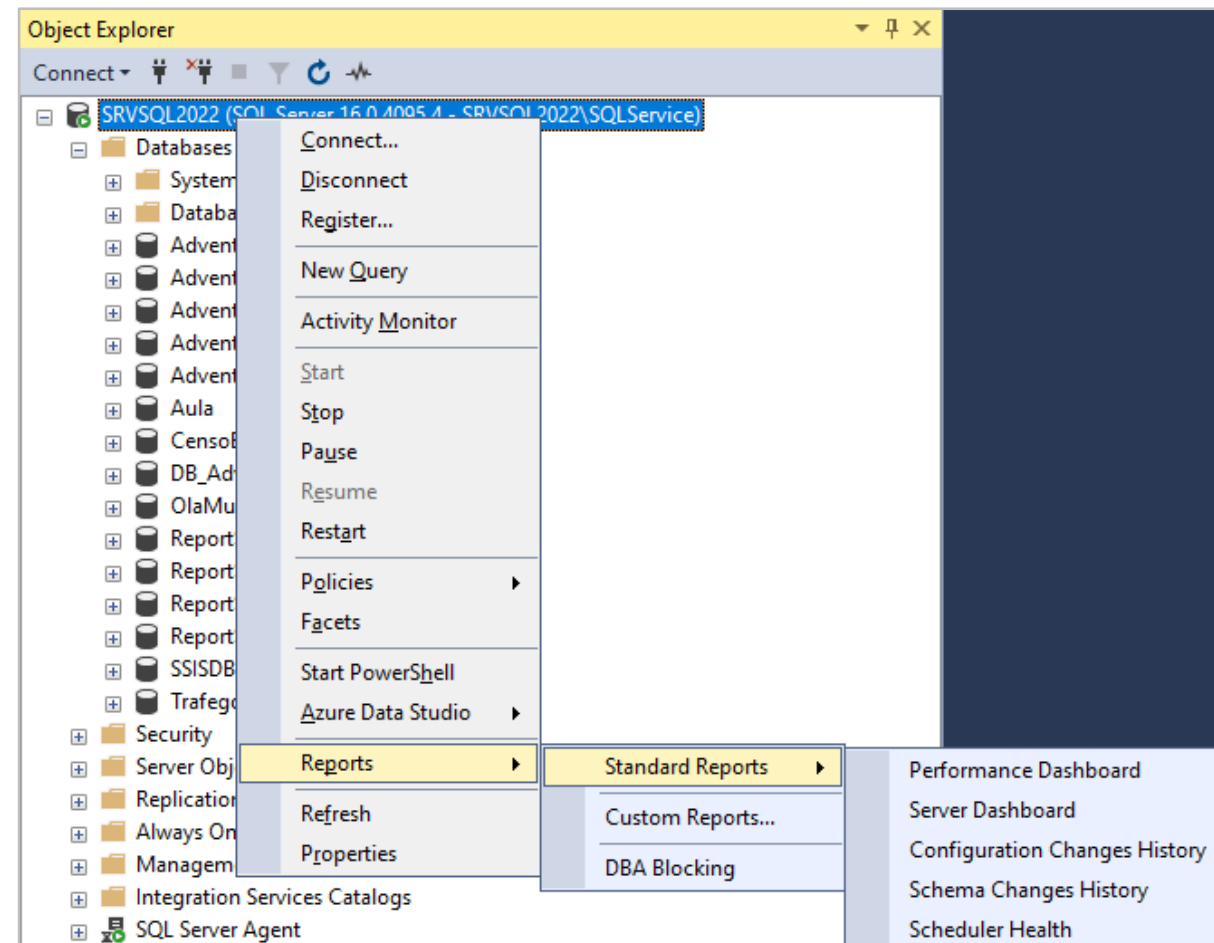


Agenda LIVE #020 - Relatórios Customizados no Management Studio (SSMS)

- Relatórios nativos do SSMS
- Como desenvolver relatórios customizados
- **Hands On:** Desenvolvendo relatório customizado de Blocking
- Aprendendo a utilizar sys.dm_os_performance_counters
- **Hands On:** Desenvolvendo relatório customizado de Contadores

Relatórios nativos do Management Studio (SSMS)

- Sensível ao contexto, mostra lista de relatórios diferentes dependendo da onde clicar.
 - ✓ Servidor
 - ✓ Banco de dados
 - ✓ SQL Server Agent





- **Hands On:** Explorando relatórios nativos



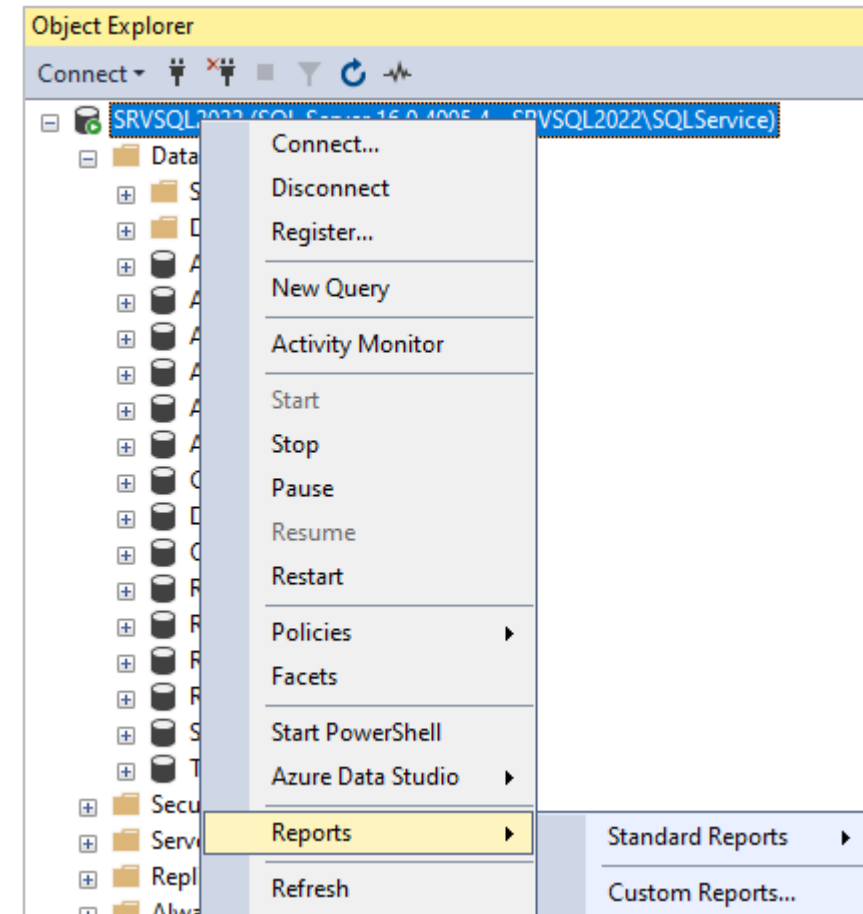


Como desenvolver relatórios customizados

- Utilizar projeto do Reporting Services.
- Datasource e Dataset devem ser locais ao relatório.
- Variáveis globais com informações do contexto do clique:
 - ✓ ObjectName (string)
 - ✓ ObjectTypeName (string)
 - ✓ Filtered (booleano True ou False)
 - ✓ ServerName (string)
 - ✓ FontName (string)
 - ✓ DatabaseName (string)

Como desenvolver relatórios customizados

- Copiar o arquivo RDL para uma pasta e publicar utilizando a opção "Custom reports".
- Disponível apenas no contexto onde foi publicado.
- Cada Login Windows tem sua própria lista de relatórios customizados.



Desenvolvendo relatório customizado de Blocking

- Consulta 1

Results Messages			
	User Connections	Active Transactions	Processes blocked
1	21	1	5

- Consulta 2

Results Messages								
	DataHora	SPID	Status	TempoEspera_Seg	SPID_Blocking	Banco	Computador	UsuarioWindows
1	2024-05-19 15:17:39.963	52	RAIZ	0	0	master	SRVSQL2022	SRVSQL2022/Landry
2	2024-05-19 15:17:39.963	59	BLOCK	321	52	Aula	SRVSQL2022	SRVSQL2022/Landry
3	2024-05-19 15:17:39.963	64	BLOCK	131	59	master	SRVSQL2022	SRVSQL2022/SQLService
4	2024-05-19 15:17:39.963	73	BLOCK	319	59	Aula	SRVSQL2022	SRVSQL2022/Landry
5	2024-05-19 15:17:39.963	79	BLOCK	316	59	Aula	SRVSQL2022	SRVSQL2022/Landry
6	2024-05-19 15:17:39.963	80	BLOCK	313	59	Aula	SRVSQL2022	SRVSQL2022/Landry



- **Hands On:** Relatório customizado de Blocking





Aprendendo a utilizar sys.dm_os_performance_counters

- Acesso aos contadores do SQL Server, os mesmos do Performance Monitor.
- Coluna CNTR_TYPE define o tipo de análise que deve ser feita:
 - ✓ 272696320: Não documentado
 - ✓ 65792 – PERF_COUNTER_LARGE_RAWCOUNT
 - ✓ 272696576 – PERF_COUNTER_BULK_COUNT
 - ✓ 537003264 – PERF_LARGE_RAW_FRACTION
 - ✓ 1073874176 – PERF_AVERAGE_BULK
 - ✓ 1073939712 – PERF_LARGE_RAW_BASE

Aprendendo a utilizar sys.dm_os_performance_counters

- 65792 – PERF_COUNTER_LARGE_RAWCOUNT
 - ✓ O mais fácil de utilizar, basta aferir um valor.

```
SELECT object_name as Objeto, counter_name as Contador,  
cntr_value as Valor  
  
FROM sys.dm_os_performance_counters  
WHERE cntr_type = 65792  
and counter_name in ('Page life expectancy',  
'Total Server Memory (KB)', 'Target Server Memory (KB)',  
'Database pages', 'User Connections')
```

	Objeto	Contador	Valor
1	SQLServer:Buffer Manager	Database pages	24135
2	SQLServer:Buffer Manager	Page life expectancy	26991
3	SQLServer:Buffer Node	Database pages	24135
4	SQLServer:Buffer Node	Page life expectancy	26991
5	SQLServer:General Statistics	User Connections	3
6	SQLServer:Memory Manager	Target Server Memory (KB)	1048576
7	SQLServer:Memory Manager	Total Server Memory (KB)	528672

Aprendendo a utilizar sys.dm_os_performance_counters

- 272696576 – PERF_COUNTER_BULK_COUNT
 - ✓ Necessita de duas coletas com um intervalo de tempo fixo entre elas.

1ª Coleta

```
SELECT object_name as Objeto, counter_name as Contador,
cntr_value as Valor

INTO #PrimeiraColeta
FROM sys.dm_os_performance_counters
WHERE cntr_type = 272696576
and instance_name in ('_Total','')
and counter_name in ('Lock Waits/sec',
'Number of Deadlocks/sec',
'Transactions/sec',
'Log Flush Waits/sec','Latch Waits/sec',
'Full Scans/sec','Index Searches/sec',
'Forwarded Records/sec','Page Splits/sec',
'Batch Requests/sec','Page lookups/sec')
ORDER BY 1,2

WAITFOR DELAY '00:00:10'
```

2ª Coleta

```
SELECT object_name as Objeto, counter_name as Contador,
cntr_value as Valor

INTO #SegundaColeta
FROM sys.dm_os_performance_counters
WHERE cntr_type = 272696576
and instance_name in ('_Total','')
and counter_name in ('Lock Waits/sec',
'Number of Deadlocks/sec',
'Transactions/sec',
'Log Flush Waits/sec','Latch Waits/sec',
'Full Scans/sec','Index Searches/sec',
'Forwarded Records/sec','Page Splits/sec',
'Batch Requests/sec','Page lookups/sec')
ORDER BY 1,2

SELECT a.Objeto,a.Contador, (b.Valor - a.Valor) / 10 as Valor
FROM #PrimeiraColeta a
JOIN #SegundaColeta b ON a.Contador = b.Contador
ORDER BY 1,2

DROP TABLE #PrimeiraColeta
DROP TABLE #SegundaColeta
```

Aprendendo a utilizar sys.dm_os_performance_counters

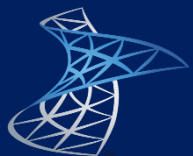
- 1073874176 – PERF_AVERAGE_BULK (A1 - A2)
- 1073939712 – PERF_LARGE_RAW_BASE (B1 - B2)
- ✓ Fórmula $(A2 - A1) / (B2 - B1)$

```
DECLARE @A1 int, @A2 int, @B1 int, @B2 int

-- 1ª amostra
SET @A1 = (SELECT cntr_value FROM sys.dm_os_performance_counters c
           where c.counter_name = 'Average Latch Wait Time (ms)') -- A1
SET @B1 = (SELECT cntr_value FROM sys.dm_os_performance_counters c
           where c.counter_name = 'Average Latch Wait Time Base') -- B1
WAITFOR DELAY '00:00:10'

-- 2ª amostra
SET @A2 = (SELECT cntr_value FROM sys.dm_os_performance_counters c
           where c.counter_name = 'Average Latch Wait Time (ms)') -- A2
SET @B2 = (SELECT cntr_value FROM sys.dm_os_performance_counters c
           where c.counter_name = 'Average Latch Wait Time Base') -- B2

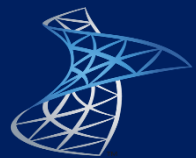
-- (A2 - A1) / (B2 - B1)
SELECT @A1 Amostra1, @B1 Base1, @A2 Amostra2, @B2 Base2,
       (@A2 - @A1) / (@B2 - @B1) AS Result
```



Aprendendo a utilizar sys.dm_os_performance_counters

- 1073939712 – PERF_LARGE_RAW_BASE (B)
- 537003264 – PERF_LARGE_RAW_FRACTION (F)
- ✓ Fórmula $100 \times (F / B)$

```
;WITH CTE_Base as (  
SELECT 1 as ID, counter_name as Contador, cntr_value as Valor  
FROM sys.dm_os_performance_counters  
WHERE counter_name = 'Buffer cache hit ratio base'),  
  
CTE_Fraction as (  
SELECT 1 as ID, counter_name as Contador, cntr_value as Valor  
FROM sys.dm_os_performance_counters  
WHERE counter_name = 'Buffer cache hit ratio')  
  
SELECT 'Buffer cache hit ratio (%)' as Contador,  
100 * (f.Valor/b.Valor) as Valor  
FROM CTE_Base b JOIN CTE_Fraction f on b.ID = f.ID
```



- **Hands On:** Relatório customizado de Contadores





Curso Turbo SQL

Aprenda a escrever consultas eficientes, clique no Link abaixo para saber mais: [Clique para saber mais](#)



E-mail: proflandry.sqlexpert@gmail.com



Youtube: @prof-landrySQLServerExpert



Instagram: @sqlserver.expert



Facebook: @sqlserver.expert.landry