SQL Optimizer for Oracle® 10.0

User Guide

Contents

1

Introduction to SQL Optimizer

About SQL Optimizer for Oracle

SQL Optimizer for Oracle® automates the SQL optimization process and maximizes the performance of your SQL statements. SQL Optimizer for Oracle® analyzes, rewrites, and evaluates SQL statements located within database objects, files, or collections of SQL statements from Oracle's System Global Area (SGA). Once SQL Optimizer identifies problematic SQL statements, it optimizes the SQL and provides replacement code that includes the optimized statement.

SQL Optimizer also provides a complete index optimization and plan change analysis solution. It provides index recommendations for multiple SQL statements or a SQL workload, simulates index impact analysis, and generates SQL execution plan alternatives.

SQL Optimizer consists of the following modules:

Optimize SQL

Optimize SQL includes a SQL Rewrite mode and a Plan Control mode.

SQL Rewrite Mode	Description
Optimize SQL Statements	Uses SQL Optimizer's Artificial Intelligence engine to execute SQL syntax rules and apply Oracle optimization hints to create semantically equivalent SQL statement alternatives. In addition, you can create user-defined alternatives to test with your database environment. See "About Optimizing SQL" in the online help for more information.
Test Run SQL Alternatives	Test run SQL statement alternatives to view execution statistics. This provides execution times that allow you to identify the best SQL statement for your database environment. See "Execute Scenarios" in the online help for more information.
Generate Index Alternatives	Analyzes SQL statement syntax and database structure to provide index alternatives that improve performance. SQL Optimizer uses virtual indexes to generate alternatives without physically creating the indexes on your database. See "About Index Generation" in the online help for more information.
Test for Scalability	Uses Benchmark Factory™ to simulate potential workload conditions to test SQL statement performance. See "Test for Scalability" in the online help for more information.
Incorporate Best Practices	Incorporates common best practices techniques to improve database performance. See "Best Practices" in the online help for more information.

Plan Control Mode	Description
Generate Execution Plan Alternatives	Generates execution plan alternatives for your SQL statements without changing the original source code. See "Generate Execution Plan Alternatives" in the online help for more information.
Deploy Baselines	Creates baselines from the execution plan alternatives and deploys these baselines to ensure optimal database performance. See "Deploy Baselines" in the online help for more information.

Optimize Indexes

Optimize Indexes analyzes a SQL workload or any group of SQL statements and determines the best index set for the workload or group of statements. See "About Optimize Indexes" in the online help for more information.

Batch Optimize SQL

Batch Optimize submits files, database objects, or SQL text for batch processing. Batch Optimize scans and extracts the SQL statements, optimizes the statements, and tests the alternatives to find the best performing SQL statements for your database environment. See "About Batch Optimize" in the online help for more information.

Scan SQL

Scan SQL identifies problematic SQL statements in your source code and database objects without execution. Scan SQL then analyzes the problematic SQL statements and categorizes them according to performance levels. See "About Scanning SQL" in the online help for more information.

Inspect SGA

Inspect SGA analyzes SQL statements from Oracle's SGA. You can specify the criteria used to retrieve SQL statements and execution statistics to review SQL performance. See "About Inspect SGA" in the online help for more information.

Analyze Impact

Use Analyze Impact to evaluate the impact on a SQL workload from database changes by tracking execution plan and Oracle cost changes for SQL statements. You can run an impact analysis to estimate the performance impacts from parameter changes and new indexing. You can also run a comparison of two different databases that run the same application. See "About Analyze Impact" in the online help for more information.

Manage Plans

Manage Plans organizes stored baselines and outlines used to improve SQL statement performance. See "About Managing Plans" in the online help for more information.

SQL Optimization Workflow

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

Procedure	Description
Identify Problematic SQL	Batch Optimize SQL extracts embedded SQL statements from your database objects. After extracting the statements, it analyzes execution plan operations and identifies potential performance bottlenecks. **i Notes:** • You can use Inspect SGA to capture dynamic SQL statements. Save the captured dynamic SQL statements into an inspector file and use Batch Optimize SQL to extract the statements. • You can also use Scan SQL to extract embedded SQL statements.
Optimize SQL Statements	Once Batch Optimize SQL identifies problematic SQL statements, it automatically optimizes these statements and generates alternatives with unique execution plans. Batch Optimize SQL generates the alternatives by analyzing SQL statement syntax and database structure. You can also use hints during the optimization process. **i Note:** You can also use SQL Rewrite mode in Optimize SQL to optimize SQL statements extracted with Scan SQL.
Test Run SQL Alternatives	After Batch Optimize SQL generates alternatives, it will automatically test run the alternatives and provide you with the best statement for your database environment. **i Note:** Since Batch Optimize SQL automates the SQL optimization process, you are only provided with the best alternative statement it generates. You can send your statement to Optimize SQL to view all statement alternatives available.
Compare SQL Alternatives	Batch Optimize SQL compares the SQL text and execution plans of your original SQL statement and the best alternative. **i Note:** If you send your statement to Optimize SQL, you can compare your original SQL statement with any of the statement alternatives available.
Replace Problematic SQL Statements	Batch Optimize SQL creates a script that you can use to replace your original source code.
Generate Index Alternatives	In addition to optimizing SQL statements, you can generate index alternatives for your statement in Optimize SQL. You can generate new indexes for a group of SQL statements or for a SQL workload using Optimize Indexes.
Generate Execution Plan Alternatives	You can use Plan Control mode in Optimize SQL to generate execution plan alternatives for your SQL statements without changing the original source code. Plan Control mode finds the best execution plan for your SQL statement and deploys it as a baseline.
Analyze Performance Changes	Analyze Impact evaluates what impact certain changes, such as indexing and parameter changes, will have on SQL statement performance.

Database Privileges

Oracle database privileges limit access for individual users. The following list summarizes the functions in SQL Optimizer that require specific Oracle database privileges.

Module	Functionality	Privilege
All Modules	Trace setup options: Enable collection of Oracle trace statistics	Requires ALTER SESSION privileges. Requires access to the following views: SYS.V_$SESSION SYS.V_$PROCESS
All Modules	Retrieve DBMS_XPLAN	Requires access to the SYS.DBMS_XPLAN package.
All Modules	General	If the Oracle init parameter O7_DICTIONARY_ACCESSIBILITY for Oracle 8 or later is set to false, you cannot access objects under SYS even if you have SELECT ANY TABLE privileges. In this case, you need SELECT ANY DICTIONARY privileges or SELECT_CATALOG_ROLE to access the objects under SYS.
Optimize SQL (SQL Rewrite)	Alter session parameters for executing SQL	Requires access to SYS.V_$PARAMETER view.
	Generate virtual indexes	Requires Oracle 8i or later.
	Check existing translation for your SQL	Requires Oracle 12c or later. Requires access to the SYS.ALL_SQL_TRANSLATIONS view.
	Create SQL Translation Profile	Requires Oracle 12c or later. Requires CREATE SQL TRANSLATION PROFILE privileges.
	Deploy outlines	Requires Oracle 8i or later. Requires CREATE ANY OUTLINE and DROP ANY OUTLINE privileges. Requires access and UPDATE privileges to the following views: OUTLN.OL$HINTS OUTLN.OL$ OUTLN.OL$NODES
	Register SQL Translation	Requires Oracle 12c or later. Requires access to the SYS.DBMS_SQL_TRANSLATOR package. Requires access to the SYS.ALL_SQL_TRANSLATION_PROFILES view.
Optimize SQL (Plan Control)	Open database connection	Requires Oracle 11g or later.
	Retrieve execution plans,	Requires ADMINISTER SQL MANAGEMENT OBJECT privileges.

Module	Functionality	Privilege
	generate alternative plans	Requires access to the following packages: SYS.DBMS_SQL SYS.DBMS_SPM SYS.DBMS_XPLAN Requires access to the following views: SYS.DBA_SQL_PLAN_BASELINES SYS.V_$SQLAREA SYS.V_$SQLTEXT_WITH_ NEWLINES
Optimize SQL and Batch Optimize SQL	Execution method option: Run on server setting	Requires access to the SYS.DBMS_SQL package.
	Retrieve run-time statistics	Requires access to the following views: SYS.V_$MYSTAT SYS.V_$STATNAME SYS.V_$PARAMETER
	Retrieve actual plan	Requires ALTER SESSION privileges. Requires access to the SYS.DBMS_XPLAN package. Requires access to the following views: SYS.V_$SQLAREA SYS.V_$ SQL_PLAN_STATISTICS_ALL SYS.V_$SESSION
	Capture bind values from database	Requires Oracle 10g or later. Requires access to the following views: SYS.V_$SQLAREA SYS.V_$SQL_BIND_CAPTURE
Optimize Indexes	Recommend indexes	Requires Oracle 8i or later. Requires access to the SYS.V_$SESSION view.
	Access the AWR	Requires Oracle 10g or later. Requires access to the following system views: SYS.DBA_HIST_SNAPSHOT SYS.DBA_HIST_SQLTEXT SYS.DBA_HIST_SQLSTAT
	Display the control information for the Workload Repository	Requires access to the following system view: SYS. DBA_HIST_WR_CONTROL
	Display the SQL summary for the Workload Repository	Requires access to the following system view: SYS. DBA_HIST_SQL_SUMMARY
	Access the SGA	Requires access to the SYS.V_$SQLAREA view.

Module	Functionality	Privilege
Inspect SGA	SQL to collect: Executed SQL from SQL area	Requires access to the following views: SYS.V_$SQLAREA SYS.V_$SQLTEXT_WITH_NEWLINES (or SYS.V_$SQLTEXT depending on your version of Oracle) Requires access to SYS.V_$SQL_PLAN view in Oracle 9 or later.
	SQL to collect: Currently executing SQL	Requires access to the following views: SYS.V_$OPEN_CURSOR SYS.V_$SESSION SYS.V_$SQLAREA SYS.V_$SQLTEXT_WITH_NEWLINES (or SYS.V_$SQLTEXT depending on your version of Oracle) Requires access to SYS.V_$SQL_PLAN view in Oracle 9 or later.
	Flush Oracle shared pool	Requires ALTER SYSTEM privileges.
	Execution plan information	Requires access to SYS.V_$SQL_PLAN view in Oracle 9 or later.
	Monitor by session	Requires access to the SYS.V_$SESSION view.
Manage Plans (Baselines Management)	Open a Baselines Management session or database connection	Requires Oracle 11g or later.
	View baselines	Requires access to the SYS.DBA_SQL_PLAN_BASELINES view.
	View baseline plans	Requires access to the SYS.DBA_SQL_PLAN_BASELINES view. Requires access to the SYS.DBMS_XPLAN package.
	Import, export, and migrate baselines	Requires ADMINISTER SQL MANAGEMENT OBJECT and CREATE TABLE privileges. Requires access to the SYS.DBA_SQL_PLAN_BASELINES view. Requires access to the SYS.DBMS_SPM package.
	View and modify baseline configuration values Obtaining the Oracle parameter values OPTIMIZER_CAPTURE_SQL_PLAN_	Requires ADMINISTER SQL MANAGEMENT OBJECT and ALTER SYSTEM privileges. Requires access to the SYS.DBMS_SPM package. Requires access to the following views: SYS.V_$SYSTEM_PARAMETER SYS.DBA_SQL_MANAGEMENT_CONFIG SYS.DBA_DATA_FILES

Module	Functionality	Privilege
	BASELINES and OPTIMIZER_USE_SQL_PLAN_BASELINES	
Manage Plans (Outlines Management)	Open an Outlines Management session or database connection	Requires Oracle 8i or later.
	View outlines	Requires OUTLN.OL$HINTS (SELECT only) and OUTLN.OL$ (SELECT only) privileges.
	Modify outline configuration values	Requires ALTER SYSTEM and OUTLN.OL$ (SELECT only) privileges.
	Modify outlines	Requires DROP ANY OUTLINE and ALTER ANY OUTLINE privileges. Requires access to the SYS.OUTLN_PKG package.
Analyze Impact	Virtual indexes for Index Impact	Requires Oracle 8i or later. Requires access to the following system view: SYS.V_$SESSION
	Altering session parameters for Parameter Impact	Requires access to the following system view: SYS.V_$PARAMETER
	Access AWR	Requires Oracle 10g or later. Requires access to the following system views: SYS.DBA_HIST_SNAPSHOT SYS.DBA_HIST_SQLTEXT SYS.DBA_HIST_SQLSTAT
	Display the control information for the Workload Repository	Requires access to the following system view: SYS. DBA_HIST_WR_CONTROL
	Display the SQL summary for the Workload Repository	Requires access to the following system view: SYS. DBA_HIST_SQL_SUMMARY
	Access SGA	Requires access to the following system view: SYS.V_$SQLAREA
	Display database information for Compare Database Impact	Requires access to the following system views: SYS.PRODUCT_COMPONENTS_VERSION SYS.V_$PARAMETER SYS.NLS_SESSION_PARAMETERS SYS.NLS_INSTANCE_PARAMETERS

Module	Functionality	Privilege
		SYS.NLS_DATABASE_PARAMETERS SYS.V_$SGAINFO SYS.V_$DATABASE SYS.V_$INSTANCE SYS.V_$STATISTICS_LEVEL SYS.V_$OPTION

Use SQL Optimizer

Tutorial: Optimize SQL (SQL Rewrite)

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

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

Step 1: Optimize the SQL Statement

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

2. Select **SQL Rewrite** from the Optimize SQL start page.

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

3. Enter a SQL statement in the Alternative Details pane.

4. Click to retrieve the execution plan for your SQL statement.The Select Connection and Schema window displays.

5. Select a connection and schema to use.

6. Click to optimize the SQL statement.

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

Step 2: Test Alternative SQL Statements

To test the alternatives SQL Optimizer generates, you can test run one or more 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 run in a transaction that is rolled back after execution.

To test a SQL statement alternative
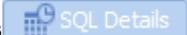
1. Click **SQL Details** [SQL Details] after you finish comparing your original SQL statement with the alternatives generated.

2. Click the drop-down arrow beside [icon] and select **Test Run - All** to execute all SQL alternatives.

3. The Test Run Settings dialog opens. Select criteria to apply to this test run. You can answer the questions on the Usage and Symptom page to allow SQL Optimizer to determine settings for you. Or you can click the **Customize Test Run Settings** link at the bottom of the page to manually specify test run settings.

   i **Tip:** You can specify default values for some Test Run Settings options by clicking **Options** [Options] and selecting **Optimize SQL | Test Run**.

4. Review the execution statistics in the Alternatives pane.

Tutorial: Optimize SQL (Plan Control)

Using Plan Control mode in Optimize SQL consists of two steps. In the first step, SQL Optimizer generates execution plan alternatives for your SQL statement without changing the source code. You can then execute the alternatives to retrieve run time statistics and identify the best alternative for your database environment. In the second step, you can use Plan Control mode to deploy the execution plan to the Manage Plans module as an Oracle plan baseline.

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

Step 1: Generate and Execute Execution Plan Alternatives

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

2. Select Plan Control from the Optimize SQL start page.

    i **Note:** If the start page does not display, click the arrow beside ![icon] and select **New Plan Control Session**.

3. Enter a SQL statement in the Original SQL pane.

    i **Tip:** Select the **This SQL is contained inside a PL/SQL block** checkbox if your SQL statement originated from a PL/SQL block. Selecting this checkbox ensures that the SQL text for the baseline you create matches the SQL text in your database.

4. Click ![icon] to generate alternative execution plans for your SQL statement. The Select Connection and Schema window displays.

5. Select a connection and schema to use.

6. Click ![icon] to execute all alternative execution plans to retrieve run time statistics.

7. Review the run time statistics in the Plans pane to identify the best alternative.

Step 2: Deploy Execution Plan as a Baseline

1. Click **Deploy** ![Deploy button].

2. Review the following for additional information:

Deploy	Description
Select a plan to deploy	Click ![icon] and select an execution plan alternative to deploy as a baseline plan.
Performance comparison	Use this to review the performance of selected plans against the original.
Mark the plan as	Review the following for additional information: • Enabled—Select whether to enable or disable this plan.

Deploy	Description
	• 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.
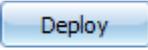
3. Click **Deploy** ⬚Deploy⬚ to deploy the plan to Manage Plans.

Tutorial: Generate Indexes in Optimize SQL

SQL Optimizer identifies columns to use as index alternatives for a SQL statement after it analyzes SQL syntax, relationships between tables, and selectivity of data. SQL Optimizer then combines identified alternatives into index sets.

To generate an index alternative

1. Select the **Optimize SQL** tab in the main menu.

2. Enter a SQL statement in the Alternative Details pane.

3. Click . The Select Connection and Schema window displays.

4. Select a connection and schema to use.

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

6. Select an index set to view index name, details and generation script in the Alternative Details pane.

7. To test an index, select it in the Alternatives pane and click .

   i **Note:** The Test Run function allows you to test an index set SQL Optimizer generated. It physically creates the indexes on the database, runs the SQL statement, retrieves execution statistics, and drops the indexes. Since this process physically creates indexes on your database, it may impact performance of other SQL statements.

Tutorial: Best Practices

You can use Best Practices to analyze your SQL statement and database to recommend common techniques for improving database performance. Since the recommendations can also affect performance of other statements in your database, you should review and test the recommendations before implementing them. When evaluating the recommendations, take into account that database performance is affected by the following:

- System resources (CPU, I/O, memory, database architecture, and more)

- Data distribution

- System architecture

- SQL execution plans

- User's usage behavior

i **Note:** The Best Practices function is only available in SQL Rewrite mode in Optimize SQL.

To view best practices

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

2. Click **Best Practices** .

   i **Tip:** To display the best practices tab, click **Options** , select **Optimize SQL | Best Practices | General**, and select the **Display Best Practices tab in SQL Rewrite mode** checkbox.

3. Enter a SQL statement into the Alternative Details pane.

4. Click . The Select Connection and Schema window displays.

5. Select a connection and schema to use.

6. Review the recommendations provided.

Tutorial: Deploy Outlines

The Deploy Outline function in Optimize SQL improves SQL statement performance without changing your original source code. Using Optimize SQL, you can generate SQL statements that are semantically equivalent to your original SQL statement with alternative execution plans. Once you identify the best alternative for your database environment, you can deploy it as a stored outline to use with your original statement.

To deploy an outline

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

2. Click the arrow beside and select **New SQL Rewrite Session**.

3. Enter your original SQL statement in the Alternative Details pane and click . The Select Connection and Schema window displays.

4. Select a connection and schema to use.

5. Right-click the alternative you want to deploy as an outline in the Alternatives pane and select **Deploy Outline**. The Deploy Outline window displays.

6. Review the following for additional information:

Outline name	Enter a name for the stored outline
Category	Click and select a previously created category or enter a new category name. **i Notes:** • The default category name is SQL_OPTIMIZIER. • You can add the outline to a disabled category until you test the SQL statement with and without using the outline.

7. Click **Deploy** .
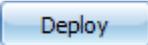
i **Note:** You can use the Outline Management feature in Manage Plans to enable and disable categories or to move outlines to different categories.

Tutorial: Batch Optimize SQL

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

To batch optimize SQL

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

2. Click **Add Code to Optimize** in the Batch Job List pane and select **All Types**. The Add Batch Optimize SQL Jobs window displays.

3. Review the following for additional information:

Connection Page	Description
Connection	Click to select a previously created database connection. **i Tips:** • Click to open the Connection Manager to create a new connection. • You can select an alternative connection for executing the SQL statement alternatives Batch Optimize generates.
Database Objects Page	**Description**
Database objects	Select a schema, database object type, or individual database object, and then click to add the object. **i Tip:** • Click to browse for database objects. • Your database privileges determine if you can scan all selected database objects.
Execute using schema	Click and select an alternative schema for executing the SQL statement alternatives.
Source Code Page	**Description**
Source code type	Select **Text/Binary files**, **Oracle SQL *Plus Script**, or **COBOL programming source code** to indicate the source code type for the file or directory you want to scan.
Add by file	Click and browse to the files you want to add.
Add by directory	Click and browse to the directories you want to add. **i Note:** Select the **Include Sub-directory** checkbox to scan sub-directories.
Scan using schema	Click and select a schema to scan.

Execute using schema	Click and select an alternative schema for executing the SQL statement alternatives.
SQL Text Page	**Description**
SQL text	Enter SQL statement text.
Scan using schema	Click and select a schema to scan.
Execute using schema	Click and select an alternative schema for executing the SQL statement alternatives.
Scan SQL Page	**Description**
Group	Select the Scanner group that contains the SQL statements you want to scan.
Scan using schema	Click and select a schema to scan.
Execute using schema	Click and select an alternative schema for executing the SQL statement alternatives.
Inspect SGA Page	**Description**
Group	Select the Inspector group that contains the SQL statements you want to scan.
Scan using schema	Click and select a schema to scan.
Execute using schema	Click and select an alternative schema for executing the SQL statement alternatives.

> **i** **Note:** Batch Optimize helps you manage jobs by organizing them into batches. Use the Batch Info page to create a new batch or to add the current job to an existing batch.

4. Click **Finish** to start batch optimization.

   Batch Optimize SQL scans the job you created, classifies and optimizes the statements, and executes the SQL statement alternatives it generates.

   > **i** **Notes:**
   >
   > - Scanning starts automatically if you select the **Automatically start extracting SQL when job is added** checkbox in the Batch Optimize SQL options page. Batch Optimize SQL selects this checkbox by default.
   >
   > - Batch Optimize SQL selects the SQL statement to optimize based on the classification types selected in the Batch Optimize SQL options page. Batch Optimize SQL selects Problematic and Complex SQL classification types by default.
   >
   > - Batch Optimize SQL executes SQL alternatives it generates based on the statement types selected in the Batch Optimize SQL options page. Batch Optimize SQL selects SELECT statements by default.

5. Select Batch List in the Batch Job List pane to view information about the jobs you created.

   The Batch List pane sorts information about your jobs by batches. Additional information displays in the Jobs Improved pane.

6. Select a batch from the batch list node to see details for the batch in the Job List pane.

The Job List pane displays the type of job, job status, and time of improvement for each job in the batch. Additional information displays in the SQL Classification and Cost and Elapsed Time Comparison panes for a selected job.

> **i** **Tip:** Select a job in the Job List pane and click to generate a replacement script with the optimized SQL statement.

7. Select a job from the batch node to see details of the job.

   The SQL List pane displays SQL classification information for the SQL statements in the job you select. The Original SQL Text and Best Alternative SQL Text panes allow you to compare your original SQL statement with the best alternative Batch Optimize SQL generated.

   > **i** **Tip:** Select a SQL statement in the SQL List pane and click to send the statement to Optimize SQL and view all SQL alternatives.

Tutorial: Scan SQL

Scan SQL helps you identify problematic SQL statements in your database environment by automatically extracting statements embedded in database objects, stored in application source code and binary files, or captured from Oracle's System Global Area. Scan SQL retrieves and analyzes execution plans for the extracted statements and classifies them according to complexity. You can then send statements that Scan SQL classifies as problematic or complex to Optimize SQL.

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

To scan SQL

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

2. Click 🔽 to select a previously created group or click ⋯ to create a new group for your scan job.

    ℹ **Note:** Scan SQL helps you manage scan jobs by organizing them into groups.

3. Click 🖼. The Add Scanner Jobs window displays.

4. Review the following for additional information:

Page	Description
Database Objects Page	
Database objects	Select a schema, database object type, or individual database object, and then click ▶ to add the object. ℹ **Tip:** Click ⋯ to browse for database objects.
Source Code Page	
Source code type	Select **Text/Binary files**, **Oracle SQL *Plus Script**, or **COBOL programming source code** to indicate the source code type for the file or directory you want to scan.
Add by file	Click ➕ and browse to the files you want to add.
Add by directory	Click ➕ and browse to the directories you want to add. ℹ **Note:** Select the **Include Sub-directory** checkbox to scan sub-directories.
Scan using schema	Click 🔽 and select a schema to scan.
Inspect SGA Page	
Group	Select the Inspector group that contains the SQL statements you want to scan.
Scan using schema	Click 🔽 and select a schema to scan.

5. Click **Finish** to start scanning.

6. Select a scan job from the Job List pane to view additional information.

Details displayed in the Job List pane include the number of SQL statements found and the classification for each statement.

> **i** **Tip:** Click ⌄ and select a different group to display scan jobs from a different group.

7. Select a SQL statement in the SQL List pane to view additional information for the selected statement in the SQL Text and Execution Plan panes.

> **i** **Tip:** Click **Send to Optimize SQL by SQL Rewrite** 🔶 Send to Optimize SQL by SQL Rewrite ▾ in the SQL Text pane to send the selected statement to Optimize SQL.

Tutorial: Inspect SGA

Inspect SGA retrieves executed SQL statements from Oracle's System Global Area or currently running SQL statements from Oracle's open cursor. Once you retrieve the statements, Inspect SGA displays the statements and their run time statistics so you can identify resource intensive statements in your database environment.

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

To retrieve a previously executed SQL statement

1. Select the Inspect SGA tab in the main window.

   i **Note**: To retrieve previously executed SQL statements, you must have privileges to view SYS.V_ $SQLAREA and either SYS.V_$SQLTEXT_WITH_NEWLINES or SYS.V_$SQLTEXT.

2. Click [icon] to select a group or click [icon] to create a new group in the **Group** list.

3. Click [icon]. The Add Inspect SGA Job wizard displays.

4. Complete the following fields in the wizard:

General Information Page	Description
Job type	Select the **Executed SQL from SQL Area** option.
Collecting Criteria Page	**Description**
Collecting Criteria	Select the **Top n records** option and enter the number of records to display.
First by	Click [icon] and select the statistic to use to extract SQL statements if you are not displaying all records. **i** **Note:** A large SGA increases processing time.
Collection Time Page	**Description**
Collection Time	Select the **Start collecting when you click the Inspect button** option.

5. Click [icon] to retrieve the SQL statements and run time statistics.

6. Select a statement that requires optimization in the SQL Statistics pane and click [icon] to send to Batch Optimize SQL or click the arrow beside **Send to Optimize SQL by SQL Rewrite** [button: Send to Optimize SQL by SQL Rewrite ▾] and select an option.

   i **Tip:** You can add an Inspect SGA job in Batch Optimize SQL to optimize all the SQL statements in the collection.
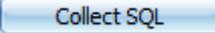
Tutorial: Analyze Impact

Use Analyze Impact to evaluate the impact that a change can have on a group of SQL statements or a SQL workload. The types of changes Analyze Impact will allow you to evaluate include: the addition of indexes and changes to database parameters. You can also run a comparison of two different databases that run the same application.
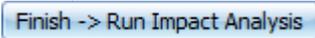
You can collect the SQL workload from one of several sources: source code, an Oracle Automatic Workload Repository (AWR), or the Oracle System Global Area (SGA).

To perform an impact analysis

1. Select Analyze Impact.

2. Select the type of change you want to analyze. For this tutorial, select **Additional Indexes**.

3. Select the database connection and schema to use.

4. Click to add a new index.

5. Enter a new index name or use the default.

6. In the Select Index Columns and Options pane, select the table containing the columns you want to index. The table's columns display in the lower pane. Select a column and click to move it to the Indexed Columns list.

7. Click to create a functional index.

8. Specify the remaining index criteria for the selected index. Review the following for additional information:

Index Type	Select an index type.
Advanced Options	Click to select from additional options for the selected index.
Compress Key	Select to add key compression. If you select *Compress Key*, you must also include the number of columns to compress in the **Columns** field.
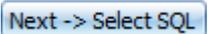
9. Click **Next -> Select SQL** [Next -> Select SQL] to start collecting the SQL to evaluate.

10. Select the source from which you want to collect SQL. For this tutorial, select **SGA**.

11. On the SQL collection page, click the **edit** link beside the parsing schema name. All the available parsing schemas are displayed in the graph. Select one of the parsing schemas by clicking it's bar in the graph. Click the bar a second time to deselect it.

12. Click the **edit** link beside **Module**. Notice that all available modules (for the selected schema) display in the graph. You can filter the SQL to collect by selecting a particular module or you can select **all** to collect SQL from all modules.

    i **Note:** Use the mouse pointer to hover over a bar in the graph to display a descriptive tooltip containing SQL workload details to help you select a workload.

13. Click the **edit** link beside **Action**. Notice that all available actions (for the selected schema) display in the graph. You can filter the SQL to collect by selecting a particular action or you can select **all** to collect SQL from all actions.

14. When you are finished selecting a SQL workload, click **Collect SQL** [Collect SQL] to start the collection process.

15. SQL Optimizer collects the specified SQL statements. The SQL page opens displaying the list of SQL statements collected.

16. In the SQL Workload pane, select a SQL statement to display the statement text and the parsed execution plan.

17. You can remove a SQL statement from the workload by deselecting (clearing) the checkbox in the Include column.

18. When you are satisfied with the SQL workload, click **Finish->Run Impact Analysis**
    Finish -> Run Impact Analysis to start the analysis process.

19. When the analysis process is finished, one of the following will display, depending on the analysis results:

    - If an impact was found, the Impacts page displays. See "Review Impact Analysis Results" in the online help for more information.

    - If the analysis determines that non of the specified SQL were impacted, the log page displays and reports "No impacted SQL found."
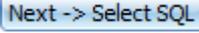
Tip: Use and to navigate back and forth between the pages of a tab.
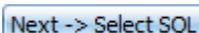
Tutorial: Send Indexes to Analyze Impact

You can analyze the impact of new indexes on your SQL statement's execution plans before you physically create the indexes on your database. You can create indexes in Optimize SQL or Optimize Indexes and then send the index or indexes to Analyze Impact to evaluate the impact on your SQL workload.

Use the Optimize SQL module to generate index alternatives for a *single* SQL statement. Or use the Optimize Indexes module to generate index alternatives for a *SQL workload* or *group* of SQL statements.

To send indexes to Analyze Impact from Optimize SQL

1. Generate index alternatives using the Index Generation feature in Optimize SQL. See Tutorial: Generate Indexes in Optimize SQL on page 16 for more information.

2. Select the virtual index alternative you want to use for the analysis in the Alternatives pane in Optimize SQL.

3. Click ⟳ in Optimize SQL. A new Analyze Impact session opens.

4. The Indexing Change Details page is populated with the new index that you sent from the Optimize SQL session. Use this page to modify the index, if necessary. See "Analyze the Impact of Additional Indexes" in the online help for more information.

5. When you have finished modifying the index, click **Next -> Select SQL** `Next -> Select SQL` to start collecting the SQL to evaluate. See "Collect SQL and Run an Impact Analysis" in the online help for more information.

To send indexes to Analyze Impact from Optimize Indexes

1. Generate index alternatives in Optimize Indexes. See "About Optimize Indexes" in the online help for more information.

2. In the Optimize Indexes session, select the Results tab.

3. On the Results page, in the Results Summary pane, click **Send to Analyze Impact**. A new Analyze Impact session opens.

4. The Indexing Change Details page is populated with the new indexes that you sent from the Optimize Indexes session. Use this page to modify the index or indexes, if necessary. See "Analyze the Impact of Additional Indexes" in the online help for more information.

5. When you have finished modifying the indexes, click **Next -> Select SQL** `Next -> Select SQL` to start collecting the SQL to evaluate. See "Collect SQL and Run an Impact Analysis" in the online help for more information.

Tutorial: Manage Outlines

Outline Management displays stored outlines deployed using SQL Rewrite mode in Optimize SQL.

To manage outlines

1. Select the Manage Plans tab in the main window.

   > ℹ **Tip:** Select the **Show Manage Plans** checkbox on the Manage Plans options page to display the Manage Plans tab in the main window.

2. Click **Manage Plans**. The Create a New Manage Plans Session window displays.

3. Select a connection to use.

4. Select the Outlines Management tab.

5. Select a category in the Category/Outline pane.

   You can delete or rename the selected category.

6. Select a stored outline from the category node.

   You can move, rename, or reset the Used Flag on the selected stored outline.

Optimize Indexes for SGA Workload

If you selected the SGA as a workload source, define the new workload and specify the search process criteria.

Define a new SGA Workload

1. If you have already created a new session, skip to 3. Otherwise, select **Optimize Indexes**.

2. Click **SGA** [SGA].
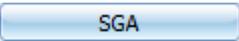
3. Enter a SQL workload name and select a database connection from which to collect SQL.

4. Specify SQL filter criteria. To modify filter criteria, click the **Edit** link beside each filter. The available options display as bars in the graph. Click an option (bar) in the graph to select it. The selected option then displays in the filter criterion field in the top pane. Review the following for additional information:

SQL filter criteria pane	Description
Collect SQL for parsing schema	You must select a parsing schema before collecting SQL.
Module, Action	Use these fields to filter the available SQL statements. *Default = All*

   i **Notes:**

   - Use the cursor to hover over an option (bar) in the graph to display option details.

   - Click **List statistics data** to display the options for the currently selected filter criterion in a grid format.

   - You can display additional data in the graph (or in the grid). To do this, select/deselect data options from the group of options at the bottom of the page.
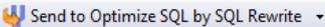
5. To begin collecting SQL, click **Configure Search Process** [Configure Search Process].
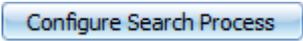
6. Optimize Indexes immediately asks you to choose a second database connection. Select a database connection to use to retrieve execution plans and evaluate index alternatives.

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

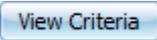7. The Search Process page opens. Before specifying search process criteria, you can review and modify your SQL workload.

   - Review collected SQL. This step is optional. You can modify the SQL workload if you want.

   - Specify search process criteria.

(Optional) Review collected SQL

1. To review and modify your SQL workload, click **Review/adjust workload SQL**. Review the following for additional information:

Pane / Button	Description
SQL Tuning Set Pane / SQL Workload Pane	This pane displays the list of SQL in the workload. Include—To include a SQL statement in the optimize indexes process, select the checkbox in the Include column.
SQL Text Pane	This pane displays the SQL text of the selected SQL statement.
Parsed Execution Plan Pane	This pane displays the execution plan of the selected SQL statement. SQL Statistics—Select this tab to display statistics of the selected statement.
Send to Optimize SQL by SQL Rewrite Send to Optimize SQL by SQL Rewrite ▾	Select to send your SQL to Optimize SQL or Batch Optimize SQL. Click the arrow for more options.

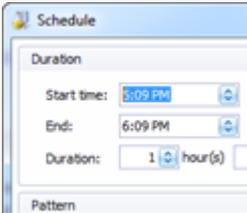**i** **Notes:**

- Click **View Criteria** View Criteria to return to the SQL Collection page. (On the SQL Collection page, click **Review SQL** to go back to the SQL Review page.)

- To collect a new SQL workload, return to the SQL Collection page and click **Collect SQL again**.

  ! **Caution: If you click Collect SQL again, your current SQL workload will be lost, as well as any analysis data already gathered for the current session.**

2. After reviewing the collected SQL, select the **Search Process** tab to return to the Search Process page.

Specify search process criteria

1. With the Search Process page open, locate the Search Process Control pane. Use these options to specify criteria for SQL Optimizer to use to find new indexes. Review the following for additional information:

Search Process Control Pane	Description
Primary goal (minimize)	Select a primary goal. SQL Optimizer evaluates indexes to optimize the workload based on this primary goal.
Scheduled to run on	Click the link to schedule a time to run the optimization process on this workload. Select a start time, end time, duration, and interval. Schedule Duration Start time: 6:09 PM End: 6:09 PM Duration: 1 hour(s) Pattern
Stop condition(s)	Select criteria to determine when to end the optimization process. You can select one or more conditions. The default setting is **When no further improvement found in 1:00:00**. **i** **Note:** If you do not specify a **Stop condition** or an **End date**, the optimization process will run indefinitely.
Advanced options	Click this link to select the number of columns in a composite

Search Process Control Pane	Description
	index, the maximum number of indexes to recommend, and an index-type time distribution.
	Recommend function-based indexes—Select this checkbox if you want SQL Optimizer to include function-based indexes.
	Search time distribution between B-Tree and Bitmap indexes—Select a pre-defined or user-defined time distribution. Determines the percentage of time spent searching for B-Tree vs Bitmap index types.
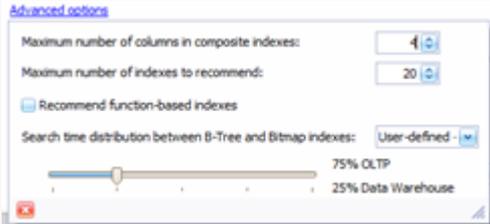
2. Click **Start** to begin searching for new indexes to improve the workload performance.

Review Optimize Indexes Results

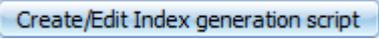You can review the results or the progress of your search for new indexes in Optimize Indexes.

To View Optimize Indexes Search Results

1. If you already have a session open, skip to 3. Otherwise, select **Optimize Indexes**.

2. Click **Open a Saved SQL Workload** and select a workload session. Click **Open** to open the session.

   > **i** **Note:** You can view a summary of the results, or the progress of the search if it is ongoing, in the bottom pane of the Open SQL Workload window.

3. Select the **Search Process** tab to review details related to the search process. Each line on the graph indicates a new indexing recommendation.

   > **i** **Notes:**
   >
   > - Use your cursor to hover over a line on the graph to display a summary of the recommendation's improvements (in a tooltip). Then click **View details** (in the tooltip) to open the Results page for the *selected* recommendation.
   > - Select **All** in the Display Range box, to display all index recommendations in the graph.
   > - Click **List all results** to display all index recommendations and their cost improvement values in a grid format. Click a recommendation in the list to open the results page for the *selected* recommendation.

4. Select the **Results** tab to view the final recommendation (or the last recommendation if the search process is ongoing).

   Review the following for additional information:

Pane / Option / Button	Description
n Recommended Indexes	Displays a list of the indexes for the recommendation.
	Index Name—Click an Index name to see the index details,

Pane / Option / Button	Description
	including a list of existing indexes which you can use to check for duplication. ℹ **Tip:** Select (highlight) an index in the Recommended Indexes list to highlight the SQL statements in the Impacted SQL list that are impacted by that recommended index.
Results Summary	(left side)—Displays a summary of workload improvement results. ℹ **Note:** If the links at the bottom of the Results Summary pane are not in view, click and drag the pane's bottom edge down to resize. (right side)—Displays the workload timeslots. Use the cursor to hover over a timeslot to display a summary of the improvement to that timeslot's SQL. ℹ **Note:** Timeslots display only for workloads collected from snapshots.
View	Displays one of two different evaluations for the Workload: • Plan Statistics view—Displays the estimated improvements based on changes in plan statistics and weighed by the number of executions for each SQL per timeslot. • Projected Actual Statistics view—Displays the estimated improvements calculated using actual statistics and changes in plan statistics.
Publish Report	Click to view the Optimize Indexes Workload Report which you can customize and then export, print, or email.
Send to Analyze Impact	Click to send the results to the Analyze Impact module where you can evaluate the impact of the new indexes on other SQL.
Chart Options	Click this link to select which charts/statistics to display.
Impacted SQL	SQL Name—Click a SQL name to review the SQL statement and to compare the recommended execution plan to the original execution plan. ℹ **Tip:** Select (highlight) a SQL statement in the Impacted SQL list to highlight the indexes in the Recommended Indexes list that contributed to the performance improvement of that statement.

Pane / Option / Button	Description
Create/Edit Index generation script Create/Edit Index generation script	Click to create a script to generate the new indexes recommended by Optimize Indexes.

i **Notes:**

- To save the contents of a grid (list) as a file, right-click the grid and select **Save As**.
- An open database connection is not required when opening a completed session, reviewing results, or creating and publishing (printing) a report.

i **Tip:** Use and to navigate back and forth between the pages of a tab.

About Us

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.

Contact Quest

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

Technical Support Resources

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

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

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

SQL Optimizer Community

You can also find help and additional information at the SQL Optimizer community at:

https://blog.toadworld.com/tag/sql-optimizer-for-oracle