Setting up for lab 3

1. Restart your SQL Server service (clears all stats)
2. Restore your StackOverflow database (Agent job)
3. Copy & run the setup script for Lab 3
4. Start SQLQueryStress with QueryLab3.json
   OR tune mqt_QueryLab3_Level1

2.3 p2

The last few topics we covered:

Changing the number of rows we return: pagination

Changing the order in which SQL Server processes: CTEs, temp tables, and index hints

Changing where business logic lives: rewriting/inlining user-defined functions

We're starting to make big changes.

2.3 p3

Your job as a T-SQL tuner

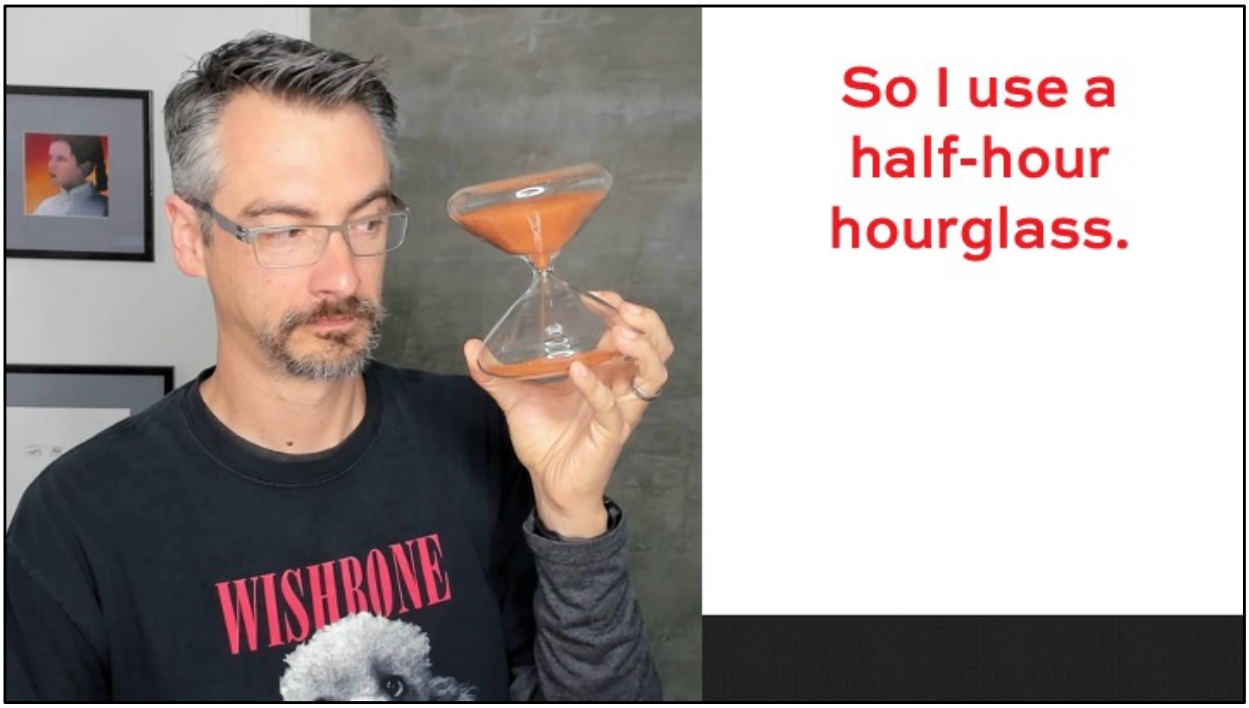
- Read the query to understand what it's aiming for
- Analyze the query and the plan to understand where SQL Server is going wrong
- Come up with a better way to write the query
- Actually write it
- Test it for faster execution

2.3 p4

And in reality, you also have to:

- Test the results for accuracy
- Find more sets of parameters to test those too
- Document your work
- Convince your team to adopt it
- Check it into source control
- Get it into production

2.3 p5



And in that half-hour, I try to:

- Read the query to understand what it's aiming for
- Analyze the query and the plan to understand where SQL Server is going wrong
- Come up with a better way to write the query
- Actually write it
- Test it for faster execution

And often, I can't do all that in 30 minutes.

2.3 p7

How to do your work faster

Use the production database contents
(same data, same distributions, same sizes)

Use the production server
(same version, config, hardware – but of course this comes with huge risks)

And today, we're doing the above two to help you.

Use a server *absolutely identical to* production
(and usually they say it is, but it ain't)

2.3 p8

How to do your work slower

Work in a database nothing like production:
different data sizes, distribution, contents

Work on a server nothing like production:
different hardware, config, versions

This is why you usually
can't make progress quickly.

2.3 p9

Timeline for this lab

5 minutes – sp_BlitzCache & query review: run sp_BlitzCache, pick a query where you can make a difference.

25-45 minutes – tune 1 query: review the plan, change query/indexes/stats, get the new actual plan. Be ruthless: if you need to change the result sets, do it. Report your before/after query plans in Slack with PasteThePlan.

If you have time: do 1 more query.

2.3 p10

Now, everything's changeable.

In prior labs, we haven't changed the result set that users get back, or a proc's surface area (parameters).

In this lab, you can, because you may need to add params or change results to do things like pagination.

So now, you can change anything.

You can even try SQL Server 2019 compat mode.

2.3 p11

What you can't do:

No indexes on columns over 200 bytes:

- ~~VARCHAR(250)~~
- ~~NVARCHAR(MAX)~~

And no, you can't use them as includes or indexed views either.

2.3 p12

Stumped? Use the clues.

Each stored procedure in the load test has an identical version with _Clue at the end, like:

- usp_GetQuestionsByDate
- usp_GetQuestionsByDate_Clue

Just spend at least 10-15 minutes working on the original proc before you open its clue.

2.3 p13