

Troubleshooting Your First Low-performing T-SQL Query

Asking Questions About a Query



Jared Westover

Data Specialist

@westoverjared sqlhabits.com



Introduction



Define a low-performing query

- Execution speed
- Better hardware

Expectations of developers

- Data landscape

Ask questions

- Checklist

Break queries down

- Smaller steps





What is low-performing?

- More than speed

Slow to return results

Difficult to understand

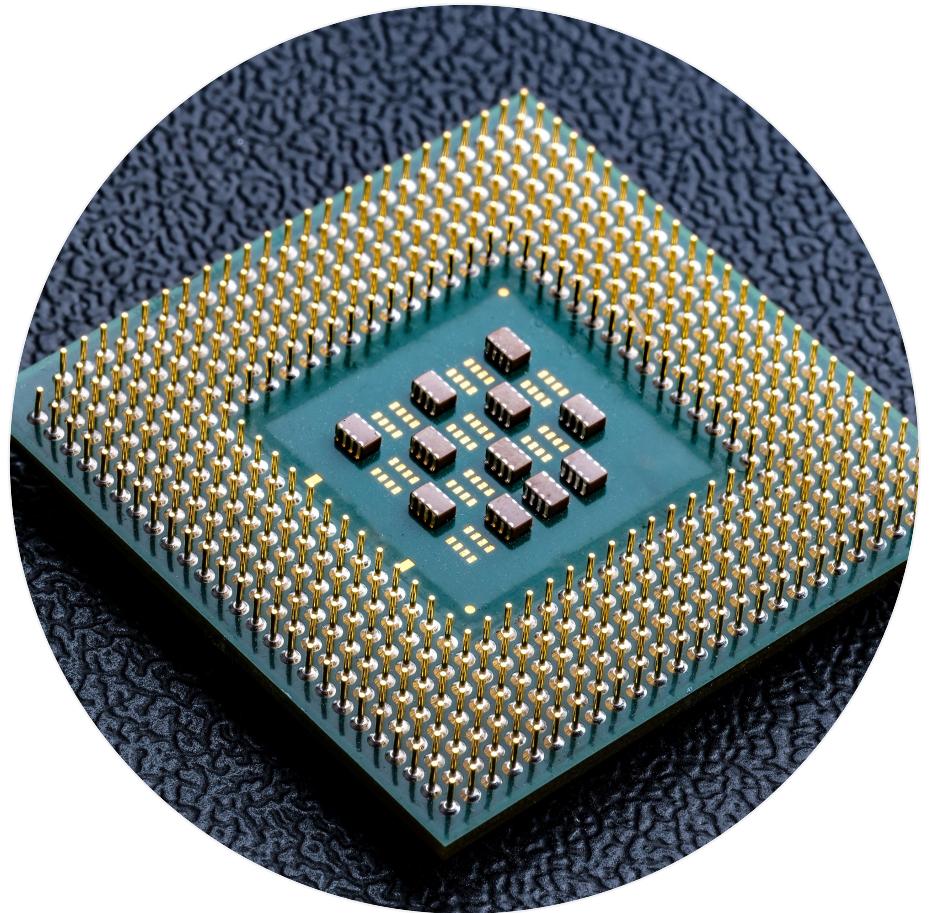
- Nested subqueries

Fails periodically

Doesn't accomplish its mission

- Mistaken updates





Hardware as a Solution

Throw hardware at the problem

Improved hardware may not help

Costly and hard to reverse



Expectations of Application Owners



What Do Application Owners Expect?



T-SQL knowledge
Return and manipulate data



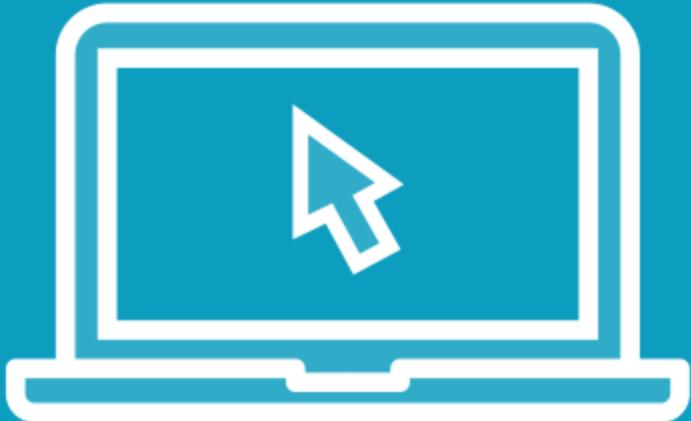
Data landscape
Knowledge of the data environment



Tune queries
Boost their performance



Demo



Create your demo environment

- Enable line numbers



Getting Started with Queries



“The only true wisdom is in
knowing you know nothing.”

Socrates



Start with Questions



What's the query used for?

Did the query perform as expected before?

What's changed since the slow down?

- Application release

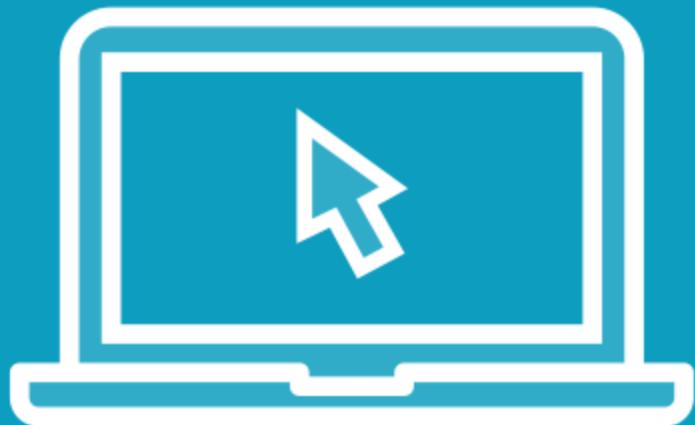
Are you expecting all this data?

Make a checklist of questions

- Closer to resolution



Demo



Create a checklist of questions



Breaking Queries Down



“When eating an elephant,
take one bite at a time.”

Creighton Abrams



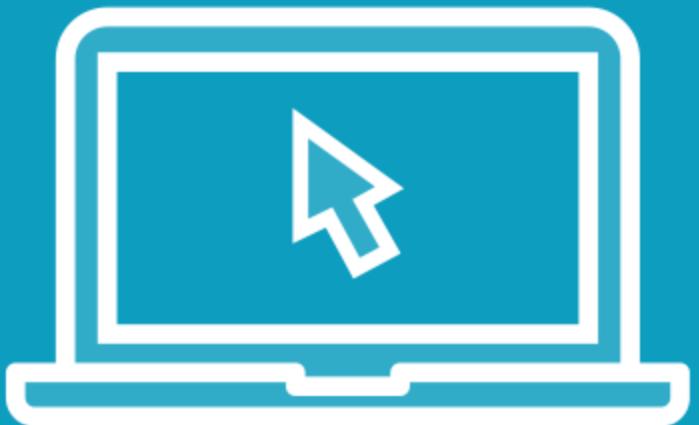
Breaking Down Complexity



- What are some of the complex parts?**
- Convert derived tables to CTEs**
- Pull apart complex date functions**
 - DATEADD & DATEDIFF
- Multiple self joins**
- Nested CASE expressions**



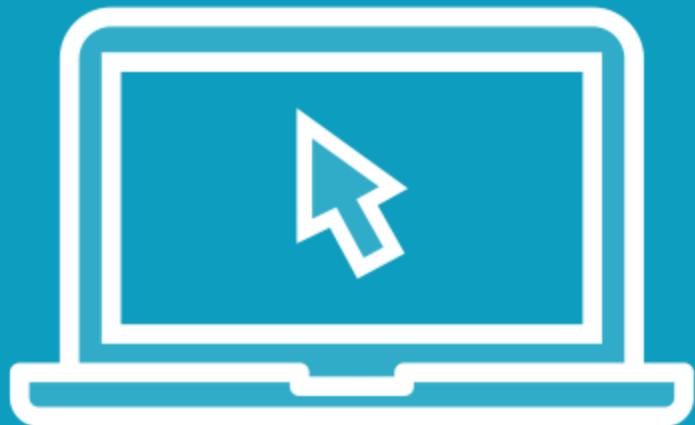
Demo



Using common table expressions



Demo



Confusing CASE expressions



Summary



Features of a low-performing query

- Doesn't behave as expected

Developer expectations

Start with questions

- Checklist

Break queries down

- Develop a habit



Up Next:
Sources of Query Performance Issues

