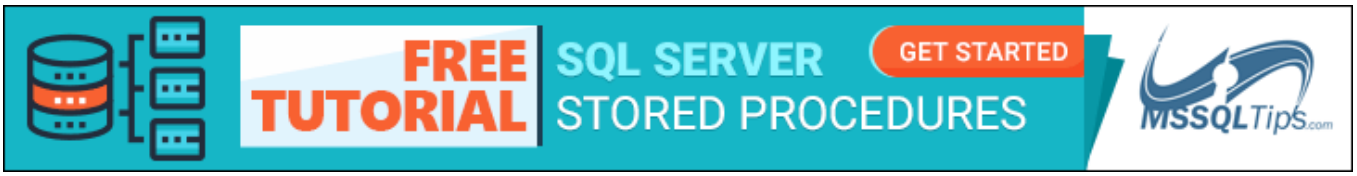
MSSQLTips Free SQL Server Conference Registration + Travel Expenses click to learn more

Discovering Unused Indexes



By: Greg Robidoux

Overview

To ensure that data access can be as fast as possible, SQL Server like other relational database systems utilizes indexing to find data quickly. SQL Server has different types of indexes that can be created such as clustered indexes, non-clustered indexes, XML indexes and Full Text indexes.

The benefit of having more indexes is that SQL Server can access the data quickly if an appropriate index exists. The downside to having too many indexes is that SQL Server has to maintain all of these indexes which can slow things down and indexes also require additional storage. So as you can see indexing can both help and hurt performance.

In this section we will focus on how to identify indexes that exist, but are not being used and therefore can be dropped to improve performance and decrease storage requirements.

Explanation

When SQL Server 2005 was introduced it added Dynamic Management Views (DMVs) that allow you to get additional insight as to what is going on within SQL Server. One of these areas is the ability to see how indexes are being used. There are two DMVs that we will discuss. Note that these views store cumulative data, so when SQL Server is restated the counters go back to zero, so be aware of this when monitoring your index usage.

DMV - sys.dm_db_index_operational_stats

This DMV allows you to see insert, update and delete information for various aspects for an index. Basically this shows how much effort was used in maintaining the index based on data changes.

If you query the table and return all columns, the output may be confusing. So the query below focuses on a few key columns. To learn more about the output for all columns you can check out Books Online.

```
SELECT OBJECT_NAME(A.[OBJECT_ID]) AS [OBJECT NAME],
       I.[NAME] AS [INDEX NAME],
       A.LEAF_INSERT_COUNT,
       A.LEAF_UPDATE_COUNT,
       A.LEAF_DELETE_COUNT
FROM   SYS.DM_DB_INDEX_OPERATIONAL_STATS (db_id(),NULL,NULL,NULL ) A
       INNER JOIN SYS.INDEXES AS I
         ON I.[OBJECT_ID] = A.[OBJECT_ID]
            AND I.INDEX_ID = A.INDEX_ID
WHERE  OBJECTPROPERTY(A.[OBJECT_ID],'IsUserTable') = 1
```

Below we can see the number of Inserts, Updates and Deletes that occurred for each index, so this shows how much work SQL Server had to do to maintain the index.

	OBJECT NAME	INDEX NAME	LEAF_INSERT_COUNT	LEAF_UPDATE_COUNT	LEAF_DELETE_COUNT
1	ContactCreditCard	PK_ContactCreditCard_ContactID_CreditCardID	1	0	0
2	CountryRegionCurrency	PK_CountryRegionCurrency_CountryRegionCode_...	13	6036	0
3	Employee	PK_Employee_EmployeeID	0	0	0

DMV - sys.dm_db_index_usage_stats

This DMV shows you how many times the index was used for user queries. Again there are several other columns that are returned if you query all columns and you can refer to Books Online for more information.

```
SELECT OBJECT_NAME(S.[OBJECT_ID]) AS [OBJECT NAME],
       I.[NAME] AS [INDEX NAME],
       USER_SEEKS,
       USER_SCANS,
       USER_LOOKUPS,
       USER_UPDATES
FROM   SYS.DM_DB_INDEX_USAGE_STATS AS S
       INNER JOIN SYS.INDEXES AS I ON I.[OBJECT_ID] = S.[OBJECT_ID] AND I.INDEX_ID = S.INDEX_ID
WHERE  OBJECTPROPERTY(S.[OBJECT_ID],'IsUserTable') = 1
       AND S.database_id = DB_ID()
```

Here we can see seeks, scans, lookups and updates.

- The seeks refer to how many times an index seek occurred for that index. A seek is the fastest way to access the data, so this is good.
- The scans refers to how many times an index scan occurred for that index. A scan is when multiple rows of data had to be searched to find the data. Scans are something you want to try to avoid.
- The lookups refer to how many times the query required data to be pulled from the clustered index or the heap (does not have a clustered index). Lookups are also something you want to try to avoid.
- The updates refers to how many times the index was updated due to data changes which should correspond to the first query above.

	OBJECT NAME	INDEX NAME	USER_SEEKS	USER_SCANS	USER_LOOKUPS	USER_UPDATES
9	CountryRegionCurrency	PK_CountryRegionCurrency_CountryRegionCode_Curre...	6036	0	0	6037
10	CustomerAddress	AK_CustomerAddress_rowguid	0	0	0	93
11	CustomerAddress	PK_CustomerAddress_CustomerID_AddressID	93	0	0	93

Identifying Unused Indexes

So based on the output above you should focus on the output from the second query. If you see indexes where there are no seeks, scans or lookups, but there are updates this means that SQL Server has not used the index to satisfy a query but still needs to maintain the index. Remember that the data from these DMVs is reset when SQL Server is restarted, so make sure you have collected data for a long enough period of time to determine which indexes may be good candidates to be dropped.

Additional Information

Here are some additional articles about indexes.

- Deeper insight into unused indexes for SQL Server
- Find unused SQL Server indexes with Policy Based Management
- Clustered Tables vs Heap Tables
- Avoid Index Redundancy in SQL Server Tables
- Retaining historical index usage statistics for SQL Server Part

<< Previous Next >>

More SQL Server Solutions

SQL Monitor
Get your free ROI calculation
Start now

SQL Server Performance Tuning Tutorial

Get Free SQL Server Tips

Post a comment or let the author know this tip helped.

All comments are reviewed, so stay on subject or we may delete your comment. Note: your email address is not published. Required fields are marked with an asterisk (*).

*Name *Email Notify for updates

*** NOTE *** - If you want to include code from SQL Server Management Studio (SSMS) in your post, please copy the code from SSMS and paste the code into a text editor like NotePad before copying the code below to remove the SSMS formatting.

p

Send me SQL tips:

I'm not a robot

reCAPTCHA
Privacy - Terms

Save Comment

Tuesday, August 01, 2017 - 9:43:42 AM - Greg Robidoux Back To Top

Hi Vicki,

yes you are correct. The index is not being used to make queries run faster, but SQL Server still has to maintain the index which is extra overhead for storage and maintenance processing.

Just make sure you have the system running for a good amount of time to collect a good sample of how the indexes are being used before dropping. Also, it is a good idea to script out the index in case there is a need to recreate it in the future.

-Greg

Tuesday, August 01, 2017 - 9:06:39 AM - Vicki Back To Top

My DBA and I have both read this article but interpret one line differently. Can you help?

*"If you see indexes where there are no seeks, scans or lookups, but there are updates this means that SQL Server has not used the index to satisfy a query but **still needs** to maintain the index."*

He believes "still needs" to mean the index is important while I believe it means the index is a shore that we really don't need. Can you clarify?

Get Free SQL Tips

enter business email Join

Copyright (c) 2006-2018 Edgewood Solutions, LLC All rights reserved

Some names and products listed are the registered trademarks of their respective owners.