SQLintersection

Session: Tuesday, 3:45pm-5:00pm

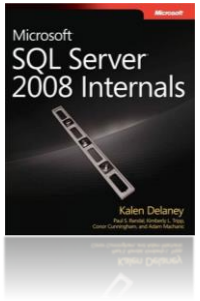
Statistics and Estimates: What you NEED to Know!

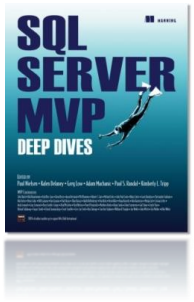
Kimberly L. Tripp
President / Founder, SQLskills.com
Kimberly@SQLskills.com
@KimberlyLTripp

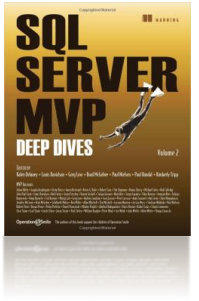
SQL *intersection*

SQLskills
immerse yourself in sql server

Author/Instructor: Kimberly L. Tripp





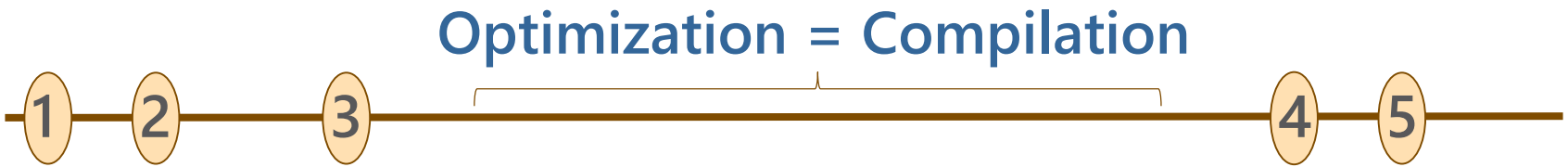


- Consultant/Trainer/Speaker/Writer
- President/Founder, SYSolutions, Inc.
  - e-mail: Kimberly@SQLskills.com
  - blog: http://www.SQLskills.com/blogs/Kimberly
  - Twitter: @KimberlyLTripp
- Author/Instructor for SQL Server Immersion Events
- Instructor for multiple rotations of both the SQL MCM & Sharepoint MCM
- Author/Manager of SQL Server 2005 & 2008 Launch Content
- Author/Speaker at Microsoft TechEd, SQLPASS, ITForum, TechDays, SQLIntersection
- Author of several SQL Server Whitepapers on MSDN/TechNet: Partitioning, Snapshot Isolation, Manageability, SQLCLR for DBAs
- Author/Presenter for more than 25 online webcasts on MSDN and TechNet
- Author/Presenter for multiple online courses at Pluralsight
- Co-author MSPress Title: SQL Server 2008 Internals, the SQL Server MVP Project (1 & 2), and SQL Server 2000 High Availability
- Owner and Technical Content Manager of the SQLintersection conference

Overview

- **Statement execution simplified**
- **Cost-based Optimization**
- **Demo**
  - Estimates
  - Statistics
  - Heuristics
- **Creating statistics**
- **Updating statistics**
- **Maintaining statistics**

Statement Execution Simplified



Optimization: this is really where you can have the most impact

1. Parse
2. Standardization / normalization / algebrization ⇒ query tree
3. Cost-based optimization
4. Compilation
5. Execution

Cost-Based Optimization

- **Find a reasonable subset of possible algorithms to access data based on:**
  - The query: sometimes a rewrite helps (join rewritten as sub-query or vice versa)
  - Any joins: sometimes a derived table helps (sub-query in the FROM clause)
  - Any SARGs: sometimes rewriting SARGs helps (well-defined predicates)
  - Data selectivity
  - Join density
- **The more information the optimizer has the better…**
  - How do you provide the BEST information?
- **Understanding optimization / estimation is the most important for troubleshooting a wide variety of solvable problems!**

Demo

Estimates / Statistics / Heuristics

What's going on when you execute that query…

SQL *intersection*

What Kinds of Statistics Exist?

- **Statistics on indexes**
- **Auto-created statistics**
  - Named _WA_SYS_
    - Paul's blog: How are auto-created column stats named? (http://bit.ly/1gjY28K)
  - Created automatically when a missing statistics event is encountered
  - Permanent objects in the database (auto-updated if option is on)
- **User-created statistics**
  - Could have been recommended by DTA and named _dta_stat_
- **Hypothetical indexes**
  - Created during DTA's analysis phase / dropped by letting DTA complete successfully
  - Can be created manually for "what if" analysis using auto pilot
  - IMPORTANT: Are not used by the optimizer, only auto-created and user-created statistics are used for executing queries / optimization

When Are They Created?

- **Automatically**
  - For all indexes
  - When “auto create statistics” is ON AND when the optimizer thinks that statistics would be a good idea (often when an column is in a SARG [search argument] or a join and does not have an index or statistic with that column as the high-order element)
- **Manually**
  - EXEC [sp_createstats]
  - Using CREATE STATISTICS

Tip: Leave Auto Create Statistics ON

What If the Data Changes?

- **Automatically updated statistics**
  - If auto update statistics is ON (for both the DB and the statistic)
    - Prior to SQL Server 2016: minimum of 500 rows + 20% of the data changes
    - SQL Server 2016+ (or, using trace flag 2371 in SQL Server 2008 R2-2014): adjustable percentage tied to the number or rows (decreasing from 20% at 25K rows)
- **Manually update statistics**
  - Executing UPDATE STATISTICS
    - Might want to decrease the frequency of updating for highly volatile tables where distribution isn't changing significantly and you see a lot of "statistics" events
    - Might want to increase the frequency of updating for large table where distribution is changing significantly but you're not reaching 20%
  - Consider turning off auto update stats at the statistic-level instead of the database-level (more granular control)
    - STATISTICS_NORECOMPUTE on the index definition
    - NORECOMPUTE on the statistics definition

Maintaining Statistics

- **Manually: but automated through a job**
  - Executing sp_updatestats
    - Sledgehammer maintenance. Only one row has to have been modified.
  - Ola Hallengren's code
    - http://ola.hallengren.com/
    - Use @UpdateStatistics parameter / settings
  - Roll your own?
    - Programmatically evaluate the stat_header data
    - Use sys.dm_db_stats_properties and do something like: *if 2% of the data has changed since last stats update AND (stats_date older than 1 week OR last update was sampled (sampled < rows))* then UPDATE STATS WITH FULLSCAN
- **Auto update stats (as a safety measure)**
- **Asynchronous update stats (unlikely to cause a problem)**

Review

- **Statement execution simplified**
- **Cost-based Optimization**
- **Demo**
  - Estimates
  - Statistics
  - Heuristics
- **Creating statistics**
- **Updating statistics**
- **Maintaining statistics**

Resources

PLURALSIGHT

- **Demo code/samples: SQLskills, Resources, Demo Scripts and Sample Databases**
- **Courses on Pluralsight: www.pluralsight.com**
  - SQL Server: Why Physical Database Design Matters
  - SQL Server: Optimizing Ad Hoc Statement Performance
  - SQL Server: Optimizing Stored Procedure Performance (Parts 1 and 2)
    - Part 2 has an entire section on session settings (for performance-related features)
  - SQL Server: Indexing for Performance
  - Coming soon: SQL Server: Query Tuning with Indexes
- **Auto pilot article: Hypothetical Indexes on SQL Server**
  - https://www.simple-talk.com/sql/database-administration/hypothetical-indexes-on-sql-server/

SQL *intersection*

Questions?

SQLskills
immerse yourself in sql server

Don’t forget to complete an online evaluation!

Cardinality Estimates with Statistics NOT Heuristics: What you NEED to Know!

Session by Kimberly L. Tripp

Your evaluation helps organizers build better conferences and helps speakers improve their sessions.

SQL *intersection*

Thank you!