Start

Grant Fritchey

Query Performance Tuning in the Cloud

Goals

Explain the importance of query tuning within the Azure SQL Database environment.

Show how the hybrid tool set can be used to identify poor performance.

Demonstrate the additional functionality available within the Database Management Portal interface.

Get in touch

Grant Fritchey

blog scarydba.com

email grant@scarydba.com

twitter @gfritchey

Why Tune Queries?

You have **no control** over:

- Where your queries run
- How many resources your query uses
- How many CPUs/Disks are dedicated to you
- When the server is just going to go away

Most performance problems are **code or structure related**

One query runs ***many*** places

Then Why Use SQL Database?

Speed of delivery

Extremely low management cost

Cheap prices

Expandable capacity

What's Old is New

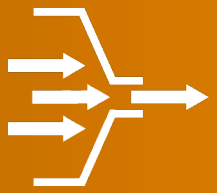
Transactions as **short** as possible

Only move the data you **need** to move

Only move the data **when** you need to move it

Tune the queries

Checking on Throttling



Throttling

Master Database

- Sys.event_log
- Sys.database_connection_stats

Checking on Throttling

	database_name	event_type	event_subtype_desc	description
1	master	connection_successful	connection_successful	Connected successfully to database.
2	master	connection_terminated	idle_connection_timeout	Connection has been idle for longer than system ...
3	MovieManagement	connection_successful	connection_successful	Connected successfully to database.
4	master	connection_successful	connection_successful	Connected successfully to database.
5	MovieManagement	connection_successful	connection_successful	Connected successfully to database.
6	MovieManagement	connection_successful	connection_successful	Connected successfully to database.
7	MovieManagement	connection_successful	connection_successful	Connected successfully to database.
8	master	connection_successful	connection_successful	Connected successfully to database.
9	MovieManagement	connection_successful	connection_successful	Connected successfully to database.
10	MovieManagement	connection_successful	connection_successful	Connected successfully to database.
11	MovieManagement	connection_successful	connection_successful	Connected successfully to database.
12	MovieManagement	connection_terminated	idle_connection_timeout	Connection has been idle for longer than system ...
13	master	connection_successful	connection_successful	Connected successfully to database.
14	master	connection_terminated	idle_connection_timeout	Connection has been idle for longer than system ...
15	MovieManagement	connection_successful	connection_successful	Connected successfully to database.

Tools for Query Tuning

Query Tuning Tools

SQL Server Management Studio

SQL Database Management Portal

Dynamic Management Objects

SQL Server Management Studio

Connectivity required

Object Explorer

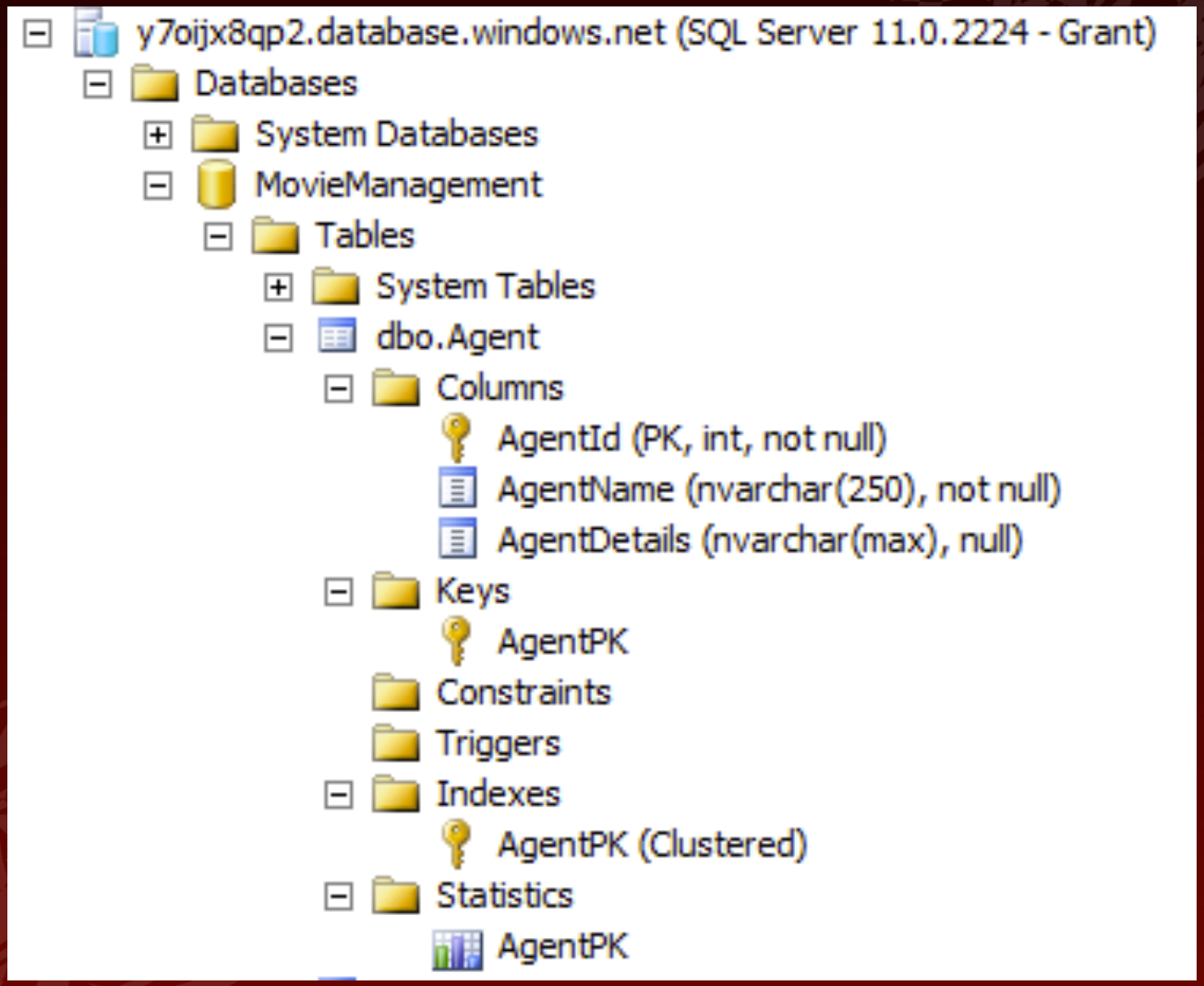
Query window
- And not much else

Statistics IO

Statistics Time

Execution Plans

SSMS – Object Explorer



SSMS - Query Window

SQLQuery4.sql - y7oi...gement (Grant (393))* | Object Explorer Details

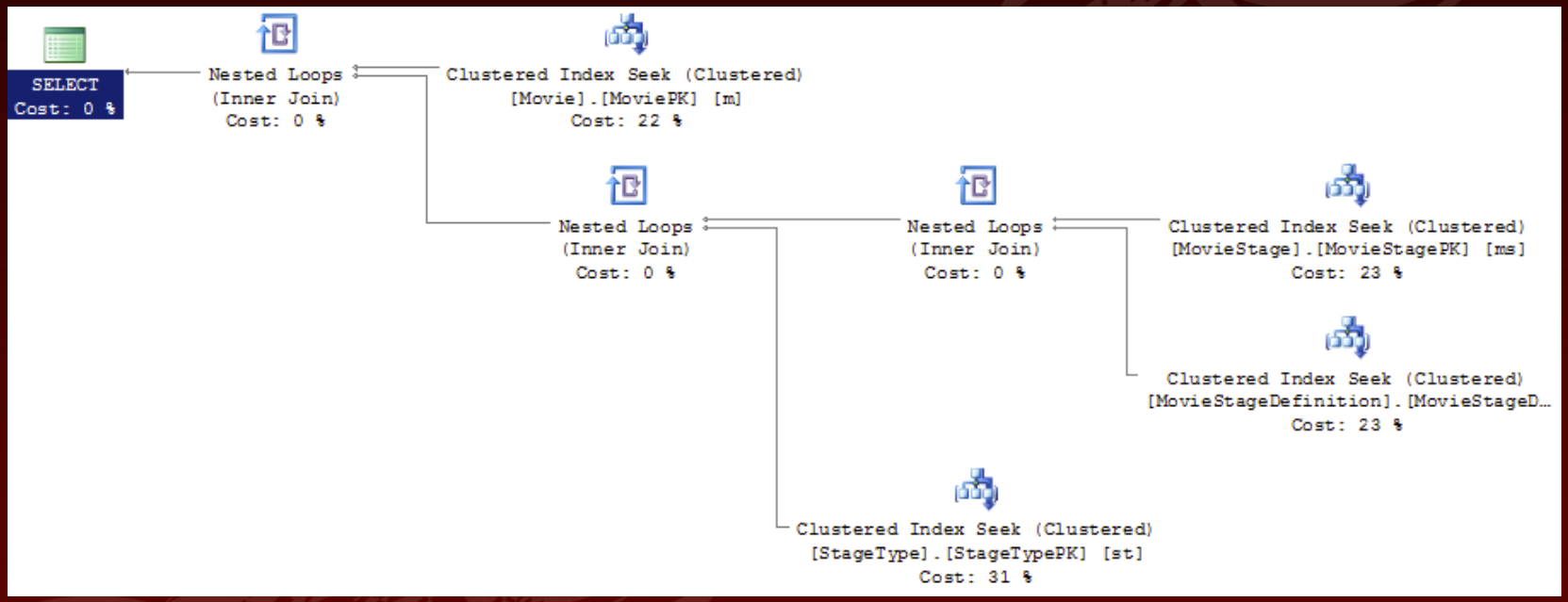
Editor | Results | Messages

```sql
SELECT  m.MovieName,
        msd.MovieStageName,
        st.StageTypeDesc
FROM    dbo.Movie AS m
        JOIN dbo.MovieStage AS ms
        ON m.MovieId = ms.MovieID
        JOIN dbo.MovieStageDefinition AS msd
        ON ms.MovieStageDefinitionId = msd.MovieStageDefinitionId
        JOIN dbo.StageType AS st
        ON msd.StageTypeId = st.StageTypeId
WHERE   m.MovieId = 42;
```

SSMS - Statistics IO/Time

```
SQL Server parse and compile time:
   CPU time = 5 ms, elapsed time = 5 ms.

(2 row(s) affected)
Table 'StageType'. Scan count 0, logical reads 4, physical r
Table 'MovieStageDefinition'. Scan count 0, logical reads 4,
Table 'MovieStage'. Scan count 1, logical reads 6, physical
Table 'Movie'. Scan count 0, logical reads 2, physical reads

 SQL Server Execution Times:
   CPU time = 0 ms,  elapsed time = 0 ms.
SQL Server parse and compile time:
   CPU time = 0 ms, elapsed time = 0 ms.

 SQL Server Execution Times:
   CPU time = 0 ms,  elapsed time = 0 ms.
```

SSMS – Execution Plans



SSMS – Execution Plans

⊟ Misc	
Cached plan size	32 KB
CompileCPU	5
CompileMemory	384
CompileTime	5
Degree of Parallelism	0
Estimated Number of Rows	1.90465
Estimated Operator Cost	0 (0%)
Estimated Subtree Cost	0.0149347
Logical Operation	
⊞ MemoryGrantInfo	
NonParallelPlanReason	CouldNotGenerateValidParallelPlan
Optimization Level	FULL
⊟ OptimizerHardwareDependentPropertie	
EstimatedAvailableDegreeOfParalleli	3
EstimatedAvailableMemoryGrant	432508
EstimatedPagesCached	189222
Physical Operation	
QueryHash	0x9266F93791E55B93
QueryPlanHash	0xCFCF7D576754A38F
Reason For Early Termination Of State	Good Enough Plan Found
RetrievedFromCache	true
⊞ Set Options	ANSI_NULLS: True, ANSI_PADDING:
Statement	SELECT m.MovieName,

SSMS – Execution Plans

⊟ Misc	
Cached plan size	32 KB
CompileCPU	5
CompileMemory	384
CompileTime	5
Degree of Parallelism	1
Estimated Number of Rows	2.19343
Estimated Operator Cost	0 (0%)
Estimated Subtree Cost	0.0153721
Logical Operation	
⊞ MemoryGrantInfo	
Optimization Level	FULL
⊟ OptimizerHardwareDependentPropertie	
EstimatedAvailableDegreeOfParalleli	2
EstimatedAvailableMemoryGrant	209689
EstimatedPagesCached	23532
Physical Operation	
QueryHash	0x9266F93791E55B93
QueryPlanHash	0xCFCF7D576754A38F
Reason For Early Termination Of State	Good Enough Plan Found
RetrievedFromCache	true
⊞ Set Options	ANSI_NULLS: True, ANSI_P
Statement	SELECT m.MovieName,

SQL Database Management Portal

- Monitor
- Statistics Time
- Query window
- Execution Plan
- Statistics IO

Portal: Monitor

Connection Activity

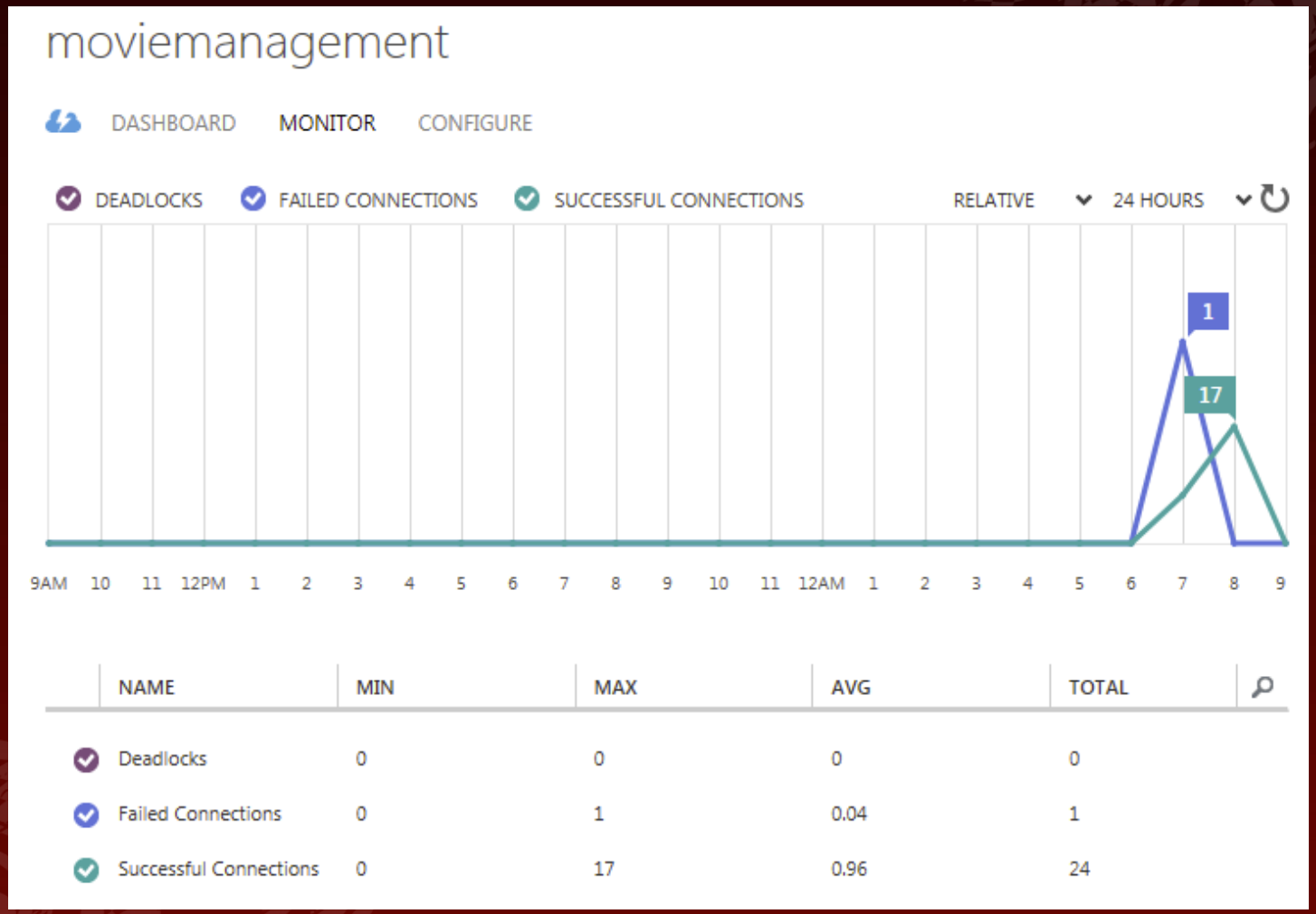
- Including throttled connections

Query Performance

- Aggregation
- Details

Cache dependent

Monitor: Connection Activity



NAME	MIN	MAX	AVG	TOTAL
Deadlocks	0	0	0	0
Failed Connections	0	1	0.04	1
Successful Connections	0	17	0.96	24

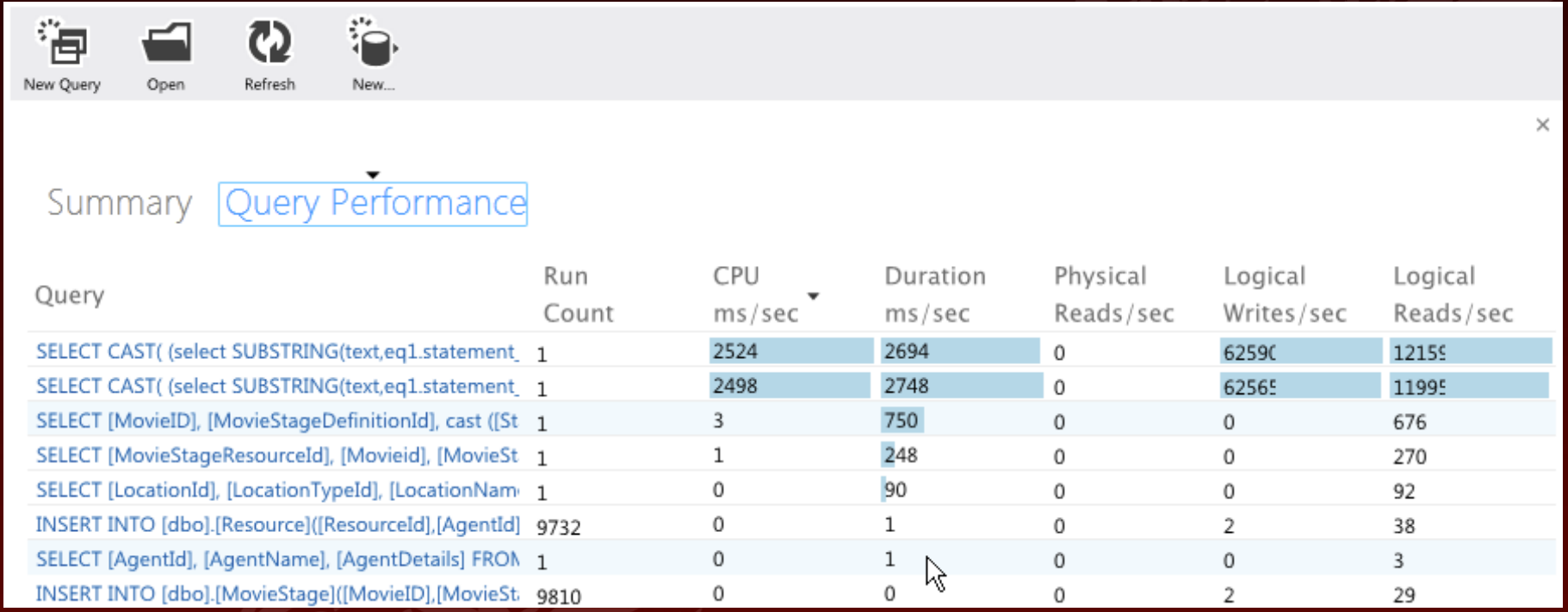
Monitor: Connection Activity

CHOOSE METRICS

Select Metrics that you'd like to Monitor

NAME	UNIT
☐ Blocked by Firewall	Count
☑ Deadlocks	Count
☑ Failed Connections	Count
☑ Successful Connections	Count
☑ Throttled Connections	Count

Monitor: Query Activity



Query	Run Count	CPU ms/sec	Duration ms/sec	Physical Reads/sec	Logical Writes/sec	Logical Reads/sec
SELECT CAST((select SUBSTRING(text,eq1.statement_	1	2524	2694	0	6259(	1215!
SELECT CAST((select SUBSTRING(text,eq1.statement_	1	2498	2748	0	6256!	1199!
SELECT [MovieID], [MovieStageDefinitionId], cast ([St	1	3	750	0	0	676
SELECT [MovieStageResourceId], [Movieid], [MovieSt	1	1	248	0	0	270
SELECT [LocationId], [LocationTypeId], [LocationNam	1	0	90	0	0	92
INSERT INTO [dbo].[Resource]([ResourceId],[AgentId]	9732	0	1	0	2	38
SELECT [AgentId], [AgentName], [AgentDetails] FROM	1	0	1	0	0	3
INSERT INTO [dbo].[MovieStage]([MovieID],[MovieSt	9810	0	0	0	2	29

Monitor: Query Details

```sql
SELECT m.MovieName,
  msd.MovieStageName,
  st.StageTypeDesc
FROM dbo.Movie AS m
  JOIN dbo.MovieStage AS ms
  ON m.MovieId = ms.MovieID
  JOIN dbo.MovieStageDefinition AS msd
  ON ms.MovieStageDefinitionId = msd.MovieStageDefinitionId
  JOIN dbo.StageType AS st
  ON msd.StageTypeId = st.StageTypeId
WHERE m.MovieId = 42;
```

Query Plan Details Query Plan

Resource Use

Resource	Total / sec	Total	Last Run	Minimum	Maximum
Duration (ms)	0	15	0	0	8
CPU (ms)	0	15	0	0	8
Logical Reads	0	96	16	16	16
Physical Reads	0	0	0	0	0
Logical Writes	0	0	0	0	0

Plan Information

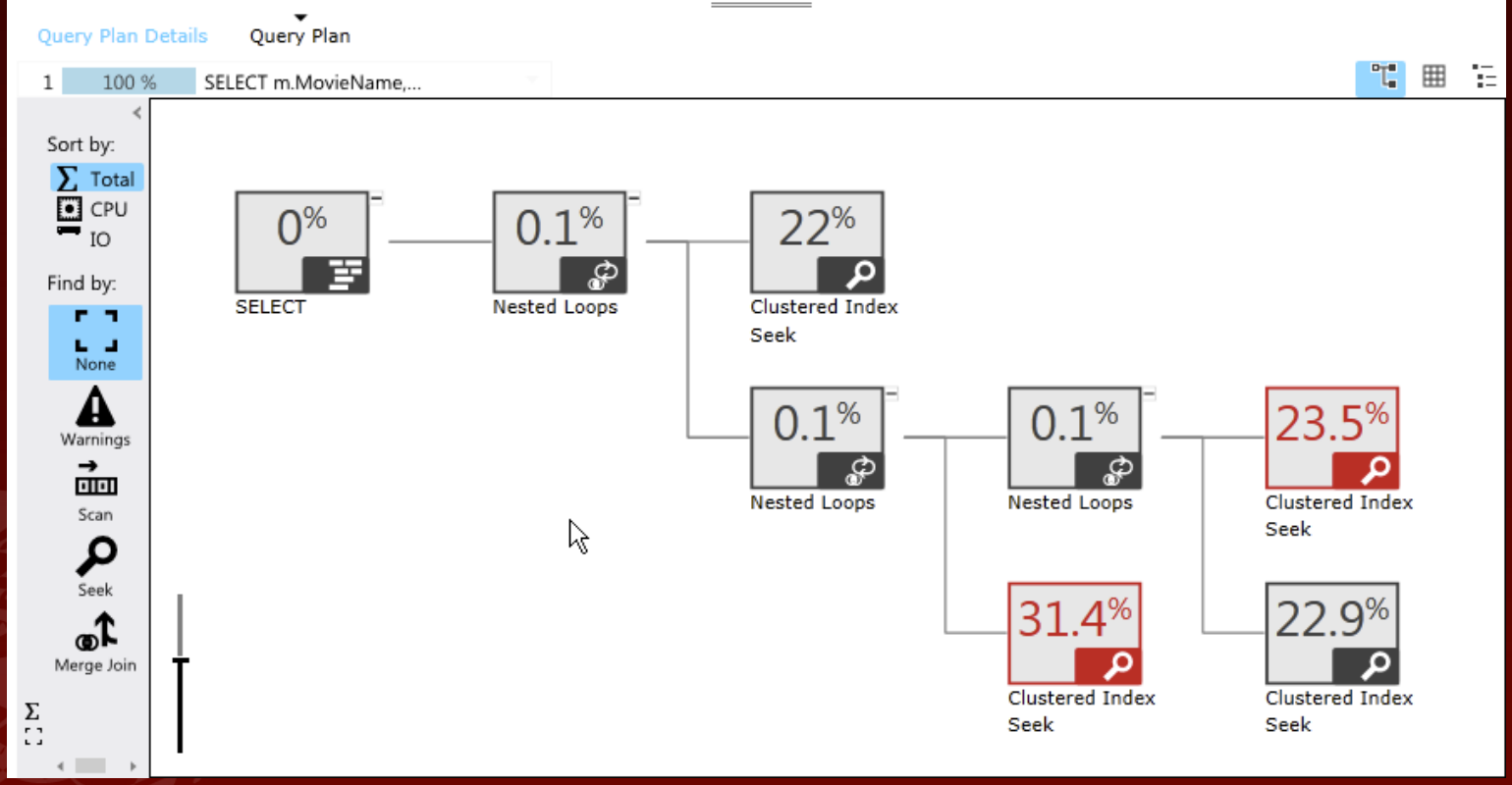
Run Count	6
Last Run Time	1/29/2013 3:30:50 PM
Plan Generation Count	1
Time Plan Cached	1/29/2013 1:42:38 PM

Advanced Information

Plan Handle	0x06009E02A34CDD36D05D502B0700000001000000000000000000000000000000(
SQL Handle	0x02000000A34CDD365C810105880D5C270CC220F745AE25C00000000000000000
Query Hash	0x9266F93791E55B93
Query Plan Hash	0xCFCF7D576754A38F

Monitor: Query Plan

```sql
SELECT m.MovieName,
  msd.MovieStageName,
  st.StageTypeDesc
FROM dbo.Movie AS m
  JOIN dbo.MovieStage AS ms
  ON m.MovieId = ms.MovieID
  JOIN dbo.MovieStageDefinition AS msd
  ON ms.MovieStageDefinitionId = msd.MovieStageDefinitionId
  JOIN dbo.StageType AS st
  ON msd.StageTypeId = st.StageTypeId
WHERE m.MovieId = 42;
```



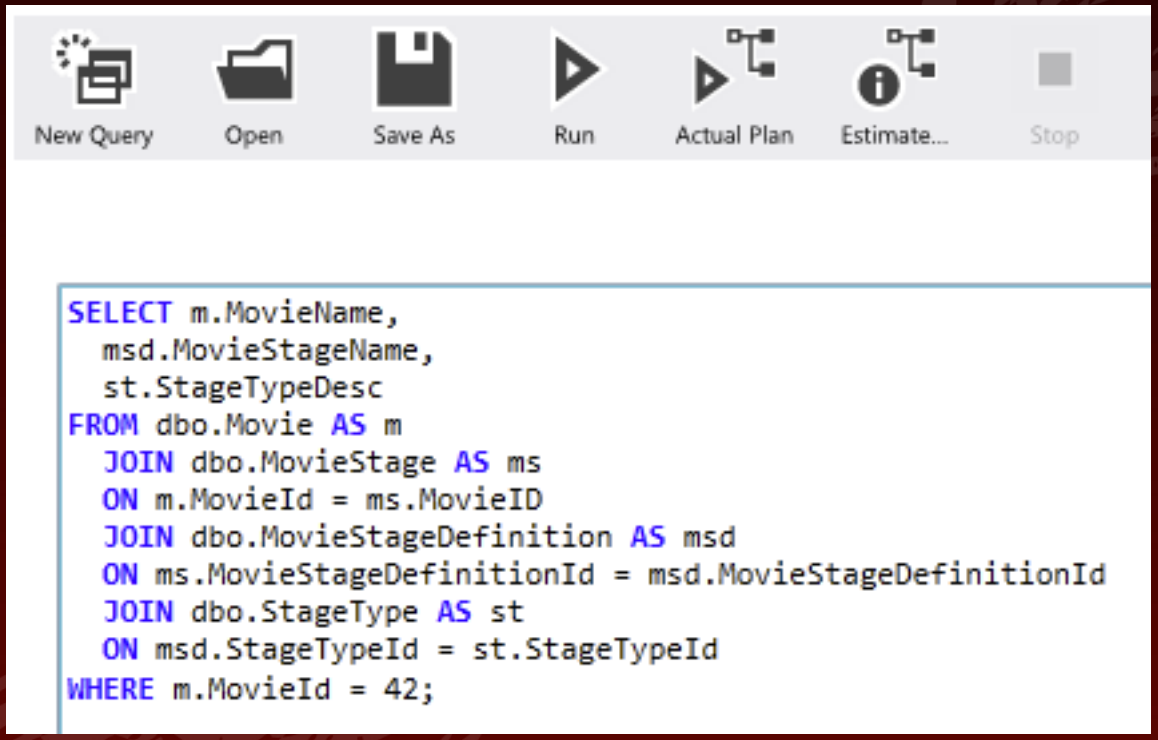
Portal: Query Window

Roughly same as SSMS query window
- » No drag & drop for object names
- » No code completion

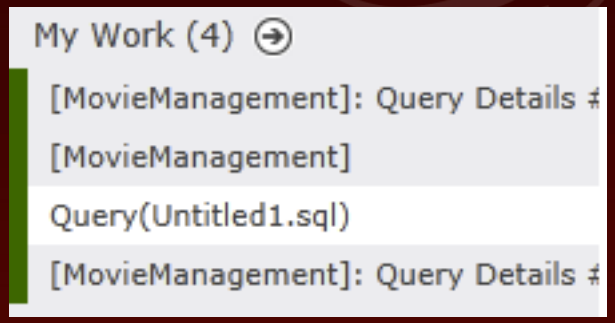
No properties
- » SET STATISTICS IO/TIME ON/OFF

Execution plans
- » Actual
- » Estimated

Query Window



Query Window: My Work



Query Window: Results

New Query | Open | Save As | Run | Actual Plan | Estimate... | Stop

```
SELECT m.MovieName,
  msd.MovieStageName,
  st.StageTypeDesc
FROM dbo.Movie AS m
  JOIN dbo.MovieStage AS ms
  ON m.MovieId = ms.MovieID
  JOIN dbo.MovieStageDefinition AS msd
  ON ms.MovieStageDefinitionId = msd.MovieStageDefinitionId
  JOIN dbo.StageType AS st
  ON msd.StageTypeId = st.StageTypeId
WHERE m.MovieId = 42;
```

Messages | Results

1 | 2 Row(s)

MovieName	MovieStageName	StageTypeDesc
Bryan	Adrienne896	fecit, plorum quo Versus...
Bryan	Brandy3	volcans et et apparens si...

Query Window: Statistics

```sql
SET STATISTICS IO ON;
SET STATISTICS TIME ON;

SELECT m.MovieName,
  msd.MovieStageName,
  st.StageTypeDesc
FROM dbo.Movie AS m
  JOIN dbo.MovieStage AS ms
  ON m.MovieId = ms.MovieID
  JOIN dbo.MovieStageDefinition AS msd
  ON ms.MovieStageDefinitionId = msd.MovieStageDefinitionId
  JOIN dbo.StageType AS st
  ON msd.StageTypeId = st.StageTypeId
WHERE m.MovieId = 42;

SET STATISTICS IO OFF;
SET STATISTICS TIME OFF;
```

Query Window: Messages

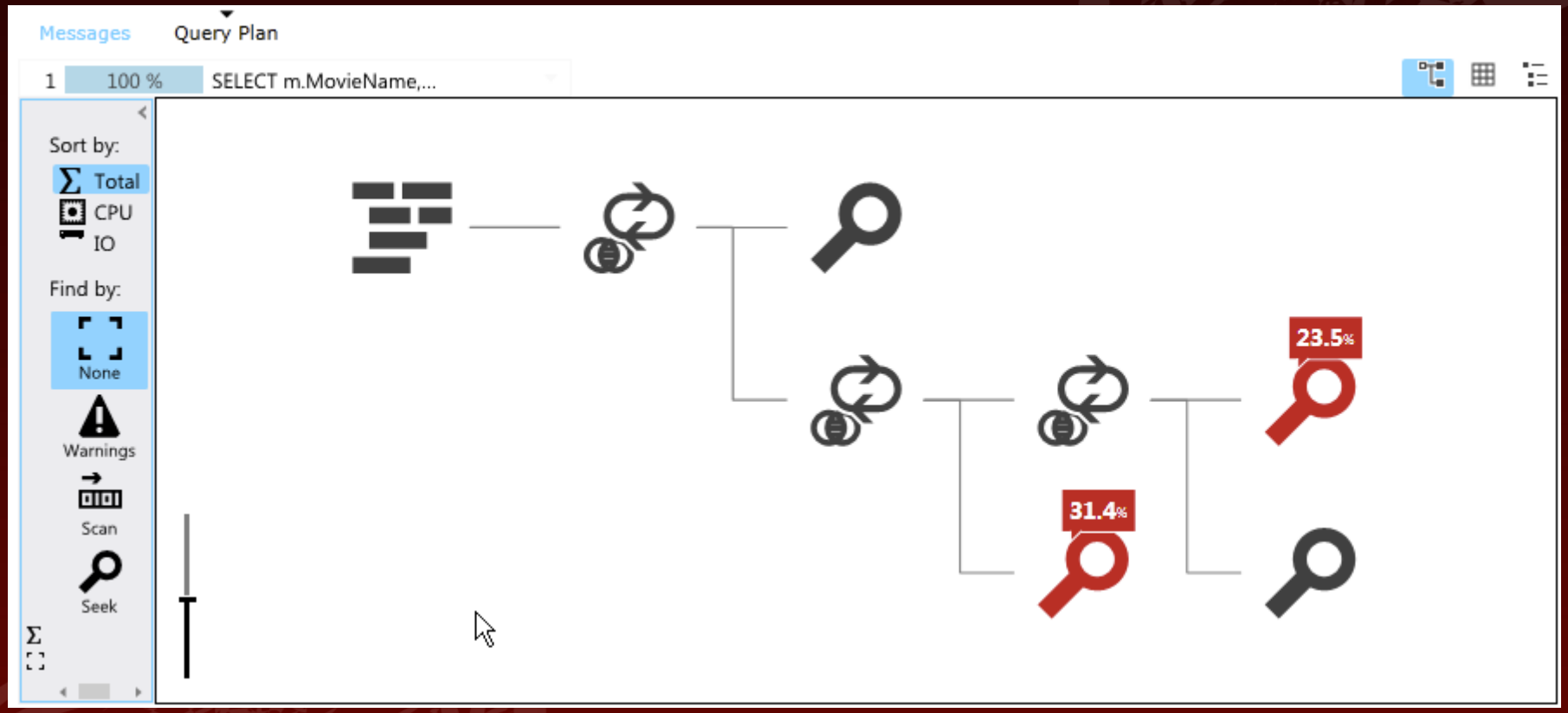
Messages Results

```
SQL Server parse and compile time:
   CPU time = 0 ms, elapsed time = 5 ms.
(2 row(s) affected)
Table 'StageType'. Scan count 0, logical reads 4, physical reads 0, read-ahead reads 0, lob logic
Table 'MovieStageDefinition'. Scan count 0, logical reads 4, physical reads 0, read-ahead reads
reads 0.
Table 'MovieStage'. Scan count 1, logical reads 6, physical reads 0, read-ahead reads 0, lob logi
Table 'Movie'. Scan count 0, logical reads 2, physical reads 0, read-ahead reads 0, lob logical re

 SQL Server Execution Times:
   CPU time = 0 ms,  elapsed time = 0 ms.

 SQL Server Execution Times:
   CPU time = 0 ms,  elapsed time = 0 ms.
```

Portal: Execution Plans

- Graphical
- Can be saved
- Properties
- Highlighting

Estimated Plan



Estimated Plan: SELECT Operator

SELECT 0%

SELECT

Estimated Subtree Cost: 0.0149347
Estimated Rows : 1.90465

Statement

```
SELECT m.MovieName,
  msd.MovieStageName,
  st.StageTypeDesc
FROM dbo.Movie AS m
  JOIN dbo.MovieStage AS ms
  ON m.MovieId = ms.MovieID
  JOIN dbo.MovieStageDefinition AS msd
  ON ms.MovieStageDefinitionId =
msd.MovieStageDefinitionId
  JOIN dbo.StageType AS st
  ON msd.StageTypeId = st.StageTypeId
WHERE m.MovieId = 42;
```

View More

SELECT 0%

SELECT

General	
Node ID	1
Operation Type	SELECT
Operator Costs	
Estimated CPU Cost	0
Estimated I/O Cost	0
Estimated Total Cost	0
Rows	
Estimated Rows	1.90465
Estimated Row Size	0 B
Miscellaneous	
Set Options	ANSI_NULLS:True, ANSI_PADDING:True, ANSI_WARNINGS:True, ARITHABORT:False, CONCAT_NULL_YIELDS_NULL:True, NUMERIC_ROUNDABORT:

View Less

Estimated Plan: SELECT Operator

SELECT 0%

SELECT

	LE:True, NUMERIC_ROUNDABORT: False, QUOTED_IDENTIFIER:True
Statement Comp Id	1
Statement Estimated Rows	1.90465
Statement Id	1
Optmization Level	FULL
Reason For Early Termination Of Statement Optimization	GoodEnoughPlanFound
Statement Subtree Cost	0.0149347
Statement Text	SELECT m.MovieName, msd.MovieStageName, st.StageTypeDesc FROM dbo.Movie AS m JOIN dbo.MovieStage AS ms ON m.MovieId = ms.MovieID

View Less

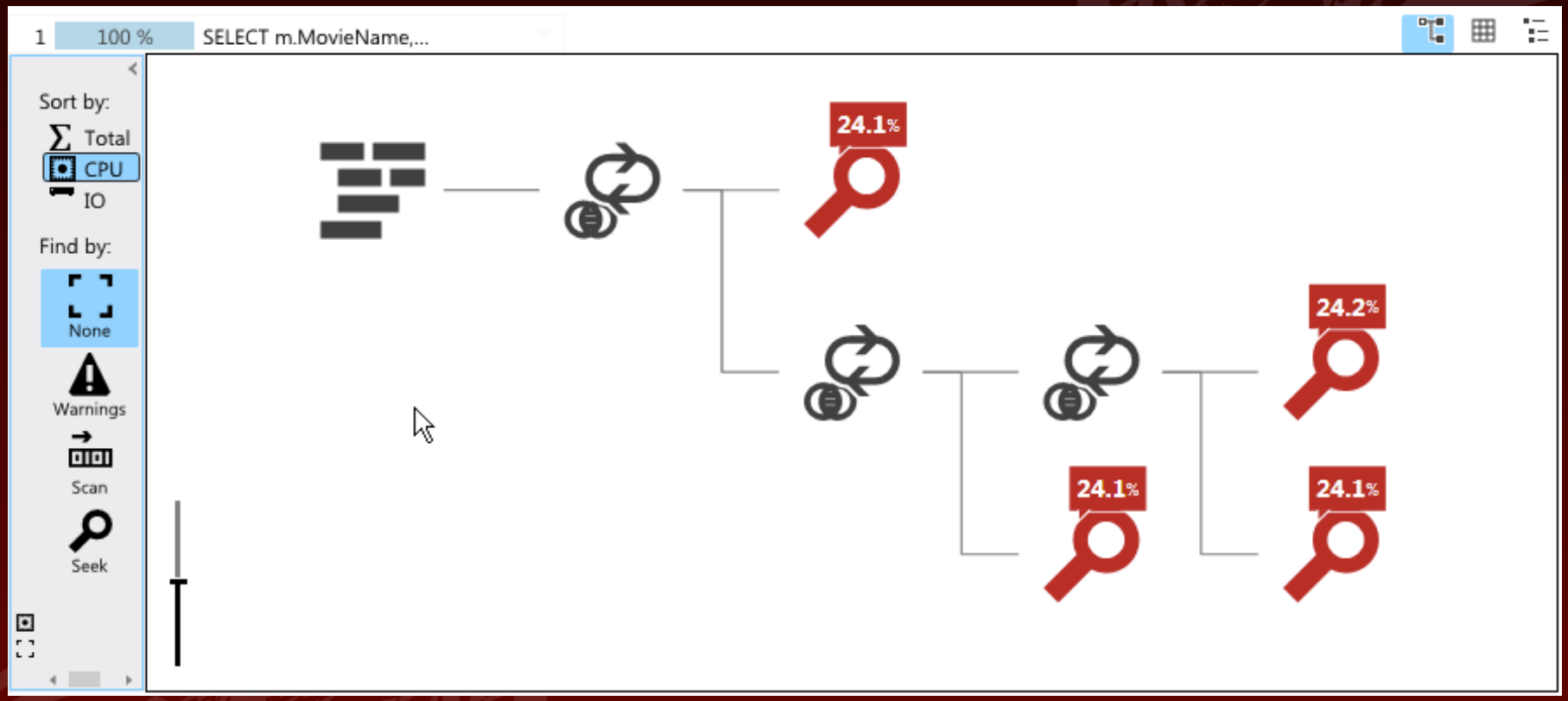
SELECT 0%

SELECT

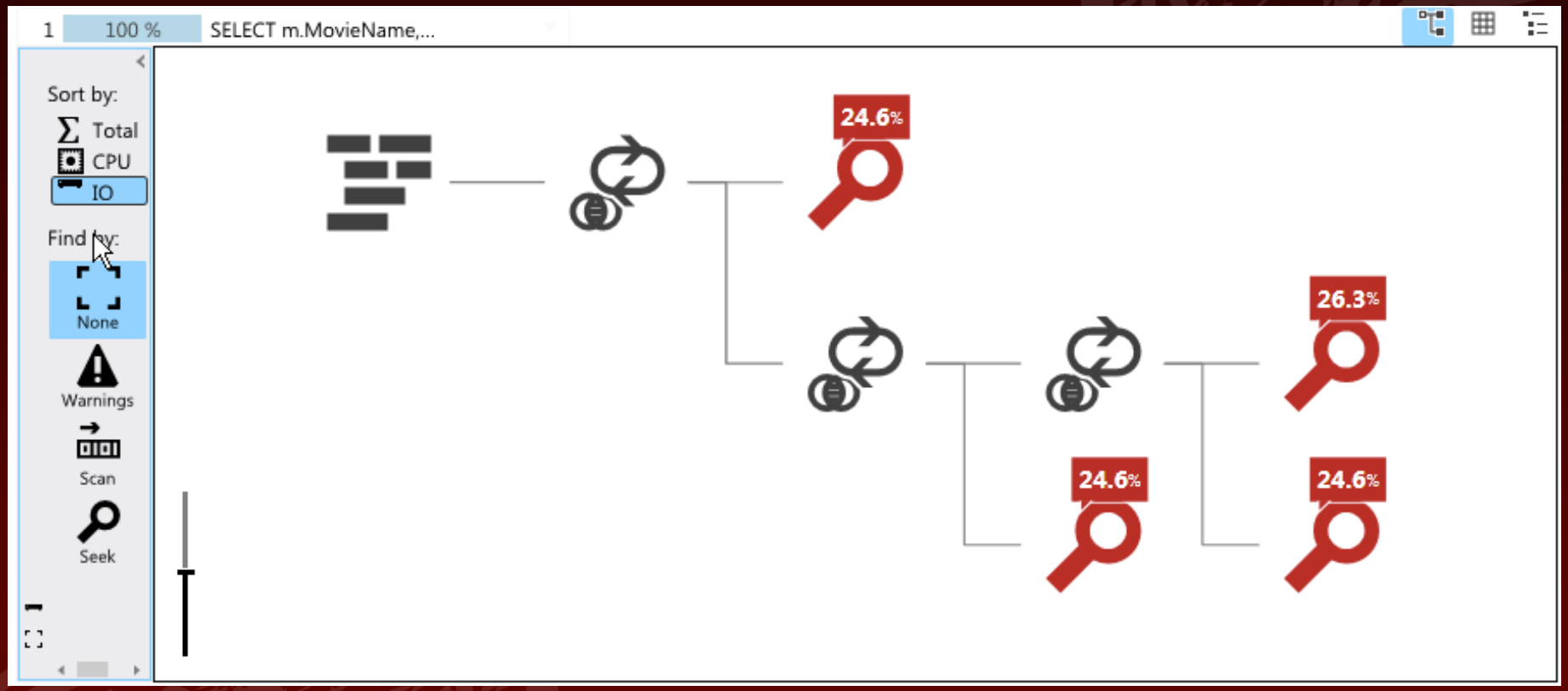
	ON msd.StageTypeId = st.StageTypeId WHERE m.MovieId = 42;
Query Hash	0x9266F93791E55B93
Query Plan Hash	0xCFCF7D576754A38F
Retrieved From Cache	false
Memory Grant Info	SerialDesiredMemory:0, SerialRequiredMemory:0
Optimizer Hardware Dependent Properties	EstimatedAvailableMemoryGrant:432508, EstimatedPagesCached:189222, EstimatedAvailableDegreeOfParallelism:3
Reason for NonParallel Plan	CouldNotGenerateValidParallelPlan
Cached plan size	32
Compile Time	4
Compile CPU	4
Compile Memory	384

View Less

Actual Plan: Sort by CPU



Actual Plan: Sort by IO



Actual Plan: Grid View

1 100 % SELECT m.MovieName,...

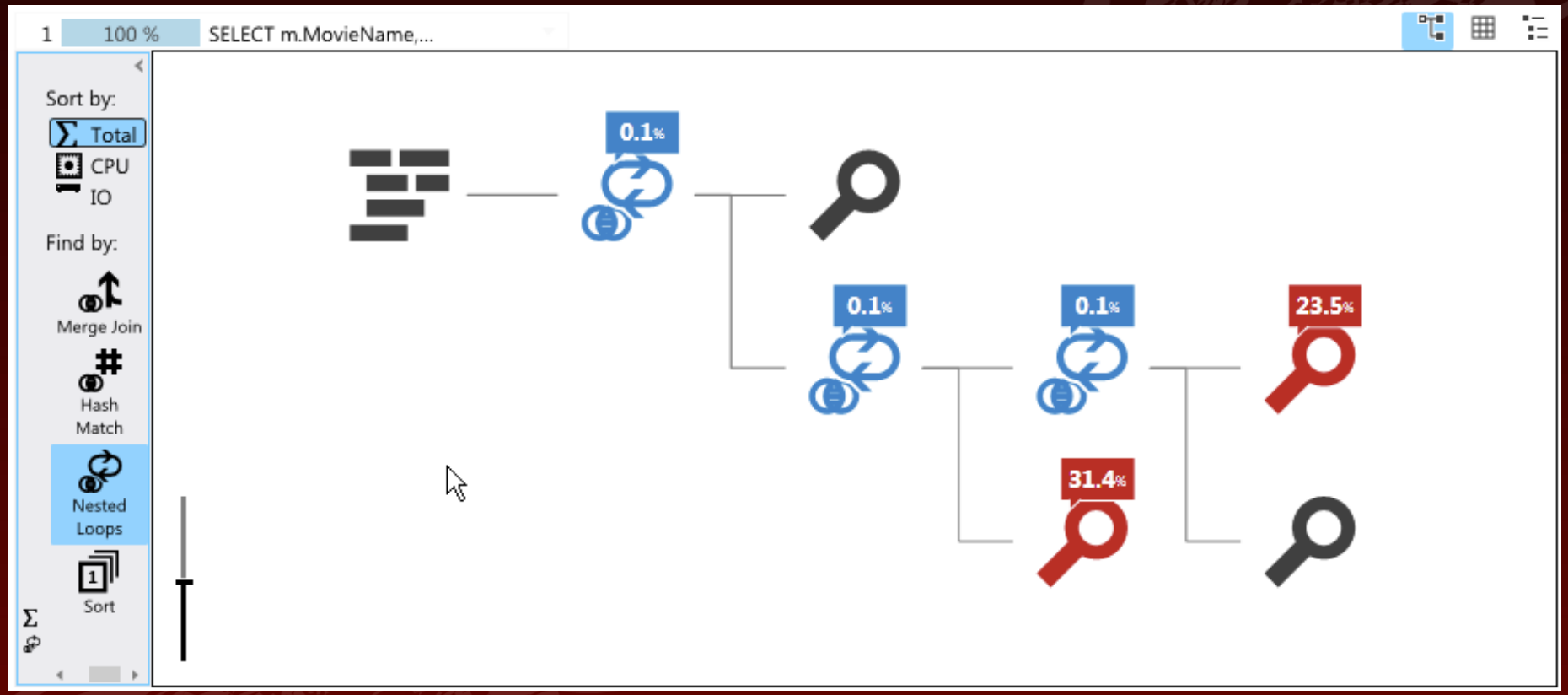
Node ID	Logical Operation	Physical Operation	Estimated CPU Cost	Estimated I/O Cost	Average Row Size(bytes)	Estimated Rows	Estimated Rebinds	Estimated Rewinds	Actual Rows
1	SELECT	SELECT	0	0	0	1.90465			
2	InnerJoin	NestedLoops	7.96143E-06	0	465	1.90465	0	0	2
3	ClusteredIndexSeek	ClusteredIndexSeek	0.0001581	0.003125	261	1	0	0	1
4	InnerJoin	NestedLoops	7.96143E-06	0	213	1.90465	0	0	2
5	InnerJoin	NestedLoops	7.96143E-06	0	113	1.90465	0	0	2
6	ClusteredIndexSeek	ClusteredIndexSeek	0.000159095	0.00334843	11	1.90465	0	0	2
7	ClusteredIndexSeek	ClusteredIndexSeek	0.0001581	0.003125	113	1	0.904647	0	2
8	ClusteredIndexSeek	ClusteredIndexSeek	0.0001581	0.003125	111	1	0.904647	0	2

Actual Plan: Tree View

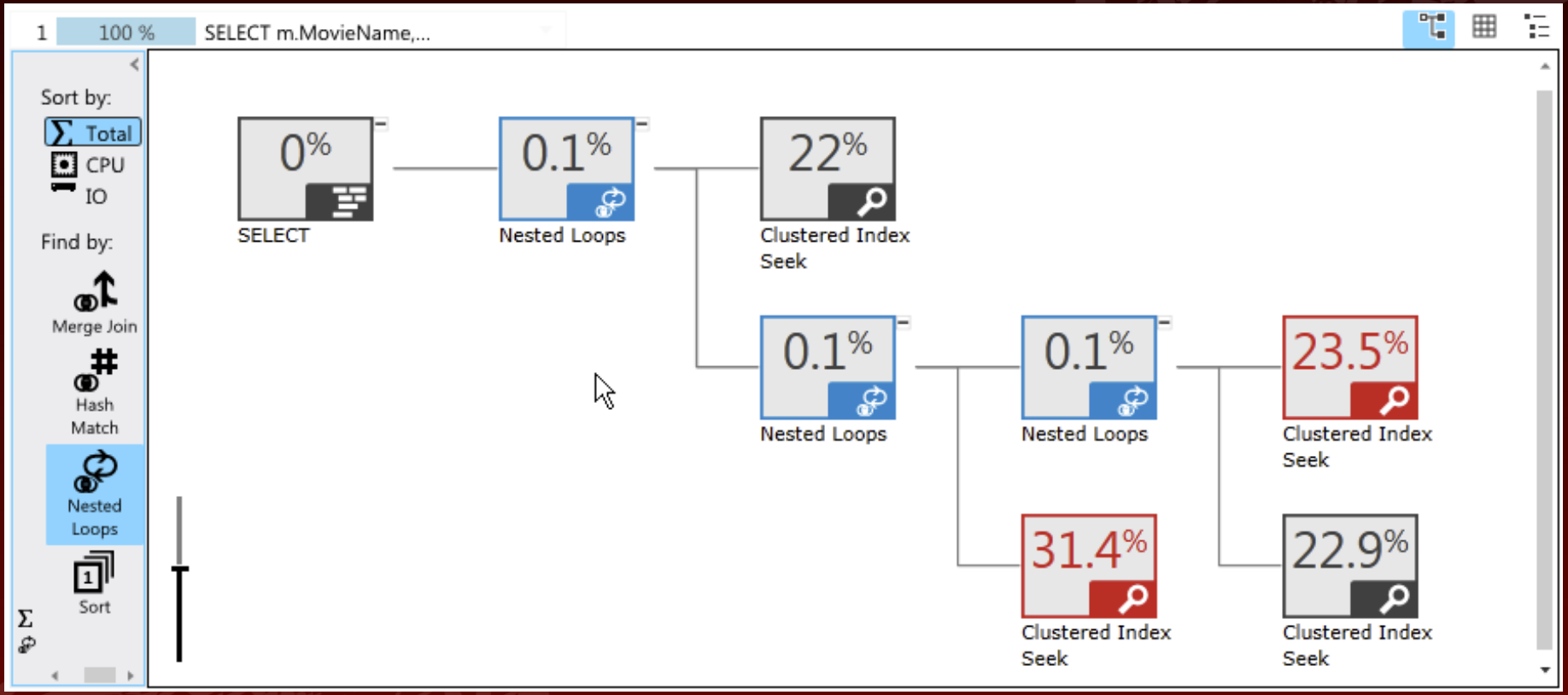
1 | 100 % | SELECT m.MovieName,...

Physical Operation	Node ID	Logical Operation	Estimated CPU Cost	Estimated I/O Cost	Average Row Size(bytes)	Estimated Rows	Estimated Rebinds	Estima Rewin
SELECT	1	SELECT	0	0	0	1.90465		
NestedLoops	2	InnerJoin	7.96143E-06	0	465	1.90465	0	0
ClusteredIndexSeek	3	ClusteredIndexSeek	0.0001581	0.003125	261	1	0	0
NestedLoops	4	InnerJoin	7.96143E-06	0	213	1.90465	0	0
NestedLoops	5	InnerJoin	7.96143E-06	0	113	1.90465	0	0
ClusteredIndexSeek	6	ClusteredIndexSeek	0.000159095	0.00334843	11	1.90465	0	0
ClusteredIndexSeek	7	ClusteredIndexSeek	0.0001581	0.003125	113	1	0.904647	0
ClusteredIndexSeek	8	ClusteredIndexSeek	0.0001581	0.003125	111	1	0.904647	0

Actual Plan: Find Nested Loops



Actual Plan: Zoomed In



DMO Differences

Reset on database move

» Not reboot, restart, attach, detach, etc.

Operating system information is gone

No clickable execution plans

Wait State Monitoring

sys.dm_exec_requests does show waits

sys.dm_db_wait_stats

Special wait types

- » SE_REPL_SLOW_SECONDARY_THROTTLE
- » SE_REPL_COMMIT_ACK

Expected wait types

- » RESOURCE_SEMAPHORE
- » *IO_LATCH
- » SOS_SCHEDULER_YIELD

Wait State Monitoring

	wait_type	waiting_tasks_count	wait_time_ms	max_wait_time_ms	signal_wait_time_ms
1	PAGEIOLATCH_SH	290	3148	403	7
2	ASYNC_NETWORK_IO	33	45	35	1
3	SOS_SCHEDULER_YIELD	186	11	0	10
4	WRITELOG	2	0	0	0

Query DMOs

- Sys.dm_exec_requests
- Sys.dm_exec_query_stats
- Sys.dm_exec_sql_text
- Sys.dm_exec_query_plan
- Sys.dm_exec_query_plan_text

Index DMO

- Sys.dm_db_index_operational_stats
- Sys.dm_db_index_physical_stats
- Sys.dm_db_index_usage_stats

Goals

Explain the importance of query tuning within the Azure SQL Database environment.

Show how the hybrid tool set can be used to identify poor performance.

Demonstrate the additional functionality available within the Database Management Portal interface.

Get in touch

Grant Fritchey

blog scarydba.com

email grant@scarydba.com

twitter @gfritchey

Resources

Scarydba.com/resources

SQL Server 2012 Query Performance Tuning *by Grant Fritchey*

Performance Tuning with SQL Server Dynamic Management Views *by Louis Davidson and Tim Ford*

Windows Azure SQL Database and SQL Server – Performance and Scalability Compared and Contrasted