MINDWORKS conexões inteligentes

A Empresa

- 14 anos de mercado, Vitória – ES
- 5 anos da inauguração da filial, Campinas - SP
- Forte atuação no mercado, público e privado
- Parceiros **estratégicos**
- Soluções **completas**
- Portfólio abrangente com foco em **resultados**
- Sistema de **Qualidade** aprimorado
- Crescimento **consistente** nos últimos 5 anos

conexões
inteligentes

MINDWORKS

Principais Clientes

Setor Privado	Setor Publico
ArcelorMittal, Garoto, BRASCOBRA, edp, CLARK THE FORKLIFT, GRUPO ÁGUIA BRANCA, Fibria, VALE, SEW EURODRIVE, Unimed Vitória	FINDES CINDES SESI SENAI IEL IDEIES, Sistema FINDES, Ministério Público Estado do Espírito Santo, TCE ES Tribunal de Contas do Estado do Espírito Santo, SEFAZ ES Secretaria de Estado da Fazenda, Prefeitura Jundiaí Oportunidade é para todos, Secretaria de Gestão e Recursos Humanos, Governo do Espírito Santo Crescer é com a gente, TJES, Prodest Tecnologia da Informação

Soluções em TI
- Serviços
- Infraestrutura
- Treinamento

conexões
inteligentes

Parceiros

Microsoft
GOLD CERTIFIED
Partner

MINDWORKS

- Learning
- Collaboration and Content
- Communications
- Devices and Deployment
- Identity and Access
- Management and Virtualization
- Messaging
- Server Platform

Soluções em TI
• Serviços
• Infraestrutura
• Treinamento

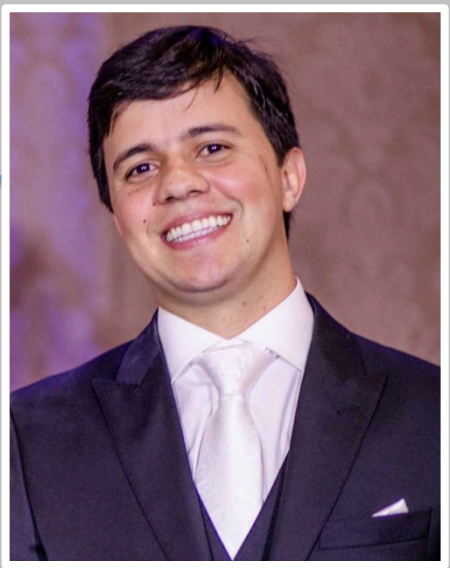
Dynamic Management Views and Functions

Vida de DBA SQL Server

conexões
inteligentes

MINDWORKS

Sobre mim



- Vithor da Silva e Silva
- 28 anos
- DBA na Accenture
- Consultor independente – VSS TI
- Instrutor na Mindworks
- Certificações
  - IBM DB2 9.7 LWU Associate;
  - Microsoft Certified Trainer;
- Sites
  - http://www.linkedin.com/in/vithorsilva
  - http://www.vssti.com.br/
  - http://facebook.com/sqles
- **Contato: vithor@vssti.com.br**

Microsoft CERTIFIED Trainer MCT

Soluções em TI
- Serviços
- Infraestrutura
- Treinamento

conexões
inteligentes

MINDWORKS

Agenda

- "Tá lento!"
- A história das DMVs;
- Categorias de DMVs;
- DMVs: Executions;
- DMVs: I/O;
- DMVs: Index;
- DMVs: SQL Server OS

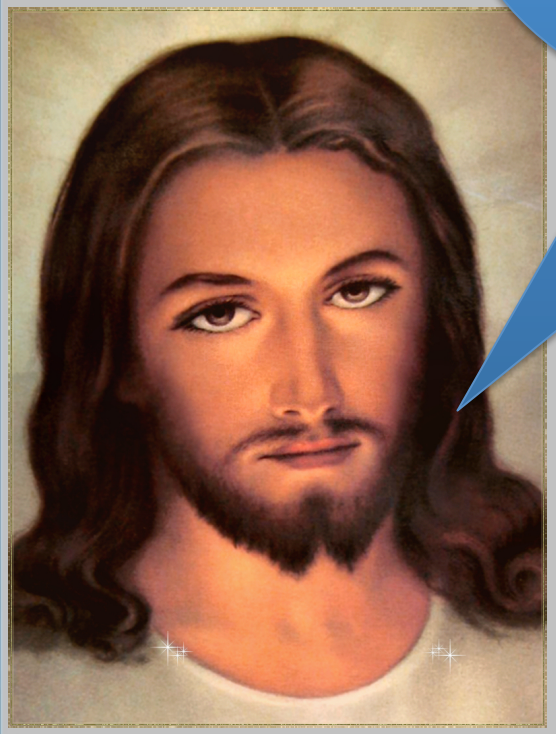
Soluções em TI
• Serviços
• Infraestrutura
• Treinamento

conexões
inteligentes

Tá lento!

conexões
inteligentes

Tá lento!



"Jesus, porém, respondendo, disse-lhes: Errais, não conhecendo as Escrituras..."
Mateus 22:29

MINDWORKS

Soluções em TI
- Serviços
- Infraestrutura
- Treinamento

conexões
inteligentes

A história das DMVs

Microsoft SQL Server

- Lançado no SQL Server 2005;
- **Dynamic Management Objects**: Fornecem informações sobre o funcionamento interno do banco de dados sem a necessidade de usar procedimentos armazenados estendidos.
- Objetos criados como *views*, são chamados **DMVs**.
- Objetos criados como *functions,* são chamados de **DMFs**.

MINDWORKS

Soluções em TI
• Serviços
• Infraestrutura
• Treinamento

conexões
inteligentes

A história das DMVs

Microsoft SQL Server

- Em constante evolução:
  - SQL Server 2014:
    - 34 - Dynamic Management Functions;
    - 166 – Dynamic Management Views;
- As informações apresentadas pelos DMOs são baseados na carga de trabalho da instância, e são de grande contribuição para solução de problemas dos DBAs;

```
--Listar Quantidade de DMVs (Total)
SELECT  [type], COUNT(*)
FROM    [master].[sys].[sysobjects]
where   [name] LIKE 'dm_%'
GROUP BY [type];

--Listar todas as DMVs (Total)
SELECT *
FROM    [master].[sys].[sysobjects]
where   [name] LIKE 'dm_%'
ORDER BY name asc;
```

MINDWORKS

Soluções em TI
- Serviços
- Infraestrutura
- Treinamento

conexões inteligentes

Categorias de DMVs

Microsoft SQL Server

- Todas as DMVs e DMFs existem no *schema* **sys** e seguem iniciando com um nome por **dm_%**.
  - **Server-scoped**: Esses objetos fornecem informações de todo servidor;
  - **Database-scoped**: Estes objetos fornecem informações específicas do banco de dados.

AlwaysOn Availability Group	I/O
Change Data Capture	Memory-Optimized Table
Change Tracking	Object
Common Language Runtime	Query Notifications
Database Mirroring	Replication
Database	Resource Governor
Execution	Security
Extended Events	Service Broker
Filestream and FileTable	SQL Server OS
Full-Text Search and Semantic Search	Transaction
Index	

MINDWORKS

Soluções em TI
• Serviços
• Infraestrutura
• Treinamento

conexões
inteligentes

DMVs: Executions

Microsoft SQL Server

Objeto	Utilidade
sys.dm_exec_connections	Retorna informações sobre as conexões estabelecidas com a instância e os detalhes de cada conexão.
sys.dm_exec_sessions	Retorna uma linha por sessão autenticada no SQL Server e é uma exibição que mostra informações sobre todas as conexões dos usuários ativos e tarefas internas. Contêm informações sobre versão do client, o nome do programa cliente, a hora do logon do cliente, o usuário de logon, a configuração da sessão atual, etc.
sys.dm_exec_requests	Retorna informações sobre cada solicitação sendo executada no SQL Server.
sys.dm_exec_sql_text	Retorna o texto do batch SQL que é identificado pelo sql_handle especificado.
sys.dm_exec_query_plan	Retorna o plano de execução em formato XML. O plano especificado pelo identificador do plano pode estar em cache ou sendo executado.

MINDWORKS

Soluções em TI
- Serviços
- Infraestrutura
- Treinamento

Demo: Usuários e aplicativos conectados

```
-- Identificar sessões, origem das conexões e o aplicativo utilizado

SELECT ec.client_net_address, es.[program_name], es.[host_name], es.login_name,
COUNT(ec.session_id) AS [connection count]
FROM sys.dm_exec_sessions AS es WITH (NOLOCK)
INNER JOIN sys.dm_exec_connections AS ec WITH (NOLOCK)
ON es.session_id = ec.session_id
GROUP BY ec.client_net_address, es.[program_name], es.[host_name], es.login_name
ORDER BY [connection count] DESC OPTION (RECOMPILE);
--ORDER BY ec.client_net_address, es.[program_name] OPTION (RECOMPILE);
```

Results | Messages

	client_net_address	program_name	host_name	login_name	connection count
1	<local machine>	Report Server	SVRELALA01	SVRELALA01\mysolutions	3
2	172.16.20.10	mySDK	SVRELALA01	mysolutions	3
3	<local machine>	Microsoft SQL Server Management Studio	SVRELALA01	SVRELALA01\mysolutions	2
4	<local machine>	Microsoft SQL Server Management Studio - Query	SVRELALA01	SVRELALA01\mysolutions	1
5	172.16.20.10	SiGO	SVRELALA01	mysolutions	1
6	<local machine>	SQLAgent - Alert Engine	SVRELALA01	SVRELALA01\mysolutions	1
7	<local machine>	SQLAgent - Email Logger	SVRELALA01	SVRELALA01\mysolutions	1
8	<local machine>	SQLAgent - Generic Refresher	SVRELALA01	SVRELALA01\mysolutions	1
9	<local machine>	SQLAgent - Job invocation engine	SVRELALA01	SVRELALA01\mysolutions	1
10	<local machine>	DatabaseMail - SQLAGENT - Id<2004>	SVRELALA01	SVRELALA01\mysolutions	1

Demo: Análise de bloqueios (*locks*)

```
SELECT t1.resource_type                    AS [lock type],
       Db_name(resource_database_id)          AS [database],
       t1.request_mode                    AS [lock req],
       t1.request_session_id               AS [waiter sid],
       t2.wait_duration_ms                AS [wait time],
       (SELECT [text]
        FROM   sys.dm_exec_requests AS r WITH (nolock)
               CROSS apply sys.Dm_exec_sql_text(r.[sql_handle])
        WHERE  r.session_id = t1.request_session_id) AS [waiter_batch],
       (SELECT Substring(qt.[text], r.statement_start_offset / 2,
                 ( CASE WHEN r.statement_end_offset = -1
                 THEN Len(CONVERT(NVARCHAR(max), qt.[text]))*2  ELSE r.statement_end_offset
                 END - r.statement_start_offset ) / 2)
        FROM   sys.dm_exec_requests AS r WITH (nolock)
               CROSS apply sys.Dm_exec_sql_text(r.[sql_handle]) AS qt
        WHERE  r.session_id = t1.request_session_id) AS [waiter_stmt],
       t2.blocking_session_id              AS [blocker sid],
       (SELECT [text] FROM   sys.sysprocesses AS p
                 CROSS apply sys.Dm_exec_sql_text(p.[sql_handle])
        WHERE  p.spid = t2.blocking_session_id)     AS [blocker_stmt]
FROM   sys.dm_tran_locks AS t1 WITH (nolock)
INNER JOIN sys.dm_os_waiting_tasks AS t2 WITH (nolock)  ON t1.lock_owner_address = t2.resource_address
OPTION (recompile);
```

Results | Messages

	lock type	database	lock req	waiter sid	wait time	waiter_batch	waiter_stmt	blocker sid	blocker_stmt
1	OBJECT	mySolutionsMonitor	X	73	238186	insert bulk [SQLSERVER].[tbSe...	insert bulk [SQLSERVER].[tbS...	61	INSERT INTO [mySolutionsMonit...
2	OBJECT	mySolutionsMonitor	X	73	238186	insert bulk [SQLSERVER].[tbSe...	insert bulk [SQLSERVER].[tbS...	70	SELECT COUNT(*) FROM SQLSE...
3	PAGE	mySolutionsMonitor	S	70	291989	SELECT COUNT(*) FROM SQL...	SELECT COUNT(*) FROM SQ...	61	INSERT INTO [mySolutionsMonit...

Demo: Utilização de CPU x Database

```
- Auxilia na identificação do banco de dados, que utilizam por mais tempo recursos de processamento (CPU)
WITH DB_CPU_Stats
AS
(SELECT DatabaseID, DB_Name(DatabaseID) AS [Database Name], SUM(total_worker_time) AS [CPU_Time_Ms]
 FROM sys.dm_exec_query_stats AS qs
 CROSS APPLY (
          SELECT CONVERT(int, value) AS [DatabaseID]
           FROM sys.dm_exec_plan_attributes(qs.plan_handle)
           WHERE attribute = N'dbid') AS F_DB
 GROUP BY DatabaseID)

SELECT ROW_NUMBER() OVER(ORDER BY [CPU_Time_Ms] DESC) AS [CPU Rank],
     [Database Name], [CPU_Time_Ms] AS [CPU Time (ms)],
     CAST([CPU_Time_Ms] * 1.0 / SUM([CPU_Time_Ms]) OVER() * 100.0 AS DECIMAL(5, 2)) AS [CPU Percent]
FROM DB_CPU_Stats
WHERE DatabaseID <> 32767 -- ResourceDB
ORDER BY [CPU Rank] OPTION (RECOMPILE);
```

Results | Messages

	CPU Rank	Database Name	CPU Time (ms)	CPU Percent
1	1	ReportServer	6141958765	62.19
2	2	mySolutionsdb	3612967846	36.58
3	3	msdb	65185752	0.66
4	4	ReportServerTempDB	51062094	0.52
5	5	master	5067936	0.05
6	6	mySolutionsMonitor	248842	0.00

Demo: TOP SQL por duração

```
SELECT TOP(50)  t.[text] AS [Query Text],
                qs.total_worker_time/1000000 AS [TotalWorkerTime],
                qs.min_worker_time/1000000 AS [MinWorkerTime],
                (qs.max_worker_time)/1000000 AS [MaxWorkerTime],
                (qs.total_worker_time/qs.execution_count)/1000000 AS [AvgWorkerTime],
                qs.execution_count AS [Execution Count],
                qp.query_plan AS [Query Plan]
 FROM sys.dm_exec_query_stats AS qs WITH (NOLOCK)
CROSS APPLY sys.dm_exec_sql_text(plan_handle) AS t
CROSS APPLY sys.dm_exec_query_plan(plan_handle) AS qp
where DB_NAME(t.[dbid]) NOT IN ('ReportServer', 'master', 'msdb') OR DB_NAME(t.[dbid]) IS NULL
ORDER BY qs.total_worker_time DESC OPTION (RECOMPILE);
```

	Query Text	TotalWorkerTime	MinWorkerTime	MaxWorkerTime	AvgWorkerTime	Execution Count	Query Plan
1	(@pCodigoOcorrencia nvarchar(6))SELECT ID, Na...	2874	0	18	0	91950	<ShowPlanXML xmlns="http://schem
2	(@pCodigoUsuario int)SELECT codigo_ocorrenci...	348	1	16	1	228	<ShowPlanXML xmlns="http://schem
3	(@pCodigoCliente int,@pDataInicial datetime,@pCodi...	326	0	146	0	663	<ShowPlanXML xmlns="http://schem
4	PRINT 'START: Report Server Tempdb - Clean'; PRI...	51	0	26	0	94	<ShowPlanXML xmlns="http://schem
5	(@pDataInicial datetime,@pDataFinal datetime,@pCo...	31	0	3	0	197	<ShowPlanXML xmlns="http://schem

Demo: TOP SQL por duração

ExecutionPlan1.sqlplan* | SQLQuery2.sql - (local)\...\mysolutio...* | SQLQuery1.sql - (local)\...\mys... (55))* | Object Explorer Details

Query 1: Query cost (relative to the batch): 100%
(@pDataInicial datetime,@pDataFinal datetime,@pCodigoCliente int)SELECT codigo_ocorrencia, data_ocorrencia, day(data…
Missing Index (Impact 98.5426): CREATE NONCLUSTERED INDEX [<Name of Missing Index, sysname,>] ON [SiGO].[tbOcorrenci…

Clustered Index Seek (Clustered) [tbClientes].[PK_tbClientes] Cost: 0 %

Nested Loops (Inner Join) Cost: 0 %

Sort Cost: 0 %

Nested Loops (Inner Join) Cost: 0 %

Parallelism (Repartition Streams) Cost: 0 %

Compute Scalar Cost: 2 %

Clustered Index Scan (Clustered) [tbOcorrencias].[PK_tbOcorrencias] Cost: 97 %

Clustered Index Seek (Clustered) [tbEmpresas].[PK_tbEmpresas] Cost: 0 %

Clustered Index Seek (Clustered) [tbTipoImpacto].[PK_tbTipoImpacto] Cost: 0 %

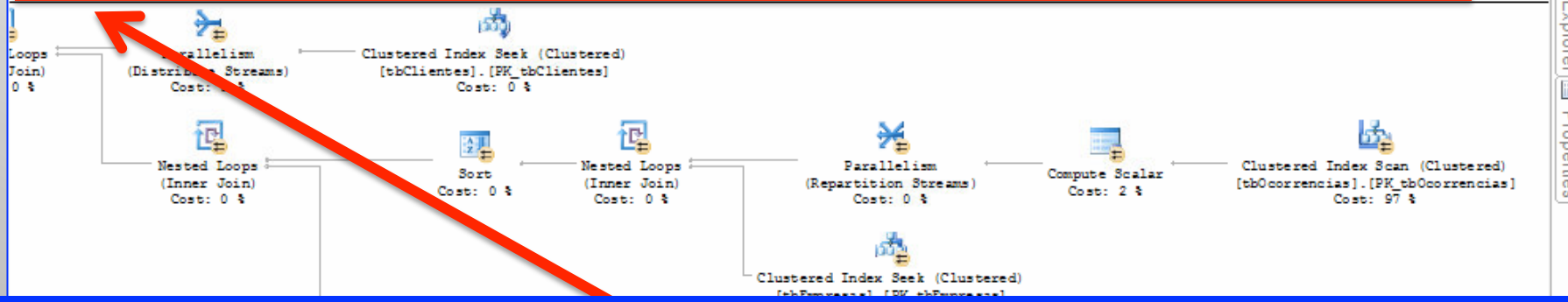
- Save Execution Plan As...
- Show Execution Plan XML...
- Edit Query Text...
- Missing Index Details...
- Zoom In
- Zoom Out
- Custom Zoom...
- Zoom to Fit
- Properties

Demo: TOP SQL por duração

Microsoft SQL Server

ExecutionPlan1.sqlplan* | SQLQuery2.sql - (local)\...\mysolutio...* | SQLQuery1.sql - (local)\...\mys... (55))* | Object Explorer Details

```
Query 1: Query cost (relative to the batch): 100%
(@pDataInicial datetime,@pDataFinal datetime,@pCodigoCliente int)SELECT codigo_ocorrencia, data_ocorrencia, day(data…
Missing Index (Impact 98.5426): CREATE NONCLUSTERED INDEX [<Name of Missing Index, sysname,>] ON [SiGO].[tbOcorrenci…
```



SQLQuery4.sql - (local)\...\mysolutio...* | ExecutionPlan1.sqlplan* | SQLQuery2.sql - (local)\...\mysolutio...* | SQLQuery1.sql - (local)\...\mys... (55))*

```
/*
Missing Index Details from ExecutionPlan1.sqlplan
The Query Processor estimates that implementing the following index could improve the query cost by 98.5426%.
*/

/*
USE [mySolutionsdb]
GO
CREATE NONCLUSTERED INDEX [<Name of Missing Index, sysname,>]
ON [SiGO].[tbOcorrencias] ([codigo_cliente],[data_ocorrencia])
INCLUDE ([[codigo_empresa])
GO
*/
```

conexões
inteligentes

DMVs: I/O

Microsoft SQL Server

Objeto	Utilidade
sys.dm_db_partition_stats	Retorna o espaço em disco e estatísticas (quantidade de linhas, quantidade de páginas, e assim por diante) para cada objeto em uma partição
sys.dm_db_index_physical_stats	Fornece estatísticas sobre o tamanho e fragmentação dos *clustered index* e *non clustered index* e *heaps*.
sys.dm_io_virtual_file_stats	Fornece estatísticas sobre a forma como todos os arquivos de banco de dados estão sendo utilizados; Excelente recurso para descobrir problemas, e identificação de oportunidades em "espalhar" a carga de I/O em *drivers*.
sys.dm_io_pending_io_requests	Fornece uma lista de todas as operações I/O que o SQL Server está esperando para completar.
sys.dm_db_file_space_usage	Fornece informações sobre como tempdb está sendo utilizado internamente.

MINDWORKS

Soluções em TI
- Serviços
- Infraestrutura
- Treinamento

Demo: Estatísticas atualizadas?

```
SELECT Schema_name(o.schema_id) + N'.' + o.NAME          AS [Object Name],
       o.type_desc                                       AS [Object Type],
       i.NAME                                            AS [Index Name],
       Stats_date(i.[object_id], i.index_id)              AS [Statistics Date],
       s.auto_created, s.no_recompute, s.user_created,
       st.row_count, st.used_page_count
FROM   sys.objects AS o WITH (nolock)
       INNER JOIN sys.indexes AS i WITH (nolock)          ON o.[object_id] = i.[object_id]
       INNER JOIN sys.stats AS s WITH (nolock)            ON i.[object_id] = s.[object_id] AND i.index_id = s.stats_id
       INNER JOIN sys.dm_db_partition_stats AS st WITH (nolock) ON o.[object_id] = st.[object_id] AND i.[index_id] = st.[index_id]
WHERE  o.[type] IN ( 'U', 'V' ) AND st.row_count > 0
ORDER  BY stats_date(i.[object_id], i.index_id) DESC
OPTION (recompile);
```

Results | Messages

	Object Name	Object Type	Index Name	Statistics Date	auto_created	no_recompute	user_created	row_count	used_page_count
1	SiGO.tbTipoO...	USER_TABLE	PK_tbTipoOcorrencias	2015-01-23 22:05:17.423	0	0	0	85	4
2	SiGO.tbTipoI...	USER_TABLE	PK_tbTipoImpacto	2015-01-23 22:05:17.420	0	0	0	3	2
3	SiGO.tbTipoC...	USER_TABLE	PK_tbTipoCriticidade	2015-01-23 22:05:17.420	0	0	0	4	2
4	SiGO.tbSiGO...	USER_TABLE	PK_tbSiGORegistro	2015-01-23 22:05:17.420	0	0	0	1	4
5	SiGO.tbPDCA	USER_TABLE	PK_tbPDCA	2015-01-23 22:05:17.413	0	0	0	1	2
6	SiGO.tbOcorr...	USER_TABLE	PK_tbOcorrenciasLog	2015-01-23 22:05:17.280	0	0	0	2117	5
7	SiGO.tbOcorr...	USER_TABLE	IX_tbOcorrenciasLog	2015-01-23 22:05:17.277	0	0	0	2117	24
8	SiGO.tbOcorr...	USER_TABLE	PK_tbOcorrencias	2015-01-23 22:05:10.237	0	0	0	389039	13612
9	SiGO.tbMotivos	USER_TABLE	PK_tbMotivos	2015-01-23 22:05:10.110	0	0	0	1	2
10	SiGO.tbChec...	USER_TABLE	PK_tbChecklist	2015-01-23 22:05:10.107	0	0	0	10	2
11	SiGO.tbAutores	USER_TABLE	PK_tbAutores	2015-01-23 22:05:10.107	0	0	0	26	2
12	SiGE.tbEmpre...	USER_TABLE	PK_tbEmpresas	2015-01-23 22:05:10.100	0	0	0	1	2
13	SiGE.tbClientes	USER_TABLE	PK_tbClientes	2015-01-23 22:05:10.053	0	0	0	30	4
14	MYSOLUTIO...	USER_TABLE	IX_tbImagens_codigo_ocorrencia	2015-01-23 22:05:10.050	0	0	0	792381	1716
15	MYSOLUTIO...	USER_TABLE	PK_tbImagens	2015-01-23 22:05:08.640	0	0	0	792381	15680310

Demo: Índices Fragmentados?

```
SELECT          DB_NAME(ps.database_id) AS [Database Name], OBJECT_NAME(ps.OBJECT_ID) AS [Object Name],
                i.name AS [Index Name],
                ps.index_id, ps.index_type_desc,
                ps.avg_fragmentation_in_percent,
                ps.fragment_count, ps.page_count
                --, i.fill_factor, i.has_filter, i.filter_definition
FROM sys.dm_db_index_physical_stats(DB_ID(),NULL, NULL, NULL , N'LIMITED') AS ps
INNER JOIN sys.indexes AS i WITH (NOLOCK)
ON ps.[object_id] = i.[object_id] AND ps.index_id = i.index_id
WHERE ps.database_id = DB_ID() AND ps.page_count > 2500
ORDER BY ps.avg_fragmentation_in_percent DESC OPTION (RECOMPILE);
```

Results | Messages

	Database Name	Object Name	Index Name	index_id	index_type_desc	avg_fragmentation_in_percent	fragment_count	page_count
1	mySolutionsdb	tbImagens	PK_tbImagens	1	CLUSTERED INDEX	29,2011635712687	1539	4469
2	mySolutionsdb	tbOcorrencias	PK_tbOcorrencias	1	CLUSTERED INDEX	22,0972130303699	3060	13599

avg_fragmentation_in_percent value	Corrective statement
> 5% and < = 30%	ALTER INDEX REORGANIZE
> 30%	ALTER INDEX REBUILD WITH (ONLINE = ON)*

https://msdn.microsoft.com/en-us/library/ms189858.aspx

Demo: Meu disco está lento?

```
SELECT [drive],
     CASE WHEN num_of_reads = 0 THEN 0 ELSE ( io_stall_read_ms / num_of_reads )END                    AS [Read Latency],
     CASE WHEN io_stall_write_ms = 0 THEN 0 ELSE ( io_stall_write_ms / num_of_writes )END             AS [Write Latency],
     CASE WHEN ( num_of_reads = 0 AND num_of_writes = 0 ) THEN 0
                                   ELSE (io_stall / ( num_of_reads + num_of_writes))END
                                                                                                      AS [Overall Latency],
     CASE WHEN num_of_reads = 0 THEN 0 ELSE ( num_of_bytes_read / num_of_reads) END                   AS [Avg Bytes/Read],
     CASE WHEN io_stall_write_ms = 0 THEN 0 ELSE ( num_of_bytes_written / num_of_writes ) END         AS [Avg Bytes/Write],
     CASE WHEN ( num_of_reads = 0 AND num_of_writes = 0 ) THEN 0
                 ELSE ((num_of_bytes_read + num_of_bytes_written) / (num_of_reads + num_of_writes))END AS [Avg Bytes/Transfer]
FROM   (SELECT LEFT(Upper(mf.physical_name), 2) AS Drive,
            Sum(num_of_reads)            AS num_of_reads,
            Sum(io_stall_read_ms)        AS io_stall_read_ms,
            Sum(num_of_writes)           AS num_of_writes,
            Sum(io_stall_write_ms)       AS io_stall_write_ms,
            Sum(num_of_bytes_read)       AS num_of_bytes_read,
            Sum(num_of_bytes_written)    AS num_of_bytes_written,
            Sum(io_stall)                AS io_stall
      FROM  sys.dm_io_virtual_file_stats(NULL, NULL) AS vfs
      INNER JOIN sys.master_files AS mf WITH (nolock)
                                        ON vfs.database_id = mf.database_id AND vfs.file_id = mf.file_id
GROUP  BY LEFT(Upper(mf.physical_name), 2)) AS tab
ORDER  BY [overall latency] OPTION (recompile);
```

Results | Messages

	drive	Read Latency	Write Latency	Overall Latency	Avg Bytes/Read	Avg Bytes/Write	Avg Bytes/Transfer
1	E:	8	22	11	25270	74089	35049

Glenn Berry diz: *"Latency above 20-25ms is usually a problem"*

Demo: Será mesmo o disco?

```
SELECT          DB_NAME(mf.database_id) AS [Database],
                mf.physical_name,
                --r.*,
                r.io_pending, r.io_pending_ms_ticks, r.io_type,
                fs.num_of_reads, fs.num_of_writes
FROM sys.dm_io_pending_io_requests AS r
INNER JOIN sys.dm_io_virtual_file_stats(null,null) AS fs
                ON r.io_handle = fs.file_handle
INNER JOIN sys.master_files AS mf
                ON fs.database_id = mf.database_id AND fs.file_id = mf.file_id
ORDER BY r.io_pending, r.io_pending_ms_ticks DESC;
```

Editor | Results | Messages

	Database	physical_name	io_pending	io_pending_ms_ticks	io_type	num_of_reads	num_of_writes
1	AdventureWorks	C:\Program Files\Microsoft SQL Server\MSSQL1...	1	80	disk	255	1
2	AdventureWorks	C:\Program Files\Microsoft SQL Server\MSSQL1...	1	79	disk	255	1
3	AdventureWorks	C:\Program Files\Microsoft SQL Server\MSSQL1...	1	79	disk	255	1

Results | Messages

	Database	physical_name	io_pending	io_pending_ms_ticks	io_type	num_of_reads	num_of_writes
1	ReportServerTempDB	E:\Database\PRODUCAO\DATA\ReportServerTempDB.mdf	1	58	disk	131809	89486
2	ReportServerTempDB	E:\Database\PRODUCAO\DATA\ReportServerTempDB.mdf	1	58	disk	131809	89486
3	ReportServerTempDB	E:\Database\PRODUCAO\DATA\ReportServerTempDB.mdf	1	58	disk	131809	89486
4	ReportServerTempDB	E:\Database\PRODUCAO\DATA\ReportServerTempDB.mdf	1	58	disk	131809	89486
5	ReportServerTempDB	E:\Database\PRODUCAO\DATA\ReportServerTempDB.mdf	1	58	disk	131809	89486

conexões
inteligentes

DMVs: Index

Microsoft SQL Server

- **Existem índices que não estão mais em uso, ou nunca foram utilizados?**
  - sys.dm_db_index_usage_stats;
- **Para índices em uso, qual é o padrão de uso?**
  - sys.dm_db_index_operational_stats;
- **Quais índices estão faltando?**
  - sys.dm_db_missing_index_details;
  - sys.dm_db_missing_index_group_stats;

MINDWORKS

Soluções em TI
• Serviços
• Infraestrutura
• Treinamento

Demo: Como estão sendo utilizados os índices?

Microsoft SQL Server

```
SELECT          OBJECT_NAME(s.[object_id]) AS [Table Name],
                                --i.name AS [Index Name],
                                i.index_id,
                                --i.is_disabled, i.is_hypothetical, i.has_filter, i.fill_factor,
                                user_updates, user_seeks, user_scans, user_lookups,
                                user_seeks + user_scans + user_lookups AS [TotalReads],
                                user_updates AS [TotalWrites],
                                user_updates - (user_seeks + user_scans + user_lookups) AS [Difference],
                                s.last_user_scan, s.last_user_seek, s.last_user_lookup, s.last_user_update
FROM sys.dm_db_index_usage_stats AS s WITH (NOLOCK)
INNER JOIN sys.indexes AS i WITH (NOLOCK)
ON s.[object_id] = i.[object_id]
AND i.index_id = s.index_id
WHERE OBJECTPROPERTY(s.[object_id],'IsUserTable') = 1
AND s.database_id = DB_ID()
--AND user_updates > (user_seeks + user_scans + user_lookups)
AND i.index_id > 1
ORDER BY [Difference] DESC, [TotalWrites] DESC, [TotalReads] ASC OPTION (RECOMPILE);
```

Results | Messages

	Table Name	index_id	user_updates	user_seeks	user_scans	user_lookups	TotalReads	TotalWrites	Difference
1	Notifications	3	2784	0	31	0	31	2784	2753
2	Event	2	2753	0	31	0	31	2753	2722
3	ExecutionLogStorage	2	476	0	0	0	0	476	476
4	ModelItemPolicy	2	90	0	0	0	0	90	90

Results | Messages

	Table Name	index_id	TotalReads	TotalWrites	Difference	last_user_scan	last_user_seek	last_user_lookup	last_user_update
1	tbOcorrenciasLog	2	0	28	28	NULL	NULL	NULL	2015-01-25 12:13:01.170
2	tbImagens	4	1883	475	-1408	NULL	2015-01-25 12:32:19.047	NULL	2015-01-25 12:31:22.133

Demo: Determinando o volume de Insert, Update e Deletes

```
SELECT --DB_NAME([database_id]) AS [Database]
   --,iops.[object_id] AS [ObjectID]
   QUOTENAME(OBJECT_SCHEMA_NAME(iops.[object_id], [database_id])) + N'.' + QUOTENAME(OBJECT_NAME(iops.[object_id],
[database_id])) AS [ObjectName]
   ,i.[index_id] AS [IndexID]
   ,i.[name] AS [IndexName]
   --,i.[fill_factor] AS [IndexFillFactor]
   --,iops.[partition_number] AS [PartitionNumber]
   ,CASE WHEN i.[is_unique] = 1 THEN 'UNIQUE '  ELSE '' END + i.[type_desc] AS [IndexType]
   ,iops.[leaf_insert_count] AS [LeafInsertCount]
   ,iops.[leaf_delete_count] AS [LeafDeleteCount]
   ,iops.[leaf_update_count] AS [LeafUpdateCount]
FROM [sys].[dm_db_index_operational_stats](DB_ID(), NULL, NULL, NULL) iops
INNER JOIN [sys].[indexes] i
   ON i.[object_id] = iops.[object_id]
      AND i.[index_id] = iops.[index_id]
WHERE OBJECT_SCHEMA_NAME(iops.[object_id], [database_id]) <> 'sys'
ORDER BY iops.[page_latch_wait_count] + iops.[page_io_latch_wait_count] DESC
```

Results | Messages

	ObjectName	IndexID	IndexName	IndexType	LeafInsertCount	LeafDeleteCount	Leaf
1	[MYSOLUTIONS]...	4	IX_tbImagens...	NONCLUSTERED	490	0	0
2	[SiGO].[tbOcorren...	2	PK_tbOcorren...	UNIQUE NONCL...	20	0	0

page_latch_wait_count: *number of waits on the physical page of the object to have the latch removed;*

page_io_latch_wait_count: *number of waits while SQL loads pages from disk into memory for an index operation.*

Demo: Quais índices estão faltando?

```
SELECT CONVERT(decimal(18,2), user_seeks * avg_total_user_cost * (avg_user_impact * 0.01)) AS [index_advantage],
                    migs.last_user_seek, mid.[statement] AS [Database.Schema.Table],
                    mid.equality_columns, mid.inequality_columns, mid.included_columns,
                    migs.unique_compiles, migs.user_seeks, migs.avg_total_user_cost, migs.avg_user_impact
FROM sys.dm_db_missing_index_group_stats AS migs WITH (NOLOCK)
INNER JOIN sys.dm_db_missing_index_groups AS mig WITH (NOLOCK)
ON migs.group_handle = mig.index_group_handle
INNER JOIN sys.dm_db_missing_index_details AS mid WITH (NOLOCK)
ON mig.index_handle = mid.index_handle
ORDER BY index_advantage DESC OPTION (RECOMPILE);
```

Results | Messages

	index_advantage	last_user_seek	Database.Schema.Table	equality_columns	inequality_columns
1	219.40	2015-01-25 07:56:52.567	[mySolutionsdb].[SiGO].[tbOcorrencias]	[codigo_cliente]	[data_ocorrencia]

included_columns	unique_compiles	user_seeks	avg_total_user_cost	avg_user_impact
[codigo_ocorrencia], [codigo_autor], [codigo_tip...	1	21	10,5960862785566	98,6

Glenn Berry diz:

- Look at last user seek time, number of user seeks to help determine source and importance
- Also look at avg_user_impact and avg_total_user_cost to help determine importance
- SQL Server is overly eager to add included columns, so beware
- Do not just blindly add indexes that show up from this query!!!

conexões
inteligentes

DMVs: SQL Server OS

Microsoft SQL Server

Objeto	Utilidade
sys.dm_os_wait_stats	Informações sobre o que tipos de esperas que seu SQL Server sofreu;
sys.dm_os_performance_counters	Informações dos contadores de desempenho do SQL Server;
sys.dm_os_sys_info	Informações sobre a configuração geral do sistema
sys.dm_os_schedulers	Informações sobre os *schedulers* que geram trabalhos para as CPUs
sys.dm_os_sys_memory	Informações sobre o gerenciamento de memória em relação ao servidor
sys.dm_os_process_memory	Informações sobre o gerenciamento de memória da instância
sys.dm_os_memory_cache_counters	Estatísticas sobre a forma como a memória cache está sendo utilizada, incluindo a quantidade de memória em uso
sys.dm_os_latch_stats	Estatísticas sobre os bloqueios de recursos em baixo nível que o SQL Server usa para bloquear páginas físicas na memória, e muito mais.

MINDWORKS

Soluções em TI
- Serviços
- Infraestrutura
- Treinamento

Demo: Top eventos de espera

```
WITH [Waits]
AS (SELECT wait_type, wait_time_ms/ 1000.0 AS [WaitS], (wait_time_ms - signal_wait_time_ms) / 1000.0 AS [ResourceS], signal_wait_time_ms /
```

Results | Messages

	WaitType	Wait_Sec	Resource_Sec	Signal_Sec	Wait Count	Wait Percentage	AvgWait_Sec	AvgRes_Sec	AvgSig_Sec
1	PAGEIOLATCH_SH	756.00	740.86	15.14	1425855	33.68	0.0005	0.0005	0.0000
2	OLEDB	552.17	552.17	0.00	857	24.60	0.6443	0.6443	0.0000
3	ASYNC_NETWORK_IO	228.27	221.99	6.28	398526	10.17	0.0006	0.0006	0.0000
4	PAGEIOLATCH_EX	200.21	200.10	0.11	22160	8.92	0.0090	0.0090	0.0000
5	PREEMPTIVE_OS_WAITFORSINGLEOBJECT	98.00	98.00	0.00	88366	4.37	0.0011	0.0011	0.0000
6	PREEMPTIVE_OS_REPORTEVENT	95.66	95.66	0.00	40	4.26	2.3915	2.3915	0.0000
7	WRITELOG	77.64	77.18	0.46	24562	3.46	0.0032	0.0031	0.0000
8	LCK_M_S	58.35	58.35	0.00	75	2.60	0.7780	0.7779	0.0001
9	PAGEIOLATCH_UP	40.04	40.04	0.00	341	1.78	0.1174	0.1174	0.0000
10	PAGELATCH_UP	30.08	30.08	0.00	114	1.34	0.2638	0.2638	0.0000
11	SOS_SCHEDULER_YIELD	28.90	0.01	28.89	11124	1.29	0.0026	0.0000	0.0026
12	IO_COMPLETION	23.17	23.16	0.01	871	1.03	0.0266	0.0266	0.0000
13	SQLTRACE_FILE_WRITE_IO_COMPLETION	16.69	16.69	0.00	22	0.74	0.7587	0.7587	0.0000
14	PREEMPTIVE_OS_GENERICOPS	5.04	5.04	0.00	15	0.22	0.3359	0.3359	0.0000
15	CXPACKET	4.71	4.53	0.19	1099	0.21	0.0043	0.0041	0.0002
16	PREEMPTIVE_OS_LIBRARYOPS	4.19	4.19	0.00	7	0.19	0.5991	0.5991	0.0000
17	SQLTRACE_FILE_BUFFER	3.57	3.57	0.00	4	0.16	0.8925	0.8925	0.0000

```
   CAST ((MAX (W1.WaitS) / MAX (W1.WaitCount)) AS DECIMAL (16,4)) AS [AvgWait_Sec],
   CAST ((MAX (W1.ResourceS) / MAX (W1.WaitCount)) AS DECIMAL (16,4)) AS [AvgRes_Sec],
   CAST ((MAX (W1.SignalS) / MAX (W1.WaitCount)) AS DECIMAL (16,4)) AS [AvgSig_Sec]
FROM Waits AS W1
INNER JOIN Waits AS W2
ON W2.RowNum <= W1.RowNum
GROUP BY W1.RowNum
HAVING SUM (W2.Percentage) - MAX (W1.Percentage) < 99 -- percentage threshold
OPTION (RECOMPILE);
```

Demo: % de Tempo de Espera - CPU

```
SELECT CAST(100.0 * SUM(signal_wait_time_ms) / SUM (wait_time_ms) AS NUMERIC(20,2)) AS [% Signal (CPU) Waits],
CAST(100.0 * SUM(wait_time_ms - signal_wait_time_ms) / SUM (wait_time_ms) AS NUMERIC(20,2)) AS [% Resource Waits]
FROM sys.dm_os_wait_stats WITH (NOLOCK)
WHERE wait_type NOT IN (
	N'BROKER_EVENTHANDLER', N'BROKER_RECEIVE_WAITFOR', N'BROKER_TASK_STOP', N'BROKER_TO_FLUSH',
N'BROKER_TRANSMITTER', N'CHECKPOINT_QUEUE', N'CHKPT', N'CLR_AUTO_EVENT', N'CLR_MANUAL_EVENT', N'CLR_SEMAPHORE',
N'DBMIRROR_DBM_EVENT', N'DBMIRROR_EVENTS_QUEUE', N'DBMIRROR_WORKER_QUEUE', N'DBMIRRORING_CMD',
N'DIRTY_PAGE_POLL', N'DISPATCHER_QUEUE_SEMAPHORE', N'EXECSYNC', N'FSAGENT', N'FT_IFTS_SCHEDULER_IDLE_WAIT',
N'FT_IFTSHC_MUTEX', N'HADR_CLUSAPI_CALL', N'HADR_FILESTREAM_IOMGR_IOCOMPLETION', N'HADR_LOGCAPTURE_WAIT',
N'HADR_NOTIFICATION_DEQUEUE', N'HADR_TIMER_TASK', N'HADR_WORK_QUEUE', N'KSOURCE_WAKEUP', N'LAZYWRITER_SLEEP',
N'LOGMGR_QUEUE', N'ONDEMAND_TASK_QUEUE', N'PWAIT_ALL_COMPONENTS_INITIALIZED',
N'QDS_PERSIST_TASK_MAIN_LOOP_SLEEP', N'QDS_CLEANUP_STALE_QUERIES_TASK_MAIN_LOOP_SLEEP',
N'REQUEST_FOR_DEADLOCK_SEARCH', N'RESOURCE_QUEUE', N'SERVER_IDLE_CHECK', N'SLEEP_BPOOL_FLUSH',
N'SLEEP_DBSTARTUP', N'SLEEP_DCOMSTARTUP', N'SLEEP_MASTERDBREADY', N'SLEEP_MASTERMDREADY',
	N'SLEEP_MASTERUPGRADED', N'SLEEP_MSDBSTARTUP', N'SLEEP_SYSTEMTASK', N'SLEEP_TASK',
	N'SLEEP_TEMPDBSTARTUP', N'SNI_HTTP_ACCEPT', N'SP_SERVER_DIAGNOSTICS_SLEEP', N'SQLTRACE_BUFFER_FLUSH',
N'SQLTRACE_INCREMENTAL_FLUSH_SLEEP', N'SQLTRACE_WAIT_ENTRIES', N'WAIT_FOR_RESULTS', N'WAITFOR',
N'WAITFOR_TASKSHUTDOWN', N'WAIT_XTP_HOST_WAIT', N'WAIT_XTP_OFFLINE_CKPT_NEW_LOG', N'WAIT_XTP_CKPT_CLOSE',
N'XE_DISPATCHER_JOIN', N'XE_DISPATCHER_WAIT', N'XE_TIMER_EVENT') OPTION (RECOMPILE);
```

Results | Messages

	% Signal (CPU) Waits	% Resource Waits
1	2.31	97.69

Glenn Berry diz:

- Signal Waits above 10-15% is usually a confirming sign of CPU pressure
- Cumulative wait stats are not as useful on an idle instance that is not under load or performance pressure
- Resource waits are non-CPU related waits

Demo: Utilização da Área de Memória

```
SELECT TOP(10) [type] AS [Memory Clerk Type], SUM(single_pages_kb)/1024 AS [SPA Memory Usage (MB)]
FROM sys.dm_os_memory_clerks WITH (NOLOCK)
GROUP BY [type]
ORDER BY SUM(single_pages_kb) DESC OPTION (RECOMPILE);
```

Results | Messages

	Memory Clerk Type	SPA Memory Usage (MB)
1	CACHESTORE_SQLCP	32
2	CACHESTORE_PHDR	22
3	CACHESTORE_OBJCP	19
4	OBJECTSTORE_LOCK_MANAGER	12
5	USERSTORE_DBMETADATA	6
6	MEMORYCLERK_SOSNODE	4
7	USERSTORE_SCHEMAMGR	3
8	CACHESTORE_SYSTEMROWSET	2
9	MEMORYCLERK_SQLSTORENG	1
10	MEMORYCLERK_SQLGENERAL	1

Glenn Berry diz:

- **CACHESTORE_SQLCP SQL Plans**
  - These are cached SQL statements or batches that aren't in stored procedures, functions and triggers
  - Watch out for high values for CACHESTORE_SQLCP
- **CACHESTORE_OBJCP Object Plans**
  - These are compiled plans for stored procedures, functions and triggers

conexões
inteligentes

Fontes

- http://download.red-gate.com/ebooks/SQL/eBook_Performance_Tuning_Davidson_Ford.pdf
- https://sqlserverperformance.wordpress.com
- http://blogs.msdn.com/b/psssql/archive/2012/01/23/how-it-works-sys-dm-io-pending-io-requests.aspx

MINDWORKS

Soluções em TI
- Serviços
- Infraestrutura
- Treinamento

Obrigado.

MINDWORKS