

Evaluating Precise for Sybase explain plans

This section includes the following topics:

- [About the SQL tab](#)
- [How the SQL tab is structured](#)
- [How the SQL tab can help you identify performance problems](#)

About the SQL tab

The SQL tab enables you to evaluate SQL statements and manage a database warehouse of your application's SQL statements together with their respective Explain Plans. The process of explaining statements is a prerequisite for tuning. The Explain process is designed to clarify the access path chosen for a statement and translate it into a visual medium.

After a statement is explained, Explain results are stored in the PMDB. This information includes the objects referenced by the statement and the operations performed on these objects. The statements are automatically explained every day. In addition, the top 10 resource-consuming statements during the last 4 hours, are automatically explained every 15 minutes.

Understanding the Execution Plan chosen by the Adaptive Server Query Optimizer is extremely important when tuning your application. You can ensure optimal system performance by ensuring that the best plans are used for your queries.

Precise for Sybase provides you with a special tab just for this purpose - the SQL tab. It provides several views of the same plan, making the process of analyzing long and complicated queries much easier to do.

To analyze the Plan, Precise for Sybase provides you with a full picture of the objects (table, indexes) participating in the Plan. You can change a plan by modifying your query or changing the schema by, for example, the addition of an index (outside of Precise for Sybase).

View the historical information of statements showing performance degradation to identify the source of the problem. A change in Execution Plan may explain the impact on performance.



The information described herein for this tab applies equally to statements and batches.

The table below shows from which tabs and entities it is possible to launch to the SQL tab, in context.

Table 1 Launching in context

Tab	Entities
Current	Session, Statement, Batch
Activity	Statement, Batch
SQL	Statement entered manually by the user through a New action.



If the SQL tab is opened with no statement in context, an error message is displayed requesting you open a statement.

See [About the Current tab](#), [About the Activity tab](#), and [About the Objects tab](#).

How the SQL tab is structured

The SQL tab lets you analyze execution plans and explain results so that you can tune statements and achieve optimal results.

Two sets of tabs control the information displayed in the SQL tab:

- View tabs
- Actions tab

About the View tabs

The SQL tab displays different information regarding the selected statement or batch, in different views. A statement is selected when you drill down to it in another tab (from Current or Activity tabs) or when you open a new or existing statement in the SQL tab. Click a View tab to display additional information regarding the statement or batch. Each view has a different layout.

The following View tabs are available in the SQL tab:

- Plan
- History
- Statements

About viewing the execution plan of a statement

The Plan view tab lets you display the execution plan of a statement and various related information such as statistics, referenced objects and operations performed in the Execution Plan.

The Plan view is divided into two panes. The Execution Plan tree is displayed in the left pane. The Details area is displayed in the right pane. The information displayed in the Details area is controlled by the information tabs (Highlights, Objects, Statistics, Operations, Properties) located above the Details area.

About the Execution plan tree

The Plan view displays the Execution Plan tree of the specified statement. When you analyze the access plan of a selected statement, you can examine the access path that was chosen by the Adaptive Server Query Optimizer. The Explain result is displayed as an Explain tree. Each branch in the Explain tree represents one operation. Each operation is displayed in the following format:

[<execution order>] <operation type> <accessed object>

It is also possible to display the Execution Plan's text, at the bottom of the Execution Plan tree. This allows you to view the text of the statement, the Execution Plan and additional information all at once, such as the objects referenced by the statement.

Actions that can be performed on the tree

It is possible to perform the following actions on the Execution Plan tree:

- The Playback controls, located at the top of the tree enable you to freely move within the Execution Plan of an explained statement.
- By moving the pointer over the Execution Plan steps, you can view a ToolTip that contains statistical information, such as Estimated Cost and Estimated Rows of the specified step.
- Selecting a specific step will highlight its sons and affect the information displayed in the information tabs.
- Clicking the plus sign (+) located at the bottom of the tree, displays the statement's text.

Available operation options

Precise for Sybase enables you to obtain a reasonable understanding of the Execution Plan selected by the Adaptive Server Query Optimizer, by displaying the textual output returned from the `SET SHOWPLAN ON` command in the Plan tree format. In addition to the Execution Plan, the Estimated I/O Costs returned from the `DBCC TRACEON (310, 311)` command are displayed in all the table-based operations.

The following options are available to different operations:

- **Clustered index.** The leaf pages of the index hold the actual data, instead of RIDs (Records ID). A table without a clustered index is called a Heap.
- **Non-clustered index.** The index consists only of the columns that combine the key and a pointer to the records in the table data. If the table has a clustered index, then the pointer is the key of the clustered index. If the table does not have clustered index, then the pointer is an RID.

There are several operations for accessing indexes or tables:

- **Seeking.** The index tree can be used for quickly locating the matching records. Seeking can only be done on indexes.
- **Scanning.** The leaf pages of the index or the data pages of the table are consecutively scanned.



Seeking is the recommended way to use indexes.

The table below describes the available operation options.

Table 2 Operation options

Operator	Description
Sort	The Sort operator presents the sorting of the rows returned from the previously executed operation (appears just below this operator) in a specific order, optionally eliminating duplicate entries. Sorting is required when no index that satisfies the requested ordering exists, or when an Index Scan is more expensive than sorting. It is usually used as the final step to retrieve the fetched data or a prior step for the Join or Stream Aggregate (Group By) operators. In some cases, the Sort operation requires more than the available memory, in which case a temporary table is used. This involves Disk I/O, which has an impact on performance. Sorting a large number of rows makes heavy use of the CPU, which can also affect overall server performance. The amount of memory needed for sorting can be estimated by multiplying the number of rows by the average row size. Therefore, minimizing the row size by selecting only the necessary fields will decrease the memory required for the Sort operation.
Nested Joins	Nested Joins (also called Nested Loops) use the Nested algorithm to join two tables. The Nested algorithm works as follows: For each row in the outer table (the first table, appearing directly below the Nested Loop operator in the Explain tree), Sybase executes the Access Plan for the inner table (the second one). A large number of iterations may indicate a problem in choosing the Access Plan due to out-of-date statistics.
Merge Joins	A Merge Join algorithm is used to join two tables by merging them. Merge Join algorithms read both tables only once; therefore it is required that the input be sorted according to the Join predicates. Merge Joins are available in Precise for Sybase from version 12.0 and above. This option is grayed-out by default. To enable it on session level, execute the <code>set sort_merge</code> on command. See the Sybase Help for more details on how to enable Merge Joins on server level.

About the operators that are available in the plan tree

The table below describes the operators that are available in the Plan tree.

Table 3 Operation options

Operator	Description
Step	This operator indicates the start of the specified step Subtree.
Table Scan	This operator indicates that a table scan was performed on the specified table to receive the required data. The More column displayed in the ToolTip and in the Statistics tab specify the scan direction (forward/backward), the I/O size and the buffer replacement strategy. When there is no useful index, the optimizer performs a table scan to receive the required data.
Index Scan	This operator indicates that an index scan was performed on the specified index to receive the required data. The More column displayed in the ToolTip and in the Statistics tab specify the scan direction (forward/backward), the I/O size and the buffer replacement strategy.
Index Seek	This operator indicates that an index seek was performed on the specified index to receive the required data. The More column displayed in the ToolTip and in the Statistics tab specify the scan direction (forward/backward), the I/O size and the buffer replacement strategy.
Clustered Index Scan	This operator indicates that a clustered index scan was performed on the specified index to receive the required data. The More column displayed in the ToolTip and in the Statistics tab specify the scan direction (forward or backward), the I/O size and the buffer replacement strategy.
Clustered Index Seek	This operator indicates that a clustered index seek was performed on the specified index to receive the required data. The More column displayed in the ToolTip and in the Statistics tab specify the scan direction (forward or backward), the I/O size and the buffer.
Table Lookup	This operator indicates that in addition to an access to the index pages, there is an access to the data pages of the table to receive the required data which is not included in the index itself. The More column displayed in the ToolTip and in the Statistics tab specify the I/O size and the buffer replacement strategy.
Nested Loops	This operator indicates that the join between the two result sets under this operator is of type Nested Loops. The first operator under the Nested Loops operator is the outer table in the join and the second operator under the Nested Loops operator is the inner table in the join.
Merge Join	This operator indicates that the join between the two result sets under this operator is of type Merge Join.
Sort	This operator indicates that a sort operation was performed on the rows returned from the previously executed operation (appears just below this operator) in a specific order, optionally eliminating duplicate entries.
Aggregate	This operator indicates that an Aggregate operation was performed on the rows returned from the previously executed operation (appears just below this operator). This operator appears in steps, which perform an evaluation of grouped or ungrouped aggregate functions such as SUM, AVERAGE and COUNT.
Subquery	This operator indicates that a subquery was executed. The Subquery tree will appear under this operator. The More column displayed in the ToolTip and in the Statistics tab specify the nesting level of the subquery, the predicate type (IN, ANY, EXISTS, or EXPRESSION), and the type of query (correlated/ non-correlated).
Spooling	This operator indicates that a temporary worktable was created to hold intermediate results. Temporary worktables are created in cases when the statement cannot be executed in one step. For example, if the statement contains an Order By clause and the optimizer cannot use an index to order the resulting rows, it creates a worktable to sort the resulting rows before returning them. The More column displayed in the ToolTip and in the Statistics tab specify why the temporary worktable was created (Order By, Distinct, Reformatting, and so on).
Join	This operator indicates that a join operation was performed between a subquery output and another result set.
Parallelism Merge	This operator indicates that a parallelism merge was done to the result sets returned by all the worker processes involved in the parallel execution. The More column displayed in the ToolTip and in the Statistics tab specify the way in which the results of parallel scans were merged.
Log Scan	This operator indicates that a log scan operation was performed.

About the costs that are available in the plan tree

Precise for Sybase uses the DBCC TRACEON (310,311) command to receive the Estimated I/O needed to execute the statement. Since Sybase supplies Estimated I/O only to table-based operations, not all operators in the Plan tree contain I/O Costs. In cases of access both to the index pages and data pages, meaning a Table Lookup is being performed, Sybase provides the Total I/O operations needed while accessing the table. Therefore, the I/O Costs are displayed in the Table Lookup operator and it applies both to the index access and table access. The following operators contain the Total Estimated I/O Costs of their Subtree operators: The entire statement, STEP N and Subquery N.

The table below shows the I/O Cost information operators available in the Plan tree.

Table 4 I/O cost information operators

Counter	Description
---------	-------------

Estimated Cost	Logical Reads * 2 milliseconds (which is the assumption of how long a logical read takes) + Physical Reads * 18 milliseconds (which is the assumption of how long a physical read takes).
Estimated Subtree Cost	Total estimated cost of all the operator's Subtree operators.
Logical Reads	Number of logical reads performed on the specified table. The number of logical reads estimated by the optimizer is based on the access method and the size of cache. In this case, the optimizer is estimating how many pages are likely to be in cache.
Physical Reads	Physical reads are estimated based on the number of logical reads and the cache size. The optimizer is estimating how many pages will not be available in cache and will need to be read from disk.
Scan Count	<p>A "scan" can represent any of these access methods:</p> <ul style="list-style-type: none"> • A table scan. • Access using a clustered index. Each time the query starts at the root page of the index and follows pointers to the data pages, it is counted as a scan. • Access using a non-clustered index. Each time the query starts at the root page of the index and follows pointers to the leaf level of the index (for a covered query) or to the data pages, it is counted. • If queries run in parallel, each worker process access to the table is counted as a scan.
Estimated Rows	Number of rows from the current table that is estimated to satisfy the join.

About viewing a statement's text

The Highlights tab displays the statement's text.

About viewing which objects are referenced by the execution plan



The Objects tab displays three tables (Tables used in the plan, Indexes of table, and Columns of table) that list all referenced objects in the Execution Plan, including their indexes and columns. Statistical details and general details are displayed for each object and its sub-entities.

About tables used in the plan

Displays a list of all the tables referenced in the tree.

The table below describes the information displayed in the tables used in the plan.



Table 5 Tables used in the plan

Column	Description
	Click to launch the Objects tab and focus on the specified table.
	Click to locate and highlight all the operators in the Execution Plan that access the specified table.
Used	Indicates whether the specified table is used in the selected operator in the Execution Plan tree.
Table	Displays the full table name ([Database Name].[Owner].[Table Name]).
Cost	Total cost of the operators that accessed the specified table.
Cost (%)	Total cost in percentage of the operator that accessed the specified table.
Rows	Number of rows in the specified table.
Pages Allocated	The number of pages allocated in the disk for the table records. The value might be incorrect due to out-of-date space usage information. If you suspect the value to be incorrect, use the <code>sp_flushstats</code> command to flush current space usage information from memory or run the <code>update statistics</code> command to recalculate the space usage fields.
Pages Used	The number of pages used in the disk for the table records. The value might be incorrect due to out-of-date space usage information. If you suspect the value to be incorrect, use the <code>sp_flushstats</code> command to flush current space usage information from memory or run the <code>update statistics</code> command to recalculate the space usage fields.

About indexes of table

Displays a list of all the indexes of the selected table displayed in Tables Used in the Plan. The table below describes the information displayed in indexes of table.

Table 6 Indexes of table



Column	Description
	Click to launch the Objects tab and focus on the specified index.
	Click to locate and highlight all the operators in the Execution Plan that access the specified index.
Used	Indicates whether the specified index is used in the selected operator in the Execution Plan tree.
Index	Displays the index name.
Pages Allocated	The number of pages allocated in the disk for the index records. The value may be incorrect due to out-of-date space usage information. If you suspect the value to be incorrect, use the sp_flushstats command to flush current space usage information from memory or run the update statistics command to recalculate the space usage fields.
Pages Used	The number of pages used in the disk for the index records. The value may be incorrect due to out-of-date space usage information. If you suspect the value to be incorrect, use the sp_flushstats command to flush current space usage information from memory or run the update statistics command to recalculate the space usage fields.
Cost	Total cost of the operators that accessed the specified index.
Cost (%)	Total cost in percentage of the operators that accessed the specified index.
Index Depth	Number of levels in the index.
Unique	Indicates whether the index is unique.
Clustered	Indicates whether the index is clustered.

About Columns of Table

Displays a list of all columns in the selected table displayed in Tables Used in Plan. When an index is selected in Indexes of Table, the first column constitutes the index column sorted by the position of the column in the index, and marked with an Ascending or Descending icon.

The table below describes the information displayed in the columns of the table.

Table 7 Columns of table


Column	Description
 	Indicates whether the column is part of the selected index and whether it is sorted in ascending or descending order.
Column	Displays the column name.
Key Number	Displays the key number of the column in the index.
Type	Displays the physical storage type of the column.

Displaying statistical information on operators in the execution plan tree

The Statistics tab displays statistical information of all operators in the Execution Plan tree. When you sort a table, you can locate operators by their estimated cost or by any other column in the table.

The table below shows the statistical information displayed for Operator ID and Type.

Table 8 About viewing statistics on Operator ID and Type

Column	Description
	Click to locate and highlight the operator in the Execution Plan tree that matches the selected operator in the grid.
Operator ID and Type	Displays the operator ID and type.
Logical Reads	Number of logical reads performed on the specified table. The number of logical reads estimated by the optimizer is based on the access method and the size of cache. In this case, the optimizer is estimating how many pages are likely to be in cache.
Physical Reads	Physical reads are estimated based on the number of logical reads and the cache size. The optimizer is estimating how many pages will not be available in cache and will need to be read from disk.


Subtree Cost	Displays the total estimated cost of all the current operation's sons.
Estimated Cost	Displays the estimated cost of the current operation. A high cost value may indicate a problem in the current implementation of the operation. Estimated cost is calculated as followed: Logical Reads * 2 milliseconds (which is the assumption of how long a logical read takes) + Physical Reads * 18 milliseconds (which is the assumption of how long a physical read takes).
Estimated Cost (%)	Displays the estimated cost (in percentage) of the current operator.
Estimated Rows	Number of rows from the current table that is estimated to satisfy the join.
Scan Count	<p>A "scan" can represent any of these access methods:</p> <ul style="list-style-type: none"> • A table scan. • Access using a clustered index. Each time the query starts at the root page of the index and follows pointers to the data pages, it is counted as a scan. • Access using a non-clustered index. Each time the query starts at the root page of the index and follows pointers to the leaf level of the index (for a covered query) or to the data pages, it is counted. • If queries run in parallel, each worker process access to the table is counted as a scan.
More	Displays additional information according to the specified operation. For example, in a table scan operation, the scan type (forward /backward), I/O size and buffer replacement strategy will be displayed.
Parallel	Indicates whether the operator is running in parallel.

About viewing statistical information on operation types

The Operations item under the Menu tab displays two tables that show statistical information broken down into operation types of all operators in the Execution Plan tree.


The table below describes the information displayed for major operation types .

Table 9 Major operations

Column	Description
	Click to locate and highlight the operators in the Execution Plan tree that match the selected major operation type.
Major Operation Type	<p>Displays the major types of the operators.</p> <ul style="list-style-type: none"> • The operators in the Execution Plan tree are separated into the following major types: • Table (any Read access to table) • Index (any Read access to index) • Sort (any sort operation) • Join (any join operation) • Spooling • Other (all other operations)
Estimated Cost	Displays the total estimated cost breakdown according to major operation types.
Operations	Indicates the number of operators of a specific major operation type.
Objects	Indicates the number of referenced tables/indexes of the specified operation type. This column is only relevant for table and index operation types.

The table below describes the information displayed for regular operation types.

Table 10 Operations

Column	Description
	Click to locate and highlight the operators in the Execution Plan tree that match the selected operation type.
Operation Type	Indicates type of operator, such as, Clustered Index Scan, Index Seek, Nested Loop, and so on.

Major Operation Type	Displays the major types of the operators. The operators in the Execution Plan tree are separated into the following major types: <ul style="list-style-type: none"> • Table (any access to table) • Index (any access to index) • Sort (any sort operation) • Join (any join operation) • Spooling • Other (all other operations)
Estimated Cost	Total estimated cost breakdown according to operation types.
Operations	Indicates the number of operators of the specified operation type.
Objects	Indicates the number of referenced tables/indexes of the specified operation type.

About viewing general properties of a selected statement

The Properties item under the More tab displays general details of the selected statement. The table below describes the properties displayed for a selected statement.

Table 11 Selected statement properties

Column	Description
Statement ID	Displays the name of the statement or batch that the SQL tab was launched with. Can also be the name of the statement opened manually.
Database	Indicates the database that was in use when the statement was run.
Parsing User	Indicates the user that was in use when the statement was run.
Cabinet	Indicates the cabinet in which the statement was saved.
Folder	Indicates the folder in which the statement was saved.
Estimate Cost	Indicates the total estimate cost of the statement as calculated for the last access plan.
Most Recent Show Plan	Indicates the last time the statement or batch was explained.
Plan Time	Indicates when the Access Plan was saved. An Access Plan is saved in two cases: <ol style="list-style-type: none"> 1. there is no Access Plan for the statement and this is the first time it is saved; 2. changes were made to the statement's Access Plan.

Observing a history of resource consumption

The History view tab displays Resource Consumption Over Time vs. Access Plan Changes, to determine how the changes affected the performance of the statements or batches.

The History view is divided into the following areas:

- Main area
- Change history area

About the Main area

The Main area (upper area) displays Resource Consumption Over Time vs. Cost and Changes Over Time.

About viewing a history of resource consumption over time

The History view displays Resource Consumption Over Time vs. Cost and Access Plan Changes Over Time in a number of overtime graphs.

- **In Sybase.** Displays the Resource Consumption of the selected statement over a specified time period.
- **Cost.** Displays the Estimated Cost of the selected statement over a specified time period.
- **Changes.** Displays Access Plan Change indications for the statement vs. Average Duration over a specified time period.
- **Executions.** Displays the number of executions of the selected statement over a specified time period.

About viewing execution plan text

The Text view displays the Execution Plan's text.


About the Change History area

The Change History area (lower area) displays all the Access Plan changes made over a specified time period.

About viewing Access Plan Changes

The Access Plan Changes view displays a list of all the Access Plans of the statement as detected by the Explain Statement process.

Table 12 Access Plan Changes table

Column	Description
	Click to switch to Plan view and display the selected Access Plan.
Date	Indicates the date the specified Access Plan was detected by the Explain Statements process.
Estimate Cost	Indicates the estimated cost of the specified Access Plan.
Table Scans	Displays the number of table scans performed in the specified Access Plan.
Index Scans	Displays the number of index scans performed in the specified Access Plan.
Parallel	Displays the number of parallel operations performed in the specified Access Plan.
Nested Loops	Displays the number of nested loops performed in the specified Access Plan.
Merge Joins	Displays the number of merge joins performed in the specified Access Plan.
Sorts	Displays the number of sort operations performed in the specified Access Plan.

Breaking down a batch's access plan into statements

In many cases a statement is part of a long batch (such as a Stored Procedure or improvised batch). Sometimes you want to examine a particular statement, and sometimes you want to view the statement in the context of the entire batch.



The Statements view breaks down the Access Plan of the entire batch into statements and correlates the statements in the Access Plan with the statements that were captured by the Precise for Sybase Collector agent. The statement IDs of these statements that were captured are displayed in Precise. It is possible to view the Access Plan of a different statement by clicking its respective Plan icon.

 Only statements that were captured by the Precise for Sybase Collector agent can be chosen.

When you switch to this view in the context of a batch, all the batch's statements are listed. In this case, all the information displayed is related to the selected batch.

The table below describes the information displayed in the Statement's View table.

Table 13 Statement's View table

Column	Description
	Click to switch to Plan view, in-context with the selected statement. <div> This icon is only enabled at batch level and in the statement that was captured by the Collector agent.</div>
ID	Indicates the ID of the statement in the batch.
Collector Statement	Displays the ID of the statement and batch as assigned by the Precise for Sybase Collector agent.
Physical Operator	Describes how a query or update was executed. The physical operator describes the physical implementation used to process a statement, for example: Batch level, Select, Update, Cond, and so on.
Operations	Displays the number of operations in the specified statement.
Table Scans	Displays the number of Table Scans performed in the specified statement.
Index Scans	Displays the number of Index Scans performed in the specified statement.
Joins	Displays the number of join operations performed in the specified statement.

Sorts	Displays the number of sort operations performed in the specified statement.
Estimated Cost	Displays the estimated cost for the specified statement.

About the Actions tab

The Actions tab, on the Precise bar, lets you perform additional actions. You can perform the following actions:

- Create a new statement
- Open an existing statement
- Edit an existing statement
- Re-Explain an existing statement
- Display the Sybase plan using Sybase SQL Advantage and Interactive SQL

Creating a new statement

You can create a new Statement and save it in the PMDB in a logical cabinet and folder hierarchy. You can also rewrite statements and view their Access Plans.

To create a new statement

1. On the Precise bar, click **Actions>New**.
2. In the New Statement dialog box, choose the relevant properties for the new statement from the drop-down lists and enter a statement name, as follows:
 - **Instance.** Indicates the name of the instance that the statement belongs to.
 - **Cabinet.** Indicates the name of the cabinet that the statement is saved in (creates a cabinet if it does not already exist).
 - **Folder.** Indicates the name of the folder that the statement is saved in (creates a folder if it does not already exist).
 - **Name.** Indicates the name of the statement (generated randomly).
 - **Database.** Indicates the name of the database where the statement is running.
 - **User.** Indicates the statement's parsing user.
3. Enter the SQL text for the statement in the text frame.
4. Click **OK**.

The new statement is saved in the PMDB in a logical cabinet and folder hierarchy.

Opening an existing statement

You can view a statement that was saved in the PMDB. To open an existing statement

1. On the Precise bar, click **Actions>Open**.
2. In the Open Statement dialog box, choose the relevant properties for the statement you want to view from the Instance, Cabinet, Folder and Name drop-down lists, as follows:
 - **Instance.** Indicates the name of the instance that the statement belongs to.
 - **Cabinet.** Indicates the name of the cabinet that the statement is saved in (creates a cabinet if it does not already exist).
 - **Folder.** Indicates the name of the folder that the statement is saved in (creates a folder if it does not already exist).
 - **Name.** Indicates the name of the statement (generated randomly).
 - **Database.** Indicates the name of the database where the statement is running.
 - **User.** Indicates the statement's parsing user.

The statement's text is displayed in the text frame.
3. Click **OK**.

Editing an existing statement

You can edit a statement that was saved in the PMDB. The statement that was collected by the Collector agent is saved as a New Statement in a logical cabinet and folder hierarchy.

To edit an existing statement

1. On the Precise bar, click **Actions>Edit**.
2. In the Edit Statement dialog box, define the statement you want to edit by choosing a relevant property from the drop-down lists, as follows:
 - **Instance.** Indicates the name of the instance that the statement belongs to.
 - **Cabinet.** Indicates the name of the cabinet that the statement is saved in (creates a cabinet if it does not already exist).
 - **Folder.** Indicates the name of the folder that the statement is saved in (creates a folder if it does not already exist).
 - **Database.** Indicates the name of the database where the statement is running.
 - **User.** Indicates the statement's parsing user.
3. Enter a New Statement name in the **Name** field.
4. Edit the SQL text for the statement in the text frame.
5. Click **OK**.

The edited statement is saved in a logical cabinet and folder hierarchy.

Re-Explaining an existing statement

You can Re-Explain an existing statement. If the new access plan is different from the most recent access plan, it is saved in the PMDB and displayed.

To re-explain an existing statement, click the Re-Explain tab

Display the Sybase plan using Sybase SQL Advantage and Interactive SQL

You can display the Sybase plan using Sybase SQL Advantage and Interactive SQL. To display the Sybase plan using Sybase SQL Advantage and Interactive SQL, click the Sybase Plan tab.

The Sybase SQL Advantage window is opened with the statement in context and commands to display its access path are executed.

How the SQL tab can help you identify performance problems

After identifying a problematic statement that is slowing down the response time of a specific application, the first step in tuning the statement is to understand the access path that Sybase chose for the statement. The Explain procedure is designed to clarify the access path chosen for a statement and to translate it into a visual medium. Therefore you can easily see whether the optimizer chose the proper Execution Plan. For example, you can see whether the optimizer performed an Index Seek as expected. In addition, you can see if the Access Plan of the statement was changed and how the changes affected the performance of the statement.

You can identify a performance problem by doing one or more of the following:

- [Identifying problematic operators in the latest access plan](#)
- [Locating referenced tables that pose potential problems](#)
- [Examining how access path changes affect statement performance](#)
- [Locating the most resource-consuming statements in a batch](#)

Identifying problematic operators in the latest access plan

The first step in tuning a statement is to identify problematic operators. Problematic operators are operators whose estimated cost is high.

To identify problematic operators

1. Launch to the SQL tab with a statement in-context or open the statement you want to analyze in the SQL tab.
2. Open the Statistics tab in Plan view
3. Sort the grid by the Estimated Cost value (desc.)



The lower Estimated Cost the better.

Following are examples of how to improve problematic operators:

- If the operator is a Table Scan, try creating an index.
- If the operator is of a Join type (such as Nested Loop or Merge), examine a different join method.
- If the operator is a Sort operation, check if the sort is required by the application or can be removed. For example, remove `distinct` or change the `union` clause to `union all`.

See [About the Execution plan tree](#).

Locating referenced tables that pose potential problems

When it is not possible to access the statement, such as in third company products, the only way to improve the statement is by improving access to the data. This can be achieved by performing changes in the referenced objects by examining the operators that access the tables and identifying potential problematic operators.

To locate reference tables that pose potential problems

1. Launch to the SQL tab with a statement in-context or open the statement you want to analyze in the SQL tab.
2. On the Plan tab, click **Objects**.
3. Click the Locator icon (to locate and highlight all the operators in the execution plan that access the specified table).
4. Examine the cost of these operators (to locate the problematic operators).
5. Identify the objects that are referenced by the problematic operators and try to tune those objects.

See [About viewing which objects are referenced by the execution plan](#).

Examining how access path changes affect statement performance

Precise for Sybase allows you to track the Access Plan changes and compare them with the Resource Consumption of your statement to understand how the changes affected the statement's performance.

To examine how access path changes affect statement performance

1. Launch to the SQL tab with a statement in-context or open the statement you want to analyze in the SQL tab.
2. On the History tab, on the View controls, click **History**.

3. Analyze the statement's resource consumption over time vs. execution cost, access path changes, and number of executions over time. This can help you to observe whether the change in the statement performance occurred due to the change in the access path or not.
4. In the Changes History table, you can view additional details on the change made. You can select the launch icon for a particular access path and launch to the Plan tab in-context to continue your analysis of the change.

Locating the most resource-consuming statements in a batch

The main difference between examining a statement's Access Plan and a batch's Access Plan is in the number of operators in the Access Plan. Understanding an Access Plan with hundreds of operators is a difficult task. The Statements view can assist you with this task since it shows the Access Plan statements and their Estimated Cost. The ID of the statements sampled by the Precise for Sybase Collector agent are also displayed.

To locate the most resource-consuming statements in a batch

1. Launch to the SQL tab with a statement in-context or open the statement you want to analyze in the SQL tab.
2. On the Statement tab, observe the access plan statements and their estimated cost.
3. Determine which of the statements has the highest estimated cost and focus your analysis on that statement.