



**BRENT OZAR**  
UNLIMITED®

# Fundamentals of Query Tuning

Recap of what we learned and what to do next.

Module 7 Slide 1

1

## Query execution has 2 phases

1. Building the plan
  - Based off the parameters we start with
  - Compiled for the whole batch, all at once
  - Plan quality can vary based on time available
  - Your window: compilation CPU, time, timeouts
2. Executing the plan
  - Based off the parameters in the cached plan
  - Not revisited when problems happen



Module 7 Slide 2

2

## Don't rely on duration or costs.

Instead, focus on somewhat-more repeatable metrics:

Common:

- Logical Reads: SET STATISTICS IO ON
- CPU time: SET STATISTICS TIME ON

Less common:

- TempDB spills
- Memory grants



Module 7 Slide 3

3

## The most important gauge

Compare estimated rows to actual rows

Start at the top right and work across/down

Find the spot where the variance is >10X

To fix it, you can:

- Change the T-SQL to be more easily understood
- Break a large query into parts, use temp tables
- Recompile, dynamic SQL, child stored procs



Module 7 Slide 4

4

## Parameters change everything

	No parameters	Has parameters
Always slow	Easy to tune	Kinda easy
Sometimes slow, sometimes fast	Probably hardware, blocking, changing data sizes	Very hard to tune

To find the right parameters to tune, you'll often have to resort to querying the data to find outliers.

Module 7 Slide 5



5

## Next steps on your learning path:

Optional, any of these in any order:

- Fundamentals of Parameter Sniffing
- Fundamentals of Columnstore
- Fundamentals of TempDB

Or, you can jump to the Mastering track:

- Mastering Index Tuning
- Mastering Query Tuning
- Mastering Server Tuning

Module 7 Slide 6



6

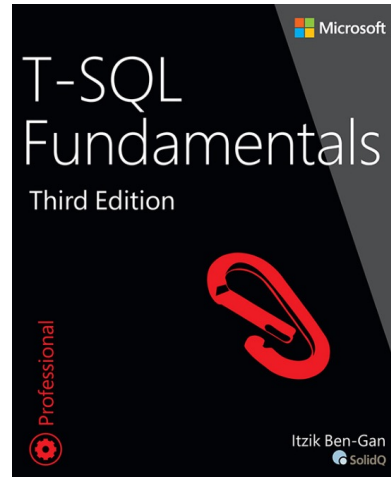
## T-SQL book, level 1

T-SQL Fundamentals by Itzik Ben-Gan

Queries, joins, data modification, transactions, concurrency, stored procs, functions

Much harder than it looks: 458 pages that will last you a year

<https://amzn.to/2Y2O7vd>



Module 7 Slide 7

7

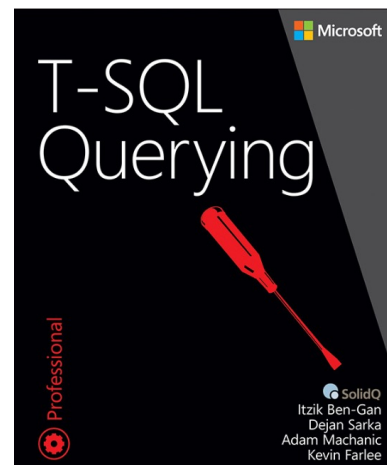
## T-SQL book, level 2

T-SQL Querying by Itzik Ben-Gan

Goes deeper: logical query processing, tuning, pivoting, windowing, offset/fetch, dates/times, T-SQL for BI

861 pages of advanced stuff

<https://amzn.to/2Y3ndDE>



Module 7 Slide 8

8

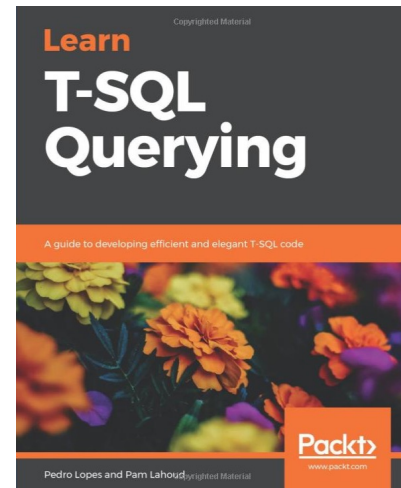
## Tuning book, level 2

Learn T-SQL Querying

Terrible name for a good book: it's really about performance tuning

Covers T-SQL anti-patterns, DMVs, tracing with XE, analyzing query plans, Query Store

<https://amzn.to/2SuBOGJ>



Module 7 Slide 9



9

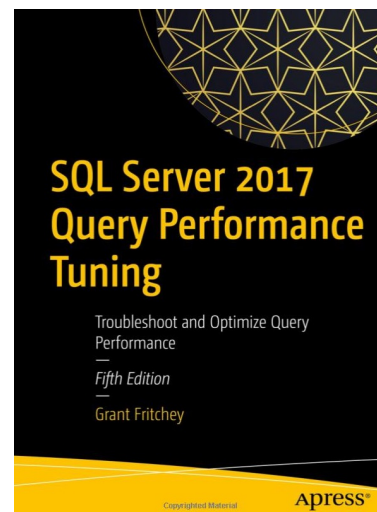
## Tuning book, level 2

SQL Server 2017 Query Performance Tuning by Grant Fritchey

Yes, this is also level 2: they're both similar levels, but this one goes into more depth on queries & indexes.

700+ pages, kept up to date

<https://amzn.to/2JQQP20>



Module 7 Slide 10



10

## You can do this.

For questions, leave comments on the relevant module.

For private help after the class, email [Help@BrentOzar.com](mailto:Help@BrentOzar.com) with:

- A note that you were in this class
- sp\_Blitz @CheckServerInfo = 1
- sp\_BlitzFirst @SinceStartup= 1

Module 7 Slide 11



11

# Next up, in Mastering...

Module 7 Slide 12



12

## Query tuning options

Index tuning

Query hints

Query Store hints

Query rewrites, temp tables, CTEs

Dynamic SQL

Adding columnstore indexes, indexed views

Creating reporting tables

Module 7 Slide 13



13

## Index tuning makes a lot of queries a lot faster.

Just getting started tuning an app or a new table?

No one's actively tuning indexes on it?

Tables have less than 5 indexes on 'em?

Plan has a missing index recommendation?

Index tuning is probably the answer.

Module 7 Slide 14



14

## Query hints

`OPTION (RECOMPILE)`

`OPTION (OPTIMIZE FOR (@Location = 'India'))`

`OPTION (MAXDOP 3)`

`OPTION (MIN_GRANT_PERCENT = 10)`

`OPTION (USE HINT ('hint_name'))`

Many, many more, like cardinality levels:

<https://docs.microsoft.com/en-us/sql/t-sql/queries/hints-transact-sql-query>

Module 7 Slide 15



15

## When using hints

Ideally, hint the data distribution, not the plan shape

Try to be as subtle as possible

You're building in technical debt:

if data distribution changes, you'll have to revisit hints

Leave SQL Server's options open for future versions

Module 7 Slide 16



16



## Query Store hints

A hint will make the query perform better

You can test the hint with lots of parameter variations

But you can't actually change the query

And you CAN find the query in Query Store

Module 7 Slide 17



17

## How Query Store hints work

Find the query in Query Store, get its ID

Run this to add the hint(s) you want:

```
EXEC sys.sp_query_store_set_hints @query_id= 39,  
@query_hints = N'OPTION(RECOMPILE)';
```

More info, video tutorial:

<https://learn.microsoft.com/en-us/sql/relational-databases/performance/query-store-hints-best-practices>

Module 7 Slide 18



18

## Query rewriting, temp tables, CTEs

Read the plan right to left, top to bottom.

Look for where estimates vs actuals are off by 10X.

After that, everything in the plan is doomed.

You've gotta find out why those estimates are wrong.

If you can fix it (by simplifying or tuning T-SQL), do it.

If you can't, break the query up into phases.

Module 7 Slide 19



19

## Dynamic SQL

Classic use case:

somebody wrote one query that's supposed to serve a lot of different purposes.

Modern use case:

allowing Intelligent Query Processing + PSPO to cache a LOT of different query plans for just 1 query, using comment injection.

Module 7 Slide 20



20

## Columnstore & indexed views

Reporting queries with a lot of aggregate functions like SUM, MAX, AVG? **Columnstore is for you.**

Reporting queries that crunch a lot of data, and go faster in batch mode? **Columnstore is for you.**

Queries with non-sargable functions, and you can't change the queries? **Indexed views.**

Queries where the toughest problem is the joins, and you run them constantly? **Indexed views.**

Module 7 Slide 21



21

## Reporting tables

Sometimes it's not a query issue.

Sometimes we're asking for reports.

Sometimes we want those reports to run quickly.

Sometimes we're okay with stale data  
(like reports that run on yesterday's sales.)

It's time to create reporting tables.

Module 7 Slide 22



22

## That's a last resort for me.

It requires changing the queries, AND the tables.

Let's be honest: users usually want real time data.

But I'd be a bad consultant if I didn't mention that sometimes the right answer is a report table process.



Module 7 Slide 23

23

## Pick the first one that gets you across the finish line, fastest.

Index tuning

Query hints

Query Store hints

Query rewrites, temp tables, CTEs

Dynamic SQL

Adding columnstore indexes, indexed views

Creating reporting tables



Module 7 Slide 24

24



**Thanks, and I hope you  
had a great time!**



Module 7 Slide 25

25