**Identify SQL Server databases that are no longer in use**

**SQL Server User Connection Count**

One suggestion to finding orphan databases is to get connection counts. In most cases, if a database has zero user connections over a long period of time, it may be time to look into removing this database. The following query will capture server name, database name, number of connections, and time the query was executed and it will also filter out system databases because they are needed:

SELECT @@ServerName AS server

,NAME AS dbname

,COUNT(STATUS) AS number\_of\_connections

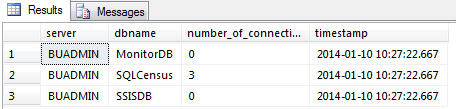
,GETDATE() AS timestamp

FROM sys.databases sd

LEFT JOIN sysprocesses sp ON sd.database\_id = sp.dbid

WHERE database\_id NOT BETWEEN 1 AND 4

GROUP BY NAME



I'm using a server named BUADMIN for this example. As you can see I have 3 active connections to the database SQLCensus. This is a good indication that this database is in use. MonitorDB and SSISDB have 0 connections, so I may need to monitor them further. The easiest way to monitor these databases is to create a stored procedure using this query so I can schedule it. You can also put this query directly into a [SQL Server Agent Job](http://www.mssqltips.com/sql-server-tip-category/27/sql-server-agent/) and set a schedule.

Before setting a schedule, you will need to create a table that will hold the results. To create a table using the following code:

SET ANSI\_NULLS ON

GO

SET QUOTED\_IDENTIFIER ON

GO

CREATE TABLE [dbo].[Connections](

[server] [nvarchar](130) NOT NULL,

[name] [nvarchar](130) NOT NULL,

[number\_of\_connections] [int] NOT NULL,

[timestamp] [datetime] NOT NULL

) ON [PRIMARY]

GO

Next, create a [stored procedure](http://www.mssqltips.com/sqlservertutorial/160/sql-server-stored-procedure/) that will [INSERT](http://www.mssqltips.com/sqlservertutorial/2514/sql-server-insert-command/) the results into the table:

SET ANSI\_NULLS ON

GO

SET QUOTED\_IDENTIFIER ON

GO

CREATE PROCEDURE usp\_ConnectionsCount

AS

BEGIN

SET NOCOUNT ON;

INSERT INTO Connections

SELECT @@ServerName AS server

,NAME AS dbname

,COUNT(STATUS) AS number\_of\_connections

,GETDATE() AS timestamp

FROM sys.databases sd

LEFT JOIN master.dbo.sysprocesses sp ON sd.database\_id = sp.dbid

WHERE database\_id NOT BETWEEN 1

AND 4

GROUP BY NAME

END

Once the stored procedure is created you can create a SQL Server Agent Job and set it to run on a schedule. I'll set it to run every 10 minutes.

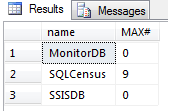
Let this run a few days, a few months or however long you think is appropriate and then go back and examine the results. Once you are happy with the timeframe chosen, use the following query to select the MAX number of connections per database:

SELECT NAME

,MAX(number\_of\_connections) AS MAX#

FROM Connections

GROUP BY NAME



From here you will be able to determine if any databases have not had a user connection in the timeframe specified.

**Detailed SQL Server Connection Information**

The above suggestion is good if you just need connection counts. However, sometimes a count isn't good enough. Sometimes you need to know exactly what is connecting. This suggestion helps in that aspect.

It's basically setup the same way, create a [stored procedure](http://www.mssqltips.com/sqlservertutorial/160/sql-server-stored-procedure/), [insert](http://www.mssqltips.com/sqlservertutorial/2514/sql-server-insert-command/) data into a table, set a [schedule](http://www.mssqltips.com/sql-server-tip-category/27/sql-server-agent/) and [examine the results](http://www.mssqltips.com/sqlservertutorial/10/select-command-for-sql-server/).

The following query gives you more information:

SELECT @@ServerName AS SERVER

,NAME

,login\_time

,last\_batch

,getdate() AS DATE

,STATUS

,hostname

,program\_name

,nt\_username

,loginame

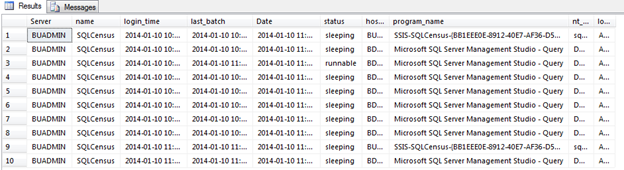
FROM sys.databases d

LEFT JOIN sysprocesses sp ON d.database\_id = sp.dbid

WHERE database\_id NOT BETWEEN 0

AND 4

AND loginame IS NOT NULL



This query is a detailed view of what is happening. As you can see, currently I have 10 connections to the SQLCensus database. You can view information such as login time, last batch, status hostname, program name, and login name. I've resized some of the columns for security purposes, but this is a good way to see where connections are coming from and what login is making the connection.

If you choose to use this suggestion, simply do what I explained above. Create a stored procedure, table, and job.