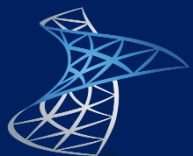




Agenda LIVE #008 – Monitorando o Microsoft SQL Server

- Windows Performance Monitor
- Contadores do Performance Monitor
- Windows Resource Monitor
- Dynamic Management Views (DMVs)
- SQL Server Profiler
- Extended Event



- **Aplicação com baixo desempenho, o que fazer?**

- ✓ Diversas causas individuais ou combinações.

- ✓ Algumas causas possíveis:

- Codificação da Aplicação
 - Servidor de Aplicação
 - Infraestrutura de Rede
 - Servidor de Banco de Dados
 - Consultas ineficientes
 - Ausência de Índices
 - Índices ineficientes
 - Fragmentação
 - Projeto Físico do Banco de Dados



- **Monitorar para Avaliar a(s) Causa(s)**

- ✓ Avaliar o consumo de recursos no Servidor SQL Server utilizando **Windows Performance Monitor**.
- ✓ Se for possível, identificar Telas ou Módulos da Aplicação com baixo desempenho. Utilizar **Extended Event** para capturar as consultas.
- ✓ Utilizar **Extended Event** para captura da atividade geral, selecionando as consultas com:
 - Maior volume de I/O
 - Maior consumo de CPU
 - Maior frequência de execução
 - Maior tempo de execução



- **Recursos Nativos para Monitorar**

- ✓ Windows

- Performance Monitor
 - Resource Monitor

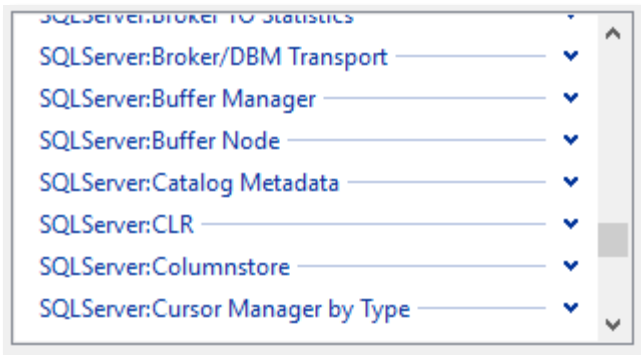
- ✓ SQL Server

- Dynamic Management Views (DMVs)
 - SQL Server Profiler
 - Extended Event

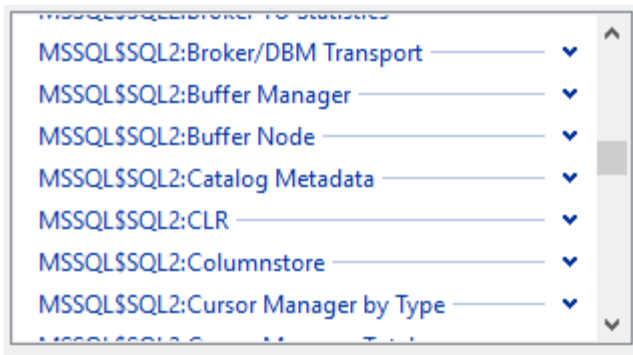
• Windows Performance Monitor

- ✓ Aplicação do Windows que disponibiliza inúmeros contadores organizados em objetos.
- ✓ **Contadores para monitorar SGBDs:**
 - ✓ Contadores do Windows relacionados ao consumo de recursos de Hardware (memória, processador, discos, etc.).
 - ✓ Contadores do SQL Server por instância.

Instância Default

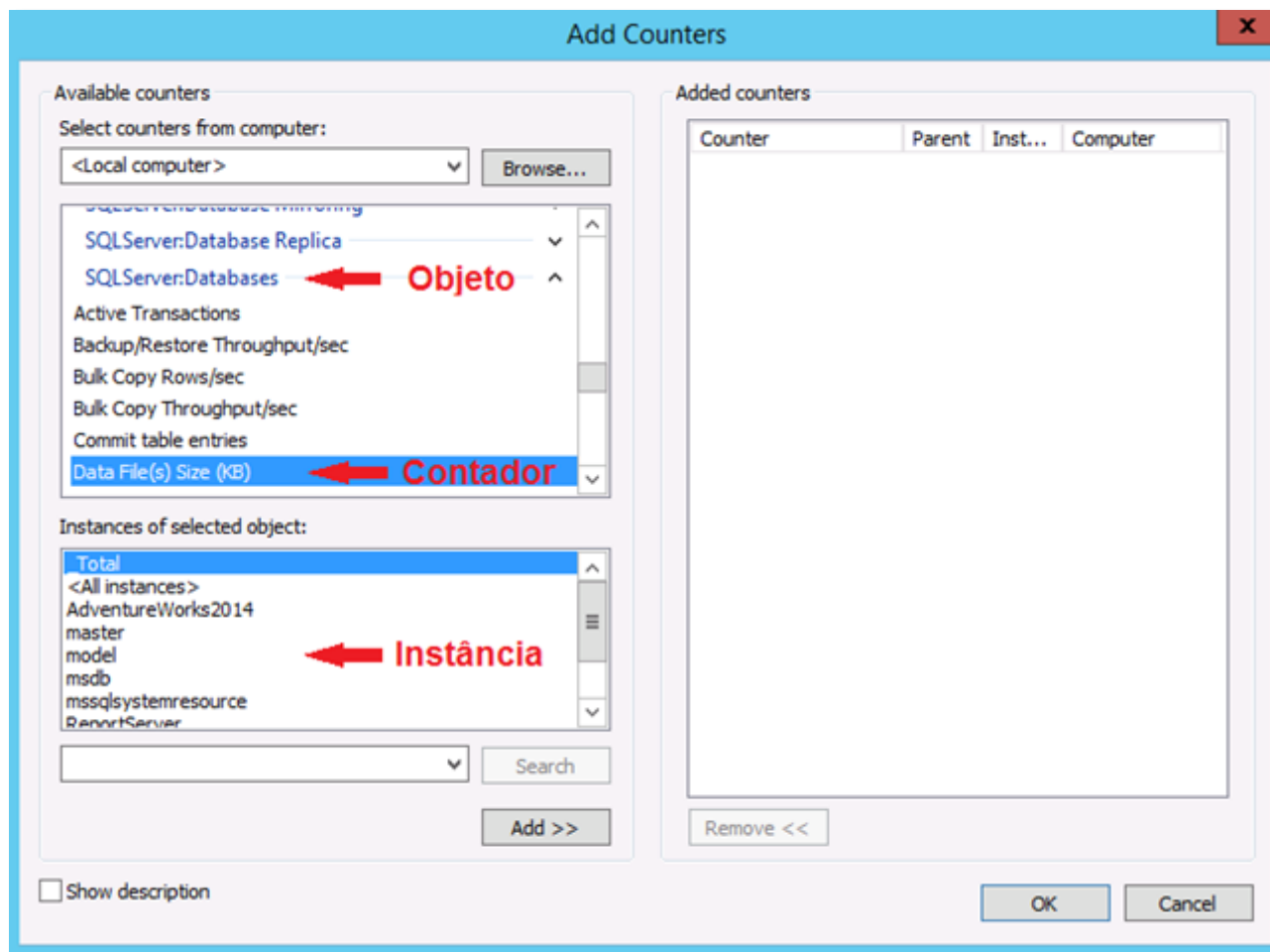


Instância nomeada "SQL2"



• Windows Performance Monitor

✓ Contadores organizados por Objeto / Contador / Instância.



✓ Contadores por Instância

- Instância Default - SQLServer:
- Instância Nomeada – MSSQL\$_____:

- **Windows Performance Monitor**

- ✓ **SQLServer: Memory Manager: Total Server Memory (KB)**

- Quantidade de memória em uso pelo SQL Server

- ✓ **SQLServer: Memory Manager: Target Server Memory (KB)**

- Quantidade de memória definida no **Max Server Memory**
 - Valor superior ao Total Server Memory significa que o SQL Server está utilizando menos memória que o configurado
 - Quando Total e Target Server Memory (KB) estão com o mesmo valor, pode indicar que o SQL Server necessite de mais memória, necessário avaliar outro contador



• Windows Performance Monitor

✓ **Memory:** Available MBytes

- Quantidade de memória RAM livre no Windows
- Não pode ser inferior a 100MB pois começa a paginar

✓ **Process:** Working Set(sqlservr)

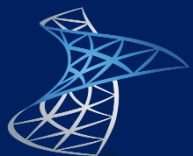
- Quantidade de memória em uso pelo processo do SQL Server
- Mantendo valor inferior ao Min e Max Server Memory, o SQL Server necessita de menos memória



• Windows Performance Monitor

✓ **SQLServer:Buffer Manager:** Page life expectancy

- Também chamado de PLE, retorna a média de tempo (em segundos) de permanência das páginas no Buffer Pool.
- Quanto maior melhor.
- Importante contador para determinar se o SQL Server está sofrendo com falta de memória.
- Utilizado como parâmetro de redução da memória do SQL Server.



- **Windows Performance Monitor**

- ✓ **SQLServer:Buffer Manager:** Page life expectancy

- O valor ideal:

$$(\text{Max Server Memory em GB}) / 4 * 300$$

- **Exemplo:** servidor com 64GB de RAM onde foi definido um teto de memória para o SQL Server de 60GB, o valor mínimo para **Page Life Expectancy** é:

$$60 / 4 * 300 = 4.500$$



• Windows Performance Monitor

✓ SQLServer:Access Methods

- **Forwarded Records/sec** – Identifica o uso de Forwarded, quanto menor a quantidade melhor. Valor de referência: < 10 por 100 Batch Request/sec.
- **Page Splits/sec** – Quantidade de Page Splits por Segundo, quando possui valor alto provoca fragmentação intensa. Valor de referência: < 20 por 100 Batch Request/sec.
- **Full Scans/sec** – Monitora ocorrências de Full Scan em Índice ou Tabela. Ocorre quando a tabela é muito pequena, quando há falta de índices ou estratégia de indexação inadequada. Quanto menor o valor melhor.
- **Index Searches/sec** – Volume de busca binária em índices, quanto maior melhor. Proporção ideal é 1 Full Scan/sec para 1000 Index Searches/sec. Valor de referência: $(\text{Index Searches/sec}) / (\text{Full Scans/sec}) > 1000$.

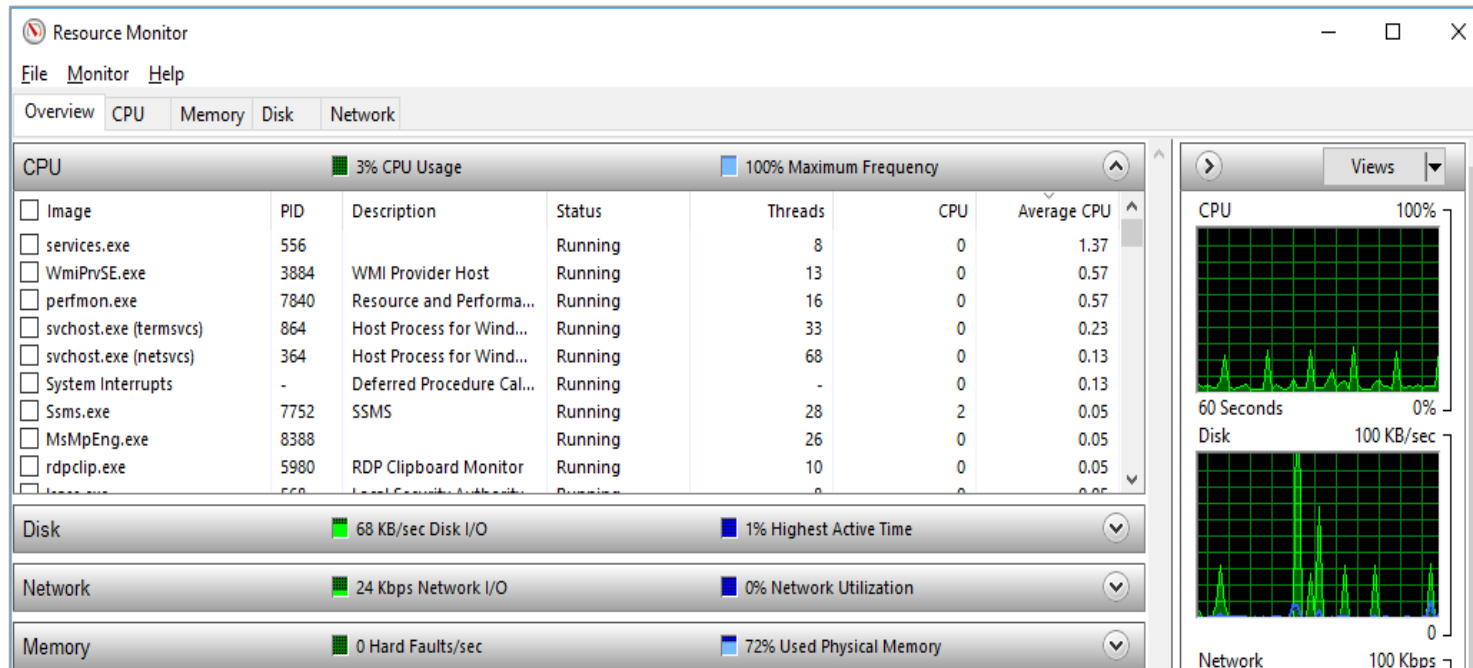


- **Hands On:** Performance Monitor



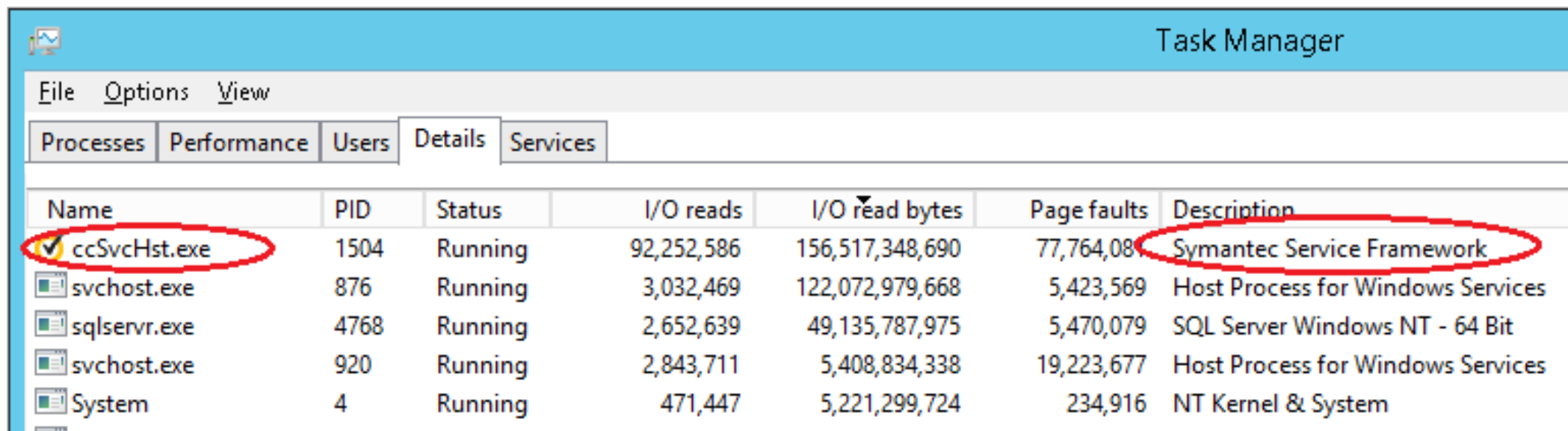
• Windows Resource Monitor

- ✓ Aplicação do Windows que fornece diversos dados importantes para análise de consumo de recursos de hardware: CPU, memória, disco e rede

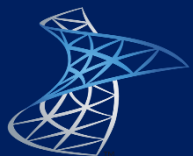


• Windows Resource Monitor

- ✓ **Exemplo:** Pode-se utilizar o **Task Manager** também para detectar a atividade de IO do antivírus, exemplo abaixo no mesmo servidor



Task Manager						
File Options View						
Processes Performance Users Details Services						
Name	PID	Status	I/O reads	I/O read bytes	Page faults	Description
✓ ccSvcHst.exe	1504	Running	92,252,586	156,517,348,690	77,764,081	Symantec Service Framework
svchost.exe	876	Running	3,032,469	122,072,979,668	5,423,569	Host Process for Windows Services
sqlservr.exe	4768	Running	2,652,639	49,135,787,975	5,470,079	SQL Server Windows NT - 64 Bit
svchost.exe	920	Running	2,843,711	5,408,834,338	19,223,677	Host Process for Windows Services
System	4	Running	471,447	5,221,299,724	234,916	NT Kernel & System



• Windows Resource Monitor

- ✓ Problema com Antivírus muito comum, relacionado a falta de filtro para arquivos que o SQL Server manipula constantemente.
- ✓ Necessário incluir filtro:
 - Arquivos de dados (.MDF e .NDF)
 - Arquivos de log (.LDF)
 - Backups (.BAK, .DIF e .TRN)
 - Traces (.XEL e .TRC), etc.



- **Hands On:** Resource Monitor



• Dynamic Management Views (DMVs)

- ✓ Views e Funções dinâmicas de sistema (DMVs).
- ✓ Retornam informações sobre a atividade em diversas áreas do servidor.
- ✓ Necessário ter permissão VIEW SERVER STATE para a instância ou VIEW DATABASE STATE para um banco de dados.
- ✓ Pode ser definido por T-SQL ou Management Studio.
- ✓ Todas as DMVs possuem o prefixo “sys.dm_...” divididas por áreas.

Categoria	Descrição
sys.dm_exec_ %	informações de conexões e execuções
sys.dm_os_ %	informações do SQL OS
sys.dm_tran_ %	informações sobre transações
sys.dm_io_ %	informações sobre IO
sys.dm_db_ %	informações de atividades em um banco de dados



• Dynamic Management Views (DMVs)

✓ Conexões e execuções

- sys.dm_exec_connections
- sys.dm_exec_sessions
- sys.dm_exec_requests
- sys.dm_exec_cached_plans
- sys.dm_exec_query_plans
- sys.dm_exec_sql_text
- sys.dm_exec_query_stats

✓ Informações de índices

- sys.dm_db_index_physical_stats
- sys.dm_db_index_usage_stats

✓ SQL OS

- sys.dm_os_performance_counters
- sys.dm_os_schedulers
- sys.dm_os_nodes
- sys.dm_os_waiting_tasks
- sys.dm_os_wait_stats

✓ I/O

- sys.dm_io_virtual_file_stats



- **Dynamic Management Views (DMVs)**

- ✓ View **sys.dm_exec_query_stats**

- Retorna as consultas em cache.
 - Utilizar com o Servidor online durante um tempo.
 - Disponível a partir do SQL Server 2008.
 - Necessário ter permissão VIEW SERVER STATE na instância.



- **Dynamic Management Views (DMVs)**

- ✓ Função **sys.dm_exec_sql_text**

- Retorna o texto da consulta.
- Recebe como parâmetro **sql_handle** da consulta.
- Disponível a partir do SQL Server 2008.

- ✓ Função **sys.dm_exec_query_plan**

- Retorna o plano de execução em formato XML.
- Recebe como parâmetro **plan_handle** da consulta.
- Disponível a partir do SQL Server 2008.



- **Hands On:** Dynamic Management Views (DMVs)



• SQL Server Profiler

- ✓ A maioria dos SGDBs disponibilizam ferramentas para monitorar ocorrências de eventos:
 - Enterprise Monitor (MySQL)
 - Event Manager (Oracle)
- ✓ No SQL Server o Profiler é a aplicação mais antiga para monitorar eventos
- ✓ Em versões futuras o Profiler será retirado, porém ainda é amplamente utilizado
- ✓ A partir do SQL Server 2012 a Microsoft criou uma nova aplicação para monitorar eventos chamada Extended Events
- ✓ O SQL Profiler pode ser utilizado de dois modos: gráfico ou T-SQL
- ✓ Indicado gerar os dados para arquivo TRC

• SQL Server Profiler

- ✓ Arquivos TRC podem ser facilmente importados para uma tabela com a função **fn_trace_gettable**

```
SELECT * INTO tmp_Trace  
FROM fn_trace_gettable('D:\Democode\Trace_QueryTempo 20141001.trc', default)
```

- ✓ Exemplo retorna as 10 consultas com maior tempo de execução:

```
SELECT top 10  
    TextData, EndTime, Duration, Reads, Writes, CPU,  
    DatabaseName, ApplicationName, HostName  
FROM tmp_Trace  
ORDER BY duration DESC
```

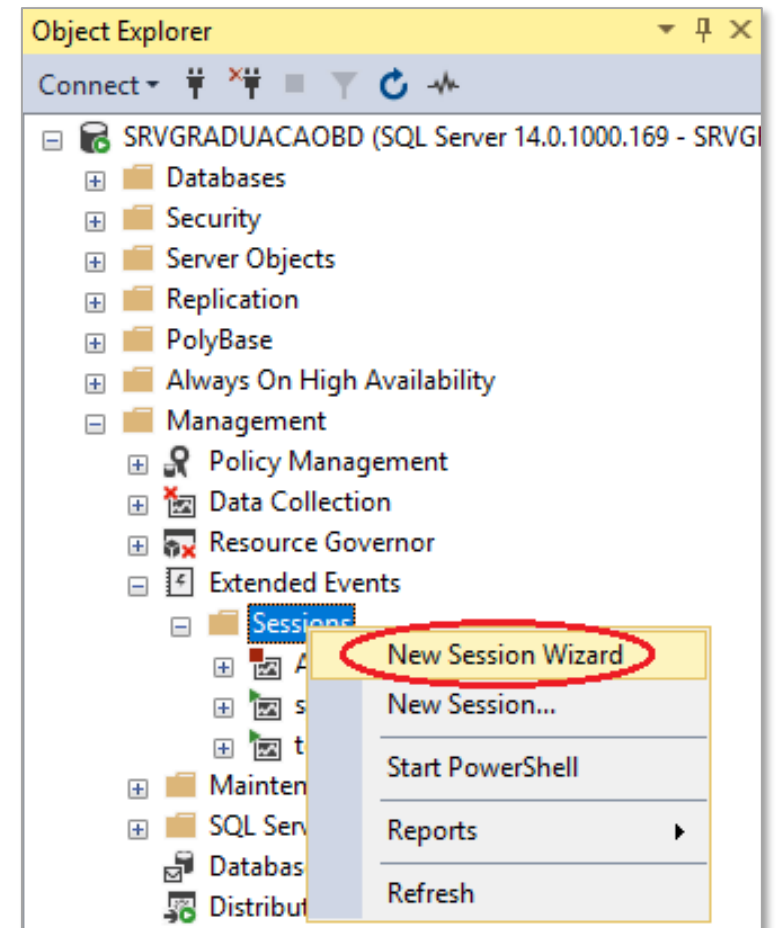


- **Hands On:** SQL Server Profiler



• Extended Events

- ✓ Novo método para monitorar eventos
- ✓ Disponível a partir do SQL Server 2012
- ✓ Bem mais leve que o SQL Server Profiler
- ✓ Pode ser definido por T-SQL ou Management Studio
- ✓ Maior flexibilidade para filtrar eventos
- ✓ Mais opções de destino



• Extended Events

✓ Para criar por T-SQL utilizar o comando CREATE EVENT

✓ Exemplo:

```
CREATE EVENT SESSION SqlStatementCompleted ON SERVER
ADD EVENT sqlserver.sql_statement_completed (ACTION
(sqlserver.sql_text,sqlserver.session_id)
WHERE server_principal_name = 'ADVENTUREWORKS\Student')
ADD TARGET package0.ring_buffer
WITH (MAX_MEMORY=4096 KB,
EVENT_RETENTION_MODE=ALLOW_SINGLE_EVENT_LOSS,
MAX_DISPATCH_LATENCY=30 SECONDS,
MAX_EVENT_SIZE=0 KB,
MEMORY_PARTITION_MODE=NONE,
TRACK_CAUSALITY=OFF,
STARTUP_STATE=OFF)
```



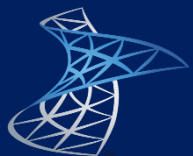
• Extended Events

- ✓ Exemplo coleta de consultas.
 - Configuração da coleta.

```
CREATE EVENT SESSION exConsultas ON SERVER
ADD EVENT sqlserver.sql_statement_completed (WHERE ([sqlserver].[database_name]=N'AdventureWorks'
and duration >= 50000)),
ADD EVENT sqlserver.rpc_completed (WHERE ([sqlserver].[database_name]=N'AdventureWorks'
and duration >= 50000))
ADD TARGET package0.event_file (SET filename=N'D:\Aulas\Consultas.xel',max_file_size=(2),
max_rollover_files=(100))
WITH (STARTUP_STATE=ON)
```

- Iniciando a coleta

```
ALTER EVENT SESSION exConsultas ON SERVER STATE = START
```



• Extended Events

✓ Exemplo coleta de consultas.

○ Consultando o arquivo .XEL

```
SELECT q.duration as TempoExec,q.cpu_time as CPU,q.logical_reads as Leituras,
q.writes as Escritas,
statement as Comando,timestamp as DataHoraUTC,
DATEADD(hh, DATEDIFF(hh, GETUTCDATE(), CURRENT_TIMESTAMP),timestamp) as DataHoraLocal,
event_data_XML as DocXML
FROM (
SELECT duration=e.event_data_XML.value('(/*data[@name="duration"]/value)[1]','int'),
cpu_time=e.event_data_XML.value('(/*data[@name="cpu_time"]/value)[1]','int'),
physical_reads=e.event_data_XML.value('(/*data[@name="physical_reads"]/value)[1]','int'),
logical_reads=e.event_data_XML.value('(/*data[@name="logical_reads"]/value)[1]','int'),
writes=e.event_data_XML.value('(/*data[@name="writes"]/value)[1]','int'),
statement=e.event_data_XML.value('(/*data[@name="statement"]/value)[1]','nvarchar(max)'),
TIMESTAMP=e.event_data_XML.value('(/*@timestamp)[1]','datetime2(7)'),*
FROM (
SELECT CAST(event_data AS XML) AS event_data_XML
FROM sys.fn_xe_file_target_read_file('D:\Aulas\Consultas*.xel', NULL, NULL, NULL)) as e
) as q
ORDER BY duration desc
```



- **Hands On:** Extended Event





Live #009

Carga de Dados no Microsoft SQL Server



E-mail: proflandry.sqlexpert@gmail.com



Youtube: @prof-landrySQLServerExpert



Instagram: @sqlserver.expert



Facebook: @sqlserver.expert.landry