

SQL Server: Temporary Objects

Module 1: Introduction

Joe Sack

Joe@SQLskills.com



pluralsight 
hardcore developer training

Course Introduction

- SQL Server's version of Transact-SQL provides the ability to create and use temporary objects
- In addition to meeting various data-tier requirements, temporary objects can also be beneficial from a performance and scalability perspective
- This demo-centric course will cover the commonly used temporary objects available in SQL Server, and also include guidelines around appropriate usage and planning considerations

Why use Temporary Objects?

- **SQL Server provides a few temporary object options**
 - Temporary tables (local and global)
 - Temporary stored procedures (not commonly used, in my experience)
 - Table variables

- **Some reasons for using temporary objects**
 - No requirement for persistent data storage
 - Intermediate state needed only temporarily
 - Use of temporary storage only visible to the current session
 - Reduce query plan complexity
 - Portability of lookup-data by the containing modules

Why a Course on Temporary Objects?

- **Your choices can significantly impact performance and scalability**
 - Query plan quality
 - Workload degradation due to unnecessary usage
 - Tempdb overhead
- **Making heavy use of demos, we'll review the mechanics of temporary tables and table variables**
 - Syntax
 - Capabilities
 - Usage scenarios
- **The demos in this course will use SQL Server 2012, however the information provided in this course also applies to earlier versions**
 - SQL Server 2014 will have some variation from what is discussed in this course, for example, inline specification of non-unique clustered and nonclustered indexes for table variables

Course Structure

- **Module 2: Temporary Tables**
- **Module 3: Table Variables**
- **Module 4: Tempdb and Temporary Object Considerations**