SQL Server: Myths and Misconceptions

Module 6: Transact-SQL

Paul S. Randal
Paul@SQLskills.com

pluralsight
hardcore developer training

Introduction

- **Coding in Transact-SQL is just like coding in another language**
  - Learning curve
  - Some things the language references don't tell you

- **In this module:**
  - Five myths around Transact-SQL usage

Transact-SQL Myth #1

UNTRUE!!

- **Myth: Nested transactions exist in SQL Server**

- **Although the syntax is there, it's meaningless**
- **Committing a 'nested' transaction does nothing**
- **Rolling back a 'nested' transaction rolls back the entire outer transaction**
  - Regardless of whether inner 'nested' transactions have been committed
- **Nothing actually commits until @@TRANCOUNT = 0**
  - This leads to problems with transaction log growth

- **To allow rolling back to a point in a transaction, use save points**

Transact-SQL Myth #2

UNTRUE!!

- **Myth: DDL triggers are INSTEAD OF triggers**

- **DDL triggers are (unfortunately) AFTER triggers**
  - The operation is performed and then rolled-back
  - This can be very expensive
- **Can be better to use user/schema separation and explicit permissions than relying on DDL triggers**
  - E.g. using DENY/REVOKE of ALTER TABLE permission rather than using a DDL trigger to catch ALTER TABLE operations

Transact-SQL Myth #3

UNTRUE!!

- **Myth: SELECT COUNT (*) will always use a table scan**

- **SELECT COUNT (*) will use the smallest structure that contains the correct number of rows**
  - I.e. it will not use a filtered index
- **The query processor will pick the structure that it computes to require the fewest number of page reads to allow the rows to be counted**
  - This will be the smallest nonclustered index (if present)

- **Unfortunately, the query processor has no knowledge of what pages are in memory**
  - This may mean that additional physical reads are required, compared to using a larger index that is already in memory

Transact-SQL Myth #4

UNTRUE!!

- **Myth: There is no way to find the physical location of a particular record in a database**

- **From SQL Server 2008 onwards, the %%PHYSLOC%% function can be used to show the physical location of a record**
  - Gives back a hexadecimal RID (Record Identifier)
  - Also a function fn_PhysLocCracker to give file, page, slot
- **Undocumented and unsupported**

- **Uses include:**
  - Investigating corruption
  - Correlating wait activity with table data

Transact-SQL Myth #5

YES AND NO!!

- **Myth: Dynamic Management Views cannot be run against a database set to the 80 compatibility-mode**

- **They can, as long as no 90+ compatibility mode constructs are used**
  - E.g. the use of OBJECT_ID as an embedded parameter to a DMV is not allowed in the 80 compatibility mode
  - E.g. when using the sys.dm_db_index_physical_stats DMV
- **To be able to use such constructs, run the DMV in the context of a 90+ compatibility mode database and point it at the 80 compatibility mode database, if possible**
  - E.g. for standardizing the code that runs index fragmentation analysis