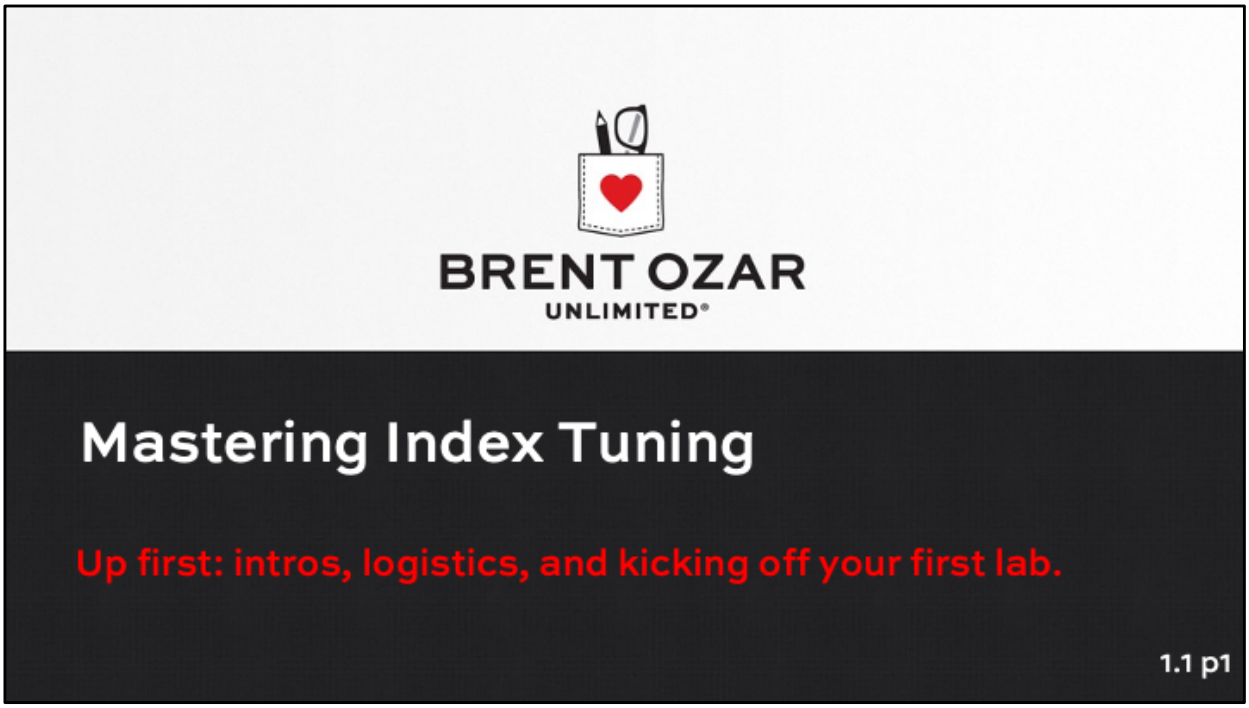


Introduce yourself in Slack:

	Developer	Development DBA	Production DBA
Write C#, Java code	Daily		
Build queries, tables	Daily	Sometimes	
Tune queries	Sometimes	Daily	
Design indexes		Daily	
Monitor performance		Daily	Sometimes
Troubleshoot outages			Daily
Manage backups, jobs			Daily
Install, config SQL			Sometimes
Install, config OS			Sometimes

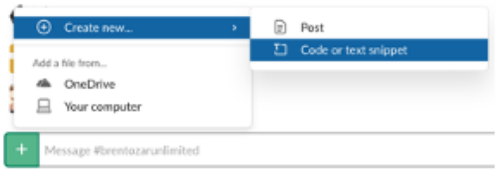
1.1 p2

Slack pro tips

Accidentally close your browser? Want to share screenshots? Lots of pro tips: BrentOzar.com/slack

No direct messages please – use the public room.

To share code or T-SQL, click the + sign next to where you type text in, and choose "code or text



1.1 p3

Slack is a great cheat code.

Get stumped on a lab?

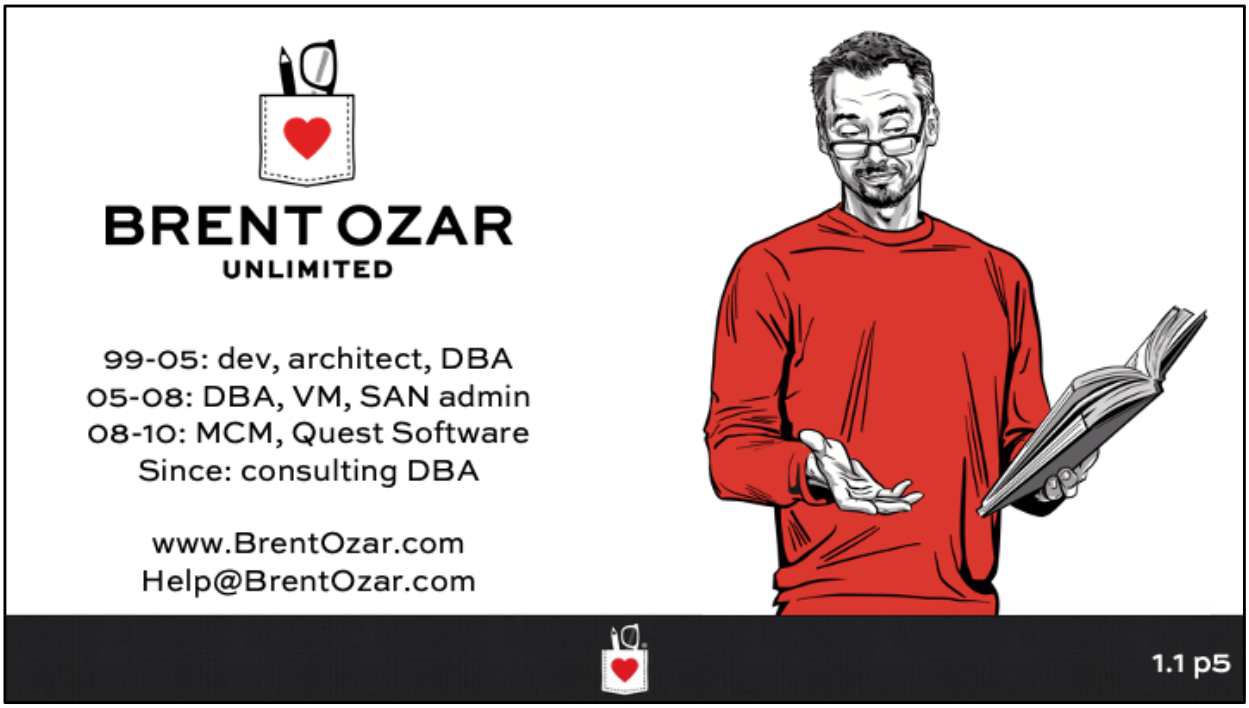
Wondering how other students solved it?

Wondering how *quickly* other students work?

Take a peek in the Slack room.

Otherwise, don't look (spoilers.)

1.1 p4



My job: 2-day SQL Critical Care®

Day 1, morning: rapidly assess a single SQL Server, database indexes, queries running against it, team

Day 1 afternoon & day 2 morning: write findings

Day 2 afternoon: deliver findings & training to get the team out of the emergency, quickly

This week: sharing my index techniques, experiences

1.1 p6

I'll show you with lectures + labs.

This isn't about theory: it's about practice.

We'll step through this cycle repeatedly:

1. Lecture: you watch me for 1-2 hours
2. Hands-on lab: you spend 1 hour working on a lab about the concepts you just saw
3. I do the lab: you watch me spend 30-45 minutes on that same lab so you can check your work

1.1 p7

Instant Replay & lab scripts

For a year from your date of purchase, you can:

- Watch the videos (I'm recording this class)
- Download the scripts and database
- Re-run the labs on your own home machines

Problems? Comment on the module.

1.1 p8

This morning's cycle

1. Lectures & demos:
    1. My D.E.A.T.H. method for index tuning
    2. Do the D.E. parts on paper
    3. Understand the problems with the index usage DMVs
    4. See how sp_BlitzIndex displays those DMVs
2. Lab & lunch:
   you clean up indexes on an over-indexed server
3. Right after lunch: I do the same lab you just did

Then the cycle starts again.

Continuing on this week

Adding indexes by interpreting the DMVs carefully

Tuning indexes for specific scenarios like:

- Blocking
- Small filters of a large table
- Reporting aggregates
- Non-sargable predicates
- Foreign keys & check constraints

1.1 p10

Live streaming tips & tricks

Everyone's stream is about 15 seconds behind

Your browser's video player may let you:

- Pause (but you'll get behind)
- Change your resolution
- Push to a TV

If your stream falls 20-30 seconds behind,
or you lose connection, refresh your browser

1.1 p11

About our lab environment

1.1 p12

VMs I use

AWS i3.xlarge:

- 4 cores so queries can go parallel
- ~30GB RAM so queries get workspace memory
- All solid state storage to create & test indexes fast

Installed & updated: SQL Server 2019,
SSMS 18, First Responder Kit scripts

Your lab VM will stay up during the class.
Do not shut it down until you're completely done:
shutting it down erases it permanently.

Rules we're breaking

"Never remote desktop into the SQL Server."

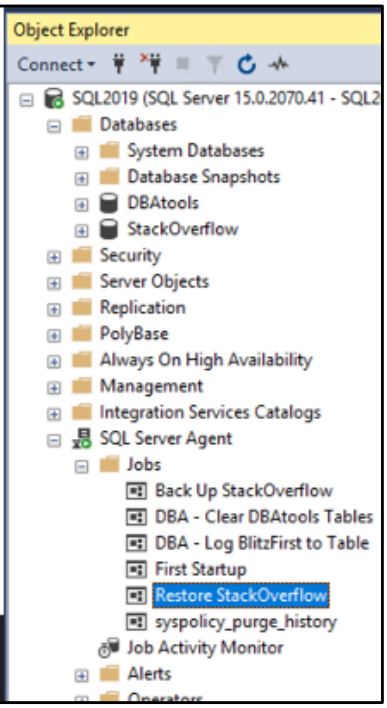
"Never run SSMS on the SQL Server itself."

"Don't run applications on the SQL Server."

None of those are the root cause for the performance issues you'll be seeing.

When you hit a problem

1. Close SQLQueryStress (end-task it if you have to)
2. Use the Agent jobs to restore your database if you have to (15min.)
3. Rerun the setup process for lab we're on (may require running a script, SQLQueryStress, etc)



In this class, I don't put any land mines in the *server setup.*

SQL Server default setup, plus a few good tweaks:

- Cost Threshold for Parallelism = 50
- MAXDOP = 4
- StackOverflow in 2017 compat mode
- Databases in simple recovery model

You can change these if you like, but
server changes aren't the point of this class.

1.1 p16

I put a lot of land mines in the *T-SQL code and indexes.*

You can't really fix these by throwing hardware at it.

In this class, solve problems by tuning indexes.

(There will be some labs where you'll have to change code to get the indexes to work, but start with index tuning alone.)

The database: StackOverflow.com

Q&A site: you ask, other people do your job

Database is available under Creative Commons

Download it via BitTorrent:
BrentOzar.com/go/querystack

1.1 p18

Important tables

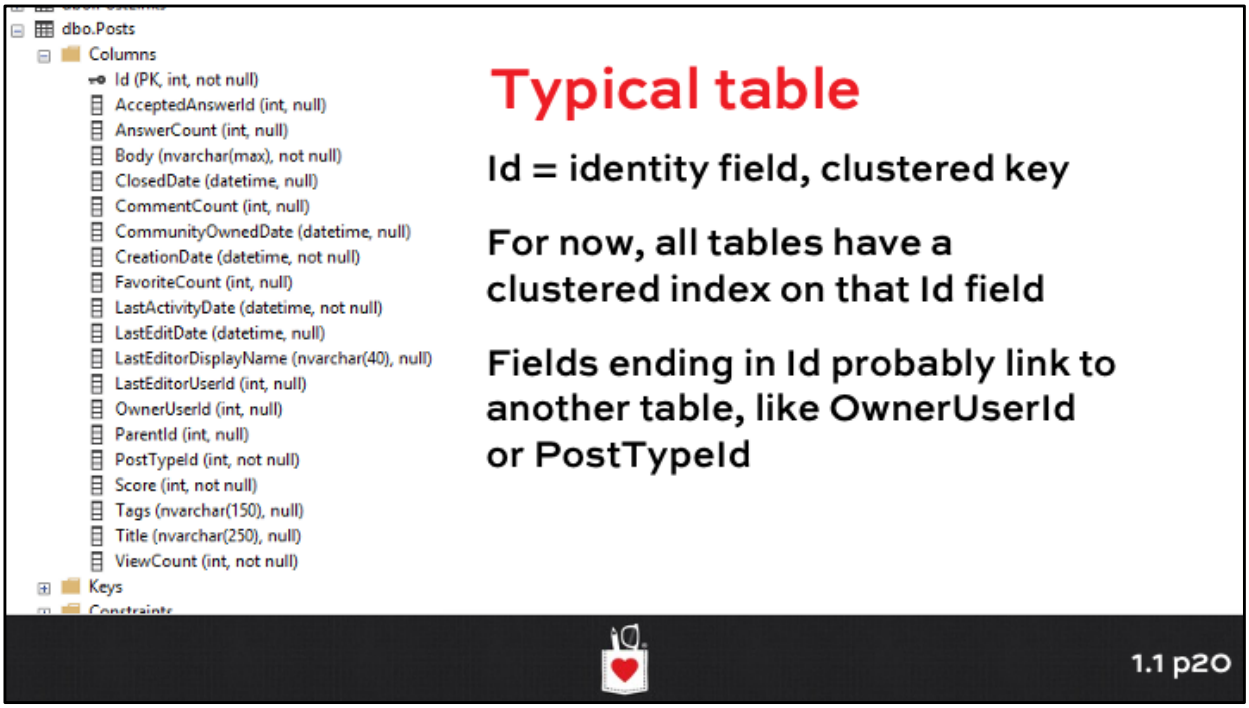
dbo.Users – from How to Think Like the Engine

dbo.Posts – all questions & answers

- PostTypeId = 1 means question
- PostTypeId = 2 means answer
- OwnerUserId = links to Users.Id

dbo.Comments

- PostId = parent (can be question or answer)
- UserId = Users.Id who left the comment



I've added a lot of indexes.

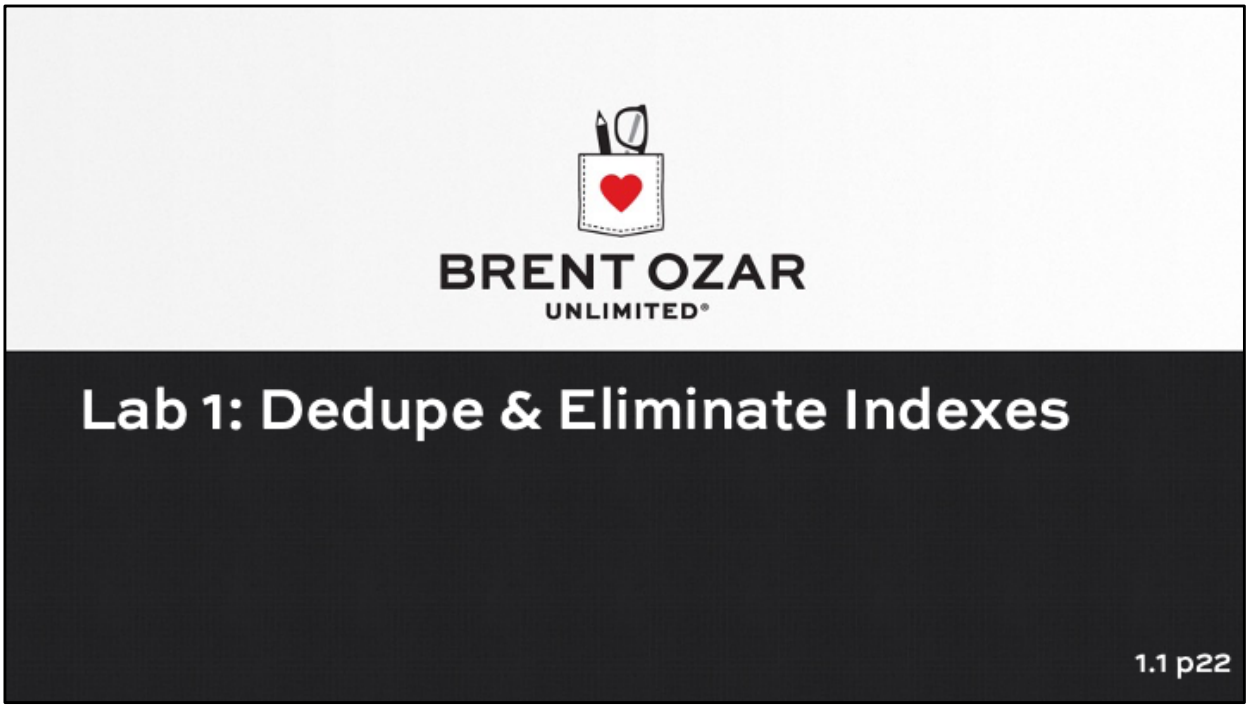
Just like in the real world, you've inherited this database from "the last person."

Your first lab is going to involve figuring them out.

To do that, we need to have a running workload.

Let's go set up your first lab!

1.1 p21



Download the latest stress tests

I changed the SQLQueryStress json files in January:

https://BrentOzar.com/go/getstressed

We need a running workload.

In your midday lab, you'll be analyzing DMVs.

To do that, we need real workloads doing selects, inserts, updates, and deletes.

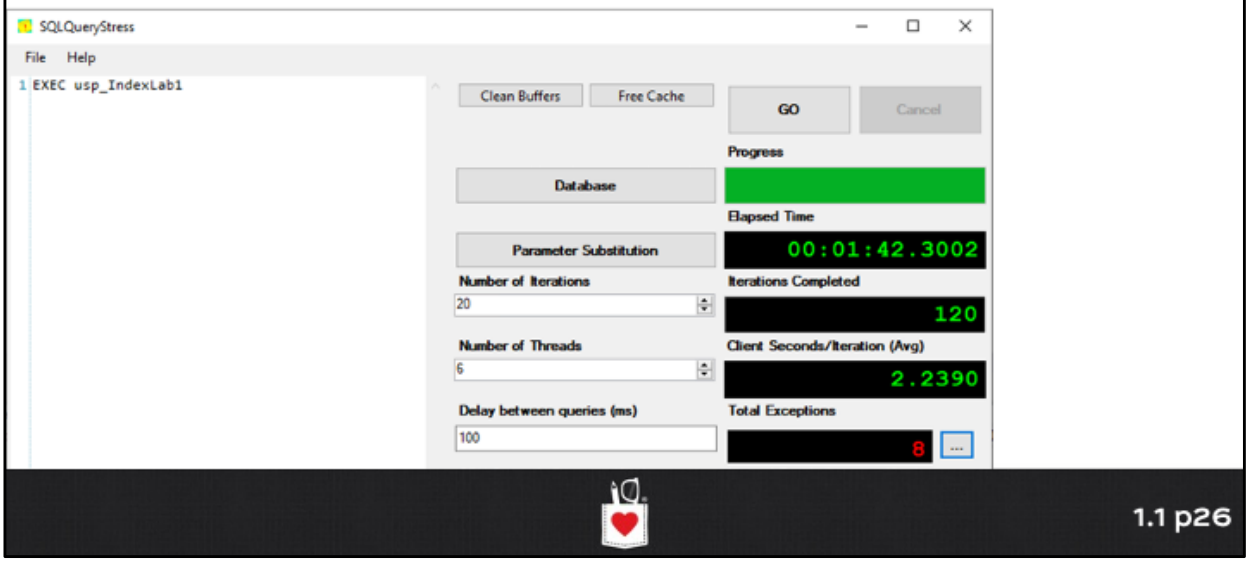
(This is also why tools like sp_BlitzIndex don't work very well in development & offline environments: none of the indexes are really getting used.)

1.1 p24

Labs 1, 2, & 3 use the same SQL:

1. If you've been playing around adding/dropping indexes, restore your StackOverflow database
2. Copy & run the setup script for Lab 1
3. Start SQLQueryStress:
    1. File Explorer, \Labs, double-click SQLQueryStress.exe
    2. Click File, Open, \Labs\IndexLab1.json
    3. Click Go

What SQLQueryStress is



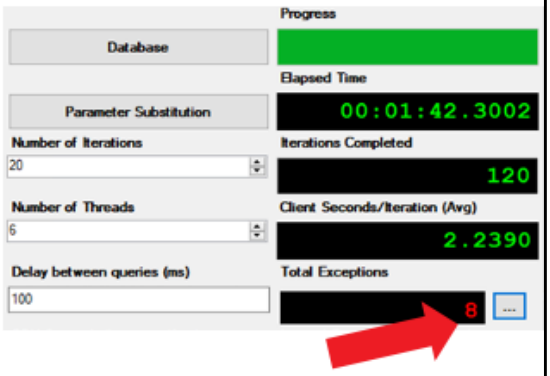
1.1 p26

A few errors are okay.
100% error rate is not.

OK errors:

- Divide by zero
- Transaction count
- Deadlock

But if you get errors about procs not being found or syntax errors, stop and ask for help, include screenshots.



1.1 p27

You can stop it at any time.

You don't need the queries running live as you work for labs 1-3. (Lab 4 will be different.)

Click Cancel on SQLQueryStress whenever you want: you don't have to wait for it to finish the lab.

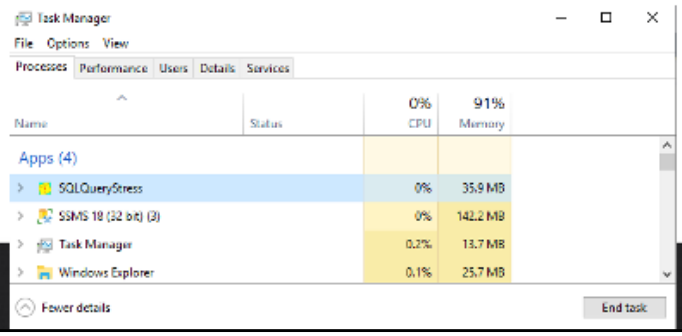
Just let it run for at least 5-10 minutes to build up data in your index usage DMVs.

1.1 p28

Sometimes, it goes rogue.

Sometimes you'll cancel it and close it, but it will keep running in the background, headless.

Go into Task Manager to make sure it's not there.
If it is, end-task it a couple of times to kill it.



1.1 p29

Set yours up and take a bio break.

Also, if you're using one of my lab VMs,
email me at Help@BrentOzar.com.

If you don't email me, we'll turn off your VM today
because we'll assume you didn't show up for class.

1.1 p30