

IDERA®

SQL Diagnostic Manager for SQL Server

KEEP YOUR SQL SERVERS RUNNING SMOOTHLY

WITH SQL DIAGNOSTIC MANAGER

BY GREG ROBIDOUX

INTRODUCTION

As an application or database user, these are the things that matter: is it easy to use, does it do what I need it to do and does it perform well? Two of these items are front end related to the design of the application and whether requirements were followed. The third item can be application or database specific, or both. In order to ensure there are no issues from the database perspective, there are several things that developers and DBAs can do both during development stages and production use of the application.

First off, you want to make sure queries are tuned and optimized for the best performance. This includes having the right indexes in place to writing efficient SQL code. Different data elements can be captured and analyzed to make improvements, such as adding indexes, tweaking code and perhaps some configuration changes. Such things as reviewing execution plans can do this, looking at wait stats, and reviewing server and SQL Server metrics.

Often, during application and database development, a significant workload of data and processes is not always in place to get a true understanding of how well the application will perform when it is rolled out. So when the application is released, issues may come into play, such as blocking, deadlocks, timeouts, tempdb contention, and resource waits such as input/output (IO), central processing unit (CPU), network, and more.



WHY ARE WE DATABASE ADMINISTRATORS AND DEVELOPERS?

I am sure most people that do database work or build applications have creativity. They want to use their skills to make things work better, whether that be improvements to existing features or rolling out new features to improve the application and database. I think these people like the challenge and always want to solve the problem. I haven't found many of these people that like to get bogged down with reading and reviewing endless logs to find issues. What ends up happening is that if the investigation work takes too much time, often they will switch gears. They will then figure out how to solve this fresh problem, how can I automate the data collection and analysis and make better decisions. But the issue with this approach is that it takes valuable resources away from what is most important, solving the application or database issue, it is not to create some new application or processes to collect and analyze data. This is where tools like SQL Diagnostic Manager (SQL DM) come into play.

USING SQL DIAGNOSTIC MANAGER AS A DEVELOPER

Development is a big undertaking and most of the time it focuses on making sure all the features are in place and working. Once this is done, tuning for performance often occurs. But in order to make adjustments, developers need data to understand where the issue is and then how to solve the problem. By using a tool like SQL DM, data collection can occur all the time and when there is a need to analyze performance, all the data is accessible. In addition, by using the SQL Query Tuner add-on, you can get automated suggestions on how to change your code to make performance improvements.

“

I am sure most people that do database work or build applications have creativity. They want to use their skills to make things work better...

”

USING SQL DIAGNOSTIC MANAGER AS A DATABASE ADMINISTRATOR

As a database administrator (DBA), constantly monitoring database systems for performance, corruption, and data issues is an ongoing task. It is also an overwhelming task depending on how large the database environment is and the number of resources there are to manage the environment. The need to have good data is key to making informed decisions. However, there are so many tools that can collect and analyze the data that often the metrics you need were not captured and therefore you cannot solve the problem. By using SQL DM, the data collection, data aggregation, data presentation, data linking, and historical data are captured. That data is ready anytime, therefore giving the DBA or whoever manages the SQL Server environment the keys to solving any issue and to keep the databases running as optimal as possible. Without the use of SQL DM, most of the time is spent setting up data collection mechanisms and writing countless queries or programs to decipher the data to help you make the right decisions. To further assist with data analysis, you can use the SQL Workload Analysis for SQL Server add-on to analyze workloads of your database systems and provide additional insights on performance improvements.

COMMON DEVELOPER TASKS WHERE SQL DIAGNOSTIC MANAGER CAN HELP

There are several cases where developers can take advantage of SQL DM capabilities. The main area is to find poor performing code, review query plans and making adjustments. This could be done in production systems, of course. But this can also be done in development environments where data is collected and reviewed to help you figure out what needs to be addressed. Other things developers may do is importing and exporting data from SQL Server. This could be another area that SQL DM can monitor activity and help you find bottlenecks so you can make improvements.

COMMON DATABASE ADMINISTRATOR TASKS WHERE SQL DIAGNOSTIC MANAGER CAN HELP

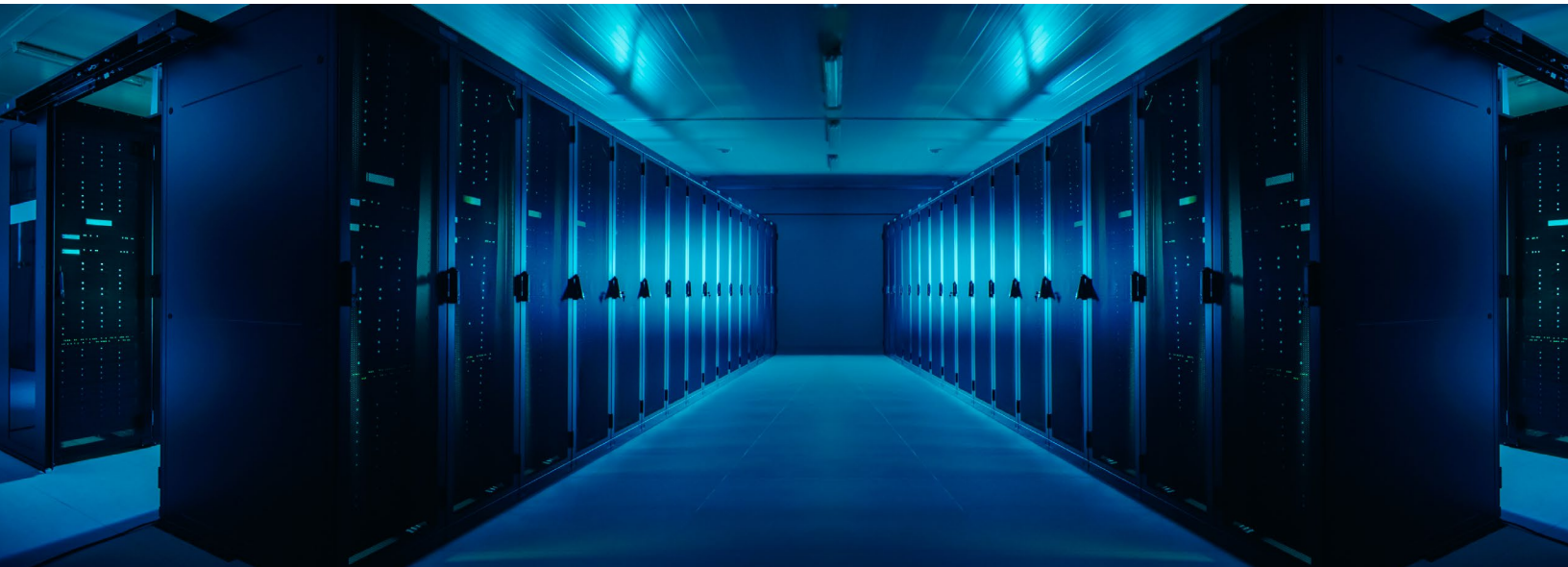
There are several things that DBAs do on a day-to-day basis. Some tasks are procedural, like setting up maintenance tasks, rolling out code, setting up security, loading data, making configuration changes, etc. But a good amount of time is spent on monitoring, analyzing and fixing issues to make sure databases are an asset and not a limitation. Would you not rather spend your time fixing problems instead of setting up data collection and analysis processes?

There are so many aspects of SQL Server that could affect performance, data integrity, and usefulness of applications that DBAs need to monitor and troubleshoot, and here are some of those items:

- Blocking
- Deadlocks
- Tempdb contention
- Availability Group issues
- Configuration changes
- Failed jobs
- Timeouts
- Resource issues like IO, memory, CPU
- Long running queries
- Execution plan regression
- Missing indexes
- Index fragmentation

Any of these items can occur anytime. Therefore, the ability for SQL DM to collect data non-stop and provide a historical repository allows you to look at issues that are happening right now or something that occurred an hour ago, a week ago or longer.

Another limiting factor to a DBA is that you can only work on one thing at a time. But you still have several SQL Server instances and databases to manage. How can you determine what is critical and needs immediate attention versus focusing on a known issue? Are you proactive and address things as they arise or are you reactive and wait for someone else to let you know there is a problem?

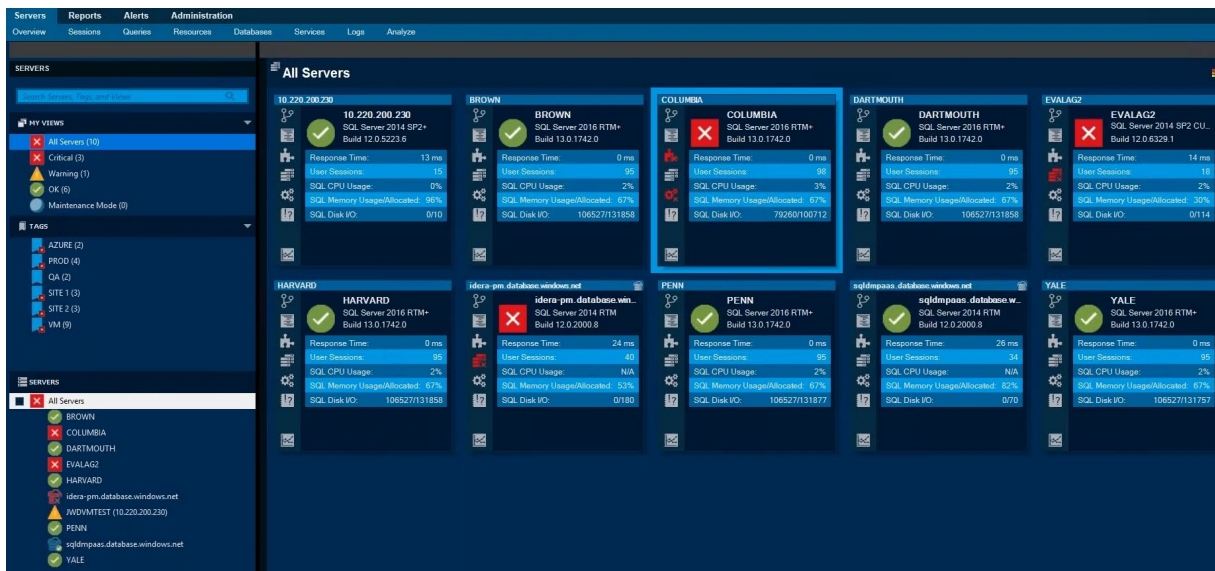


SQL DM IN ACTION

In the following sections, we will look at SQL DM in action and how you can use this tool to help with common database monitoring and management tasks.

ENTERPRISE MANAGEMENT

There are many advantages of using SQL DM over building your own tools and one of these is the enterprise management view of all your SQL Servers in one place. We can see the entire environment fast and also see issues that need attention. With constant data collection and analysis, SQL DM can let you know where to focus your time.

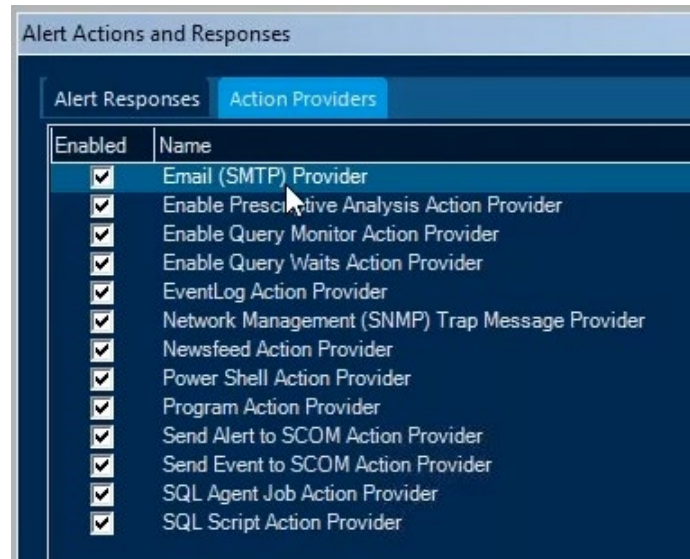


By drilling into a particular server that needs attention, SQL DM lets you know what issues need to be addressed. All of this is configurable, so that SQL DM only notifies you of things that are important to your system.

| Change | Time | Instance | Summary |
|---|----------------------|----------|---|
| Instance : COLUMBIA (10 items) | | | |
| Remained Critical | 8/5/2021 10:14:38 AM | COLUMBIA | Blocked sessions count is 5 |
| Remained Critical | 8/5/2021 10:14:38 AM | COLUMBIA | Session 119 blocking for 1744 seconds, since 8/5/2021 9:43:10 AM. |
| Remained Critical | 8/5/2021 10:14:38 AM | COLUMBIA | Session 125 blocking for 1724 seconds, since 8/5/2021 9:43:30 AM. |
| Remained Critical | 8/5/2021 10:14:38 AM | COLUMBIA | SQL Agent job [Idera SQL Enterprise Job Manager, Calendar Maintena... |
| Remained Critical | 8/5/2021 10:14:38 AM | COLUMBIA | SQL Agent job [Idera SQL Enterprise Job Manager, Grooming Data [Ide... |
| Remained Critical | 8/5/2021 10:14:38 AM | COLUMBIA | SQL Agent job [SQLsafe Full backup for policy SIMPLE RECOVERY M... |
| Remained Critical | 8/5/2021 10:14:38 AM | COLUMBIA | SQL Agent job [Deadlock Job Healthcare(second)] completed with a sta... |
| Remained Warning | 8/5/2021 10:14:38 AM | COLUMBIA | Page life expectancy for COLUMBIA is currently 2881. |
| Remained Warning | 8/5/2021 10:14:38 AM | COLUMBIA | Total Deadlocks Per Hour is 5.00 |
| Raised to Warning | 8/5/2021 10:12:41 AM | COLUMBIA | Recommendations Are Available |
| Instance : EVALAG2 (2 items) | | | |
| Instance : idera-pm.database.windows.net (16 items) | | | |
| Instance : sqlmpaas.database.windows.net (6 items) | | | |

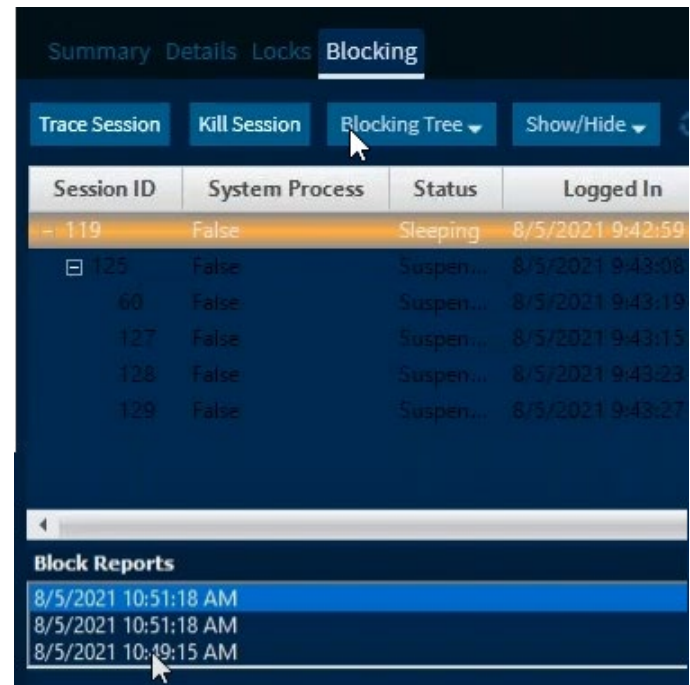
AUTOMATE FIXES OR SEND ALERTS

Another outstanding feature of SQL DM is the ability to either send out alerts or automatically fix a problem. The following is a list of actions that can be taken when an event occurs. This can all be automated using these action providers when there is something that you know happens and you just need to run a script to resolve.



BLOCKING

Blocking is a normal activity that occurs in SQL Server and it does that to control data integrity and make sure only one resource can change a set of data at one time. The problem with blocking is if a session is blocked for a long period, this can cause slowdowns and multiple sessions being blocked, creating a blocking chain. In SQL DM, if there are blocking issues, you can discover fast what is occurring now and for previous points in time. The first screen below shows when blocking occurred and if you drill into one of the block reports, you can view the detail of the blocked process and the blocking process.



Here are the details of the blocking process and blocked process. So you have a history of when blocking occurred. But more importantly, you have the sessions and statements that were causing the blocking.

The screenshot displays two panels from the SQL Server Enterprise Manager interface. The top panel, titled "Blocking Process Details", shows session 119 in a sleeping state. Its last command is a transaction with a table lock: `BEGIN TRANSACTION` followed by `SELECT ShipName FROM dbo.Orders WITH (TABLOCKX)` and `SELECT * FROM dbo.Orders`. The bottom panel, titled "Blocked Process Details", shows session 125 in a suspended state. Its last command is `USE Northwind` followed by `SELECT ShipName FROM dbo.Orders`. The blocked process is waiting for a resource (OBJECT: 17:149575571:0) for 3914684 ms.

| Blocking Process Details | | |
|---|---|---------------------|
| Connection | Usage | Lock Information |
| Session: 119 | Last Batch Started: 8/5/2021 9:43:02 AM | Wait Type: sleeping |
| Status: sleeping | Last Batch Completed: 8/5/2021 9:43:03 AM | |
| Host: JWDVM3 | Transaction Id: 2544371 | |
| User: JWDVM3\Jwilkes | Transaction Name: | |
| Execution Context: 0 | Open Transactions: 1 | |
| Application: Microsoft SQL Server Management Studi... | Last Command: | |
| Database: Northwind | <code>BEGIN TRANSACTION</code> <code>SELECT ShipName FROM dbo.Orders WITH (TABLOCKX)</code> <code>SELECT * FROM dbo.Orders</code> | |

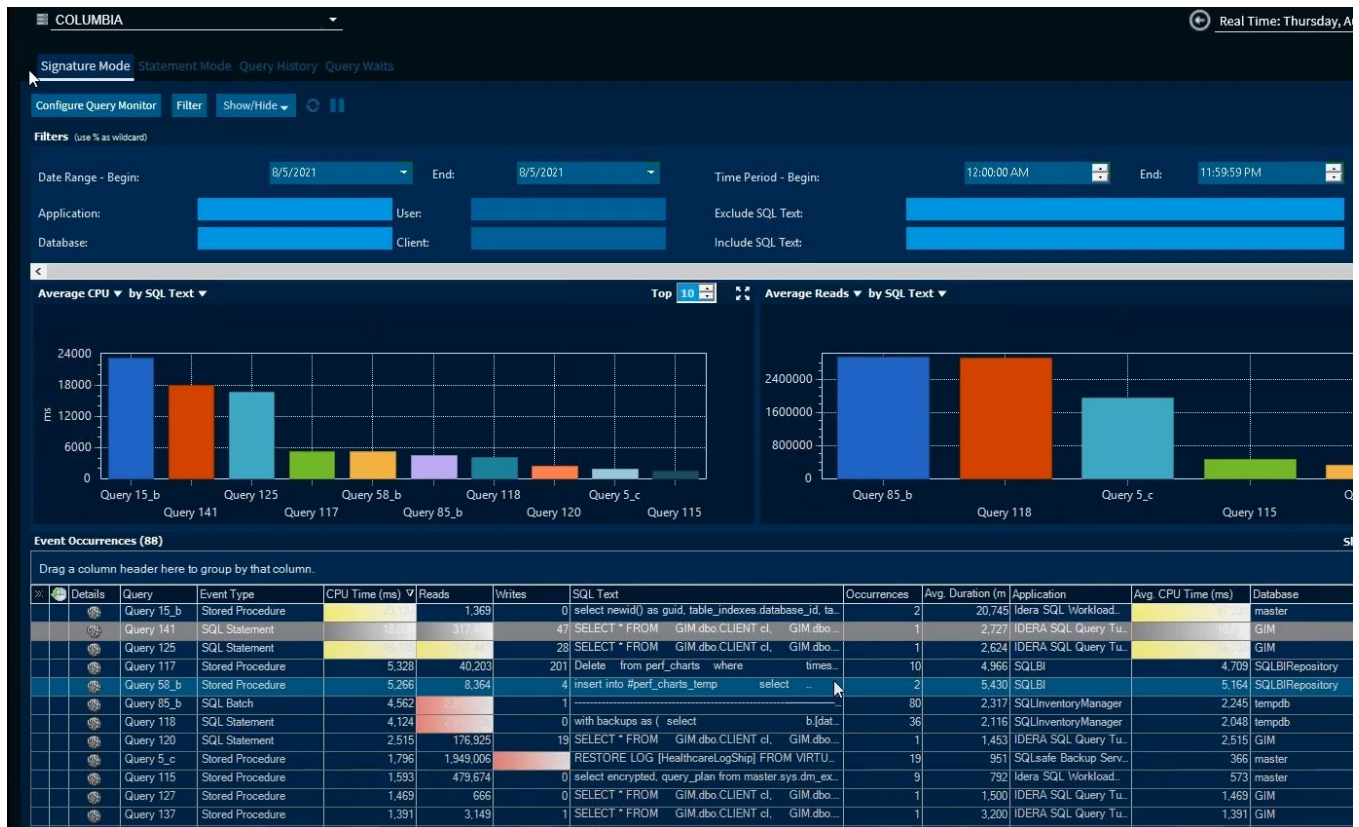
| Blocked Process Details | | |
|---|--|---------------------------------------|
| Connection | Usage | Lock Information |
| Session: 125 | Last Batch Started: 8/5/2021 9:43:10 AM | Wait Time (ms): 3914684 |
| Status: suspended | Last Batch Completed: 8/5/2021 9:43:10 AM | Wait Type: suspended |
| Host: JWDVM3 | Last Transaction Started: 8/5/2021 9:43:10 AM | Wait Resource: OBJECT: 17:149575571:0 |
| User: JWDVM3\Jwilkes | Transaction Id: 2707541 | |
| Execution Context: 0 | Transaction Name: SELECT | |
| Application: Microsoft SQL Server Management Studi... | Open Transactions: 0 | |
| Database: Northwind | Last Command: | |
| | <code>USE Northwind</code> <code>SELECT ShipName FROM dbo.Orders</code> | |

DEADLOCKING

Deadlocking is another common problem that can occur in an SQL Server database. This is like blocking. But instead of being blocked forever, one of the sessions is chosen as the deadlock victim and is rolled back so the other process can continue. SQL DM allows you to drill into the deadlock to find the sessions that are involved as well as the code that is causing the problem for quick resolution. You can also see a history of deadlocks like was shown for blocking. This lets you determine if there is a pattern with particular processes and also how frequently deadlocks occur.

QUERY METRICS

There are many things that can cause slowdowns in SQL Server, such as resources like disk, memory, and network. But the one place where DBAs and developers have the most control and can make the biggest impact is tuning queries. SQL DM collected this data non-stop and then aggregated it so you can focus time on code fixes that would have the biggest impact overall versus just focusing on the worst performing code. In the below screenshot, we can see how SQL DM rolls up similar statements and provides aggregate insights, so you know where to spend your time.



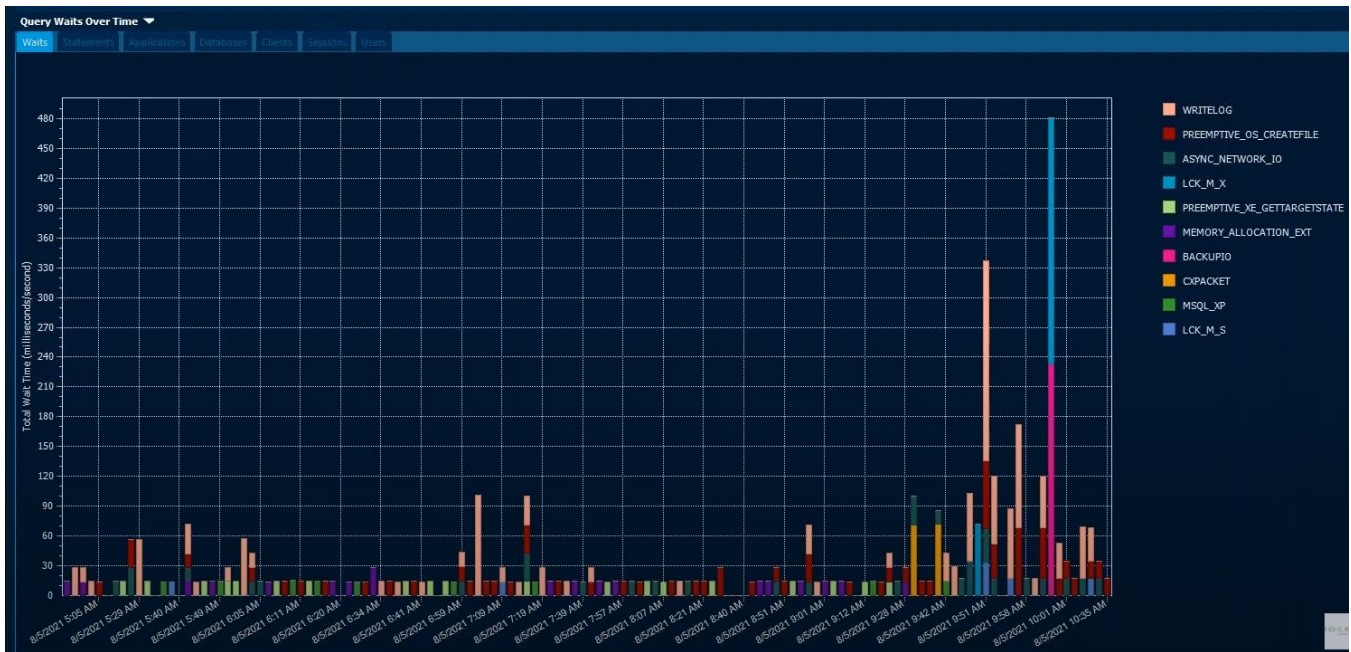
“

SQL DM collected this data... then aggregated it so you can focus time on code fixes that would have the biggest impact overall versus just focusing on the worst performing code.

”

WAIT STATS

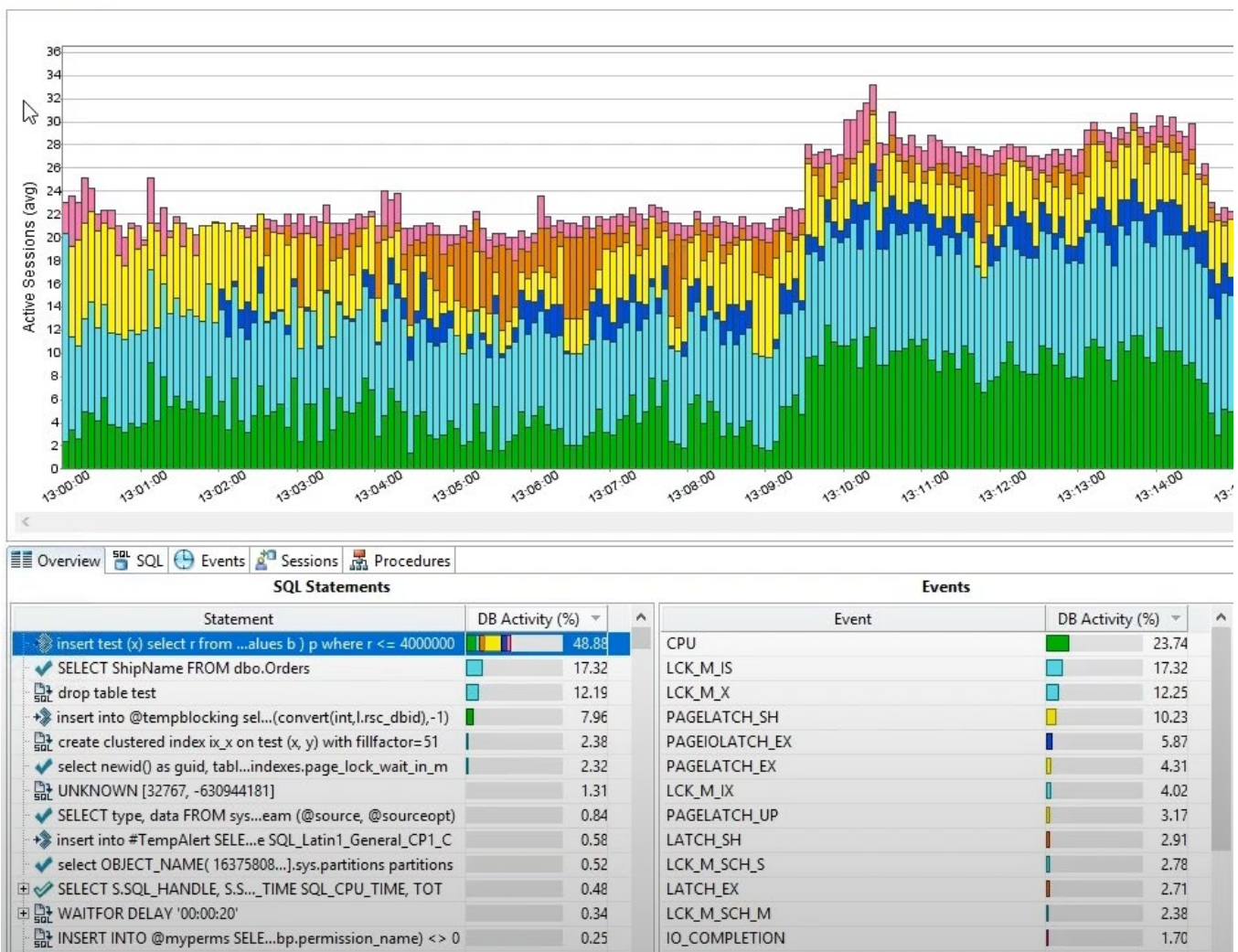
As mentioned several times already, there are several things that could cause performance issues for a query. These can be identified by breaking down the processing time for a query into various wait states, such as locks, IO, CPU, etc. This can give you an idea of what steps you might take to improve the overall performance or for a specific statement. In the screenshot below, we can see a breakdown of wait times for various time periods. SQL DM then let's you drill into the specified time to get a sense of queries that were running and identify what problem queries need to be reviewed.



SQL QUERY TUNER

Writing and tuning SQL code are two different things. There are many ways to write code so that it does the intended function. But there are also ways to make code more efficient. In most cases, development techniques are followed in repetition based on what the DBA or developer knows. Sometimes, this might yield the best performing code. But in other cases, perhaps not. To tune a query, you can review execution plans and identify where the problem is and then change to optimize the code. Another approach to tuning queries is to use the SQL Query Tuner add-on for SQL DM. You can find what statements need to be addressed by looking at duration and wait stats and then use SQL Query Tuner to suggest automatically different ways to rewrite the code along with metrics to show overall improvements. Below, we can view an aggregate view of statements run over time.

Profile Session



If we drill into a particular statement, we can then get suggestions on how to improve the code. Besides showing the improvements, the SQL Query Tuner also generates the code for you to use in your development. Below we can see several suggestions to improve the code along with metrics for each case, which includes elapsed time, reads, CPU, and more.

Tuning Statements Generate cases Perform detail analysis

Schema Same count Same order

| Statement | | | | | | | Time | | Analysis | | | | |
|-------------------------------------|----------|---------|--------|-------------|--------|-------|-------------|--------------|----------|---------|---|---|---|
| | Name | Catalog | Schema | Text | Tables | Views | Elapsed (s) | Improved (s) | Cases | Indexes | | | |
| <input checked="" type="checkbox"/> | SELECT 2 | GIM | dbo | select from | 6 | 0 | 1.00 | 0.20 | 50 | 7 | 1 | 2 | 4 |

Generated Cases Winners found:122 Percent improved:0.92%

| SQL Statements and Cases | | Cost | Executi...istics | Other Execution Statistics | | |
|--------------------------|-----------------------------|---------|------------------|----------------------------|---------------|--------------|
| Name | Text | Value | Elapsed Time (s) | Physi...eads | Logical Reads | CPU Time (s) |
| SELECT 2 | select from GIM.dbo.CLIENT, | 1.6702 | 1.00 | 0 | 644 | 0.97 |
| [Transitiv...sformation | | 1.6549 | 0.27 | 0 | 647 | 0.23 |
| IGNORE_N...RE_INDEX | | 1.6702 | 0.92 | 0 | 690 | 0.92 |
| TABLE_INDEX1 | | 1.675 | 0.92 | 0 | 653 | 0.84 |
| TABLE_INDEX2 | | 1.6771 | 0.72 | 0 | 666 | 0.72 |
| TABLE_INDEX1 | | 1.6771 | 0.80 | 0 | 666 | 0.77 |
| [EXISTS_TO...LRewrite | | 1.6777 | 0.69 | 0 | 644 | 0.62 |
| TABLE_INDEX3 | | 1.6988 | 0.70 | 0 | 668 | 0.67 |
| TABLE_INDEX4 | | 1.7025 | 0.74 | 0 | 648 | 0.66 |
| TABLE_INDEX2 | | 1.7532 | 0.67 | 0 | 1059 | 0.67 |
| FORCESEEK4 | | 1.999 | 0.67 | 0 | 2712 | 0.66 |
| TABLE_INDEX4 | | 2.8682 | 0.72 | 0 | 6502 | 0.66 |
| FORCESEEK2 | | 3.5262 | 0.70 | 0 | 56666 | 0.67 |
| FORCE_ORDER | | 3.6737 | 0.81 | 1032 | 3117 | 0.75 |
| TABLE_INDEX | | 5.3455 | 0.69 | 0 | 20296 | 0.67 |
| TABLE_INDEX5 | | 5.3455 | 0.85 | 0 | 20296 | 0.84 |
| FORCESEEK4 | | 7.1323 | 0.72 | 0 | 60077 | 0.75 |
| LOOP_JOIN_QUERY | | 22.6726 | 2.37 | 0 | 317450 | 17.76 |

THE POWER OF SQL DIAGNOSTIC MANAGER

As a seasoned SQL Server DBA or developer, you already know that there is an abundance of tools you can use to collect server and database information to help you make informed decisions. As someone just starting out, you will find out fast that this is the case once you monitor and improve your systems.

SQL DM is based on these data collection tools that are available for SQL Server. But it provides a cohesive and interactive environment for data presentation, linking of objects, and faster resolution time. The need to set up different collection methods and analysis efforts all goes away and SQL DM lets you use a standardized interface for problem solving. I often wonder why when I observe people creating their own tools to recreate what already exists in a tool like SQL DM. You can never build a tool as robust and comprehensive nor would you be able to keep it current as Microsoft extends the feature set in SQL Server. Take the time to download your free trial and begin taking advantage of everything SQL DM offers and begin improving your environment right now.

“

You can never build a tool as robust and comprehensive [as SQL DM] nor would you be able to keep it current as Microsoft extends the feature set in SQL Server.

”

ABOUT THE AUTHOR

Greg Robidoux is the President and founder of Edgewood Solutions, a technology services company delivering services and solutions for Microsoft SQL Server. Greg is also a co-founder of MSSQLTips.com. He has been working with SQL Server since 1999 and has authored many articles and delivered several presentations online and at local and national SQL Server events.

[Start for Free](#)