|  |
| --- |
| Close-up image showing the leaf-sides of two oversized books side-by-side on a bookshelf, with additional books in soft focus background |
| SQL QUERY HELP  Enhance Your Findings |
| |  |  |  | | --- | --- | --- | | Kumar, Raghavendra | 4/4/16 | SQL | |

1. **Blockings Finding Queries :-**

* use SYSAdmin

Select \* from EventInfoMain\_His order by dtStampFound desc

* use SYSAdmin

Select \* from EventInfoMain\_His where dtStampReleased is null order by dtStampFound desc

* select \* from sysprocesses where blocked <> 0
* select \* from sys.sysprocesses where spid >= 50 and blocked <> 0
* select \* from sys.sysprocesses CROSS APPLY sys.dm\_exec\_sql\_text(sql\_handle) where spid in ('148', '178')
* SP\_WHO2 @SPID

1. **SQL Jobs Complete % :-**

* USE MASTER

GO

SELECT SPID,ER.percent\_complete,

CAST(((DATEDIFF(s,start\_time,GetDate()))/3600) as varchar) + ' hour(s), '

+ CAST((DATEDIFF(s,start\_time,GetDate())%3600)/60 as varchar) + 'min, '

+ CAST((DATEDIFF(s,start\_time,GetDate())%60) as varchar) + ' sec' as running\_time,

CAST((estimated\_completion\_time/3600000) as varchar) + ' hour(s), '

+ CAST((estimated\_completion\_time %3600000)/60000 as varchar) + 'min, '

+ CAST((estimated\_completion\_time %60000)/1000 as varchar) + ' sec' as est\_time\_to\_go,

DATEADD(second,estimated\_completion\_time/1000, getdate()) as est\_completion\_time,

/\* End of Article Code \*/

ER.command,ER.blocking\_session\_id, SP.DBID,LASTWAITTYPE,

DB\_NAME(SP.DBID) AS DBNAME,

SUBSTRING(est.text, (ER.statement\_start\_offset/2)+1,

((CASE ER.statement\_end\_offset

WHEN -1 THEN DATALENGTH(est.text)

ELSE ER.statement\_end\_offset

END - ER.statement\_start\_offset)/2) + 1) AS QueryText,

TEXT,CPU,HOSTNAME,LOGIN\_TIME,LOGINAME,

SP.status,PROGRAM\_NAME,NT\_DOMAIN, NT\_USERNAME

FROM SYSPROCESSES SP

INNER JOIN sys.dm\_exec\_requests ER

ON sp.spid = ER.session\_id

CROSS APPLY SYS.DM\_EXEC\_SQL\_TEXT(er.sql\_handle) EST

ORDER BY CPU DESC

* SELECT command, percent\_complete,

'elapsed' = total\_elapsed\_time / 60000.0,

'remaining' = estimated\_completion\_time / 60000.0

FROM sys.dm\_exec\_requests

WHERE command like 'BACKUP%'

* SELECT command, percent\_complete, start\_time FROM sys.dm\_exec\_requests WHERE session\_id=@SPID
* --You can use msdb.dbo.sysjobactivity, checking for a record with a non-null start\_execution\_date and a null stop\_execution\_date, meaning the job was started, but has not yet completed.

SELECT sj.name , sja.\* FROM msdb.dbo.sysjobactivity AS sja

INNER JOIN msdb.dbo.sysjobs AS sj ON sja.job\_id = sj.job\_id

WHERE sja.start\_execution\_date IS NOT NULL AND sja.stop\_execution\_date IS NULL

* EXEC msdb.dbo.sp\_help\_job @Job\_name = 'Your Job Name'

/\* Check field execution\_status

0 - Returns only those jobs that are not idle or suspended.

1 - Executing.

2 - Waiting for thread.

3 - Between retries.

4 - Idle.

5 - Suspended.

7 - Performing completion actions.

If you need the result of execution, check the field last\_run\_outcome

0 = Failed

1 = Succeeded

3 = Canceled

5 = Unknown \*/

* SELECT ja.job\_id, j.name AS job\_name, ja.start\_execution\_date,

ISNULL(last\_executed\_step\_id,0)+1 AS current\_executed\_step\_id, Js.step\_name

FROM msdb.dbo.sysjobactivity ja LEFT JOIN msdb.dbo.sysjobhistory jh

ON ja.job\_history\_id = jh.instance\_id JOIN msdb.dbo.sysjobs j

ON ja.job\_id = j.job\_id JOIN msdb.dbo.sysjobsteps js ON ja.job\_id = js.job\_id

AND ISNULL(ja.last\_executed\_step\_id,0)+1 = js.step\_id

WHERE ja.session\_id = (SELECT TOP 1 session\_id FROM msdb.dbo.syssessions ORDER BY agent\_start\_date DESC) AND start\_execution\_date is not null AND stop\_execution\_date is null;

* SELECT \* FROM sys.dm\_exec\_requests WHERE session\_id = @spid of your job

--see "percent\_complete" column.

SELECT \* FROM sys.dm\_exec\_sessions WHERE session\_id = @spid

1. **Current Running Query Info :-**

* DBCC INPUTBUFFER(@SPID)
* DECLARE @sqltext VARBINARY(128)

SELECT @sqltext = sql\_handle

FROM sys.sysprocesses

WHERE spid = (YourSessionID)

SELECT TEXT

FROM sys.dm\_exec\_sql\_text(@sqltext)

GO

* SELECT TEXT

FROM sys.dm\_exec\_connections

CROSS APPLY sys.dm\_exec\_sql\_text(most\_recent\_sql\_handle)

WHERE session\_id = (yoursessionID)

GO

* --Find Current SQL Statements that are Running

SELECT SPID = er.session\_id

,STATUS = ses.STATUS

,[Login] = ses.login\_name

,Host = ses.host\_name

,BlkBy = er.blocking\_session\_id

,DBName = DB\_Name(er.database\_id)

,CommandType = er.command

,ObjectName = OBJECT\_NAME(st.objectid)

,CPUTime = er.cpu\_time

,StartTime = er.start\_time

,TimeElapsed = CAST(GETDATE() - er.start\_time AS TIME)

,SQLStatement = st.text

FROM sys.dm\_exec\_requests er

OUTER APPLY sys.dm\_exec\_sql\_text(er.sql\_handle) st

LEFT JOIN sys.dm\_exec\_sessions ses

ON ses.session\_id = er.session\_id

LEFT JOIN sys.dm\_exec\_connections con

ON con.session\_id = ses.session\_id

WHERE st.text IS NOT NULL

* SELECT

CASE

WHEN statement\_end\_offset = -1

THEN text

ELSE SUBSTRING(text,statement\_start\_offset/2,(statement\_end\_offset- statement\_start\_offset)/2)

END, \*

FROM sys.dm\_exec\_requests

CROSS APPLY sys.dm\_exec\_sql\_text(sql\_handle)

WHERE session\_id = <whatever>

1. **SQL SERVER MIN MAX memory status check :-**

* SELECT name, value, value\_in\_use, [description]

FROM sys.configurations WHERE name like '%server memory%'

ORDER BY name OPTION (RECOMPILE);

* SELECT (physical\_memory\_in\_use\_kb/1024) AS Memory\_usedby\_Sqlserver\_MB,

(locked\_page\_allocations\_kb/1024) AS Locked\_pages\_used\_Sqlserver\_MB,

(total\_virtual\_address\_space\_kb/1024) AS Total\_VAS\_in\_MB,

process\_physical\_memory\_low, process\_virtual\_memory\_low

FROM sys.dm\_os\_process\_memory;

1. **DBs and recovery models info :-**

* Select name, database\_id, recovery\_model, recovery\_model\_desc, log\_reuse\_wait, log\_reuse\_wait\_desc from sys.databases

1. **DBs Backup info :-**

* SELECT CONVERT(CHAR(100), SERVERPROPERTY('Servername')) AS Server,

msdb.dbo.backupset.database\_name,

msdb.dbo.backupset.backup\_start\_date,

msdb.dbo.backupset.backup\_finish\_date,

msdb.dbo.backupset.expiration\_date,

CASE msdb..backupset.type

WHEN 'D' THEN 'Database' WHEN 'L' THEN 'Log'

END AS backup\_type,

msdb.dbo.backupset.backup\_size, msdb.dbo.backupmediafamily.logical\_device\_name,

msdb.dbo.backupmediafamily.physical\_device\_name, msdb.dbo.backupset.name AS backupset\_name,

msdb.dbo.backupset.description

FROM msdb.dbo.backupmediafamily

INNER JOIN msdb.dbo.backupset ON msdb.dbo.backupmediafamily.media\_set\_id = msdb.dbo.backupset.media\_set\_id

WHERE (CONVERT(datetime, msdb.dbo.backupset.backup\_start\_date, 102) >= GETDATE() - 2)

ORDER BY

msdb.dbo.backupset.database\_name, msdb.dbo.backupset.backup\_finish\_date

1. **Check Index Fragmentation info :-**

-- Change nn to correct DBID sys.dm\_db\_index\_physical\_stats (nn,NULL, NULL, NULL, 'sampled')

-- This script should get fragmentation information fairly quickly. Please run this script in the database where you want to check the fragmentation of indexes.

* SELECT DB\_NAME() AS "Database", OBJECT\_NAME(IPS.object\_id) AS "Table", SI.[NAME] AS "Index Name",

IPS.index\_id, IPS.OBJECT\_ID, --These fields included for joins only

IPS.index\_type\_desc, --Heap, Non-clustered, or Clustered

IPS.alloc\_unit\_type\_desc, --In-row data or BLOB data

CAST(IPS.avg\_fragmentation\_in\_percent AS decimal(5,2)) AS AvgFragmentationPct,

IPS.record\_count FROM sys.dm\_db\_index\_physical\_stats (nn,NULL, NULL, NULL, 'sampled') IPS

LEFT JOIN sys.sysindexes SI ON IPS.OBJECT\_ID = SI.id AND IPS.index\_id = SI.indid

WHERE IPS.index\_id <> 0 ORDER BY CAST(IPS.avg\_fragmentation\_in\_percent AS decimal(5,2)) DESC