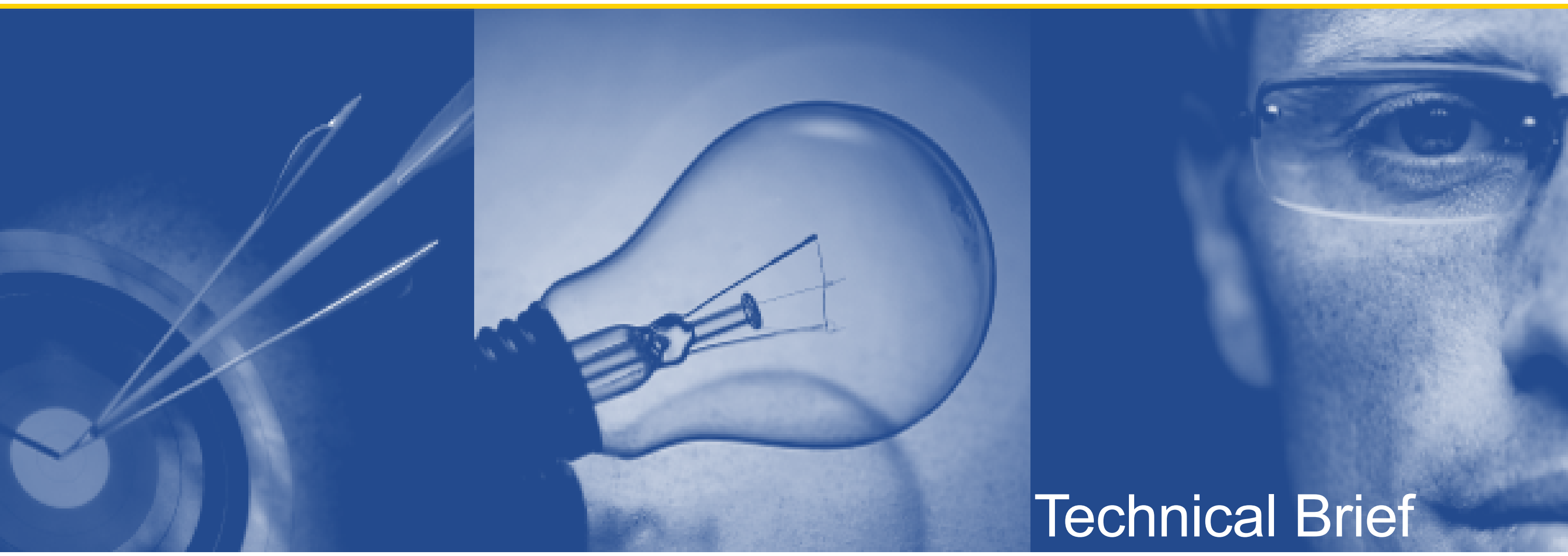
Identifying Problematic SQL in Sybase ASE

No Matter What Your Role, Quest Has the Right Tool to Help

Written by
Darren Mallette
Senior Technical Consultant
Quest Software, Inc.



WARRANTY

The information contained in this document is subject to change without notice. Quest Software makes no warranty of any kind with respect to this information. QUEST SOFTWARE SPECIFICALLY DISCLAIMS THE IMPLIED WARRANTY OF THE MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. Quest Software shall not be liable for any direct, indirect, incidental, consequential, or other damage alleged in connection with the furnishing or use of this information.

TRADEMARKS

World Headquarters:
5 Polaris Way
Aliso Viejo, CA 92656
e-mail: info@quest.com

Please refer to our Web site (www.quest.com) for regional and international office information.

Updated—April 2009

CONTENTS

OVERVIEW

This technical brief describes the different Quest products that can be used to identify problematic SQL statements from Sybase Adaptive Server Enterprise (ASE) versions 12.5.0.3 and higher.

The document is directed toward developers, QA analysts, database administrators (DBAs), performance engineers and database managers.

BACKGROUND: MDA TABLES

Sybase introduced Monitoring Data Access (MDA) tables in ASE 12.5.0.3. MDA tables are actually proxy tables that have no persistent data storage. They are implemented as in-memory “ring buffers” that hold as much data as you tell them to via sp_configure parameters. How to install and configure MDA tables is beyond the scope of this paper, but will be mentioned several times.

For excellent resources on MDA tables, please visit Rob Verschoor’s site at http://www.sypron.nl/mda or consult your ASE documentation.

IDENTIFYING PROBLEMATIC SQL

The tool you need for identifying problematic SQL depends on your role in the organization. A production database administrator (DBA), a developer, a performance engineer, and a quality assurance analyst all require different features in a tool.

Introduction of MDA tables has simplified the task of capturing SQL statements from the ASE server. However, several challenges remain. First, MDA tables are not persistent–that is, they do not store any permanent data. If the ASE server is restarted, then previous performance data is lost. Second, the tables themselves have a fixed size, and they follow a first-in first-out method of purging data. Proper sizing needs to be done so that you don't miss data. Finally, although the MDA tables provide you with the SQL that has executed, you must then narrow down to the problematic SQL statements.

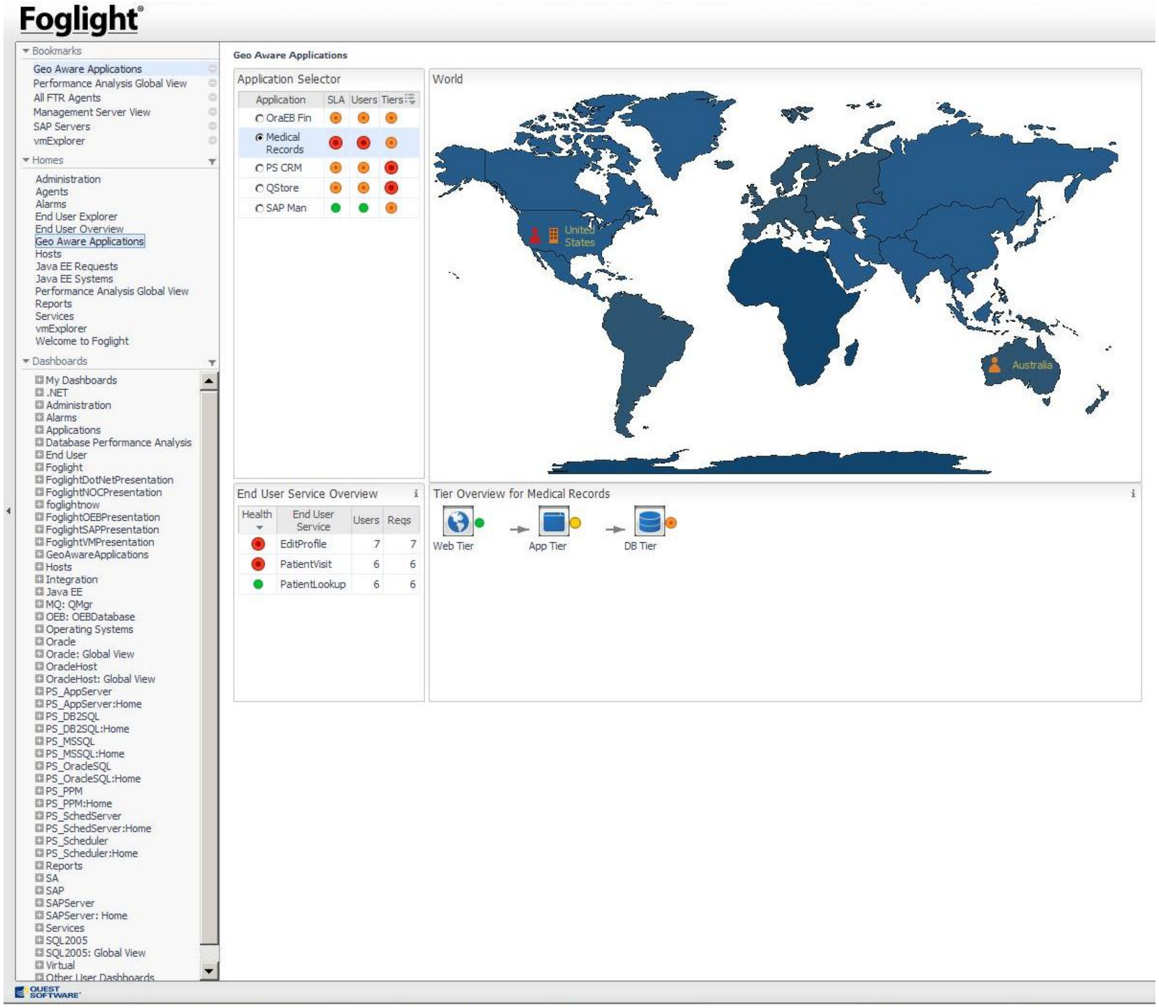
In this brief, several use cases are presented that represent scenarios that are commonly found in organizations. Organizational environments and responsibilities may vary so it's best to explore the capabilities of each product listed, and use those that best suit your needs.

Monitoring the Sybase Server—the DBA

The DBA is primarily concerned with the performance of Sybase ASE and, perhaps, to a lesser extent, the operating system (Unix, Linux, or Windows), though that is typically the realm of the system administrator. Foglight provides unattended, 24x7 monitoring for the entire application stack—from the end user to the application server to the database and the operating system.

Foglight Overview

The high-level Foglight dashboard below shows the applications and their SLA, user impact, and infrastructure status. Drilldowns give further insight into transactions, locations, and infrastructure tiers.



Application Selector

Application	SLA	Users	Tiers
○ OraEB Fin	●	●	●
● Medical Records	●	●	●
○ PS CRM	●	●	●
○ QStore	●	●	●
○ SAP Man	●	●	●

End User Service Overview

Health	End User Service	Users	Reqs
●	EditProfile	7	7
●	PatientVisit	6	6
●	PatientLookup	6	6

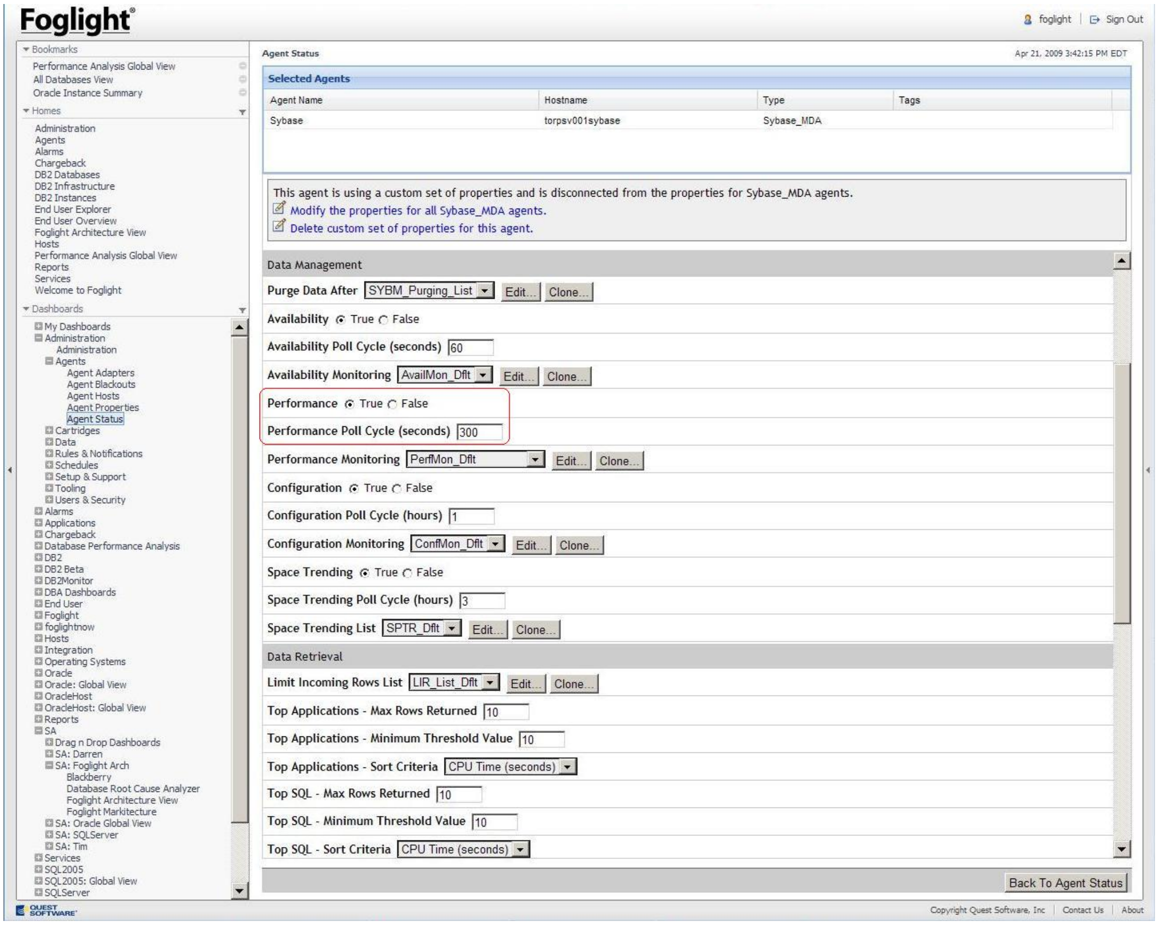
Figure 1. Foglight dashboard

Configuring Sybase Data Collection

A few of the DBA's tasks are to see how ASE is using resources (such as CPU, memory, and disk), determine whether any users are blocked or deadlocked, and identify any resource "hogs." The Sybase_MDA cartridge available in Foglight allows the periodic capture and storage of performance data and the ability to define criteria to identify resource hogs. These can be individual SQL statements, users, or programs.

Specify what Performance Data to Collect

Once the Sybase cartridge has been deployed to your target ASE server, a simple configuration change is all that is needed to start capturing these statements. First we need to modify the agent properties for the Sybase cartridge. The Data Management section allows us to specify what performance data to collect, and how frequently. The Performance option needs to be checked in order to collect the top resource consuming SQL statements. Note also that the default sample interval is 300 seconds, or five minutes.



Agent Status

Apr 21, 2009 3:42:15 PM EDT

Selected Agents

Agent Name	Hostname	Type	Tags
Sybase	torpsv001sybase	Sybase_MDA	

This agent is using a custom set of properties and is disconnected from the properties for Sybase_MDA agents.
Modify the properties for all Sybase_MDA agents.
Delete custom set of properties for this agent.

Data Management

Purge Data After SYBM_Purging_List Edit... Clone...
Availability ● True ○ False
Availability Poll Cycle (seconds) 60
Availability Monitoring AvailMon_Dflt Edit... Clone...
Performance ● True ○ False
Performance Poll Cycle (seconds) 300
Performance Monitoring PerfMon_Dflt Edit... Clone...
Configuration ● True ○ False
Configuration Poll Cycle (hours) 1
Configuration Monitoring ConfMon_Dflt Edit... Clone...
Space Trending ● True ○ False
Space Trending Poll Cycle (hours) 3
Space Trending List SPTR_Dflt Edit... Clone...

Data Retrieval

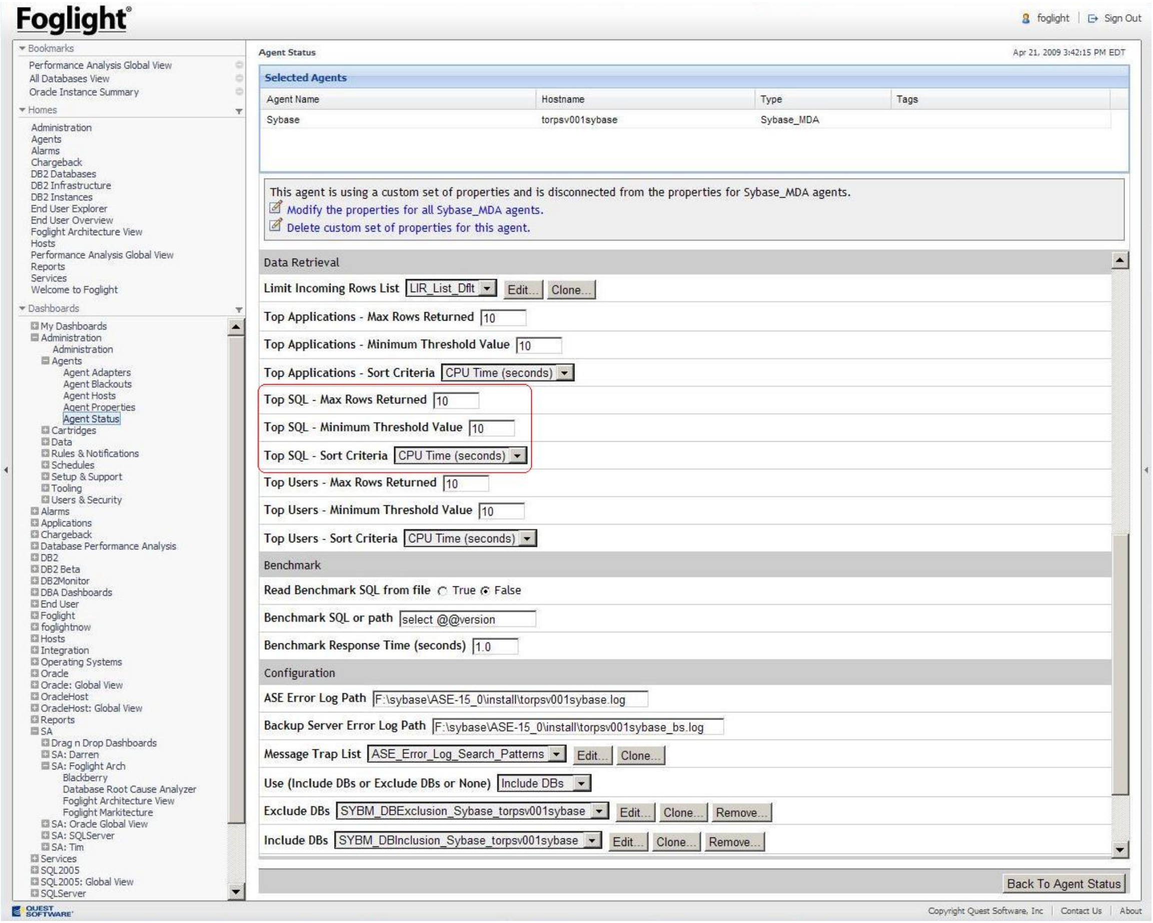
Limit Incoming Rows List LIR_List_Dflt Edit... Clone...
Top Applications - Max Rows Returned 10
Top Applications - Minimum Threshold Value 10
Top Applications - Sort Criteria CPU Time (seconds)
Top SQL - Max Rows Returned 10
Top SQL - Minimum Threshold Value 10
Top SQL - Sort Criteria CPU Time (seconds)

Back To Agent Status

Figure 2. Configuring Sybase data collection

Specify Collection Criteria for SQL Statements

Next we need to specify collection criteria for SQL statements. We do this from the Data Retrieval section of the properties page. For top applications, SQL, and users, we specify the number of rows that Foglight will retrieve at each interval, in addition to the minimum resource consumption limit that has to occur. In the screenshot below, we have specified that the top 10 SQL statements be retrieved at each sample interval, and these must be statements that have at least 10 seconds of CPU time per execution.



Agent Status

Apr 21, 2009 3:42:15 PM EDT

Selected Agents

Agent Name	Hostname	Type	Tags
Sybase	torpsv001sybase	Sybase_MDA	

This agent is using a custom set of properties and is disconnected from the properties for Sybase_MDA agents.

Modify the properties for all Sybase_MDA agents.

Delete custom set of properties for this agent.

Data Retrieval

Limit Incoming Rows List: LIR_List_Dflt [Edit...] [Clone...]

Top Applications - Max Rows Returned: 10

Top Applications - Minimum Threshold Value: 10

Top Applications - Sort Criteria: CPU Time (seconds)

Top SQL - Max Rows Returned: 10

Top SQL - Minimum Threshold Value: 10

Top SQL - Sort Criteria: CPU Time (seconds)

Top Users - Max Rows Returned: 10

Top Users - Minimum Threshold Value: 10

Top Users - Sort Criteria: CPU Time (seconds)

Benchmark

Read Benchmark SQL from file ○ True ● False

Benchmark SQL or path: select @@version

Benchmark Response Time (seconds): 1.0

Configuration

ASE Error Log Path: F:\sybase\ASE-15_0\install\torpsv001sybase.log

Backup Server Error Log Path: F:\sybase\ASE-15_0\install\torpsv001sybase_bs.log

Message Trap List: ASE_Error_Log_Search_Patterns [Edit...] [Clone...]

Use (Include DBs or Exclude DBs or None): Include DBs

Exclude DBs: SYBM_DBExclusion_Sybase_torpsv001sybase [Edit...] [Clone...] [Remove...]

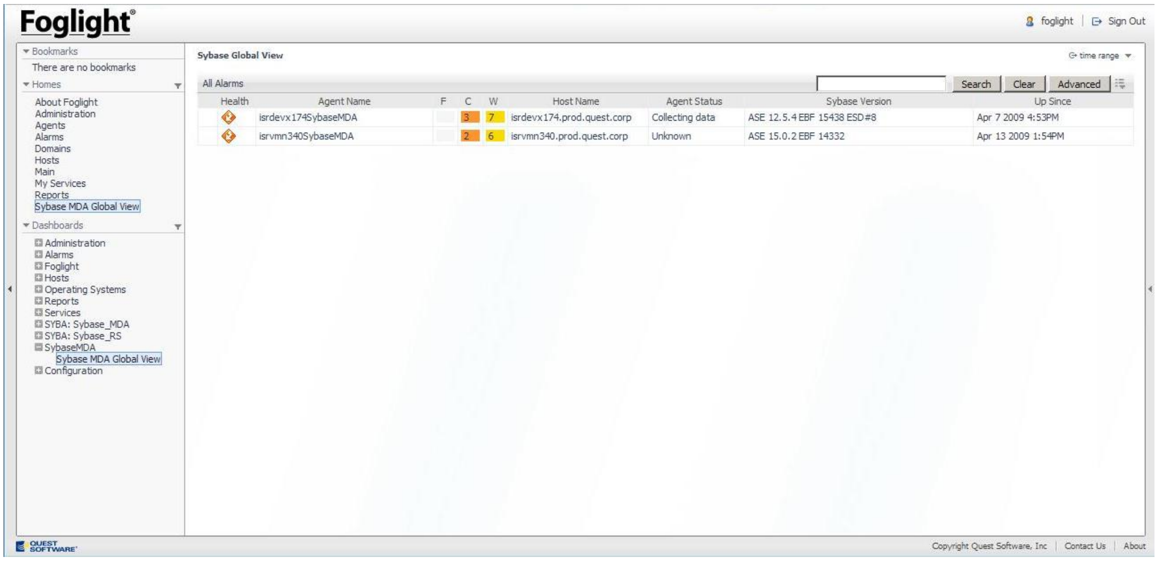
Include DBs: SYBM_DBInclusion_Sybase_torpsv001sybase [Edit...] [Clone...] [Remove...]

[Back To Agent Status]

Figure 3. Specifying collection criteria for SQL statements

Reviewing the Top SQL Statements

Once some time has passed to allow for data collection, the DBA can review top SQL using the Sybase dashboards. The DBA can also drill down to a specific ASE server from the Sybase Global View dashboard.



Sybase Global View

All Alarms

Health	Agent Name	F	C	W	Host Name	Agent Status	Sybase Version	Up Since
◈	isrdevx174SybaseMDA		3	7	isrdevx174.prod.quest.corp	Collecting data	ASE 12.5.4 EBF 15438 ESD#8	Apr 7 2009 4:53PM
◈	isrvmn340SybaseMDA		2	6	isrvmn340.prod.quest.corp	Unknown	ASE 15.0.2 EBF 14332	Apr 13 2009 1:54PM

Figure 4. Viewing the top SQL statements

Drill Down into a Particular ASE Server

Drilling down into a specific ASE Server gives the DBA an end-to-end view of its processes and data flows.

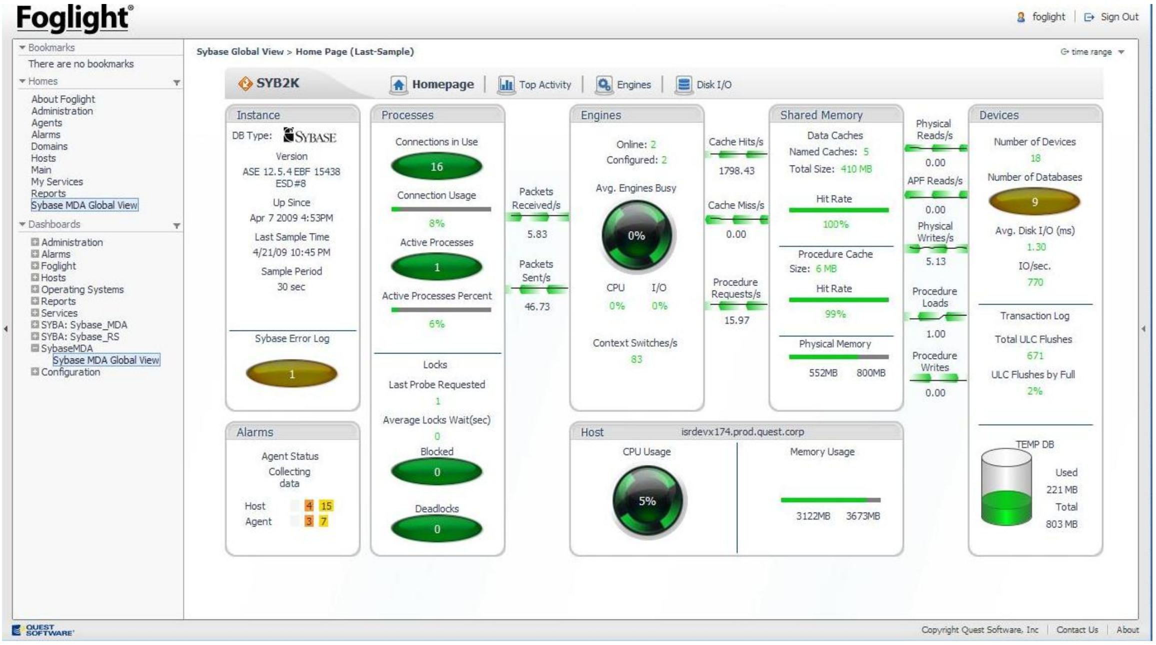
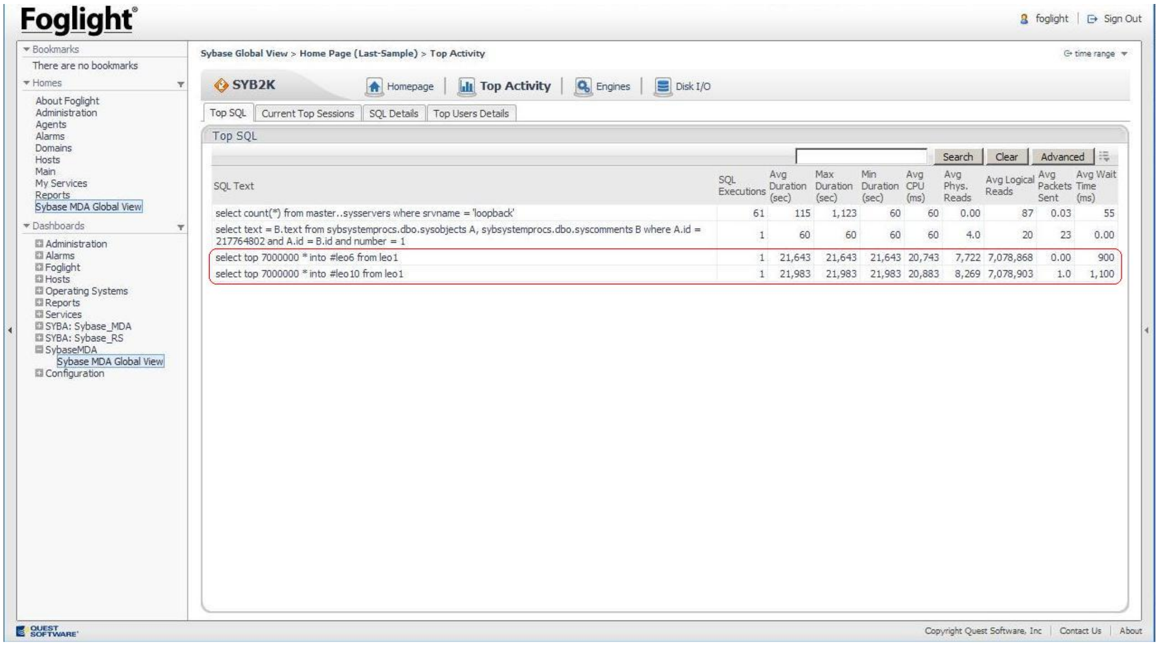


Figure 5. Drilling down into a specific ASE Server

Viewing the Top SQL for a Given Time Period

The Top Activity tab then allows the DBA to review the top SQL that was collected for a time range. The page for TopSQL defaults to the past 4 hours, but can easily be changed. Since the data is retrieved from the MDA tables and stored locally to the Foglight server, it is preserved for later review. On the Top SQL page below, a SQL was executed with high logical reads and CPU time. The DBA can now take this data back to the developers so they can investigate the code and tune the SQL to prevent the problem in the future.



SQL Text	SQL Executions	Avg Duration (sec)	Max Duration (sec)	Min Duration (sec)	Avg CPU (ms)	Avg Phys. Reads	Avg Logical Reads	Avg Packets Sent	Avg Wait Time (ms)
select count(*) from master..sysservers where srvname = 'loopback'	61	115	1,123	60	60	0.00	87	0.03	55
select text = B.text from sybsystemprocs.dbo.sysobjects A, sybsystemprocs.dbo.syscomments B where A.id = 217764802 and A.id = B.id and number = 1	1	60	60	60	60	4.0	20	23	0.00
select top 7000000 * into #leo6 from leo1	1	21,643	21,643	21,643	20,743	7,722	7,078,868	0.00	900
select top 7000000 * into #leo10 from leo1	1	21,983	21,983	21,983	20,883	8,269	7,078,903	1.0	1,100

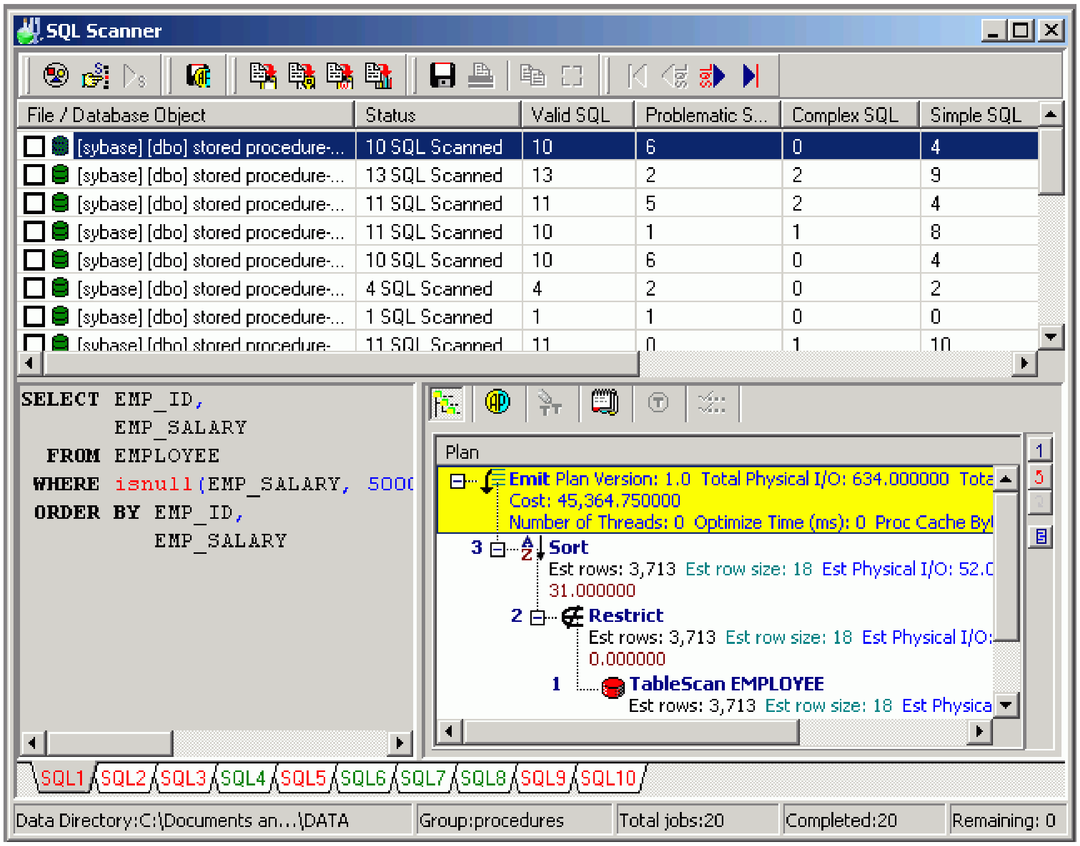
Figure 6. Viewing the top SQL statements for the ASE Server

Investigating SQL within the Application Code—the Developer

Often, developers write SQL code to be functionally correct but give little regard to performance, saying, “Performance is the DBA’s job.” Similarly, code maintenance and SQL tuning is often regarded as drudge work—who wants to tune legacy SQL code when they could be doing cool Java or .Net stuff? Tuning code can be quite difficult, because the SQL could be embedded in stored procedures or views in the ASE server, in text files on the LAN, or compiled into executable code.

However, a SQL statement is correct only if it returns the proper result and minimizes impact on other system resources. Performance is not only the DBA’s job; it is also the developer’s job. The SQL Scanner module of SQL Optimizer for Sybase provides an easy way to extract and analyze embedded SQL, so developers can determine which statements really need tuning without having to run the application.

To start, simply add objects containing SQL to the SQL Scanner. The figure below shows the stored procedures from an ASE server; text and binary files from a Windows file system could also be added. Then with one click, the SQL Scanner will extract the SQL and analyze the query plan to pinpoint the problematic statements.



File / Database Object	Status	Valid SQL	Problematic S...	Complex SQL	Simple SQL
□ [sybase] [dbo] stored procedure-...	10 SQL Scanned	10	6	0	4
□ [sybase] [dbo] stored procedure-...	13 SQL Scanned	13	2	2	9
□ [sybase] [dbo] stored procedure-...	11 SQL Scanned	11	5	2	4
□ [sybase] [dbo] stored procedure-...	11 SQL Scanned	10	1	1	8
□ [sybase] [dbo] stored procedure-...	10 SQL Scanned	10	6	0	4
□ [sybase] [dbo] stored procedure-...	4 SQL Scanned	4	2	0	2
□ [sybase] [dbo] stored procedure-...	1 SQL Scanned	1	1	0	0
□ [sybase] [dbo] stored procedure-	11 SQL Scanned	11	0	1	10

```
SELECT EMP_ID,
       EMP_SALARY
  FROM EMPLOYEE
 WHERE isnull(EMP_SALARY, 500(
 ORDER BY EMP_ID,
          EMP_SALARY
```

```
Emit Plan Version: 1.0  Total Physical I/O: 634.000000  Tota
Cost: 45,364.750000
Number of Threads: 0  Optimize Time (ms): 0  Proc Cache Byt
  3 Sort
    Est rows: 3,713  Est row size: 18  Est Physical I/O: 52.0
    31.000000
    2 Restrict
      Est rows: 3,713  Est row size: 18  Est Physical I/O:
      0.000000
      1 TableScan EMPLOYEE
        Est rows: 3,713  Est row size: 18  Est Physica
```

Data Directory:C:\Documents an...\DATA | Group:procedures | Total jobs:20 | Completed:20 | Remaining: 0

Figure 7. Analysis of the query plan for the stored procedures from an ASE server

When the analysis is complete, the developer can review the extracted SELECT statement from the stored procedure, the abstract plan, the query plan (showplan), the XML query plan (in ASE 15), the trace on data, and the analysis of the plan. Any temp tables in the stored procedure are also extracted to facilitate proper analysis and tuning.

If the developer decides to tune this statement, just one click will bring it into the Syntactical SQL Optimizer for a comprehensive re-write and benchmark test to optimize performance.

Understanding What the Application is Doing—the QA Analyst

Before an application goes into production, the quality assurance team is responsible for making sure it works as expected. The team performs functional, unit and load tests to evaluate the application. Unfortunately, the database is often seen as a black box, and performance testing is frequently rudimentary, limited to system commands such as top, perfmon, or sp_who (which shows the ASE users).

Getting an Overview of ASE Server Health

Spotlight on Sybase enables QA analysts to easily visualize the health of the ASE server at a glance, in real time, during the load test. If any problems arise, an icon changes color to draw attention to the problem. With the aid of context-sensitive help, the analyst can drill down to the root cause of the problem.

For example, the Spotlight screen below shows three problem users and one blocked process. Problem users are identified based on thresholds you specify; in this case it is due to excessive time blocked and high CPU usage.

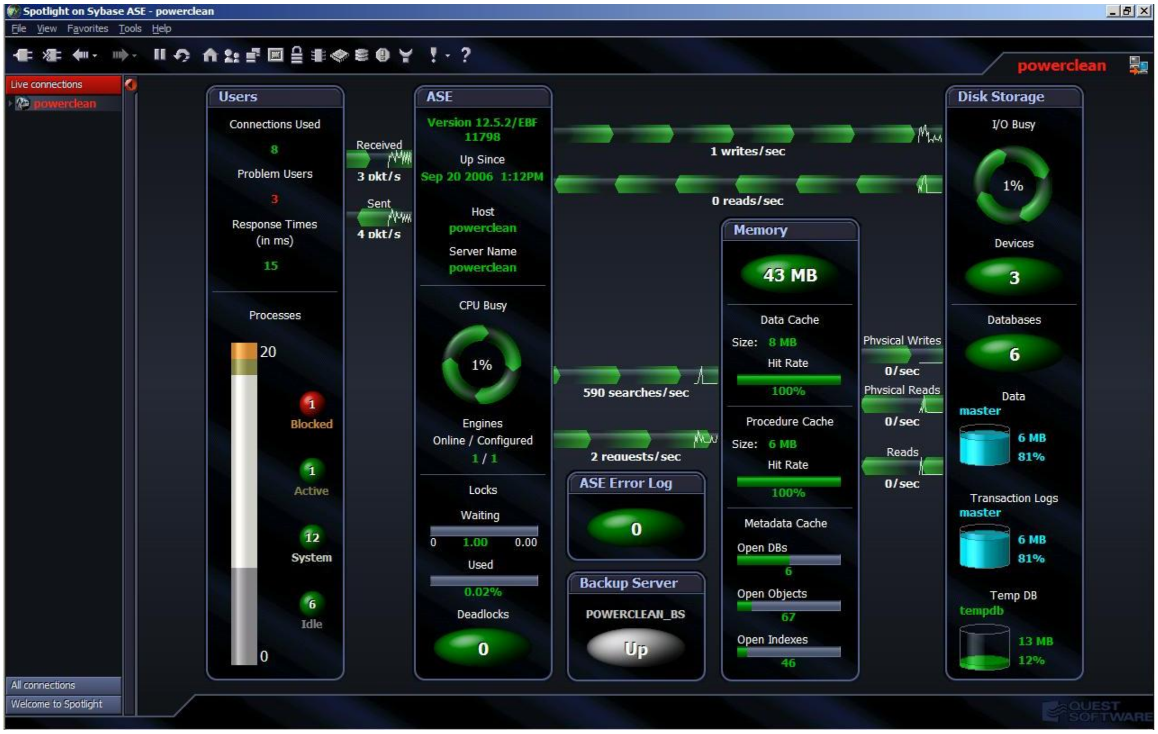
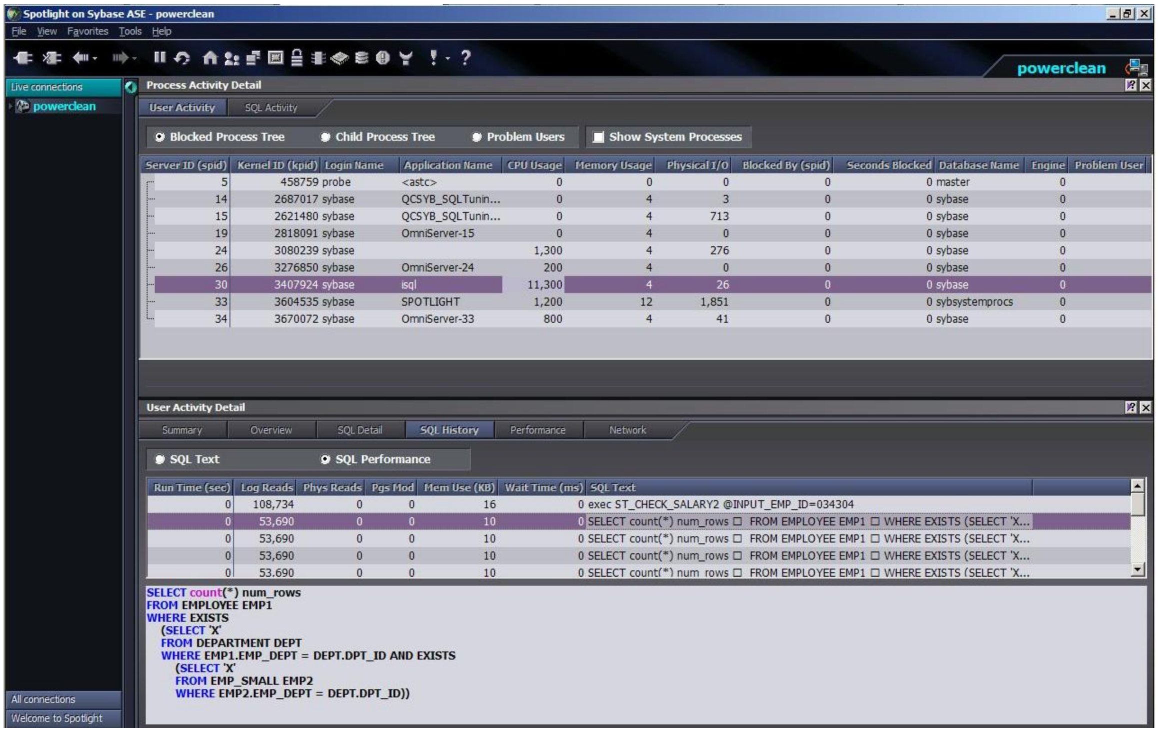


Figure 8. Identifying problem users and blocked processes with Spotlight

Drilling Down to the SQL Generated by a Specific Session

The analyst can then drill down to current process activity and sort the list of sessions to identify the session consuming the most resources. The session information provides many metrics, including a history of the SQL statements that executed. The analyst can click a statement to see the full text, which can then be copied and sent back to a triage team for resolution.



Server ID (spid)	Kernel ID (kpid)	Login Name	Application Name	CPU Usage	Memory Usage	Physical I/O	Blocked By (spid)	Seconds Blocked	Database Name	Engine	Problem User
5	458759	probe	<astc>	0	0	0	0	0	master	0	
14	2687017	sybase	QCSYB_SQLTunin...	0	4	3	0	0	sybase	0	
15	2621480	sybase	QCSYB_SQLTunin...	0	4	713	0	0	sybase	0	
19	2818091	sybase	OmniServer-15	0	4	0	0	0	sybase	0	
24	3080239	sybase		1,300	4	276	0	0	sybase	0	
26	3276850	sybase	OmniServer-24	200	4	0	0	0	sybase	0	
30	3407924	sybase	isql	11,300	4	26	0	0	sybase	0	
33	3604535	sybase	SPOTLIGHT	1,200	12	1,851	0	0	sybsystemprocs	0	
34	3670072	sybase	OmniServer-33	800	4	41	0	0	sybase	0	

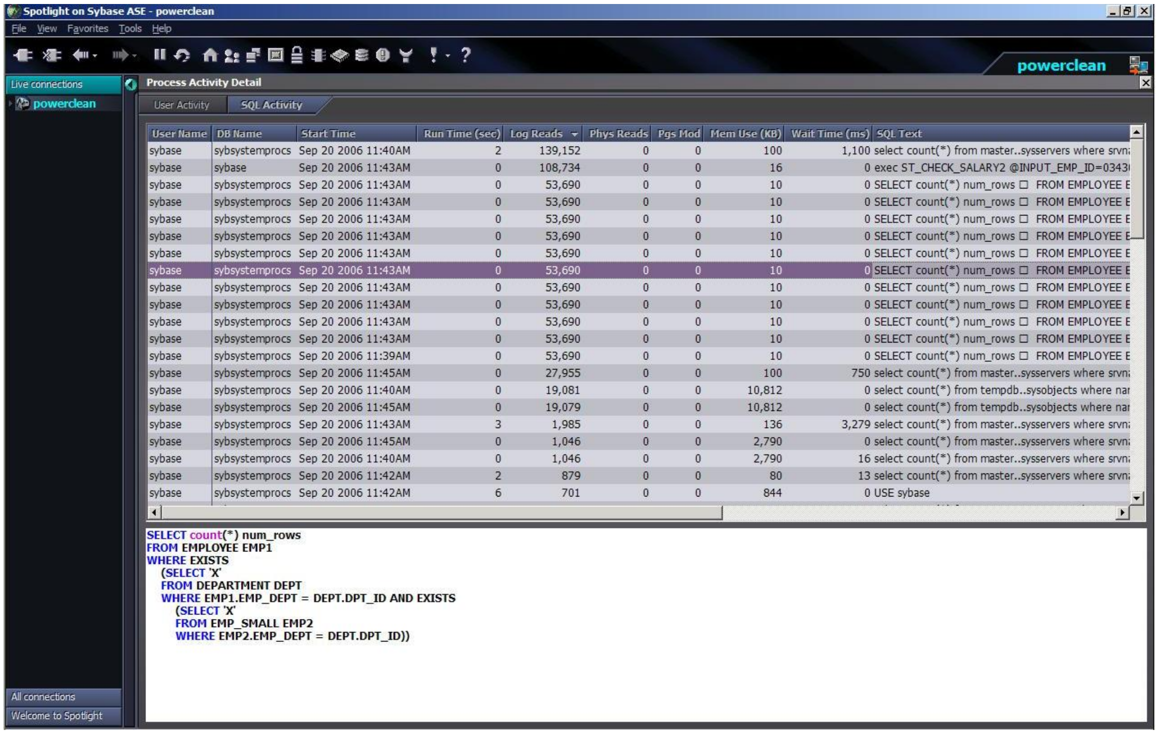
Run Time (sec)	Log Reads	Phys Reads	Pgs Mod	Mem Use (KB)	Wait Time (ms)	SQL Text
0	108,734	0	0	16	0	exec ST_CHECK_SALARY2 @INPUT_EMP_ID=034304
0	53,690	0	0	10	0	SELECT count(*) num_rows □ FROM EMPLOYEE EMP1 □ WHERE EXISTS (SELECT 'X...
0	53,690	0	0	10	0	SELECT count(*) num_rows □ FROM EMPLOYEE EMP1 □ WHERE EXISTS (SELECT 'X...
0	53,690	0	0	10	0	SELECT count(*) num_rows □ FROM EMPLOYEE EMP1 □ WHERE EXISTS (SELECT 'X...
0	53,690	0	0	10	0	SELECT count(*) num_rows □ FROM EMPLOYEE EMP1 □ WHERE EXISTS (SELECT 'X...

```
SELECT count(*) num_rows
FROM EMPLOYEE EMP1
WHERE EXISTS
   (SELECT 'X'
   FROM DEPARTMENT DEPT
   WHERE EMP1.EMP_DEPT = DEPT.DPT_ID AND EXISTS
      (SELECT 'X'
      FROM EMP_SMALL EMP2
      WHERE EMP2.EMP_DEPT = DEPT.DPT_ID))
```

Figure 9. Drilling down to problematic SQL in a particular session with Spotlight

Finding the Most Resource-Intensive SQL Server-wide

In addition to evaluating the SQL generated by a specific session, the QA analyst can view a current "trace" of SQL statements server-wide in the SQL Activity tab. This list can also be sorted to identify the most resource-intensive SQL.



User Name	DB Name	Start Time	Run Time (sec)	Log Reads	Phys Reads	Pgs Mod	Mem Use (KB)	Wait Time (ms)	SQL Text
sybase	sybsystemprocs	Sep 20 2006 11:40AM	2	139,152	0	0	100	1,100	select count(*) from master..sysservers where srvn
sybase	sybase	Sep 20 2006 11:43AM	0	108,734	0	0	16	0	exec ST_CHECK_SALARY2 @INPUT_EMP_ID=0343
sybase	sybsystemprocs	Sep 20 2006 11:43AM	0	53,690	0	0	10	0	SELECT count(*) num_rows □ FROM EMPLOYEE E
sybase	sybsystemprocs	Sep 20 2006 11:43AM	0	53,690	0	0	10	0	SELECT count(*) num_rows □ FROM EMPLOYEE E
sybase	sybsystemprocs	Sep 20 2006 11:43AM	0	53,690	0	0	10	0	SELECT count(*) num_rows □ FROM EMPLOYEE E
sybase	sybsystemprocs	Sep 20 2006 11:43AM	0	53,690	0	0	10	0	SELECT count(*) num_rows □ FROM EMPLOYEE E
sybase	sybsystemprocs	Sep 20 2006 11:43AM	0	53,690	0	0	10	0	SELECT count(*) num_rows □ FROM EMPLOYEE E
sybase	sybsystemprocs	Sep 20 2006 11:43AM	0	53,690	0	0	10	0	SELECT count(*) num_rows □ FROM EMPLOYEE E
sybase	sybsystemprocs	Sep 20 2006 11:43AM	0	53,690	0	0	10	0	SELECT count(*) num_rows □ FROM EMPLOYEE E
sybase	sybsystemprocs	Sep 20 2006 11:43AM	0	53,690	0	0	10	0	SELECT count(*) num_rows □ FROM EMPLOYEE E
sybase	sybsystemprocs	Sep 20 2006 11:43AM	0	53,690	0	0	10	0	SELECT count(*) num_rows □ FROM EMPLOYEE E
sybase	sybsystemprocs	Sep 20 2006 11:43AM	0	53,690	0	0	10	0	SELECT count(*) num_rows □ FROM EMPLOYEE E
sybase	sybsystemprocs	Sep 20 2006 11:39AM	0	53,690	0	0	10	0	SELECT count(*) num_rows □ FROM EMPLOYEE E
sybase	sybsystemprocs	Sep 20 2006 11:45AM	0	27,955	0	0	100	750	select count(*) from master..sysservers where srvn
sybase	sybsystemprocs	Sep 20 2006 11:40AM	0	19,081	0	0	10,812	0	select count(*) from tempdb..sysobjects where nar
sybase	sybsystemprocs	Sep 20 2006 11:45AM	0	19,079	0	0	10,812	0	select count(*) from tempdb..sysobjects where nar
sybase	sybsystemprocs	Sep 20 2006 11:43AM	3	1,985	0	0	136	3,279	select count(*) from master..sysservers where srvn
sybase	sybsystemprocs	Sep 20 2006 11:45AM	0	1,046	0	0	2,790	0	select count(*) from master..sysservers where srvn
sybase	sybsystemprocs	Sep 20 2006 11:40AM	0	1,046	0	0	2,790	16	select count(*) from master..sysservers where srvn
sybase	sybsystemprocs	Sep 20 2006 11:42AM	2	879	0	0	80	13	select count(*) from master..sysservers where srvn
sybase	sybsystemprocs	Sep 20 2006 11:42AM	6	701	0	0	844	0	USE sybase

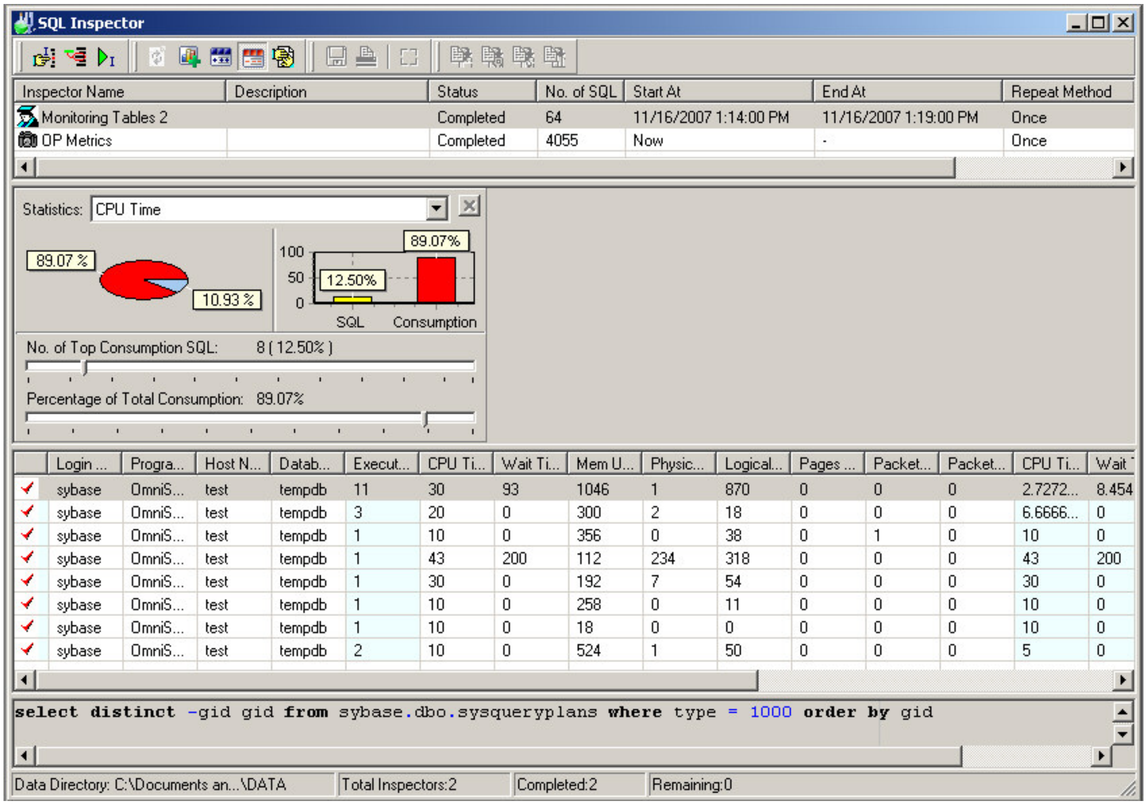
```
SELECT count(*) num_rows
FROM EMPLOYEE EMP1
WHERE EXISTS
   (SELECT 'X'
   FROM DEPARTMENT DEPT
   WHERE EMP1.EMP_DEPT = DEPT.DPT_ID AND EXISTS
      (SELECT 'X'
      FROM EMP_SMALL EMP2
      WHERE EMP2.EMP_DEPT = DEPT.DPT_ID))
```

Figure 10. Finding the most resource-intensive SQL server-wide with Spotlight

Ad-Hoc Identification of SQL—the Performance Engineer

Performance engineers have a difficult job. In theory, everyone should love them, because their job is to optimize system performance. But in reality, they need to tread lightly between the developers and DBAs: no one likes to be told that their database or code isn't performing up to par.

The SQL Inspector module within SQL Optimizer for Sybase enables the performance engineer to take a quick snapshot of SQL activity, either from the whole server or from specific spids or logins. The SQL Inspector monitors SQL statements and collects SQL performance statistics from the Adaptive Server monitoring tables or the QP Metrics. Then it graphically displays and compares SQL activity statistics, as shown in the figure below, to diagnose performance bottlenecks at the SQL level. The performance engineer can then identify the most resource-intensive SQL statements for corrective action.



Inspector Name	Description	Status	No. of SQL	Start At	End At	Repeat Method
Monitoring Tables 2		Completed	64	11/16/2007 1:14:00 PM	11/16/2007 1:19:00 PM	Once
OP Metrics		Completed	4055	Now	-	Once

Statistics: CPU Time

No. of Top Consumption SQL: 8 (12.50%)

Percentage of Total Consumption: 89.07%

	Login ...	Progra...	Host N...	Datab...	Execut...	CPU Ti...	Wait Ti...	Mem U...	Physic...	Logical...	Pages ...	Packet...	Packet...	CPU Ti...	Wait
✓	sybase	OmniS...	test	tempdb	11	30	93	1046	1	870	0	0	0	2.7272...	8.454
✓	sybase	OmniS...	test	tempdb	3	20	0	300	2	18	0	0	0	6.6666...	0
✓	sybase	OmniS...	test	tempdb	1	10	0	356	0	38	0	1	0	10	0
✓	sybase	OmniS...	test	tempdb	1	43	200	112	234	318	0	0	0	43	200
✓	sybase	OmniS...	test	tempdb	1	30	0	192	7	54	0	0	0	30	0
✓	sybase	OmniS...	test	tempdb	1	10	0	258	0	11	0	0	0	10	0
✓	sybase	OmniS...	test	tempdb	1	10	0	18	0	0	0	0	0	10	0
✓	sybase	OmniS...	test	tempdb	2	10	0	524	1	50	0	0	0	5	0

```
select distinct -gid gid from sybase.dbo.sysqueryplans where type = 1000 order by gid
```

Data Directory: C:\Documents an...\DATA | Total Inspectors:2 | Completed:2 | Remaining:0

Figure 11. Finding the most resource-intensive SQL with SQL Inspector

AFTER YOU'VE IDENTIFIED THE PROBLEMATIC SQL

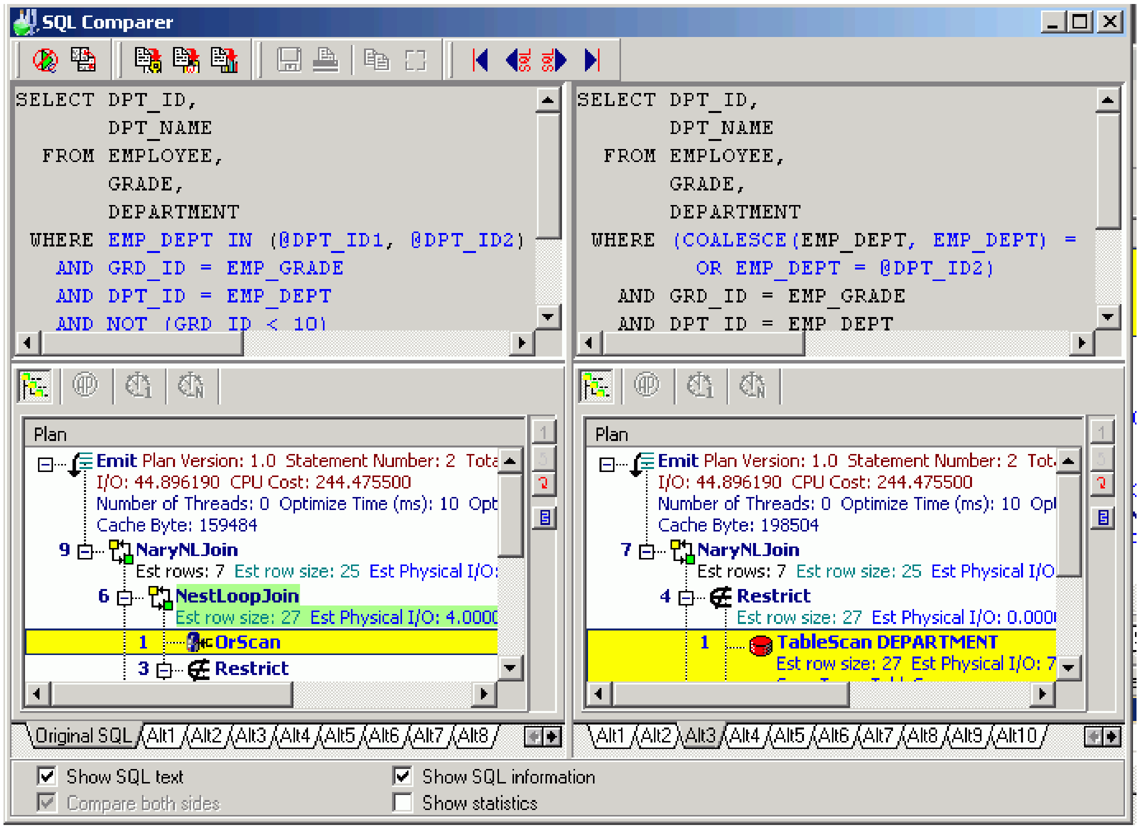
You may be wondering what to do with all of the problematic SQL you've identified. Using SQL Optimizer for Sybase, you have four options:

- Tune the SQL
- Test for concurrent user scalability
- Recommend indexes
- Perform "what if" analysis

Tune the SQL

SQL Optimizer can automatically perform a syntax rewrite of problematic SQL statements. Many SQL statements can return the same result set, but only a few of them are likely to be efficient. To ensure optimal performance, a SQL statement needs to be transformed in multiple ways to produce every equivalent SQL statement, and then those alternatives need to be evaluated to find the most efficient SQL.

SQL Optimizer analyzes source SQL statements and then applies a unique artificial intelligence engine to reproduce a list of semantically equivalent and syntactically correct SQL statements. Then it compares the performance of each alternative statement, as shown in the figure below, so you can perform a test run to see if the alternative statement offers performance improvement in your environment.



```
SELECT DPT_ID,
       DPT_NAME
  FROM EMPLOYEE,
       GRADE,
       DEPARTMENT
 WHERE EMP_DEPT IN (@DPT_ID1, @DPT_ID2)
   AND GRD_ID = EMP_GRADE
   AND DPT_ID = EMP_DEPT
   AND NOT (GRD_ID < 10)
```

```
SELECT DPT_ID,
       DPT_NAME
  FROM EMPLOYEE,
       GRADE,
       DEPARTMENT
 WHERE (COALESCE(EMP_DEPT, EMP_DEPT) =
        OR EMP_DEPT = @DPT_ID2)
   AND GRD_ID = EMP_GRADE
   AND DPT_ID = EMP_DEPT
```

```
Emit Plan Version: 1.0  Statement Number: 2  Tota
I/O: 44.896190  CPU Cost: 244.475500
Number of Threads: 0  Optimize Time (ms): 10  Opt
Cache Byte: 159484
9  NaryNLJoin
     Est rows: 7  Est row size: 25  Est Physical I/O:
  6  NestLoopJoin
       Est row size: 27  Est Physical I/O: 4.0000
     1  OrScan
     3  Restrict
```

```
Emit Plan Version: 1.0  Statement Number: 2  Tot
I/O: 44.896190  CPU Cost: 244.475500
Number of Threads: 0  Optimize Time (ms): 10  Op
Cache Byte: 198504
7  NaryNLJoin
     Est rows: 7  Est row size: 25  Est Physical I/O
  4  Restrict
       Est row size: 27  Est Physical I/O: 0.000
     1  TableScan DEPARTMENT
          Est row size: 27  Est Physical I/O: 7
```

☑ Show SQL text ☑ Show SQL information
☑ Compare both sides ☐ Show statistics

Figure 12. Generating and comparing alternative SQL statements with SQL Optimizer

Test for Concurrent User Scalability

The results of the test run from the SQL Optimizer are for a single user running each statement. However, multiple concurrent users typically run a given application. Performing a scalability test with Benchmark Factory for Databases will enable you to determine which statement from the SQL Optimizer performs best at different user loads.

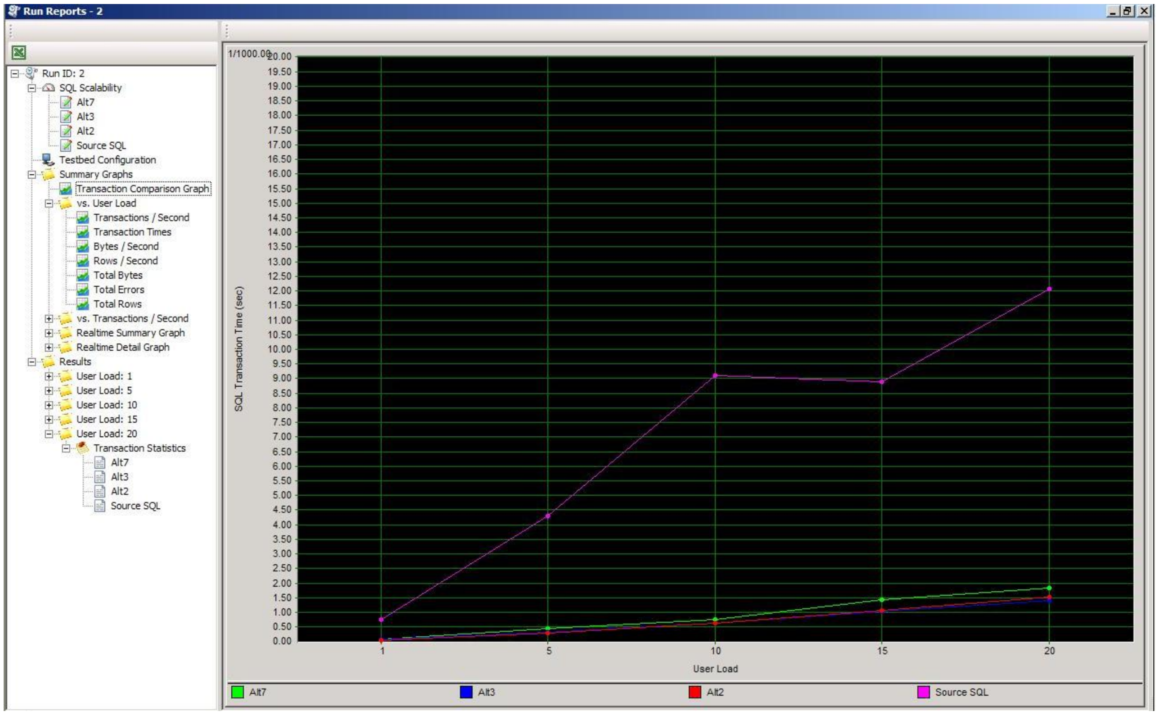
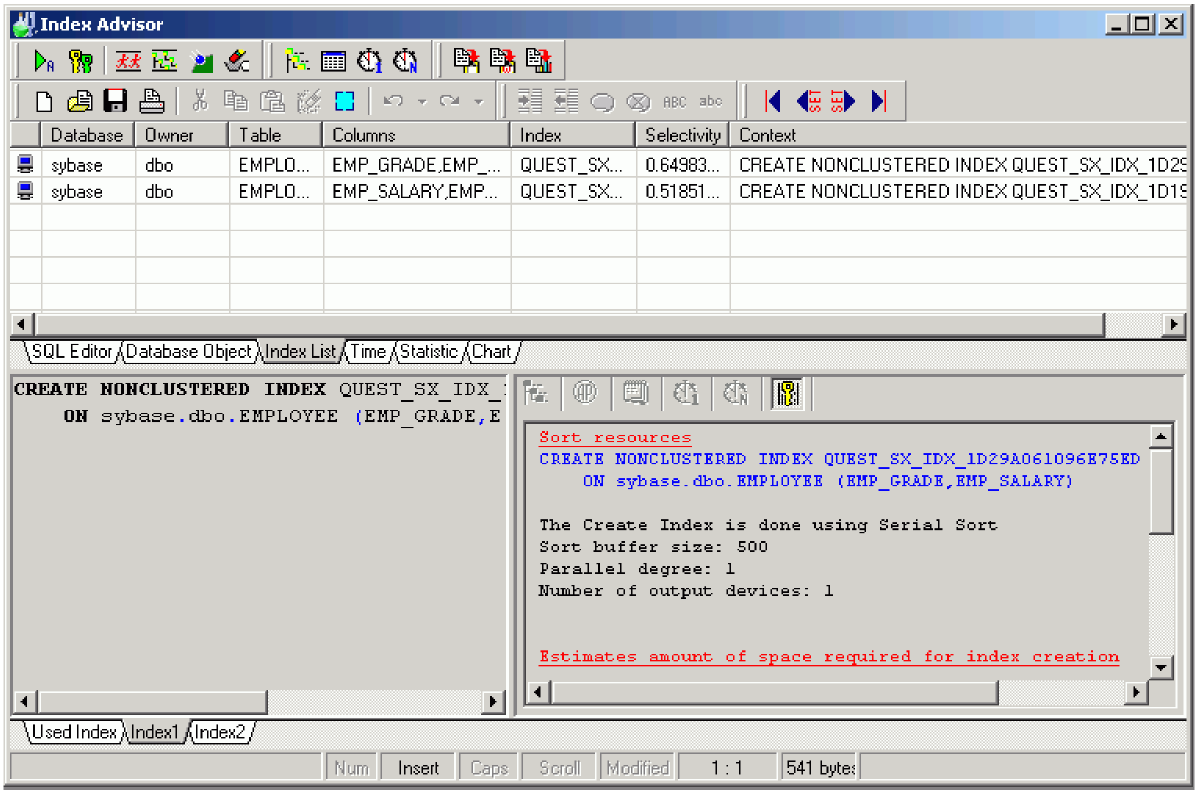


Figure 13. Testing SQL statements at different load with Benchmark Factory for Databases

Recommend Indexes

If you cannot modify problematic source code, another alternative is to modify the indexes on your tables. This approach is useful when your best efforts to optimize a SQL statement are insufficient—for example, when the original problematic SQL is actually faster than the alternatives, or the alternatives improve performance only marginally.

SQL Optimizer for Sybase's Index Advisor automatically generates a set of index recommendations for a problematic SQL statement by analyzing the structure of a SQL statement (including the search arguments and table join conditions) and identifying all the tables and indexes it uses. You can also add your own candidate indexes and group candidates into compound indexes, and then evaluate the effect the new indexes could have on database performance.



Database	Owner	Table	Columns	Index	Selectivity	Context
sybase	dbo	EMPLO...	EMP_GRADE,EMP_...	QUEST_SX...	0.64983...	CREATE NONCLUSTERED INDEX QUEST_SX_IDX_1D29
sybase	dbo	EMPLO...	EMP_SALARY,EMP...	QUEST_SX...	0.51851...	CREATE NONCLUSTERED INDEX QUEST_SX_IDX_1D19

```
CREATE NONCLUSTERED INDEX QUEST_SX_IDX_1
    ON sybase.dbo.EMPLOYEE (EMP_GRADE,E
```

```
Sort resources
CREATE NONCLUSTERED INDEX QUEST_SX_IDX_1D29A061096E75ED
    ON sybase.dbo.EMPLOYEE (EMP_GRADE,EMP_SALARY)

The Create Index is done using Serial Sort
Sort buffer size: 500
Parallel degree: 1
Number of output devices: 1


Estimates amount of space required for index creation
```

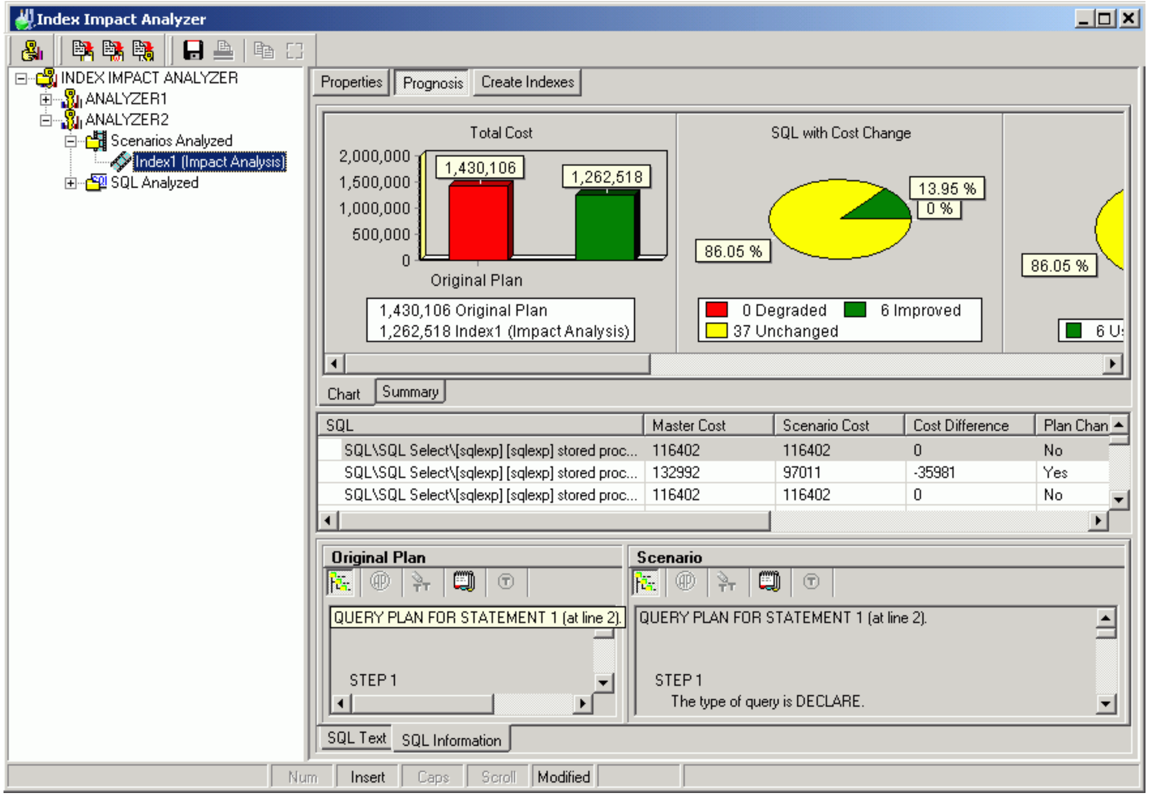
Figure 14. Generating index recommendations for a problematic SQL statement with Index Advisor

Perform "What-If" Analysis

We may want to know ahead of time what will happen to SQL performance if we make a change, such as adding an index, migrating to ASE 15, or making a sp_configure change. The analysis modules within SQL Optimizer for Sybase help answer those questions.

From a common SQL repository, you can analyze query plans using before and after scenarios and show the impact of the change. You can get an overall application view, and then drill down to individual SQLs for detailed analysis.

The figure below shows an analysis provided by the Index Impact Analyzer.



1,430,106 Original Plan
1,262,518 Index1 (Impact Analysis)

🟥 0 Degraded 🟩 6 Improved
🟨 37 Unchanged

SQL	Master Cost	Scenario Cost	Cost Difference	Plan Chan
SQL\SQL Select\[sqlexp] [sqlexp] stored proc...	116402	116402	0	No
SQL\SQL Select\[sqlexp] [sqlexp] stored proc...	132992	97011	-35981	Yes
SQL\SQL Select\[sqlexp] [sqlexp] stored proc...	116402	116402	0	No

Original Plan:

```
QUERY PLAN FOR STATEMENT 1 (at line 2).

  STEP 1
```

Scenario:

```
QUERY PLAN FOR STATEMENT 1 (at line 2).

  STEP 1
    The type of query is DECLARE.
```

Figure 15. "What if" analysis from the Index Impact Analyzer

SUMMARY

DBAs, developers, QA analysts, and performance engineers all need to identify problematic SQL statements, and Quest offers tools to fit each of their differing needs.

Foglight provides unattended monitoring of applications, alerting DBAs to issues as they arise. In particular, Foglight can be used in the ASE environment to capture and identify poorly-performing SQL statements. Since DBAs are often fighting multiple fires at once, this automated capture tends to suit their needs best.

The SQL Scanner of SQL Optimizer for Sybase enables developers to quickly extract SQL from application code and pinpoint the SQL most in need of tuning. It simplifies the tedious task of wading through stored procedures and manually pulling out SQL statements.

Spotlight on Sybase provides visibility into the current activity of Sybase ASE so the QA analyst can easily determine what a load test is doing to the server. The analyst can also drill down to identify the users and SQL statements consuming the most resources.

Finally, the SQL Inspector of SQL Optimizer for Sybase enables performance engineers to take a snapshot of database activity. By analyzing the most resource-intensive statements, they can identify candidates for performance tuning.

To explore the capabilities of each tool and decide which are best for your environment, visit http://www.quest.com/sybase . You can get more information on these solutions and download trial versions to find the right combination to suit your needs.

ABOUT THE AUTHOR

Darren Mallette has been with Quest Software for five years. He currently serves as a Solutions Architect helping customers get the most out of their database and virtualization platforms. In the past 15 years, he has worked with Oracle, SQL Server, Sybase, DB2 and VMWare technologies with a focus on performance monitoring and tuning. When not staring at the 1> prompt, he enjoys spending time with his family and competing in strongman competitions.

About Quest Software, Inc.

Quest Software, Inc., a leading enterprise systems management vendor, delivers innovative products that help organizations get more performance and productivity from their applications, databases, Windows infrastructure and virtual environments. Quest also provides customers with client management through its ScriptLogic subsidiary and server virtualization management through its Vizioncore subsidiary. Through a deep expertise in IT operations and a continued focus on what works best, Quest helps more than 100,000 customers worldwide meet higher expectations for enterprise IT. Quest Software helps organizations deliver, manage and control complex database environments through award-winning products for Oracle, SQL Server, IBM DB2, Sybase and MySQL. Quest Software can be found in offices around the globe and at www.quest.com.

Contacting Quest Software

Phone:	949.754.8000 (United States and Canada)
Email:	info@quest.com
Mail:	Quest Software, Inc. World Headquarters 5 Polaris Way Aliso Viejo, CA 92656 USA
Web site	www.quest.com

Please refer to our Web site for regional and international office information.

Contacting Quest Support

Quest Support is available to customers who have a trial version of a Quest product or who have purchased a commercial version and have a valid maintenance contract. Quest Support provides around the clock coverage with SupportLink, our web self-service. Visit SupportLink at http://support.quest.com

From SupportLink, you can do the following:

- Quickly find thousands of solutions (Knowledgebase articles/documents).
- Download patches and upgrades.
- Seek help from a Support engineer.
- Log and update your case, and check its status.

View the ***Global Support Guide*** for a detailed explanation of support programs, online services, contact information, and policy and procedures. The guide is available at: http://support.quest.com/pdfs/Global Support Guide.pdf