



# **Banishing Slow Queries with SQL Server 2019 Intelligent Query Processing**

**John Morehouse**  
**john@dcac.com**

03 June 2022

Data Grillen

**DCAC**✱



# John Morehouse

Principal Consultant


Denny Cherry & Associates

✉ [john@dcac.com](mailto:john@dcac.com)

 [/in/johnmorehouse](https://www.linkedin.com/in/johnmorehouse)

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

 [Sqlrus.com](http://Sqlrus.com)

 He/Him

MVP - Data Platform  
Avid VMWare vExpert  
Friend of Red Gate  
Nerd

---

# Denny Cherry & Associates

---



Certified IT professionals to help achieve IT goals

Clients ranging from small business to Fortune 10 corporations

Help save on costs while improving IT reliability and solving challenges



# AGENDA

---



01

**Lightweight  
Query  
Profiling**

02

**Table  
Variable  
Deferred  
Compilation**

03

**Scalar UDF  
Inlining**

04

**Row Mode  
Memory  
Grant  
Feedback**

05

**Batch Mode  
on Rowstore**

06

**Demos**



**Prince**

**Artist Formerly Known As**

Adaptive  
Query  
Processing



A silhouette of a person walking a tightrope high above a canyon. The person is holding a long pole for balance. The background shows a blue sky with scattered white clouds and the dark, jagged peaks of the canyon below. The scene conveys a sense of risk and challenge.

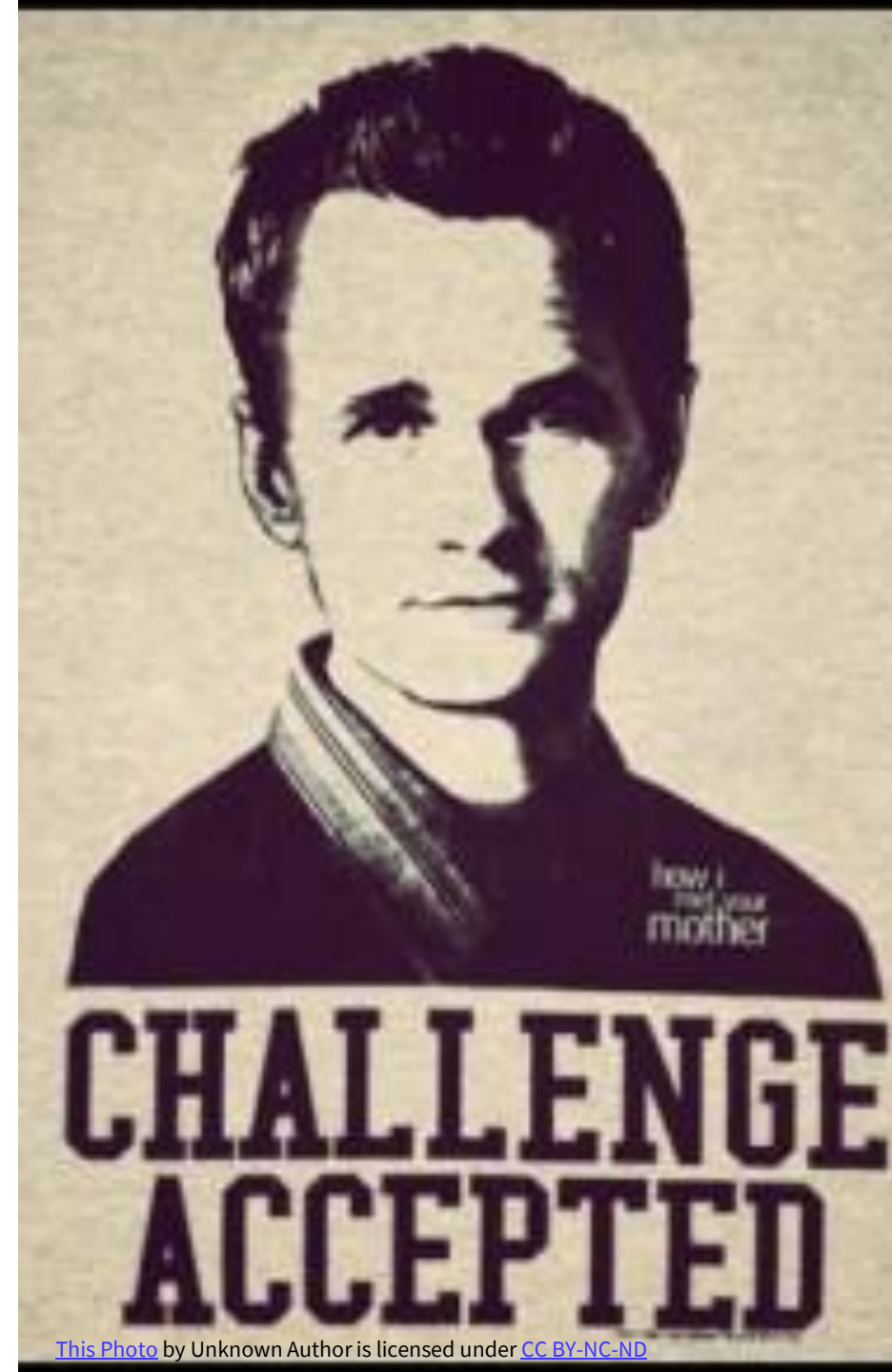
# CHALLENGES

# CHALLENGE ACCEPTED

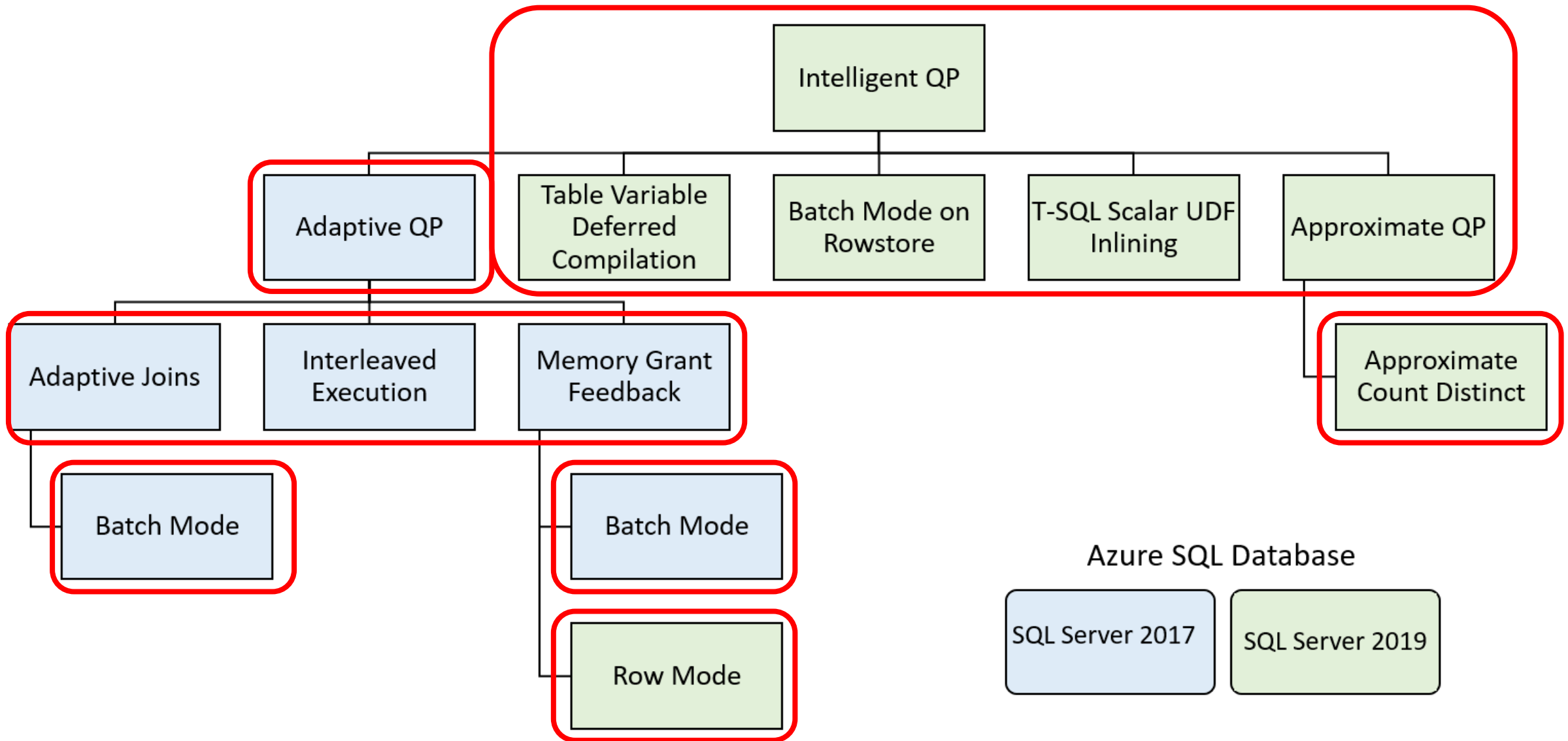
Lightweight Query Profiling

Provides framework for  
performance improvements

< SQL Server 2019 = Opted In



# SQL Server 2019 Intelligent Performance







# **COMPATIBILITY LEVEL**

# DMVS

## Query Inflight - 2016 SP1 or higher

*sys.dm\_exec\_query\_statistics\_xml(sessionID)*

## Query Completion - 2019+

*sys.dm\_exec\_query\_plan\_stats(plan\_handle)*

LAST\_QUERY\_PLAN\_STATS



# Table Variables Issues



Estimated Subtree Cost	0.00000001
Number of Executions	1
Estimated Number of Executions	1
Estimated Number of Rows to be Read	1
Estimated Number of Rows Per Execution	1
Estimated Row Size	15 B
Actual Rebinds	0
Actual Rewinds	0
Ordered	False
Node ID	4

Object

[@table] [a]

Output List

@table.objectid

# Table Variable Deferred Compilation

---



< SQL Server 2019 = 1 row fixed estimate

Too many rows = bad execution plan

Temporary Table vs Table Variable

# Table Variable Deferred Compilation

---



Compilation is shifted to first actual execution of a statement

Compilation frequency remains the same

Now mimics temporary table behavior





**Scalar**

**UDF**

**Inlining**

# Scalar UDF Inlining

---



Serial vs Parallel Execution Plans

Scalar UDFs = Poor Performance  
(usually)

Scalar UDFs previously not costed  
during compilation

# Scalar UDF Inlining

```
ALTER DATABASE SCOPED CONFIGURATION SET  
TSQL_SCALAR_UDF_INLINING = OFF;
```

```
SELECT Sales.SalesQuantity(sol.Description)  
as quantity  
FROM Sales.Orders so  
      JOIN Sales.OrderLines as sol on  
so.OrderId=sol.OrderID  
OPTION (USE  
HINT( 'DISABLE_TSQL_SCALAR_UDF_INLINING' ));
```

```
CREATE OR ALTER FUNCTION  
dbo.SalesPricesCalculation (@itemID, @cost  
decimal(10,2))  
RETURNS DECIMAL(10,2)  
WITH INLINE = OFF
```

# Scalar UDF Inlining

## Inlineable scalar UDFs requirements

A scalar T-SQL UDF can be inline if all of the following conditions are true:

Improved  
Performance

Limitations →

HOPEFULLY...



Query executed successfully.

```
3
4 SELECT inline_type
5       , is_inlineable
6       , object_id
7       , definition
8 FROM sys.sql_modules
9 WHERE object_id = Object_Id('sales.SalesQuantity')
10
```

1 %

Results Messages

	inline_type	is_inlineable	object_id	definition
1	1	1	935674381	CREATE FUNCTION Sales.SalesQuantity (@D...





# Memory Grant Feedback

# Memory Grant Feedback

---



Added in SQL Server 2017 (COMPAT = 140), but for batch mode only.

Does not actually change the execution plan but changes the context.

# Memory Grant Feedback

---



Increase memory grant if the query spills to TempDB

Decrease memory grant if deemed excessive





# Batch Mode on Rowstore

# Batch Mode on Rowstore

---



Introduced with SQL Server 2012 & columnstore indexes

Batch mode processes ~900 rows at a time



# Batch Mode on Rowstore

Reduces CPU  
consumption, so great  
for analytical queries

Batch or Row; Not both





# DEMO

# Feature Availability



Feature	Enterprise	Standard
Batch Mode for Row Store	Yes	No
Row Mode Memory Grant Feedback	Yes	No
Table Variable Deferred Compilation	Yes	Yes
Scalar UDF Inlining*	Yes	Yes

# 2022 Enhancements (Forthcoming)

---



Parameter Sensitivity Plan (PSP)

Memory Grant Feedback - Persistence and Percentile

Degree of Parallelism Feedback

# Summary

---



Performance Game Changer

Increased flexibility

Reminder: Compatibility Level = 150

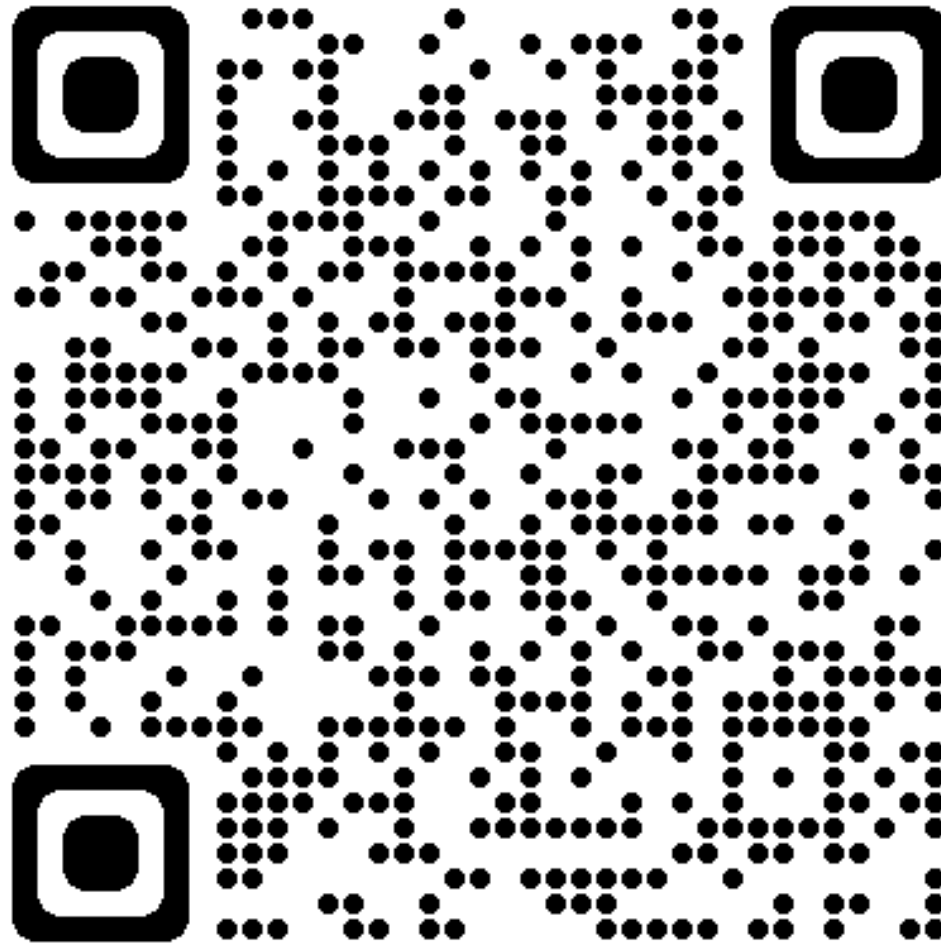




**BYE BYE BAD ANTI-PATTERNS!!**

# Evaluations

---



<https://evals.datagrillen.com/evals.aspx>

**Got  
Questions?**

**Follow Me  
on Twitter!**

**Check out  
my blog!**

# John Morehouse

**Denny Cherry & Associates Consulting**

**DCAC\***



**[john@dcac.com](mailto:john@dcac.com)**



**Sqlrus.com**



**@SQLRUS**



**/in/johnmorehouse**