

Custom Solution to Measure SQL Server Query Performance Improvements



Free MSSQLTips Webinar: [Azure SQL Database Performance Monitoring and Optimization](#)

Many organizations have already deployed or plan to deploy databases in the cloud, both in Infrastructure as a Service (IaaS) and Platform as a Service (PaaS) implementations. People who move to the cloud might think everything's completely hands-off, but monitoring and optimization is even more critical because you share resources and have no control over the infrastructure. So, how do you address your SQL Server performance challenges?

Problem

As part of a performance optimization project for one of my customers, I changed multiple indexes and wanted to assess the performance improvement from these changes. The problem was that each index had been used by a number of different queries and identifying those queries, as well as measuring their key performance metrics before/after the index changes required a lot of manual work. As you may know, the [SQL 2016 Query Store](#) feature makes this task much easier, however what do you do if your SQL Server version is earlier than 2016? In this tip I'll show how you can automate this task using custom performance collection tools and reports.

Solution

Overview and Use Case Scenarios

There are many different scenarios which would require assessing SQL Server query performance changes, like:

- I tuned my existing indexes and want to see improvement rates.

Custom Solution to Measure SQL Server Query Performance Improvements



Free MSSQLTips Webinar: [Azure SQL Database Performance Monitoring and Optimization](#)

Many organizations have already deployed or plan to deploy databases in the cloud, both in Infrastructure as a Service (IaaS) and Platform as a Service (PaaS) implementations. People who move to the cloud might think everything's completely hands-off, but monitoring and optimization is even more critical because you share resources and have no control over the infrastructure. So, how do you address your SQL Server performance challenges?

Problem

As part of a performance optimization project for one of my customers, I changed multiple indexes and wanted to assess the performance improvement from these changes. The problem was that each index had been used by a number of different queries and identifying those queries, as well as measuring their key performance metrics before/after the index changes required a lot of manual work. As you may know, the [SQL 2016 Query Store](#) feature makes this task much easier, however what do you do if your SQL Server version is earlier than 2016? In this tip I'll show how you can automate this task using custom performance collection tools and reports.

Solution

Overview and Use Case Scenarios

There are many different scenarios which would require assessing SQL Server query performance changes, like:

- I tuned my existing indexes and want to see improvement rates.

Custom Solution to Measure SQL Server Query Performance Improvements



Free MSSQLTips Webinar: [Azure SQL Database Performance Monitoring and Optimization](#)

Many organizations have already deployed or plan to deploy databases in the cloud, both in Infrastructure as a Service (IaaS) and Platform as a Service (PaaS) implementations. People who move to the cloud might think everything's completely hands-off, but monitoring and optimization is even more critical because you share resources and have no control over the infrastructure. So, how do you address your SQL Server performance challenges?

Problem

As part of a performance optimization project for one of my customers, I changed multiple indexes and wanted to assess the performance improvement from these changes. The problem was that each index had been used by a number of different queries and identifying those queries, as well as measuring their key performance metrics before/after the index changes required a lot of manual work. As you may know, the [SQL 2016 Query Store](#) feature makes this task much easier, however what do you do if your SQL Server version is earlier than 2016? In this tip I'll show how you can automate this task using custom performance collection tools and reports.

Solution

Overview and Use Case Scenarios

There are many different scenarios which would require assessing SQL Server query performance changes, like:

- I tuned my existing indexes and want to see improvement rates.

Custom Solution to Measure SQL Server Query Performance Improvements



Free MSSQLTips Webinar: [Azure SQL Database Performance Monitoring and Optimization](#)

Many organizations have already deployed or plan to deploy databases in the cloud, both in Infrastructure as a Service (IaaS) and Platform as a Service (PaaS) implementations. People who move to the cloud might think everything's completely hands-off, but monitoring and optimization is even more critical because you share resources and have no control over the infrastructure. So, how do you address your SQL Server performance challenges?

Problem

As part of a performance optimization project for one of my customers, I changed multiple indexes and wanted to assess the performance improvement from these changes. The problem was that each index had been used by a number of different queries and identifying those queries, as well as measuring their key performance metrics before/after the index changes required a lot of manual work. As you may know, the [SQL 2016 Query Store](#) feature makes this task much easier, however what do you do if your SQL Server version is earlier than 2016? In this tip I'll show how you can automate this task using custom performance collection tools and reports.

Solution

Overview and Use Case Scenarios

There are many different scenarios which would require assessing SQL Server query performance changes, like:

- I tuned my existing indexes and want to see improvement rates.

Custom Solution to Measure SQL Server Query Performance Improvements



Free MSSQLTips Webinar: [Azure SQL Database Performance Monitoring and Optimization](#)

Many organizations have already deployed or plan to deploy databases in the cloud, both in Infrastructure as a Service (IaaS) and Platform as a Service (PaaS) implementations. People who move to the cloud might think everything's completely hands-off, but monitoring and optimization is even more critical because you share resources and have no control over the infrastructure. So, how do you address your SQL Server performance challenges?

Problem

As part of a performance optimization project for one of my customers, I changed multiple indexes and wanted to assess the performance improvement from these changes. The problem was that each index had been used by a number of different queries and identifying those queries, as well as measuring their key performance metrics before/after the index changes required a lot of manual work. As you may know, the [SQL 2016 Query Store](#) feature makes this task much easier, however what do you do if your SQL Server version is earlier than 2016? In this tip I'll show how you can automate this task using custom performance collection tools and reports.

Solution

Overview and Use Case Scenarios

There are many different scenarios which would require assessing SQL Server query performance changes, like:

- I tuned my existing indexes and want to see improvement rates.

Custom Solution to Measure SQL Server Query Performance Improvements



Free MSSQLTips Webinar: [Azure SQL Database Performance Monitoring and Optimization](#)

Many organizations have already deployed or plan to deploy databases in the cloud, both in Infrastructure as a Service (IaaS) and Platform as a Service (PaaS) implementations. People who move to the cloud might think everything's completely hands-off, but monitoring and optimization is even more critical because you share resources and have no control over the infrastructure. So, how do you address your SQL Server performance challenges?

Problem

As part of a performance optimization project for one of my customers, I changed multiple indexes and wanted to assess the performance improvement from these changes. The problem was that each index had been used by a number of different queries and identifying those queries, as well as measuring their key performance metrics before/after the index changes required a lot of manual work. As you may know, the [SQL 2016 Query Store](#) feature makes this task much easier, however what do you do if your SQL Server version is earlier than 2016? In this tip I'll show how you can automate this task using custom performance collection tools and reports.

Solution

Overview and Use Case Scenarios

There are many different scenarios which would require assessing SQL Server query performance changes, like:

- I tuned my existing indexes and want to see improvement rates.

Custom Solution to Measure SQL Server Query Performance Improvements



Free MSSQLTips Webinar: [Azure SQL Database Performance Monitoring and Optimization](#)

Many organizations have already deployed or plan to deploy databases in the cloud, both in Infrastructure as a Service (IaaS) and Platform as a Service (PaaS) implementations. People who move to the cloud might think everything's completely hands-off, but monitoring and optimization is even more critical because you share resources and have no control over the infrastructure. So, how do you address your SQL Server performance challenges?

Problem

As part of a performance optimization project for one of my customers, I changed multiple indexes and wanted to assess the performance improvement from these changes. The problem was that each index had been used by a number of different queries and identifying those queries, as well as measuring their key performance metrics before/after the index changes required a lot of manual work. As you may know, the [SQL 2016 Query Store](#) feature makes this task much easier, however what do you do if your SQL Server version is earlier than 2016? In this tip I'll show how you can automate this task using custom performance collection tools and reports.

Solution

Overview and Use Case Scenarios

There are many different scenarios which would require assessing SQL Server query performance changes, like:

- I tuned my existing indexes and want to see improvement rates.

Custom Solution to Measure SQL Server Query Performance Improvements



Free MSSQLTips Webinar: [Azure SQL Database Performance Monitoring and Optimization](#)

Many organizations have already deployed or plan to deploy databases in the cloud, both in Infrastructure as a Service (IaaS) and Platform as a Service (PaaS) implementations. People who move to the cloud might think everything's completely hands-off, but monitoring and optimization is even more critical because you share resources and have no control over the infrastructure. So, how do you address your SQL Server performance challenges?

Problem

As part of a performance optimization project for one of my customers, I changed multiple indexes and wanted to assess the performance improvement from these changes. The problem was that each index had been used by a number of different queries and identifying those queries, as well as measuring their key performance metrics before/after the index changes required a lot of manual work. As you may know, the [SQL 2016 Query Store](#) feature makes this task much easier, however what do you do if your SQL Server version is earlier than 2016? In this tip I'll show how you can automate this task using custom performance collection tools and reports.

Solution

Overview and Use Case Scenarios

- There are many different scenarios which would require assessing SQL Server query performance changes, like:
 - I tuned my existing indexes and want to see improvement rates.

Custom Solution to Measure SQL Server Query Performance Improvements



Free MSSQLTips Webinar: [Azure SQL Database Performance Monitoring and Optimization](#)

Many organizations have already deployed or plan to deploy databases in the cloud, both in Infrastructure as a Service (IaaS) and Platform as a Service (PaaS) implementations. People who move to the cloud might think everything's completely hands-off, but monitoring and optimization is even more critical because you share resources and have no control over the infrastructure. So, how do you address your SQL Server performance challenges?

Problem

As part of a performance optimization project for one of my customers, I changed multiple indexes and wanted to assess the performance improvement from these changes. The problem was that each index had been used by a number of different queries and identifying those queries, as well as measuring their key performance metrics before/after the index changes required a lot of manual work. As you may know, the [SQL 2016 Query Store](#) feature makes this task much easier, however what do you do if your SQL Server version is earlier than 2016? In this tip I'll show how you can automate this task using custom performance collection tools and reports.

Solution

Overview and Use Case Scenarios

There are many different scenarios which would require assessing SQL Server query performance changes, like:

- I tuned my existing indexes and want to see improvement rates.