# SQL Server: Deadlock Analysis and Prevention

## Module 1: Introduction

Jonathan M. Kehayias
Jonathan@SQLskills.com

pluralsight
hardcore developer training

# Introduction

- **This course applies to all versions from SQL Server 2005 onwards**
  - Version-specific features will be pointed out
- **Understanding how to collect and analyze diagnostic data when deadlocks occur is critical for being able to identify and fix the root cause**
- **Deadlocks don't have to be problematic in SQL Server applications**
  - Most deadlocks can be prevented with appropriate design and indexing
  - Proper error handling will prevent deadlocks from negatively affecting end-user experiences
- **There are a number of misconceptions and a lot of bad advice about deadlocks that need to be corrected**

# Misconceptions

- Deadlocks are a bug in SQL Server
- Deadlocks cannot be prevented
- Using NOLOCK on all SELECT statements is the best way to prevent deadlocks from occurring
- Adding covering indexes for every type of query will prevent deadlocks from occurring
- Troubleshooting deadlocks is a complex task that requires an experienced SQL Server developer or administrator

# Course Structure

- This course is designed to make anyone proficient at troubleshooting deadlocks in SQL Server

- Module 2: Locking Overview
- Module 3: Deadlock Detection
- Module 4: Collecting Deadlock Information
- Module 5: Analyzing Deadlock Graphs
- Module 6: Example Deadlock Scenarios
- Module 7: Handling Deadlocks