney World® Resort. Located in the heart of the most magical place on earth, the Walt Disney World Swan and Dolphin Resort provides a truly extraordinary backdrop for our event! Beautiful tropical landscaping, tranquil waterways, and classic art and architecture work together to create a stunning landmark!*