**[4 Lightweight Ways to Tell if a Database is Used](http://www.brentozar.com/archive/2014/05/4-lightweight-ways-tell-database-used/)**

by [Kendra Little](http://www.brentozar.com/archive/author/kendra-little/) May 5, 2014

FROM: <http://www.brentozar.com/archive/2014/05/4-lightweight-ways-tell-database-used/>

[19 comments](http://www.brentozar.com/archive/2014/05/4-lightweight-ways-tell-database-used/#comments)

We’ve all found those databases. They’re on your production SQL Server instance, but nobody seems to know if they’re being used, or what’s using them.



Is anyone using this?

You could run some sort of trace, but you’d probably drag down performance on your SQL Server. What you need is a quicker way to find out if the database is in use.

**Is a Login Using the database?**

And if so, what query did they run last? My favorite way to see this is with [Adam Machanic](https://twitter.com/AdamMachanic)‘s [sp\_whoisactive](http://www.brentozar.com/archive/2010/09/sql-server-dba-scripts-how-to-find-slow-sql-server-queries/).

You want to run sp\_whoisactive with a couple of special parameters, so it shows you sessions that are connected even if they aren’t running, and limits the output to sessions using that database:

[view source](http://www.brentozar.com/archive/2014/05/4-lightweight-ways-tell-database-used/#viewSource)



[print](http://www.brentozar.com/archive/2014/05/4-lightweight-ways-tell-database-used/#printSource)[?](http://www.brentozar.com/archive/2014/05/4-lightweight-ways-tell-database-used/#about)

|  |
| --- |
| exec sp\_whoisactive |
| @show\_sleeping\_spids=2, | |

|  |
| --- |
| @filter\_type='database', |
| @filter='AdventureWorks2012'; | |

|  |
| --- |
| GO |

* **Pros**: Super fast, super easy. Lets you see the host name and the last query they ran.
* **Cons**: Whatever’s using the database might not be connected at the moment, or might be using a different database context.
* **Verdict**: A good first step.

**Are Reads and Writes Happening on Tables in the Database?**

Sure, it’s possible that stored procedures or functions are in use that don’t use any local tables, but just looking at whether data access is being done tells you a lot fast.

My favorite way to do this is with [sp\_BlitzIndex®](http://www.brentozar.com/blitzindex/). Run it with the @mode=2 parameter, copy and paste the data into a spreadsheet, and look at the “Index Usage” column. You can also see the timestamps of most recent seeks, scans, and writes.

[view source](http://www.brentozar.com/archive/2014/05/4-lightweight-ways-tell-database-used/#viewSource)



[print](http://www.brentozar.com/archive/2014/05/4-lightweight-ways-tell-database-used/#printSource)[?](http://www.brentozar.com/archive/2014/05/4-lightweight-ways-tell-database-used/#about)

|  |
| --- |
| exec sp\_BlitzIndex |
| @DatabaseName='AdventureWorks2012', | |

|  |  |
| --- | --- |
| @mode=2; | |
| GO |

* **Pros**: Very lightweight, and lets you know which tables are in use.
* **Cons**: If you have fulltext indexes, you may see reads from the fulltext indexing service on those tables (even if users aren’t querying the fulltext index).
* **Verdict**: Very useful information. A good second step.

**Is the Transaction Counter Going Up for the Database?**

SQL Server keeps a cumulative track of activity, and you can check this to see if it’s currently going up:

[view source](http://www.brentozar.com/archive/2014/05/4-lightweight-ways-tell-database-used/#viewSource)



[print](http://www.brentozar.com/archive/2014/05/4-lightweight-ways-tell-database-used/#printSource)[?](http://www.brentozar.com/archive/2014/05/4-lightweight-ways-tell-database-used/#about)

|  |
| --- |
| SELECT \* |
| FROM sys.dm\_os\_performance\_counters | |

|  |
| --- |
| WHERE counter\_name like 'Transactions/sec%' |
| and instance\_name like 'AdventureWorks2012%'; | |

|  |
| --- |
| GO |

* **Pros**: Super lightweight, no extra tools needed.
* **Cons**: Some reads will occur just in starting up SQL Server. You can’t tell reads from maintenance from reads from user transactions. (Full and log backups will increment the counter.) Also, strangely enough, just plain SELECT statements outside of an explicitly defined transaction don’t cause this counter to go up. (I know, weird, right???)
* **How to use it**: Take a couple of samples and see if the counter is going up dramatically. If it is going up, that means something’s fairly busy in there. I wouldn’t try to get too scientific with this one, though– it’s just too limited.
* **Verdict**: Kind of clunky, but could be useful along with the other methods here.

**Are there user Execution Plans in the Cache for the Database?**

You can also ask the execution plan cache if queries have been running against the database. This query takes advantage of the dm\_exec\_text\_query\_plan DMV (as recommended here by the very clever [Grant Fritchey](http://www.scarydba.com/2012/07/02/querying-data-from-the-plan-cache/)):

[view source](http://www.brentozar.com/archive/2014/05/4-lightweight-ways-tell-database-used/#viewSource)



[print](http://www.brentozar.com/archive/2014/05/4-lightweight-ways-tell-database-used/#printSource)[?](http://www.brentozar.com/archive/2014/05/4-lightweight-ways-tell-database-used/#about)

|  |
| --- |
| SELECT |
| SUBSTRING(tx.1, | |

|  |
| --- |
| (qs.statement\_start\_offset / 2) + 1, |
| (CASE WHEN qs.statement\_end\_offset =-1 THEN DATALENGTH(tx.text) ELSE qs.statement\_end\_offset END - qs.statement\_start\_offset) | |

|  |
| --- |
| / 2 + 1) AS QueryText, |
| case when pl.query\_plan LIKE '%<MissingIndexes>%' then 1 else 0 end as [Missing Indexes?], | |

|  |
| --- |
| qs.execution\_count, |
| qs.total\_worker\_time/execution\_count AS avg\_cpu\_time, | |

|  |
| --- |
| qs.total\_worker\_time AS total\_cpu\_time, |
| qs.total\_logical\_reads/execution\_count AS avg\_logical\_reads, | |

|  |
| --- |
| qs.total\_logical\_reads, |
| qs.creation\_time AS [plan creation time], | |

|  |  |
| --- | --- |
| qs.last\_execution\_time [last execution time], | |
| CAST(pl.query\_plan AS XML) AS sqlplan |

|  |
| --- |
| FROM    sys.dm\_exec\_query\_stats AS qs |
| CROSS APPLY sys.dm\_exec\_text\_query\_plan(qs.plan\_handle, qs.statement\_start\_offset, qs.statement\_end\_offset) AS pl | |

|  |  |
| --- | --- |
| CROSS APPLY sys.dm\_exec\_sql\_text(qs.sql\_handle) AS tx | |
| WHERE pl.query\_plan LIKE '%[AdventureWorks2012]%' |

|  |  |
| --- | --- |
| ORDER BY execution\_count DESC OPTION (RECOMPILE); | |
| GO |

* **Pros**: Gives you insight into *what* is using the database when it finds results
* **Cons**: Doesn’t catch anything that’s not in the cache due to RECOMPILE hints or memory pressure over time. You’ll probably have to slog through some rows of system procedures.
* **Verdict**: not perfect, but very information-rich if something does turn out to be using the database (and a whole lot quicker and more lightweight than tracing).

**I know there’s a way I haven’t THOUGHT of.**

Do you have a lightweight technique you use for this which I haven’t mentioned here? Let me know in the comments!

[↑ Back to top](http://www.brentozar.com/archive/2014/05/4-lightweight-ways-tell-database-used/#top)

19 comments

[Leave a comment](http://www.brentozar.com/archive/2014/05/4-lightweight-ways-tell-database-used/#comments)

1. http://1.gravatar.com/avatar/71309d968667122293252a9b19ae1076?s=50&d=http%3A%2F%2F1.gravatar.com%2Favatar%2Fad516503a11cd5ca435acc9bb6523536%3Fs%3D50&r=G

[*Klaus Aschenbrenner*](http://www.SQLpassion.at) [May 5, 2014 | 9:45 am](http://www.brentozar.com/archive/2014/05/4-lightweight-ways-tell-database-used/#comment-895763)

Another solution: take the database offfline, and wait until the phone rings…

-Klaus

[Reply](http://www.brentozar.com/archive/2014/05/4-lightweight-ways-tell-database-used/?replytocom=895763#respond)

* + http://1.gravatar.com/avatar/188c850f0ea2fea2bababc9414158e8b?s=50&d=http%3A%2F%2F1.gravatar.com%2Favatar%2Fad516503a11cd5ca435acc9bb6523536%3Fs%3D50&r=G

*Per Hejndorf* [May 5, 2014 | 11:04 am](http://www.brentozar.com/archive/2014/05/4-lightweight-ways-tell-database-used/#comment-895899)

This is what in the army is known as “recon by fire” :)

[Reply](http://www.brentozar.com/archive/2014/05/4-lightweight-ways-tell-database-used/?replytocom=895899#respond)

* + http://0.gravatar.com/avatar/a478f8bd3eaddbbcb15087896c2a58d2?s=50&d=http%3A%2F%2F0.gravatar.com%2Favatar%2Fad516503a11cd5ca435acc9bb6523536%3Fs%3D50&r=G

*Kendra Little* [May 5, 2014 | 11:16 am](http://www.brentozar.com/archive/2014/05/4-lightweight-ways-tell-database-used/#comment-895918)

LOL!

[Reply](http://www.brentozar.com/archive/2014/05/4-lightweight-ways-tell-database-used/?replytocom=895918#respond)

* + http://0.gravatar.com/avatar/8f9ec496df70d9a601fed3e06dad35b0?s=50&d=http%3A%2F%2F0.gravatar.com%2Favatar%2Fad516503a11cd5ca435acc9bb6523536%3Fs%3D50&r=G

*Andrew Notarian* [May 5, 2014 | 12:07 pm](http://www.brentozar.com/archive/2014/05/4-lightweight-ways-tell-database-used/#comment-896035)

Wenn Nutzer kommt, flach auf den Boden legen und auf Hilfe warten.

(Whenever I see anything from Klaus I think of a great joke in German)

[Reply](http://www.brentozar.com/archive/2014/05/4-lightweight-ways-tell-database-used/?replytocom=896035#respond)

1. http://0.gravatar.com/avatar/8f9ec496df70d9a601fed3e06dad35b0?s=50&d=http%3A%2F%2F0.gravatar.com%2Favatar%2Fad516503a11cd5ca435acc9bb6523536%3Fs%3D50&r=G

*Andrew Notarian* [May 5, 2014 | 9:49 am](http://www.brentozar.com/archive/2014/05/4-lightweight-ways-tell-database-used/#comment-895771)

I had to do this a lot in the SQL 2000 days and some things I would look at would be if the transaction log was growing or do differential backups and see how big they were. Is this too simplistic? Obviously that would only show transaction activity, which can be not enough to prove a database is used.

[Reply](http://www.brentozar.com/archive/2014/05/4-lightweight-ways-tell-database-used/?replytocom=895771#respond)

* + http://0.gravatar.com/avatar/a478f8bd3eaddbbcb15087896c2a58d2?s=50&d=http%3A%2F%2F0.gravatar.com%2Favatar%2Fad516503a11cd5ca435acc9bb6523536%3Fs%3D50&r=G

*Kendra Little* [May 5, 2014 | 11:17 am](http://www.brentozar.com/archive/2014/05/4-lightweight-ways-tell-database-used/#comment-895922)

Oh interesting! Yeah, it only covers modification and if you have index maintenance running that could cause some false positives, but it definitely is a possible source of info.

[Reply](http://www.brentozar.com/archive/2014/05/4-lightweight-ways-tell-database-used/?replytocom=895922#respond)

* + - http://0.gravatar.com/avatar/8f9ec496df70d9a601fed3e06dad35b0?s=50&d=http%3A%2F%2F0.gravatar.com%2Favatar%2Fad516503a11cd5ca435acc9bb6523536%3Fs%3D50&r=G

*Andrew Notarian* [May 5, 2014 | 12:05 pm](http://www.brentozar.com/archive/2014/05/4-lightweight-ways-tell-database-used/#comment-896030)

True. If there was index maintenance. These were mostly tiny databases that never got an index and we were just trying to tell if they could be safely dropped.

[Reply](http://www.brentozar.com/archive/2014/05/4-lightweight-ways-tell-database-used/?replytocom=896030#respond)

1. http://0.gravatar.com/avatar/af56894ff951a62ad9b7ce83c5755f4c?s=50&d=http%3A%2F%2F0.gravatar.com%2Favatar%2Fad516503a11cd5ca435acc9bb6523536%3Fs%3D50&r=G

[*Ben Thul*](http://www.spartansql.com) [May 5, 2014 | 10:22 am](http://www.brentozar.com/archive/2014/05/4-lightweight-ways-tell-database-used/#comment-895824)

If you suspect that a given table isn’t being used (or, worse, is only being written to implying that the data isn’t ever being read) is SQL Audit. Granted, you have to have some idea as to whether or not it’s being used or the audit file gets large quickly, but I’ve trimmed some junk using this process. As a bonus, you get the statements that are running.

[Reply](http://www.brentozar.com/archive/2014/05/4-lightweight-ways-tell-database-used/?replytocom=895824#respond)

* + http://0.gravatar.com/avatar/a478f8bd3eaddbbcb15087896c2a58d2?s=50&d=http%3A%2F%2F0.gravatar.com%2Favatar%2Fad516503a11cd5ca435acc9bb6523536%3Fs%3D50&r=G

*Kendra Little* [May 5, 2014 | 11:18 am](http://www.brentozar.com/archive/2014/05/4-lightweight-ways-tell-database-used/#comment-895925)

Heh, I’m not sure I’d call auditing simple OR easy, but if you’ve got EE and the know how to get Audit rigged up then it absolutely gives a lot of info. Cool to hear that you’ve used this successfully!

[Reply](http://www.brentozar.com/archive/2014/05/4-lightweight-ways-tell-database-used/?replytocom=895925#respond)

* + http://0.gravatar.com/avatar/c49b08fb87089c954b8aa2f4c3607239?s=50&d=http%3A%2F%2F0.gravatar.com%2Favatar%2Fad516503a11cd5ca435acc9bb6523536%3Fs%3D50&r=G

*Nic Neufeld* [May 5, 2014 | 3:57 pm](http://www.brentozar.com/archive/2014/05/4-lightweight-ways-tell-database-used/#comment-896421)

Timely stuff…just started looking at Audit today to solve that very problem (auditing specific table access and specific proc execution when the devs assume they aren’t used any longer). I was first worried about performance impact as compared to a server side trace but it sounds like its actually more lightweight than a trace, being built off of XEvents.

[Reply](http://www.brentozar.com/archive/2014/05/4-lightweight-ways-tell-database-used/?replytocom=896421#respond)

1. http://0.gravatar.com/avatar/e7decfa8df3ffb705665620aab80dbcb?s=50&d=http%3A%2F%2F0.gravatar.com%2Favatar%2Fad516503a11cd5ca435acc9bb6523536%3Fs%3D50&r=G

*anand* [May 5, 2014 | 10:39 am](http://www.brentozar.com/archive/2014/05/4-lightweight-ways-tell-database-used/#comment-895858)

Is dm index usage stats not a good idea? We keep a daily snapshot of indeed usage stats for 2 months, if we see no usage we offline for a month and then delete after a external tape backup with 12 month retention.

Have a semi automated process For SQL 2000. We had a trace running for 30 min window to capture which db is used and the next trace filters out the for DB that have already captured as used. The more DB we captured, we increased the time for which the trace would run. We automated stopping trace after 30 mins, reading the trace to capture the db used and add filter to the New trace.

Any better idea please let us know, have removed 30 SQL 2000 instances, 20 more to go :).

Thanks  
Anand

[Reply](http://www.brentozar.com/archive/2014/05/4-lightweight-ways-tell-database-used/?replytocom=895858#respond)

* + http://0.gravatar.com/avatar/a478f8bd3eaddbbcb15087896c2a58d2?s=50&d=http%3A%2F%2F0.gravatar.com%2Favatar%2Fad516503a11cd5ca435acc9bb6523536%3Fs%3D50&r=G

*Kendra Little* [May 5, 2014 | 11:22 am](http://www.brentozar.com/archive/2014/05/4-lightweight-ways-tell-database-used/#comment-895936)

Usage stats can definitely be helpful– that’s basically what I’m referring to in the “Are reads and writes happening?” section.

On SQL Server 2012 and higher, index rebuilds can clear out those index usage stats, so that might give you bad info if you sampled right after index maintenance and you’re using rebuilds.

For SQL Server 2000, that’s a pretty tricky situation. I like how you’ve approached it: you don’t have a ton of tools or options, and you’ve designed a way where you’re trying to minimize your trace footprint.

[Reply](http://www.brentozar.com/archive/2014/05/4-lightweight-ways-tell-database-used/?replytocom=895936#respond)

1. http://0.gravatar.com/avatar/65196baecb5ae284a595d72a80eb51a6?s=50&d=http%3A%2F%2F0.gravatar.com%2Favatar%2Fad516503a11cd5ca435acc9bb6523536%3Fs%3D50&r=G

*JohnR* [May 5, 2014 | 1:24 pm](http://www.brentozar.com/archive/2014/05/4-lightweight-ways-tell-database-used/#comment-896178)

So I have this run every 5 to 10 min. It does 2 things…  
1. will show what hosts are connected (but not necessarily whats being executed) and how many connections each host has to each db  
2. if there are more than x threshold it starts to dump what is being executed by said connection.

Its my “go to” when I want to find whats connected and if there’s a service that isn’t letting go of the db connection like it should.

FYI … dbname and author were removed to protect … well just to protect ;)

USE []  
GO  
/\*\*\*\*\*\* Object: StoredProcedure [dbo].[ConnectionsStatus] Script Date: 05/05/2014 13:58:55 \*\*\*\*\*\*/  
SET ANSI\_NULLS ON  
GO  
SET QUOTED\_IDENTIFIER ON  
GO  
– =============================================  
– Author:  
– Create date: 04/07/2012  
– Description: Inserts current connections, active and possibly inactive, into .dbo.ConnectionsActive & .dbo.ConnectionsInactive tables  
– =============================================  
ALTER PROCEDURE [dbo].[ConnectionsStatus]

AS  
BEGIN  
– SET NOCOUNT ON added to prevent extra result sets from  
– interfering with SELECT statements.  
SET NOCOUNT ON;

— connection count by database, host, login  
INSERT INTO .dbo.ConnectionsActive  
(DatabaseName, HostName, LoginName, ConnectionCount)  
SELECT d.name as DatabaseName, p.hostname, loginame, count(\*) as ConnectionCount  
FROM master..sysprocesses p  
join master..sysdatabases d on p.dbid = d.dbid  
GROUP BY d.name, p.hostname, loginame  
ORDER BY 4 desc

– inactive connection count by database, host, login  
INSERT INTO .dbo.ConnectionsInactive  
(DatabaseName, HostName, LoginName, ConnectionCount)  
SELECT d.name as DatabaseName, p.hostname, loginame, count(\*) as ConnectionCount  
FROM master..sysprocesses p  
JOIN master..sysdatabases d on p.dbid = d.dbid  
WHERE datediff (minute,p.last\_batch, getdate()) > 2  
GROUP BY d.name, p.hostname, loginame  
ORDER BY 4 desc

— Connections Active Detail

DECLARE @MaxConnectionCount INT = 95 –set max amount of connection count to begin showing last executed sprocs  
DECLARE @cntr int =1  
DECLARE @max INT  
DECLARE @spid INT  
DECLARE @spids AS TABLE (id INT IDENTITY (1,1), spid INT, DatabaseName VARCHAR(255), HostName VARCHAR(255), LoginName VARCHAR(255), textdata VARCHAR(Max))

–SELECT d.name as DatabaseName, p.hostname, loginame, count(\*) as ConnectionCount  
–FROM master..sysprocesses p  
–JOIN master..sysdatabases d on p.dbid = d.dbid  
–GROUP BY d.name, p.hostname, loginame  
–ORDER BY 4 desc

INSERT INTO @spids  
(  
spid ,  
DatabaseName ,  
HostName ,  
LoginName  
)  
SELECT spid, d.name AS databasename, p.hostname, p.loginame  
FROM master..sysprocesses p  
JOIN master..sysdatabases d on p.dbid = d.dbid  
JOIN  
( SELECT d.name as DatabaseName, p.hostname, loginame  
FROM master..sysprocesses p  
join master..sysdatabases d on p.dbid = d.dbid  
GROUP BY d.name, p.hostname, loginame  
HAVING COUNT(\*) > @MaxConnectionCount  
) RU  
ON RU.DatabaseName = d.name AND RU.hostname = p.hostname AND RU.loginame = p.loginame

SELECT @max = (SELECT MAX(id) FROM @spids)

WHILE @cntr <= @max

BEGIN

DECLARE @sqltext VARBINARY(128)  
SELECT @sqltext = sp.sql\_handle  
FROM sys.sysprocesses sp  
JOIN @spids s  
ON s.id = @cntr  
WHERE sp.spid = s.spid

SELECT @spid = (SELECT spid FROM @spids WHERE id = @cntr)

UPDATE s  
SET Textdata = txt.textdata  
FROM @spids s  
JOIN (  
SELECT TEXT AS textdata, @spid spid  
FROM sys.dm\_exec\_sql\_text(@sqltext)  
) txt  
ON txt.spid = s.spid

SELECT @cntr = @cntr + 1

END

INSERT INTO .dbo.ConnectionsActiveDetail  
( DatabaseName ,  
HostName ,  
LoginName ,  
spid ,  
TextData  
)  
SELECT  
DatabaseName,  
HostName,  
LoginName,  
spid,  
textdata  
FROM @spids  
WHERE textdata IS NOT NULL

DELETE  
FROM .dbo.ConnectionsActiveDetail  
WHERE TimeStamp < GETDATE() – 60

END

[Reply](http://www.brentozar.com/archive/2014/05/4-lightweight-ways-tell-database-used/?replytocom=896178#respond)

* + http://0.gravatar.com/avatar/65196baecb5ae284a595d72a80eb51a6?s=50&d=http%3A%2F%2F0.gravatar.com%2Favatar%2Fad516503a11cd5ca435acc9bb6523536%3Fs%3D50&r=G

*JohnR* [May 5, 2014 | 1:27 pm](http://www.brentozar.com/archive/2014/05/4-lightweight-ways-tell-database-used/#comment-896180)

Looks like db name got removed… well you should get the idea anyway

[Reply](http://www.brentozar.com/archive/2014/05/4-lightweight-ways-tell-database-used/?replytocom=896180#respond)

1. http://1.gravatar.com/avatar/5d53f9e28d9bde9778cf4832cad345b5?s=50&d=http%3A%2F%2F1.gravatar.com%2Favatar%2Fad516503a11cd5ca435acc9bb6523536%3Fs%3D50&r=G

[*Michael J Swart*](http://michaeljswart.com) [May 5, 2014 | 1:35 pm](http://www.brentozar.com/archive/2014/05/4-lightweight-ways-tell-database-used/#comment-896199)

Clunky or not, I’ve really come to like the Transactions/sec counter this year. It’s been especially useful when filtering on tempdb. (Even though I find it hard to understand what that counter is doing without experimenting.)

BTW: You know you’ve truly made it when you illustrate a blog article with a picture of a [burning database.](http://michaeljswart.com/wp-content/uploads/2013/04/charmax.png)

[Reply](http://www.brentozar.com/archive/2014/05/4-lightweight-ways-tell-database-used/?replytocom=896199#respond)

1. http://0.gravatar.com/avatar/8d13f379fc0a663cbf2e1a0aeae3968e?s=50&d=http%3A%2F%2F0.gravatar.com%2Favatar%2Fad516503a11cd5ca435acc9bb6523536%3Fs%3D50&r=G

*Stefan P* [May 6, 2014 | 8:56 am](http://www.brentozar.com/archive/2014/05/4-lightweight-ways-tell-database-used/#comment-898307)

I always start by running a ‘netstat’. Of course this is server-wide, not per database.

[Reply](http://www.brentozar.com/archive/2014/05/4-lightweight-ways-tell-database-used/?replytocom=898307#respond)

* + http://1.gravatar.com/avatar/77f776c2eaf0cc691e8a0880bb8a191f?s=50&d=http%3A%2F%2F1.gravatar.com%2Favatar%2Fad516503a11cd5ca435acc9bb6523536%3Fs%3D50&r=G

[*Brent Ozar*](http://www.brentozar.com) [May 6, 2014 | 1:29 pm](http://www.brentozar.com/archive/2014/05/4-lightweight-ways-tell-database-used/#comment-898852)

Stefan – right, how does that help here with multiple DBs?

[Reply](http://www.brentozar.com/archive/2014/05/4-lightweight-ways-tell-database-used/?replytocom=898852#respond)

1. http://1.gravatar.com/avatar/f82daed15ec9fa3fdcfbb7a7b672880e?s=50&d=http%3A%2F%2F1.gravatar.com%2Favatar%2Fad516503a11cd5ca435acc9bb6523536%3Fs%3D50&r=G

*Jaime González* [May 12, 2014 | 12:20 pm](http://www.brentozar.com/archive/2014/05/4-lightweight-ways-tell-database-used/#comment-916352)

Here’s my solution based in sys.dm\_db\_index\_usage\_stats view.  
Please comment, is secure ?

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/  
/\* SCRIPT DATABASE REPORTING FROM UNUSED SQL LAST RESET SERVICE \*/  
/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/  
use master  
go  
SELECT DB\_NAME() as ‘dbname’,  
convert(varchar(255), t.name) AS ‘Table’,  
SUM(i.user\_seeks + i.user\_scans + i.user\_lookups)  
AS ‘Total accesses’,  
SUM(i.user\_seeks) AS ‘Seeks’,  
SUM(i.user\_scans) AS ‘Scans’,  
SUM(i.user\_lookups) AS ‘Lookups’  
INTO ##tb\_tables\_used  
FROM  
sys.dm\_db\_index\_usage\_stats i RIGHT OUTER JOIN  
sys.tables t ON (t.object\_id = i.object\_id)  
GROUP BY  
i.object\_id,  
t.name ORDER BY [Total accesses] DESC  
go  
delete ##tb\_tables\_used  
go

exec sp\_msforeachdb ‘use ?;  
insert into ##tb\_tables\_used  
SELECT ”?” as ”dbname”,  
t.name AS ”Table”,  
SUM(i.user\_seeks + i.user\_scans + i.user\_lookups)  
AS ”Total accesses”,  
SUM(i.user\_seeks) AS ”Seeks”,  
SUM(i.user\_scans) AS ”Scans”,  
SUM(i.user\_lookups) AS ”Lookups”  
FROM  
sys.dm\_db\_index\_usage\_stats i RIGHT OUTER JOIN  
sys.tables t ON (t.object\_id = i.object\_id)  
GROUP BY  
i.object\_id,  
t.name ORDER BY [Total accesses] DESC ‘  
go

SELECT DATEDIFF(D, create\_date, GETDATE()) as ‘Running Days’, CREATE\_DATE AS ‘Restarting from the’  
FROM SYS.databases  
WHERE name = ‘TEMPDB’

select name as ‘Databases without Use’  
from sys.databases  
where name not in (‘model’)  
except  
select dbname  
from ##tb\_tables\_used  
order by 1

[Reply](http://www.brentozar.com/archive/2014/05/4-lightweight-ways-tell-database-used/?replytocom=916352#respond)

* + http://0.gravatar.com/avatar/a478f8bd3eaddbbcb15087896c2a58d2?s=50&d=http%3A%2F%2F0.gravatar.com%2Favatar%2Fad516503a11cd5ca435acc9bb6523536%3Fs%3D50&r=G

*Kendra Little* [May 12, 2014 | 2:16 pm](http://www.brentozar.com/archive/2014/05/4-lightweight-ways-tell-database-used/#comment-916643)

Hey there,

That’s essentially what I’m describing in the section, “ARE READS AND WRITES HAPPENING ON TABLES IN THE DATABASE?” So it has the same pros and cons.

Kendra