Duplicate Indexes, What a Waste

Today on the tuning minute on the SQL Data Partners Podcast, we discussed duplicate indexes, which lead me to think more about and and write this post.

You know there are many different ways of doing things in SQL Server, and often times you can argue that one way or the other is better, and given the right situation anything might be a good idea. However duplicate indexes are a different story.

When I talk about duplicate indexes, what I mean is 2 or more indexes on the same table that are exactly the same columns. Something like this:

Two indexes on exactly the same column. There is nothing to be gained here.

SQL Server will let you create many duplicate indexes on the same table. I think is is something like 999 in SQL 2008R2 or older, and even more in newer versions. But don't do it, duplicate indexes have the following negative impact:

- Increases database size with no value.
- Increase database backup size with no value.
- Slows inserts, updates and deletes because there are duplicate indexes that need to be updated.
- Can slow queries as the query optimizer needs to consider multiple indexes when analyzing query performance and determining a plan.
- Increase Deadlocks.
- Bloated transaction log size.
- Add to the amount of data that replicated databases need to transfer.

You can see how many negatives there are associated with duplicate indexes that have no reason to be there.

How can I find duplicate indexes?

The way I find duplicate indexes is with the Database Health Monitor application, you can use the duplicate indexes report and the duplicated indexes advisor to get a script to drop those duplicate indexes.

The duplicate indexes report gives you the details on the tables with the biggest duplicate indexes.

And by double clicking on one of the tables listed in the duplicate indexes report, you can get to the duplicate index advisor. where you can see the details, and the recommendations on what indexes could be dropped.

You can clean up your system by tracking down duplicate indexes and removing them.

Another Point Of View... Perhaps?

However.... I wonder if you are using SQL Server Standard Edition, where you don't have online index rebuilding operations if it would make sense to have two identical indexes, one to cover for the other when it is getting rebuilt, and vice versa. I am not promoting this, just wondering if anyone has tried this as an option to get around the offline index rebuilds in Standard Edition.

Related Links

- SQL Data Partners Podcast #61
- Database Health Monitor
- Database Health Monitor Download Page
- Duplicate Indexes at DatabaseHealth.com