

SQL Optimizer for SQL Server® 10.1.2

User Guide



Copyright

Copyright 2020 Quest Software Inc.
ALL RIGHTS RESERVED.

This guide contains proprietary information protected by copyright. The software described in this guide is furnished under a software license or nondisclosure agreement. This software may be used or copied only in accordance with the terms of the applicable agreement. No part of this guide may be reproduced or transmitted in any form or by any means, electronic or mechanical, including photocopying and recording for any purpose other than the purchaser's personal use without the written permission of Quest Software Inc.

The information in this document is provided in connection with Quest Software products. No license, express or implied, by estoppel or otherwise, to any intellectual property right is granted by this document or in connection with the sale of Quest Software products. EXCEPT AS SET FORTH IN THE TERMS AND CONDITIONS AS SPECIFIED IN THE LICENSE AGREEMENT FOR THIS PRODUCT, QUEST SOFTWARE ASSUMES NO LIABILITY WHATSOEVER AND DISCLAIMS ANY EXPRESS, IMPLIED OR STATUTORY WARRANTY RELATING TO ITS PRODUCTS INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTY OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, OR NON-INFRINGEMENT. IN NO EVENT SHALL QUEST SOFTWARE BE LIABLE FOR ANY DIRECT, INDIRECT, CONSEQUENTIAL, PUNITIVE, SPECIAL OR INCIDENTAL DAMAGES (INCLUDING, WITHOUT LIMITATION, DAMAGES FOR LOSS OF PROFITS, BUSINESS INTERRUPTION OR LOSS OF INFORMATION) ARISING OUT OF THE USE OR INABILITY TO USE THIS DOCUMENT, EVEN IF QUEST SOFTWARE HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES. Quest Software makes no representations or warranties with respect to the accuracy or completeness of the contents of this document and reserves the right to make changes to specifications and product descriptions at any time without notice. Quest Software does not make any commitment to update the information contained in this document.

If you have any questions regarding your potential use of this material, contact:

Quest Software Inc.
Attn: LEGAL Dept
4 Polaris Way
Aliso Viejo, CA 92656

Refer to our Web site (<https://www.quest.com>) for regional and international office information.

Patents

SQL Optimizer for SQL Server is protected by U.S. Patents # 8,332,346 and 8,499,001. Additional patents pending. For the most current information about applicable patents for this product, please visit our website at <https://www.quest.com/legal>.

Trademarks

Quest, Quest Software, Benchmark Factory, Foglight, Spotlight, Toad, and the Quest logo are trademarks of Quest Software Inc. in the U.S.A. and other countries. For a complete list of Quest Software trademarks, please visit our website at <https://www.quest.com/legal>. Microsoft, Windows, Windows Server, SQL Server, Excel, and Windows Vista are either registered trademarks or trademarks of Microsoft Corporation in the United States and/or other countries. Citrix and XenApp are trademarks of Citrix Systems, Inc. and/or one or more of its subsidiaries, and may be registered in the United States Patent and Trademark Office and in other countries. Other trademarks are property of their respective owners.

Legend

i **IMPORTANT, NOTE, TIP, MOBILE, VIDEO:** An information icon indicates supporting information.

! **CAUTION:** A CAUTION indicates potential damage to hardware or loss of data if instructions are not followed.

Contents

Copyright	2
Contents	3
Introduction	4
SQL Optimization Workflow	4
Identify Problematic SQL	5
Tutorial: SQL Scanner	5
View SQL Classification	6
Send to Optimize SQL	8
Tutorial: Find SQL from Plan Cache	9
Tutorial: Find SQL from SQL Trace	10
SQL Trace Templates	12
Optimize SQL Statements and Execute Alternatives	14
Tutorial: SQL Optimizer (SQL Rewrite)	14
Step 1: Optimize the SQL Statement	14
Step 2: Benchmark Alternative SQL Statements	15
Auto Optimize SQL Statements	15
Auto Optimize Using Plan Guide	16
Test Run SQL Alternatives	17
About Generating Index Alternatives	17
Index Generation for SQL Server 2005 or later	18
Index Generation for SQL Server 2000	18
Tutorial: Index Alternatives	18
Tutorial: User-Defined Temp Tables	19
Compare SQL Alternatives	21
Compare SQL Statements	21
Generate Reports	23
Generate Scanned Jobs Reports	23
Generate Group Summary Reports	23
Generate Optimize SQL Resolution Report	24
Tutorial: Optimize Indexes	25
About Us	27
Contact Quest	27
Technical Support Resources	27
SQL Optimizer Community	27

Introduction

SQL Optimization Workflow

SQL Optimizer for SQL Server® uses the following SQL optimization workflow to help you ensure that your SQL statements perform optimally in your database environment.

Procedure	Description
Identify Problematic SQL Statements	<p>The SQL Scanner locates and extracts problematic SQL statements directly from your source code without execution. SQL Scanner analyzes the operations of the execution plan and identifies potential performance bottlenecks. See "About Scanning SQL" in the online help for more information.</p> <p>i Note: You need to use Microsoft® SQL Server Profiler to capture dynamic SQL statements. Save the captured dynamic SQL statements into a trace file or trace table and use SQL Scanner to extract the statements.</p>
Optimize SQL Statements	<p>The SQL Optimizer uses artificial intelligence to generate alternatives with unique execution plans for problematic SQL statements. SQL Optimizer generates the alternatives by analyzing SQL statement syntax and database structure. You can also use hints during the optimization process. See "Optimize SQL Statements" in the online help for more information.</p>
Test Run SQL Alternatives	<p>The Test Run function tests the performance of each alternative SQL Optimizer generates. This provides execution times for each alternative to determine the best statement for your database environment. See "Execute SQL Alternatives" in the online help for more information.</p>
Compare SQL Alternatives	<p>The Compare feature displays details for two SQL statements side-by-side. You can compare an alternative to the original SQL statement or compare two alternatives. See "Compare SQL Statements" in the online help for more information.</p>
Generate Reports	<p>The Report feature creates a SQL resolution report after you complete the optimization process. See "Generate Optimize SQL Resolution Report" in the online help for more information.</p>

Identify Problematic SQL

Tutorial: SQL Scanner

SQL Scanner extracts SQL statements embedded in database objects, captured from SQL Profiler, or stored in application source code and binary files. SQL Scanner retrieves and analyzes execution plans for extracted SQL statements from SQL Server to identify statements with performance bottlenecks. You can copy statements classified as problematic (first priority) or complex (second priority) into SQL Optimizer for analysis.

To scan SQL

1. Select the Scan SQL tab in the main window.
2. Select **Welcome** from the Scanner Explorer.
3. Click **Add Scanner Job**. The Add Scanner Jobs window displays.
4. Select a connection to use. Review the following for additional information:

Select Connection	Description
Connection	Click  to select a previously created connection. i Tip: Click  to open the Connection Manager to create a new connection.
Select Database and User	Description
Database	Click  to select the database to match your SQL statement.
Set User	Click  to select your user name.

5. Select the page for the item you want to scan in the Add Scanner Jobs pane. Review the following for additional information:

Database Objects Page	Description
Database Objects	Select a database object and click  to add the object to scan. i Tip: Click  to browse for database objects to add.
SQL Profiler Page	Description

Available trace files/table	Click  to add SQL Server Profiler trace files or trace tables to scan.
Database	Click  to select the database of the SQL to scan.
Set User	Click  to select your user name.
Source Code Page	Description
Source code type	Select Text/Binary files or COBOL programming source code to indicate the source code type.
Add by file	Click  and browse to the files you want to scan.
Add by directory	Click  and browse to the directories you want to scan.  Note: Select the Include Sub-directory checkbox to scan sub-directories.
Connection for scanning	Select the database and user name settings.
 Note: SQL Scanner helps you manage scan jobs by organizing them into groups. Use the Group Information page to create a new group or to add the current scan job to an existing group.	

- Click **Finish** to start SQL Scanner.
- Select **Task** from the Scanner Explorer.
- Select Scanner in the Task pane to view information about your scan jobs.

The Group Summary pane sorts information about your scan jobs by groups. Additional information displays in the Type of Jobs and Job Status panes.

- Select a group from the scanner node to see details for the group in the Job List pane.

Details displayed in the Job List pane include the number of SQL statements found and the classification for each statement. Additional information displays in the SQL Classification and Job Detail Information panes.

- Select a scan job from the group node to see details for the scan job.

The SQL List pane displays SQL statements identified by classification. Selecting a SQL statement in the SQL List pane displays information about the statement in the SQL Text and Execution Plan panes.

View SQL Classification

SQL Scanner extracts syntactically correct SQL statements and retrieves their execution plans. SQL Scanner then classifies the extracted statements as Simple, Complex, Problematic, or Invalid. You can view the classification of extracted statements by scan jobs or individually.

To view SQL classification for a scan job

- Select the Scan SQL tab in the main window.
- Create a scan job using the Add Scanner Jobs wizard. See "Scan SQL" in the online help for more information.

3. Select a group in the Scanner node in the Task pane.
4. Select a scan job in the Job List pane to view classification information in the SQL Classification pane. Review the following for additional information:

Classification	Description
Simple	SQL statements are classified as simple when the number of tables referenced in the execution plan is less than the lower limit of the complex table scan operations range.
Complex	SQL statements are classified as complex when the number of tables referenced in the execution plan exceeds the lower limit of the complex table scan operations range.
Problematic	SQL statements are classified as problematic when SQL Optimizer determines they can be optimized. Problematic SQL statements satisfy one or more of the following criteria: <ul style="list-style-type: none"> • The number of tables referenced in the execution plan exceeds the upper limit of the complex table scan operations range. • A full table scan executed in a nested loop exceeds the user-defined threshold table size. • A single table scan executed exceeds the user-defined threshold table size.
Invalid	SQL statements are classified as invalid for one of the following reasons: <ul style="list-style-type: none"> • The current user does not have privileges to use the tables, views, or other database objects referenced in the SQL statement. • The database and/or user do not match the database and user (object owner) for the SQL statement. • The SQL statement is dynamically constructed but is not a Single Command Line Dynamic (SCLD) SQL statement. SQL Optimizer can only identify SCLD SQL statements. • In Scan SQL, the SQL statement uses a temporary table that was not created before starting the Scan SQL session. See "Use Temporary Tables in Scan SQL" in the online help for more information.
<p> Note: You can specify SQL classification rules in the Options dialog. See "SQL Classification Options" in the online help for more information.</p>	

To view SQL classification for individual statements

1. Select a group in the Scanner node in the Task pane.
2. Select a scan job from the Group node.
3. Review the list of SQL in the SQL List pane.

 **Tips:**

- Click  to group the SQL by classification category.
- Click  to list the SQL numerically.

4. Select a SQL statement in the list to view SQL text.

Send to Optimize SQL

You can optimize SQL statements extracted by SQL Scanner using SQL Optimizer. SQL Optimizer supports single INSERT, UPDATE, SELECT, DELETE, or MERGE SQL statements.

To send a statement to SQL Optimizer

1. Select the Scan SQL tab in the main window.
2. Create a scan job using the Add Scanner Jobs wizard. See "Scan SQL" in the online help for more information.
3. Select a group in the Scanner node in the Task pane.
4. Select a scan job from the Group node.
5. Select the SQL statement in the Job List pane.
6. Click  **Send to Optimize SQL by SQL Rewrite** in the SQL Text pane.

i Notes:

- If the selected SQL statement uses a temporary table and SQL Scanner finds a CREATE TABLE or SELECT INTO statement, the User-Defined Temp Table window displays automatically. See "Use Temporary Tables in Scan SQL" in the online help for more information.
- If the selected SQL statement is used within a cursor declaration, the Cursor Settings window displays automatically. See "Optimize SQL Statements" in the online help for more information.
- If the selected SQL statement uses a variable, SQL Scanner automatically assigns the CHAR datatype to the variable when it retrieves the execution plan. To assign a different datatype, click  after you send the statement to SQL Optimizer and select the new datatype in the Set Bind Variables window that displays. See "Set Bind Variables" in the online help for more information.

Tutorial: Find SQL from Plan Cache

Find SQL helps you locate the most resource-intensive SQL in your server. It evaluates existing SQL performance by different statistics (such as CPU time and elapsed time) from different areas (such as batches and database objects).

i Note: The Find SQL from Plan Cache module supports only SQL Server 2005 (Service Pack 2) or later.

Additionally, you can:

- Extract from Plan Cache any SQL executed in your server
- Review execution statistics and query plans of the SQL
- Check resource consumption at SQL, Batch, and Database Object levels
- Optimize the performance of your SQL by sending them to Optimize SQL or Optimize Indexes
- Save a SQL and its XML plan for others to review

To create a new Plan Cache search session

1. Select **Find SQL**.

i Note: If you are creating your first search session, select **Click here to start find SQL** and select a connection. Then select **Click to start finding SQL in Plan Cache**.

2. To conduct a search on a server for which you have previously conducted a search, select the server from the list. Or click **Show all servers** to view entire list and select the server.
 - Then click **Start a new Plan Cache search**.
 - Or select **Click to start finding SQL in Plan Cache** if you have never conducted a Plan Cache search on this server.
3. To conduct a search on a new server, click **Find SQL in another server**.
 - a. Select a connection.
 - b. Select **Click to start finding SQL in Plan Cache** in the *Find SQL from Plan Cache* panel that displays.
4. The Add Plan Cache Search dialog opens. Select criteria for collecting SQL. Review the following for additional information:

Criteria to collect SQL	Description
Databases	Click the link and select from the following options: <ul style="list-style-type: none">• All (excluding system databases)• All (including system databases)• Specific databases only—Select the databases to include in the search.
Select the number of SQL to retrieve	Click the link to specify the number of SQL and the retrieval method.

	<p>i Note: The second option allows you to specify the number of SQL according to a specific criterion.</p>
--	--

- Click **OK**. The session name is added to the list of saved sessions and the search process starts.
- A search progress bar displays next to the search name during the search process. The following information and options are displayed in the Find SQL from Plan Cache pane.

Start a new search	Select to begin a new search and to display the Criteria to collect SQL dialog box.
Search name	Name you have given to your search. Click to open the dashboard-style Search Result page for this search. i Notes: <ul style="list-style-type: none"> The default format is the <i>current date and time</i>. Right-click the name to rename or delete the search.
Status	Displays status of search: <ul style="list-style-type: none"> Collecting i Note: Click  to abort search. Complete
Show all Plan Cache searches in this server	Select to display all the searches you have conducted on the selected server. i Note: Right-click the search to delete the selected search.

- When the search is finished, a dashboard-style Search Results page displays.
Click a pie chart to view the top-consuming SQL for that resource. Each pie chart represents a different view of the most-expensive (top-consuming) SQL. In each view, the SQL statements are filtered by a different resource consumed and by a different SQL location.
- After viewing top-consuming SQL on the Summary Chart page, to return to the dashboard, click  or click the session name in the breadcrumb.
- For more information about reviewing Find SQL results, see the online Help.

Tutorial: Find SQL from SQL Trace

Find SQL from SQL Trace allows you to extract SQL statements and their performance statistics from trace files or trace tables collected by SQL Profiler. You can identify the most resource-intensive SQL statements in your traces.

Additionally, you can:

- Extract SQL executed in your server through SQL Trace
- Review execution statistics and query plans of the SQL
- Check resource consumption in SQL, Batch and Database Object levels

- Optimize the performance of your SQL by sending them to Optimize SQL or Optimize Indexes
- Save a SQL and its XML plan for others to review

To create a new SQL Trace search session

1. Select **Find SQL**.
 - i Note:** If you are creating your first search session, select **Click here to start find SQL** and select a connection. Then select **Click to start finding SQL in SQL Trace**.
2. To conduct a search on a server for which you have previously conducted a search, select the server from the list. Or click **Show all servers** to view entire list and select the server.
 - Then click **Start a new SQL Trace search**.
 - Or select **Click to start finding SQL in SQL Trace** if you have never conducted a SQL Trace search on this server.
3. To conduct a search on a new server, click **Find SQL in another server**.
 - a. Select a connection.
 - b. Select **Click to start finding SQL in SQL Trace** in the *Find SQL from SQL Trace* panel that displays.
4. The Add SQL Trace Search dialog opens. Select a trace file or table. Review the following for more information:
 - i Notes:**
 - You can extract SQL statements and their performance statistics from trace files or trace tables.
 - You can use trace templates provided by SQL Optimizer or use your own templates.

Add from Trace Files	Click to add SQL from trace files.
Add from Trace Tables	<p>Click to add SQL from trace tables. The Add Trace Table dialog opens. Select a connection.</p> <ul style="list-style-type: none"> • Connection—Select the connection where the trace table is located. • Database—Specify the database where the trace table is located. • Schema—Specifies the owner of the trace table. <p>Then select a trace table.</p> <p>i Note: Find SQL requires certain event columns in your trace in order to analyze it. For convenience, SQL Optimizer provides trace templates for you to use. These template files are located in the Find SQL - Trace Templates folder in the SQL Optimizer installation directory.</p> <p>See SQL Trace Templates on page 12 for more information.</p>
Show Collect Criteria	<p>Click to filter the SQL to collect. You can filter by:</p> <ul style="list-style-type: none"> • Application Name • Database Name • Login Name • Host Name

- Click **OK**. The session name is added to the list of saved sessions and the search process starts.
- A search progress bar displays next to the search name during the search process. The following information and options are displayed in the Find SQL from SQL Trace pane.

Start a new search	Select to begin a new search and to display the Add SQL Trace Search dialog box.
Search name	Name you have given to your search. Click to open the dashboard-style Search Result page for this search. i Notes: <ul style="list-style-type: none"> The default format is the <i>current date and time</i>. Right-click the name to rename or delete the search.
Status	Displays status of search: <ul style="list-style-type: none"> Collecting i Note: Click  to abort search. Complete
Show all SQL Trace searches in this server	Select to display all the searches you have conducted on the selected server. i Note: Right-click the search to delete the selected search.

- When the search is finished, a dashboard-style Search Results page displays.
Click a pie chart to view the top-consuming SQL for that resource. Each pie chart represents a different view of the most-expensive (top-consuming) SQL. In each view, the SQL statements are filtered by a different resource consumed and by a different SQL location.
- After viewing top-consuming SQL on a Summary Graph page, to return to the dashboard click  or click the Search session name in the breadcrumb.
- For more information about reviewing Find SQL results, see the online Help.

SQL Trace Templates

Find SQL requires certain event columns in your trace in order to analyze it. For your convenience, SQL Optimizer provides trace template files (.tdf) for you to use. The template files are located in the **Find SQL - Trace Templates** folder in the SQL Optimizer installation directory.

If you create your own template, make sure it defines the events and columns listed in the following table.

Methods for setting up SQL Trace	Description
Use trace templates provided by SQL Optimizer	Use trace templates provided by SQL Optimizer to collect SQL. You can locate the templates in the installation directory.

Methods for setting up SQL Trace	Description																								
Use your own templates	<p data-bbox="571 315 1283 344">Ensure that the following trace events and columns are captured.</p> <table border="1" data-bbox="571 365 1382 1227"> <thead> <tr> <th data-bbox="571 365 963 421">Events</th> <th data-bbox="963 365 1382 421">Columns</th> </tr> </thead> <tbody> <tr> <td data-bbox="571 421 963 477">RPC: Completed</td> <td data-bbox="963 421 1382 477">EventClass</td> </tr> <tr> <td data-bbox="571 477 963 533">RPC: Starting</td> <td data-bbox="963 477 1382 533">TextData</td> </tr> <tr> <td data-bbox="571 533 963 589">SP:Completed</td> <td data-bbox="963 533 1382 589">CPU</td> </tr> <tr> <td data-bbox="571 589 963 645">SP:Starting</td> <td data-bbox="963 589 1382 645">Writes</td> </tr> <tr> <td data-bbox="571 645 963 701">SP:StmtCompleted</td> <td data-bbox="963 645 1382 701">Reads</td> </tr> <tr> <td data-bbox="571 701 963 757">SQL:BatchCompleted</td> <td data-bbox="963 701 1382 757">Duration</td> </tr> <tr> <td data-bbox="571 757 963 813">SQL:BatchStarting</td> <td data-bbox="963 757 1382 813">SPID</td> </tr> <tr> <td data-bbox="571 813 963 1227" rowspan="7">SQL:StmtCompleted</td> <td data-bbox="963 813 1382 880">DatabaseName</td> </tr> <tr> <td data-bbox="963 880 1382 947">EventSequence</td> </tr> <tr> <td data-bbox="963 947 1382 1014">LineNumber</td> </tr> <tr> <td data-bbox="963 1014 1382 1081">ObjectID</td> </tr> <tr> <td data-bbox="963 1081 1382 1149">ObjectName</td> </tr> <tr> <td data-bbox="963 1149 1382 1216">Offset</td> </tr> <tr> <td data-bbox="963 1216 1382 1227">SourceDatabaseID</td> </tr> </tbody> </table>	Events	Columns	RPC: Completed	EventClass	RPC: Starting	TextData	SP:Completed	CPU	SP:Starting	Writes	SP:StmtCompleted	Reads	SQL:BatchCompleted	Duration	SQL:BatchStarting	SPID	SQL:StmtCompleted	DatabaseName	EventSequence	LineNumber	ObjectID	ObjectName	Offset	SourceDatabaseID
Events	Columns																								
RPC: Completed	EventClass																								
RPC: Starting	TextData																								
SP:Completed	CPU																								
SP:Starting	Writes																								
SP:StmtCompleted	Reads																								
SQL:BatchCompleted	Duration																								
SQL:BatchStarting	SPID																								
SQL:StmtCompleted	DatabaseName																								
	EventSequence																								
	LineNumber																								
	ObjectID																								
	ObjectName																								
	Offset																								
	SourceDatabaseID																								

Optimize SQL Statements and Execute Alternatives

Tutorial: SQL Optimizer (SQL Rewrite)

Using SQL Rewrite mode in Optimize SQL consists of two steps. In the first step, SQL Optimizer generates semantically equivalent alternatives with unique execution plans for your original SQL statement. A SQL Server cost estimate displays for each alternative generated. In the second step, SQL Optimizer executes the alternatives to benchmark each statement's performance. This provides execution times and run time statistics that allow you to find the best SQL statement for your database environment.

i Tip: SQL Server cost only provides an estimate of resource usage to execute a SQL statement. Since statements with higher cost may perform better, you should test alternatives generated to determine the best statements for your database environment.

Step 1: Optimize the SQL Statement

1. Select the Optimize SQL tab in the main window.
2. Select SQL Rewrite from the Optimize SQL start page.

i Note: If the start page does not display, click the arrow beside  and select **New SQL Rewrite Session**.

3. Enter a SQL statement in the Alternative Details pane.
4. Click . The Select Connection window displays.
5. Select a connection. Review the following for additional information:

Select Connection	Description
Connection	Click  to select a previously created connection. i Tip: Click  to open the Connection Manager to create a new connection.
Select Database and User	Description
Database	Click  to select the database to match your SQL statement.
Set User	Click  to select your user name.

6. Optimize SQL to use in a cursor. SQL Server generates different execution plans for SQL statements embedded in a cursor declaration. This needs to be considered when using SQL Optimizer to retrieve execution plans, retrieve run times, and generate SQL alternatives. Specific cursor settings need to be selected before SQL statements that come from or will be embedded in a cursor declaration. These settings tell SQL Optimizer to use cursor simulation when retrieving execution plans and run time information.

Complete the following steps to select cursor settings:

- a. Select the **Optimize for Cursor** checkbox.
 - b. Click **Settings**.
 - c. Select the cursor type and the associated cursor argument.
7. Click  to optimize the SQL statement or click  to optimize the SQL statement and generate index alternatives in one step.

You can configure hints and other optimization settings, such as temporary table generation and ANSI JOIN syntax, in the Options pages before performing this step. The Intelligence Level that determines the number of SQL Server hints applied and the number of alternatives generated can also be configured from the Options pages.

i Note: You can create a temporary table for your SQL statement. See "Use Temporary Tables in Scan SQL" in the online help for more information.

8. Click  after SQL Optimizer completes the SQL rewrite process to compare your original SQL statement with the alternatives generated.

Step 2: Benchmark Alternative SQL Statements

The Test Run function provides an efficient way to benchmark alternatives generated by SQL Optimizer. You can execute selected alternatives to obtain actual execution statistics. This function does not affect network traffic since SQL Optimizer can provide these statistics without having to retrieve result sets from the database server. Additionally, data consistency is maintained when using SELECT, SELECT INTO, INSERT, DELETE, and UPDATE statements because these statements are run in a transaction that is rolled back after execution.

To benchmark a SQL statement alternative

1. Click the drop-down arrow beside  and select **Test Run All**.
2. The Test Run Settings dialog opens. Select test run options to apply to this test run.
 - To allow SQL Optimizer to determine the best test run settings, answer the questions in the Test Run Settings dialog and click **Start Test Run**.
 - To customize test run settings, click **Customize Test Run Settings** at the bottom of the page and specify options for this test run.
3. The execution statistics display in the Alternatives pane once the test run is finished. Select an alternative to see more information in the Alternative Details and Execution Plan panes.

Auto Optimize SQL Statements

Use the Auto Optimize function to perform the optimization and testing processes simultaneously. The function optimizes your original SQL statement by generating alternatives and then starts testing once SQL Optimizer generates the first alternative.

The Auto Optimize function reduces the time required to find the best alternative because it begins the testing process early instead of waiting until SQL Optimizer generates all alternatives. You can stop the Auto Optimize function once you find a satisfactory SQL statement alternative.

To automatically optimize a SQL statement

1. Select the Optimize SQL tab in the main window.
2. Select SQL Rewrite from the Optimize SQL start page.
3. Enter a SQL statement in the Alternative Details pane.
4. Click .
5. If you did not already select a connection for this session, the Select Connection dialog opens. Select a connection.
6. After you select a connection, the Test Run Settings dialog opens. Specify test run options.
7. When the optimization and testing process is finished, the execution statistics display in the Alternatives pane. Select an alternative to view details in the Alternative Details pane and the Execution Plan pane.
8. Select the Compare tab to compare SQL alternatives to the original SQL.

i Tips:

- Click  to stop the optimization and testing processes.
- You can send SQL statements from SQL Optimizer back to Toad for SQL Server after the optimization process is finished. Click  to send a SQL statement back to Toad from SQL Optimizer.

Auto Optimize Using Plan Guide

Use Auto Optimize to generate plan alternatives and then test run the alternatives all in one process.

To automatically optimize using plan guide

1. Select the Optimize SQL tab in the main window.
 2. Select Plan Control from the Optimize SQL start page.
 3. Enter a SQL statement in the Original SQL pane.
 4. Click .
- i Tip:** Click  to abort the process.
5. If you did not already select a connection for this session, the Select Connection dialog opens. Select a connection.
 6. After you select a connection, the Test Run Settings dialog opens. Specify test run options.
 7. When the optimization and testing process is finished, the execution statistics display in the Plans pane. Select a plan alternative to view details in the Execution Plan pane.
 8. Select the Compare tab to compare plan alternatives to the original plan.
 9. Select the Deploy tab to deploy a plan to the Manage Plan Guides module.

Test Run SQL Alternatives

After SQL Optimizer generates SQL alternatives, you can test run the alternatives. Test run statement alternatives to view their execution statistics. You can test run a single statement, a selected group of statements, or all statements simultaneously.

i Note: SQL Optimizer test runs the statements against the database during this process to obtain accurate run-time statistics. SQL Optimizer automatically rolls back any changes made to the database after it executes SELECT, SELECT INTO, INSERT, DELETE, or UPDATE statements.

To test run SQL alternatives

1. After generating SQL alternatives in a SQL Rewrite session, use one of the following methods to test run the SQL alternatives.
 - To test run a single SQL alternative, select the alternative and click .
 - To test run a single SQL alternative and specify test run criteria, select the alternative and click the arrow beside  and select **Test Run Special - Current**.
 - To test run multiple alternatives simultaneously, select the alternatives and click the arrow beside  and select **Test Run Selected**.
 - To test run all alternatives simultaneously, click the arrow beside  and select **Test Run All**.
2. If you selected to test run a single alternative using special test run criteria, the Test Run Special Settings dialog opens. Specify criteria for this test run only.
3. If you selected to test run multiple or all alternatives simultaneously, the Test Run Settings dialog opens. Specify test run options to apply to this test run only.
4. After the test run process is finished, the execution statistics display in the Alternatives pane. Select an alternative to view details in the Alternative Details pane and the Execution Plan pane.
5. Select the Compare tab to compare SQL alternatives to the original SQL.

i Notes:

- Click  to stop the test run process for multiple alternatives.
- To stop the process for a single alternative, select the running plan alternative and click .

To clear the SQL Rewrite window

To clear the results, right-click inside the Alternative Details pane and do one of the following:

- Select **Clear Optimization Results | Keep Original Scenario** to clear only the results and retain both the original SQL statement and the database connection.
- Select **Clear Optimization Results | Clear Original Scenario** to clear the SQL Rewrite window, but retain the database connection.

About Generating Index Alternatives

SQL Optimizer analyzes the following in your original SQL statement and table references to generate index alternatives:

- SQL statement syntax
- Related tables and indexes

- Search arguments
- Table join conditions

Once SQL Optimizer generates alternatives, you can test them to evaluate improvements in database performance.

i NOTE: SQL Optimizer does not physically create indexes on your database when generating alternatives.

Index Generation for SQL Server 2005 or later

When connected to Microsoft SQL Server 2005 or later, SQL Optimizer uses virtual indexes and Database Engine Tuning Advisor (DTA) information to provide index alternatives. SQL Optimizer generates virtual indexes and combines them into index sets. DTA information for the original SQL statement is then retrieved using each index set. If the DTA information indicates the indexes can be used by the SQL statement, the index set is saved. Index sets that cannot not be used are automatically eliminated. This ensures SQL Optimizer only provides index sets that impact performance.

i Tips:

- When SQL Optimizer generates virtual index alternatives, it also creates virtual execution plans for each index set. You can create temporary indexes on your database to retrieve the actual execution plans for the index sets. See "Fill Missing Execution Plans" in the online help for more information.
- Once you retrieve the execution plans, you can use the plan cost to help you select alternatives to test with your database environment. See "Compare Plan Cost" in the online help for more information.

Index Generation for SQL Server 2000

When connected to Microsoft SQL Server 2000, SQL Optimizer provides you with individual index alternatives to evaluate, test or implement. Since SQL Server 2000 cannot create virtual indexes, DTA information is not used to eliminate alternatives that do not impact performance.

Tutorial: Index Alternatives

SQL Optimizer analyzes the following in your original SQL statement and table references to generate index alternatives:

- SQL statement syntax
- Related tables and indexes
- Search arguments
- Table join conditions

Once SQL Optimizer generates alternatives, you can test them to evaluate improvements in database performance.

To generate and benchmark an index alternative

1. Select the Optimize SQL tab in the main window.
2. Enter a SQL statement in the Alternative Details pane of the SQL Details tab.
3. Click . The Select Connection window displays.
4. Review the following for additional information:

Select Connection	Description
Connection	Click  to select a previously created connection. i Tip: Click  to open the Connection Manager to create a new connection.
Select Database and User	Description
Database	Click  to select the database to match your SQL statement.
Set User	Click  to select your user name.

5. Click   to generate index alternatives or click  to generate index alternatives and optimize the SQL statement in one step.
6. Benchmark index alternatives with the original SQL statement. The Test Run function provides an efficient way to test index alternatives. It physically creates the index on the database, runs the SQL statement, and drops the indexes after the test is finished.
 - i** **Important:** Since indexes are physically created on the database, this process may impact database performance and the performance of other SQL statements.
 - a. Select the index alternative to test.
 - b. Click . Execution statistics display in the Alternatives pane once the test is finished. Use the tabs available in the Execution Plan pane to view more information about the index alternative.

Tutorial: User-Defined Temp Tables

When your SQL statement uses a temp table, SQL Optimizer requires you to create the temporary table before optimizing the SQL statement or generating index alternatives. SQL Optimizer automatically drops all temp tables created when you close your session.

To create a temporary table

1. Select the Optimize SQL tab in the main window.
2. Click  at the bottom of the Alternative Details pane. The Temp Table Manager window displays.

3. Enter a SQL statement to create a temporary table. Review the following for additional information:

Temp Table Manager	Description
SQL Script Editor	Allows you to enter SQL statements to create temporary tables. i Tip: Click  to open a file with your SQL statement.
SQL Script	Displays the SQL script for the temporary table you select.
Temp Table List	Displays a list of temporary tables for your current session. i Tip: Click  to drop all temporary tables for your current session.
i Note: If the selected SQL statement uses a variable, the Set Bind Variables window displays so you can define the variable. See "Set Bind Variables" in the online help for more information.	

4. Click . The Select Connection window displays. Review the following for additional information:

Select Connection	Description
Connection	Click  to select a previously created connection. i Tip: Click  to open the Connection Manager to create a new connection.
Select Database and User	Description
Database	Click  to select the database to match your SQL statement.
Set User	Click  to select your user name.

Compare SQL Alternatives

Compare SQL Statements

Use the Compare window to view the SQL text, execution plan, and execution statistics for your original SQL statement and all alternatives SQL Optimizer generated. The Compare window consists of the Alternatives window and the Comparison window. The Alternatives window displays execution statistics and the Comparison window displays SQL statements and execution plans. You can compare your original SQL statement with an alternative SQL Optimizer generated or compare two different alternatives.

i Note: This topic focuses on information that may be unfamiliar to you. It does not include all step and field descriptions.

To compare SQL statements

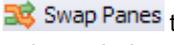
1. Select the Optimize SQL tab in the main window.

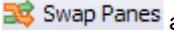
2. Click .

3. Select an alternative in the Alternatives window to compare the statement with your original SQL statement.

i Note: The Comparison window displays the SQL text and execution plan for the original SQL statement in both panes by default.

i Tip: The alternative you selected is shown under your original SQL statement in the Alternatives window by default. To unfreeze the alternative selected, right-click the alternative and select **Unfreeze Comparing Rows**.

4. Click  to switch the location of the original SQL statement and selected alternative in the Comparison window.

i Note: Change the location of the original SQL statement to compare two alternative statements. Once you select the first alternative, click  and select the second alternative.

5. Click the current layout option in the Comparison Window to change the layout for the statements you are comparing. Review the following for additional information:

SQL and Plan (Left-Right)	Displays the SQL text and execution plan for the statements you are comparing side by side.
SQL Only (Left-Right)	Displays the SQL text for the statements you are comparing side by side.
Plan Only (Left-Right)	Displays the execution plan for the statements you are comparing side by side.

SQL and Plan (Top-Bottom)	Displays the SQL text and execution plan for the statements you are comparing beneath each other.
SQL Only (Top-Bottom)	Displays the SQL text for the statements you are comparing beneath each other.
Plan Only (Top-Bottom)	Displays the execution plan for the statements you are comparing beneath each other.
<p>i Tips:</p> <ul style="list-style-type: none"> • Click  Maximize to maximize the comparison window. • Click  Restore to restore the comparison window to its original size. 	

Generate Reports

Generate Scanned Jobs Reports

You can generate a scanned jobs report in SQL Scanner. The scanned jobs report provides information about the SQL statements for a selected scan job that includes:

- Chart for SQL Classification
- SQL conversion
- Temp table dependencies
- SQL position
- SQL Text
- Execution Plan

i Tip: You can display execution plans in text mode, grayscale graphic mode, or full color graphic mode.

- SQL Classification

To generate a scanned jobs report

1. Select the Scan SQL tab in the main window.
2. Create a scan job using the Add Scanner Jobs wizard. See "Scan SQL" in the online help for more information.
3. Select a group in the Scanner node in the Task pane.
4. Select the scan job that you want to create a report for in the Job List pane.
5. Click .
- i Tip:** You can also generate a selected SQL report by selecting a scan job from the Group node of the Task pane and clicking  in the SQL List pane.
6. Select the checkboxes for the criteria you want to include in the report.

Generate Group Summary Reports

You can generate a group summary report in SQL Scanner. The group summary report provides information about all the scan jobs for a selected group and includes:

- Job status
- Job type
- SQL classification

To generate a group summary report

1. Select the Scan SQL tab in the main window.
2. In the Task pane, select the **Scanner** node.
3. In the Group Summary pane, select a group and click .
4. Select the sections you want to include in the report from the Group Summary Report Options dialog.
5. Click **OK** to generate the report.

Generate Optimize SQL Resolution Report

You can generate a resolution report in SQL Optimizer after you optimize your SQL statement and execute the alternatives. The resolution report includes the following:

- Resolution summary
- Original SQL statement
- Optimization Session Information
- Scenario List
- Comparison Graph
- Best Alternative Scenario

To generate an Optimize SQL resolution report

1. Select the Optimize SQL tab in the main window.
2. Click .
3. Click  in the Report window to customize the information displayed in the resolution report.

i Tip: You can display execution plans in text mode, grayscale graphic mode, or full color graphic mode.

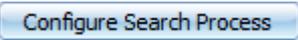
Tutorial: Optimize Indexes

To optimize indexes

1. Select the Optimize Indexes tab.
2. Click **Define New SQL Workload**.
3. Select the type of SQL workload source from which you want to collect SQL.

SQL Workload Source	Description
Scan Code	Scan SQL statements from file, database object source, or clipboard.
Plan Cache	Collect SQL from the Plan Cache.
SQL Profiler	Collect SQL with built-in SQL Profiler or load SQL from a trace file or a trace table.
Spotlight Statistics Repository	Collect SQL from Spotlight Statistics Repository.

i Note: Depending on the workload source you select, you may need to select a database connection from which to collect SQL.

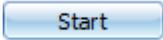
4. Specify the SQL filter criteria.
5. To begin collecting SQL, click .
6. Optimize Indexes immediately asks you to choose a second database connection. Select a database connection to use to retrieve execution plans and evaluate index alternatives.

i Note: This step allows you to select an alternate database (other than the one used to collect SQL) to run the index optimization process. In other words, you can collect your SQL from one database, but run the optimize indexes process on a second database. The second database must have the same application environment and data volume statistics as the first.

The Search Process page opens during the collection process. Before specifying search process criteria, you can review and modify your SQL workload.

i Notes:

- To restart the SQL collection process, click **Collected from <source>** to return to the SQL Criteria page. Then click **restarted**.
- Click **Advanced options** to modify your SQL workload before starting the index evaluation process.

7. Click  to begin the index evaluation process.

i Note: You can schedule the index evaluation process on a specific date and time.

8. Select the **Search Process** tab to review details related to the search process. Each line on the graph indicates a new indexing recommendation.
9. Select the **Results** tab to view the final recommendation (or the last recommendation if the search process is ongoing).

About Us

Quest creates software solutions that make the benefits of new technology real in an increasingly complex IT landscape. From database and systems management, to Active Directory and Office 365 management, and cyber security resilience, Quest helps customers solve their next IT challenge now.

Around the globe, more than 130,000 companies and 95% of the Fortune 500 count on Quest to deliver proactive management and monitoring for the next enterprise initiative, find the next solution for complex Microsoft challenges and stay ahead of the next threat.

Quest Software. Where next meets now. For more information, visit www.quest.com.

Contact Quest

For sales or other inquiries, visit www.quest.com/contact.

Technical Support Resources

Technical support is available to Quest customers with a valid maintenance contract and customers who have trial versions. You can access the Quest Support Portal at <https://support.quest.com>.

The Support Portal provides self-help tools you can use to solve problems quickly and independently, 24 hours a day, 365 days a year. The Support Portal enables you to:

- Submit and manage a Service Request
- View Knowledge Base articles
- Sign up for product notifications
- Download software and technical documentation
- View how-to-videos
- Engage in community discussions
- Chat with support engineers online
- View services to assist you with your product

SQL Optimizer Community

You can also find help and additional information at the SQL Optimizer community at: <https://blog.toadworld.com/tag/sql-optimizer-for-sql-server>