Vroom
PERFORMANCE TECHNOLOGIES

Microsoft SQL Server 2017

10 WAYS TO IMPROVE SQL QUERY PERFORMANCE

BOOST SYSTEM PERFORMANCE

MAXIMIZE CUSTOMER SATISFACTION

IMPROVE SERVICE DELIVERY

Enabling Growth Through Performance™

5 Steps For Improving Database Performance

TOP 10

Information Technology solutions are a critical element of customer satisfaction and business strategy. Improving performance improves customer experience and lowers costs. Potential customers form their impression based the speed, quality, and reliability of services delivered. Data shows that up to 88% of poor application performance issues can be attributed to the database.

Here are 5 steps that deliver higher performing systems that require less cloud or datacenter resources .

1 Avoid Multiple Joins in a Single Query

Avoid writing SQL queries that use multiple joins. This applies to all joins like outer joins, outer apply and other complex sub queries. Using multiple joins reduces the choices available to the Optimizer. The Optimizer may lose the ability to choose the join order and join type and be forced to use nested loop joins. Using queries with excessively complex cross apply and/or sub queries, will probably show evidence of severe performance consequences in the associated execution plans.

2 Understand the Data

Its critical to understanding the data, how it is stored and how queries are being performed to retrieve the data. A thorough understanding of data behavior leads to better decisions about index strategy. Decisions like which columns should have either a clustered index or non-clustered index will be stronger. If a clustered index is not on a unique column then SQL Server will maintain uniqueness by adding a unique identifier to every duplicate key, which adds significant overhead.

3

Have a Strategy for Managing Indexes

While indexes can significantly reduce the data retrieval time they can have the reverse effect on DML operations and may degrade query performance. This is why "understanding the data" plays such an important role. Managing indexing is a challenging task, but could help to improve SQL query performance and give the best query response time. It is critical to understand the overhead costs of DML operations of indexes.

4

Create a Highly Selective Index

Where possible use highly selective indexes. Selectivity can be defined as the percentage of qualifying rows in the table (qualifying number of rows/total number of rows). The lower the percentage the more useful the index. A non-clustered index is most useful if the ratio is around 5% or less. If the index can eliminate 95% of the rows from consideration it probably will be used. If not, then either a different index will be chosen or the table will be scanned.

5

Column Position in an Index

Order or position of a column in an index also plays a vital role to improve SQL query performance. An index can help improve the performance of a SQL query if the criteria of the query matches the columns that are leftmost in the index key. As a best practice, most selective columns should be placed leftmost in the key of a non-clustered index.

6

Drop All Unused Indexes

Dropping unused indexes can help to speed up data modifications without affecting data retrieval. Be careful that any infrequently run batch processes may use these indexes. An approach to handle such cases is creating these indexes in advance of batch processes and dropping them when the batch processes are complete. This will help reduce the overhead on the database.

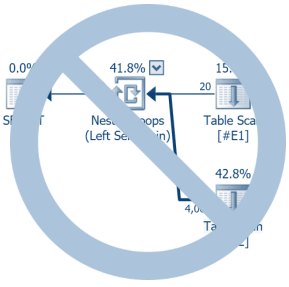
7

Eliminate Cursors from the Query

Try to remove cursors from the query and use set-based query; set-based query is more efficient than cursor-based. If there is a need to use cursor than avoid dynamic cursors as they tend to limit the choice of plans available to the query optimizer. Example: Dynamic cursors limit the optimizer to using nested loop joins.

8

Avoid Non-Correlated Scalar Sub Queries



Rewrite queries to remove any non-correlated scalar sub queries as separate queries instead of part of a main query. Store any output in a variable, which can be referred to in the main query or later part of the batch. This gives better options for the Optimizer, which will help return accurate cardinality estimates as well as a better execution plan.

9

Statistic Creation and Updates

It's critical to take care of statistic creation and regular updates for computed columns and multi-columns referred in the query; the query optimizer uses information about the distribution of values in one or more columns of a table statistics to estimate the cardinality, or number of rows, in the query result. These cardinality estimates enable the query optimizer to create a high-quality query plan.

10

Revisit Schema Definitions

Last but not least, revisit the schema definitions. Review whether appropriate FOREIGN KEY, NOT NULL and CEHCK constraints are in place or not. Availability of the right constraint on the right place will always improve the query performance. For example, FOREIGN KEY constraint help simplify joins by converting some outer or semi joins to inner joins and CHECK constraint also helps by removing unnecessary or redundant predicates.

SUMMARY

Recap

The ten tips listed have resulted in dramatic performance improvements for our clients. Understanding how data your data and how it's used is fundamental to lasting solutions of problems. Addressing Indexes is a great place to start and can result in significant quick wins. Reengineering queries can take more time but should be pursued none the less. Additionally these tips should be incorporated in any design standards utilized by your organization's SQL team.

Vroom Performance helps customers identify and solve the true root causes of poor system performance in MS SQL bases systems.

Our Products and Services help customers maximize the performance of their IT assets.

What our customers are saying

VPS has really helped us optimize our system integrations to minimize the impact on performance

Nancy

Application Support and Development

VPS allowed me focus on my job and not constantly chase performance and stability problems."

Ron

Software Specialist

Our solutions

Our core expertise

- IT performance
- database development
- database management
- regulatory compliance
- service management
- capacity management

Vroom PERFORMANCE TECHNOLOGIES

info@VroomPerformance.com

www.VroomPerformance.com