# List CPU Usage Per Database in SQL Server

WITH DB\_CPU\_Stats

AS

(

SELECT DatabaseID, DB\_Name(DatabaseID) AS [DatabaseName],

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 [row\_num],

DatabaseName,

[CPU\_Time\_Ms],

CAST([CPU\_Time\_Ms] \* 1.0 / SUM([CPU\_Time\_Ms]) OVER() \* 100.0 AS DECIMAL(5, 2)) AS [CPUPercent]

FROM DB\_CPU\_Stats

--WHERE DatabaseID > 4 -- system databases

--AND DatabaseID <> 32767 -- ResourceDB

ORDER BY row\_num OPTION (RECOMPILE);

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

WITH CPU\_Per\_Db

AS

(SELECT

dmpa.DatabaseID

, DB\_Name(dmpa.DatabaseID) AS [Database]

, SUM(dmqs.total\_worker\_time) AS CPUTimeAsMS

FROM sys.dm\_exec\_query\_stats dmqs

CROSS APPLY

(SELECT

CONVERT(INT, value) AS [DatabaseID]

FROM sys.dm\_exec\_plan\_attributes(dmqs.plan\_handle)

WHERE attribute = N'dbid') dmpa

GROUP BY dmpa.DatabaseID)

SELECT

[Database]

,[CPUTimeAsMS]

,CAST([CPUTimeAsMS] \* 1.0 / SUM([CPUTimeAsMS]) OVER() \* 100.0 AS DECIMAL(5, 2)) AS [CPUTimeAs%]

FROM CPU\_Per\_Db

ORDER BY [CPUTimeAs%] DESC;

**------------------------------------------------------**

WITH DB\_CPU\_Stats

AS

(SELECT DatabaseID, DB\_Name(DatabaseID) AS [DatabaseName], SUM(total\_worker\_time)/1000000 AS [CPU\_Time\_s]

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\_s] DESC) AS [row\_num],

DatabaseName, [CPU\_Time\_s],

CAST([CPU\_Time\_s] \* 1.0 / SUM([CPU\_Time\_s]) OVER() \* 100.0 AS DECIMAL(5, 2)) AS [CPUPercent]

FROM DB\_CPU\_Stats

WHERE DatabaseID > 4 -- system databases

group by DatabaseID, [DatabaseName], [CPU\_Time\_s]

--and DatabaseName='CMS' -- ResourceDB

ORDER BY row\_num OPTION (RECOMPILE);

# I/O Usage Per Database in SQL Server

WITH IO\_Per\_DB

AS

(SELECT

DB\_NAME(database\_id) AS Db

, CONVERT(DECIMAL(12,2), SUM(num\_of\_bytes\_read + num\_of\_bytes\_written) / 1024 / 1024) AS TotalMb

FROM sys.dm\_io\_virtual\_file\_stats(NULL, NULL) dmivfs

GROUP BY database\_id)

SELECT

Db

,TotalMb

,CAST(TotalMb / SUM(TotalMb) OVER() \* 100 AS DECIMAL(5,2)) AS [I/O%]

FROM IO\_Per\_DB

ORDER BY [I/O%] DESC;

I/O usage at the database file level in SQL Server

WITH IO\_Per\_DB

AS

(SELECT

DB\_NAME(dmivfs.database\_id) AS Db

, CONVERT(DECIMAL(12,2), SUM(num\_of\_bytes\_read + num\_of\_bytes\_written) / 1024 / 1024) AS TotalMb

, CONVERT(DECIMAL(12,2), SUM(num\_of\_bytes\_read) / 1024 / 1024) AS TotalMbRead

, CONVERT(DECIMAL(12,2), SUM(num\_of\_bytes\_written) / 1024 / 1024) AS TotalMbWritten

, CASE WHEN dmmf.type\_desc = 'ROWS' THEN 'Data File' WHEN dmmf.type\_desc = 'LOG' THEN 'Log File' END AS DataFileOrLogFile

FROM sys.dm\_io\_virtual\_file\_stats(NULL, NULL) dmivfs

JOIN sys.master\_files dmmf ON dmivfs.file\_id = dmmf.file\_id AND dmivfs.database\_id = dmmf.database\_id

GROUP BY dmivfs.database\_id, dmmf.type\_desc)

SELECT

Db

, TotalMb

, TotalMbRead

, TotalMbWritten

, DataFileOrLogFile

, CAST(TotalMb / SUM(TotalMb) OVER() \* 100 AS DECIMAL(5,2)) AS [I/O%]

FROM IO\_Per\_DB

ORDER BY [I/O%] DESC;

**READ & WRITE usage on tables**

DECLARE @dbid int

SELECT @dbid = db\_id('CMS')

SELECT TableName = object\_name(s.object\_id),

Reads = SUM(user\_seeks + user\_scans + user\_lookups), Writes = SUM(user\_updates)

FROM sys.dm\_db\_index\_usage\_stats AS s

INNER JOIN sys.indexes AS i

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 = @dbid

GROUP BY object\_name(s.object\_id)

ORDER BY writes DESC

# [Sql server table usage statistics](https://stackoverflow.com/questions/5129114/sql-server-table-usage-statistics)

Select object\_schema\_name(UStat.object\_id)

+ '.' + object\_name(UStat.object\_id) As [Object Name]

,Case

When Sum(User\_Updates + User\_Seeks + User\_Scans + User\_Lookups) = 0 Then Null

Else Cast(Sum(User\_Seeks + User\_Scans + User\_Lookups) As Decimal)

/ Cast(Sum(User\_Updates

+ User\_Seeks

+ User\_Scans

+ User\_Lookups) As Decimal(19,2))

End As [Proportion of Reads]

, Case

When Sum(User\_Updates + User\_Seeks + User\_Scans + User\_Lookups) = 0 Then Null

Else Cast(Sum(User\_Updates) As Decimal)

/ Cast(Sum(User\_Updates

+ User\_Seeks

+ User\_Scans

+ User\_Lookups) As Decimal(19,2))

End As [Proportion Of Writes]

, Sum(User\_Seeks + User\_Scans + User\_Lookups) As [Total Read Ops]

, Sum(User\_Updates) As [Total Write Ops]

From sys.dm\_db\_Index\_Usage\_Stats As UStat

Join Sys.Indexes As I

On UStat.object\_id = I.object\_id

And UStat.index\_Id = I.index\_Id

Join sys.tables As T

On T.object\_id = UStat.object\_id

Where I.Type\_Desc In ( 'Clustered', 'Heap' )

Group By UStat.object\_id

Order By object\_schema\_name(UStat.object\_id)

+ '.' + object\_name(UStat.object\_id)

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

;WITH cte AS

(

SELECT stat.[sql\_handle],

stat.statement\_start\_offset,

stat.statement\_end\_offset,

COUNT(\*) AS [NumExecutionPlans],

SUM(stat.execution\_count) AS [TotalExecutions],

((SUM(stat.total\_logical\_reads) \* 1.0) / SUM(stat.execution\_count)) AS [AvgLogicalReads],

((SUM(stat.total\_worker\_time) \* 1.0) / SUM(stat.execution\_count)) AS [AvgCPU]

FROM sys.dm\_exec\_query\_stats stat

GROUP BY stat.[sql\_handle], stat.statement\_start\_offset, stat.statement\_end\_offset

)

SELECT CONVERT(DECIMAL(15, 5), cte.AvgCPU) AS [AvgCPU],

CONVERT(DECIMAL(15, 5), cte.AvgLogicalReads) AS [AvgLogicalReads],

cte.NumExecutionPlans,

cte.TotalExecutions,

DB\_NAME(txt.[dbid]) AS [DatabaseName],

OBJECT\_NAME(txt.objectid, txt.[dbid]) AS [ObjectName],

SUBSTRING(txt.[text], (cte.statement\_start\_offset / 2) + 1,

(

(CASE cte.statement\_end\_offset

WHEN -1 THEN DATALENGTH(txt.[text])

ELSE cte.statement\_end\_offset

END - cte.statement\_start\_offset) / 2

) + 1

)

FROM cte

CROSS APPLY sys.dm\_exec\_sql\_text(cte.[sql\_handle]) txt

ORDER BY cte.AvgCPU DESC;