**Top 10 expensive queries in SQL**

Often, DBA gets this message from the application developers and clients that their system or applications are running deadly slow. There could be several reasons for a Query or System is running slow. A few common reasons can include a temporary blocking, a network slowness or sometimes even an issue from the application end. To verify if it is a temporary blocking we can simply use this system stored-procedure- sp\_who2. Upon execution of this system stored procedure we are presented with all the granular level details of the system and user processes which are running behind the scene in SQL Server. I have pasted a sample output of the execution below to get a fair idea of how the output will look like-

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| SPID | Status | Login | BlkBy | DBName | Command | CPUTime | DiskIO |
| 1 | BACKGROUND | sa | . | NULL | UNKNOWN TOKEN | 0 | 0 |
| 2 | BACKGROUND | sa | . | NULL | UNKNOWN TOKEN | 0 | 0 |
| 3 | BACKGROUND | sa | . | NULL | UNKNOWN TOKEN | 328 | 0 |
| 4 | BACKGROUND | sa | . | NULL | LOG WRITER | 125 | 0 |
| 5 | BACKGROUND | sa | . | NULL | RECOVERY WRITER | 78 | 0 |
| 6 | BACKGROUND | sa | . | NULL | LAZY WRITER | 9765 | 0 |
| 7 | BACKGROUND | sa | . | NULL | LOCK MONITOR | 15 | 0 |
| 8 | BACKGROUND | sa | . | master | SIGNAL HANDLER | 0 | 0 |
| 9 | BACKGROUND | sa | . | master | BRKR TASK | 0 | 0 |
| 10 | BACKGROUND | sa | . | NULL | XE DISPATCHER | 31 | 0 |
| 11 | BACKGROUND | sa | . | NULL | RESOURCE MONITOR | 503875 | 0 |
| 12 | BACKGROUND | sa | . | NULL | XE TIMER | 1562 | 0 |
| 13 | sleeping | sa | . | master | TASK MANAGER | 0 | 0 |
| 14 | BACKGROUND | sa | . | master | TRACE QUEUE TASK | 31 | 0 |

The process ids which have their SPID lesser than 50 are system processes and others are user processes. If there is a block then you can see the same in blk by column of the above output. Blocking is intermittent and generally resolves in sometime once the resource is set free by the process but if it does not then we have to identify the cause and try to resolve it.  
  
  
While there could be a blocking issue sometimes but most of the times the culprit will be poorly created queries. Such queries can eat up all the resources and make the system severely slow.In order to tune up those queries we first need to identify them.

**Below is the Query that can help you find out the TOP 10 expensive queries in SQL Server by CPU-**

SELECT TOP 10 SUBSTRING(qt.TEXT, (qs.statement\_start\_offset/2)+1,

((CASE qs.statement\_end\_offset

WHEN -1 THEN DATALENGTH(qt.TEXT)

ELSE qs.statement\_end\_offset

END - qs.statement\_start\_offset)/2)+1),

qs.execution\_count,

qs.total\_logical\_reads, qs.last\_logical\_reads,

qs.total\_logical\_writes, qs.last\_logical\_writes,

qs.total\_worker\_time,

qs.last\_worker\_time,

qs.total\_elapsed\_time/1000000 total\_elapsed\_time\_in\_S,

qs.last\_elapsed\_time/1000000 last\_elapsed\_time\_in\_S,

qs.last\_execution\_time,

qp.query\_plan

FROM sys.dm\_exec\_query\_stats qs

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

CROSS APPLY sys.dm\_exec\_query\_plan(qs.plan\_handle) qp

--ORDER BY qs.total\_logical\_reads DESC -- logical reads

 --ORDER BY qs.total\_logical\_writes DESC -- logical writes

ORDER BY qs.total\_worker\_time DESC -- CPU time

--Check the execution plans and Tune them