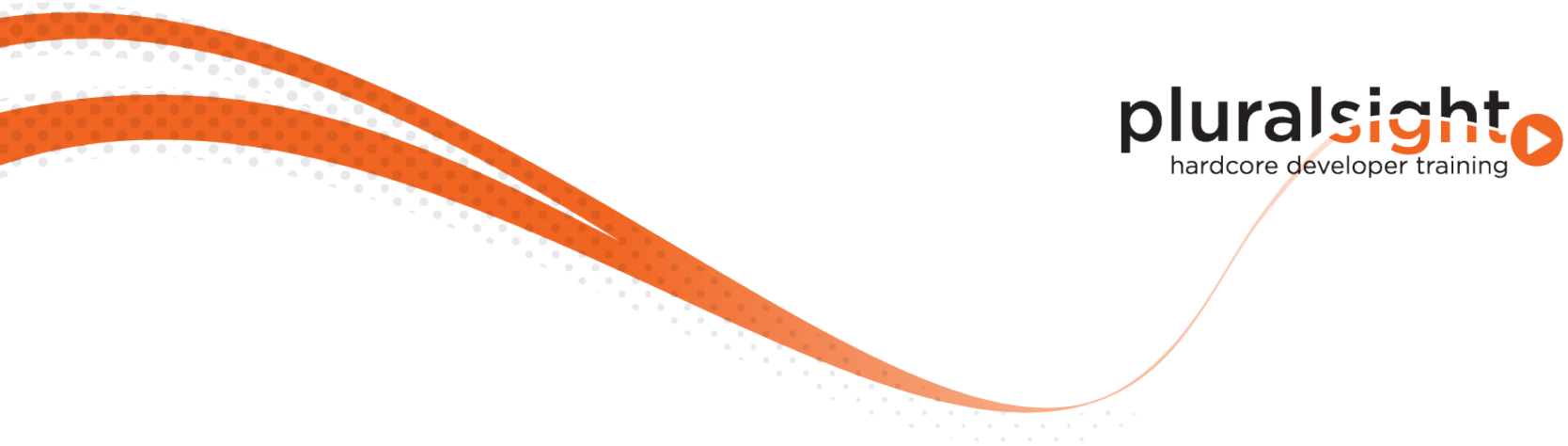
SQL Server: Myths and Misconceptions

Module 4: Indexing

Paul S. Randal
Paul@SQLskills.com



pluralsight
hardcore developer training

Introduction

- **A proper indexing strategy is the nearest thing SQL Server has a /FASTER switch**
  - Indexes need to be chosen carefully
  - Index maintenance needs to be performed

- **In this module:**
  - Seven myths around indexes and index maintenance

Indexing Myth #1

UNTRUE!!

- **Myth: An index FILLFACTOR is used by regular DML operations**

- **FILLFACTOR only applies during index build and rebuild operations**
- **The same goes for the PAD_INDEX setting**

- **Also, a FILLFACTOR of 0 is the same as a FILLFACTOR of 100**
- **And there is no formula for determining the correct FILLFACTOR**
  - Pick a number (usually 70) and put it in production
  - Monitor fragmentation
  - Tweak the FILLFACTOR up or down, or change the periodicity of index maintenance

Indexing Myth #2

UNTRUE!!

- **Myth: Rebuilding a clustered index always rebuilds all nonclustered indexes**

- **This does not happen in SQL Server 2005 onwards**
- **In SQL Server 2000 and before, this could happen…**
  - A non-unique clustered index has a 'uniquifier' column that serves to allow a unique index key to be generated
  - Non-clustered indexes include the clustered index key (and any uniquifier)
  - SQL Server 2000 would regenerate the clustered index uniquifiers on a rebuild, invalidating all the nonclustered indexes
  - SQL Server 2000 SP1 also had a bug that caused nonclustered indexes to always be rebuilt when the clustered index was rebuilt

Indexing Myth #3

UNTRUE!!

- **Myth: A nonclustered index should be created for each column in the table**

- **Nonclustered indexes should only be created where needed**
- **Nonclustered indexes are only used in query plans if:**
  - They are selective enough AND
  - They are the most efficient choice for getting the query results
- **It's unlikely that an index on each column will be useful**
- **Use sys.dm_db_index_usage_stats to check whether an index is being used**
  - Make sure to check over an entire business cycle
  - Potentially drop those that are not being used

Indexing Myth #4

UNTRUE!!

- **Myth: Using SSDs means you can ignore index fragmentation**

- **You must always be concerned about index fragmentation**
- **Index fragmentation has two forms:**
  - Logical fragmentation that stops efficient readahead
  - Page density that wastes space
- **SSDs make I/Os faster, but do you want to have an expensive SSD storing a lot of empty space?**
- **Index fragmentation occurs through page splits**
  - Page splits are expensive operations
  - Page splits generate a lot of transaction log, which has to be processed

Indexing Myth #5

TRUE!!

- **Myth: Heap fragmentation can be fixed by creating and dropping a clustered index**

- **Well, yes, you can do it that way but you shouldn't**
  - Creating the clustered index rebuilds all nonclustered indexes
  - Dropping the clustered index rebuilds all nonclustered indexes
- **And don't use ALTER TABLE … REBUILD either**
  - It will cause all nonclustered indexes to be rebuilt too
- **If fragmentation is an issue, create a good clustered index… and leave it there!**

Indexing Myth #6

UNTRUE!!

- **Myth: A GUID is a good cluster key**

- **GUIDs are usually random and so cause index fragmentation**
  - As a GUID is a random value, that make the index key random
  - A random insertion will make index pages fill up and eventually have to split
- **GUIDs are 16 bytes long**
  - This leads to extra space required, including in nonclustered indexes
- **GUIDs created using NEWSEQUENTIALID are not random**
  - But are still 16 bytes long
- **A better choice would be an INT or BIGINT**
  - 4 and 8 bytes wide, respectively
  - Append-only insertion pattern

Indexing Myth #7

MAYBE!!

- **Myth: Statistics must be rebuilt after an index rebuild**

- **Index rebuilds always rebuild index column statistics**
  - They are rebuilt with the equivalent of a full scan
- **Non-index column statistics are not rebuilt**
- **Index reorganizes do not rebuild any statistics**
- **Statistics maintenance should be performed for those statistics that are not automatically rebuilt by an index rebuild**