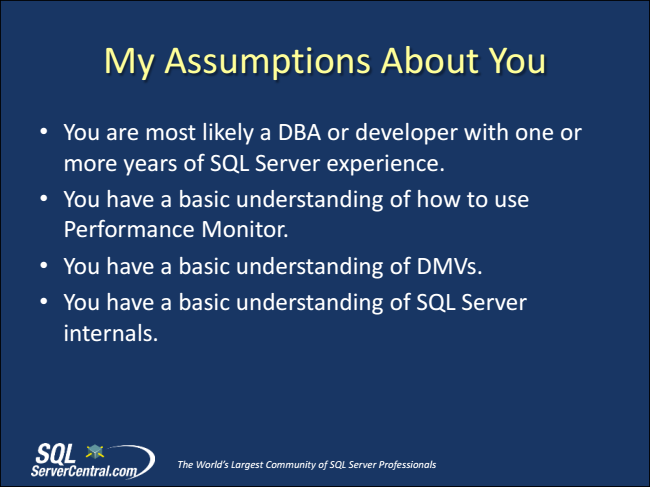


*The World’s Largest Community of SQL Server Professionals*

How to Optimize TEMPDB Performance

Brad M. McGehee Director of DBA Education Red Gate Software www.bradmcgehee.com/presentations



My Assumptions About You

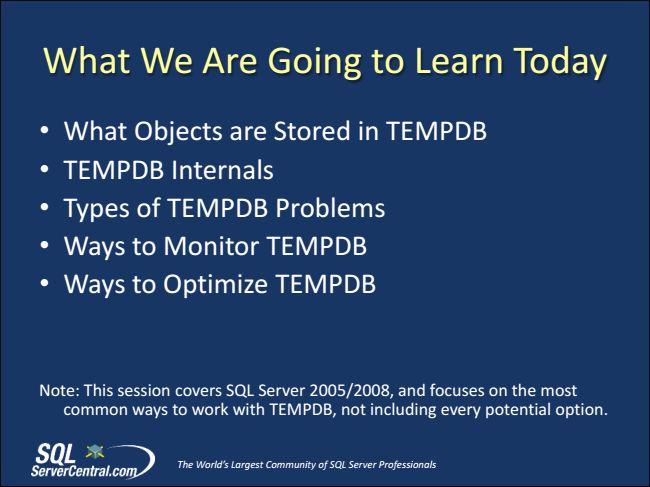
• You are most likely a DBA or developer with one or more years of SQL Server experience.

• You have a basic understanding of how to use Performance Monitor.

• You have a basic understanding of DMVs.

• You have a basic understanding of SQL Server internals.

*The World’s Largest Community of SQL Server Professionals*

**

What We Are Going to Learn Today

• What Objects are Stored in TEMPDB

• TEMPDB Internals

• Types of TEMPDB Problems

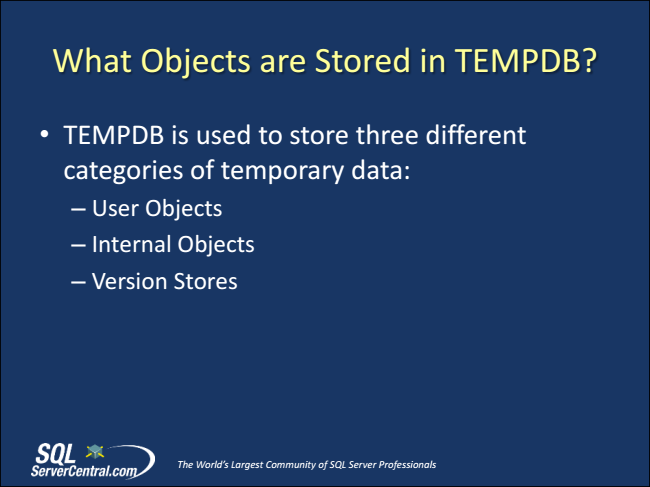
• Ways to Monitor TEMPDB

• Ways to Optimize TEMPDB

Note: This session covers SQL Server 2005/2008, and focuses on the most

common ways to work with TEMPDB, not including every potential option.

*The World’s Largest Community of SQL Server Professionals*

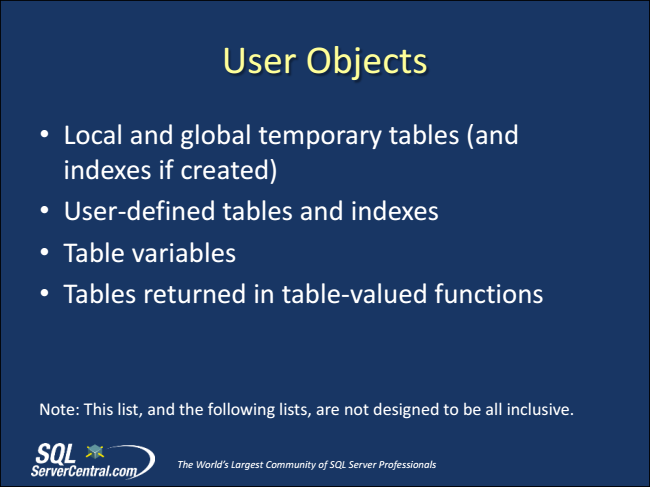
**

What Objects are Stored in TEMPDB?

• TEMPDB is used to store three different categories of temporary data:

– User Objects – Internal Objects – Version Stores

*The World’s Largest Community of SQL Server Professionals*

**

User Objects

• Local and global temporary tables (and indexes if created)

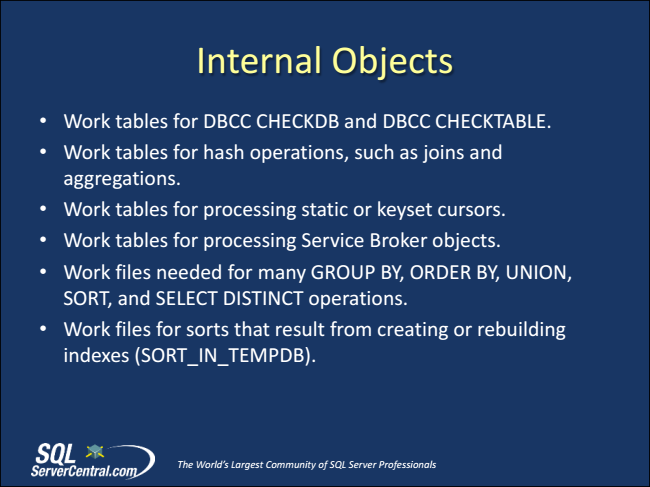
• User-defined tables and indexes

• Table variables

• Tables returned in table-valued functions

Note: This list, and the following lists, are not designed to be all inclusive.

*The World’s Largest Community of SQL Server Professionals*

**

Internal Objects

• Work tables for DBCC CHECKDB and DBCC CHECKTABLE.

• Work tables for hash operations, such as joins and aggregations.

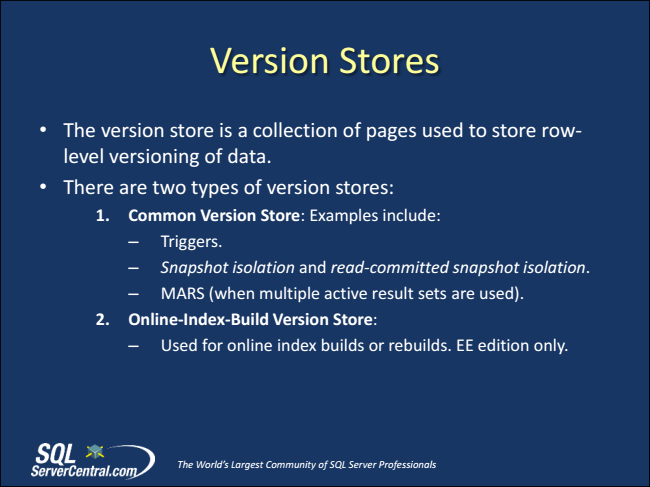
• Work tables for processing static or keyset cursors.

• Work tables for processing Service Broker objects.

• Work files needed for many GROUP BY, ORDER BY, UNION, SORT, and SELECT DISTINCT operations.

• Work files for sorts that result from creating or rebuilding indexes (SORT\_IN\_TEMPDB).

*The World’s Largest Community of SQL Server Professionals*

**

Version Stores

• The version store is a collection of pages used to store row- level versioning of data.

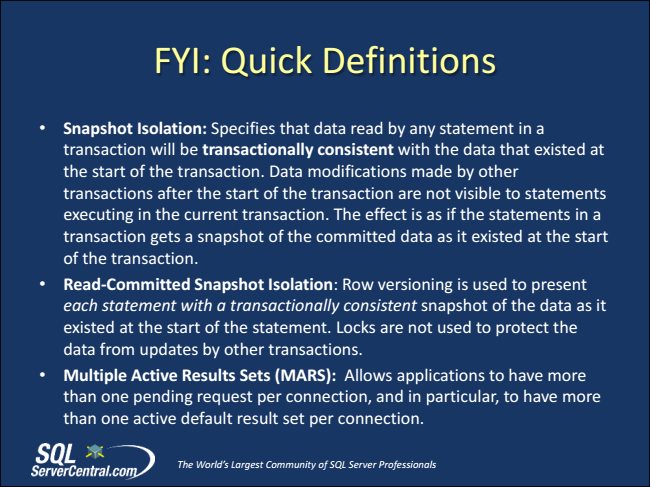
• There are two types of version stores:

**1. Common Version Store: Examples include:**

– Triggers. – Snapshot isolation and read-committed snapshot isolation. – MARS (when multiple active result sets are used). 2. Online-Index-Build Version Store:

– Used for online index builds or rebuilds. EE edition only.

*The World’s Largest Community of SQL Server Professionals*

**

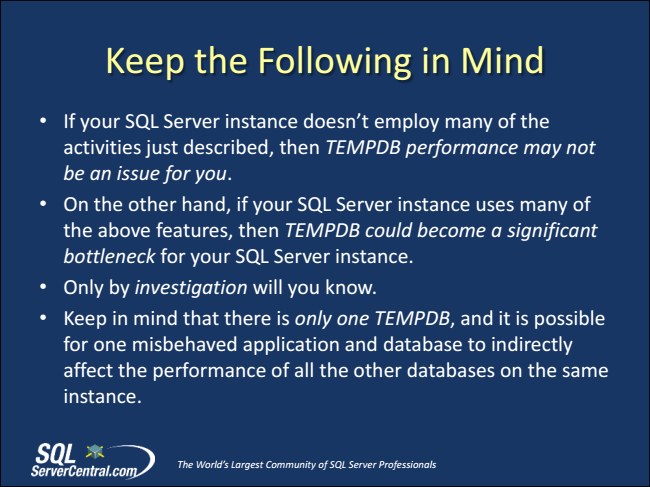
FYI: Quick Definitions

• Snapshot Isolation: Specifies that data read by any statement in a transaction will be transactionally consistent with the data that existed at the start of the transaction. Data modifications made by other transactions after the start of the transaction are not visible to statements executing in the current transaction. The effect is as if the statements in a transaction gets a snapshot of the committed data as it existed at the start of the transaction.

• Read-Committed Snapshot Isolation: Row versioning is used to present each statement with a transactionally consistent snapshot of the data as it existed at the start of the statement. Locks are not used to protect the data from updates by other transactions.

• Multiple Active Results Sets (MARS): Allows applications to have more than one pending request per connection, and in particular, to have more than one active default result set per connection.

*The World’s Largest Community of SQL Server Professionals*

**

Keep the Following in Mind

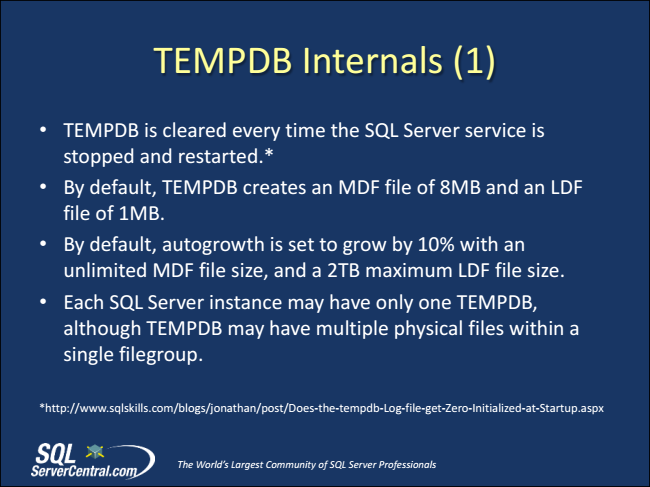
• If your SQL Server instance doesn’t employ many of the activities just described, then TEMPDB performance may not be an issue for you.

• On the other hand, if your SQL Server instance uses many of the above features, then TEMPDB could become a significant bottleneck for your SQL Server instance.

• Only by investigation will you know.

• Keep in mind that there is only one TEMPDB, and it is possible for one misbehaved application and database to indirectly affect the performance of all the other databases on the same instance.

*The World’s Largest Community of SQL Server Professionals*

**

TEMPDB Internals (1)

• TEMPDB is cleared every time the SQL Server service is stopped and restarted.\*

• By default, TEMPDB creates an MDF file of 8MB and an LDF file of 1MB.

• By default, autogrowth is set to grow by 10% with an unlimited MDF file size, and a 2TB maximum LDF file size.

• Each SQL Server instance may have only one TEMPDB, although TEMPDB may have multiple physical files within a single filegroup.

\*http://www.sqlskills.com/blogs/jonathan/post/Does-the-tempdb-Log-file-get-Zero-Initialized-at-Startup.aspx

*The World’s Largest Community of SQL Server Professionals*