Module 5

Concurrency and transaction handling

Module Overview

- Concurrency and transactions
- Locking internals

Lesson: Concurrency and transactions

- Concurrency problems
- Transaction isolation levels
- Working with row versioning
- Transactions
- Working with transactions

Concurrency problems

- Protected from by default
  - Dirty read
    - Uncommitted data is included in results
  - Double read
    - Data in a range is read twice because the rows moves during the read
  - Missing rows
    - Opposite of above, data is moved during read so that rows are missed
- Not protected from by default
  - Non-repeatable read
    - ”Inconsistent analysis”
    - Data changes between two identical SELECT statements within a transaction
  - Phantom read
    - Data inserted into range after read

The ACID Properties of Statements & Transactions
https://sqlperformance.com/2014/02/t-sql-queries/confusion-caused-by-trusting-acid

Transaction Isolation Levels (ODBC)
https://docs.microsoft.com/en-us/sql/odbc/reference/develop-app/transaction-isolation-levels?view=sql-server-ver15

Dirty read, Non-repeatable read, and Phantom read
https://jennyttt.medium.com/dirty-read-non-repeatable-read-and-phantom-read-bd75dd69d03a

Transaction isolation levels

- Pessimistic isolation levels:
  - READ UNCOMMITTED
  - READ COMMITTED
    - READ_COMMMITTED_SNAPSHOT OFF
  - REPEATABLE READ
  - SERIALIZABLE
- Optimistic (row versioning) isolation levels:
  - READ COMMITTED
    - READ_COMMMITTED_SNAPSHOT ON
  - SNAPSHOT
    - Be aware of possibilities for update conflict errors

SQL Server Isolation Levels : A Series
https://sqlperformance.com/2014/07/t-sql-queries/isolation-levels

SET TRANSACTION ISOLATION LEVEL (Transact-SQL)
https://docs.microsoft.com/en-us/sql/t-sql/statements/set-transaction-isolation-level-transact-sql

Working with row versioning

- Row versioning benefits:
  - Fewer locks
  - Less blocking
- Row versioning aspects:
  - Versioning adds load to tempdb
  - Applications need to handle update conflicts for true SNAPSHOT
- Other considerations:
  - Lock hints still apply
  - Writers still block writers
  - Setting READ_COMMITTED_SNAPSHOT ON requires zero connections to the database

Row Versioning Concurrency in SQL Server
https://www.red-gate.com/simple-talk/databases/sql-server/t-sql-programming-sql-server/row-versioning-concurrency-in-sql-server/

Transactions

- A logical unit of work, made up of one or more Transact-SQL statements
  - **A**tomicity
  - **C**onsistency
  - **I**solation
  - **D**urability
- Transaction management modes:
  - Auto-commit
  - Explicit transactions
  - Implicit transactions

Demo Transaction semantics

The ACID Properties of Statements & Transactions
https://sqlperformance.com/2014/02/t-sql-queries/confusion-caused-by-trusting-acid

Working with transactions

- Naming transactions:
  - Label only; no effect on code
- Nesting transactions:
  - Only the state of the outer transaction has any effect
  - @@TRANCOUNT track transaction nesting
- Terminating transactions:
  - Resource error
  - SET XACT_ABORT
  - Connection closure
- Transaction best practices:
  - Keep transactions as short as possible

Demo Concurrency

Transactions in SQL Server for beginners
https://www.sqlshack.com/transactions-in-sql-server-for-beginners/

Error and Transaction Handling in SQL Server
https://www.sommarskog.se/error_handling/Part1.html

BEGIN TRANSACTION (Transact-SQL)
https://docs.microsoft.com/en-us/sql/t-sql/language-elements/begin-transaction-transact-sql

COMMIT TRANSACTION (Transact-SQL)
https://docs.microsoft.com/en-us/sql/t-sql/language-elements/commit-transaction-transact-sql

ROLLBACK TRANSACTION (Transact-SQL)
https://docs.microsoft.com/en-us/sql/t-sql/language-elements/rollback-transaction-transact-sql

SET XACT_ABORT (Transact-SQL)
https://docs.microsoft.com/en-us/sql/t-sql/statements/set-xact-abort-transact-sql

Lesson: Locking Internals

- Lock granularity and escalation
- Types of locks
- Lock mode compatibility
- Locking during data modifications
- Deadlocks
- Locking hints
- Monitoring locking and blocking

Lock granularity and escalation

- The lock manager decides at what level locks should be held:
  - Row (RID or KEY)
  - Page (PAGE)
  - Table (OBJECT)
- Each lock costs 128 bytes in memory
  - I.e., 10 million locks costs approx. 1.2 GB memory
- Locks can escalate at run-time
  - Row and page locks escalate to table locks
  - Control at table level with ALTER TABLE SET LOCK_ESCALATION
    - Table (default)
    - Disable (be careful)
    - Partition
  - Control at session or instance level with trace flags 1224 and 1211

Lock granularity and hierarchies
https://docs.microsoft.com/en-us/sql/relational-databases/sql-server-transaction-locking-and-row-versioning-guide#lock-granularity-and-hierarchies

SQL Server, Locks object
https://docs.microsoft.com/en-us/sql/relational-databases/performance-monitor/sql-server-locks-object

SQL Server Concurrency: Locking, Blocking and Row Versioning (free e-book)
https://www.red-gate.com/library/sql-server-concurrency-locking-blocking-and-row-versioning

Types of locks

- Also known as Lock Modes
- Shared Sh
- Exclusive X
- Update U
- Intent I
- Key-range
- Schema stability Sch-S
- Schema modification Sch-M
- Bulk update BU

All about locking in SQL Server
https://www.sqlshack.com/locking-sql-server/

Lock modes
https://docs.microsoft.com/en-us/sql/relational-databases/sql-server-transaction-locking-and-row-versioning-guide#lock_modes

Lock mode compatibility

Compatibility between common lock modes

	Existing granted mode					
Requested mode	IS	S	U	IX	SIX	X
Intent shared (IS)	Yes	Yes	Yes	Yes	Yes	No
Shared (S)	Yes	Yes	Yes	No	No	No
Update (U)	Yes	Yes	No	No	No	No
Intent exclusive (IX)	Yes	No	No	Yes	No	No
Shared with intent exclusive (SIX)	Yes	No	No	No	No	No
Exclusive (X)	No	No	No	No	No	No

Lock compatibility
https://docs.microsoft.com/en-us/sql/relational-databases/sql-server-transaction-locking-and-row-versioning-guide#lock_compatibility

Locking during data modifications

- Relevant data pages located in the Buffer Pool
- Locks for modifications:
  - Update lock on possibly affected rows
    - If the row turn out to qualify, “convert” to exclusive
    - If not, then release update lock and go to next row
      - Intent exclusive lock on pages
      - Intent exclusive lock on table
  - Shared lock on database

Deadlocks

- Deadlocks are resolved by the Lock Manager:
  - Runs every five seconds by default
  - Frequency increases as deadlocks are detected
  - Deadlock victim is selected and terminated
- Deadlock priority can be set
  - SET DEADLOCK PRIORITY
- Manage using Extended Events
  - xml_deadlock_report

Demo Deadlocks

Deadlocks
https://docs.microsoft.com/en-us/sql/relational-databases/sql-server-transaction-locking-and-row-versioning-guide#deadlocks

What are SQL Server deadlocks and how to monitor them
https://www.sqlshack.com/what-are-sql-server-deadlocks-and-how-to-monitor-them/

Locking hints

- Hints affecting Lock Mode:
  - ROWLOCK
  - PAGLOCK
  - TABLOCK
  - TABLOCKX
  - UPDLOCK
  - XLOCK
  - READPAST
- Hints affecting Isolation level at table level:
  - READCOMMITTED
  - READCOMMITTEDLOCK
  - READUNCOMMITTED or NOLOCK
  - REPEATABLEREAD
  - SERIALIZABLE or HOLDLOCK

Hints (Transact-SQL) – Table
https://docs.microsoft.com/en-us/sql/t-sql/queries/hints-transact-sql-table

Understanding the SQL Server NOLOCK hint
https://www.mssqltips.com/sqlservertip/2470/understanding-the-sql-server-nolock-hint/

Using NOLOCK? Here's How You'll Get the Wrong Query Results.
https://www.brentozar.com/archive/2018/10/using-nolock-heres-how-youll-get-the-wrong-query-results/

"But NOLOCK Is Okay When My Data Isn't Changing, Right?"
https://www.brentozar.com/archive/2019/08/but-nolock-is-okay-when-the-data-isnt-changing-right/

Monitoring locking and blocking

- sys.dm_tran_locks
- Download any of below
  - sp_whoisactive
  - beta_lockinfo
  - sp_blitzlock
- Activity Monitor in SSMS
- Troubleshoot deadlock by capturing below event in an Extended Event trace
  - sqlserver.xml_deadlock_report

Demo Locking information

sp_whoisactive
http://whoisactive.com/

beta_lockinfo
https://www.sommarskog.se/sqlutil/beta_lockinfo.html

Lab 5: Concurrency and transaction handling

- Ex 1: Improve concurrency
- Ex 2: Capture deadlock information into a trace

Estimated Time: 30 minutes