Script

den 15 mars 2019 14:30

http://FirstResponderKit.org/ (Bra skit som jag tror är gratis)

Drop all views in databas

den 15 mars 2019 14:42

Ibland behöver man att ta bort vyer och peka om dom igen mot en annan databas

```
DECLARE @sql VARCHAR(MAX) = ''
      , @crlf VARCHAR(2) = CHAR(13) + CHAR(10) ;

SELECT @sql = @sql + 'DROP VIEW ' + QUOTENAME(SCHEMA_NAME(schema_id)) + '.' +
QUOTENAME(v.name) +';' + @crlf
FROM   sys.views v

PRINT @sql;
EXEC(@sql);
```

Find the duplicates in OCR

den 6 december 2019 16:15

```
select [OCR No_], count([OCR No_]) from Cust_LedgerEntry where [Document Type] = 2 and [Posting Date] > convert(datetime,'2019-07-01T00:00:00.000')  group by [OCR No_] Having count([OCR No_]) > 1
```

Hitta deadlocks

den 12 april 2019 09:25

```
use intilityDBA
exec sp_blitzlock
```

https://www.brentozar.com/archive/2011/07/difficulty-deadlocks/

WhoLock

den 1 oktober 2019 15:31

```
--Create Procedure WhoLock
--AS
set nocount on
if object_id('tempdb..#locksummary') is not null Drop table #locksummary
if object_id('tempdb..#lock') is not null Drop table #lock
create table #lock (  spid int,  dbid int,  objId int,  indId int,  Type char(4),  resource nchar(32),
Mode char(8),  status char(6))
Insert into #lock exec sp_lock
if object_id('tempdb..#who') is not null Drop table #who
create table #who (  spid int, ecid int, status char(30),
        loginame char(128), hostname char(128),
        blk char(5), dbname char(128), cmd char(16)
        --
        , request_id INT --Needed for SQL 2008 onwards
        --
      )
Insert into #who exec sp_who
Print '---------------------------------------'
Print 'Lock Summary for ' + @@servername  + ' (excluding tempdb):'
Print '---------------------------------------' + Char(10)
Select  left(loginame, 28) as loginame,
  left(db_name(dbid),128) as DB,
  left(object_name(objID),30) as object,
  max(mode) as [ToLevel],
  Count(*) as [How Many],
  Max(Case When mode= 'X' Then cmd Else null End) as [Xclusive lock for command],
  l.spid, hostname
into #LockSummary
from #lock l join #who w on l.spid= w.spid
where dbID != db_id('tempdb') and l.status='GRANT'
group by dbID, objID, l.spid, hostname, loginame

Select * from #LockSummary order by [ToLevel] Desc, [How Many] Desc, loginame, DB, object

Print '--------'
Print 'Who is blocking:'
Print '--------' + char(10)
SELECT p.spid
,convert(char(12), d.name) db_name
, program_name
, p.loginame
, convert(char(12), hostname) hostname
, cmd
, p.status
, p.blocked
, login_time
, last_batch
, p.spid
FROM   master..sysprocesses p
JOIN   master..sysdatabases d ON p.dbid =  d.dbid
WHERE  EXISTS (  SELECT 1
```

```
        FROM      master..sysprocesses p2
        WHERE     p2.blocked = p.spid )

Print '--------'
Print 'Details:'
Print '--------' + char(10)
Select     left(loginame, 30) as loginame,  l.spid,
   left(db_name(dbid),15) as DB,
   left(object_name(objID),40) as object,
   mode ,
   blk,
   l.status
from #lock l join #who w on l.spid= w.spid
where dbID != db_id('tempdb') and blk <>0
Order by mode desc, blk, loginame, dbID, objID, l.status
```

Running query

den 2 oktober 2019 17:18

```
SELECT sqltext.TEXT,
req.session_id,
req.status,
req.start_time,
req.command,
req.cpu_time,
req.total_elapsed_time
FROM sys.dm_exec_requests req
CROSS APPLY sys.dm_exec_sql_text(sql_handle) AS sqltext
```

```
DECLARE @sqltext VARBINARY(128)
SELECT @sqltext = sql_handle
FROM sys.sysprocesses
WHERE spid = 97
SELECT TEXT
FROM sys.dm_exec_sql_text(@sqltext)
GO
```

Performance / Monitoring

den 15 mars 2019 14:43

Basic RAM consumption

den 15 mars 2019 14:44

```
select
(physical_memory_in_use_kb/1024)Phy_Memory_usedby_Sqlserver_MB,
(locked_page_allocations_kb/1024 )Locked_pages_used_Sqlserver_MB,
(virtual_address_space_committed_kb/1024 )Total_Memory_UsedBySQLServer_MB,
process_physical_memory_low,
process_virtual_memory_low
from sys. dm_os_process_memory

------------------------------------------------------------

-- Good basic information about OS memory amounts and state  (Query 14) (System Memory)
SELECT total_physical_memory_kb/1024 AS [Physical Memory (MB)],
      available_physical_memory_kb/1024 AS [Available Memory (MB)],
      total_page_file_kb/1024 AS [Total Page File (MB)],
      available_page_file_kb/1024 AS [Available Page File (MB)],
      system_cache_kb/1024 AS [System Cache (MB)],
      system_memory_state_desc AS [System Memory State]
FROM sys.dm_os_sys_memory WITH (NOLOCK) OPTION (RECOMPILE);
------

-- You want to see "Available physical memory is high" for System Memory State
-- This indicates that you are not under external memory pressure

-- Possible System Memory State values:
-- Available physical memory is high
-- Physical memory usage is steady
-- Available physical memory is low
-- Available physical memory is running low
-- Physical memory state is transitioning

------------------------------------------------------------

-- Memory Grants Pending value for current instance  (Query 47) (Memory Grants Pending)
SELECT @@SERVERNAME AS [Server Name], RTRIM([object_name]) AS [Object Name], cntr_value AS [Memory Grants Pending]
FROM sys.dm_os_performance_counters WITH (NOLOCK)
WHERE [object_name] LIKE N'%Memory Manager%' -- Handles named instances
AND counter_name = N'Memory Grants Pending' OPTION (RECOMPILE);
------

-- Run multiple times, and run periodically if you suspect you are under memory pressure
-- Memory Grants Pending above zero for a sustained period is a very strong indicator of internal memory pressure

------------------------------------------------------------

-- Top Cached SPs By Total Logical Reads. Logical reads relate to memory pressure  (Query 62) (SP Logical Reads)
SELECT TOP(25) p.name AS [SP Name], qs.total_logical_reads AS [TotalLogicalReads],
qs.total_logical_reads/qs.execution_count AS [AvgLogicalReads],qs.execution_count,
ISNULL(qs.execution_count/DATEDIFF(Minute, qs.cached_time, GETDATE()), 0) AS [Calls/Minute],
qs.total_elapsed_time, qs.total_elapsed_time/qs.execution_count AS [avg_elapsed_time],
CASE WHEN CONVERT(nvarchar(max), qp.query_plan) LIKE N'%<MissingIndexes>%' THEN 1 ELSE 0 END AS [Has Missing Index],
FORMAT(qs.last_execution_time, 'yyyy-MM-dd HH:mm:ss', 'en-US') AS [Last Execution Time],
FORMAT(qs.cached_time, 'yyyy-MM-dd HH:mm:ss', 'en-US') AS [Plan Cached Time]
-- ,qp.query_plan AS [Query Plan] -- Uncomment if you want the Query Plan
FROM sys.procedures AS p WITH (NOLOCK)
INNER JOIN sys.dm_exec_procedure_stats AS qs WITH (NOLOCK)
ON p.[object_id] = qs.[object_id]
CROSS APPLY sys.dm_exec_query_plan(qs.plan_handle) AS qp
WHERE qs.database_id = DB_ID()
AND DATEDIFF(Minute, qs.cached_time, GETDATE()) > 0
```

```
ORDER BY qs.total_logical_reads DESC OPTION (RECOMPILE);
------

-- This helps you find the most expensive cached stored procedures from a memory perspective
-- You should look at this if you see signs of memory pressure

------------------------------------------------------------

-- Top Cached SPs By Total Physical Reads. Physical reads relate to disk read I/O pressure  (Query 63) (SP Physical Reads)
SELECT TOP(25) p.name AS [SP Name],qs.total_physical_reads AS [TotalPhysicalReads],
qs.total_physical_reads/qs.execution_count AS [AvgPhysicalReads], qs.execution_count,
qs.total_logical_reads,qs.total_elapsed_time, qs.total_elapsed_time/qs.execution_count AS [avg_elapsed_time],
CASE WHEN CONVERT(nvarchar(max), qp.query_plan) LIKE N'%<MissingIndexes>%' THEN 1 ELSE 0 END AS [Has Missing Index],
FORMAT(qs.last_execution_time, 'yyyy-MM-dd HH:mm:ss', 'en-US') AS [Last Execution Time],
FORMAT(qs.cached_time, 'yyyy-MM-dd HH:mm:ss', 'en-US') AS [Plan Cached Time]
-- ,qp.query_plan AS [Query Plan] -- Uncomment if you want the Query Plan
FROM sys.procedures AS p WITH (NOLOCK)
INNER JOIN sys.dm_exec_procedure_stats AS qs WITH (NOLOCK)
ON p.[object_id] = qs.[object_id]
CROSS APPLY sys.dm_exec_query_plan(qs.plan_handle) AS qp
WHERE qs.database_id = DB_ID()
AND qs.total_physical_reads > 0
ORDER BY qs.total_physical_reads DESC, qs.total_logical_reads DESC OPTION (RECOMPILE);
------

-- This helps you find the most expensive cached stored procedures from a read I/O perspective
-- You should look at this if you see signs of I/O pressure or of memory pressure

------------------------------------------------------------

-- Top Cached SPs By Total Logical Writes (Query 64) (SP Logical Writes)
-- Logical writes relate to both memory and disk I/O pressure
SELECT TOP(25) p.name AS [SP Name], qs.total_logical_writes AS [TotalLogicalWrites],
qs.total_logical_writes/qs.execution_count AS [AvgLogicalWrites], qs.execution_count,
ISNULL(qs.execution_count/DATEDIFF(Minute, qs.cached_time, GETDATE()), 0) AS [Calls/Minute],
qs.total_elapsed_time, qs.total_elapsed_time/qs.execution_count AS [avg_elapsed_time],
CASE WHEN CONVERT(nvarchar(max), qp.query_plan) LIKE N'%<MissingIndexes>%' THEN 1 ELSE 0 END AS [Has Missing Index],
FORMAT(qs.last_execution_time, 'yyyy-MM-dd HH:mm:ss', 'en-US') AS [Last Execution Time],
FORMAT(qs.cached_time, 'yyyy-MM-dd HH:mm:ss', 'en-US') AS [Plan Cached Time]
-- ,qp.query_plan AS [Query Plan] -- Uncomment if you want the Query Plan
FROM sys.procedures AS p WITH (NOLOCK)
INNER JOIN sys.dm_exec_procedure_stats AS qs WITH (NOLOCK)
ON p.[object_id] = qs.[object_id]
CROSS APPLY sys.dm_exec_query_plan(qs.plan_handle) AS qp
WHERE qs.database_id = DB_ID()
AND qs.total_logical_writes > 0
AND DATEDIFF(Minute, qs.cached_time, GETDATE()) > 0
ORDER BY qs.total_logical_writes DESC OPTION (RECOMPILE);
------

-- This helps you find the most expensive cached stored procedures from a write I/O perspective
-- You should look at this if you see signs of I/O pressure or of memory pressure
```

History

den 15 mars 2019 14:46

Find in cache

den 15 mars 2019 14:46

Find in transactionlog

den 15 mars 2019 14:47

Finding data or data containers

den 15 september 2020 11:09

Find value in database

den 29 oktober 2019 13:04

```
-- This takes about 6 min to run in Johans company.
-- 1 h in Nox prod. (No wildcard on SearchStr2)

DECLARE @SearchStr nvarchar(100)
SET @SearchStr = 'YourString'
DECLARE @TableFilter nvarchar(100)
SET @TableFilter = 'Johan'

CREATE TABLE #Results (ColumnName nvarchar(370), ColumnValue nvarchar(3630))

SET NOCOUNT ON

DECLARE @TableName nvarchar(256), @ColumnName nvarchar(128), @SearchStr2 nvarchar(110), @TableFilter2 nvarchar(110)
SET  @TableName = ''
SET @SearchStr2 = QUOTENAME('%' + @SearchStr + '%','''') -- Remove the % for exact match.
SET @TableFilter2 = '%' + @TableFilter + '%'

WHILE @TableName IS NOT NULL

BEGIN
    SET @ColumnName = ''
    SET @TableName =
    (
        SELECT MIN(QUOTENAME(TABLE_SCHEMA) + '.' + QUOTENAME(TABLE_NAME))
        FROM     INFORMATION_SCHEMA.TABLES
        WHERE         TABLE_TYPE = 'BASE TABLE'
            AND    QUOTENAME(TABLE_SCHEMA) + '.' + QUOTENAME(TABLE_NAME) > @TableName
            AND    OBJECTPROPERTY(
                    OBJECT_ID(
                        QUOTENAME(TABLE_SCHEMA) + '.' + QUOTENAME(TABLE_NAME)
                         ), 'IsMSShipped'
                           ) = 0
        and TABLE_NAME like @TableFilter2 -- Revers to exclude.
    )

    WHILE (@TableName IS NOT NULL) AND (@ColumnName IS NOT NULL)

    BEGIN
        SET @ColumnName =
        (
            SELECT MIN(QUOTENAME(COLUMN_NAME))
            FROM     INFORMATION_SCHEMA.COLUMNS
            WHERE         TABLE_SCHEMA    = PARSENAME(@TableName, 2)
                AND    TABLE_NAME    = PARSENAME(@TableName, 1)
                AND    DATA_TYPE IN ('char', 'varchar', 'nchar', 'nvarchar', 'int', 'decimal')
                AND    QUOTENAME(COLUMN_NAME) > @ColumnName
        )

        IF @ColumnName IS NOT NULL

        BEGIN
            INSERT INTO #Results
            EXEC
            (
                'SELECT ''' + @TableName + '.' + @ColumnName + ''', LEFT(' + @ColumnName + ', 3630) FROM ' + @TableName + ' (NOLOCK) ' +
                ' WHERE ' + @ColumnName + ' LIKE ' + @SearchStr2
            )
        END
    END
END

SELECT ColumnName, ColumnValue FROM #Results

DROP TABLE #Results
```

Find all tables containing column with specified name

den 7 februari 2020 09:20

```
SELECT      c.name  AS 'ColumnName'
          ,t.name AS 'TableName'
FROM        sys.columns c
JOIN        sys.tables  t   ON c.object_id = t.object_id
WHERE       c.name LIKE '%MyName%'
ORDER BY    TableName
          ,ColumnName;
```

Find GUID in allt tables in database.sql

den 15 april 2020 10:52

```
CREATE TABLE #Results (TheTable nvarchar(370), TheColumn nvarchar(3630))

DECLARE @searchValue uniqueidentifier
SET @searchValue = '{872B1C9A-5407-485F-9DBD-AB5D00BB5B64}'

DECLARE abc CURSOR FOR
  SELECT
    c.TABLE_SCHEMA, c.TABLE_NAME, c.COLUMN_NAME
  FROM INFORMATION_SCHEMA.Columns c
    INNER JOIN INFORMATION_SCHEMA.Tables t
    ON c.TABLE_NAME = t.TABLE_NAME
    AND t.TABLE_TYPE = 'BASE TABLE'
  WHERE DATA_TYPE = 'uniqueidentifier'

DECLARE @tableSchema varchar(200)
DECLARE @tableName varchar(200)
DECLARE @columnName varchar(200)
DECLARE @szQuery varchar(8000)

OPEN ABC

FETCH NEXT FROM abc INTO @tableSchema, @tableName, @columnName
WHILE (@@FETCH_STATUS = 0)
BEGIN
  SET @szQuery =
    'SELECT '''+@tableSchema+'.'+@tableName+''' AS ''TheTable'', '''+@columnName+''' AS
''TheColumn'' '+
    'FROM '+@tableSchema+'.'+@tableName+' '+
    'WHERE '+@columnName+' = '''+CAST(@searchValue AS varchar(50))+''''

          INSERT INTO #Results EXEC (@szQuery)

  FETCH NEXT FROM abc INTO @tableSchema, @tableName, @columnName
END

CLOSE abc
DEALLOCATE abc

SELECT TheTable, TheColumn FROM #Results

DROP TABLE #Results
```

List all new rows in database

den 15 september 2020 11:05

```
-- For sql server 2012 and up.
-- The listing only works if the table has a primary key for its first column. (Often the case in Arc, not so much Navision…)
-- Just show the difference in row count if the new rows are not inserted last.

create table #TempRC1 ([name] varchar(100), [count] int)

insert into #TempRC1
	SELECT o.NAME, i.rowcnt
		FROM sysindexes AS i
		  INNER JOIN sysobjects AS o ON i.id = o.id
		WHERE i.indid < 2  AND OBJECTPROPERTY(o.id, 'IsMSShipped') = 0
		ORDER BY o.NAME

-- => Do stuff that inserts data anywhere in the database - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -

create table #TempRC2 ([name] varchar(100), [count] int)

insert into #TempRC2
	SELECT o.NAME, i.rowcnt
		FROM sysindexes AS i
		  INNER JOIN sysobjects AS o ON i.id = o.id
		WHERE i.indid < 2  AND OBJECTPROPERTY(o.id, 'IsMSShipped') = 0
		ORDER BY o.NAME

DECLARE @NAME VARCHAR(100)
DECLARE @COUNT VARCHAR(100)
DECLARE @SQL NVARCHAR(300)
DECLARE @KEY NVARCHAR(100)

DECLARE CUR CURSOR FOR
	select one.[name] as 'Table name', two.[count] - one.[count] as 'New rows', col.Column_Name
		from #TempRC1 one
			join #TempRC2 two
				on two.[name] = one.[name] and two.[count] <> one.[count]
			join INFORMATION_SCHEMA.TABLE_CONSTRAINTS tab
				on tab.Table_Name = one.[name] COLLATE DATABASE_DEFAULT AND
				tab.Constraint_Type = 'PRIMARY KEY' COLLATE DATABASE_DEFAULT
			join INFORMATION_SCHEMA.CONSTRAINT_COLUMN_USAGE col
				on col.Table_Name = tab.Table_Name COLLATE DATABASE_DEFAULT and
				col.Constraint_Name = tab.Constraint_Name COLLATE DATABASE_DEFAULT
		order by one.[name]

OPEN CUR
FETCH NEXT FROM CUR INTO @NAME, @COUNT, @KEY
WHILE @@FETCH_STATUS = 0
 BEGIN
	select @NAME as 'Table name'
   SET @SQL = 'select top ' + @COUNT + ' * from [' + @NAME + '] order by [' + @KEY + '] desc'
   EXEC Sp_executesql @SQL
```

```
   FETCH NEXT FROM CUR INTO @NAME, @COUNT, @KEY
  END

CLOSE CUR
DEALLOCATE CUR

--drop table #TempRC1
--drop table #TempRC2
```

Ola Hallengrens index

den 3 april 2020 15:26

Collector use Ola Hallengren scripts to maintain indexes.

https://ola.hallengren.com/

https://www.sqlskills.com/blogs/glenn/category/dmv-queries/

Extended events

Check all SERVICEs and QUEUEs in SQL server

den 19 maj 2020 16:36

```
SELECT t1.name AS [Service_Name],  t3.name AS [Schema_Name],  t2.name AS [Queue_Name],
CASE WHEN t4.state IS NULL THEN 'Not available'
ELSE t4.state
END AS [Queue_State],
CASE WHEN t4.tasks_waiting IS NULL THEN '--'
ELSE CONVERT(VARCHAR, t4.tasks_waiting)
END AS tasks_waiting,
CASE WHEN t4.last_activated_time IS NULL THEN '--'
ELSE CONVERT(varchar, t4.last_activated_time)
END AS last_activated_time ,
CASE WHEN t4.last_empty_rowset_time IS NULL THEN '--'
ELSE CONVERT(varchar,t4.last_empty_rowset_time)
END AS last_empty_rowset_time,
(
SELECT COUNT(*)
FROM sys.transmission_queue t6
WHERE (t6.from_service_name = t1.name) ) AS [Tran_Message_Count]
FROM sys.services t1    INNER JOIN sys.service_queues t2
ON ( t1.service_queue_id = t2.object_id )
INNER JOIN sys.schemas t3 ON ( t2.schema_id = t3.schema_id )
LEFT OUTER JOIN sys.dm_broker_queue_monitors t4
ON ( t2.object_id = t4.queue_id  AND t4.database_id = DB_ID() )
INNER JOIN sys.databases t5 ON ( t5.database_id = DB_ID() );
```

Check sessions and transactions

den 27 juli 2020 17:05

```
select *
   from sys.dm_exec_requests
   cross apply sys.dm_exec_sql_text(plan_handle)
   where session_id = 69

        select *from sys.dm_exec_requests

        SELECT * FROM sys.sysprocesses WHERE open_tran = 1

        DBCC OPENTRAN

        select transaction_id, name, transaction_begin_time
 ,case transaction_type
   when 1 then '1 = Read/write transaction'
   when 2 then '2 = Read-only transaction'
   when 3 then '3 = System transaction'
   when 4 then '4 = Distributed transaction'
end as transaction_type
,case transaction_state
   when 0 then '0 = The transaction has not been completely initialized yet'
   when 1 then '1 = The transaction has been initialized but has not started'
   when 2 then '2 = The transaction is active'
   when 3 then '3 = The transaction has ended. This is used for read-only transactions'
   when 4 then '4 = The commit process has been initiated on the distributed transaction'
   when 5 then '5 = The transaction is in a prepared state and waiting resolution'
   when 6 then '6 = The transaction has been committed'
   when 7 then '7 = The transaction is being rolled back'
   when 8 then '8 = The transaction has been rolled back'
end as transaction_state
,case dtc_state
   when 1 then '1 = ACTIVE'
   when 2 then '2 = PREPARED'
   when 3 then '3 = COMMITTED'
   when 4 then '4 = ABORTED'
   when 5 then '5 = RECOVERED'
end as dtc_state
,transaction_status, transaction_status2,dtc_status, dtc_isolation_level, filestream_transaction_id
from sys.dm_tran_active_transactions


SELECT
trans.session_id AS [SESSION ID],
ESes.host_name AS [HOST NAME],login_name AS [Login NAME],
trans.transaction_id AS [TRANSACTION ID],
tas.name AS [TRANSACTION NAME],tas.transaction_begin_time AS [TRANSACTION
BEGIN TIME],
tds.database_id AS [DATABASE ID],DBs.name AS [DATABASE NAME]
FROM sys.dm_tran_active_transactions tas
JOIN sys.dm_tran_session_transactions trans
ON (trans.transaction_id=tas.transaction_id)
LEFT OUTER JOIN sys.dm_tran_database_transactions tds
```

```
ON (tas.transaction_id = tds.transaction_id )
LEFT OUTER JOIN sys.databases AS DBs
ON tds.database_id = DBs.database_id
LEFT OUTER JOIN sys.dm_exec_sessions AS ESes
ON trans.session_id = ESes.session_id
WHERE ESes.session_id IS NOT NULL



SELECT
  GETDATE() AS [Current Time],
  [des].[login_name] AS [Login Name],
  DB_NAME ([dtdt].database_id) AS [Database Name],
  [dtdt].[database_transaction_begin_time] AS [Transaction Begin Time],
  [dtdt].[database_transaction_log_bytes_used] AS [Log Used Bytes],
  [dtdt].[database_transaction_log_bytes_reserved] AS [Log Reserved Bytes],
  SUBSTRING([dest].text, [der].statement_start_offset/2 + 1,(CASE WHEN
[der].statement_end_offset = -1 THEN LEN(CONVERT(nvarchar(max),[dest].text)) * 2 ELSE
[der].statement_end_offset END - [der].statement_start_offset)/2) as [Query Text]
FROM
  sys.dm_tran_database_transactions [dtdt]
  INNER JOIN sys.dm_tran_session_transactions [dtst] ON  [dtst].[transaction_id] =
[dtdt].[transaction_id]
  INNER JOIN sys.dm_exec_sessions [des] ON  [des].[session_id] = [dtst].[session_id]
  INNER JOIN sys.dm_exec_connections [dec] ON   [dec].[session_id] = [dtst].[session_id]
  LEFT OUTER JOIN sys.dm_exec_requests [der] ON [der].[session_id] = [dtst].[session_id]
  CROSS APPLY sys.dm_exec_sql_text ([dec].[most_recent_sql_handle]) AS [dest]
GO
```