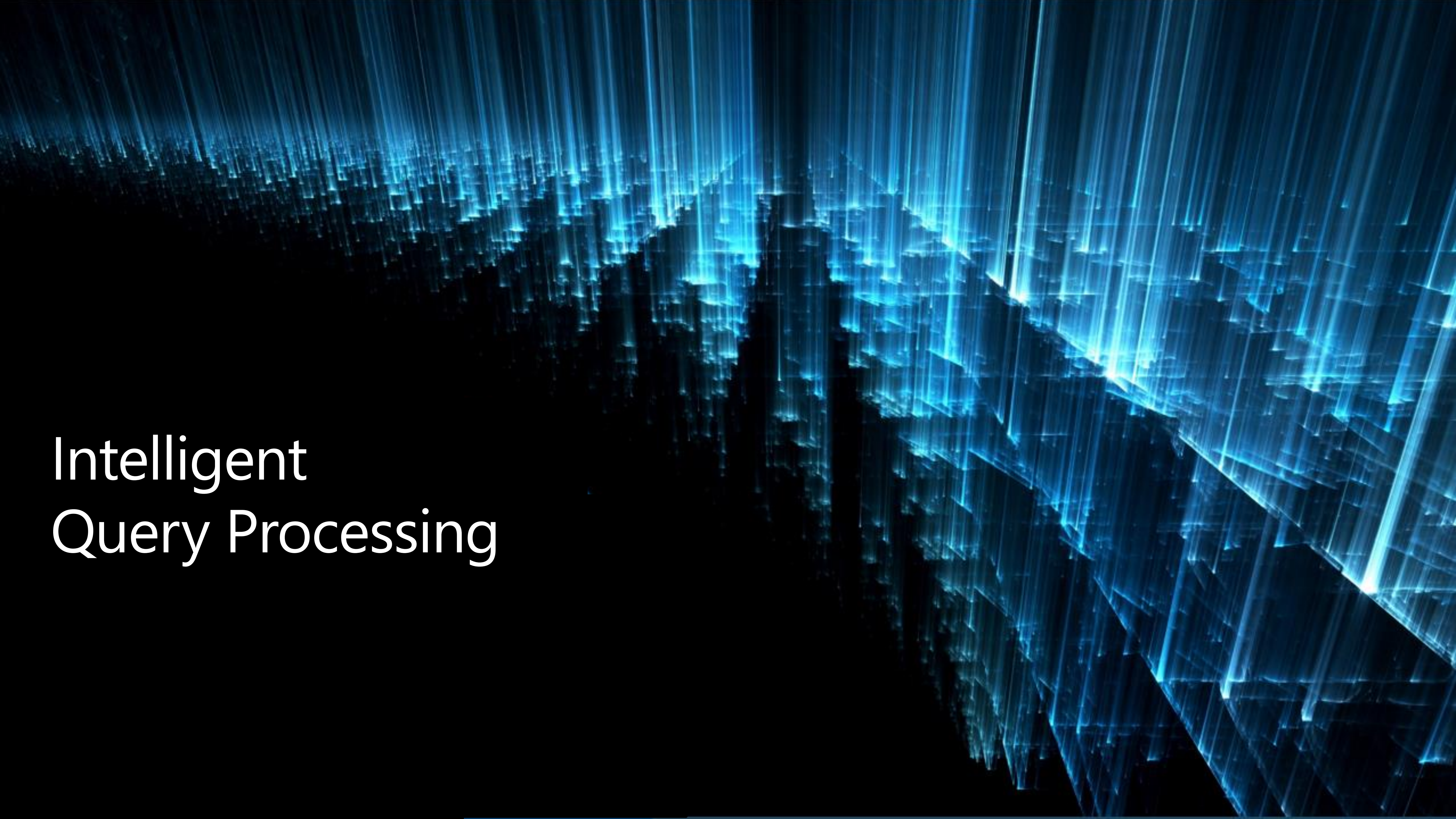
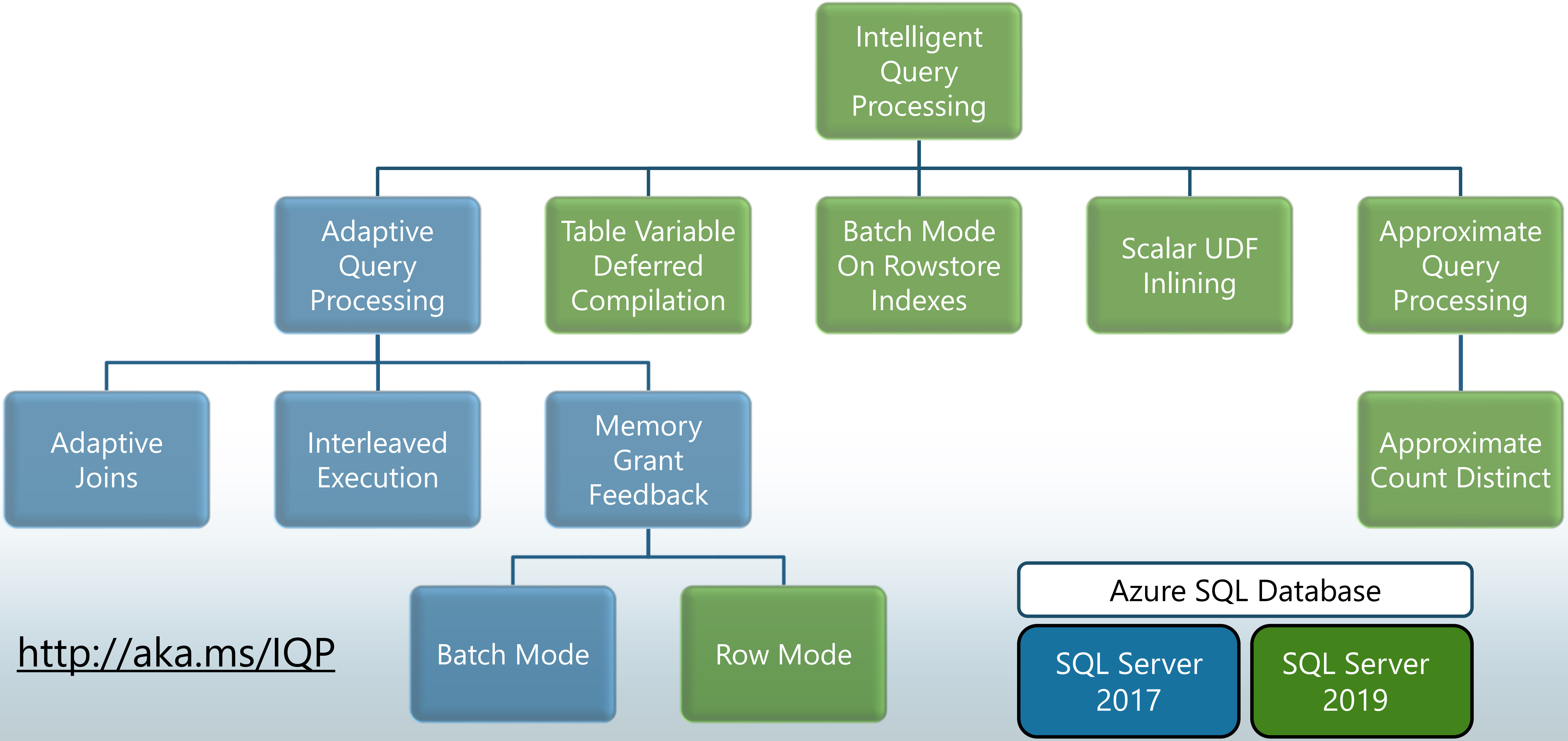
Intelligent Query Processing

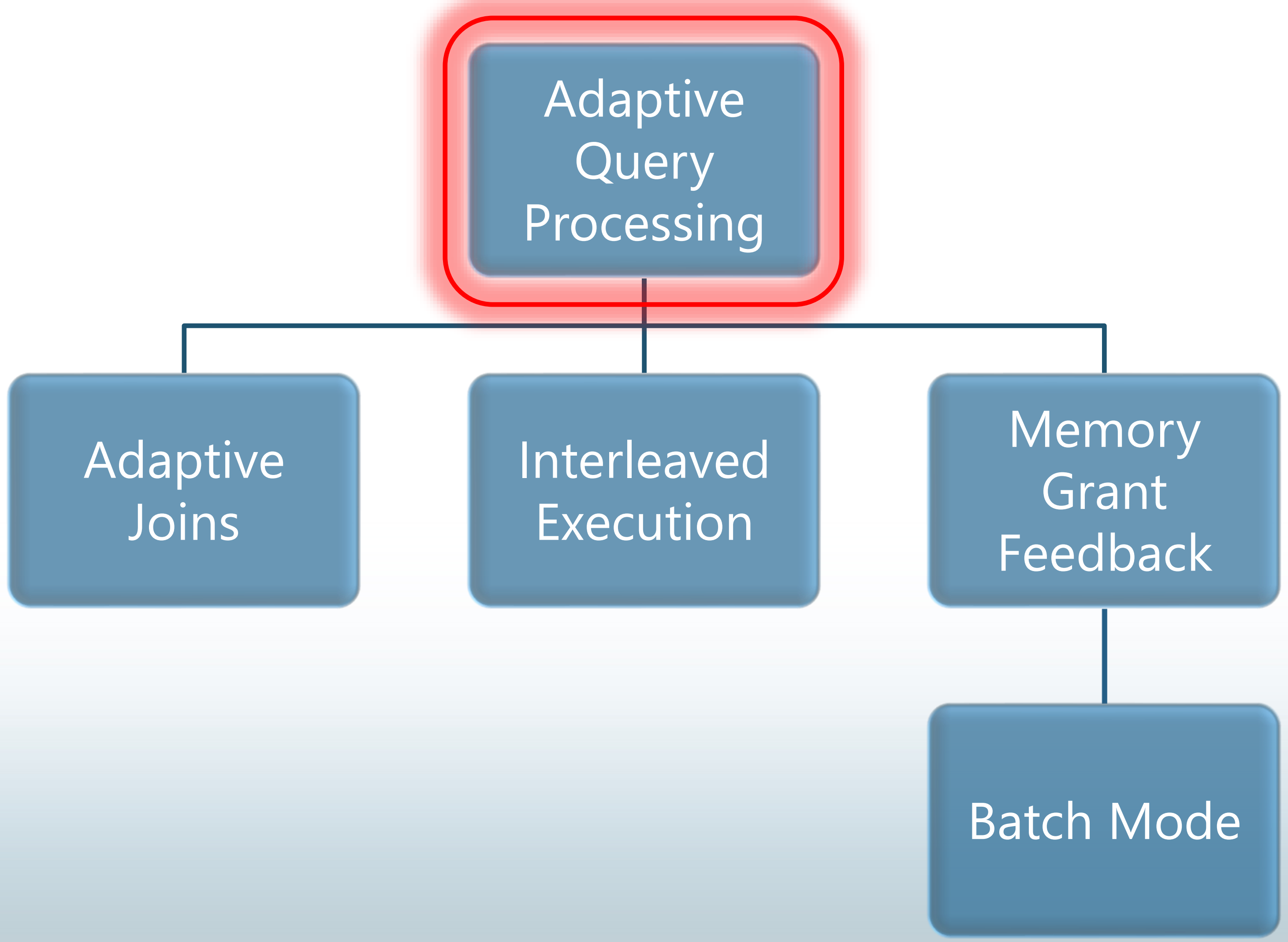


Intelligent Query Processing


Intelligent Query Processing
Adaptive Query Processing
Table Variable Deferred Compilation
Batch Mode On Rowstore Indexes
Scalar UDF Inlining
Approximate Query Processing
Adaptive Joins
Interleaved Execution
Memory Grant Feedback
Approximate Count Distinct
Batch Mode
Row Mode
Azure SQL Database
SQL Server 2017
SQL Server 2019

http://aka.ms/IQP

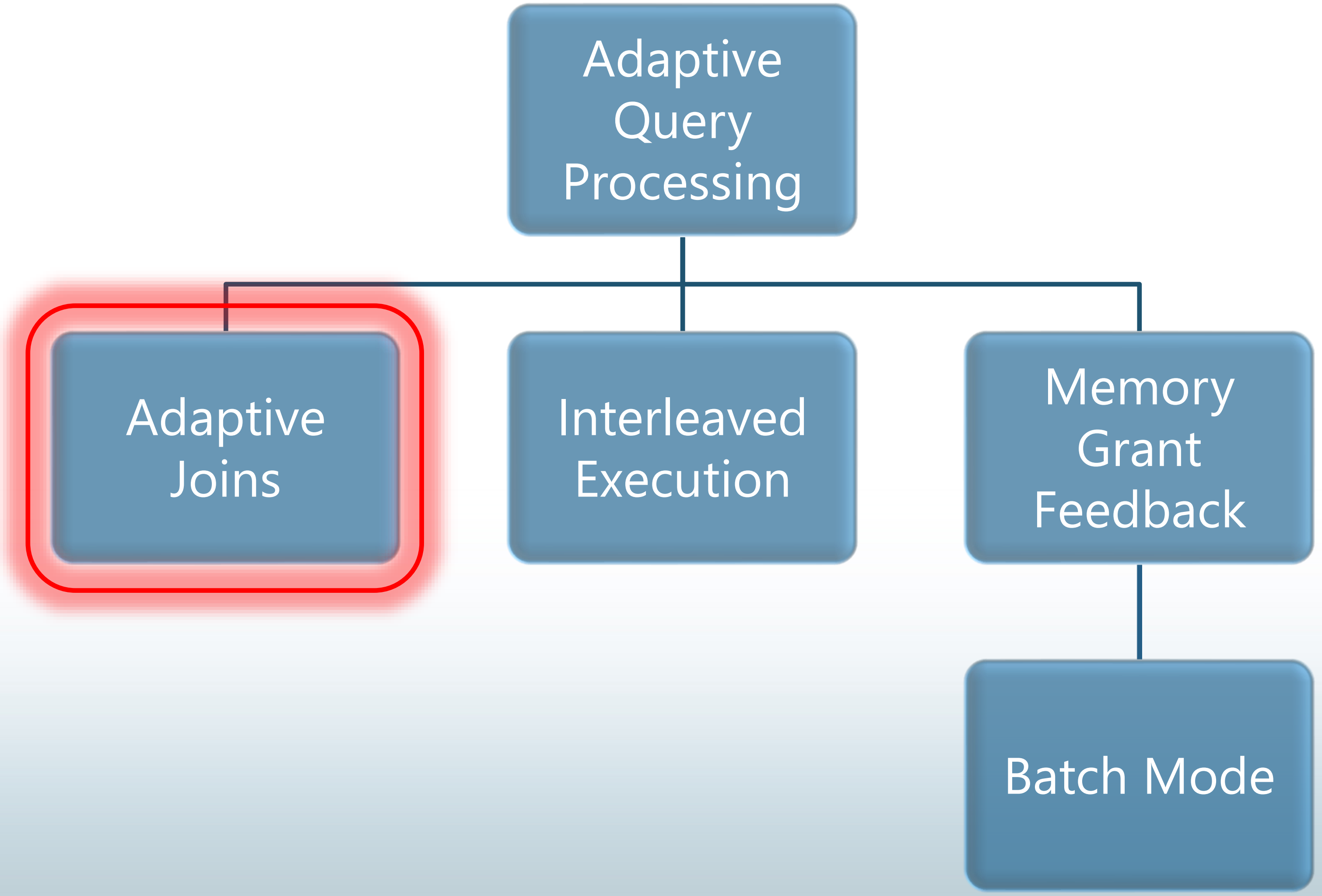
Adaptive Query Processing (2017)



Addresses performance issues related to the cardinality estimation of an execution plan.

These options can provide improved join type selection, row-calculations for Multi-Statement Table-Valued Functions, and memory allocation of row storage.

Batch Mode Adaptive Joins (2017)



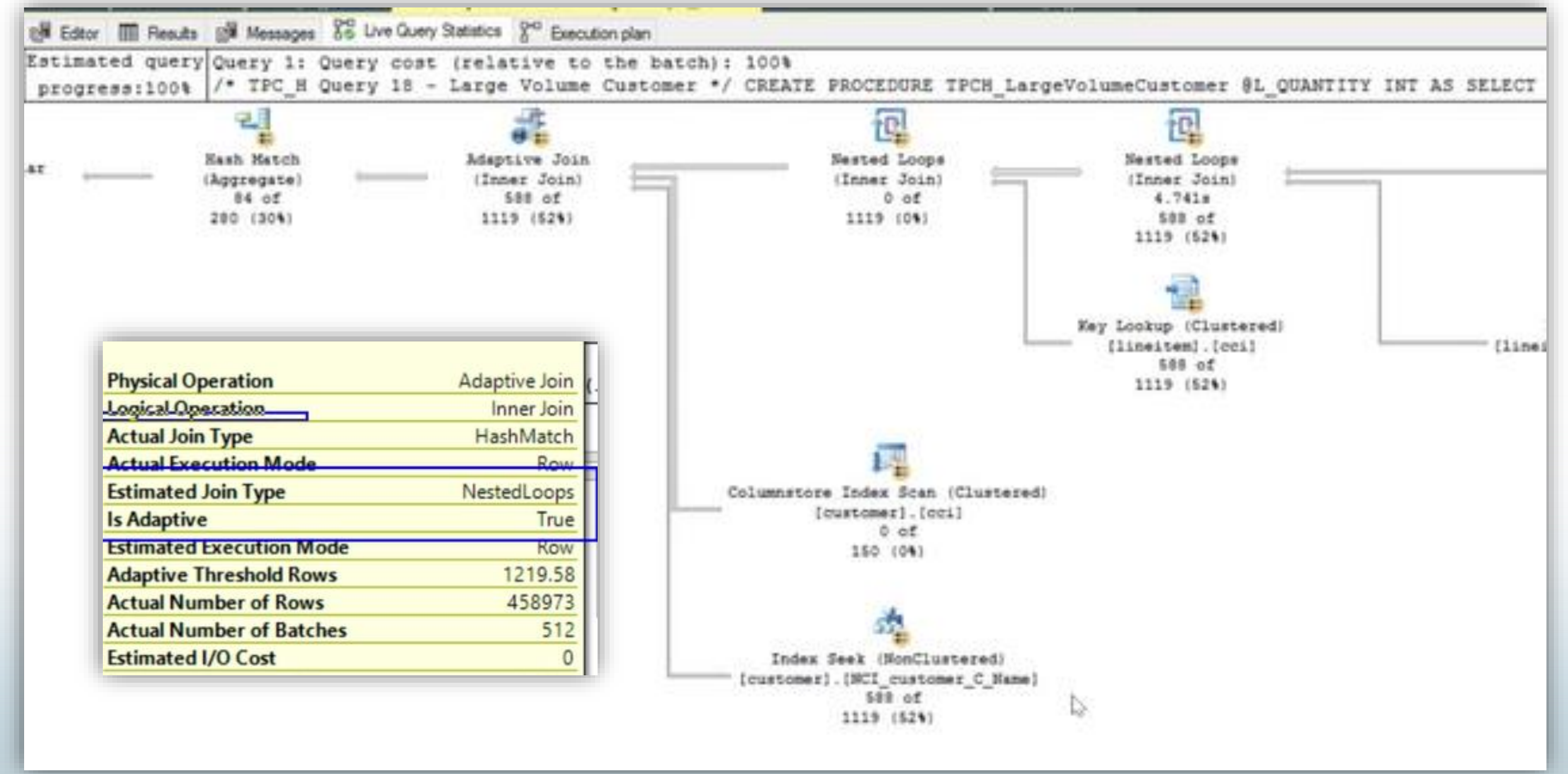
This feature enables the choice of either the Hash or the Nested Loop join type.

Decision is deferred until statement execution.

No need to use join hints in queries.

Batch Mode Adaptive Joins (2017)

Adaptive Joins



```
Estimated query   Query 1: Query cost (relative to the batch): 100%
 progress:100%    /* TPC_H Query 18 - Large Volume Customer */ CREATE PROCEDURE TPCH_LargeVolumeCustomer @L_QUANTITY INT AS SELECT
```

Hash Match (Aggregate) 84 of 280 (30%)

Adaptive Join (Inner Join) 588 of 1119 (52%)

Nested Loops (Inner Join) 0 of 1119 (0%)

Nested Loops (Inner Join) 4.741s 588 of 1119 (52%)

Key Lookup (Clustered) [lineitem].[cci] 588 of 1119 (52%)

Columnstore Index Scan (Clustered) [customer].[cci] 0 of 150 (0%)

Index Seek (NonClustered) [customer].[NCI_customer_C_Name] 588 of 1119 (52%)

Physical Operation	Adaptive Join
Logical Operation	Inner Join
Actual Join Type	HashMatch
Actual Execution Mode	Row
Estimated Join Type	NestedLoops
Is Adaptive	True
Estimated Execution Mode	Row
Adaptive Threshold Rows	1219.58
Actual Number of Rows	458973
Actual Number of Batches	512
Estimated I/O Cost	0

Batch Mode Adaptive Joins (2017)

Adaptive Joins

Enabled by default in Compatibility level 140 or higher.

To disable change compatibility level to 130 or lower

You can enable/disable it at the database level in SQL Server 2017.

```
ALTER DATABASE SCOPED CONFIGURATION SET DISABLE_BATCH_MODE_ADAPTIVE_JOINS = ON|OFF;
```

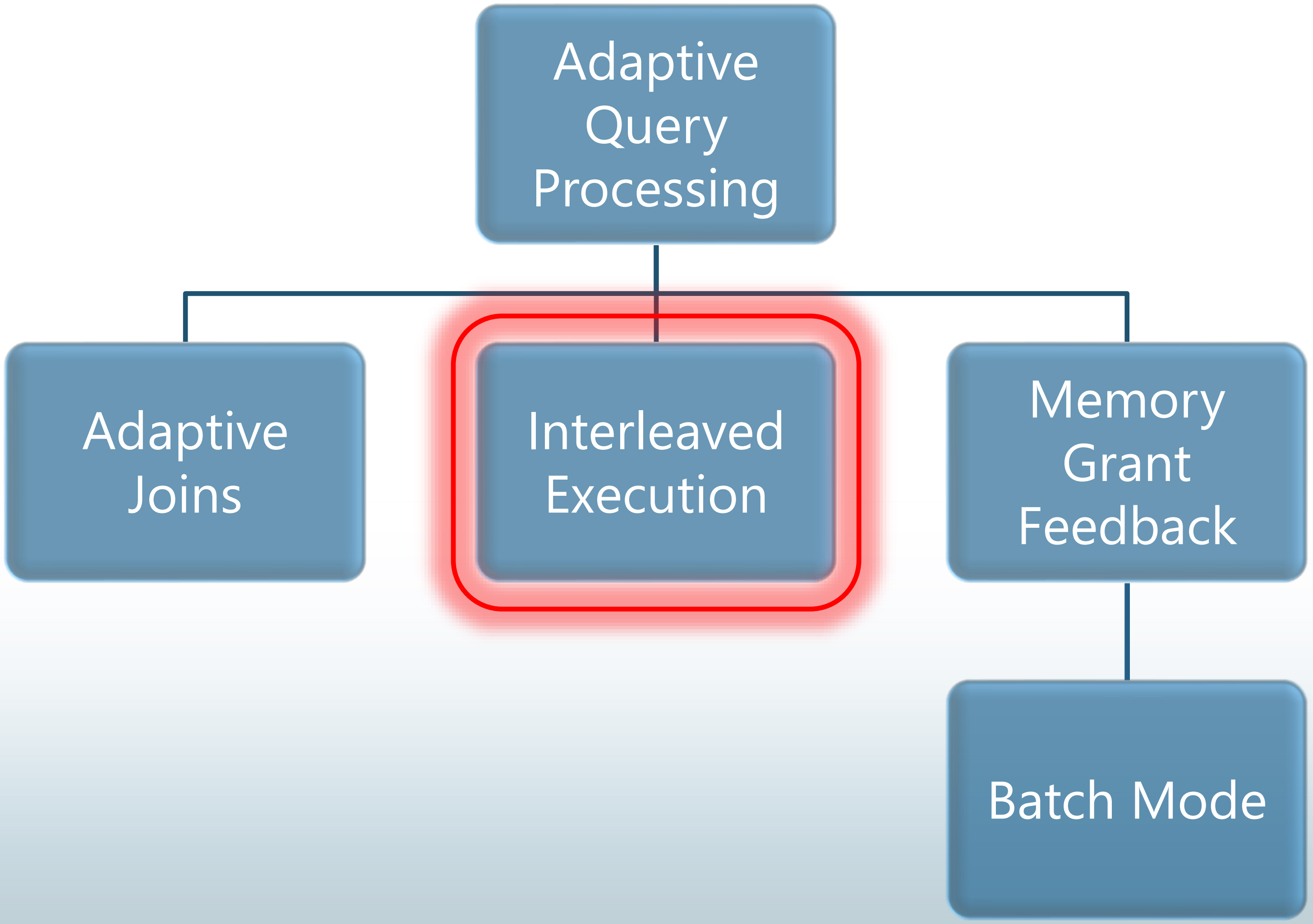
Azure SQL Database, SQL Server 2019 and higher

```
ALTER DATABASE SCOPED CONFIGURATION SET BATCH_MODE_ADAPTIVE_JOINS = ON|OFF;
```

You can disable it at statement scope if necessary.

```
<statement>
OPTION (USE HINT('DISABLE_BATCH_MODE_ADAPTIVE_JOINS'));
```

Interleaved Execution (2017)



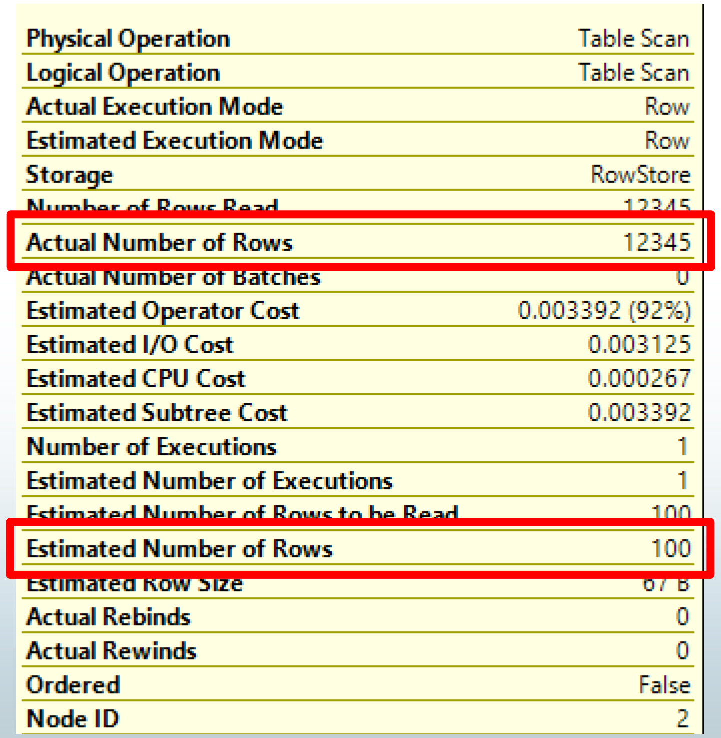
Previously, when a Multi-Statement Table-Valued Function was executed, it used a fixed row estimate of 100 rows.

Now execution is paused so a better cardinality estimate can be captured.

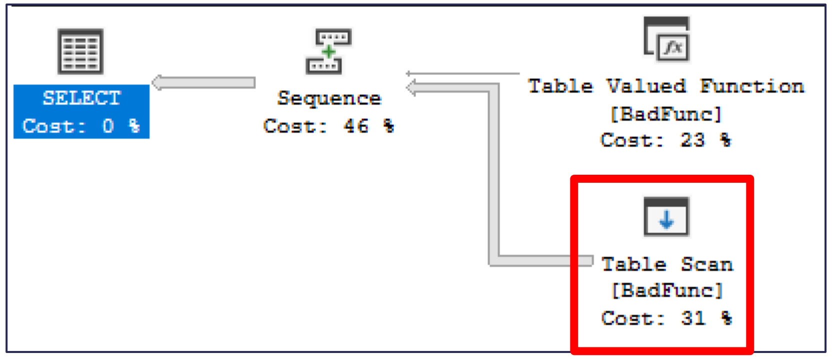
Interleaved Execution (2017)

Interleaved Execution

Compatibility Level 120/130

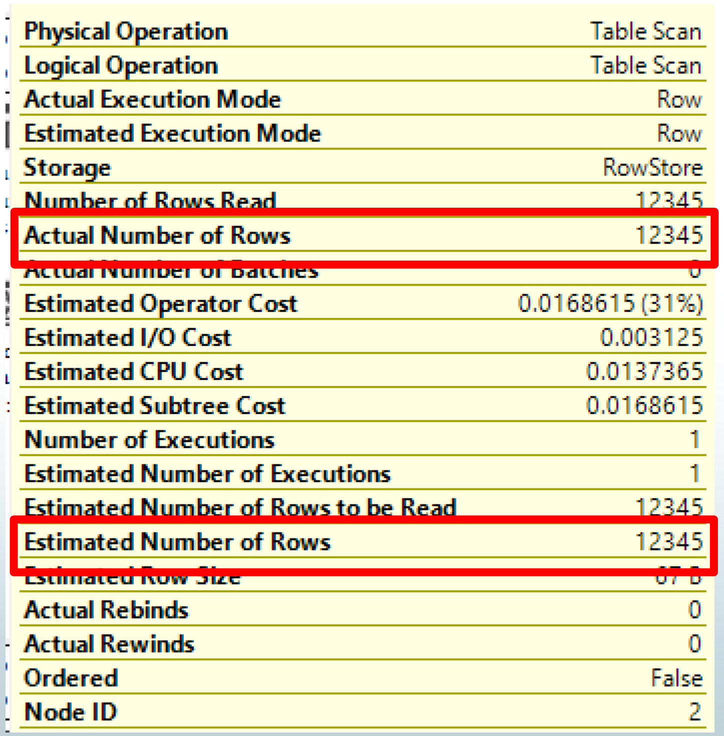


Physical Operation	Table Scan
Logical Operation	Table Scan
Actual Execution Mode	Row
Estimated Execution Mode	Row
Storage	RowStore
Number of Rows Read	12345
Actual Number of Rows	12345
Actual Number of Batches	0
Estimated Operator Cost	0.003392 (92%)
Estimated I/O Cost	0.003125
Estimated CPU Cost	0.000267
Estimated Subtree Cost	0.003392
Number of Executions	1
Estimated Number of Executions	1
Estimated Number of Rows to be Read	100
Estimated Number of Rows	100
Estimated Row Size	67 B
Actual Rebinds	0
Actual Rewinds	0
Ordered	False
Node ID	2



During optimization if SQL Server encounter a read-only multi-statement table-valued function (MSTVF), it will pause optimization, execute the applicable subtree, capture accurate cardinality estimates, and then resume optimization for downstream operations.

Compatibility Level 140 or higher



Physical Operation	Table Scan
Logical Operation	Table Scan
Actual Execution Mode	Row
Estimated Execution Mode	Row
Storage	RowStore
Number of Rows Read	12345
Actual Number of Rows	12345
Actual Number of Batches	0
Estimated Operator Cost	0.0168615 (31%)
Estimated I/O Cost	0.003125
Estimated CPU Cost	0.0137365
Estimated Subtree Cost	0.0168615
Number of Executions	1
Estimated Number of Executions	1
Estimated Number of Rows to be Read	12345
Estimated Number of Rows	12345
Estimated Row Size	67 B
Actual Rebinds	0
Actual Rewinds	0
Ordered	False
Node ID	2

Interleaved Execution (2017)

Interleaved Execution

Enabled by default in Compatibility level 140 or higher.

To disable change compatibility level to 130 or lower

You can enable/disable it at the database level in SQL Server 2017.

```
ALTER DATABASE SCOPED CONFIGURATION SET DISABLE_INTERLEAVED_EXECUTION_TVF = ON|OFF;
```

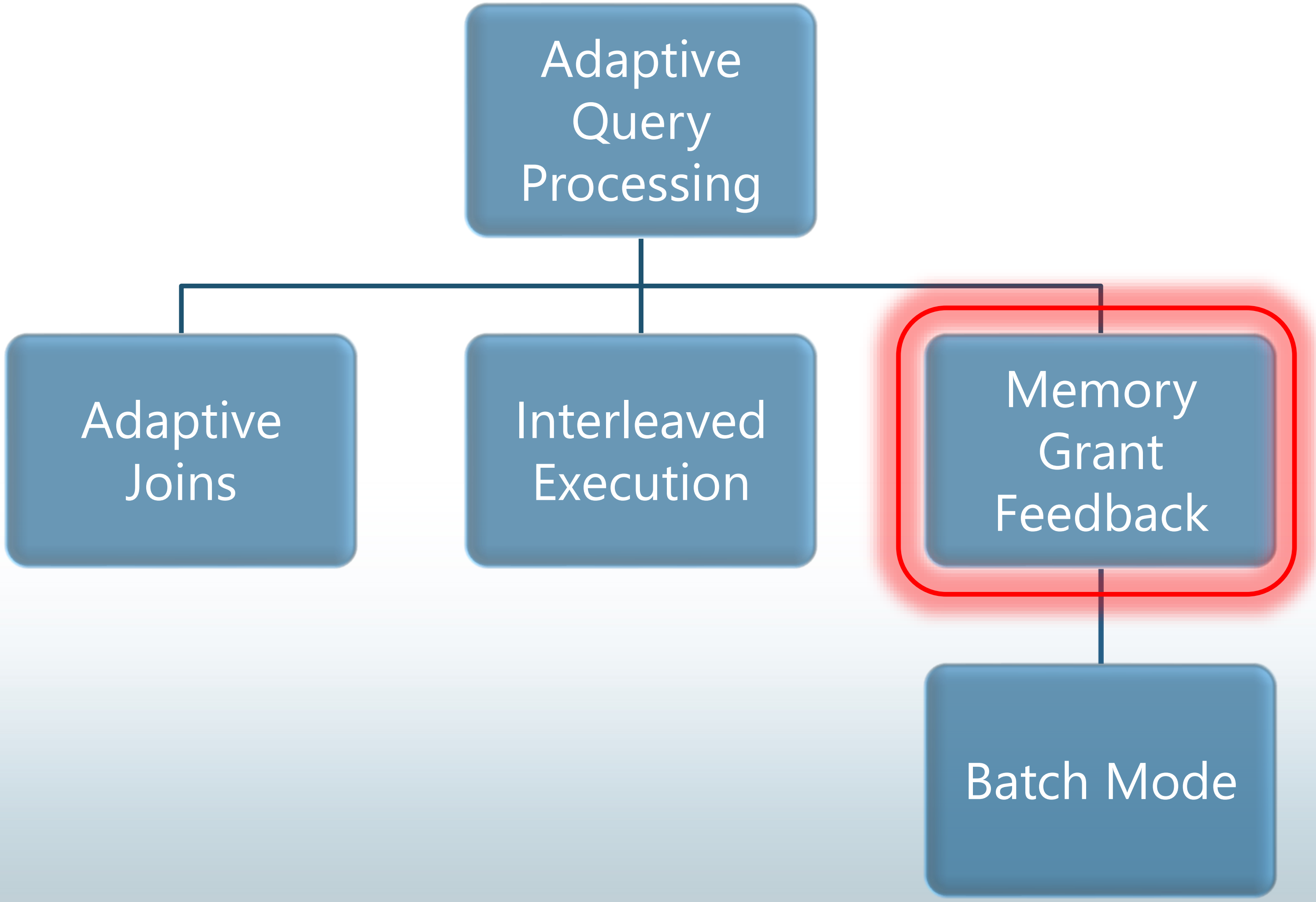
Azure SQL Database, SQL Server 2019 and higher

```
ALTER DATABASE SCOPED CONFIGURATION SET INTERLEAVED_EXECUTION_TVF = ON|OFF;
```

You can disable it at statement scope if necessary.

```
<statement>
OPTION (USE HINT('DISABLE_INTERLEAVED_EXECUTION_TVF'));
```

Batch Mode Memory Grant Feedback (2017)



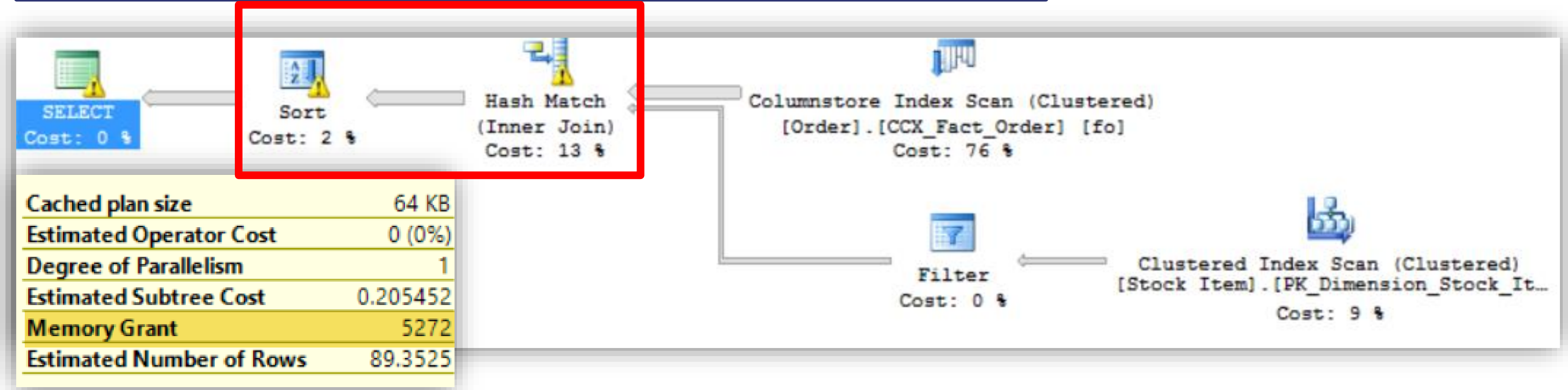
When compiling an execution plan, the query engine estimates how much memory is needed to store rows during join and sort operations.

Too much memory allocation may impact performance of other operations. Not enough will cause a spill over to disk.

This feature recalculates memory on first execution and updates the cached plan.

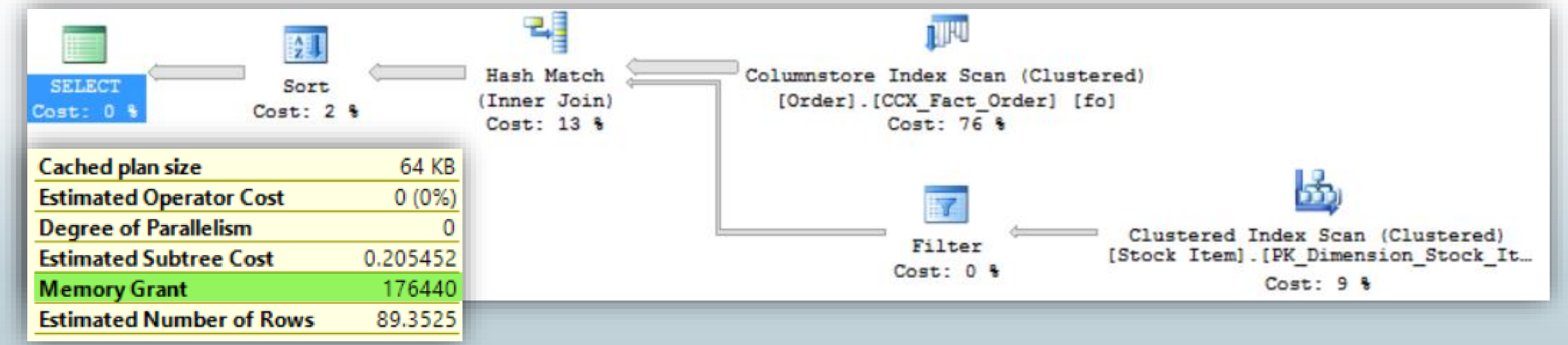
Batch Mode Memory Grant Feedback (2017)

First Execution (Spills detected; feedback generated)


SELECT
Cost: 0 %
Sort
Cost: 2 %
Hash Match
(Inner Join)
Cost: 13 %
Columnstore Index Scan (Clustered)
[Order].[CCX_Fact_Order] [fo]
Cost: 76 %
Filter
Cost: 0 %
Clustered Index Scan (Clustered)
[Stock Item].[PK_Dimension_Stock_It…
Cost: 9 %

Cached plan size	64 KB
Estimated Operator Cost	0 (0%)
Degree of Parallelism	1
Estimated Subtree Cost	0.205452
Memory Grant	5272
Estimated Number of Rows	89.3525

Second Execution (Memory grant adjusted)


SELECT
Cost: 0 %
Sort
Cost: 2 %
Hash Match
(Inner Join)
Cost: 13 %
Columnstore Index Scan (Clustered)
[Order].[CCX_Fact_Order] [fo]
Cost: 76 %
Filter
Cost: 0 %
Clustered Index Scan (Clustered)
[Stock Item].[PK_Dimension_Stock_It…
Cost: 9 %

Cached plan size	64 KB
Estimated Operator Cost	0 (0%)
Degree of Parallelism	0
Estimated Subtree Cost	0.205452
Memory Grant	176440
Estimated Number of Rows	89.3525

Memory Grant Feedback (Batch Mode)

Batch Mode Memory Grant Feedback (2017)

Memory Grant Feedback (Batch Mode)

Enabled by default in Compatibility level 140 or higher.

To disable change compatibility level to 130 or lower

You can enable/disable it at the database level in SQL Server 2017.

```
ALTER DATABASE SCOPED CONFIGURATION SET DISABLE_BATCH_MODE_MEMORY_GRANT_FEEDBACK=ON|OFF;
```

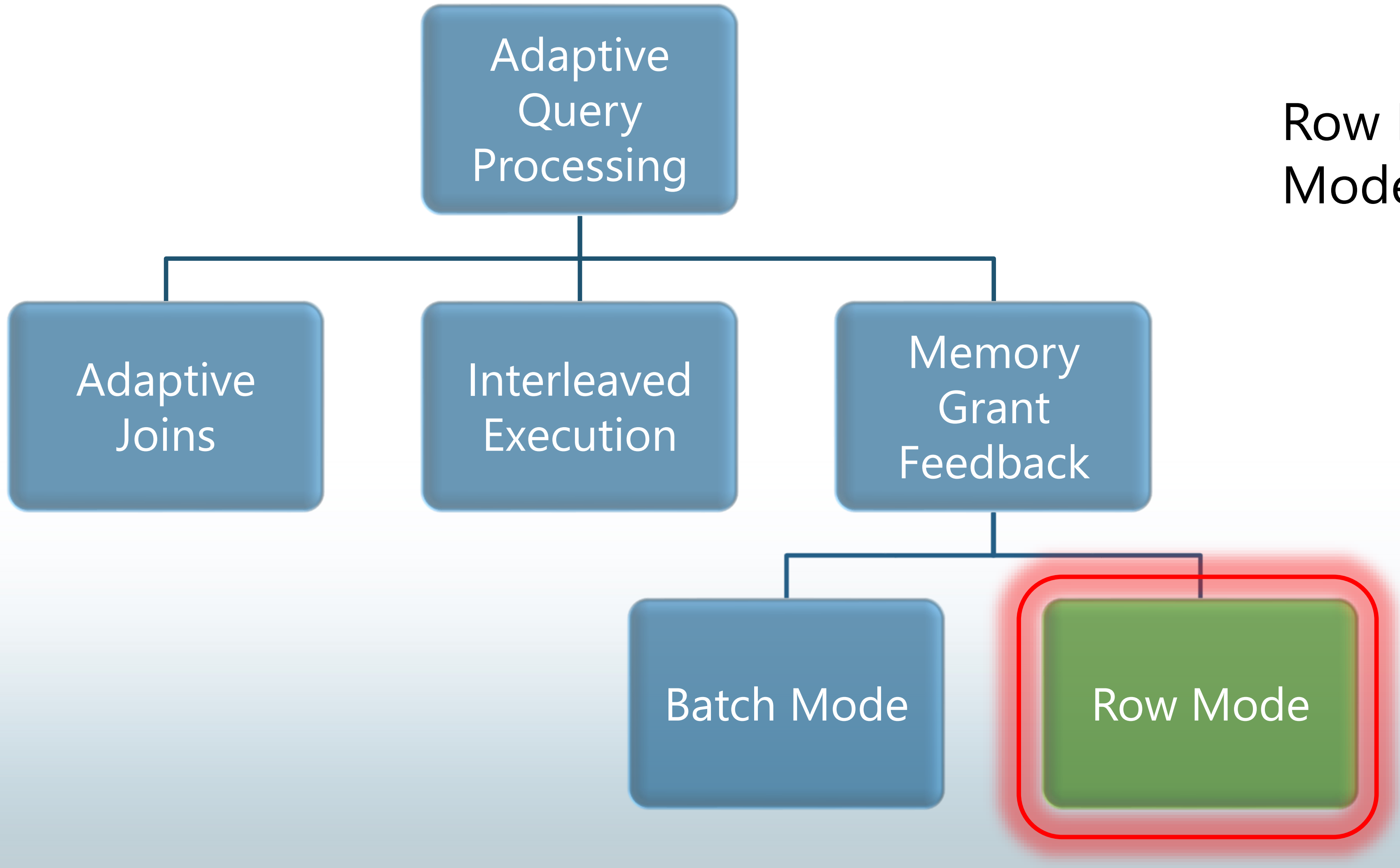
Azure SQL Database, SQL Server 2019 and higher

```
ALTER DATABASE SCOPED CONFIGURATION SET BATCH_MODE_MEMORY_GRANT_FEEDBACK = ON|OFF;
```

You can disable it at statement scope if necessary.

```
<statement>
OPTION (USE HINT ('DISABLE_BATCH_MODE_MEMORY_GRANT_FEEDBACK'))
```

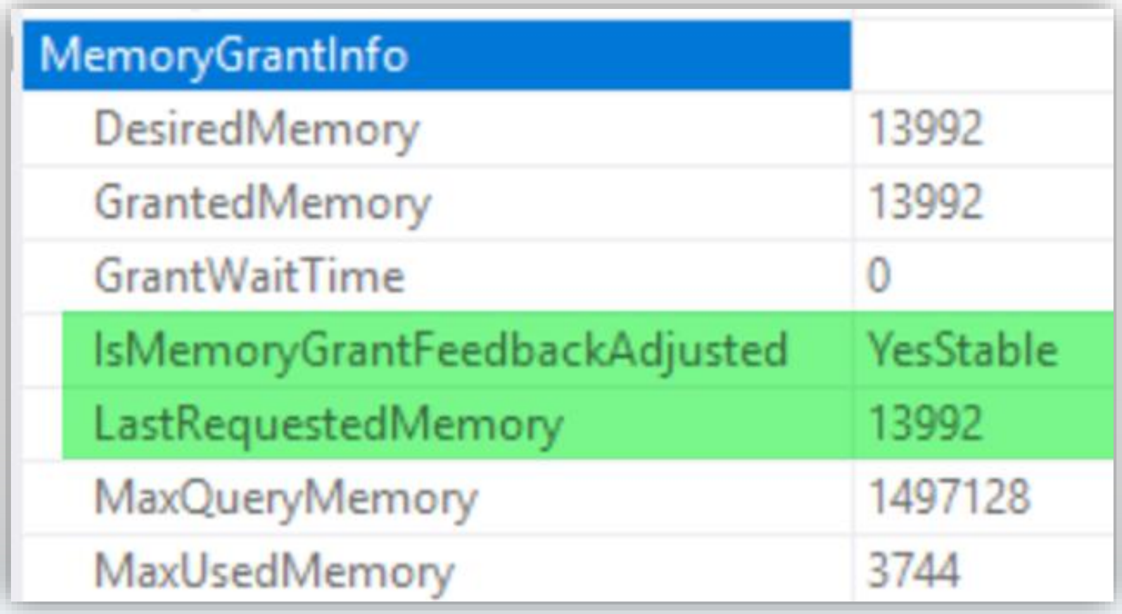
Row Mode Memory Grant Feedback



Row Mode is just like Batch Mode, but different.

Row Mode Memory Grant Feedback

Expands on the batch mode memory grant feedback feature by also adjusting memory grant sizes for row mode operators.



MemoryGrantInfo	
DesiredMemory	13992
GrantedMemory	13992
GrantWaitTime	0
IsMemoryGrantFeedbackAdjusted	YesStable
LastRequestedMemory	13992
MaxQueryMemory	1497128
MaxUsedMemory	3744

Memory Grant Feedback (Row Mode)

Two new query plan attributes will be shown for actual post-execution plans.

Row Mode Memory Grant Feedback

Enabled by default in Compatibility level 150 or higher.

To disable change compatibility level to 140 or lower

Memory Grant Feedback (Row Mode)

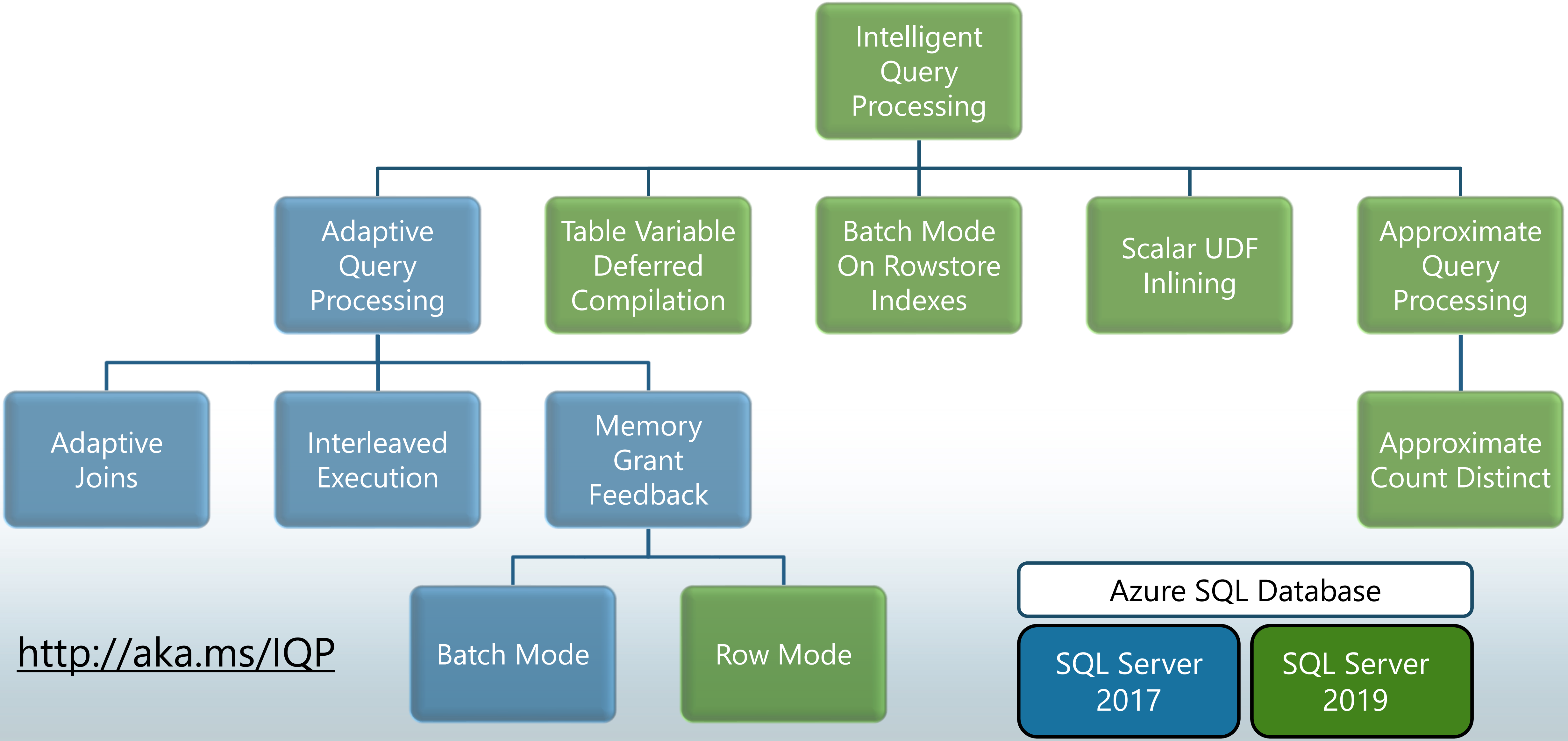
You can enable/disable it at the database level

```
ALTER DATABASE SCOPED CONFIGURATION SET ROW_MODE_MEMORY_GRANT_FEEDBACK = ON|OFF;
```

You can disable it at statement scope if necessary.

```
<statement>
OPTION (USE HINT ('DISABLE_ROW_MODE_MEMORY_GRANT_FEEDBACK'));
```

Intelligent Query Processing


Intelligent Query Processing
Adaptive Query Processing
Table Variable Deferred Compilation
Batch Mode On Rowstore Indexes
Scalar UDF Inlining
Approximate Query Processing
Adaptive Joins
Interleaved Execution
Memory Grant Feedback
Approximate Count Distinct
Batch Mode
Row Mode
Azure SQL Database
SQL Server 2017
SQL Server 2019

http://aka.ms/IQP

Intelligent Query Processing (2019)

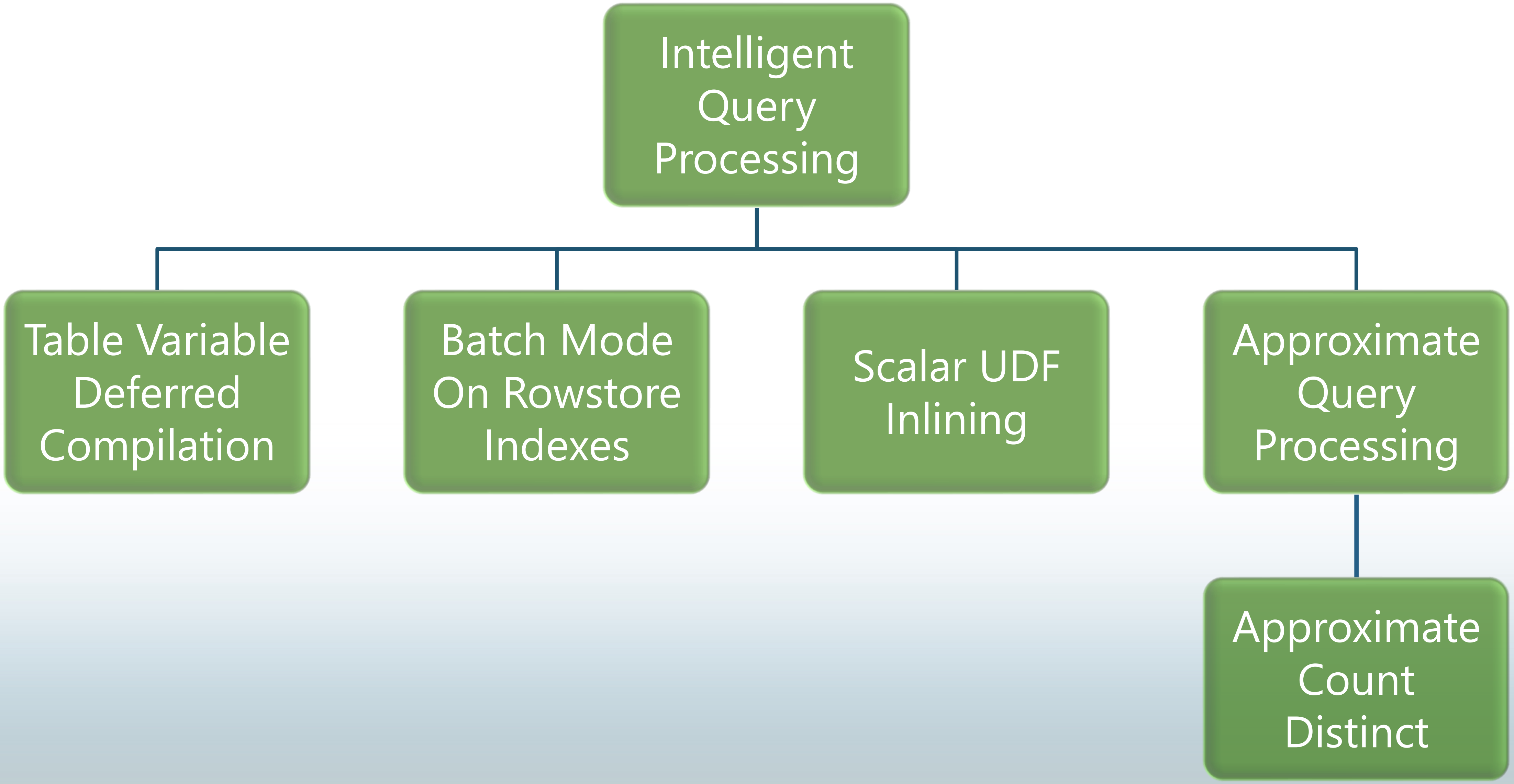
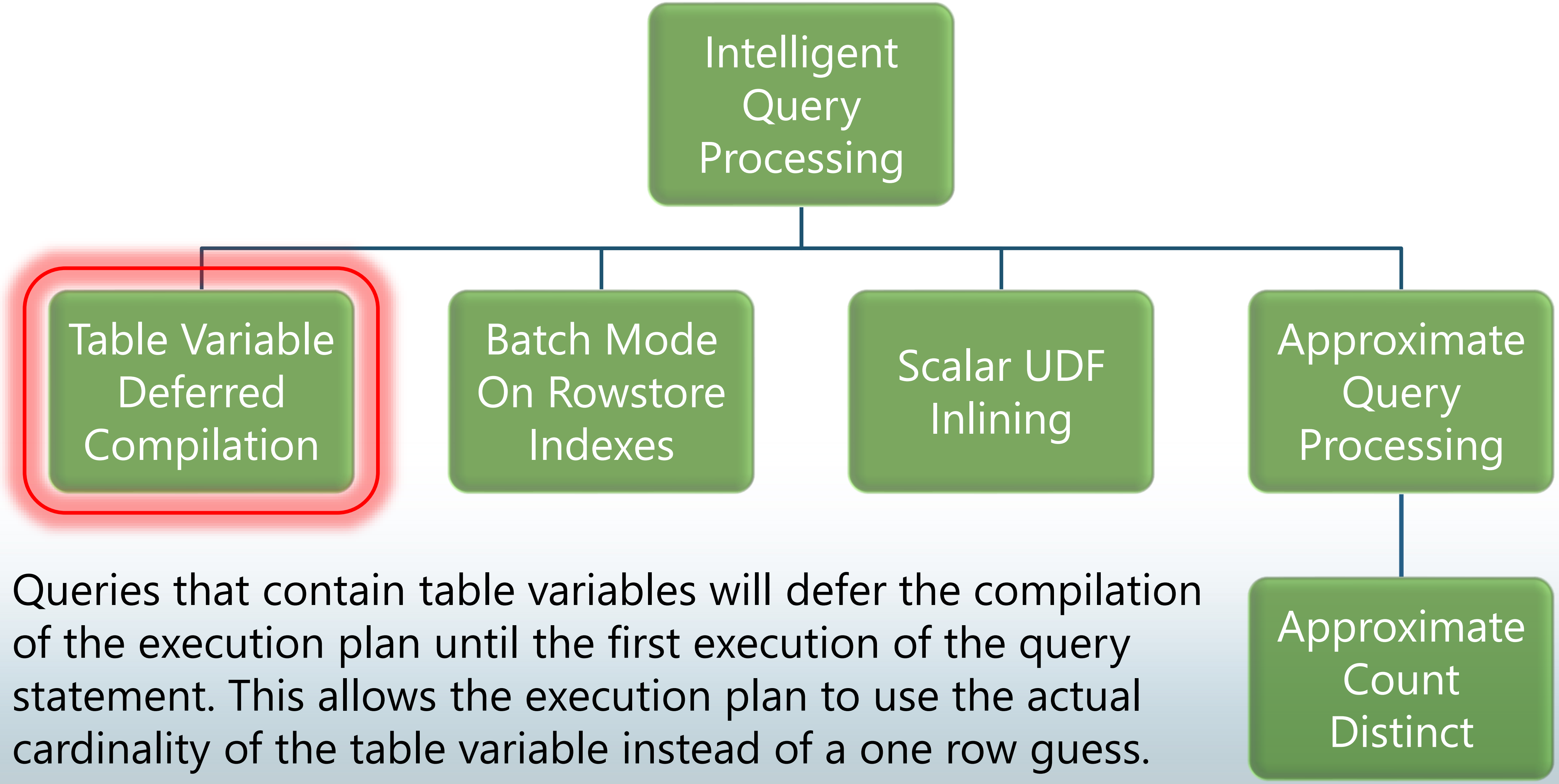

Intelligent Query Processing
Table Variable Deferred Compilation
Batch Mode On Rowstore Indexes
Scalar UDF Inlining
Approximate Query Processing
Approximate Count Distinct

Table Variable Deferred Compilation


Intelligent Query Processing
Table Variable Deferred Compilation
Batch Mode On Rowstore Indexes
Scalar UDF Inlining
Approximate Query Processing
Approximate Count Distinct

Queries that contain table variables will defer the compilation of the execution plan until the first execution of the query statement. This allows the execution plan to use the actual cardinality of the table variable instead of a one row guess.

Table Variable Deferred Compilation

Table Variable Deferred Compilation

Enabled by default in Compatibility level 150 or higher.

To disable change compatibility level to 140 or lower

You can enable/disable it at the database level

```
ALTER DATABASE SCOPED CONFIGURATION SET DEFERRED_COMPILATION_TV = ON|OFF;
```

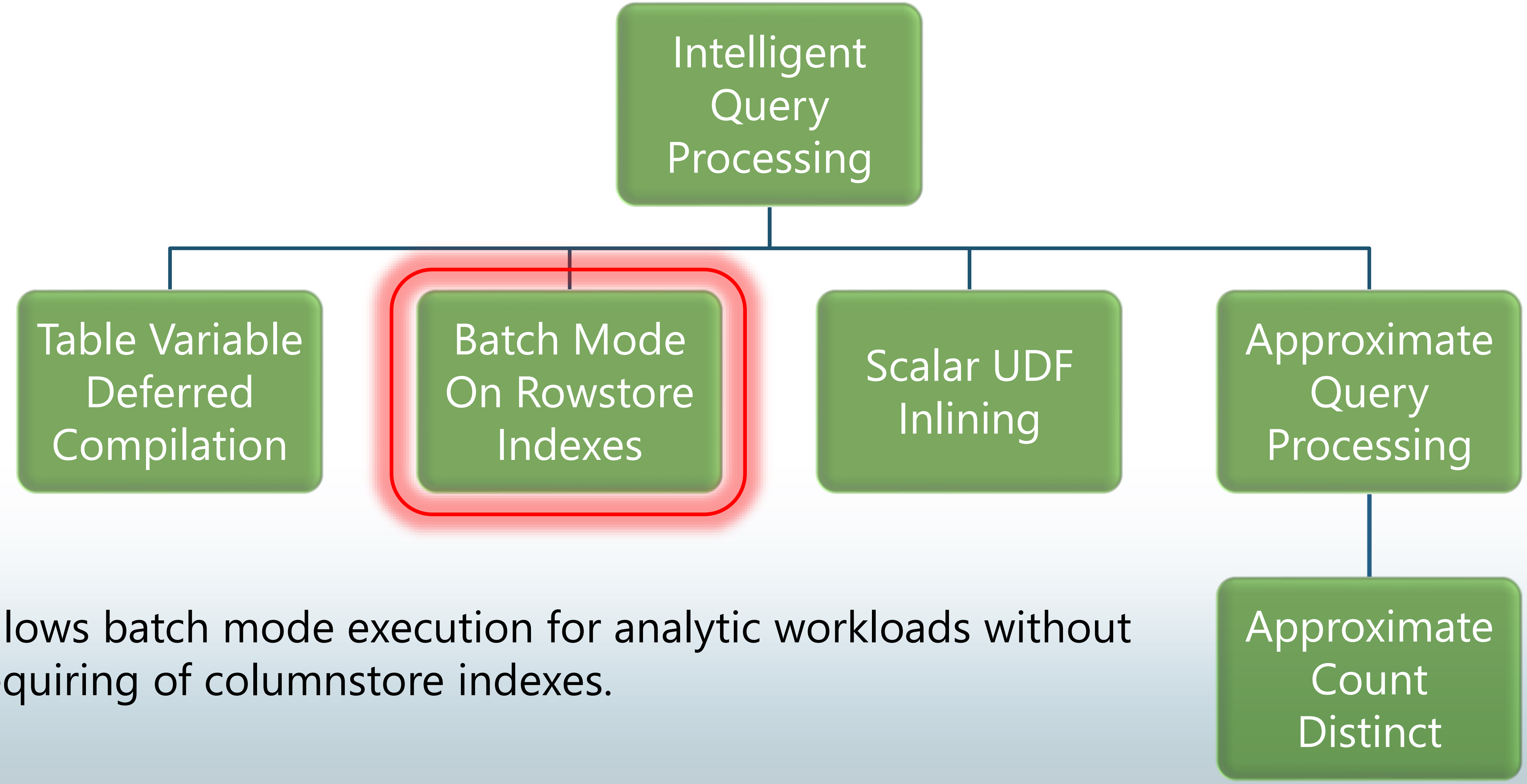
You can disable it at statement scope if necessary.

```
<table variable declaration>

<data inserted into table variable >

<statement that uses the table variable>
OPTION (USE HINT('DISABLE_DEFERRED_COMPILATION_TV'));
```

Batch Mode on Rowstore Indexes


Intelligent Query Processing
Table Variable Deferred Compilation
Batch Mode On Rowstore Indexes
Scalar UDF Inlining
Approximate Query Processing
Approximate Count Distinct

Allows batch mode execution for analytic workloads without requiring of columnstore indexes.

Batch Mode on Rowstore Indexes

Batch Mode On Rowstore Indexes

Enabled by default in Compatibility level 150 or higher.

To disable change compatibility level to 140 or lower

You can enable/disable it at the database level

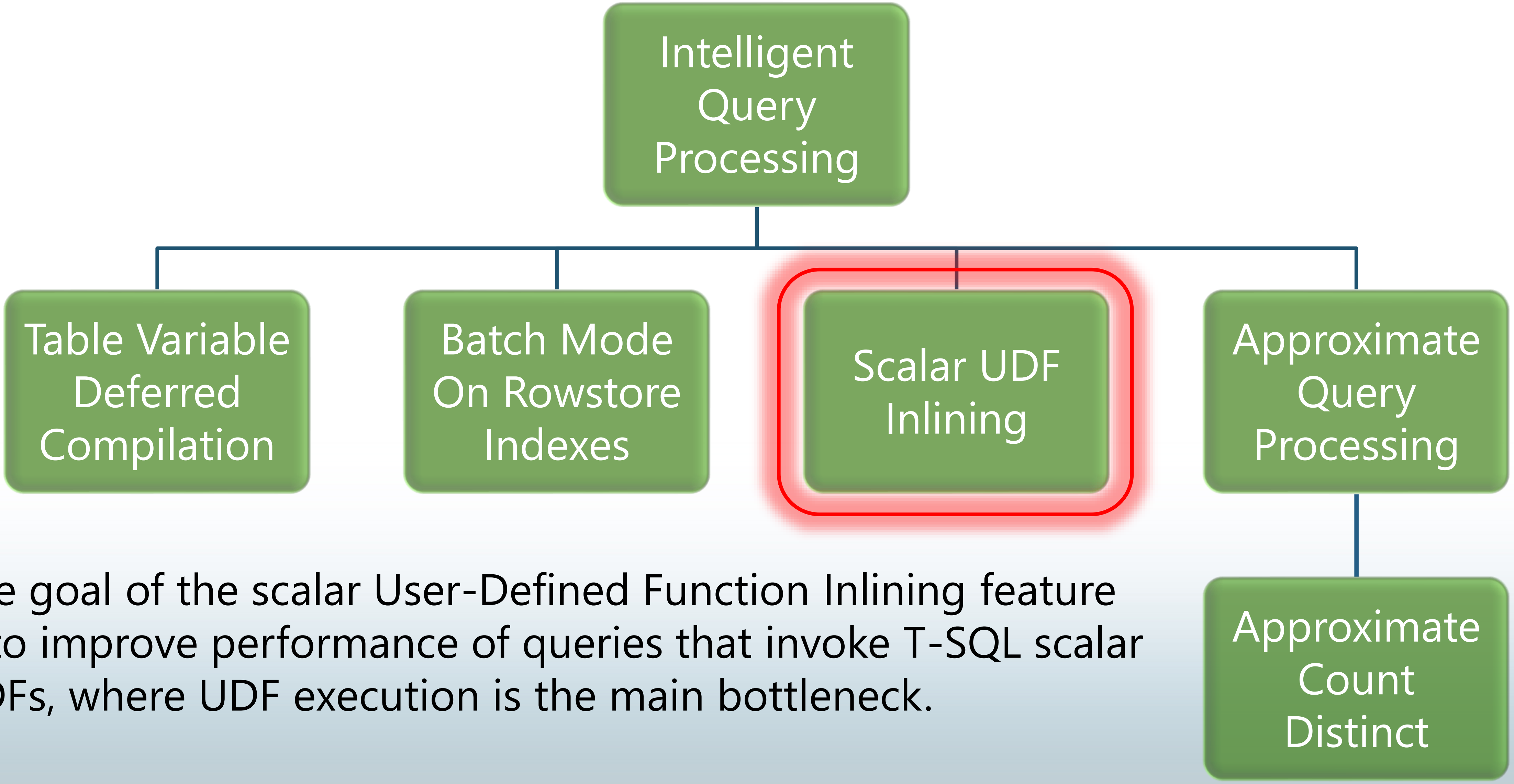
```
ALTER DATABASE SCOPED CONFIGURATION SET BATCH_MODE_ON_ROWSTORE = ON|OFF;
```

You can enable/disable it at statement scope if necessary.

```
<statement>
OPTION(RECOMPILE, USE HINT('ALLOW_BATCH_MODE'));
```

```
<statement>
OPTION(RECOMPILE, USE HINT('DISALLOW_BATCH_MODE'));
```

Scalar User-Defined Function Inlining


Intelligent Query Processing
Table Variable Deferred Compilation
Batch Mode On Rowstore Indexes
Scalar UDF Inlining
Approximate Query Processing
Approximate Count Distinct

The goal of the scalar User-Defined Function Inlining feature is to improve performance of queries that invoke T-SQL scalar UDFs, where UDF execution is the main bottleneck.

Scalar User-Defined Function Inlining

The goal of the scalar User-Defined Function Inlining feature is to improve performance of queries that invoke T-SQL scalar UDFs, where UDF execution is the main bottleneck.

Scalar UDF Inlining

Scalar UDFs are automatically transformed into scalar expressions or scalar subqueries that are substituted in the calling query in place of the UDF operator.

These expressions and subqueries are then optimized. As a result, the query plan will no longer have a user-defined function operator, but its effects will be observed in the plan, like views or inline TVFs.

Scalar User-Defined Function Inlining

Enabled by default in Compatibility level 150 or higher.

To disable change compatibility level to 140 or lower

Scalar UDF Inlining

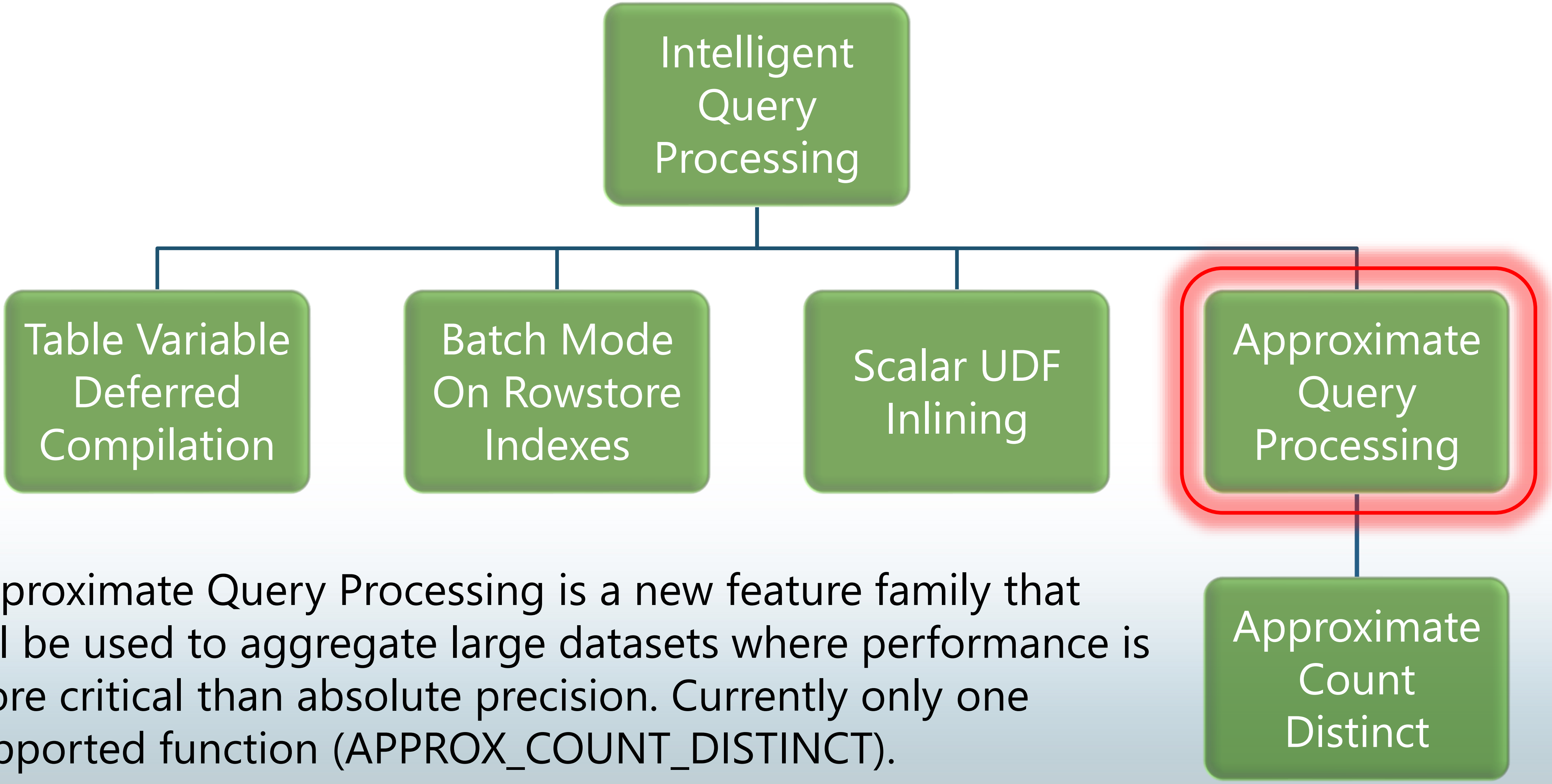
You can enable/disable it at the database level

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

You can disable it at statement scope if necessary.

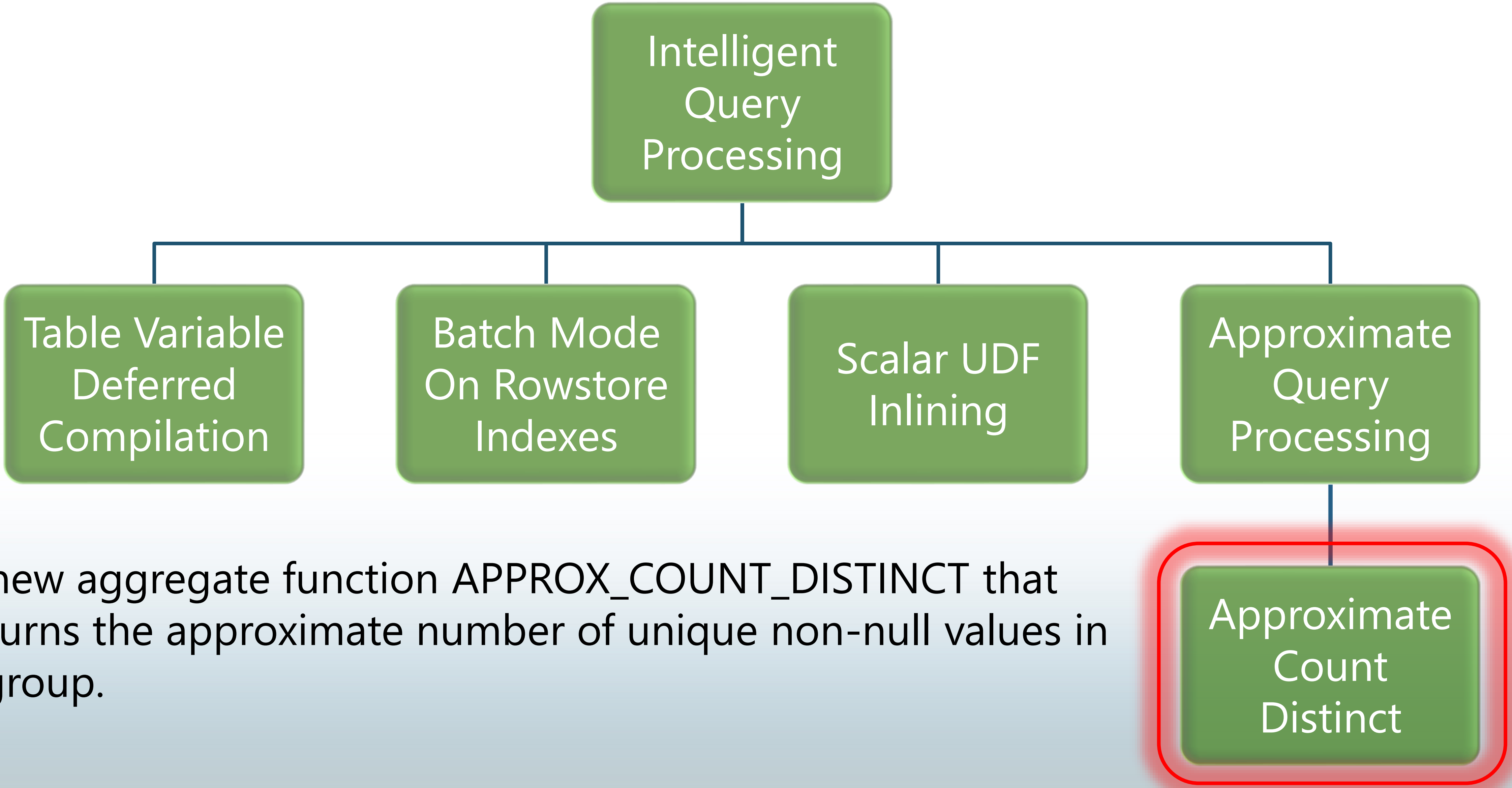
```
<statement>
OPTION (USE HINT('DISABLE_TSQL_SCALAR_UDF_INLINING'));
```

Approximate Query Processing


Intelligent Query Processing
Table Variable Deferred Compilation
Batch Mode On Rowstore Indexes
Scalar UDF Inlining
Approximate Query Processing
Approximate Count Distinct

Approximate Query Processing is a new feature family that will be used to aggregate large datasets where performance is more critical than absolute precision. Currently only one supported function (APPROX_COUNT_DISTINCT).

Approximate Count Distinct


Intelligent Query Processing
Table Variable Deferred Compilation
Batch Mode On Rowstore Indexes
Scalar UDF Inlining
Approximate Query Processing
Approximate Count Distinct

A new aggregate function APPROX_COUNT_DISTINCT that returns the approximate number of unique non-null values in a group.

Approximate Count Distinct

It returns the approximate number of unique non-null values in a group.

It is designed to provide aggregations across large data sets where responsiveness is more critical than absolute precision.

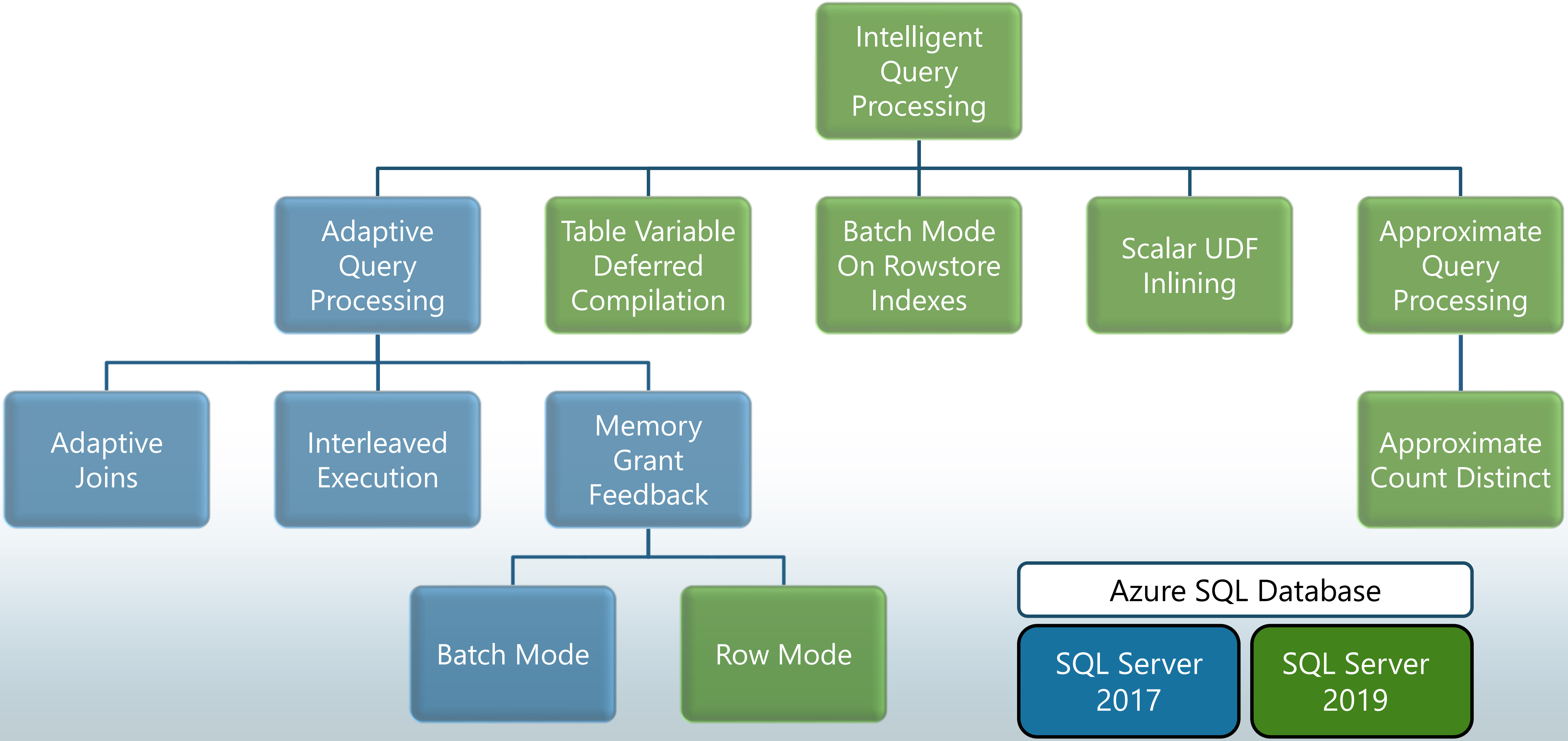
Its implementation guarantees up to a 2% error rate within a 97% probability.

Requires less memory than an exhaustive COUNT DISTINCT operation so it is less likely to spill memory to disk compared to COUNT DISTINCT.

Approximate Count Distinct

```sql
SELECT APPROX_COUNT_DISTINCT(O_OrderKey) AS Approx_Distinct_OrderKey
FROM dbo.Orders;
```

Intelligent Query Processing


Intelligent Query Processing
Adaptive Query Processing
Table Variable Deferred Compilation
Batch Mode On Rowstore Indexes
Scalar UDF Inlining
Approximate Query Processing
Adaptive Joins
Interleaved Execution
Memory Grant Feedback
Approximate Count Distinct
Batch Mode
Row Mode
Azure SQL Database
SQL Server 2017
SQL Server 2019

Questions?