Faster Queries In ColdFusion & SQL Server

Who is this guy?

Eric Cobb

Database Development Manager MCSE: Data Platform | MCSE: Data Management and Analytics

1999-2013: "Webmaster", Programmer, Developer 2013+: SQL Server Database Administrator

GitHub: <u>https://github.com/ericcobb</u> Blog: <u>http://www.sqlnuggets.com</u> Twitter: @cfgears, @sqlnugg



What Tools Are We Using?

- SQL Server 2017 Developer Edition (Free Download)
- Public Stack Overflow Database (<u>Free Download</u>)
- ColdFusion 2018 Developer Edition (<u>Free Download</u>)
- ColdFusion Builder 2018 (<u>Trial Download</u>)
- SentryOne Plan Explorer (<u>Free Download</u>)

CFML Query Rules To Live By

Always Use CFQUERYPARAM

- Security, Data Validation, Enhanced Performance
- Uses Bind Variables to prevent SQL injection, and reuse Execution Plans
- Always Look For Caching Opportunities
 - Cache data that seldom changes (long-term caching)
 - Think of caching in terms of reducing hits on the database, even if for just a short period of time
 - Consider caching frequently run queries for reduced database load
 - Short-term caching can have big impact on busy systems

Caching

- Mark Kruger, ColdFusion Muse <u>Good Developers Practice Safe Query</u> <u>Caching</u>
 - Page on busy site was receiving over 200 views per minute (12,000 times/hour)
 - Query populated choices in a drop down list that changed frequently several times an hour
 - By caching query results for 3 minutes the number of database hits went from 12,000 per hour to 20 per hour
 - "Minimizing the number of calls to the database has an exponential effect on your server. Remember, database activity is virtually always the single most costly process on your web server."

STOP: {DEMO TIME}

Common Query Problems

ORDER BY, GROUP BY, and DISTINCT May Be Slowing Your Query Down

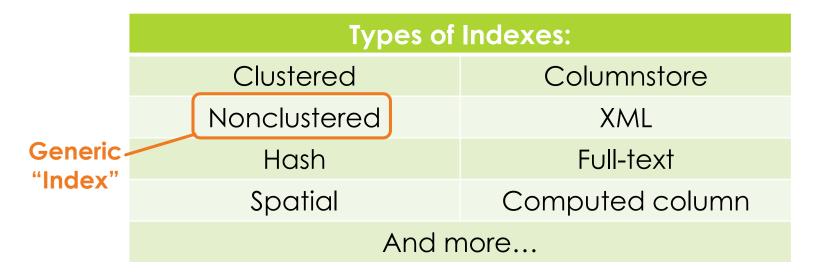
May have to use TempDB to process large or complex result sets

Data Type Mismatches

- Causes Implicit Conversions in SQL Server
- Indexing
 - "Poorly designed indexes and a lack of indexes are primary sources of database application bottlenecks." - <u>SQL Server Index Design Guide</u>

Index Basics

"Proper indexing is one of the best performance enhancements you can make to your database" – Microsoft



Index Basics

Clustered Index

- Tells SQL Server how to physically sort the records on disk
- The most important index you can apply to a table
- Primary Key = Clustered Index (usually) If you don't specify a Clustered Index SQL Server automatically creates one on a table's Primary Key

Nonclustered Index

- Copies the values from the specified columns
- Points to the actual data rows (via Clustered Index or Heap Row ID)
- Can have multiple Nonclustered Indexes on a table
 - ▶ SQL Server 2005 allows up to 249 per table; SQL Server 2008+ allows up to 999 per table

STOP: {DEMO TIME}

Eric Cobb

GitHub: <u>https://github.com/ericcobb</u> Blog: <u>http://www.sqlnuggets.com</u> Twitters: @cfgears, @sqlnugg