



Banishing Slow Queries with Intelligent Query Processing

John Morehouse
john@dcac.com



DCAC*



John Morehouse

Principal Consultant

Denny Cherry & Associates

✉ john@dcac.com

in /in/johnmorehouse

🐦 @SQLRUS

🌐 Sqlrus.com

👤 He/Him



Found in Kentucky, USA

Blogger/Tweeter

Community Junkie

Microsoft Data Platform
MVP

VMWare vExpert

Basically, a Nerd

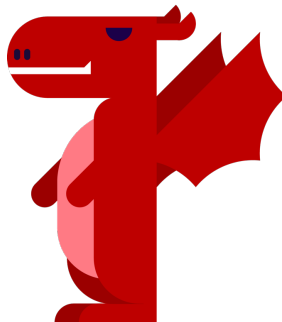
Denny Cherry & Associates



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

**Memory
Grant
Feedback**

04

**Parameter
Sensitive
Plans**

05

**Degree of
Parallelism
Feedback**

06

Demos





Prince

Artist Formerly Known As

Adaptive
Query
Processing



CHALLENGES

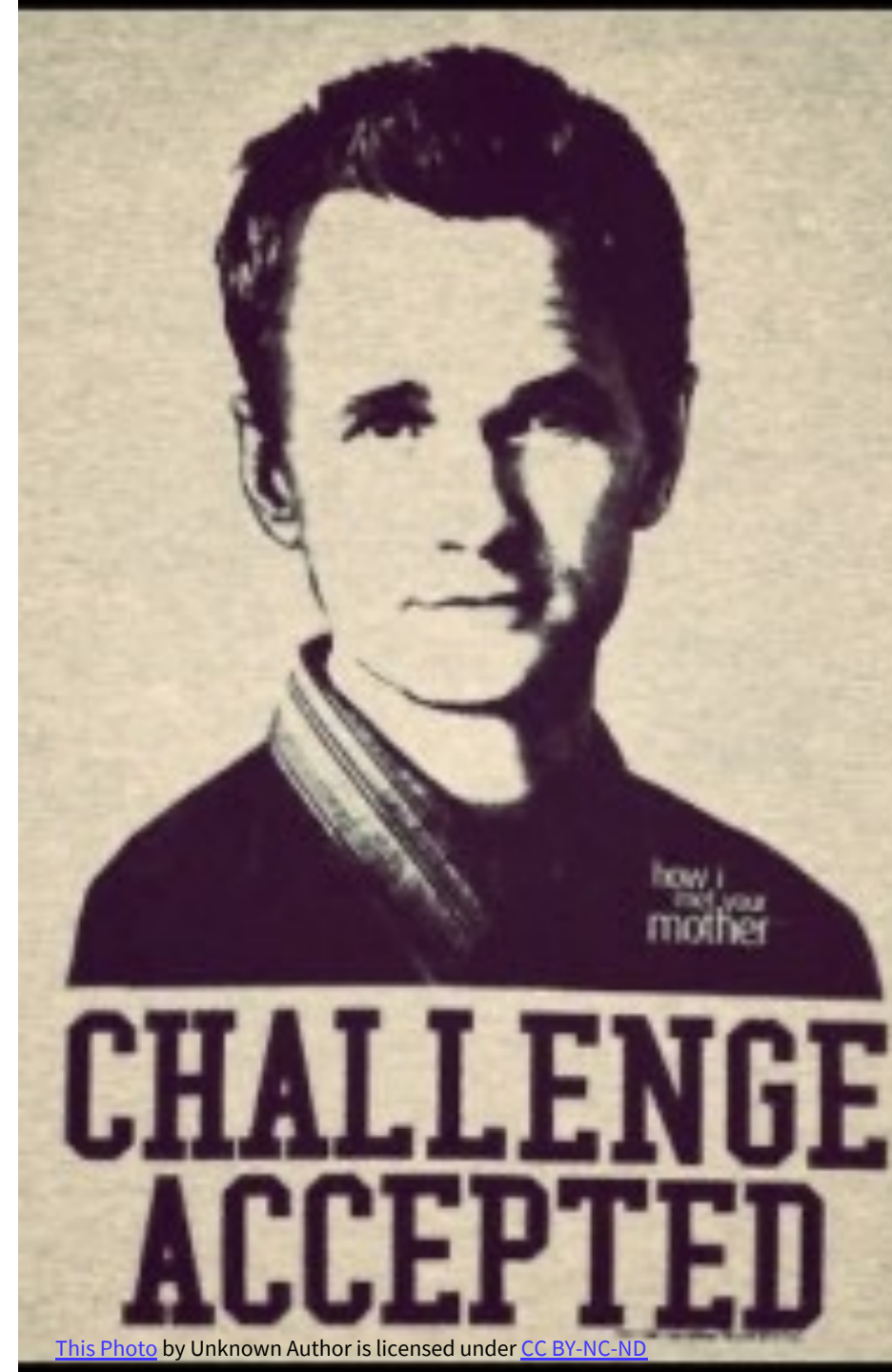
CHALLENGE ACCEPTED

Lightweight Query Profiling

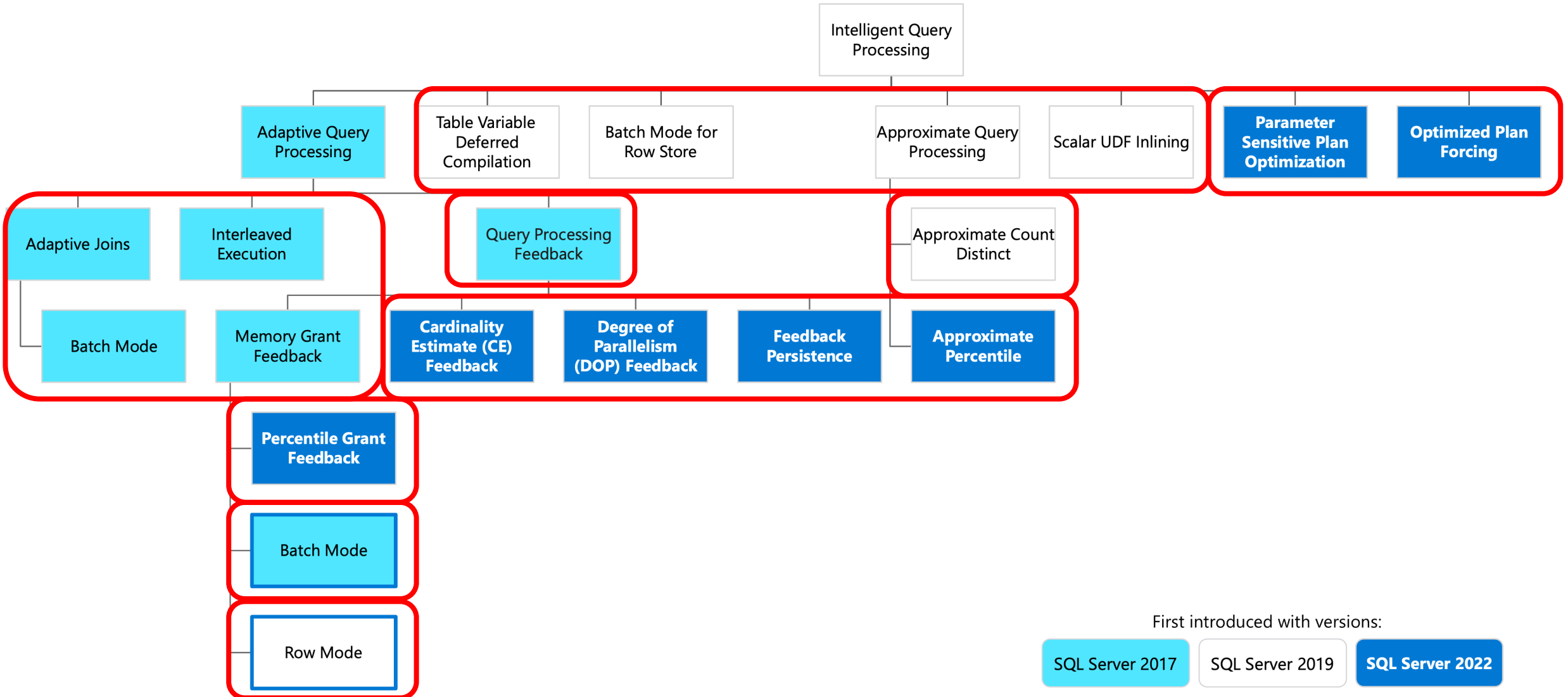
Provides framework for
performance improvements

< SQL Server 2019 = Opted In

Query Store utilized in cases



SQL Server 2022 IQP





COMPATIBILITY LEVEL



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 Variable Deferred Compilation

Table Variables Issues



Estimated Subtree Cost	0.0002031
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

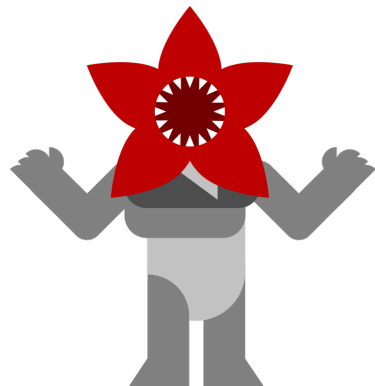


Table Variable Deferred Compilation



Compilation is shifted to first actual execution of a statement

Utilizes actual cardinality vs wild guess

Mimics temporary table behavior





Memory Grant Feedback

Memory Grant Feedback



Added in SQL Server 2017 but for batch mode only.

SQL 2019 Added Row Mode

Partially requires query store (persistence)

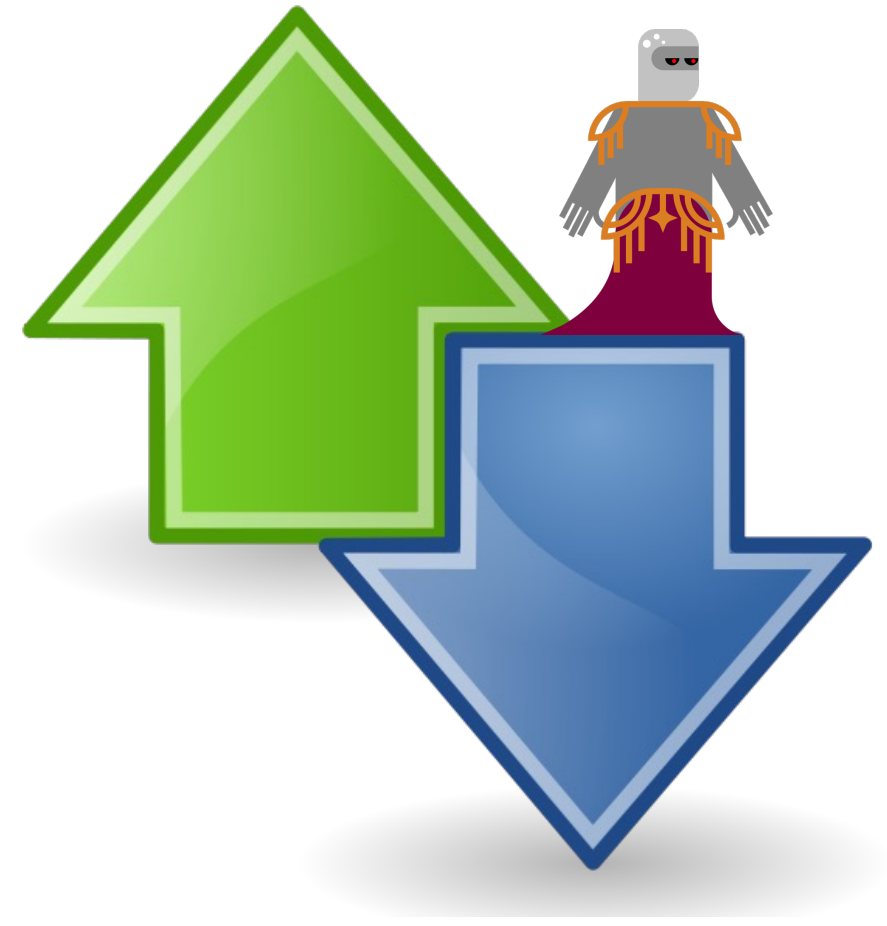


Memory Grant Feedback



Increase memory grant
if the query spills to
TempDB

Decrease memory grant
if deemed excessive





Parameter Sensitive Plan

Parameter Sensitive Plan



SQL Server 2022

Helps solve parameter sniffing issues

Only works on equality predicates

Query/Dispatcher Plan/Query Variant

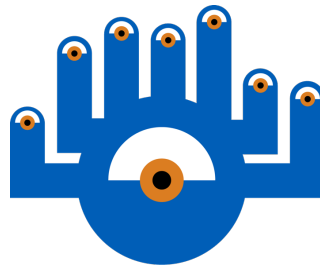
Parameter Sensitive Plan



Predicate Cardinality evaluated at runtime for parameter values.

Cardinality values placed into three predicate cardinality "buckets" via Dispatcher.

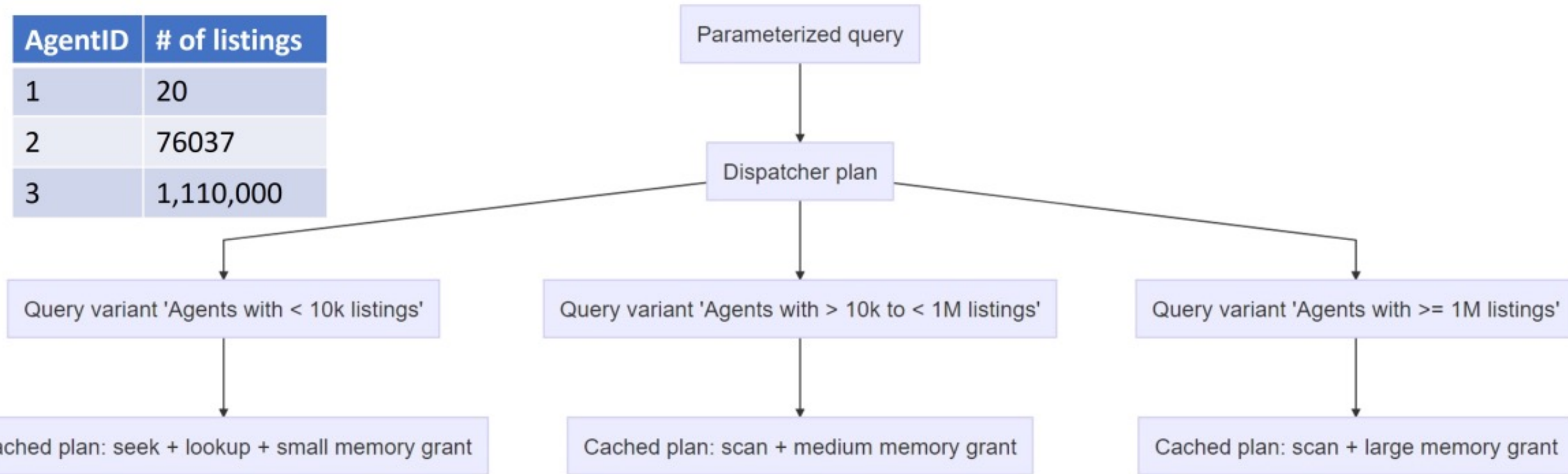
Child queries are then compiled and executed generating query variants. They have their own execution plan in Query Store.



Parameter Sensitive Plan



AgentID	# of listings
1	20
2	76037
3	1,110,000



sys.query_store_plan

sys.query_store_query_variant



Degree of Parallelism Feedback

DOP Feedback

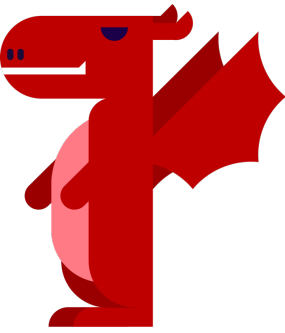


Uses elapsed time and waits to identify inefficient parallelism for repeating queries

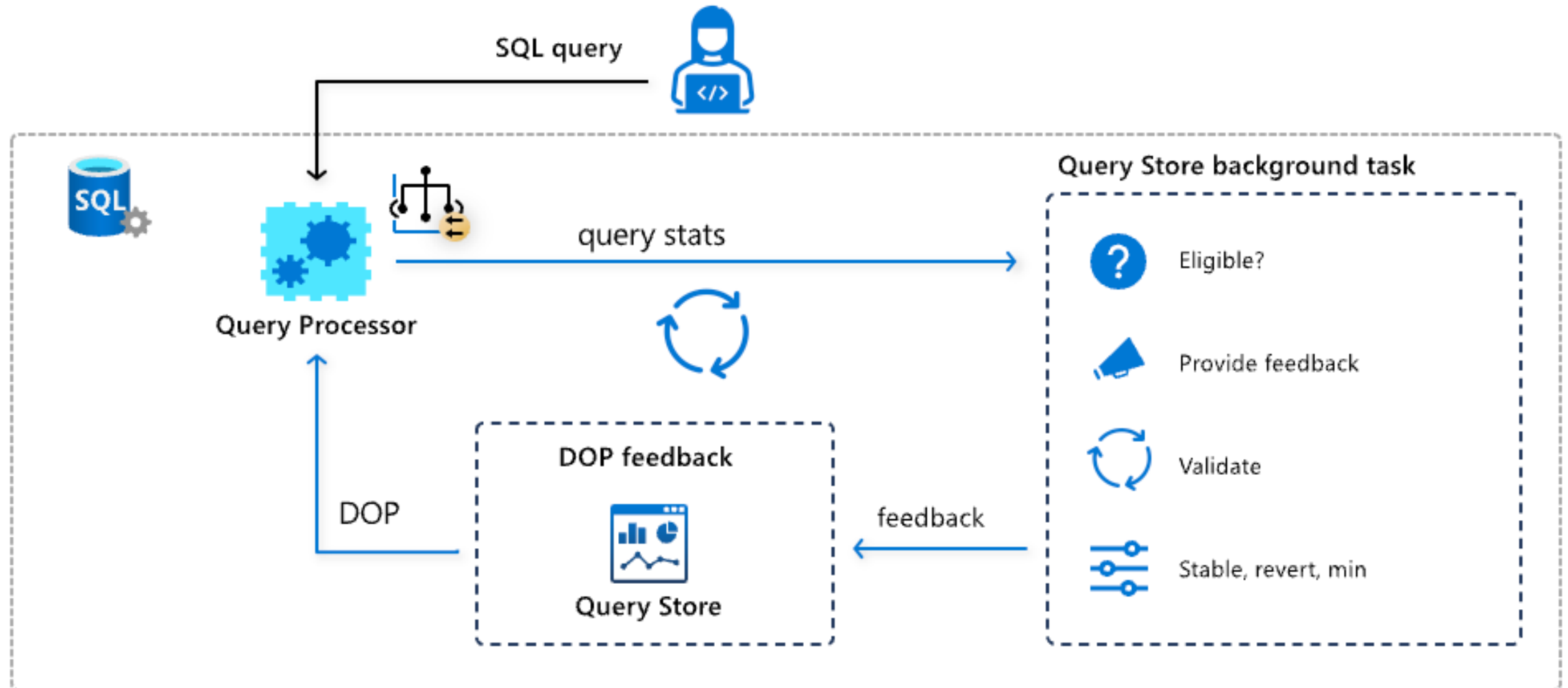
Query store is required

Plan regression = last known good DOP

Honors Plan forcing



DOP Feedback Architecture



DOP Feedback



Won't rotate up unless reverting

DOP of 2 is the minimum value

Query Hints will be the new ceiling limit if > 2

DOP Feedback is Replica aware





DEMO

Feature Availability



Feature	Enterprise	Standard
Table Variable Deferred Compilation	Yes	Yes
Memory Grant Feedback	Yes	No
Parameter Sensitive Plan	Yes	Yes
Degrees of Parallelism Feedback*	Yes	No

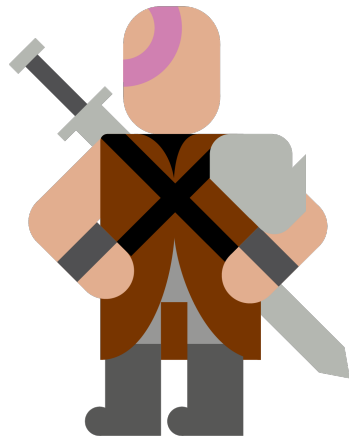
Summary



Performance Game Changer

Increased flexibility

Reminder: Compatibility Level





BYE BYE BAD ANTI-PATTERNS!!

**Feedback
always
appreciated**



John Morehouse

Denny Cherry & Associates Consulting



DCAC*



john@dcac.com



Sqlrus.com



@SQLRUS



/in/johnmorehouse