Agenda

The two SQL Server index usage views:

- By plan: sys.dm_db_index_usage_stats
- By index: sys.dm_db_index_operational_stats

Why they're not as accurate as you might suspect

How sp_BlitzIndex shows the results

How I interpret the results to do the D.E. parts

1.3 p2

SQL Server has a lot of metrics.

Old-school, operating system level: Perfmon counters

System & database level: "system tables & views"

- Dynamic Management Views (DMVs)
- Dynamic Management Functions (DMFs)

Tracing: Profiler, Extended Events

1.3 p3

Dynamic Management Views

The good:

- Well-documented by both Microsoft and blogs
- It's easy to find scripts and tools that use 'em

The bad:

- A lot of the user-written documentation is wrong
- Many of the DMVs don't mean what you think
- Contents can reset at surprising times
- Hit-or-miss coverage in Azure, keeps changing

1.3 p4

sys.dm_db_index_usage_stats

Shows # of executions where a plan included an operator

- Does NOT show if the operator was used (or how often it was accessed)

Number and last date of reads (seeks, scans, lookups)

Number and last date of last write

- Insert/update/deletes all called “updates”

Data is since startup or when the index was modified

1.3 p5

sys.dm_db_index_operational_stats

Lower level, more transitory

Lock waits (page and row)

Access counts

- Doesn't distinguish between full scans/range scans, or even range scans and seeks

Data only persisted while object's metadata is in memory

No good way when to tell it was last cleared

1.3 p6

A lot of tools use this data.

T-SQL scripts: sp_BlitzIndex, sp_HelpIndex, Glenn Berry's DMV scripts

Apps: SentryOne Plan Explorer, lots of monitoring tools like Quest Spotlight, Red Gate SQL Monitor

You may have written your own scripts too (let's be honest, you copied it from someone else's online, and you don't really know what it's doing)

1.3 p7

I'm going to be using my favorite.

sp_BlitzIndex in the First Responder Kit:

- Github repo: FirstResponderKit.org
- Zip download: BrentOzar.com/first-aid
- Slack chat: SQLslack.com, #FirstResponderKit

Lots of code contributors, used all over the world

Open source, free as in speech, MIT License

1.3 p8

1.3 p9

What this means

You can deploy this anywhere, forever, for free.

No one can take it from you.

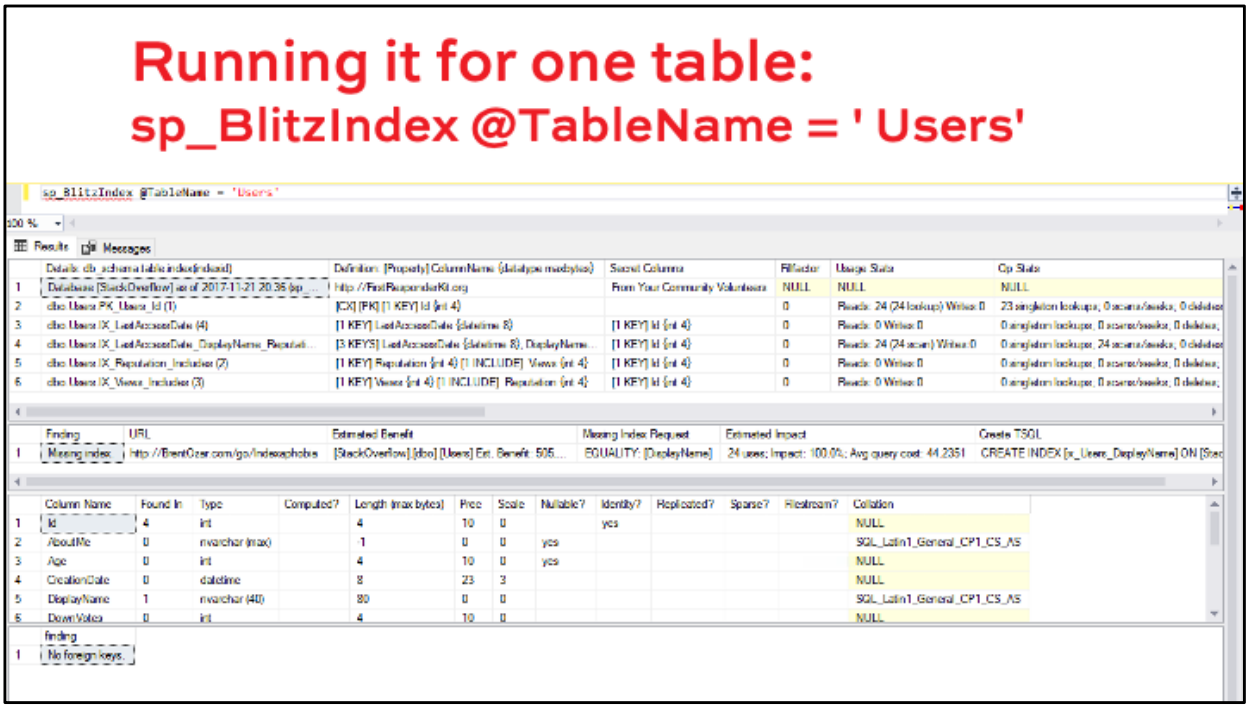
You can use it at clients or other companies.

You can even take the code and use it in products.

1.3 p10

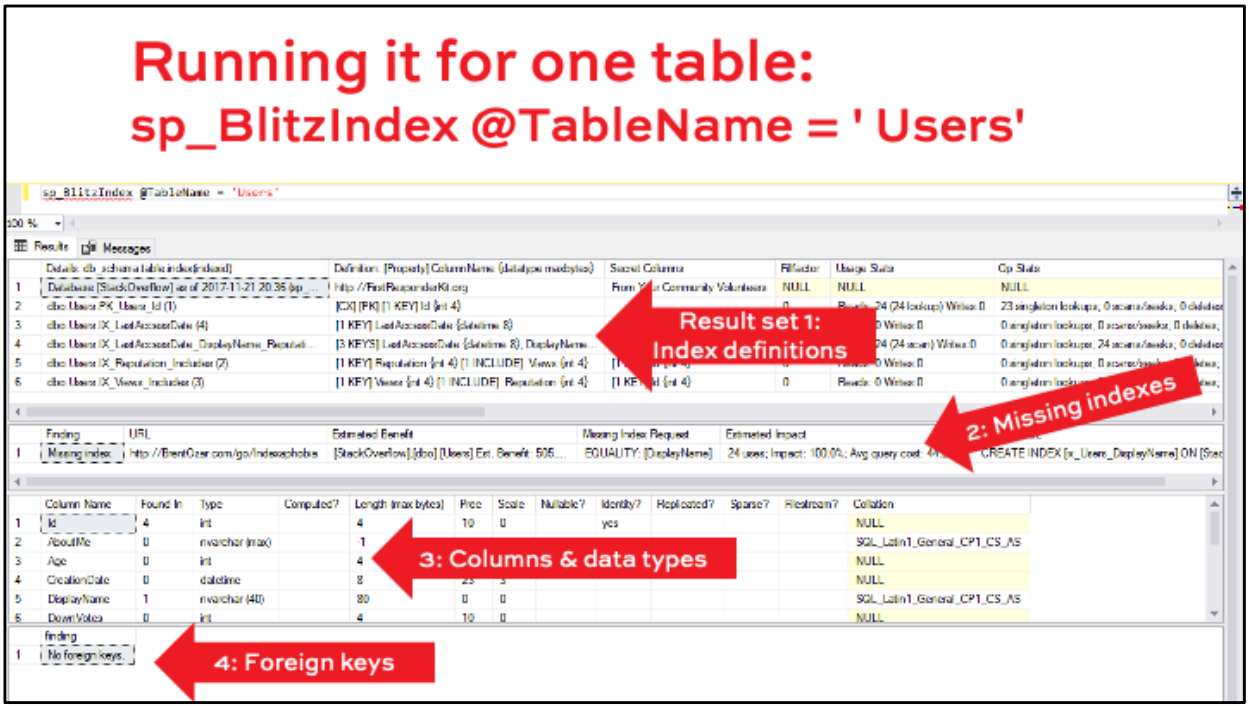
Running it for one table:

sp_BlitzIndex @TableName = 'Users'

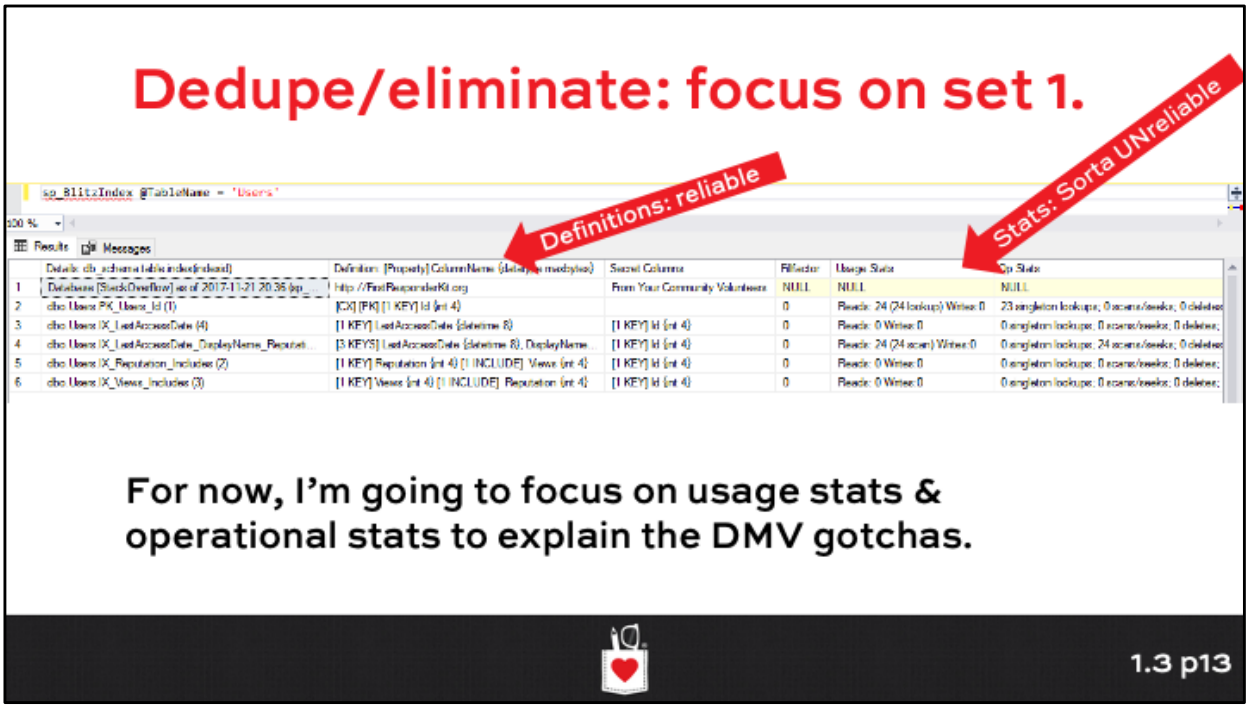


Running it for one table:

sp_BlitzIndex @TableName = 'Users'

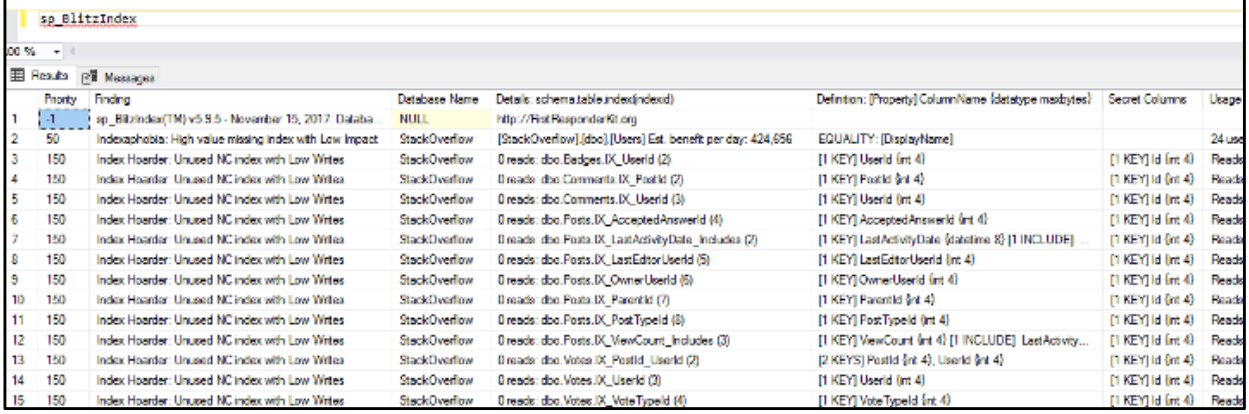


Dedupe/eliminate: focus on set 1.



Later, I'll use it across a database.

Defaults to the current database,
or you can pick one with @DatabaseName parameter

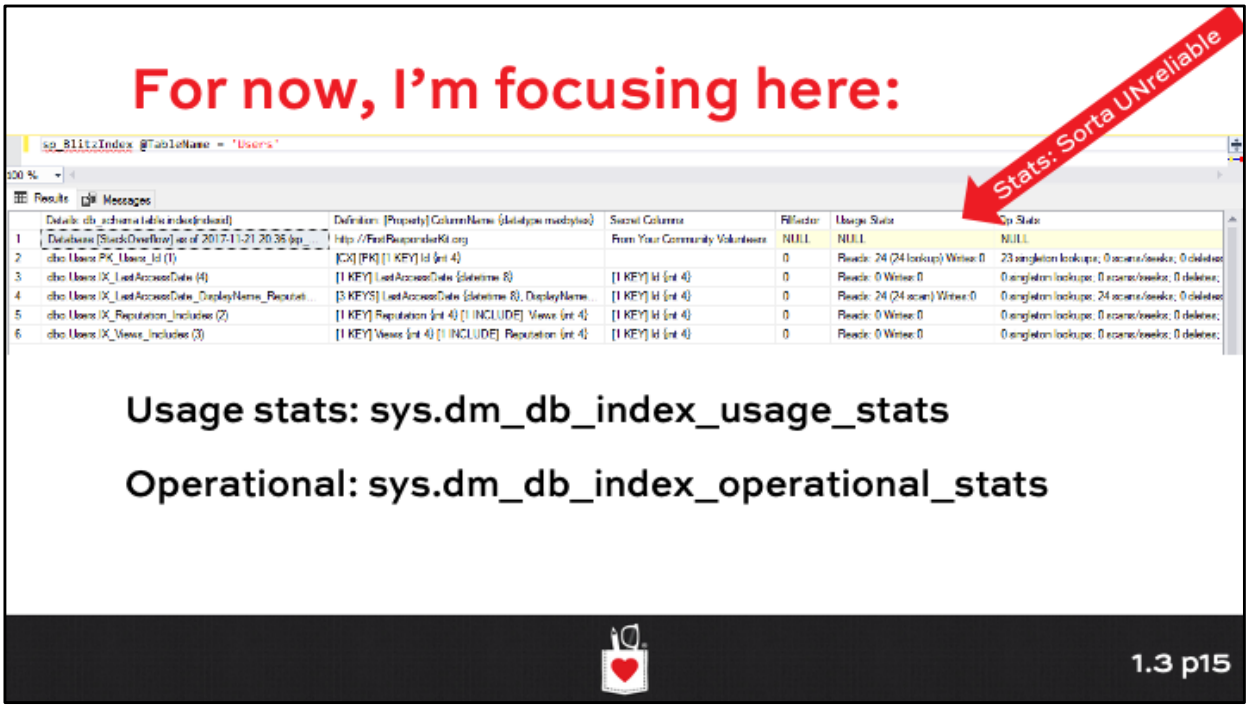


```
sp_BlitzIndex
```

00 %

Results | Messages

	Priority	Finding	Database Name	Details: schema.table.index(indexid)	Definition: [Property] ColumnName {datatype maxbytes}	Secret Columns	Usage
1	-1	sp_BlitzIndex(TM) v5.9.5 - November 15, 2017 Databa...	NULL	http://FirstResponderKit.org			
2	50	Indexaphobia: High value missing index with Low Impact	StackOverflow	[StackOverflow].[dbo].[Users] Est. benefit per day: 424,656	EQUALITY: [DisplayName]		24 use
3	150	Index Hoarder: Unused NC index with Low Writes	StackOverflow	0 reads: dbo.Badges.IX_UserId (2)	[1 KEY] UserId {int 4}	[1 KEY] Id {int 4}	Reads
4	150	Index Hoarder: Unused NC index with Low Writes	StackOverflow	0 reads: dbo.Comments.IX_PostId (2)	[1 KEY] PostId {int 4}	[1 KEY] Id {int 4}	Reads
5	150	Index Hoarder: Unused NC index with Low Writes	StackOverflow	0 reads: dbo.Comments.IX_UserId (3)	[1 KEY] UserId {int 4}	[1 KEY] Id {int 4}	Reads
6	150	Index Hoarder: Unused NC index with Low Writes	StackOverflow	0 reads: dbo.Posts.IX_AcceptedAnswerId (4)	[1 KEY] AcceptedAnswerId {int 4}	[1 KEY] Id {int 4}	Reads
7	150	Index Hoarder: Unused NC index with Low Writes	StackOverflow	0 reads: dbo.Posts.IX_LastActivityDate_Includes (2)	[1 KEY] LastActivityDate {datetime 8} [1 INCLUDE]...	[1 KEY] Id {int 4}	Reads
8	150	Index Hoarder: Unused NC index with Low Writes	StackOverflow	0 reads: dbo.Posts.IX_LastEditorUserId (5)	[1 KEY] LastEditorUserId {int 4}	[1 KEY] Id {int 4}	Reads
9	150	Index Hoarder: Unused NC index with Low Writes	StackOverflow	0 reads: dbo.Posts.IX_OwnerUserId (6)	[1 KEY] OwnerUserId {int 4}	[1 KEY] Id {int 4}	Reads
10	150	Index Hoarder: Unused NC index with Low Writes	StackOverflow	0 reads: dbo.Posts.IX_ParentId (7)	[1 KEY] ParentId {int 4}	[1 KEY] Id {int 4}	Reads
11	150	Index Hoarder: Unused NC index with Low Writes	StackOverflow	0 reads: dbo.Posts.IX_PostTypeId (8)	[1 KEY] PostTypeId {int 4}	[1 KEY] Id {int 4}	Reads
12	150	Index Hoarder: Unused NC index with Low Writes	StackOverflow	0 reads: dbo.Posts.IX_ViewCount_Includes (3)	[1 KEY] ViewCount {int 4} [1 INCLUDE] LastActivity...	[1 KEY] Id {int 4}	Reads
13	150	Index Hoarder: Unused NC index with Low Writes	StackOverflow	0 reads: dbo.Votes.IX_PostId_UserId (2)	[2 KEYS] PostId {int 4}, UserId {int 4}	[1 KEY] Id {int 4}	Reads
14	150	Index Hoarder: Unused NC index with Low Writes	StackOverflow	0 reads: dbo.Votes.IX_UserId (3)	[1 KEY] UserId {int 4}	[1 KEY] Id {int 4}	Reads
15	150	Index Hoarder: Unused NC index with Low Writes	StackOverflow	0 reads: dbo.Votes.IX_VoteTypeId (4)	[1 KEY] VoteTypeId {int 4}	[1 KEY] Id {int 4}	Reads



Build the execution plan for this.

```
SELECT TOP 10 Id
FROM dbo.Users
ORDER BY LastAccessDate;
GO
```

Flash back to How to Think Like the Engine:

- Clustered index on Id (white pages)
- Nonclustered on LastAccessDate, Id (black pages)

1.3 p16

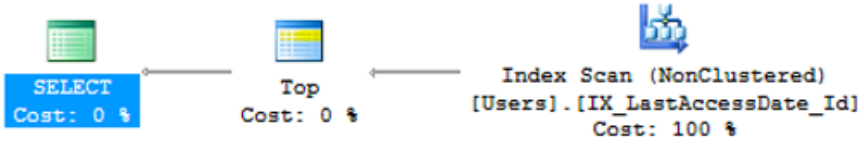
It's a "scan" – but what does that mean?

```
23  SELECT TOP 10 Id
24  FROM dbo.Users
25  ORDER BY LastAccessDate;
26  GO
```

100 %

Results | Messages | Execution plan

```
Query 1: Query cost (relative to the batch): 100%
SELECT TOP 10 Id FROM dbo.Users ORDER BY LastAccessDate
```



1.3 p17

It's efficient –
it doesn't scan the whole index.

```
23  SELECT TOP 10 Id
24  FROM dbo.Users
25  ORDER BY LastAccessDate;
26  GO
```

100 %

Results | Messages | Execution plan

```
SQL Server parse and compile time:
   CPU time = 0 ms, elapsed time = 0 ms.

 SQL Server Execution Times:
   CPU time = 0 ms,  elapsed time = 0 ms.
SQL Server parse and compile time:
   CPU time = 0 ms, elapsed time = 0 ms.

(10 row(s) affected)
Table 'Users'. Scan count 1, logical reads 3, physical reads 0, read-ahead reads 0,
```

1.3 p18

It's just labeled a scan

The DMVs don't distinguish between types of scans

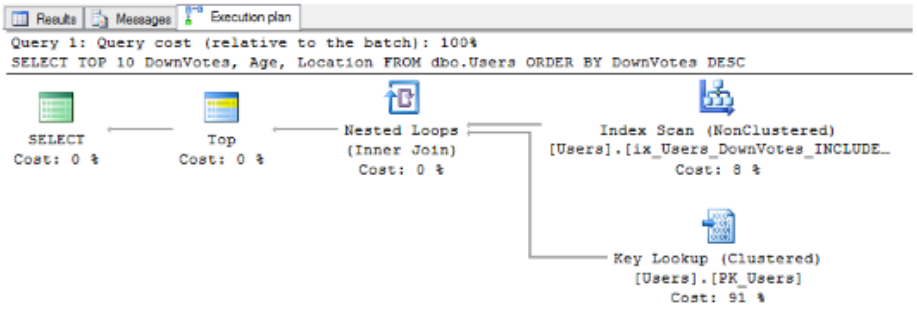
sp_BlitzIndex @TableName = 'Users'

	Details: db_schema.table.index(indexid)	Definition: [Property] ColumnName {datatype maxbytes}	Usage Stats	Op Stats
1	Database [StackOverflow] as of 2017-07-23 17:09 (sp_...	http://FirstResponderKit.org	NULL	NULL
2	dbo.Users.IX_CreationDate_Reputation_Filtered (7)	[2 KEYS] CreationDate {datetime 8}, Reputation {int ...	Reads: 0 Writes:0	0 singleton lookups; 0 scans/seeks; 0 deletes; 0...
3	dbo.Users.IX_DownVotes_Includes (5)	[1 KEY] DownVotes {int 4} [7 INCLUDES] Age {int 4...	Reads: 0 Writes:0	0 singleton lookups; 0 scans/seeks; 0 deletes; 0...
4	dbo.Users.IX_Id (8)	[1 KEY] Id {int 4}	Reads: 0 Writes:0	0 singleton lookups; 0 scans/seeks; 0 deletes; 0...
5	dbo.Users.PK_Users_Id (1)	[CX] [PK] [1 KEY] Id {int 4}	Reads: 0 Writes:0	0 singleton lookups; 0 scans/seeks; 0 deletes; 0...
6	dbo.Users.IX_LastAccessDate_Id (2)	[2 KEYS] LastAccessDate {datetime 8}, Id {int 4}	Reads: 1 (1 scan) Writes:0	0 singleton lookups; 1 scans/seeks; 0 deletes; 0...
7	dbo.Users.IX_LastAccessDate_Id_DisplayName_Age (3)	[4 KEYS] LastAccessDate {datetime 8}, Id {int 4}, Dis...	Reads: 0 Wri...	0 singleton lookups; 0 s.../seeks; 0 deletes; 0...

1.3 p19

This plan has a key lookup

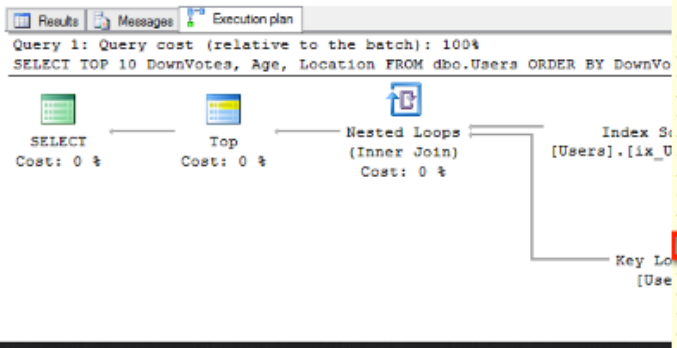
For every row from the nonclustered index scan, it looks up related values in the clustered index



Key lookup is executed 10 times

Index scan returns 10 rows

Each one has to go do a lookup



Key Lookup (Clustered)

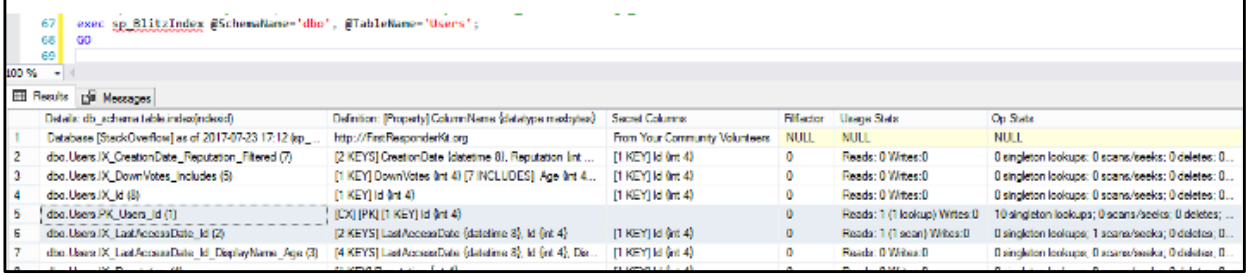
Uses a supplied clustering key to lookup on a table that has a clustered index.

Physical Operation	Key Lookup
Logical Operation	Key Lookup
Actual Execution Mode	Row
Estimated Execution Mode	Row
Storage	RowStore
Actual Number of Rows	10
Actual Number of Batches	0
Estimated I/O Cost	0.003125
Estimated Operator Cost	0.0361113 (91%)
Estimated CPU Cost	0.0001581
Estimated Subtree Cost	0.0361113
Number of Executions	10
Estimated Number of Executions	11
Estimated Number of Rows	1
Estimated Row Size	111 B
Actual Rebinds	0
Actual Rewinds	0
Ordered	True
Node ID	5

3 p21

The stats are different

- **sys.dm_db_index_usage_stats:** Number of times the operator appeared in an execution plan since last reset
- **sys.dm_db_index_operational_stats:** Number of times the operator was executed (recently)



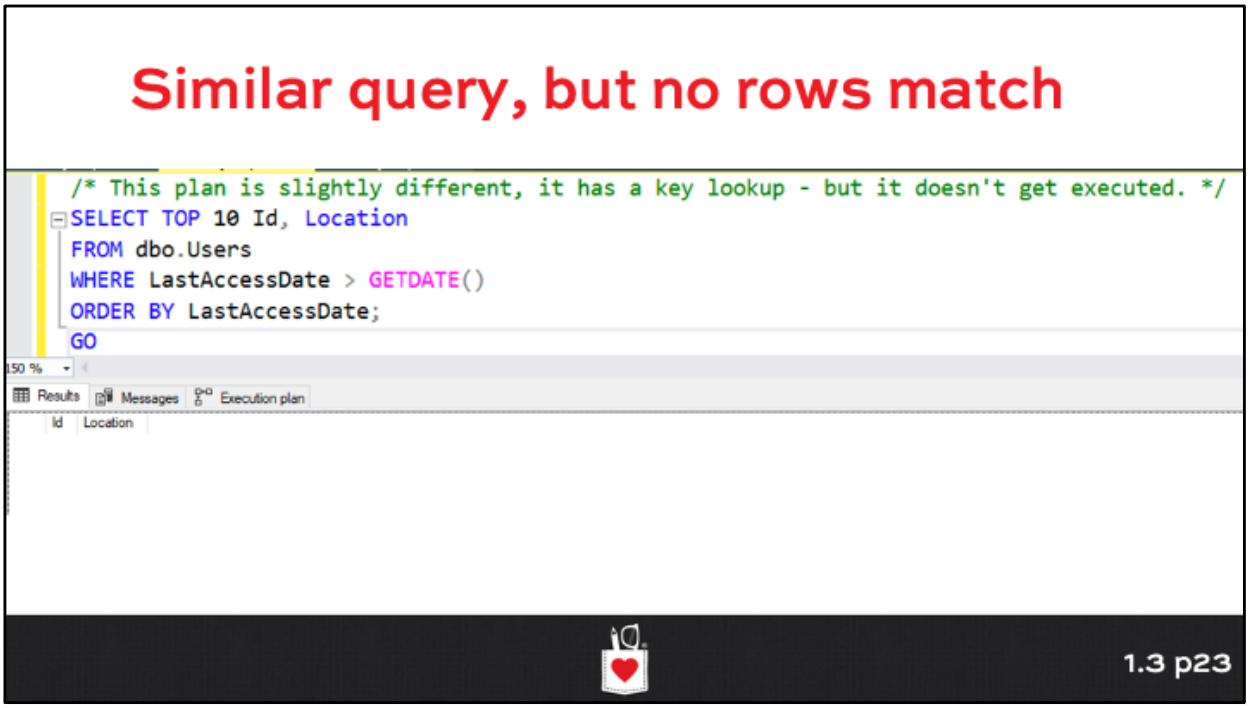
```
exec sp_BlitzIndex @SchemaName='dbo', @TableName='Users';
GO
```

100 %

Results | Messages

	Details: db_schema.table.index(indexid)	Definition: [Property] ColumnName (datatype maxbytes)	Secret Columns	Fillfactor	Usage Stats	Op Stats
1	Database [StackOverflow] as of 2017-07-23 17:12 (sp_...	http://FirstResponderKit.org	From Your Community Volunteers	NULL	NULL	NULL
2	dbo.Users.IX_CreationDate_Reputation_Filtered (7)	[2 KEYS] CreationDate (datetime 8), Reputation (int ...	[1 KEY] Id (int 4)	0	Reads: 0 Writes:0	0 singleton lookups: 0 scans/seeks: 0 deletes: 0...
3	dbo.Users.IX_DownVotes_Includes (5)	[1 KEY] DownVotes (int 4) [7 INCLUDES] Age (int 4...	[1 KEY] Id (int 4)	0	Reads: 0 Writes:0	0 singleton lookups: 0 scans/seeks: 0 deletes: 0...
4	dbo.Users.IX_Id (6)	[1 KEY] Id (int 4)	[1 KEY] Id (int 4)	0	Reads: 0 Writes:0	0 singleton lookups: 0 scans/seeks: 0 deletes: 0...
5	dbo.Users.PK_Users_Id (1)	[CX] [PK] [1 KEY] Id (int 4)		0	Reads: 1 (1 lookup) Writes:0	10 singleton lookups; 0 scans/seeks: 0 deletes; ...
6	dbo.Users.IX_LastAccessDate_Id (2)	[2 KEYS] LastAccessDate (datetime 8), Id (int 4)	[1 KEY] Id (int 4)	0	Reads: 1 (1 scan) Writes:0	0 singleton lookups; 1 scans/seeks; 0 deletes; 0...
7	dbo.Users.IX_LastAccessDate_Id_DisplayName_Age (3)	[4 KEYS] LastAccessDate (datetime 8), Id (int 4), Dis...	[1 KEY] Id (int 4)	0	Reads: 0 Writes:0	0 singleton lookups: 0 scans/seeks: 0 deletes: 0...

Similar query, but no rows match



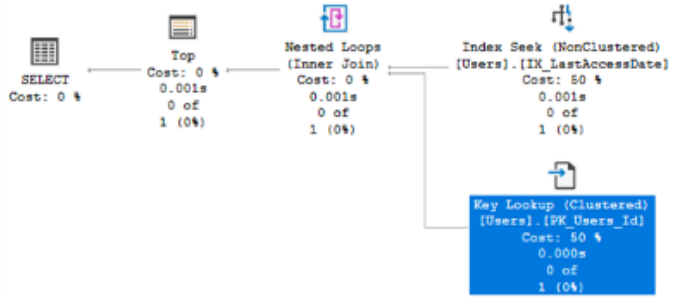
It doesn't execute the lookup

```
/* This plan is slightly different, it has a key lookup - but it doesn't get executed. */
SELECT TOP 10 Id, Location
FROM dbo.Users
WHERE LastAccessDate > GETDATE()
ORDER BY LastAccessDate;
GO
```

150 %

Results | Messages | Execution plan

Query 1: Query cost (relative to the batch): 100%
SELECT TOP 10 Id, Location FROM dbo.Users WHERE LastAccessDate > GETDATE() ORDER BY LastAccessDate



What the DMVs say

Usage stats says the plan included the seek

Operational stats says it wasn't executed

```
/* How does that show up in the DMVs? */
exec sp_BlitzIndex @SchemaName='dbo', @TableName='Users';
```

150 %

Results | Messages

	Details: db_schema.table index(indexid)	Definition: [Property] ColumnName {datatype maxbytes}	Usage Stats	Op Stats
1	Database [StackOverflow2013] as of 2019-06-11 10...	http://FirstResponderKit.org	Server: SQL2019 Days Uptime: 0.04	NULL
2	dbo.Users.PK_Users_Id (1)	[CX] [PK] [1 KEY] Id {int 4}	Reads: 1 (1 lookup) Writes:0	0 singleton lookups; 0 scans/seeks; 0
3	dbo.Users.IX_LastAccessDate (2)	[1 KEY] LastAccessDate {datetime 8}	Reads: 1 (1 seek) Writes:0	0 singleton lookups; 1 scans/seeks; 0

1.3 p25

So, the contents don't match.

When I use them, I'm really just asking:

- Is this index helping? (reads)
- Is this index hurting? (writes)
- Roughly how much? (quantity: millions, billions)

But your next question is,
"When did these numbers reset?"

1.3 p26

Uh, well…I can't tell.

When SQL Server restarts (which we can measure)

When an Availability Group failed over (harder to tell)

When Azure SQL DB fails over, restarts (can't see)

SQL 2012, 2014: resets on ALTER INDEX REBUILD

- SQL 2012: fixed in SP2 CU12, or SP3 CU3
- SQL 2014: fixed in RTM CU14, or SP1 CU8, or SP2
- SQL Server 2016 & newer: unaffected

1.3 p27

Index DMVs: Your takeaways

- "Scan" may not be the whole table
- "Seek" might actually be the whole table

sys.dm_db_index_usage_stats - "usage stats"

- Show # of times an operator appeared in a query plan that was run
- The operator may have been accessed many times, or not at all
- Reset by system restart, or by index rebuild if on buggy versions

sys.dm_db_index_operational_stats – "op stats"

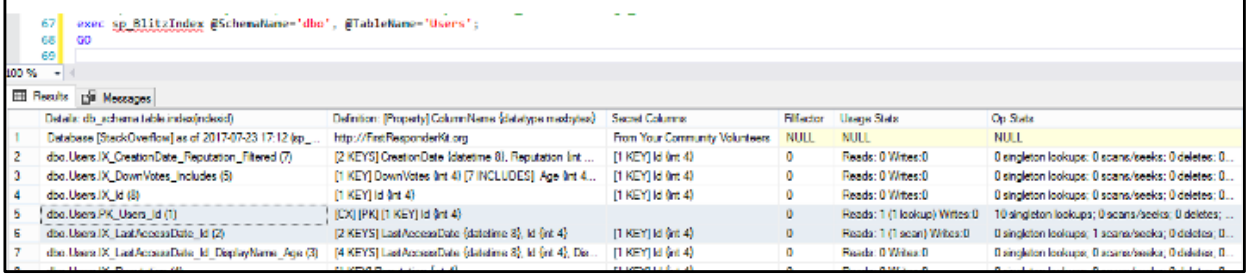
- Show number of times an operator was accessed
- Very volatile, can be reset by memory pressure

Only check when you have enough uptime to reflect business processes.

I'm not saying not to use these.

I'm just saying don't put too much faith in the details.

Your goal is just to know, "Are these indexes kinda getting used, or totally ignored?"



	Details: db_schema.table.index(indexid)	Definition: [Property] ColumnName (datatype maxbytes)	Secret Columns	Fillfactor	Usage Stats	Op Stats
1	Database [StackOverflow] as of 2017-07-23 17:12 (sp_...	http://FirstResponderKit.org	From Your Community Volunteers	NULL	NULL	NULL
2	dbo.Users IX_CreationDate_Reputation_Filtered (7)	[2 KEYS] CreationDate (datetime 8), Reputation (int ...	[1 KEY] Id (int 4)	0	Reads: 0 Writes:0	0 singleton lookups: 0 scans/seeks: 0 deletes: 0...
3	dbo.Users IX_DownVotes_Includes (5)	[1 KEY] DownVotes (int 4) [7 INCLUDES] Age (int 4...	[1 KEY] Id (int 4)	0	Reads: 0 Writes:0	0 singleton lookups: 0 scans/seeks: 0 deletes: 0...
4	dbo.Users IX_Id (6)	[1 KEY] Id (int 4)	[1 KEY] Id (int 4)	0	Reads: 0 Writes:0	0 singleton lookups: 0 scans/seeks: 0 deletes: 0...
5	dbo.Users PK_Users_Id (1)	[CX] [PK] [1 KEY] Id (int 4)		0	Reads: 1 (1 lookup) Writes:0	10 singleton lookups; 0 scans/seeks: 0 deletes; ...
6	dbo.Users IX_LastAccessDate_Id (2)	[2 KEYS] LastAccessDate (datetime 8), Id (int 4)	[1 KEY] Id (int 4)	0	Reads: 1 (1 scan) Writes:0	0 singleton lookups; 1 scans/seeks; 0 deletes; 0...
7	dbo.Users IX_LastAccessDate_Id_DisplayName_Age (3)	[4 KEYS] LastAccessDate (datetime 8), Id (int 4), Dis...	[1 KEY] Id (int 4)	0	Reads: 0 Writes:0	0 singleton lookups: 0 scans/seeks: 0 deletes: 0...

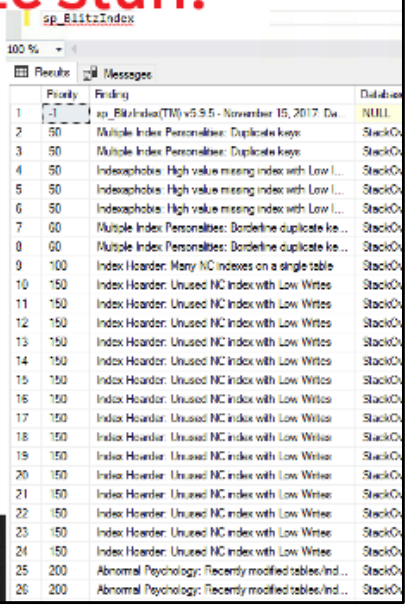
Our scripts try to prioritize stuff.

We want you to focus on the biggest bang-for-the-buck first.

Findings here are a little more fun – we use psychiatry terms.

There's nothing *wrong* with some psychiatric disorders, either.

Let your freak flag fly.

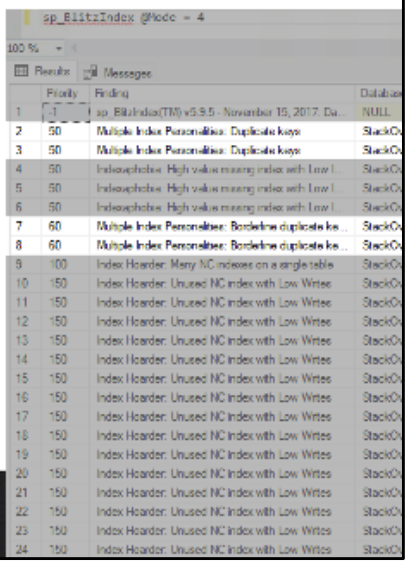


	Priority	Finding	Database
1	-1	sp_BlitzIndex(TM) v5.9.5 - November 15, 2017: Da...	NULL
2	50	Multiple Index Personalities: Duplicate keys	StackO
3	50	Multiple Index Personalities: Duplicate keys	StackO
4	50	Indexaphobia: High value missing index with Low I...	StackO
5	50	Indexaphobia: High value missing index with Low I...	StackO
6	50	Indexaphobia: High value missing index with Low I...	StackO
7	60	Multiple Index Personalities: Borderline duplicate ke...	StackO
8	60	Multiple Index Personalities: Borderline duplicate ke...	StackO
9	100	Index Hoarder: Many NC indexes on a single table	StackO
10	150	Index Hoarder: Unused NC Index with Low Writes	StackO
11	150	Index Hoarder: Unused NC Index with Low Writes	StackO
12	150	Index Hoarder: Unused NC Index with Low Writes	StackO
13	150	Index Hoarder: Unused NC Index with Low Writes	StackO
14	150	Index Hoarder: Unused NC Index with Low Writes	StackO
15	150	Index Hoarder: Unused NC Index with Low Writes	StackO
16	150	Index Hoarder: Unused NC Index with Low Writes	StackO
17	150	Index Hoarder: Unused NC Index with Low Writes	StackO
18	150	Index Hoarder: Unused NC Index with Low Writes	StackO
19	150	Index Hoarder: Unused NC Index with Low Writes	StackO
20	150	Index Hoarder: Unused NC Index with Low Writes	StackO
21	150	Index Hoarder: Unused NC Index with Low Writes	StackO
22	150	Index Hoarder: Unused NC Index with Low Writes	StackO
23	150	Index Hoarder: Unused NC Index with Low Writes	StackO
24	150	Index Hoarder: Unused NC Index with Low Writes	StackO
25	200	Abnormal Psychology: Recently modified tables/Ind...	StackO
26	200	Abnormal Psychology: Recently modified tables/Ind...	StackO

Our scripts try to prioritize stuff.

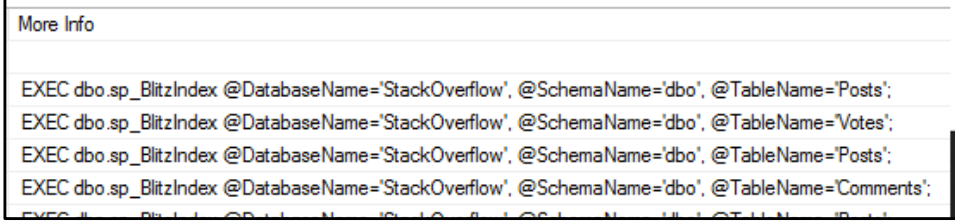
50: Duplicate keys:
These are no-brainers to dedupe.

60: Borderline duplicate:
These have the same leading field, but may have different subsequent fields. Will take a little bit more work.



How I use sp_BlitzIndex to D/E

1. sp_BlitzIndex @GetAllDatabases = 1
   (and figure out what database to tune)
2. Run sp_BlitzIndex in the database I want to tune
   (and figure out what table I want to focus on)
3. Scroll across to the More Info column and run it for the particular table I want to tune



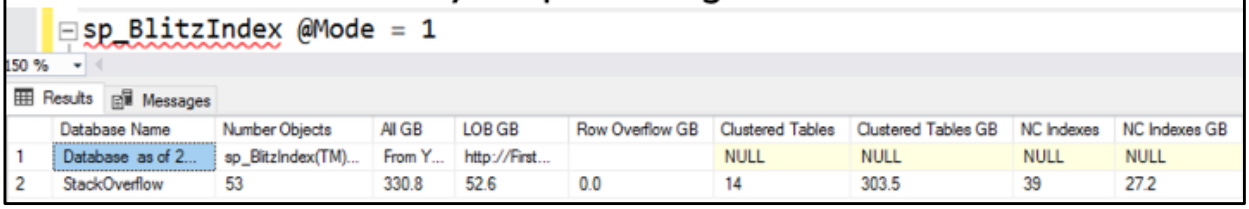
1.3 p32

Advanced sp_BlitzIndex tips

@ThresholdMB: default 250MB, only alerts you for problems with indexes at least this large

@Mode:

- 0 = default, most urgent problems
- 4 = more analysis, includes more warnings
- 2 = inventory of all your indexes & metrics
- 1 = summary of space usage



```
sp_BlitzIndex @Mode = 1
```

150 %

Results | Messages

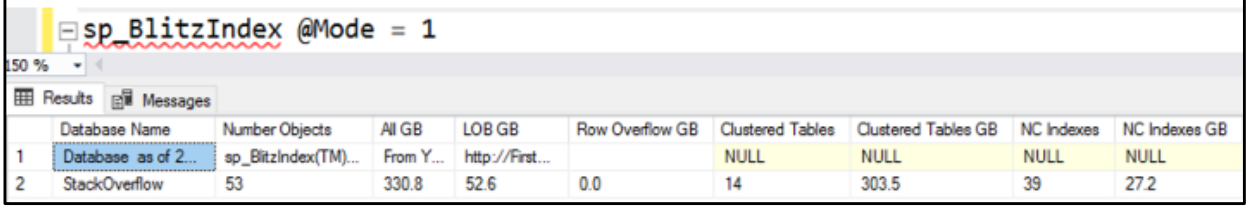
	Database Name	Number Objects	All GB	LOB GB	Row Overflow GB	Clustered Tables	Clustered Tables GB	NC Indexes	NC Indexes GB
1	Database as of 2...	sp_BlitzIndex(TM)...	From Y...	http://First...		NULL	NULL	NULL	NULL
2	StackOverflow	53	330.8	52.6	0.0	14	303.5	39	27.2

Example writeup

"On the Users table, we have:

- 3 duplicate indexes of 15GB total
- 2 unused indexes of 8GB total

By removing these, I saved 23GB of drive space,
made deletes/updates/inserts go faster,
and now I can add more appropriate indexes."



```
sp_BlitzIndex @Mode = 1
```

150 %

Results | Messages

	Database Name	Number Objects	All GB	LOB GB	Row Overflow GB	Clustered Tables	Clustered Tables GB	NC Indexes	NC Indexes GB
1	Database as of 2...	sp_BlitzIndex(TM)...	From Y...	http://First...		NULL	NULL	NULL	NULL
2	StackOverflow	53	330.8	52.6	0.0	14	303.5	39	27.2

What we covered

The two SQL Server index usage views:

- Usage by plan: sys.dm_db_index_usage_stats
- Usage by index: sys.dm_db_index_operational_stats

Why they're not as accurate as you might suspect:

- Seek doesn't mean one row
- Scan doesn't mean the whole table
- Reads doesn't mean the index was actually read
- 1 write doesn't mean 1 row was updated
- They even reset at unusual times
- Analyze with enough uptime to reflect business processes

1.3 p35