# SQL Server: Optimizing Stored Procedure Performance – Part 1

Introduction

Kimberly L. Tripp
SQLskills.com
@KimberlyLTripp

pluralsight
hardcore dev and IT training

# Background

- **My first two Pluralsight courses were *SQL Server: Why Physical Database Design Matters* and *SQL Server: Optimizing Ad Hoc Statement Performance***
- **This course builds on the base concepts that:**
  - Performance tuning is not just about writing good code
  - Performance tuning is not just about knowing your data
  - Performance tuning is not just about knowing your workload / priorities
- **Performance tuning takes multiple things:**
  - Knowing your data
  - Knowing your workload
  - Knowing how SQL Server works

# This Course

- **There are lots of different ways to write stored procedures**
- **The default options for caching and compilation are NOT always best for all stored procedures**
- **The course will show you that having a precompiled (and therefore an already-optimized plan) is not always good**
  - However, recompiling isn't always the best choice either
- **A point that I make in all of my courses:**
  - Truly scalable applications don't happen by accident
  - There is no platform (or default) that's perfect for every possible use case
  - There is no single method for writing stored procedures that works well ALL THE TIME and requires minimal-to-no-work

# What Does Optimizing Stored Procedure Performance Mean?

- **There are different options for storing and dealing with plans for stored procedures in SQL Server**
- **Each of these ways has pros and cons around:**
  - Cost
    - Do we have to compile a plan or is one already available?
    - How did SQL Server arrive at that plan? Was it a good one? Is it good for all subsequent executions?
      - This is all tied to statistics and data estimates
      - Can we "sniff" the value?
      - Or, is the value completely "unknown" during optimization?
  - Complexity
    - Different methods sometimes require additional strategies to reduce exposure (e.g. SQL injection)
    - Different methods sometimes require explicit type conversion to reduce plan cache pollution

# Does This Sound Familiar?

- **All of my slides (so far) are almost identical to the opening slides in the *SQL Server: Optimizing Ad Hoc Statement Performance* course**
- **Have you watched that course?**
  - It's an absolute requirement before watching this course!
  - It might not seem like it would be, but the ideas behind estimations, selectivity, and "sniffing" parameters are EXACTLY the same with stored procedures
    - However, there are some different ways of dealing with them
  - The more you understand plan cache, optimization, selectivity, estimates (and heuristics), the easier you'll be able to understand all of this course
  - Many of the statement execution methods described in that course are the basis for the solutions to some of your ugliest stored procedures
- **Please, stop now… watch that course first!** ☺

# What Does Optimizing Stored Procedure Performance NOT Mean?

- **This is not a course on how to create a stored procedure**
- **This is not a course on how to write / re-write the Transact-SQL of a stored procedure to get a faster result**
  - However, sometimes that does help
- **This is not a course on indexing**
  - However, that sometimes helps A LOT
  - We will look at execution plans and some of them will use indexes
  - We will see a couple of cases where indexes change the way a statement is handled by SQL Server
  - We will *not* be diving deeply into index creation techniques and strategies
- **This is not a course on statistics**
  - However, those can also help A LOT

# Why This Course Is Relevant

- **There are some very common scenarios that are not well understood**
- **I've watched arguments on web sites, blog posts, in-person…often based on assumptions and misunderstandings about what SQL Server is doing (some of the most heated are around using [or, NEVER EVER, EVER USING] stored procedures ☺)**
- **Microsoft SQL Server is a multi-purpose relational database engine**
  - It can do anything
  - It can house any data
  - But, that doesn't mean that the defaults for EVERYTHING are good for EVERYONE and EVERY database

# Course Focus and Structure (1)

- **This course expects basic knowledge of database terminology**
  - e.g. database, transaction log, backup
- **This course applies to all Editions of SQL Server 2005 onward and includes SQL Server 2014 with version-specific notes and exceptions**
- **This course talks about the different ways that statements can be executed within stored procedures and how they affect caching, reuse, complexity, and ultimately performance**
  - This course is not a general course about writing Transact-SQL constructs or stored procedures
  - This course is not about extended stored procedures or SQLCLR assemblies
  - But, these techniques apply to ALL Transact-SQL stored procedures (it's critical for your best overall performance and understanding of SQL Server)

# Course Focus and Structure (2)

- **Module 2: Why use stored procedures?**
- **Module 3: Creation, compilation, and invalidation**
- **Module 4: Optimization and recompilation**
- **Module 5: Other concerns and considerations**
- **Consider watching all components of this course, in order**
  - It might not seem like some of the components are necessary but almost every module has cross references
- **Please review the *SQL Server: Optimizing Ad Hoc Statement Performance* course to cover topics such as estimates, cost-based optimization, and statistics**