

SQLServerFast.com

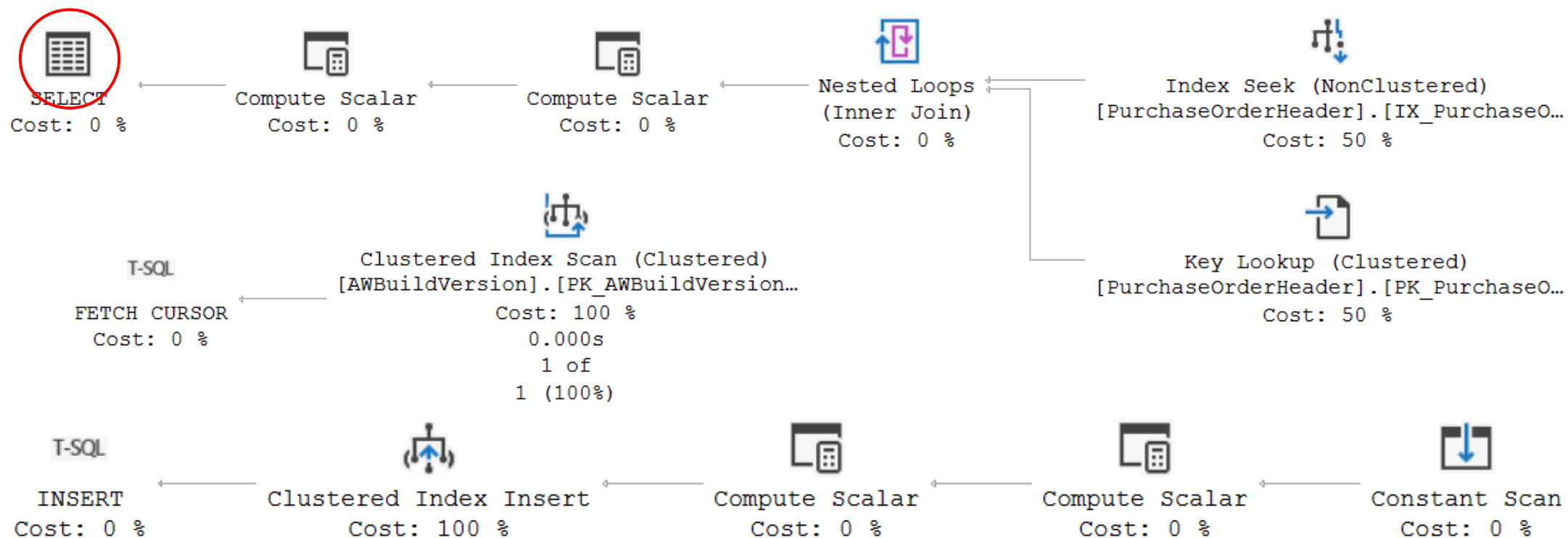
Execution Plan Video Training

Block 1: Understanding execution plans

Level: Advanced

Chapter 1: Properties of the plan

Properties of the plan



Properties of the plan

“Execution Plan Reference”:
sqlserverfast.com/epr

Result operator

- Generic name for the top left operator

- Typically labeled with the statement type

- Not actually a real operator!

 - Represents the client

- Container for properties that apply to the entire plan

 - Only very few (and not very useful) properties in the popup

 - Many properties in the full properties window

 - Some of these are actually VERY useful!

 - Several new properties added over past versions

 - Some, but not all, backported in Service Packs and Cumulative Updates



SELECT

Properties of the plan

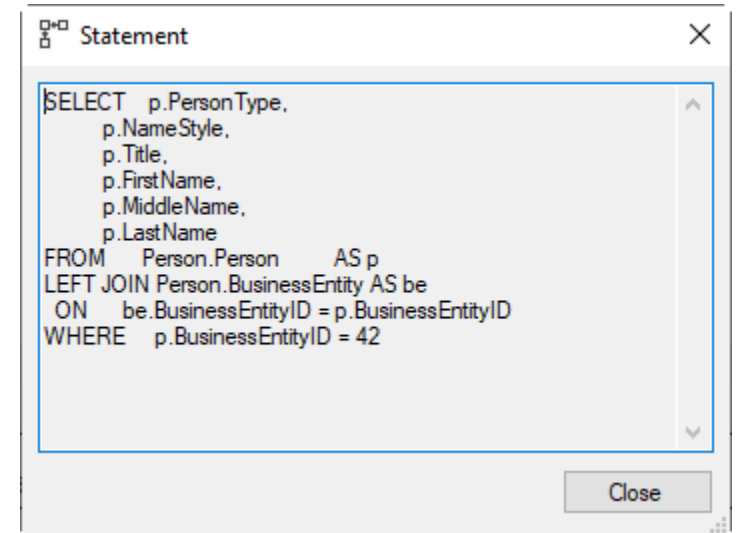
Statement

SQL of the statement that the execution plan executes

Sometimes changed by automatic parameterization

Otherwise the full, unchanged text

Truncated after 4000 characters



Properties of the plan

*Querying the plan cache is explained
in block 1, basic level, chapter 5*

Statement

SQL of the statement that the execution plan executes

Useful when only execution plan available

Execution plan sent to consultant for tuning advice

Execution plan extracted from plan cache

Properties of the plan

Statement

SQL of the statement that the execution plan executes

Useful when only execution plan available

Privacy concerns when sharing

Query text may be considered company confidential

Constant values in the query may be protected personal information

More information about privacy concerns when sharing execution plans:

*<https://www.red-gate.com/simple-talk/sysadmin/data-protection-and-privacy/execution-plans-data-protection/>
(or: <https://tinyurl.com/ExecPlanPrivacy>)*

Properties of the plan

Warnings

Data type conversion *may* affect cardinality estimate

Given whenever any data type conversion is done in the execution plan

Might affect cardinality estimate ... or might not affect it at all!

But might also prevent optimal index usage

SELECT	
Cached plan size	64 KB
Estimated Operator Cost	0 (0%)
Degree of Parallelism	1
Estimated Subtree Cost	0,0065706
Estimated Number of Rows Per Execution	1
Statement	
SELECT * FROM [Sales].[SalesOrderHeader] [soh] WHERE [soh].[SalesOrderNumber]=@1	
Warnings	
Type conversion in expression (CONVERT(nvarchar(23), [soh].[SalesOrderID],0)) may affect "CardinalityEstimate" in query plan choice	

Properties of the plan

Warnings

Data type conversion *may* affect cardinality estimate

Excessive memory grant

SELECT	
Cached plan size	32 KB
Estimated Operator Cost	0 (0%)
Degree of Parallelism	1
Estimated Subtree Cost	8,52439
Memory Grant	12304
Estimated Number of Rows Per Execution	200
Statement	
SELECT TOP(200) SalesOrderID, SalesOrderDetailID, ProductID, UnitPrice, OrderQty, LineTotal FROM (SELECT TOP(@SubsetSize) * FROM Sales.SalesOrderDetail ORDER BY SalesOrderID, SalesOrderDetailID) AS x ORDER BY LineTotal DESC OPTION (OPTIMIZE FOR (@SubsetSize = 100000))	
Warnings	
The query memory grant detected "ExcessiveGrant", which may impact the reliability. Grant size: Initial 12304 KB, Final 12304 KB, Used 88 KB.	

Properties of the plan

Warnings

- Data type conversion *may* affect cardinality estimate

- Excessive memory grant

- Probably other possible causes

Properties of the plan

Missing Indexes

Stored in sys.dm_db_missing_indexes

Also stored in execution plan property MissingIndexes

Highlighted (maximum one) on top of screen in SSMS

MissingIndexes	
Impact	89,5835
MissingIndex	
ColumnGroup	
[1]	
Column	
Column	6
Name	[MiddleName]
Usage	EQUALITY
[2]	
Column	

Query 1: Query cost (relative to the batch): 100%

~~SELECT p.PersonType, p.NameStyle, p.Title, be.ModifiedDate, p.FirstName, p.MiddleName, p.LastName FROM Person.Person AS p LEFT JOIN Person.BusinessEntity AS be ON p.BusinessEntityID = be.BusinessEntityID~~

Missing Index (Impact 89.5835): CREATE NONCLUSTERED INDEX [Name of Missing Index, sysname,>] ON [Person].[Person] ([MiddleName]) INCLUDE ([PersonType],[NameStyle]...

Execution Plan:

- SELECT (Cost: 0 %)
- Compute Scalar (Cost: 0 %)
- Nested Loops (Left Outer Join) (Cost: 0 %)
- Compute Scalar (Cost: 0 %)
- Nested Loops (Inner Join) (Cost: 8 %)
- Index Scan (NonClustered) (Person). [IX_Person_LastName_FirstN... (Cost: 86 %)
- Clustered Index Seek (Clustered) [BusinessEntity]. [PK_BusinessEntity... (Cost: 3 %)
- Key Lookup (Clustered) (Person). [PK_Person_BusinessEntityI... (Cost: 3 %)

INCLUDE

- [AdventureWorks201
- [Person]
- [Person]

Properties of the plan

Missing Indexes

- Stored in sys.dm_db_missing_indexes

- Also stored in execution plan property MissingIndexes

- Highlighted (maximum one) on top of screen in SSMS

- Based on single query optimization only

 - No attempt to synergize with other queries

 - Cost of updating index as data changes not taken into account

 - Order of columns not necessarily optimal, even for the single query

 - Many index types not considered

 - Use as guideline only, not as recommendation to create that index.

Properties of the plan

Optimization Level

TRIVIAL: Just one option for execution plan available; use that option

No parallelism considered, no MissingIndexes property generated

FULL: Multiple options available, go through full optimization.

Reason for Early Termination of Statement Optimization

Only used for FULL optimization

Not present when compilation stops at end of any compilation phase

“Good Enough Plan Found” – plan found below cost threshold for current phase

“Time Out” – too much time spend in current phase

“Memory Limit Exceeded” – exactly what it sounds like

} Execution plan likely not optimal

Properties of the plan

CardinalityEstimationModelVersion

Which version of the cardinality estimator was used?

Values correspond to values used for database compatibility level

Value “70” represents SQL Server 2012 and older

If cardinalities are very wrong, consider using a different version

Properties of the plan

OptimizerStatsUsage

Lists the statistics that were used during compilation

Includes various details about the statistics object

Use this to gauge whether performance issues might be caused by outdated or under-sampled statistics

[-] OptimizerStatsUsage	
[-] [1]	
Database	[AdventureWorks2017]
LastUpdate	27-10-2017 14:33
ModificationCount	0
SamplingPercentage	100
Schema	[Person]
Statistics	[PK_BusinessEntity_Employee]
Table	[BusinessEntity]
[+] [2]	
[+] [3]	
[+] [4]	
[+] [5]	

Properties of the plan

Memory Grant

Memory required to run the execution plan (in kilobytes)

Query waits until memory available

RESOURCE_SEMAPHORE

Memory reserved; execution starts

Spill to tempdb when insufficient memory

Warnings

Operator used tempdb to spill data during execution with spill level 1 and 1 spilled thread(s); Hash wrote 24 pages to and read 24 pages from tempdb with granted memory 3328KB and used memory 2840KB



Hash Match
(Inner Join)

Properties of the plan

Memory Grant

Memory required to run the execution plan (in kilobytes)

Query waits until memory available

RESOURCE_SEMAPHORE

Memory reserved; execution starts

Spill to tempdb when insufficient memory

Property is NOT included ...

In execution plan only and in live execution plan

When only “regular” (low memory) operators are used

SELECT	
Cached plan size	56 KB
Estimated Operator Cost	0 (0%)
Degree of Parallelism	1
Estimated Subtree Cost	1,84557
Memory Grant	3360
Estimated Number of Rows Per Execution	19972
Statement	
SELECT p.PersonType, p.NameStyle, p.Title, be.ModifiedDate, p.FirstName, p.MiddleName, p.LastName--, CLR_database.dbo.Delay(1,) FROM Person.Person AS p LEFT HASH JOIN Person.BusinessEntity AS be ON be.BusinessEntityID = p.BusinessEntityID	

Properties of the plan

Memory Grant

Memory required to run the execution plan (in kilobytes)

Query waits until memory available

RESOURCE_SEMAPHORE

Memory reserved; execution starts

Spill to tempdb when insufficient memory

Property is NOT included ...

In execution plan only and in live execution plan

When only “regular” (low memory) operators are used

MemoryGrantInfo

[-] MemoryGrantInfo	
DesiredMemory	3360
GrantedMemory	3360
GrantWaitTime	0
IsMemoryGrantI	YesAdjusting
LastRequestedM	8352
MaxQueryMem	3004400
MaxUsedMemo	2840
RequestedMem	3360
RequiredMemo	1024
SerialDesiredMe	3360
SerialRequiredM	1024

Properties of the plan

Parameter List

Use to investigate “bad parameter sniffing”

Query optimized for specific value of a parameter (variable)

Plan cached and later reused, when parameter has a different value

Parameter Compiled Value

Values used to estimate row counts

Plan is optimized for this value

Parameter Runtime Value

Only in execution plan plus run-time statistics

Can be reason for estimation errors

Parameter List	@CheckDate; @StartProduct
[1]	@CheckDate
Column	@CheckDate
Parameter Compiled Value	'2015-01-01 00:00:00.000'
Parameter Data Type	datetime
Parameter Runtime Value	'2012-01-01 00:00:00.000'
[2]	@StartProductID

Properties of the plan

QueryTimeStats

Same information as SET STATISTICS TIME

Included in execution plan plus run-time statistics

Additional details when query uses user-defined functions

[-] QueryTimeStats	
CpuTime	64
ElapsedTime	9162
UdfCpuTime	32
UdfElapsedTime	9130

Properties of the plan

QueryTimeStats

WaitStats

Included in execution plan plus run-time statistics

Shows top ten wait reasons, and their time

[-] WaitStats	
[-] [1]	
WaitCount	1
WaitTimeMs	4846
WaitType	LCK_M_S
[-] [2]	
WaitCount	4
WaitTimeMs	131
WaitType	ASYNC_NETWORK_IO

Properties of the plan

*For more information about parallelism,
see block 6: Technical operators*

QueryTimeStats

WaitStats

NonParallelPlanReason

Use to understand why no parallelism was used

MaxDOPSetToOne

MAXDOP setting of 1 prevents parallelism

TSQLUserDefinedFunctionsNotParallelizable

Any scalar T-SQL user-defined function prevents parallelism

Many other possible reasons

Not included when estimated cost < cost threshold for parallelism

Properties of the plan

QueryTimeStats

WaitStats

NonParallelPlanReason

TraceFlags

Shows what trace flags were enabled at compile time

In all execution plans

Shows what trace flags were enabled at run time

In execution plan plus run-time statistics

TraceFlags	
[1]	
IsCompileTime	True
TraceFlag	
[1]	
Scope	Global
Value	1117
[2]	
[3]	
[4]	
[5]	
[2]	
IsCompileTime	False
TraceFlag	
[1]	
Scope	Global
Value	1117
[2]	
[3]	
[4]	
[5]	

Summary

“Execution Plan Reference”:
sqlserverfast.com/epr

Plan properties

- Exposed on top left operator of the execution plan

 - Popup too limited

 - Use full properties window

Many properties available

- Not all of them covered in this course

- New properties often added in major versions or intermediate updates

Next chapters

Chapter 2: Common operator properties

- Properties found on all operators

- Properties found on many operators

Chapter 3: Ordering of the data stream

Chapter 4: Missing nodes

Chapter 5: Batch mode versus row mode