SQL Server:
Why Physical Database Design Matters

Module 4: Data Types and Query Performance

Kimberly L. Tripp
Kimberly@SQLskills.com
http://www.SQLskills.com/blogs/Kimberly

pluralsight
hardcore developer training

Introduction: Does Parameter Type Matter?

- **You allow your users the ability to write ad hoc queries (scary!) and they've recently started to learn that they need to precede Unicode columns with an N – this is good, right?**
  - It's good when they're accessing Unicode columns
  - It might be really bad when they're not (but, it doesn't error)
- **If they use N'string' with a non-Unicode (i.e. ASCII) column, won't SQL Server will just discard it?**
  - If the variable/value is ASCII and the column is Unicode then only the variable/value has to be converted
  - If the column is ASCII and the variable/value is Unicode then the entire column needs to be converted
- **This happens "implicitly"**
- **This inconsistency can be very costly**

Implicit Conversions

- **These can be found anywhere a column is being compared to a:**
    - Literal
    - Variable
    - Parameter
- **The main question is: are they defined as the same data type?**
    - If yes, then no problem
    - If no, then there could be a problem
- **To successfully compare, the lower data type must be converted to the higher data type (based on the data type hierarchy and still only if conversions are allowed implicitly)**
    - If the column is the higher type, only the literal/variable/parameter needs to be converted, and this is cheap
    - If the column is the lower type then the ENTIRE column needs to be converted, and this can be tremendously expensive as this requires a scan of some sort

More on Implicit Conversions

- There are many conversions that can be expensive
- My colleague Jonathan Kehayias has done a fantastic job documenting all the data type conversions that cause problems here:
  - “Implicit Conversions that cause Index Scans” at http://bit.ly/XFS6On
- Jonathan also wrote some excellent plan cache analysis code to see if any implicit conversions have occurred
  - “Finding Implicit Column Conversions in the Plan Cache” at http://bit.ly/17MdijL

Probe Residual

- **Similar to an implicit conversion but performed when the columns of a join are only implicitly compatible and one column is of a lower type**
- **To be able to do the comparison between the columns, the column of a lower type must be brought up to the higher type and then evaluated**
- **To do the comparison/evaluation a “residual” is created**
- **Sometimes you can see this directly in the plan (as a scalar computation, immediately prior to the join) and in other cases you cannot**
- **But you can always see the Probe Residual:**
  - In Showplan
  - In the plan cache

Summary: Inconsistencies in Data Types

- **Implicit_conversion**
  - Query doesn’t match the column definition
- **Probe Residual**
  - May add a residual value for comparisons
  - Requires temporary residual to be created on EVERY execution
- **Inconsistencies in any layer of your application ecosystem can be costly:**
  - Tables
  - Stored procedures/functions
  - Ad hoc queries/application interface
- **Consider tools like those in Visual Studio for refactoring and static code analysis**

Just the Tip of the Iceberg

- **Data types: column size / row size / consistency**
  - This course
- **Ad hoc statements: plan cache / parameter sniffing**
  - **Coming soon:** *SQL Server: Optimizing Ad Hoc Statement Performance*
- **Stored Procedures: Parameter Sniffing / Recompilation**
  - **Coming soon:** *SQL Server: Optimizing Stored Procedure Performance*
- **Indexes: creation / overhead / maintenance**
- **Queries: predicates / functions / WHERE clause vs. FROM clause / isolating expressions**
- **Statistics: accuracy / cardinality estimation and limitations**

Where To Go Next?

- **Check for new Pluralsight courses from me**
  - I'm going to stay within the developer/database development area for my first few courses
    - Targeting best practices and typically using a "Problem/Solution" approach
- **Check out these SQLskills courses on Pluralsight that are the most appropriate courses for you to consider next:**
  - *Developing and Deploying SQL Server ISV Applications*
  - *SQL Server: Common Performance Issue Patterns*
  - *SQL Server: Troubleshooting Query Plan Quality Issues*
- **Everyone using SQL Server should watch Paul Randal's course: *SQL Server: Myths and Misconceptions***
  - You'd be surprised at how many of these you might think you know
  - It gives you all sorts of great advice – across the entire product!

Course Summary

- **Performance doesn’t just “happen”**
- **Do not just expect the SQL Server defaults to perfectly support every environment**
  - It’s not all One-Size-Fits-All!
- **The effect on performance of incorrect data type and table structure choice can be huge**
  - Design your tables with knowledge of the data/application
  - Use the correct data type VERY consistently
  - Consistency, consistency, consistency!
- **There’s so much more to know about designing for performance**
- **Thanks for watching!**