Implementing Snapshot Isolation

Gerald Britton
IT SOLUTIONS DESIGNER, SQL SERVER SPECIALIST
@GeraldBritton www.linkedin.com/in/geraldbritton

Overview

- Introducing snapshot isolation
- Database settings
- READ COMMITTED
- DML operations and conflicts
- tempdb space usage
- Demos

Snapshot Isolation

The SQL Server Database Engine maintains versions of each row that is modified

The chance that a read operation will block other transactions is greatly reduced

SQL Server uses a copy-on-write mechanism when a row is modified or deleted

tempdb is used to hold the version store !

Demo

Database setup

Read committed snapshot isolation

Demo

SNAPSHOT isolation level

Demo

Dynamic Management Views (DMVs)

Locking vs. Row Versioning

Locking (pessimistic)	Row versioning (optimistic)
Read uncommitted	Read committed snapshot isolation
Read committed	Snapshot isolation level
Repeatable read	
Serializable	
ANSI SQL-92 compliant	Proprietary
Better for long-running updates	Better for read-heavy operations
Normal tempdb usage	Extra usage of tempdb (version store)
More blocking = less concurrency	Less blocking = greater concurrency

Summary

Row versioning via snapshot isolation

- Optimistic concurrency

Read committed snapshot isolation

- RCSI

Transactional snapshot isolation

DMVs

Locking vs row versioning

- No one-size-fits-all
- Benchmark and test
- SET READ_COMMITTED_SNAPSHOT ON