

Quest SQL Optimizer for IBM DB2 z/OS 5.6.1

User Guide



Contents

About Quest SQL Optimizer for IBM DB2 z/OS	9
Optimize SQL	9
About Quest SQL Optimizer for IBM DB2 z/OS	10
Optimize SQL	10
About Product Improvement Program	11
Select Connection	12
Connect to the Database	13
SQL Optimization Workflow	14
Create Support Bundles	15
Join the SQL Optimizer Community	16
Product Licensing	17
To register the trial or production version	17
Additional Resources	18
Toad World	18
Search	18
BTBSCAN	19
CORSUB	20
DELCUR	21
DELETE	22
DFETCH	23
DIXSCAN	24
EXCEPT	25
EXCEPTA	26
FETCH	27
FFETCH	28

FIXSCAN	29
HBJOIN	30
HSSCAN	31
INLIST	32
INSERT	33
INTERSECT	34
INTERSECTA	35
IXAND	36
IXOR	37
IXSCAN	38
MERGE	39
MERGESTMT	40
MIXSCAN	41
MSJOIN	42
NLJOIN	43
PARTITION	44
PIXSCAN	45
PRUNED	46
QB	47
RETURN	48
REPARTITION	49
RGLIST	50
RIDFETCH	51
SELUPD	52
SEMIJOIN	53

SIXSCAN	54
SORT	55
SORTRID	56
STARJOIN	57
TBSCAN	58
TRUNCATE	59
UNION	60
UNIONA	61
UPDATE	62
UPDCUR	63
WFSCAN	64
WORKFILE	65
XIXAND	66
XIXOR	67
XIXSCAN	68
About Optimizing SQL	69
About Optimizing SQL	70
Create Optimize SQL Sessions	71
Open Optimizer SQL Sessions	72
Optimize SQL Statements	73
Specify Session Test Run Criteria	75
Execution Method and Run Time	75
Order and Termination	75
SQL to Execute	76
Automatically Optimize SQL Statements	78
About Execution Plans	79
Execution Plan Window	79

Execution Plan Actions	79
About Execution Plans	80
Execution Plan Window	80
Execution Plan Actions	80
Review Execution Plans	81
Get Execution Plans	82
Fill Missing Execution Plans	83
Copy Execution Plans	84
Generate Execution Plan Alternatives	85
Compare Execution Plan Alternatives	86
Execute Execution Plan Alternatives	88
Execute Function	88
Batch Run Function	88
Session Batch Run Criteria	88
Batch Run Multiple Function	89
Retrieve Run Results	90
Deploy Plan Guides	91
Animate Execution Plans	92
About Generating Index Alternatives	93
Test Run Index Alternatives	94
Generate Index Alternatives	95
Delete Remaining Indexes	96
Test Run SQL Alternatives	97
Test Run Single SQL Function	97
Test Run Function	97
Session Test Run Criteria	98
Multiple Test Runs Function	98
Retrieve Run Results	99
Create User Defined Alternatives	100

Set Bind Variables	101
Review Alternatives	103
Alternative SQL Text	104
View Optimization Details	105
View Schema Information	106
View Bind Variables	107
Show or Hide Table Attributes	108
Show or Hide Column Attributes	109
Compare SQL Statements	110
Generate Optimize SQL Resolution Report	112
Troubleshoot Optimization Results	113
Execution Plan Options	114
Directory Setup Options	115
General Options	116
Appearance Options	117
Intelligence Options	118
Optimization Options	119
View to Nested Table Transformation	119
Query to Derived Table Transformation	119
Advanced SQL transformation	120
Join Tables	120
SQL Options	122
Quota Options	123
Index Options	124
Intelligence Options (Index Generation)	125
Test Run Settings Options	128
Execution Method Options	129

Order and Termination Options	130
Best Alternative Criteria Options	131
Tutorial: SQL Optimizer	132
Step 1: Optimize the SQL Statement	132
Step 2: Test Run Alternative SQL Statements	132
Tutorial: SQL Optimizer	134
Step 1: Optimize the SQL Statement	134
Step 2: Test Run Alternative SQL Statements	134
About us	136
Contacting Quest	136
Technical support resources	136
Third-party contributions	137

Copyright 2019 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

Quest Software is proud of our advanced technology. Patents and pending patents may apply to this product. For the most current information about applicable patents for this product, please visit our website at <https://www.quest.com/legal>.

Trademarks

Quest, and the Quest logo are trademarks and registered 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>. All other trademarks, servicemarks, registered trademarks, and registered servicemarks are the property of their respective owners.

Legend

-  **WARNING:** A WARNING icon indicates a potential for property damage, personal injury, or death.
-  **CAUTION:** A CAUTION icon indicates potential damage to hardware or loss of data if instructions are not followed.
-  **IMPORTANT, NOTE, TIP, MOBILE, or VIDEO:** An information icon indicates supporting information.

About Quest SQL Optimizer for IBM DB2 z/OS

Quest SQL Optimizer for IBM DB2 z/OS ensures optimal SQL performance by generating alternatives and testing SQL for maximum application performance.

Optimize SQL

SQL Optimizer analyzes your SQL statements and generates alternatives for improving performance. It can test these alternatives to identify the best SQL statement for your database environment. Optimizing SQL consists of the following functions:

Functions	Description
Optimize SQL Statement	SQL Optimizer uses an Artificial Intelligence engine to execute SQL syntax rules that produce semantically equivalent SQL statement alternatives.
Test Run SQL Alternatives	SQL Optimizer executes statement alternatives to view their execution statistics. This provides execution times for identifying the best SQL statement for your database environment. Test Run SQL Alternatives

About Quest SQL Optimizer for IBM DB2 z/OS

Quest SQL Optimizer for IBM DB2 z/OS ensures optimal SQL performance by generating alternatives and testing SQL for maximum application performance.

Optimize SQL

SQL Optimizer analyzes your SQL statements and generates alternatives for improving performance. It can test these alternatives to identify the best SQL statement for your database environment. Optimizing SQL consists of the following functions:

Functions	Description
Optimize SQL Statement	SQL Optimizer uses an Artificial Intelligence engine to execute SQL syntax rules that produce semantically equivalent SQL statement alternatives.
Test Run SQL Alternatives	SQL Optimizer executes statement alternatives to view their execution statistics. This provides execution times for identifying the best SQL statement for your database environment. Test Run SQL Alternatives

About Product Improvement Program

To prioritize enhancements in future releases, SQL Optimizer for IBM DB2 z/OS collects data about the use of its different features, and periodically, this data is communicated back to us. Initially, this usage data includes an IP address. Upon its receipt at a temporary server in the U.S.A., the IP address is removed, and then the anonymous data is aggregated before it is sent to our servers in California. Our product team analyses the aggregated data to understand our user community's preferences and common practices. This analysis influences our future releases. Click [here](#) for more information on the data we collect and on our privacy policy.

- No personal information is collected
- You can stop participating at any time

To initiate participation in Product Improvement Program

Select **Help | Product Improvement Program** and select **Yes, I want to participate**

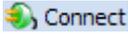
To cancel participation in Product Improvement Program

Select **Help | Product Improvement Program** and select **No, thank you**

Select Connection

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

To select a connection

1. Select the SQL Details tab in the main window.
2. Click  **Connect**
3. Review the following for additional information:

Select Connection	Description
Connection	Click  to select a previously created connection. Tip: Click  to open the Connection Manager to create a new connection. Connect to the Database
Select Schema and SQLID	Description
Schema	Click  to select a schema for the connection.
SQLID	Click  to select your SQLID.

Related Topics

[Create Optimize SQL Sessions](#)

[About Quest SQL Optimizer for IBM DB2 z/OS](#)

[Open Optimizer SQL Sessions](#)

[Connect to the Database](#)

[Select Connection](#)

[Tutorial: SQL Optimizer](#)

Connect to the Database

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

To create a connection

1. Click [Connection](#).
2. Review the following for additional information:

Connection Manager	Description
Database alias	Select name of the database to which you want to connect.
User	Enter the user name required to connect to the database.
Password	Enter the password associated with the user name. Click Save password to remember the password.
Connection Details	Description
Catalog node	Displays the local catalog node name.
Database name	Displays the local database name or DCS Alias.
Subsystem	Displays the name of the subsystem.
System	Displays system IP address and port number.
Comment	Displays comments.
Installed Database Clients	Description
Connect using	Select the appropriate database client or click  to browse.

3. Click **Connect** or **Close**.

To display connection information about a z/OS database in Toad.

1. Connect to a z/OS DB2 database.
2. Open **Editor**, type: "-DIS DDF" and execute.

Note: Consult the IBM DB2 documentation for more information on z/OS DB2 databases.

Note: If you use a non-administrator windows account to run SQL Optimizer for IBM DB2 z/OS, the login account must belong to the "DB2ADMIN" or "DB2USER" user group. You may also need to add "DB2ADMIN" or "DB2USER" if the user login belongs to the Administrator user group.

SQL Optimization Workflow

The SQL optimization workflow ensures that your SQL statements perform optimally in your database environment.

Procedure	Description
Optimize SQL Statements	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. Optimize SQL Statements
Test Run SQL	The Test Run SQL 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. Test Run SQL Alternatives
Compare SQL Alternatives	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.
Generate Reports	The Report feature creates a SQL resolution report after you complete the optimization process.

Create Support Bundles

The Support Bundle provides any easy way for you to contact Quest technical support to report a problem or ask a question. It includes information about each module of the application and your local computer; and can be used to help troubleshoot problems.

To create a support bundle

1. Select **Help | Support Bundle**.
2. Complete the Contact and Problem Details sections.

Join the SQL Optimizer Community

The SQL Optimizer for DB2 z/OS Community is available for you to find the latest information about SQL Optimizer. By joining the community, you can participate in the beta program, communicate with the development team and other users in the discussion forums, and read blog entries. Click [Toad for DB2 community](#) to join the community or follow the instructions below to access the community from SQL Optimizer.

To access the community

Select the Community tab in the main window.

Note: You need a connection to the Internet to access most community content.

Related Topic

[About Quest SQL Optimizer for IBM DB2 z/OS](#)

Product Licensing

To register the trial or production version

1. Select **Help | Licensing**.
2. Enter your Authorization key.
3. Enter your site message if you have a Production key.
4. Click **Apply**.

Additional Resources

Toad World

Visit other Toad Communities. Find DBMS and SQL knowledge, find software downloads, and find answers to your database questions. Click [here](#) to go to Toad World or follow the instructions below.

To go to Toad World

Click **Help** and select **Go to Toad World**.

Search

SQL Optimizer allows you to conduct searches in Toad World, Help, and Google.

To conduct a search

1. Enter your search item in the **Search** field at the top right..
2. Select an option from the list.

BTBSCAN

Scan a buffer table.

A buffer table construct contains two nodes: a BTBSCAN node and a buffer table node. The buffer table scan is the only way to access a buffer table.

CORSUB

Use a correlated subquery for access.

The CORSUB operation provides access by a correlated subquery.

DELCUR

Perform a DELETE WHERE CURRENT OF CURSOR operation.

DELETE

Remove rows from a table or view.

A delete construct contains a DELETE node and one of the following: table space scan, single-index access, or multiple-index access.

DFETCH

Represents a direct fetch data using ROWID.

ROWID is used as a search condition. This is the most efficient way to access a base table if a ROWID-type column is defined within the table.

DIXSCAN

Represents access by XML DOCID index scan.

EXCEPT

Denotes the EXCEPT operation.

This operation returns rows in the outer table only and eliminates redundant duplicate rows.

EXCEPTA

Denotes the EXCEPT ALL operation.

This operation returns rows in the outer table only and retains redundant duplicate rows.

FETCH

Represents the retrieval of columns for a row.
The FETCH operator uses RIDs from an IXSCAN or MIXSCAN.

FFETCH

Employs a fact table index in order to fetch a fact table in a data manager pushdown star join.

FIXSCAN

Represents a fact table index scan in a data manager pushdown star join.

HBJOIN

Use to perform a hybrid join.

A hybrid join is used only on inner joins. The construct contains a single HBJOIN node and two subtrees.

HSSCAN

Use to perform a HASH SCAN.

INLIST

Denotes the in-memory table that stores IN-list elements.

INSERT

Insert rows into table or view.

The insert construct contains two nodes: one INSERT node and one node that represents either the target view or table.

INTERSECT

Denotes the INTERSECT operation.

This node returns rows in both the inner and outer tables and eliminates redundant duplicate rows.

INTERSECTA

Denotes the INTERSECT ALL operation.

This node returns rows in both the inner and outer tables and retains redundant duplicate rows.

IXAND

Represents the intersection of two sorted ROWID lists. Output includes only ROWIDs in both lists.

IXOR

Return the union of two sorted ROWID lists.

Output includes any ROWID that is present in at least one ROWID list. The output removes duplicate ROWIDs.

IXSCAN

Use an index to retrieve the rows.

MERGE

Merge multiply data streams into a single data stream.

MERGESTMT

Denotes the MERGE statement.

This node merges the conditional UPDATE and INSERT operation on a target table or view into one statement.

MIXSCAN

Scan a multiple-index.

MSJOIN

Use to perform a merge join.

NLJOIN

Use to perform a nested loop join.

This join is used to join tables and is represented by a NLJOIN node and right and left subtrees. The inner table is represented by the right subtree and the outer table by the left subtree. Use either a single-index scan, a multiple-index scan, or a table space scan to access both the inner and outer tables. You can also include another join operation with the left subtree.

PARTITION

Separate a single data stream into multiple data streams.

PIXSCAN

Scan Dynamic Pair-Wise Index.

PRUNED

No access path is generated for the query since the query only qualifies zero rows.

Auxiliary node that represents a subquery in a particular SQL statement.

A subquery can be any of the following operations:

- MERGE
- INSERT
- SELECT
- UPDATE
- DELETE
- SELECT with FOR UPDATE OF
- UPDATE WHERE CURRENT OF CURSOR
- DELETE WHERE CURRENT OF CURSOR
- TRUNCATE
- Table expression
- Correlated subselect or full select
- Non-correlated subselect or fullselect
- EXCEPT
- EXCEPT ALL
- EXCEPT
- EXCEPT ALL
- UNION
- UNION ALL

Notes:

- The query block node is the root of a subtree and represents the whole subquery.
- QB_n denotes a query block node, where n represents the query block number.

RETURN

Auxiliary node that represents an entire SQL statement.

Labeled QUERY, this node can represent any of the following operations:

- MERGE
- INSERT
- SELECT
- UPDATE
- DELETE

REPARTITION

Re-partition multiple input data streams for multiple output data.

RGLIST

Use to perform a range-list Scan.

RIDFETCH

Use built-in RID function for access.

SELUPD

Denotes value SELECT with FOR UPDATE OF.

SEMIJOIN

Use to perform a semi join.

During a hybrid join, this node denotes a join operation where the outer table records merge with one of the indexes on the inner table.

SIXSCAN

Access and scan with a sparse index.

A sparse index scan construct contains two nodes: a SIXSCAN node and a sparse index node labeled SPARSE INDEX.

SORT

Sort result set in the order specified from the join process or the query.

SORTRID

Denotes operation where DB2 sorts qualified index entries.

STARJOIN

Use to perform a star join.

Star join allows you to join a dimension column of a fact table to the key column in the corresponding dimension table.

TBSCAN

Use to read every row in a table.

A table space scan construct consists of two nodes: a TBSCAN node and some type of table node such as a base table, a materialized query table, a work file, a pipe, or a table function.

TRUNCATE

Denotes TRUNCATE statement.

A TRUNCATE statement deletes all rows for either declared global temporary tables or base tables.

UNION

Combine rows from multiple tables and remove duplicate rows.

A UNION construct contains a UNION node and at least one subtree. Each subtree contains a WFSCAN node, a WORKFILE node, a SORT node, and the source query block node.

UNIONA

Combine rows from multiple tables and retain duplicate rows.

An UNION ALL construct contains a UNIONA node and at least one subtree. Subtrees denote any source query blocks.

UPDATE

Modify rows in a table.

An update construct contains one UPDATE node and one of the following: single-index access, multiple-index access construct, or table space scan.

UPDCUR

Represents UPDATE WHERE CURRENT OF CURSOR value.

WFSCAN

Scan a work file.

A work file scan construct contains a WFSCAN node plus a work file node that is labeled WORKFILE.

WORKFILE

Denotes a work file or temporary table that DB2 generates.

DB2 labels a work file node with the word WORKFILE or with specific name. Sort (for JOIN, UNION, DISTINCT, GROUP BY, or ORDER BY operations) is the most common operation for generating work files.

XIXAND

Denotes operation where DB2 returns intersection of two sorted DOCID lists. Output includes only DOCIDs in both lists.

XIXOR

Denotes operation for XML data where DB2 returns the union of two sorted DOCID lists.
The output includes any DOCID that exists in at least one of the DOCID lists and removes duplicate DOCIDs.

XIXSCAN

Access and scan an XML index.

This operation returns DOCID and NODWID pairs.

About Optimizing SQL

The Optimize SQL function analyzes the input SQL statement and uses an Artificial Intelligence Engine to produce a group of semantically equivalent versions of the statement, known as SQL alternatives. You can then test run these alternatives in the SQL Optimizer window to determine the best-performing version of the SQL.

Note: The Intelligence Level setting you select affects the duration of the optimization process and the number of alternatives SQL Optimizer generates.

About Optimizing SQL

The Optimize SQL function analyzes the input SQL statement and uses an Artificial Intelligence Engine to produce a group of semantically equivalent versions of the statement, known as SQL alternatives. You can then test run these alternatives in the SQL Optimizer window to determine the best-performing version of the SQL.

Note: The Intelligence Level setting you select affects the duration of the optimization process and the number of alternatives SQL Optimizer generates.

Create Optimize SQL Sessions

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

To create a new session

1. Select the SQL Details tab in the main window.
2. Click .

To add a session

Click  besides the Session tab.

Note: You need to create a new connection for each new SQL Optimizer session.

Related Topics

[About Quest SQL Optimizer for IBM DB2 z/OS](#)

[Open Optimizer SQL Sessions](#)

[Connect to the Database](#)

[Select Connection](#)

[Tutorial: SQL Optimizer](#)

Open Optimizer SQL Sessions

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

To open a session

1. Select the SQL Details tab in the main window.
2. Click .
3. Select a Optimize session to open..

Related Topics

[Create Optimize SQL Sessions](#)

[About Quest SQL Optimizer for IBM DB2 z/OS](#)

[Open Optimizer SQL Sessions](#)

[Connect to the Database](#)

[Select Connection](#)

[Tutorial: SQL Optimizer](#)

Optimize SQL Statements

SQL Optimizer generates semantically equivalent alternatives with unique execution plans for your original SQL statement. You can then test run the alternatives to benchmark their performance. SQL Optimizer provides execution times and run-time statistics for each alternative to help you identify the best SQL statement for your database environment.

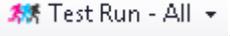
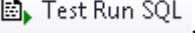
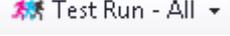
Note: This topic focuses on information that may be unfamiliar to you. It does not include all step and field descriptions.

To optimize a SQL statement

1. Select the SQL Details tab in the main window.
2. Enter a SQL statement in the Alternative Details pane of the SQL Details tab.
3. Click  to optimizer the SQL statement.
4. Select a connection to use. Review the following for additional information:

Select Connection	Description
Connection	Click  to select a previously created connection. Tip: Click  to open the Connection Manager to create a new connection. Connect to the Database
Select Schema and SQLID	Description
Schema	Click  to select a schema for the connection.
SQLID	Click  to select your SQLID.

5. When SQL Optimizer is finished generating alternatives, use one of the following methods to test run the SQL alternatives:

- To test run all SQL alternatives, click .
- To test run a single SQL alternative, select the alternative and click .
- To test run two or more alternatives simultaneously, select the alternatives, click the arrow beside  and select **Test Run - Selected**.

The execution statistics display in the Alternatives pane once the test run completes. Select an alternative to see more information in the Alternative Details and Execution Plan panes.

[Test Run SQL Alternatives](#)

Tips:

- Click  to stop the test run process.
- You can select an executing SQL statement and click  to abort only the selected statement.

To clear the SQL Rewrite window

To clear only the results and retain the original SQL statement, right-click within the Alternative Details pane and select **Clear Optimization Results | Keep Original Scenario**.

To clear the SQL Rewrite window, right-click within the Alternative Details pane and select **Clear Optimization Results | Clear Original Scenario**.

Related Topics

[Automatically Optimize SQL Statements](#)

[Test Run SQL Alternatives](#)

Specify Session Test Run Criteria

Use the Session Test Run Criteria dialog to specify options to use for the current test run of SQL statements in the Optimize SQL module. The options you specify are used for the current test run only. Global test run options for Optimize SQL are modified at [Options | Optimize SQL | Test Run](#).

Execution Method and Run Time

Select from the following options:

Execution Method	Description
Maximum rows to be retrieved (records)	Select this checkbox and then enter the maximum number of rows to retrieve for a test run of all records.
Run Time Retrieval Method	Description
Run SQL options	Select one of the following options: <ul style="list-style-type: none">• Run all SQL twice if original SQL runs faster than (seconds)—Combines the Original SQL twice and all others once and the All SQL twice options into one option and allows you to determine (by the number of seconds a SQL statement runs) which option to use. The original SQL statement always runs twice. The SQL alternatives run twice if the original SQL statement runs in less time than the value specified. Otherwise, the SQL alternatives all run once.• Original SQL twice and all others once—Caches data from a table into memory the first time you access it. The next time you access that data, it is already in memory so the following SQL statements run faster. To provide an accurate comparison, the first SQL statement runs twice but only the time from the second run is compared to the times for the other statements.• All SQL twice—Executes all SQL statements twice to eliminate factors that can affect the accuracy of the results. If you recently executed a SQL statement, the information for that statement may be cached and the statement may execute faster. This option eliminates time variation caused by caching since it runs all SQL statements twice but only uses the second run time for comparison.• All SQL once—Executes all SQL statements once. For long running SQL, you do not need to run any statement twice since the effect from caching diminishes over time.

Order and Termination

Select from the following options:

Execution Order	Description
Execution order for	Select one of the following options:

- Intelligent order—Executes representative SQL statements with various plan cost according to SQL Optimizer's intelligence engine.
- Plan cost—Executes SQL statements in order of plan cost.

SQL Termination Criteria	Description
Terminate execution of SQL alternative if it runs longer than	<p>Select one of the following options:</p> <ul style="list-style-type: none"> • Run time of fastest SQL—Cancels SQL statements that run longer than the current fastest run time. With this option, the first SQL statement runs and the time from that statement is used as the termination time for the next SQL statement. When a SQL statement runs faster than this time, the faster time is used as the new termination time. • This percentage of the original SQL run time—Cancels SQL statements whose total elapsed time is the specified % of the total elapsed time for the original SQL statement. It terminates all SQL statements that run longer than the calculated termination time. • User defined time (mins/secs)—Cancels SQL statements that run longer than a time you specify. • Run without termination—Runs all SQL statements to completion regardless of run time.
Cancellation delay (seconds)	Adds a specified time to the termination time. It is important to factor a delay into the overall termination time to account for the time needed to send the SQL statement to the database server.

SQL to Execute

Execute Criteria	Description
Execute all alternatives with criteria	<p>Select one of the following:</p> <ul style="list-style-type: none"> • % of alternatives with lowest cost—Enter the percentage of SQL alternatives to execute with the lowest cost. You can also enter the minimum and maximum number of alternatives to execute. • Number of alternatives with lowest cost—Enter the number of SQL alternatives to execute with the lowest cost. • All alternatives with cost less than or equal to original SQL—Select to execute all SQL alternatives with a cost less than or equal to the cost of the original SQL statement. • All alternatives with cost less than the average of all alternatives—Select to execute all SQL alternatives with a cost less than the average cost of all alternatives. • All alternatives with cost less than the original SQL by percentage—Enter a percentage used to determine the SQL alternatives selected for execution. Optimize SQL executes alternatives with a cost that is the specified percentage lower than the cost of the original statement.

- All alternatives with cost less than the original SQL by N times—Enter a value for N used to determine the SQL alternatives selected for execution. Optimize SQL executes alternatives with a cost N times lower than the original SQL statement.
- All alternatives—Select to execute all SQL alternatives.

Tips:

- Click  to stop the test run process.
- You can select an executing SQL statement and click  to abort only the selected statement.

Related Topics

[About Quest SQL Optimizer for IBM DB2 z/OS](#)

[Open Optimizer SQL Sessions](#)

[Execution Method Options](#)

[Order and Termination Options](#)

Automatically 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 starts testing once SQL Optimizer generates the first alternative. The Auto Optimize function reduces the time required to find the best alternative by not waiting until SQL Optimizer generates all alternatives before starting the testing process. You can stop the Auto Optimize function once you find a satisfactory SQL statement alternative.

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

To automatically optimize a SQL statement

1. Select the SQL Details tab in the main window.
2. Enter a SQL statement in the Alternative Details pane.
3. Click .

Tip: Click  to stop the optimization and testing processes.

Related Topics

[About Quest SQL Optimizer for IBM DB2 z/OS](#)

[Optimize SQL Statements](#)

About Execution Plans

The execution plan displays the steps a database takes to execute a SQL statement. You can use the execution plan to determine if a statement is efficient.

Each step of the tree indicates how SQL Optimizer retrieves rows of data. The first line of the execution plan displays the SQL statement type, such as SELECT. The remaining lines represent an operation. The operations are numbered in the order of execution to make the plan easier to read.

The database executes each child operation before the parent operation. For some SQL statements, the database executes the parent operation once it retrieves a single row from the child operation. Other SQL statements require that the database retrieve all rows from the child operation before it executes the parent operation.

Execution Plan Window

The Execution Plan pane shows types of information for the original SQL or the SQL associated with the currently selected alternative.

Execution Plan Actions

Right-click in the Execution Plan window to select from the following actions:

Action	Description
Copy	Copies the execution plan to the clipboard.
Save	Saves the execution plan as a JPG file.
Print	Opens the print window so you can print the execution plan.
Style	Select option to display different plan details (operators, predicates, statistics).
View Plan	Allows you to change how the execution plan is displayed.
Step Description	Select this option to display a description of the step selected in the execution plan.
Animate Plan Steps	Highlights, one-by-one, the execution plan steps.
Plan Options	Opens the Execution Plan Options window so you can select which information is displayed in the execution plan and whether to display specific items in a column.
Help on <i>RETURN</i>	Displays the help text for the currently selected operation in the execution plan.
Help on Execution Plan	Opens online help for the Execution Plan window.

Related Topics

[Animate Execution Plans](#)

[Copy Execution Plans](#)

[Fill Missing Execution Plans](#)

[Get Execution Plans](#)

About Execution Plans

The execution plan displays the steps a database takes to execute a SQL statement. You can use the execution plan to determine if a statement is efficient.

Each step of the tree indicates how SQL Optimizer retrieves rows of data. The first line of the execution plan displays the SQL statement type, such as SELECT. The remaining lines represent an operation. The operations are numbered in the order of execution to make the plan easier to read.

The database executes each child operation before the parent operation. For some SQL statements, the database executes the parent operation once it retrieves a single row from the child operation. Other SQL statements require that the database retrieve all rows from the child operation before it executes the parent operation.

Execution Plan Window

The Execution Plan pane shows types of information for the original SQL or the SQL associated with the currently selected alternative.

Execution Plan Actions

Right-click in the Execution Plan window to select from the following actions:

Action	Description
Copy	Copies the execution plan to the clipboard.
Save	Saves the execution plan as a JPG file.
Print	Opens the print window so you can print the execution plan.
Style	Select option to display different plan details (operators, predicates, statistics).
View Plan	Allows you to change how the execution plan is displayed.
Step Description	Select this option to display a description of the step selected in the execution plan.
Animate Plan Steps	Highlights, one-by-one, the execution plan steps.
Plan Options	Opens the Execution Plan Options window so you can select which information is displayed in the execution plan and whether to display specific items in a column.
Help on <i>RETURN</i>	Displays the help text for the currently selected operation in the execution plan.
Help on Execution Plan	Opens online help for the Execution Plan window.

Related Topics

[Animate Execution Plans](#)

[Copy Execution Plans](#)

[Fill Missing Execution Plans](#)

[Get Execution Plans](#)

Review Execution Plans

The execution plan can be displayed in different ways to help you get more detailed information.

To change how the execution plan is displayed

1. Right-click the plan and select **View Plan**.
2. Select one of the following display options:
 - As Tree Plan
 - As Plain Language Plan
 - As Graphic Plan
 - As MS Graphic Plan

Tips:

- You can change the colors used to highlight different items for the current plan only by right-clicking the plan and selecting **Plan Options**. You can change colors for all execution plans in **Options | General | Execution Plan**. [Execution Plan Options](#)
- You can use the toolbar to the right of the Execution Plan pane to change the details displayed.



Related Topics

[About Execution Plans](#)

[Animate Execution Plans](#)

[Copy Execution Plans](#)

Get Execution Plans

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

SQL Optimizer automatically retrieves the execution plan when you optimize or test run a SQL statement.

To get the execution plan

1. Select the Optimize SQL tab.
2. Enter a SQL statement in the Alternative Details pane.

3. Click  .

Related Topics

[About Execution Plans](#)

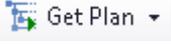
[Animate Execution Plans](#)

[Copy Execution Plans](#)

Fill Missing Execution Plans

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

To fill a missing execution plan

1. Select the Optimize SQL tab.
2. Enter a SQL statement in the Alternative Details pane.
3. Click the arrow beside  and select **Fill Missing Plans**.

Related Topics

[About Execution Plans](#)

[Get Execution Plans](#)

Copy Execution Plans

To copy an execution plan

Right-click the execution plan and select **Copy**.

Note: You can paste execution plans displayed as a Tree Plan or Plain Language Plan in text or bitmap format. Applications such as Microsoft Word allow you to choose the format using the **Paste Special** option. You can use text applications such as Notepad to paste execution plans in text format or graphic applications such as MS Paint to paste execution plans in bitmap format.

Related Topics

[About Execution Plans](#)

[Fill Missing Execution Plans](#)

[Get Execution Plans](#)

Generate Execution Plan Alternatives

Plan Control sessions optimize your SQL statement by generating execution plan alternatives for the statement. You can then select an alternative execution plan for your SQL statement and deploy it as a plan guide to improve database performance.

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

To generate execution plan alternatives

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

Note: If the start page does not display, click the arrow beside  and select New Plan Control Session.

3. Enter a SQL statement in the Original SQL pane.
4. Click  to retrieve the execution plan for your SQL statement. The Select Connection window displays.
5. Select a connection.
6. Click  to generate execution plan alternatives.

Related Topics

[Execute Execution Plan Alternatives](#)

Compare Execution Plan Alternatives

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.

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

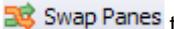
1. Select the SQL Details tab in the main window.

2. Click .

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

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

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**.

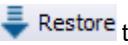
4. Click  to switch the location of the original SQL statement and selected alternative in the Comparison window.

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.
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.
Plan Only (Top-Bottom)	Displays the execution plan for the statements you are comparing beneath each other.

Tips:

- Click  to maximize the comparison window.
- Click  to restore the comparison window to its original size.

Related Topics

Optimize SQL Statements
Test Run SQL Alternatives

Execute Execution Plan Alternatives

Once SQL Optimizer generates execution plan alternatives, you can execute them to test each alternative or use Batch Run to test all alternatives simultaneously. Executing the alternatives retrieves run time statistics and identifies the best alternative for your database environment.

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

Execute Function

To execute a single statement

1. Select the SQL Details tab in the main window.
2. Select a Plan Control session.
3. Select a single SQL statement in the Plans pane.
4. Click  and select **Execute (All Records)** or **Execute (First Record)**.

Batch Run Function

To execute selected statements

1. Select the SQL Details tab in the main window.
2. Select a Plan Control session.
3. In the Plans pane, select multiple SQL statements in one of the following ways:
 - To select a consecutive group of SQL statements, click the first SQL, press and hold SHIFT, and then click the last SQL.
 - To select non-consecutive SQL statements, press and hold CTRL, and then click each SQL that you want to select.
4. Click  and select **All Records** or **First Record, Batch Run Selected**.

To execute all statements

1. Select the SQL Details tab in the main window.
2. Select a Plan Control session.
3. Click  and select **All Records** or **First Record, Batch Run Selected**.

Session Batch Run Criteria

Every Plan Control session can have its own batch run settings. See the following topics for more information:

- [Execution Method Options](#)
- [Order and Termination Options](#)

Tips:

- You can change Batch Run settings globally on the Batch Run Options page. [Test Run Settings Options](#)
- Click  to stop the batch run process.

Batch Run Multiple Function

You can use the Batch Run Multiple Function to obtain statistics for statements with run times in the millisecond range. Statements with run times in the millisecond range can be skewed by other active processes because run time is based on a CPU's clock time. Multiple active processes can slow down CPU performance and result in a longer than normal run time for a statement. Use this function to execute selected or all statements multiple times. The statistics reported using this function are an average of the results from these multiple executions.

To execute selected statements multiple times

1. Select the SQL Details tab in the main window.
2. Select a Plan Control session.
3. Select multiple alternatives in the Plans pane.
4. Click  and select **All Records**) or **First Record, Batch Run Multiple Selected**.
5. Enter the number of times you want each SQL statement executed.

To execute all statements multiple times

1. Select the SQL Details tab in the main window.
2. Select a Plan Control session.
3. Click  and select **All Records** or **First Record, Batch Run Multiple All**.
4. Enter the number of times you want each SQL statement executed.

Related Topics

[About Optimizing SQL \(Plan Control\)](#)

[Generate Execution Plan Alternatives](#)

[Deploy Plan Guides](#)

Retrieve Run Results

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

To retrieve run results

1. Right-click the Alternative Details or SQL Text pane and select Run Result.

2. Click .

Tip: Click  to copy the SQL statement to execute in another Questproduct.

Deploy Plan Guides

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

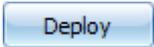
To deploy a plan guide

1. Select the SQL Details tab in the main window.
2. Select a Plan Control Session.
3. Click  and select the alternative from the **Select a Plan to Deploy** list.

Note: You can also right-click the alternative you want to deploy as a plan guide in the **Alternatives** pane and select **Deploy Plan Guide**.

4. Review the following for additional information:

Deploy	Description
Select a plan to deploy	Click  and select an execution plan alternative to deploy.
Performance Comparison	Description
Mark the plan as	Review the following for additional information: <ul style="list-style-type: none">• Enabled—Select whether to enable or disable this plan.• Fixed—Select whether to deploy this plan as fixed or non-fixed.• Not Auto-Purged—Select whether to auto-purge when it is not used.
Plan name	Enter a name for the plan.
Description	Enter a description for this plan.

5. Click .

Related Topics

[Generate Execution Plan Alternatives](#)

[Execute Execution Plan Alternatives](#)

Animate Execution Plans

You can animate each step in the execution plan in the order SQL Optimizer executes them.

To animate the execution plan or cancel the animation

Right-click the execution plan and select **Animate Plan Steps**.

Related Topics

[About Execution Plans](#)

[Copy Execution Plans](#)

[Fill Missing Execution Plans](#)

[Get Execution Plans](#)

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.

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

Related Topics

[Test Run Index Alternatives](#)

[Generate Index Alternatives](#)

Test Run Index Alternatives

After SQL Optimizer generates index alternatives, you can test each alternative. When SQL Optimizer tests indexes, the indexes are physically created on the database and dropped after SQL Optimizer executes the statement. When you test run index alternatives, you must select a storage group in which to store the created indexes.

(missing or bad snippet)

To execute index alternatives

1. (missing or bad snippet)
2. Select a SQL Rewrite session.
3. Select an index set in the Alternatives pane.
4. Click .
5. Select a storage group in which to store the created indexes. (missing or bad snippet)

Tip: Click  to test run all index alternatives.

Index Information	Description
Index	Shows SQL text of index selected.
Storage group	Storage group to store indexes created.
	Note: Once you select the storage group for the first index, SQL Optimizer automatically selects the same storage group for the remaining indexes.

Note: Review the information in the DDL Script pane.

Related Topic

[About Generating Index Alternatives](#)

[Generate Index Alternatives](#)

Generate Index Alternatives

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

To generate index alternatives for a SQL statement

1. Select the SQL Details tab in the main window.
2. Enter the SQL statement in the Alternative Details pane.
3. Click **Index** button.

Note: Delete Remaining Indexes Window may appear if it has been detected that some virtual/real indexes exist in the database and these might affect index searching result. It is recommended to remove the unused indexes before starting the search.

1. Select **Index Details** in the SQL Information pane to view index generation information.

Tip: Click  to stop the index generation process.

Note: When you generate virtual index alternatives, the execution plan is not retrieved for the SQL statement because SQL Optimizer does not physically create indexes on the database. [Fill Missing Execution Plans](#)

Related Topics

[About Generating Index Alternatives](#)

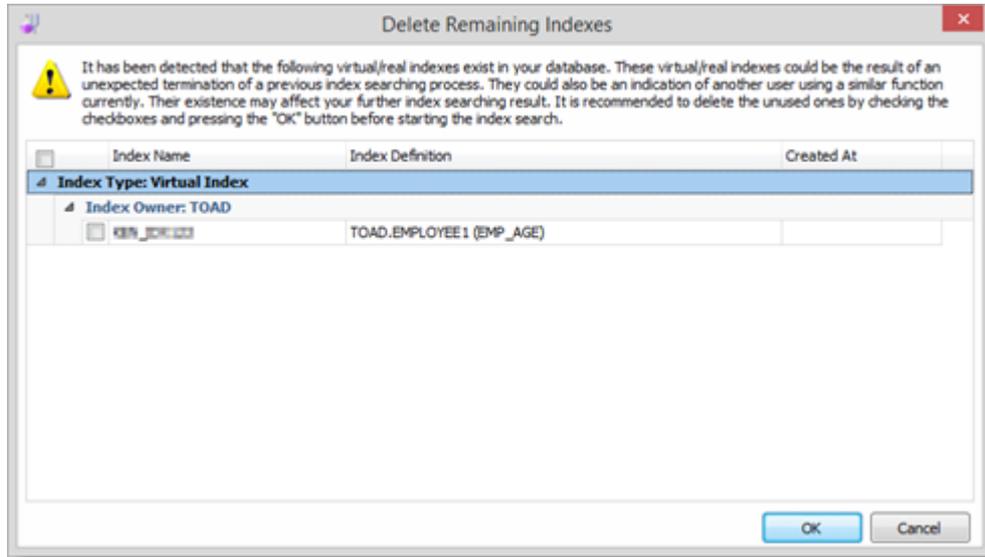
[Delete Remaining Indexes](#)

[Test Run Index Alternatives](#)

Delete Remaining Indexes

Delete Remaining Indexes window appears if it has been detected that some virtual/real indexes exist in the database and these might affect index searching result.

It is recommended to remove the unused indexes before starting the search. Click **OK** to remove the selected indexes.



Test Run SQL Alternatives

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

Use SQL Optimizer to perform test runs on alternatives in order to view their run-time statistics. You can test run a single statement, selected statements, or all statements. Use the Test Run Single SQL function to retrieve run-time statistics for a single statement.

Use the Test Run function to perform multiple test runs on selected or all statements. These functions allow you to get run-time statistics for selected SQL or average run-time statistics for all SQL.

Note: SQL Optimizer automatically rolls back any changes made to the database if the SQL is an INSERT, UPDATE, DELETE, or MERGE statement. This maintains the integrity of your data and provides that the initial data is the same for each SQL alternative so that the test is comparable.

Test Run Single SQL Function

Perform a test run on a single SQL statement run to get run-time statistics.

To test run a single statement

1. Select the Optimize tab.
2. Select a SQL statement in the Alternatives pane.
3. Click  Test Run SQL.

Test Run Function

To test run all statements

1. Select the Optimize tab.
2. Click the arrow beside  Test Run - All and select **Test Run - All**.
3. Use the tabs to select test run criteria.

To test run selected statements

1. Select the Optimize tab.
2. In the Alternatives pane, select multiple SQL statements in one of the following ways:
 - To select a consecutive group of SQL statements, click the first SQL, press and hold SHIFT, and then click the last SQL.
 - To select non-consecutive SQL statements, press and hold CTRL, and then click each SQL that you want to select.
3. Click the arrow beside  Test Run - All and select **Test Run - Selected**.
4. Use the tabs to select test run criteria.

Session Test Run Criteria

Every session can have its own test run settings. See the following topics for more information:

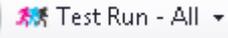
- [Execution Method Options](#)
- [Order and Termination Options](#)

Tips:

- You can change Test Run settings globally on the Test Run Options page. [Test Run Settings Options](#)
- Click  to stop the test run process.
- You can select an executing SQL statement and click  to abort only the selected statement.

Multiple Test Runs Function

To test run all statements multiple times

1. Select the Optimize tab.
2. Click the arrow beside  and select **Multiple Test Runs - All**.
3. Enter the number of times you want each SQL statement executed.

To test run selected statements multiple times

1. Select the Optimize tab.
2. In the Alternatives pane, select multiple SQL statements in one of the following ways:
 - To select a consecutive group of SQL statements, click the first SQL, press and hold SHIFT, and then click the last SQL.
 - To select non-consecutive SQL statements, press and hold CTRL, and then click each SQL that you want to select.
3. Click the arrow beside  and select **Multiple Test Runs - Selected**.
4. Enter the number of times you want each SQL statement executed..

Related Topics

[Optimize SQL Statements](#)

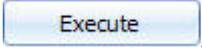
[Automatically Optimize SQL Statements](#)

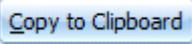
Retrieve Run Results

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

To retrieve run results

1. Select the Optimize SQL tab.
2. Enter a SQL statement in the Alternative Details pane of the SQL Details tab.
3. Right-click the Alternative Details or SQL Text pane and select **Run Result**.
4. Select the maximum number of rows to retrieve.

5. Click .

Tip: Click  to copy the SQL statement to execute in another Questproduct.

Create User Defined Alternatives

In addition to using SQL Optimizer to generate alternative statements, you can create user defined alternatives. You can create user defined alternatives using SQL text from your original SQL statement or from SQL text of an alternative SQL Optimizer generated.

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

To create a user defined alternative

1. Select the SQL Details tab in the main window.
2. Enter the SQL statement in the in the Alternative Details pane.
3. Click .
4. Edit the statement in the SQL Text pane to create a user defined alternative.

Note: To create a user defined alternative using SQL text of an alternative SQL Optimizer generated, right-click an alternative and select **User Alternative**.

Caution: SQL Optimizer does not check that user defined alternatives are semantically equivalent to your original SQL statement. Review the execution statistics for user defined alternatives to compare the results with your original statement. See View Execution Statistics for more information.

Related Topics

[Compare SQL Statements](#)

[Test Run SQL Alternatives](#)

Set Bind Variables

You need to define the values of SQL statement variables before you optimize the statement. The Set Bind Variables window displays automatically when you use SQL Optimizer for a statement with variables.

Troubleshooting: If the Set Bind Variables window displays when you optimize a statement without variables, make sure you spelled the column and table names correctly, you selected the correct database or user, and you selected a table or column that exists in the database.

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

To set a bind variable

1. Select the SQL Details tab in the main window.
2. Enter a SQL statement with a bind variable.

Note: SQL Optimizer currently supports the following bind variables data type:

- CHAR
- NUMERIC
- DECIMAL
- INTEGER
- SMALLINT
- FLOAT
- REAL
- DOUBLE
- DATETIME
- VARCHAR
- TYPE_DATE
- TYPE_TIME
- TYPE_TIMESTAMP
- GRAPHIC
- VARGRAPHIC
- LONGVARGRAPHIC
- BLOB
- CLOB
- DBCLOB
- DATALINK

3. Click .

4. Review the following for additional information:

Bind Variables List	Description
Datatype	Click  and select the variable datatype.
Null	Select the checkbox if the value for the variable is null.
Variable Value	Enter a value for the variable.
Browse Data	Description
SELECT	Click  and select column references for the variables. Note: SQL Optimizer selects the All Columns checkbox by default.
FROM	Click  and select table references for the variables. Tip: Click  to browse to tables in a different database.
WHERE	Enter a WHERE clause or click  to select a previously entered clause.
ORDER BY	Enter a ORDER BY clause or click  to select a previously entered clause.

Related Topics

[Optimize SQL Statements](#)

[Test Run SQL Alternatives](#)

Review Alternatives

Use the Alternatives pane to review the SQL text and execution plan for the alternatives SQL Optimizer generates. You can also review the run-time statistics for your original SQL statement and the alternatives after you execute the statements.

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

To review details for an alternative

1. Select the SQL Details tab in the main window.
2. Select an alternative in the Alternatives pane.
3. Review details for the alternative in the Alternative Details and Execution Plan panes.

Related Topics

[Optimize SQL Statements](#)

[Automatically Optimize SQL Statements](#)

[Alternative SQL Text](#)

Status	Displays client status.
Status	Displays client status.
Elapsed Time	Displays the execution time to retrieve the entire result set.

Classification

SQL Information	Description
SQL name	Name of the SQL statement.
Classification	Statement type as classified by SQL Optimizer. Types include Simple, Complex, or Problematic.
Classification Rules Detail	Description
Details	Applicable SQL Classification Rules. Rules used for the classification process are set in the SQL Classification Options menu.

Note: SQL Classification information only displays for the original SQL statement.

Alternative SQL Text

The text editor is used to display the SQL text of the selected alternative. When there's only original alternative or the selected alternative is user defined, the SQL text can be edited. Otherwise, it's read-only.

View Optimization Details

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

To view optimization details

1. Select the SQL Details tab in the main window.
2. In the Alternatives pane, click .
3. Select the SQL Information tab in the Optimize SQL window. Review the following for additional information:

Optimization Details	Description
Intelligence level	Intelligence level used for optimization. This option is set on the Optimizer Intelligence options page. Intelligence Options
Semantically equivalent SQL statements	Number of semantically equivalent SQL statements generated by SQL Optimizer.
Alternative execution plans produced	Number of alternative execution plans produced for the original SQL statement.
SQL statements eliminated due to identical execution plans	Number of statement alternatives eliminated with the same execution plan as the original SQL statement. Note: SQL statements with the same execution plan produce the same run time because the database executes these statements in the same way. Therefore, SQL Optimizer eliminates alternatives with equivalent execution plans.
Optimization Time Details	Description
Optimization started at	Time optimization started.
Optimization finished at	Time optimization finished.
Total optimization time	Total time to complete optimization.
Average optimization time	Average time to generate each statement alternative.

Notes:

- This information is only available after you generate SQL alternatives.
- Optimization details only display for the original SQL statement.

View Schema Information

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

To view schema information

1. Select the SQL Details tab in the main window.
2. Select the Schema Information tab in the Optimize SQL window.
3. Click  beside the Summary list and select a table for additional information.

Related Topics

[About Optimizing SQL](#)

[View Bind Variables](#)

View Bind Variables

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

To view bind variables

1. Select the SQL Details tab in the main window.
2. Select the Bind Variables tab in the Optimize SQL window.

The parameters for the bind variable defined for your SQL statement display. [Set Bind Variables](#)

Note: Information is only displayed in this tab if you used variables for optimization.

Related Topics

[About Optimizing SQL](#)

[View Schema Information](#)

Show or Hide Table Attributes

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

To show or hide a table attribute

1. Select the SQL Details tab in the main window.
2. Select the Schema Information tab in the Optimize SQL window.
3. Right-click the header row in the Summary Details pane and select **Column Chooser**.
4. Add or remove table attributes by dragging the column names to and from the Customization dialog.

Related Topics

[View Schema Information](#)

[Show or Hide Column Attributes](#)

Show or Hide Column Attributes

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

To show or hide a column attribute

1. Select the SQL Details tab in the main window.
2. Select the Schema Information tab in the Optimize SQL window.
3. Click  besides the Summary list and select a table.
4. Right-click the header row of the Column Attributes pane and select **Column Chooser**.
5. Add or remove column attributes by dragging the column names to and from the Customization dialog.

Related Topics

[View Schema Information](#)

[Show or Hide Table Attributes](#)

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.

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

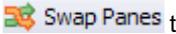
1. Select the SQL Details tab in the main window.

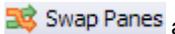
2. Click .

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

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

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.

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.

Tips:

- Click  to maximize the comparison window.
- Click  to restore the comparison window to its original size.

Related Topics

[Optimize SQL Statements](#)

[Test Run SQL Alternatives](#)

Generate Optimize SQL Resolution Report

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

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

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

To generate an Optimize SQL resolution report

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

2. Click  Report.

3. Click  in the Report window to customize the information displayed in the resolution report.

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

Related Topics

[Optimize SQL Statements](#)

[Test Run SQL Alternatives](#)

Troubleshoot Optimization Results

After you optimize the original SQL statement and determine the best alternatives, you may find the performance of the selected statements unsatisfactory in your database environment.

Make adjustments to the intelligence, optimization, SQL options, or quota setting options if SQL Optimizer has reached any of the quotas. To access these settings, click [Options](#) and select **Optimize SQL | SQL Rewrite**.

Additionally, review the execution plan of the original and optimized SQL statements to check if you need to change the database structure.

Related Topics

[Optimize SQL Statements](#)

[Automatically Optimize SQL Statements](#)

[Test Run SQL Alternatives](#)

Execution Plan Options

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

To set execution plan options

1. Click [Options](#).
2. Select **General | Execution Plan**.
3. Review the following for additional information:

Execution Plan	Description
TABLE ACCESS FULL warning threshold	Specify a table size threshold at which the TABLE ACCESS FULL operation is highlighted in red.
Color	Specify the color of the individual items in the query plan by clicking the Color column in the row for the item and selecting the new color from the dialog.
Font	Specify the font settings of the individual items in the query plan by clicking the Font column in the row for the item and selecting the new settings from the dialog.

Note: The execution plan displays in several panes. The color and font settings in this window controls the settings in all the Execution Plan panes. Right-click the execution plan and select Plan Options to customize display options for individual panes.

Note: These options apply to SQL Optimizer..

Related Topic

[About Execution Plans](#)

[Directory Setup Options](#)

[General Options](#)

Directory Setup Options

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

To set directory setup options

1. Click [Options](#).
2. Select **General | Directory Setup**.
3. Review the following for additional information:

Directory for	Description	Default Directory
Optimize SQL data directory	Optimize SQL files	C:\Documents and Settings\User\Application Data\Quest Software\Quest SQL Optimizer for DB2 z/OS\version\Optimize SQL Data

Related Topics

[Execution Plan Options](#)

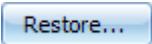
[General Options](#)

General Options

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

To restore layouts

1. Click [Options](#).
2. Select **General | Appearance**.
3. Review the following for additional information:

General Options	Description
Restore Layouts	Click  and Yes .

Appearance Options

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

To set appearance options

1. Click [Options](#).
2. Select **General | Appearance**.
3. Review the following for additional information:

Font Settings	Description
SQL Text Editor	Select the font settings for the SQL Text Editor.
Other Data	Select the font settings for other data.
Size	Select the font size.

Related Topics

[Execution Plan Options](#)

[Directory Setup Options](#)

Intelligence Options

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

To set intelligence options

1. Click [Options](#).
2. Select **Optimize SQL | SQL Rewrite | Intelligence**
3. Review the following for additional information:

Intelligence Level	Description
Custom	Enables you to customize the settings.
Predefined	Uses the predefined set of options.

Related Topics

[Quota Options](#)

[SQL Options](#)

[Optimization Options](#)

Optimization Options

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

To set Optimization options

1. Click [Options](#).
2. Select **Optimize SQL | SQL Rewrite | Optimization**

View to Nested Table Transformation

This transformation is only applicable if the SQL statement is using a view to access information from the database. When a SQL statement is using a view, the SQL Rewrite optimizes the view's SQL statement along with the original SQL statement. SQL Rewrite inserts the view's SQL into the original SQL statement in every place the view is referenced. Therefore the view's SQL is going to be rewritten along with the original SQL. This is very useful when you want to optimize a SQL statement that is using a poor performing view but you cannot change the view's SQL.

View to nested table transformation options	Description
Transform view to nested table	Specify whether to transform view to nested table.
Transformation level	Specify the recursive level to transform views inside the subquery of a nested table. You can control whether the view's SQL is rewritten with the original SQL statement with the Transform view to nested table option. SQL Rewrite can also transform a view that is being used by another view. You control how many views will be included when the original SQL is rewritten by specifying the transformation levels that you would like to perform during the optimization.

Query to Derived Table Transformation

This transformation takes a original SQL statement with an IN or EXISTS clause and rewrites it as a derived table.

Query to derived table transformation option	Description
Transform query to derived table	Specify whether to transform the query to a derived table - a subquery used as a table in a FROM clause.

Advanced SQL transformation

Advanced SQL transformation option	Description
Enable transformation that adds COALESCE (Default = enabled)	<p>Specify to apply the SQL syntax transformation rule that adds COALESCE to a column. When the data is retrieved, the COALESCE function, which in this case is not actually doing anything to change the value of the column, causes a full table scan or the database to pick another index to use. For example:</p> <pre>SELECT * FROM EMPLOYEE, DEPARTMENT WHERE COALESCE(DPT_ID, DPT_ID) = EMP_DEPT</pre>

Join Tables

Join Tables options	Description
Rewrite SQL using the same JOIN syntax as the source SQL (Default)	<p>Specify that the alternative SQL statements join the tables in the FROM clause using the same SQL syntax that is used in the original SQL statement. If the original SQL statement contains both syntax types, the optimization process rewrites the syntax using the Ansi-92 JOIN syntax. The outer join is not included in this conversion.</p>
Rewrite SQL using the Ansi-92 JOIN syntax	<p>Specify to use the JOIN clause from the Ansi-92 SQL standard when generating the SQL alternatives. During the optimization, the SQL statement is converted to the Ansi-92 SQL standard and then SQL syntax transformation rules are applied to rewrite the converted SQL statement. Next, the DB2 SQL Options are applied to the original SQL and the transformed SQL. So you can see SQL alternatives that use the JOIN syntax from the original SQL, but these SQL alternatives are simply the original SQL with a SQL Option applied.</p> <p>The outer join is not including in this conversion because Ansi-92 OUTER JOIN syntax does not always retrieve the same result set as the outer join using the (+) operator. So to avoid producing the wrong result set, the conversion of the outer join syntax cannot be applied.</p> <p>For example:</p> <pre>SELECT DPT_ID FROM EMPLOYEE INNER JOIN DEPARTMENT ON EMP_DEPT = DPT_ID</pre>
Rewrite SQL using none Ansi-92 JOIN syntax	<p>Specify to join tables in the FROM clause without the Ansi-92 JOIN syntax or using comma. The join analysis occurs in the WHERE clause which specifies the column in one table that is compared to a column in another table. During the optimization, the SQL statement is converted from the Ansi-92 SQL standard and then SQL syntax transformation rules are applied to rewrite the converted SQL. Next, the DB2 SQL</p>

Options are applied to the original SQL and the transformed SQL. So you may see SQL alternatives that use the JOIN syntax from the original SQL, but these SQL alternatives are simply the original SQL with a SQL Option applied.

The outer join is not including in this conversion because Ansi-92 OUTER JOIN syntax does not always retrieve the same result set as the outer join using the (+) operator. So to avoid producing the wrong result set, the conversion of the outer join syntax cannot be applied.

For example:

```
SELECT DPT_ID
FROM EMPLOYEE,
DEPARTMENT
WHERE DPT_ID = EMP_DEPT
```

Rewrite SQL using Ansi-92 JOIN and none Ansi-92 JOIN syntax

Specify to use the both types of SQL syntax for joining the tables. Each type of join syntax will result in a different alternative.

Related Topics

[Intelligence Options](#)

[Quota Options](#)

[SQL Options](#)

SQL Options

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

To set SQL Options

1. Click [Options](#).
2. Select **Optimize SQL | SQL Rewrite | SQL Options**
3. Review the following for additional information:

Set Current Degree	Specify whether to set the current degree of intra-partition parallelism. Note the following: <ul style="list-style-type: none">• The value ANY lets the database optimizer choose an appropriate degree depending on the resources.• The value 1 prevents the database optimizer from running queries concurrently.
Set Value for the "Optimize for" Option	Specify the number of rows to optimize. If this option is not selected, then it is assumed that all rows will be retrieved.

Related Topics

[Intelligence Options](#)

[Quota Options](#)

[Optimization Options](#)

Quota Options

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

To set quota options

1. Click [Options](#).
2. Select **Optimize SQL | SQL Rewrite | Quota**
3. Review the following for additional information:

Quota	Description
Syntax Transformation Quota	Specify the maximum number of SQL statements generated by applying SQL transformation rules. This quota affects how many SQL statements are created by changing the SQL syntax. The default quota of 100 is normally sufficient for most of the complicated SQL statements. However, the quota can be increased to tackle exceptionally complicated SQL statements with very high levels of table joins or multiple levels of nested sub-queries.
SQL Options Quota	Specify the maximum number of SQL statements to be generated by applying SQL options.
Total Quota	This is a read-only field indicating the maximum number of SQL statements generated during SQL Rewrite. This figure consists of: <i>Syntax Transformation Quota</i> + <i>SQL Options Quota</i> .

Related Topics

[Intelligence Options](#)

[SQL Options](#)

[Optimization Options](#)

Index Options

To set index options

1. Click [Options](#).
2. Select **Optimize SQL | Index Generation | Options**.
3. Review the following for additional information:

Option	Description
Retrieve index selectivity by sampling a maximum number of records:	Enter the maximum number of records to retrieve for selectivity sampling.
Maximum number of columns in a composite index:	Enter the maximum number of columns in a composite index. <i>Range: 1 to 99</i>
Maximum number of indexes in an index set:	Enter the maximum number of indexes in an index set. <i>Range: 1 to 99</i>
Evaluate columns in SELECT list:	Select to evaluate creating an index on columns in the SELECT list.
Quota	Description
Index Generation Quota:	Enter the maximum number of indexes to generate. <i>Range: 10 to 9999</i>
Index Set Generation Quota:	Enter the maximum number of index sets to generate by combining two or more indexes. <i>Range: 1 to 99,999</i>
Index Prefix	Description
Default prefix of index name:	Enter the prefix placed on the index name when SQL Optimizer generates index candidates. <i>Default: SQLOPT_IDX</i>

Related Topics

Index Setting Options

[Intelligence Options \(Index Generation\)](#)

Intelligence Options (Index Generation)

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

To set intelligence options

1. Click [Options](#).
2. Select **Optimize SQL | Index Generation | Intelligence**.

3. Select whether to use custom or predefined intelligence levels. If you select *predefined*, the levels range as follows:

Intelligence Level 1

Index Option

Option Section

Top percentage rows from table = 5

Minimum number of rows = 300

Maximum number of rows = 600

Maximum number of columns in a composite index = 2

Maximum number of indexes to generate per alternative = 2

Quota Section

Index Generation Quota = 20

Index Set Generation Quota = 30

Intelligence Level 2

Index Option

Option Section

Top percentage rows from table = 10

Minimum number of rows = 500

Maximum number of rows = 1000

Maximum number of columns in a composite index = 4

Maximum number of indexes to generate per alternative = 4

Quota Section

Index Generation Quota = 50

Index Set Generation Quota = 100

Intelligence Level 3

Index Option

Option Section

Top percentage rows from table = 15

Minimum number of rows = 800

Maximum number of rows = 1600

Maximum number of columns in a composite index = 5

Maximum number of indexes to generate per alternative = 5

Quota Section

Index Generation Quota = 80

Index Set Generation Quota = 160

Intelligence Level 4

Index Option

Option Section

Top percentage rows from table = 20

Minimum number of rows = 1500

Maximum number of rows = 3000

Maximum number of columns in a composite index = 6

Maximum number of indexes to generate per alternative = 6

Quota Section

Index Generation Quota = 200

Index Set Generation Quota = 400

Intelligence Level 5

Index Option

Option Section

Top percentage rows from table = 25

Minimum number of rows = 4000

Maximum number of rows = 8000

Maximum number of columns in a composite index = 8

Maximum number of indexes to generate per alternative = 8

Quota Section

Index Generation Quota = 500

Index Set Generation Quota = 1000

Related Topics

[Index Options](#)

[Index Setting Options](#)

Test Run Settings Options

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

To set test run settings options

1. Click [Options](#).
2. Select **Optimize SQL | Test Run | Test Run Settings**
3. Review the following for additional information:

Test Run Settings	Description
Custom	Enables you to customize the settings.
Predefined	Uses the predefined set of options.

Related Topics

[Best Alternative Criteria Options](#)

[Execution Method Options](#)

[Order and Termination Options](#)

Execution Method Options

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

To set execution method options

1. Click [Options](#).
2. Select **Optimize SQL | Test Run | Execution Method**
3. Review the following for additional information:

Execution Method	Description
Maximum rows to be retrieved (records)	Select this checkbox and then enter the maximum number of rows to retrieve for a test run of all records.
Run Time Retrieval Method	Description
Run SQL options	Select one of the following options: <ul style="list-style-type: none">• Run all SQL twice if original SQL runs faster than (seconds)—Combines the Original SQL twice and all others once and the All SQL twice options into one option and allows you to determine (by the number of seconds a SQL statement runs) which option to use. The original SQL statement always runs twice. The SQL alternatives run twice if the original SQL statement runs in less time than the value specified. Otherwise, the SQL alternatives all run once.• Original SQL twice and all others once—Caches data from a table into memory the first time you access it. The next time you access that data, it is already in memory so the following SQL statements run faster. To provide an accurate comparison, the first SQL statement runs twice but only the time from the second run is compared to the times for the other statements.• All SQL twice—Executes all SQL statements twice to eliminate factors that can affect the accuracy of the results. If you recently executed a SQL statement, the information for that statement may be cached and the statement may execute faster. This option eliminates time variation caused by caching since it runs all SQL statements twice but only uses the second run time for comparison.• All SQL once—Executes all SQL statements once. For long running SQL, you do not need to run any statement twice since the effect from caching diminishes over time.

Related Topics

[Test Run Settings Options](#)

[Best Alternative Criteria Options](#)

[Order and Termination Options](#)

Order and Termination Options

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

To set order and termination options

1. Click [Options](#).
2. Select **Optimize SQL | Test Run | Order and Termination**
3. Review the following for additional information:

Execution Order	Description
Execution order for SQL	Select one of the following options: <ul style="list-style-type: none">• Intelligent order—Executes representative SQL statements with various plan cost according to SQL Optimizer's intelligence engine.• Plan cost—Executes SQL statements in order of plan cost.
SQL Termination Criteria	Description
Terminate execution of SQL alternative if it runs longer than	Select one of the following options: <ul style="list-style-type: none">• Run time of fastest SQL—Cancels SQL statements that run longer than the current fastest run time. With this option, the first SQL statement runs and the time from that statement is used as the termination time for the next SQL statement. When a SQL statement runs faster than this time, the faster time is used as the new termination time.• This percentage of the original SQL run time—Cancels SQL statements whose total elapsed time is the specified % of the total elapsed time for the original SQL statement. It terminates all SQL statements that run longer than the calculated termination time.• User defined time (mins/secs)—Cancels SQL statements that run longer than a time you specify.• Run without termination—Runs all SQL statements to completion regardless of run time.
Cancellation delay (seconds)	Adds a specified time to the termination time. It is important to factor a delay into the overall termination time to account for the time needed to send the SQL statement to the database server.

Related Topics

[Test Run Settings Options](#)

[Best Alternative Criteria Options](#)

[Execution Method Options](#)

Best Alternative Criteria Options

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

To set the best alternative options

1. Click [Options](#).
2. Select **Optimize SQL | Test Run | Best Alternative Criteria**
3. Review the following for additional information:

Best SQL Alternative Selection Criteria	Description
Best alternative selected based on lowest	Select one of the following : <ul style="list-style-type: none">• Total Elapsed Time—Select elapsed run time as the criteria to find the best SQL alternative t.• First Row Time—Uses the time to retrieve the first record to find the best SQL alternative. <p><i>Default: Elapsed Time</i></p>

Related Topics

[Test Run Settings Options](#)

[Execution Method Options](#)

[Order and Termination Options](#)

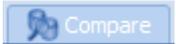
Tutorial: SQL Optimizer

Using 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. In the second step, SQL Optimizer executes the alternatives to test runs 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.

Step 1: Optimize the SQL Statement

1. Select the SQL Details tab in the main window.
2. Enter a SQL statement in the Alternative Details pane.
3. Click  to optimize the SQL statement
4. Select a connection. Review the following for additional information:

Select Connection	Description
Connection	Click  to select a previously created connection. Tip: Click  to open the Connection Manager to create a new connection. Connect to the Database
Select Schema and SQLID	Description
Schema	Click  to select a schema for the connection.
SQLID	Click  to select your SQLID.

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

Step 2: Test Run Alternative SQL Statements

The Test Run function provides an efficient way to benchmark alternatives generated by SQL Optimizer. You can test run 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.

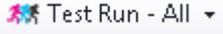
To test run a SQL statement alternative

1. Click .

2. Select **Optimize SQL | Test Run | Test Run Settings**.
3. Select **Custom Settings** or **Predefined Settings**.
4. Select **Optimize SQL | Test Run | Execution Method**.
5. Review the following for additional information:

Execution Method	Description
Maximum rows to be retrieved (records)	Select this checkbox and then enter the maximum number of rows to retrieve for a test run of all records.

Run Time Retrieval Method	Description
Run SQL options	<p>Select one of the following options:</p> <ul style="list-style-type: none"> • Run all SQL twice if original SQL runs faster than (seconds)—Combines the Original SQL twice and all others once and the All SQL twice options into one option and allows you to determine (by the number of seconds a SQL statement runs) which option to use. The original SQL statement always runs twice. The SQL alternatives run twice if the original SQL statement runs in less time than the value specified. Otherwise, the SQL alternatives all run once. • Original SQL twice and all others once—Caches data from a table into memory the first time you access it. The next time you access that data, it is already in memory so the following SQL statements run faster. To provide an accurate comparison, the first SQL statement runs twice but only the time from the second run is compared to the times for the other statements. • All SQL twice—Executes all SQL statements twice to eliminate factors that can affect the accuracy of the results. If you recently executed a SQL statement, the information for that statement may be cached and the statement may execute faster. This option eliminates time variation caused by caching since it runs all SQL statements twice but only uses the second run time for comparison. • All SQL once—Executes all SQL statements once. For long running SQL, you do not need to run any statement twice since the effect from caching diminishes over time.

6. Select **Optimize SQL | Test Run | Order and Termination**.
7. Select the **This percentage of the original SQL run time** checkbox.
8. Click .

The execution statistics display in the Alternatives pane once the test run completes. Select an alternative to see more information in the Alternative Details and Execution Plan panes.

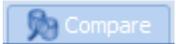
Tutorial: SQL Optimizer

Using 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. In the second step, SQL Optimizer executes the alternatives to test runs 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.

Step 1: Optimize the SQL Statement

1. Select the SQL Details tab in the main window.
2. Enter a SQL statement in the Alternative Details pane.
3. Click  to optimize the SQL statement
4. Select a connection. Review the following for additional information:

Select Connection	Description
Connection	Click  to select a previously created connection. Tip: Click  to open the Connection Manager to create a new connection. Connect to the Database
Select Schema and SQLID	Description
Schema	Click  to select a schema for the connection.
SQLID	Click  to select your SQLID.

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

Step 2: Test Run Alternative SQL Statements

The Test Run function provides an efficient way to benchmark alternatives generated by SQL Optimizer. You can test run 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.

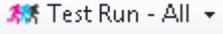
To test run a SQL statement alternative

1. Click .

2. Select **Optimize SQL | Test Run | Test Run Settings**.
3. Select **Custom Settings** or **Predefined Settings**.
4. Select **Optimize SQL | Test Run | Execution Method**.
5. Review the following for additional information:

Execution Method	Description
Maximum rows to be retrieved (records)	Select this checkbox and then enter the maximum number of rows to retrieve for a test run of all records.

Run Time Retrieval Method	Description
Run SQL options	<p>Select one of the following options:</p> <ul style="list-style-type: none"> • Run all SQL twice if original SQL runs faster than (seconds)—Combines the Original SQL twice and all others once and the All SQL twice options into one option and allows you to determine (by the number of seconds a SQL statement runs) which option to use. The original SQL statement always runs twice. The SQL alternatives run twice if the original SQL statement runs in less time than the value specified. Otherwise, the SQL alternatives all run once. • Original SQL twice and all others once—Caches data from a table into memory the first time you access it. The next time you access that data, it is already in memory so the following SQL statements run faster. To provide an accurate comparison, the first SQL statement runs twice but only the time from the second run is compared to the times for the other statements. • All SQL twice—Executes all SQL statements twice to eliminate factors that can affect the accuracy of the results. If you recently executed a SQL statement, the information for that statement may be cached and the statement may execute faster. This option eliminates time variation caused by caching since it runs all SQL statements twice but only uses the second run time for comparison. • All SQL once—Executes all SQL statements once. For long running SQL, you do not need to run any statement twice since the effect from caching diminishes over time.

6. Select **Optimize SQL | Test Run | Order and Termination**.
7. Select the **This percentage of the original SQL run time** checkbox.
8. Click .

The execution statistics display in the Alternatives pane once the test run completes. Select an alternative to see more information in the Alternative Details and Execution Plan panes.

We are more than just a name

We are on a quest to make your information technology work harder for you. That is why we build community-driven software solutions that help you spend less time on IT administration and more time on business innovation. We help you modernize your data center, get you to the cloud quicker and provide the expertise, security and accessibility you need to grow your data-driven business. Combined with Quest's invitation to the global community to be a part of its innovation, and our firm commitment to ensuring customer satisfaction, we continue to deliver solutions that have a real impact on our customers today and leave a legacy we are proud of. We are challenging the status quo by transforming into a new software company. And as your partner, we work tirelessly to make sure your information technology is designed for you and by you. This is our mission, and we are in this together. Welcome to a new Quest. You are invited to Join the Innovation.

Our brand, our vision. Together.

Our logo reflects our story: innovation, community and support. An important part of this story begins with the letter Q. It is a perfect circle, representing our commitment to technological precision and strength. The space in the Q itself symbolizes our need to add the missing piece — you — to the community, to the new Quest.

Contacting Quest

For sales or other inquiries, visit <https://www.quest.com/company/contact-us.aspx>.

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

Third-party contributions

This product contains some third-party components (listed below). Copies of their licenses may be found at referencing <https://www.quest.com/legal/license-agreements.aspx>. Source code for components marked with an asterisk (*) is available at <https://opensource.quest.com>.

Table 1: List of Third-Party Contributions

Component	License or Acknowledgement
DevExpress WinForms 16.1.4	jQuery JavaScript Library (Open Source - MIT License) Copyright jQuery Foundation and other contributors http://jquery.com/ jQueryUI JavaScript Library (Open Source - MIT License) Copyright jQuery Foundation and other contributors http://jqueryui.com/ Knockout JavaScript Library (Open Source - MIT License) Copyright Knockoutjs.com http://knockoutjs.com/ http://opensource.org/licenses/mit-license.php Globalize JavaScript Library (Open Source - MIT License) Copyright Software Freedom Conservancy, Inc. http://jquery.org/license Ace (Ajax.org Cloud9 Editor) (Open Source - BSD License) Copyright 2010, Ajax.org B.V. https://github.com/ajaxorg/ace/blob/master/LICENSE JS Beautifier (Open Source - MIT License) Copyright 2007-2013 Einar Lielmanis and contributors https://github.com/beautify-web/js-beautify/blob/master/LICENSE CodeMirror (Open Source - MIT License) Copyright 2015 Marijn Haverbeke https://codemirror.net/LICENSE

Copyright 2019 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

Quest Software is proud of our advanced technology. Patents and pending patents may apply to this product. For the most current information about applicable patents for this product, please visit our website at <https://www.quest.com/legal>.

Trademarks

Quest, and the Quest logo are trademarks and registered 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>. All other trademarks, servicemarks, registered trademarks, and registered servicemarks are the property of their respective owners.

Legend

-  **WARNING:** A WARNING icon indicates a potential for property damage, personal injury, or death.
-  **CAUTION:** A CAUTION icon indicates potential damage to hardware or loss of data if instructions are not followed.
-  **IMPORTANT, NOTE, TIP, MOBILE, or VIDEO:** An information icon indicates supporting information.