



# Introduction to SQL Server Query Optimization with Database Engine Tuning Advisor

By: [Diogo Souza \(/sqlserverauthor/314/diogo-souza/\)](/sqlserverauthor/314/diogo-souza/) | Updated: 2018-11-12 | [Comments \(1\)](#) | Related: [More \(/sql-server-dba-resources/\)](/sql-server-dba-resources/) > [Performance Tuning \(/sql-server-tip-category/9/performance-tuning/\)](/sql-server-tip-category/9/performance-tuning/)



[\(/sql-server-webcast-signup/?id=828&src=tip\)](/sql-server-webcast-signup/?id=828&src=tip)

Free MSSQLTips Webinar: [Using SQL Server Query Store & Plan Guides to Fix Problem Queries \(/sql-server-webcast-signup/?id=828&src=tip\)](/sql-server-webcast-signup/?id=828&src=tip)

This webinar will cover how to use SQL Server plan guides and the query store to fix problematic queries and ways to improve query performance.

## Problem

Most of the times that we need to optimize a SQL Server query, a lot of questions and uncertainties come to our minds, just like if we could use any tool that would help with this type of improvement not only regarding the performance itself, but also in structural terms like with indexes, partitioning, DDL and DML, etc. Imagine you, being a SQL Server developer that needs to deal with lots of queries every day and still must consult the SQL Server DBA to check each of them, having such kind of optimizer in hands. That's when the *Database Engine Tuning Advisor*, present in SQL Server, comes to the table, providing a lot of great analysis and recommendations based on our queries and workloads. In this tip we will cover the use of this tool, especially improving the performance of our queries based on structural changes.

## Solution

The famous DTA (*Database Engine Tuning Advisor*), basically analyzes databases and gives some recommendations. From the [official docs \(https://docs.microsoft.com/en-us/sql/relational-databases/performance/database-engine-tuning-advisor?view=sql-server-2017\)](https://docs.microsoft.com/en-us/sql/relational-databases/performance/database-engine-tuning-advisor?view=sql-server-2017), you can find a series of things it helps with:



- Troubleshoot the performance of a specific problem query
- Tune a large set of queries across one or more databases
- Perform an exploratory what-if analysis of potential physical design changes
- Manage storage space

For this example, we're going to use the [AdventureWorks Databases and Scripts for SQL Server \(https://docs.microsoft.com/en-us/sql/samples/adventureworks-install-configure?view=sql-server-2017\)](https://docs.microsoft.com/en-us/sql/samples/adventureworks-install-configure?view=sql-server-2017), which is great as a data sample for test purposes. So, first, open SQL Server Management Studio and import the database following the official steps.

Then, let's define the query we want to optimize

```
-- sales for group of customers by year
SELECT
    year = YEAR(header.OrderDate),
    sum = SUM(detail.LineTotal)
FROM
    Sales.SalesOrderDetail detail
INNER JOIN
    Sales.SalesOrderHeader header ON detail.SalesOrderID = header.SalesOrderID
INNER JOIN
    Sales.Customer cust ON header.CustomerID = cust.CustomerID
WHERE
    cust.CustomerID > 10000 and cust.CustomerID < 10000000
GROUP BY
    YEAR(header.OrderDate)
```

Make sure to have the proper database selected in order to run the query and get the following results

The screenshot shows the SQL Server Enterprise Manager interface. In the left pane, the 'AdventureWorks2016CTP3' database is selected. The right pane displays a SQL query titled 'SQLQuery1.sql - DE...36C9F9\diogo (51))' with the following code:

```
-- sales for group of customers by year
SELECT
    year = YEAR(header.OrderDate),
    sum = SUM(detail.LineTotal)
FROM
    Sales.SalesOrderDetail detail
INNER JOIN
    Sales.SalesOrderHeader header ON detail.SalesOrderID = header.SalesOrderID
INNER JOIN
    Sales.Customer cust ON header.CustomerID = cust.CustomerID
WHERE
    cust.CustomerID > 10000 and cust.CustomerID < 10000000
GROUP BY
    YEAR(header.OrderDate)
```

Below the query, the 'Results' tab shows the output of the query:

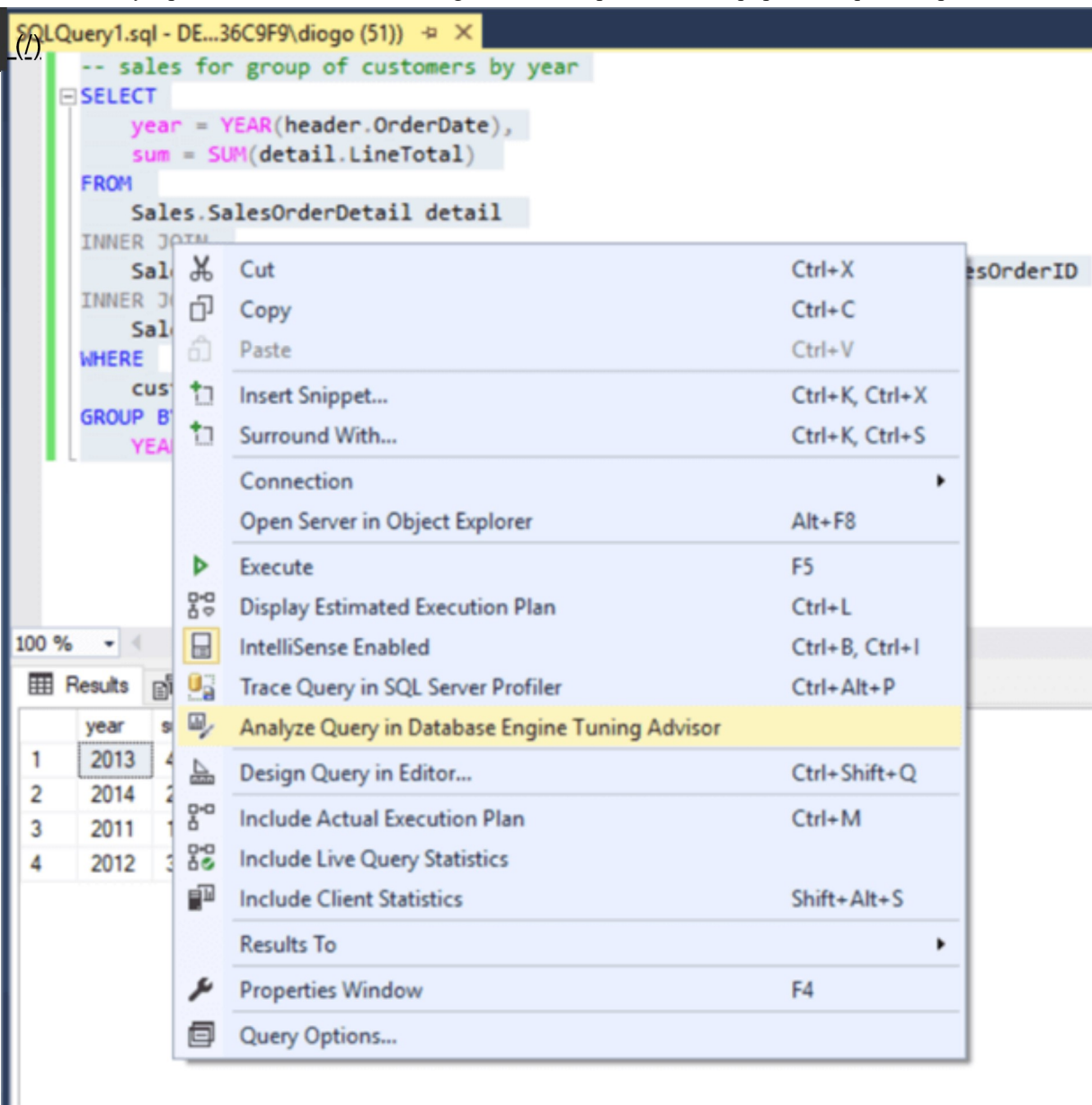
	year	sum
1	2013	43622479.051635
2	2014	20057928.810865
3	2011	12641672.212954
4	2012	33524301.324434

What we need to do now is to open the "SQL Server Database Engine Tuning Advisor" going to the menu *Tools > Database Engine Tuning Advisor*. Authenticate and create a new session name with the same options we have below.

The screenshot shows the 'SQL Server Database Engine Tuning Advisor' configuration window. The 'Session name' field is set to 'mssqltips-diogo-test'. The 'Workload' section has 'Query Store' selected. The 'Database for workload analysis' is set to 'AdventureWorks2016CTP3'. The 'Select databases and tables to tune' section shows a list of databases with 'AdventureWorks2016CTP3' selected.

Name	Selected Tables
<input checked="" type="checkbox"/> AdventureWorks2016CTP3	94 of 94
<input type="checkbox"/> master	<a href="#">Click to view individual tables</a>
<input type="checkbox"/> model	<a href="#">Click to view individual tables</a>
<input type="checkbox"/> msdb	<a href="#">Click to view individual tables</a>
<input type="checkbox"/> tempdb	<a href="#">Click to view individual tables</a>
<input type="checkbox"/> TuningAdvisorDevmedia	<a href="#">Click to view individual tables</a>
<input type="checkbox"/> TuningAdvisorMSSLtips	<a href="#">Click to view individual tables</a>

Then, get back to the query, right click it and click on the option "Analyze Query in Database Engine Tuning Advisor":



The screenshot shows a SQL Server Enterprise Manager interface. A query window titled "SQLQuery1.sql - DE...36C9F9\diogo (51)" is open, displaying a SQL query. The query is as follows:

```
-- sales for group of customers by year
SELECT
    year = YEAR(header.OrderDate),
    sum = SUM(detail.LineTotal)
FROM
    Sales.SalesOrderDetail detail
INNER JOIN
    Sales.SalesOrder header ON detail.SalesOrderID = header.SalesOrderID
WHERE
    customer_id = 1
GROUP BY
    YEAR(header.OrderDate)
```

A context menu is open over the query, listing various actions and their keyboard shortcuts. The "Analyze Query in Database Engine Tuning Advisor" option is highlighted in yellow.

Year	Sum
2013	4
2014	2
2011	1
2012	3

The Analyzer is going to run and give us an estimated improvement over the performance of our queries:

Estimated improvement: 33%

Partition Recommendations

Index Recommendations

Database Name	Object Name	Recommendation	Target of Recommendation	Details	Partition Scheme	Size (KB)	Definition
AdventureWorks2016CTP3	[Sales].[Customer]	create	_dta_index_Customer_5_1317579732__K1			72	([CustomerID] asc)
AdventureWorks2016CTP3	[Sales].[SalesOrderDetail]	create	_dta_index_SalesOrderDetail_5_1474104292__K1_9			2960	([SalesOrderID] asc) include ([LineTotal])
AdventureWorks2016CTP3	[Sales].[SalesOrderHeader]	create	_dta_index_SalesOrderHeader_5_1586104691__K11_K1_3			512	([CustomerID] asc, [SalesOrderID] asc) include ([OrderDate])

DTA includes a lot of different reports, from the performance and how costly the query is, to the way each entity relates to the other, the balance between the keys (indexes, etc.), and much more:

General | Tuning Options | Progress | Recommendations | Reports

Database(s) to tune: [DETA]

Workload: Inline

Maximum tuning time: 57 Minutes

Time taken for tuning: 1 Minute

Estimated percentage improvement: 64.09

Maximum space for recommendation (MB): 54

Tuning Reports

Select report:

- Statement cost report
- Statement cost report
- Event frequency report
- Statement detail report
- Statement-index relations report (current)
- Statement-index relations report (recommended)
- Statement cost range report
- Index usage report (current)
- Index usage report (recommended)
- Index detail report (current)
- Index detail report (recommended)
- View-Table relations report
- Workload analysis report
- Database access report
- Table access report
- Column access report

With them, we can check, for example, how much improvement we'll have when we decide to apply one of the suggested improvements. Let's look at the "Statement cost report", e.g., which says that our query can be improved by 33.07% if we apply the suggestions:

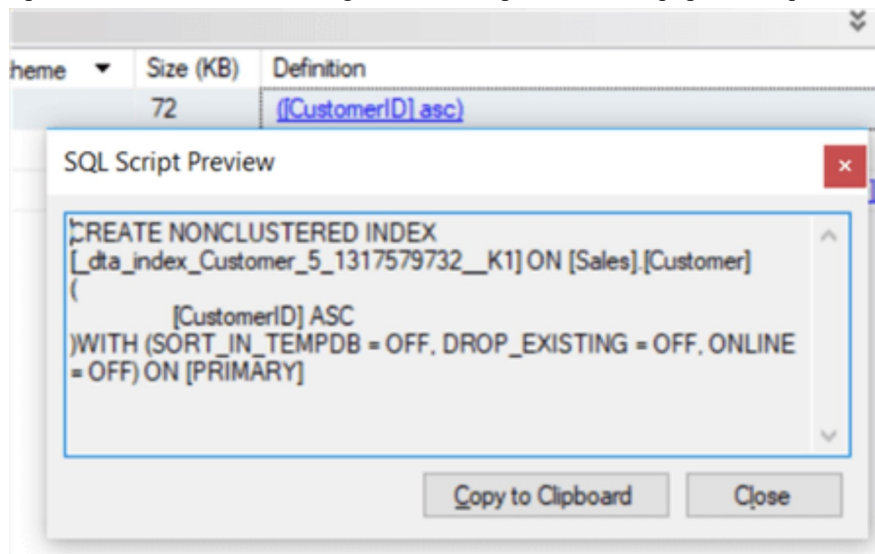
Tuning Reports

Select report: Statement cost report

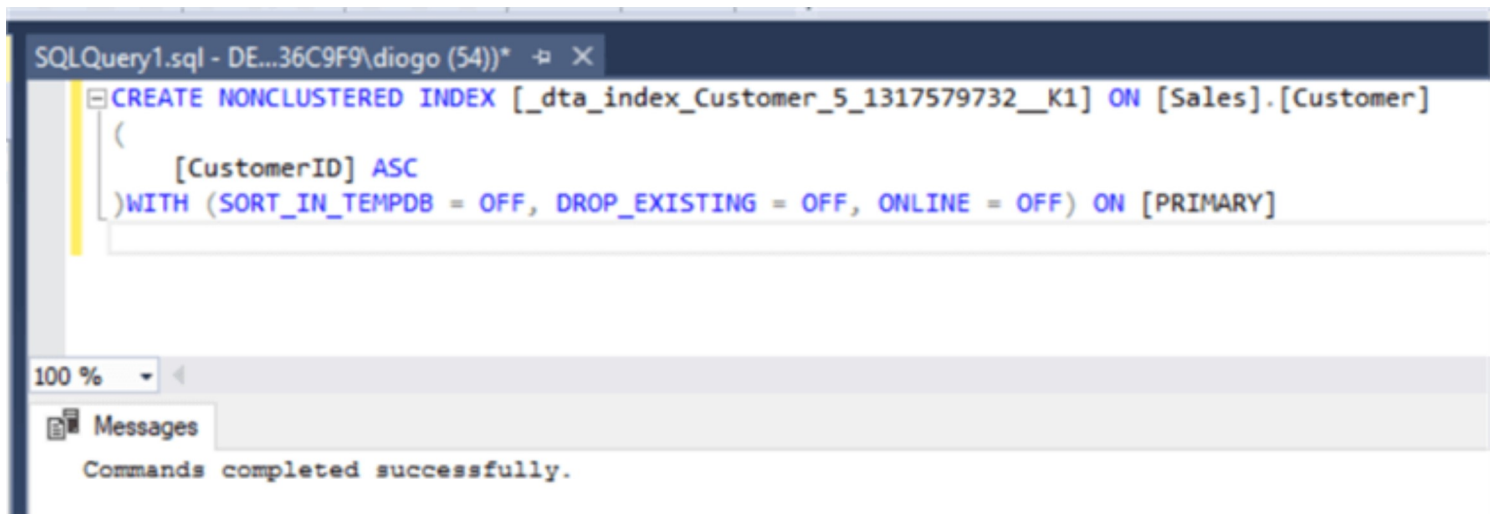
Statement Id	Statement String	Percent Improvement	Statement Type
1	-- sales for group of customers by year	33.07	Select

In order to check each improvement, go to the "Definition" column and click in one of them. Let's try the first one, related to an index recommendation:





Then, go back to SQL Server Management Studio and run the recommended script that'll create a new non-clustered index to the CustomerID column, just like:



When finished, let's run the Tuning Advisor once again, go to the *Reports* tab and check the "Statement cost report" again:

Tuning Reports			
Select report:		Statement cost report	
Statement Id	Statement String	Percent Improvement	Statement Type
1	-- sales for group of customers by yea...	32.30	Select

The index creation allowed our select to execute in a more optimized way. If you follow the other recommendations, you can further improve the query and the percent improvement should get smaller with each change you make.



This concludes this article, where we covered a small fraction of the range of information we will have regarding the Database Engine Tuning Advisor and how it helps us to get recommendations based on the workloads we present for performance optimization.

## Next Steps

- Check out some other related content:
  - [Database Engine Tuning Advisor \(/sqlservertutorial/286/database-engine-tuning-advisor/\)](/sqlservertutorial/286/database-engine-tuning-advisor/)
  - [SQL Server Database Engine Tuning Advisor for Performance Tuning \(/sqlservertip/1872/sql-server-database-engine-tuning-advisor-for-performance-tuning/\)](/sqlservertip/1872/sql-server-database-engine-tuning-advisor-for-performance-tuning/)
  - [SQL Server Performance Tuning with Hypothetical Indexes \(/sqlservertip/3246/sql-server-performance-tuning-with-hypothetical-indexes/\)](/sqlservertip/3246/sql-server-performance-tuning-with-hypothetical-indexes/)

Last Updated: 2018-11-12

## About the author



[\(/sqlserverauthor/314/diogo-souza/\)](/sqlserverauthor/314/diogo-souza/) Diogo Souza has been passionate about clean code, data manipulation, software design and development for almost ten years.

[View all my tips \(/sqlserverauthor/314/diogo-souza/\)](/sqlserverauthor/314/diogo-souza/)

### Related Resources

- [More SQL Server DBA Tips... \(/sql-server-dba-resources/\)](/sql-server-dba-resources/)

## Follow

- [Get Free SQL Tips \(/get-free-sql-server-tips/?ref=GetFooterMenu\)](/get-free-sql-server-tips/?ref=GetFooterMenu)
- [Twitter \(https://twitter.com/mssqltips\)](https://twitter.com/mssqltips)
- [LinkedIn \(https://www.linkedin.com/groups/2320891/\)](https://www.linkedin.com/groups/2320891/)
- [Facebook \(https://www.facebook.com/mssqltips/\)](https://www.facebook.com/mssqltips/)
- [Pinterest \(https://www.pinterest.com/mssqltips/\)](https://www.pinterest.com/mssqltips/)
- [RSS \(https://feeds.feedburner.com/MSSQLTips-LatestSqlServerTips\)](https://feeds.feedburner.com/MSSQLTips-LatestSqlServerTips)

## Learning

- [DBAs \(/sql-server-dba-resources/\)](/sql-server-dba-resources/)
- [Developers \(/sql-server-developer-resources/\)](/sql-server-developer-resources/)
- [BI Professionals \(/sql-server-business-intelligence-resources/\)](/sql-server-business-intelligence-resources/)
- [Careers \(/sql-server-professional-development-resources/\)](/sql-server-professional-development-resources/)
- [Today's Tip \(/todays-sql-server-tip/\)](/todays-sql-server-tip/)



- [Tutorials \(/sql-server-tutorials/\)](/sql-server-tutorials/)
- [Webcasts \(/sql-server-webcasts/\)](/sql-server-webcasts/)
- [Whitepapers \(/sql-server-whitepapers/\)](/sql-server-whitepapers/)
- [Tools \(/sql-server-tools/\)](/sql-server-tools/)

## Search

- [Tip Categories \(/sql-server-categories/\)](/sql-server-categories/)
- [Search By TipID \(/search-tip-id/\)](/search-tip-id/)
- [Authors \(/sql-server-mssqltips-authors/\)](/sql-server-mssqltips-authors/)

## Community

### MENU

- [First Timer? \(/learn-more-about-mssqltips/\)](/learn-more-about-mssqltips/)
- [Pictures \(/mssqltips-community/1/\)](/mssqltips-community/1/)
- [Contribute \(/contribute/\)](/contribute/)
- [Event Calendar \(/sql-server-event-list/\)](/sql-server-event-list/)
- [User Groups \(/sql-server-user-groups/\)](/sql-server-user-groups/)
- [Author of the Year \(/mssqltips-author-of-year/\)](/mssqltips-author-of-year/)

## More Info

- [Join \(/get-free-sql-server-tips/?ref=JoinFooterMenu\)](/get-free-sql-server-tips/?ref=JoinFooterMenu)
- [About \(/about/\)](/about/)
- [Copyright \(/copyright/\)](/copyright/)
- [Privacy \(/privacy/\)](/privacy/)
- [Disclaimer \(/disclaimer/\)](/disclaimer/)
- [Feedback \(/feedback/\)](/feedback/)
- [Advertise \(/advertise/\)](/advertise/)

---

Copyright (c) 2006-2020 [Edgewood Solutions, LLC \(https://www.edgewoodsolutions.com\)](https://www.edgewoodsolutions.com) All rights reserved

Some names and products listed are the registered trademarks of their respective owners.