

### Introduction

SQL query performance improvement is a very thought-provoking topic between developers and the user community. Users always want a fast response on their data retrieval action and developers put forth their best efforts to provide the data in the shortest time, however, there is no straightforward way to define what is the best performance. Sometime it's debatable what is good and what is bad performance of a query but overall if you follow best practices during development, you can provide the best query response to users and avoid such discussions.. You can choose multiple ways to improve SQL query performance, which falls under various categories like re-writing the SQL query, creation and use of Indexes, proper management of statistics, etc.

<http://www.developer.com/db/slideshows/10-ways-to-improve-sql-query-performance.html>

### Avoid Multiple Joins in a Single Query

Try to avoid writing a SQL query using multiple joins that includes outer joins, cross apply, outer apply and other complex sub queries. It reduces the choices for Optimizer to decide the join order and join type. Sometime, Optimizer is forced to use nested loop joins, irrespective of the performance consequences for queries with excessively complex cross apply or sub queries.

### 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 it tends to limit the choice of plans available to the query optimizer. For example, dynamic cursor limits the optimizer to using nested loop joins.

### Avoid Use of Non-correlated Scalar Sub Query

You can re-write your query to remove non-correlated scalar sub query as a separate query instead of part of the main query and store the output in a variable, which can be referred to in the main query or later part of the batch. This will give better options to Optimizer, which may help to return accurate cardinality estimates along with a better plan.

### Avoid Multi-statement Table Valued Functions

Multi-statement TVFs are more costly than inline TVFs. SQL Server expands inline TVFs into the main query like it expands views but evaluates multi-statement TVFs in a separate context from the main query and materializes the results of multi-statement into temporary work tables. The separate context and work table make multi-statement TVFs costly.

### Creation and Use of Indexes

We are aware of the fact that Index can magically reduce the data retrieval time but have a reverse effect on DML operations, which may degrade query performance. With this fact, Indexing is a challenging task, but could help to improve SQL query performance and give you best query response time.

### Understand the Data

Understand the data, its type and how queries are being performed to retrieve the data before making any decision to create an index. If you understand the behavior of data thoroughly, it will help you to decide which column should have either a clustered index or non-clustered index. 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 leads to overhead. To avoid this type of overhead choose the column correctly or make the appropriate changes..

### Create a Highly Selective Index

Selectivity define the percentage of qualifying rows in the table (qualifying number of rows/total number of rows). If the ratio of the qualifying number of rows to the total number of rows is low, the index is highly selective and is most useful. A non-clustered index is most useful if the ratio is around 5% or less, which means if the index can eliminate 95% of the rows from consideration. If index is returning more than 5% of the rows in a table, it probably will not be used; either a different index will be chosen or created or the table will be scanned..

### Position a Column 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 to improve the SQL query performance if the criteria of the query matches the columns that are left most in the index key. As a best practice, most selective columns should be placed leftmost in the key of a non-clustered index.

### Drop Unused Indexes

Dropping unused indexes can help to speed up data modifications without affecting data retrieval. Also, you need to define a strategy for batch processes that run infrequently and use certain indexes. In such cases, creating indexes in advance of batch processes and then dropping them when the batch processes are done helps to reduce the overhead on the database.

### Statistic Creation and Updates

You need 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.

### Revisit Your Schema Definitions

Last but not least, revisit your schema definitions; keep on eye out that appropriate FOREIGN KEY, NOT NULL and CHECK constraints are in place or not. Availability of the right constraint on the right place always helps to improve the query performance, like FOREIGN KEY constraint helps to simplify joins by converting some outer or semi-joins to inner joins and CHECK constraint also helps a bit by removing unnecessary or redundant predicates.

### Conclusion

We discussed how SQL query performance can be improved by re-writing a SQL query, creation and use of Indexes, proper management of statistics and we revisited schema definitions. There are many more areas that can be looked at to improve the SQL query performance like using query hints, table hints and plan hints, etc.

<http://msdn.microsoft.com/en-us/library/ff647793.aspx>

<http://msdn.microsoft.com/en-us/library/ff650689.aspx>

<http://technet.microsoft.com/en-us/library/ms172984.aspx>



By [deleted]  
[cheatography.com/deleted-2754/](https://cheatography.com/deleted-2754/)

Not published yet.  
Last updated 20th January, 2017.  
Page 2 of 2.

Sponsored by **ApolloPad.com**  
Everyone has a novel in them. Finish Yours!  
<https://apollopadd.com>