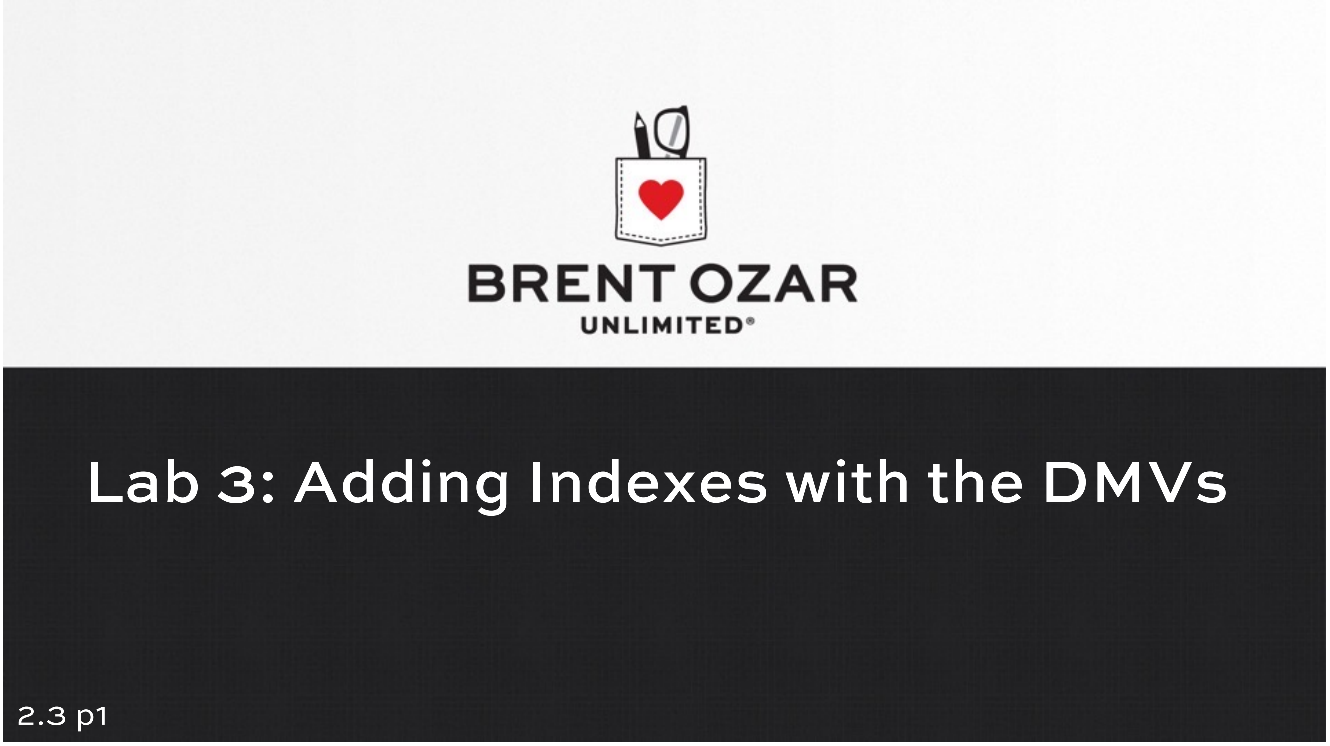
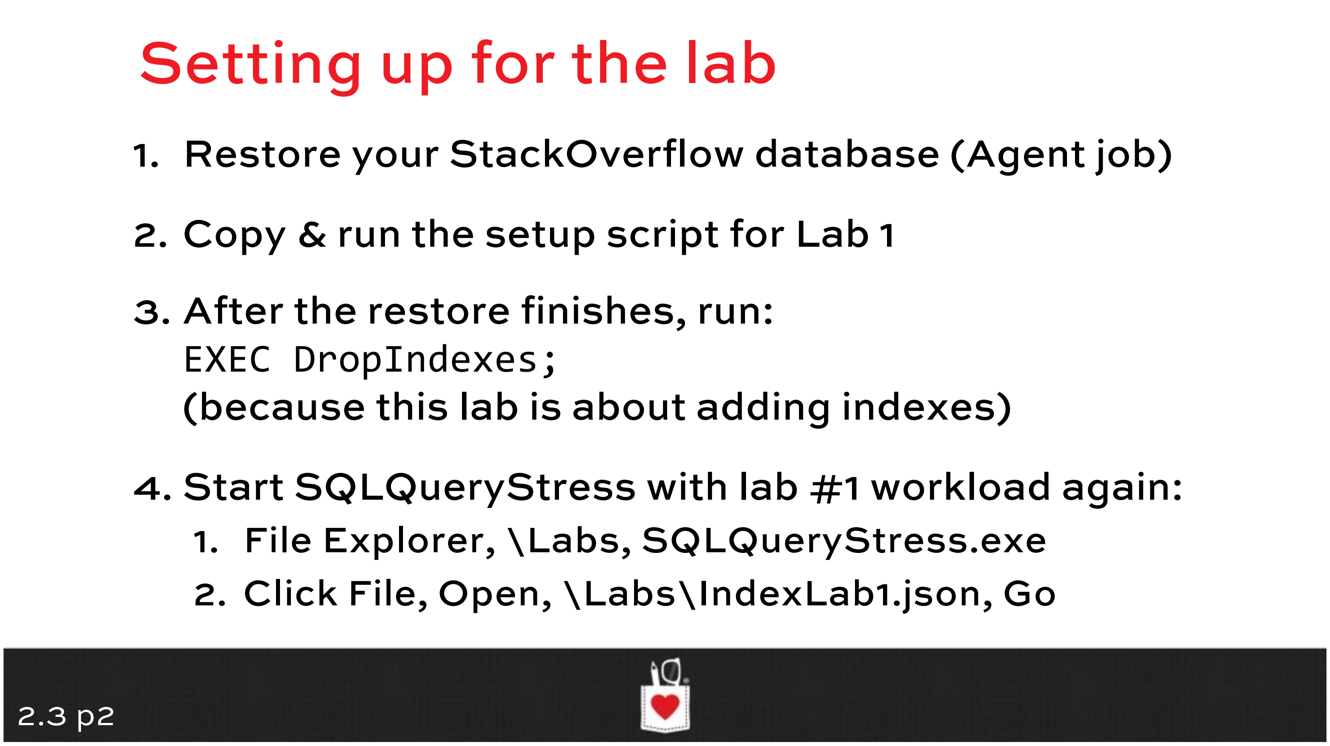


BRENT OZAR UNLIMITED®

Lab 3: Adding Indexes with the DMVs



Setting up for the lab

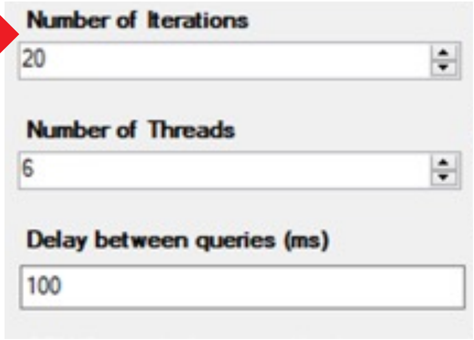
1. Restore your StackOverflow database (Agent job)
2. Copy & run the setup script for Lab 1
3. After the restore finishes, run:
   `EXEC DropIndexes;`
   (because this lab is about adding indexes)
4. Start SQLQueryStress with lab #1 workload again:
   1. File Explorer, \Labs, SQLQueryStress.exe
   2. Click File, Open, \Labs\IndexLab1.json, Go

Doing the lab later?

Set this to 100000.

This makes sure your VM stays busy til you're ready.

For the success test, you'll be using the regular iterations to see if the test completes quickly enough.



Number of Iterations

20

Number of Threads

6

Delay between queries (ms)

100

My index tuning strategy

Just once	**Dedupe** – reduce overlapping indexes **Eliminate** – unused indexes
Weekly for 1 month	**Add** – badly needed missing indexes
Do this only AFTER the easy stuff above	**Tune** – indexes for specific queries **Heaps** – usually need clustered indexes

The D/E/A steps use sp_BlitzIndex.

The T step uses sp_BlitzCache.

Time to rely on our buddy.

The DropIndexes proc dropped all your nonclustered indexes.

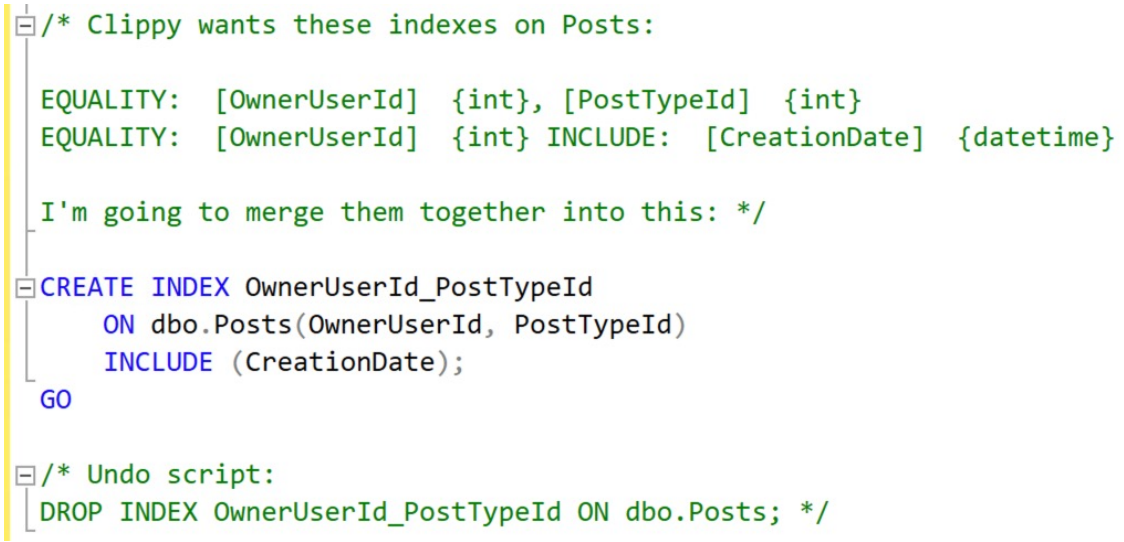
The load test pushed a bunch of queries through, which populated your missing index DMVs.

This time: use the missing index recommendations in sp_BlitzIndex, but think carefully about their contents.

Now, add the right ones.

1. Run sp_BlitzIndex for missing index suggestions.
2. Think about the queries that might trigger each index. (Remember, you've seen them in your last lab.)
3. Script out changes you want to make, aiming for:
   - 5 or less indexes per table
   - 5 or less fields per index
4. Show your work in Slack. (But you don't have to wait for permission to run it – run 'em as you go, because it takes a while to add new indexes.

Lab answer format



```
/* Clippy wants these indexes on Posts:

EQUALITY:  [OwnerUserId]  {int}, [PostTypeId]  {int}
EQUALITY:  [OwnerUserId]  {int} INCLUDE:  [CreationDate]  {datetime}

I'm going to merge them together into this: */

CREATE INDEX OwnerUserId_PostTypeId
    ON dbo.Posts(OwnerUserId, PostTypeId)
    INCLUDE (CreationDate);
GO

/* Undo script:
DROP INDEX OwnerUserId_PostTypeId ON dbo.Posts; */
```

How I’d budget this hour

Run sp_BlitzIndex, and keep that window open. (As you create indexes, the suggestions will vanish.)

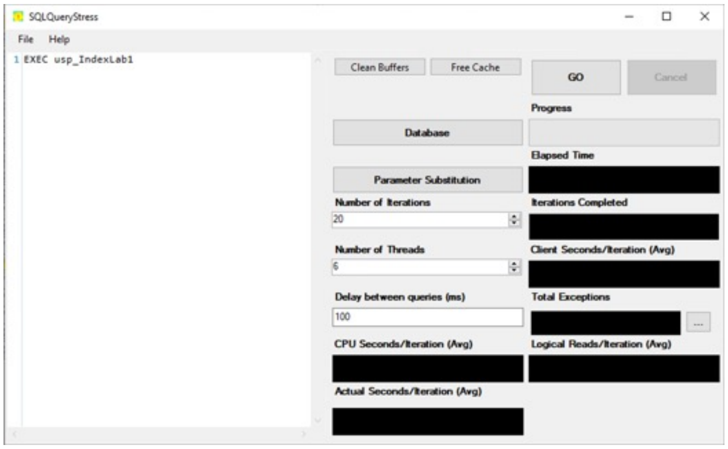
- 15 minutes to work on one table: combine the index recommendations for it, script ‘em, run ‘em.
- Repeat for 2-3 tables. Run ‘em as you go.
- Start the load test again. Let it run for several minutes, and make sure your indexes get used.

Stop your load generator now.

Click Cancel and close SQLQueryStress.

If it won't close, you may need to kill it and its queries.

(Make sure it doesn't show up in Task Manager either.)



```
EXEC usp_IndexLab1
```

Number of Iterations: 20

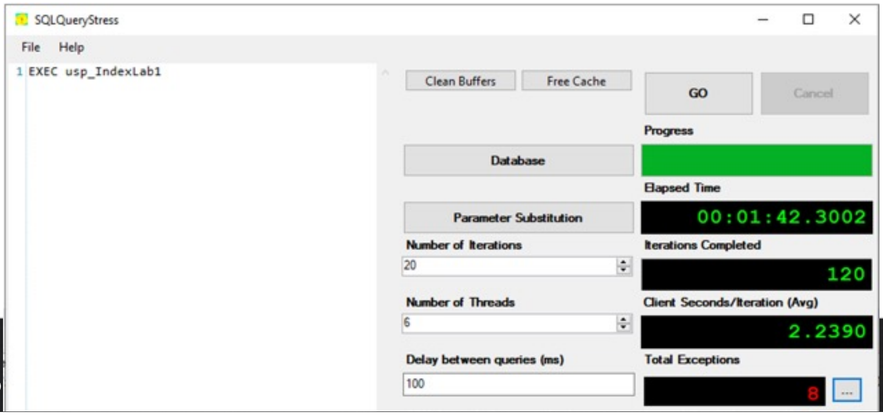
Number of Threads: 6

Delay between queries (ms): 100

Bonus points: rerun the load test.

If you've done a good job of adding indexes, the workload should finish in 1-2 minutes.

Some exceptions are expected, and that's okay.



```
EXEC usp_IndexLab1
```

Elapsed Time: 00:01:42.3002

Number of Iterations: 20

Iterations Completed: 120

Number of Threads: 6

Client Seconds/Iteration (Avg): 2.2390

Delay between queries (ms): 100

Total Exceptions: 8