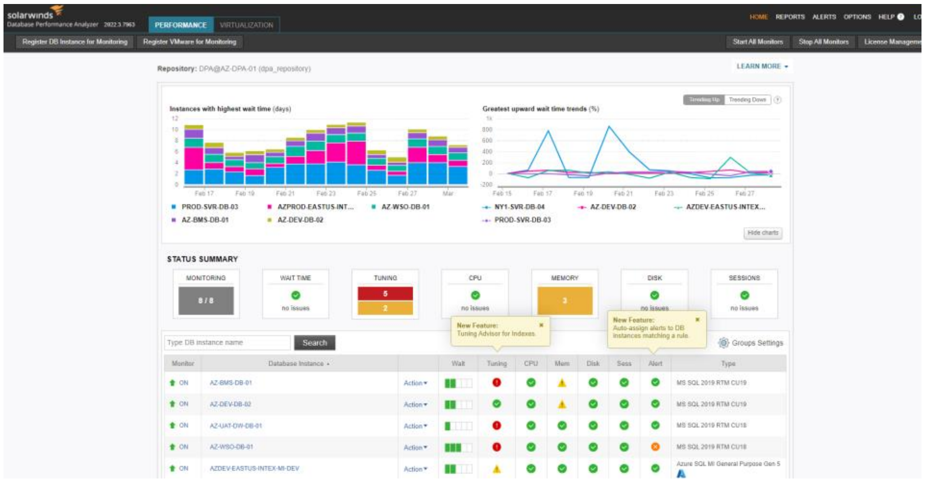
Step by step guidelines DPA tool.

1. Log in DPA web page [DPA - Login Page (cifc.com)]
2. Once you are inside, you will see this main page:



Repository: DPA@AZ-DPA-01 (dpa_repository)

New Feature:
Tuning Advisor for Indexes

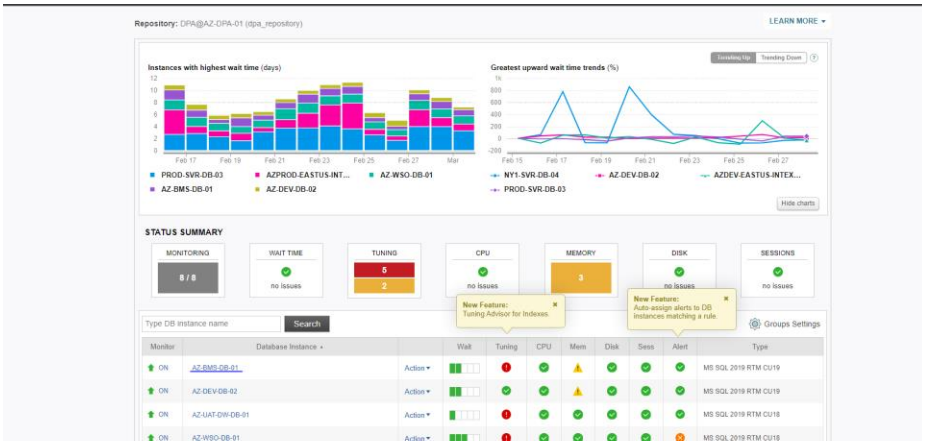
New Feature:
Auto-assign alerts to DB instances matching a rule.

Monitor	Database Instance		Wait	Tuning	CPU	Mem	Disk	Sess	Alert	Type
ON	AZ-BMS-DB-01	Action		!	✓	⚠	✓	✓	✓	MS SQL 2019 RTM CU19
ON	AZ-DEV-DB-02	Action		✓	✓	⚠	✓	✓	✓	MS SQL 2019 RTM CU19
ON	AZ-UAT-DW-DB-01	Action		!	✓	✓	✓	✓	✓	MS SQL 2019 RTM CU18
ON	AZ-WSO-DB-01	Action		!	✓	✓	✓	✓	✗	MS SQL 2019 RTM CU18
ON	AZDEV-EASTUS-INTEX-MI-DEV	Action		⚠	✓	✓	✓	✓	✓	Azure SQL MI General Purpose Gen 5

This is a general view about different trends in all the monitored servers.

3. Click on one the machines/servers that you want to track:

In this case I chose the **AZ-BMS-DB-01** server



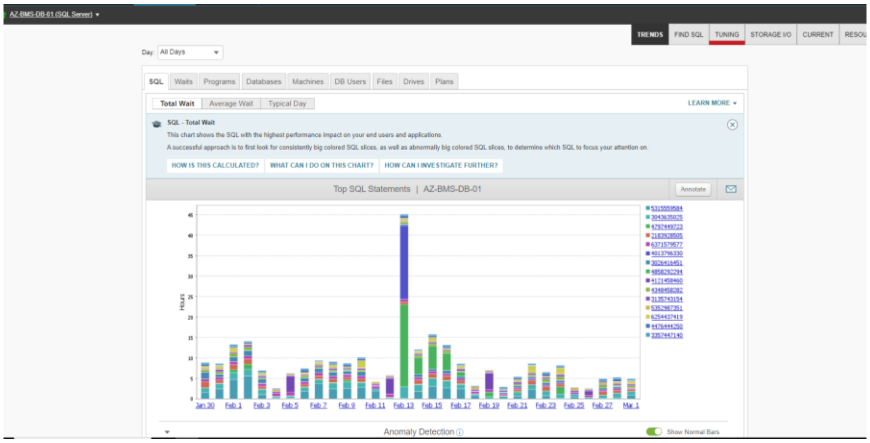
Repository: DPA@AZ-DPA-01 (dpa_repository)

New Feature:
Tuning Advisor for Indexes

New Feature:
Auto-assign alerts to DB instances matching a rule.

Monitor	Database Instance		Wait	Tuning	CPU	Mem	Disk	Sess	Alert	Type
ON	AZ-BMS-DB-01	Action		!	✓	⚠	✓	✓	✓	MS SQL 2019 RTM CU19
ON	AZ-DEV-DB-02	Action		✓	✓	⚠	✓	✓	✓	MS SQL 2019 RTM CU19
ON	AZ-UAT-DW-DB-01	Action		!	✓	✓	✓	✓	✓	MS SQL 2019 RTM CU18
ON	AZ-WSO-DB-01	Action		!	✓	✓	✓	✓	✗	MS SQL 2019 RTM CU18

4. Here you will find several useful options:

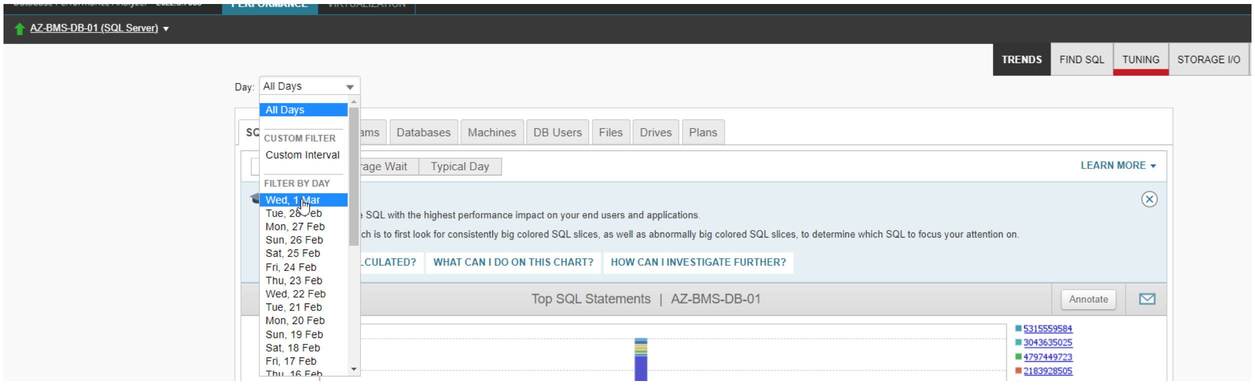


SQL - Total Wait

This chart shows the SQL with the highest performance impact on your end users and applications.

A successful approach is to first look for consistently big colored SQL slices, as well as abnormally big colored SQL slices, to determine which SQL to focus your attention on.

5. Let’s filter the statistics for the required day, clicking in Day dropbox:



AZ-BMS-DB-01 (SQL Server)

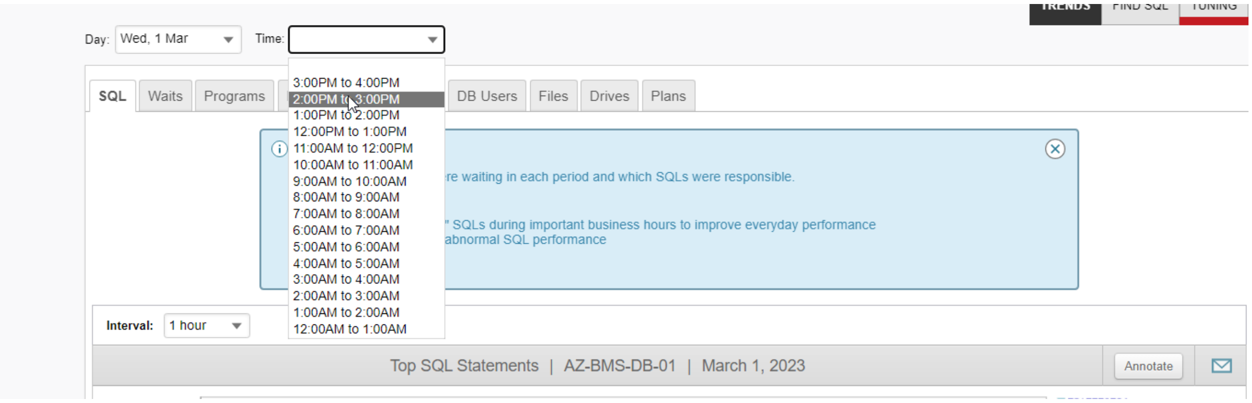
Day: All Days

Top SQL Statements | AZ-BMS-DB-01

…SQL with the highest performance impact on your end users and applications.

…ch is to first look for consistently big colored SQL slices, as well as abnormally big colored SQL slices, to determine which SQL to focus your attention on.

Then let’s filter the range of time when the performance issue occurred:



Day: Wed, 1 Mar Time:

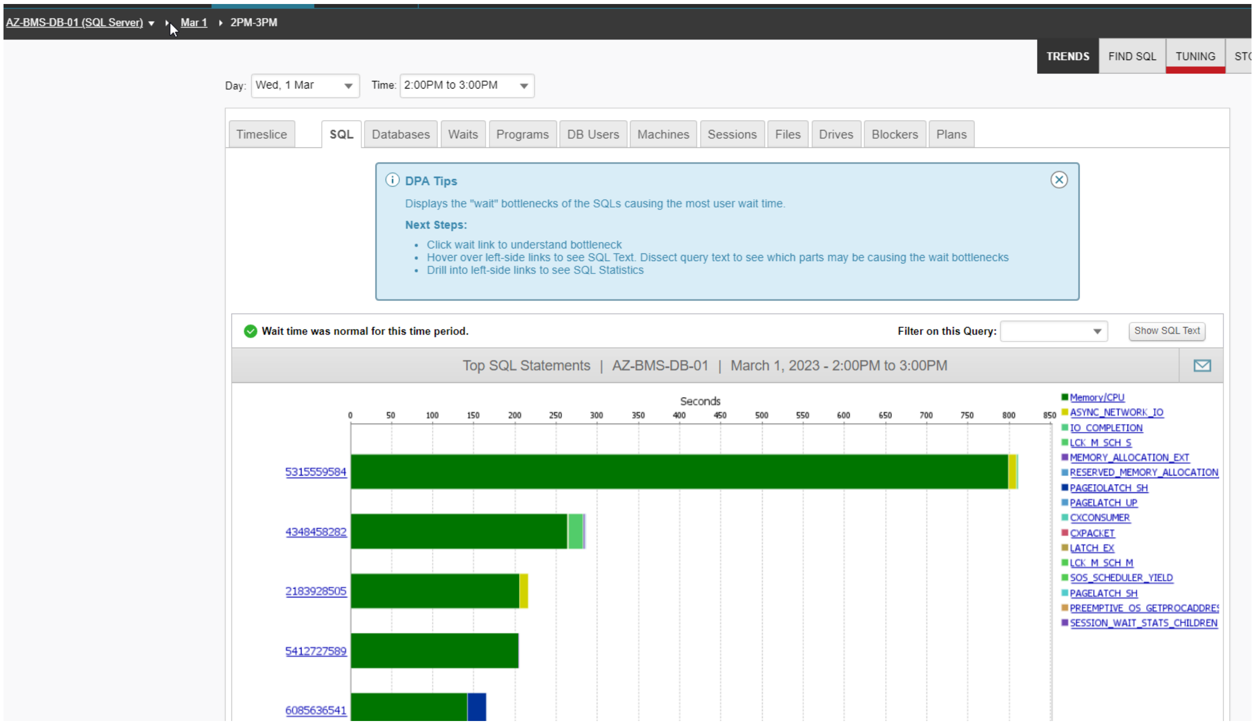
Interval: 1 hour

…re waiting in each period and which SQLs were responsible.

…" SQLs during important business hours to improve everyday performance
…abnormal SQL performance

Top SQL Statements | AZ-BMS-DB-01 | March 1, 2023

Now we have the TOP SQL statements by duration in that range of time:



AZ-BMS-DB-01 (SQL Server) › Mar 1 › 2PM-3PM

Day: Wed, 1 Mar Time: 2:00PM to 3:00PM

DPA Tips

Displays the "wait" bottlenecks of the SQLs causing the most user wait time.

Next Steps:

- Click wait link to understand bottleneck
- Hover over left-side links to see SQL Text. Dissect query text to see which parts may be causing the wait bottlenecks
- Drill into left-side links to see SQL Statistics

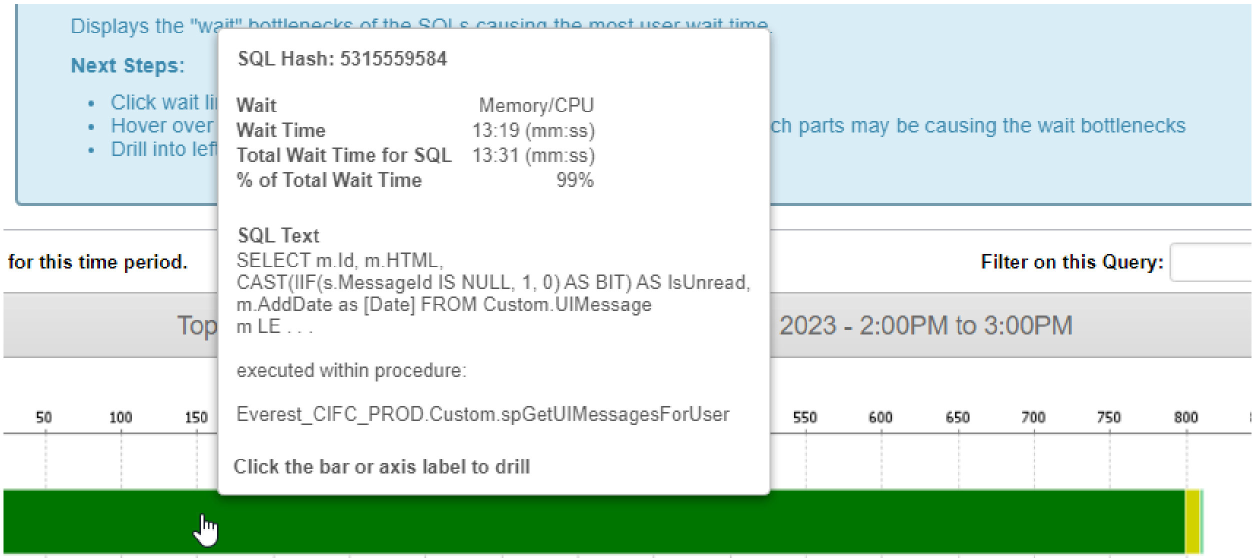
Wait time was normal for this time period.

Filter on this Query:

Top SQL Statements | AZ-BMS-DB-01 | March 1, 2023 - 2:00PM to 3:00PM

The color of the bars indicates the type of wait present in that query specified in the right side.

6. If we hover over one of the bars it will show us more information:



SQL Hash: 5315559584

Wait	Memory/CPU
Wait Time	13:19 (mm:ss)
Total Wait Time for SQL	13:31 (mm:ss)
% of Total Wait Time	99%

SQL Text
SELECT m.Id, m.HTML, CAST(IIF(s.MessageId IS NULL, 1, 0) AS BIT) AS IsUnread, m.AddDate as [Date] FROM Custom.UIMessage m LE . . .

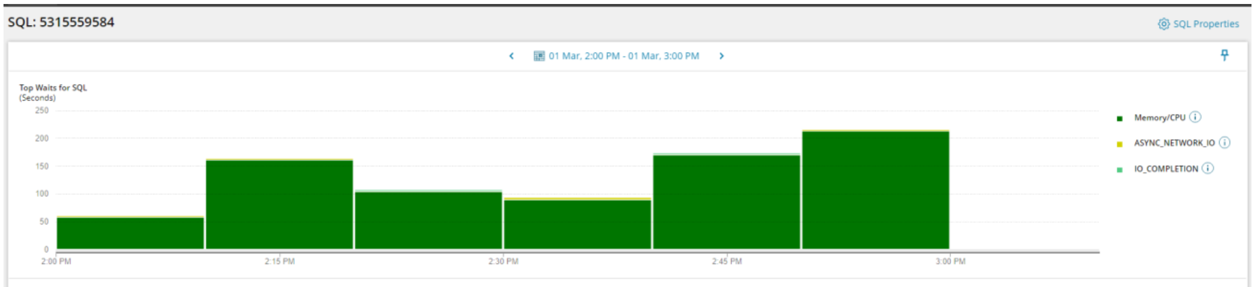
executed within procedure:

Everest_CIFC_PROD.Custom.spGetUIMessagesForUser

Click the bar or axis label to drill

It shows what type of wait, how much time it took and the query itself.

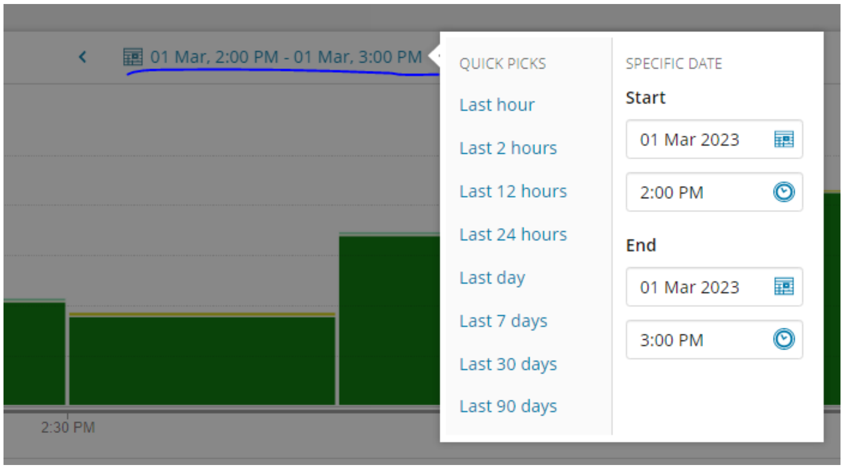
7. Now let's click in that bar(query) directly to see more details:



SQL: 5315559584

Here you will visualize how long it took and what type of wait along the hour was used.

Now notice that in the top part we have an option to filter it out in a more specific date:



Start

01 Mar 2023

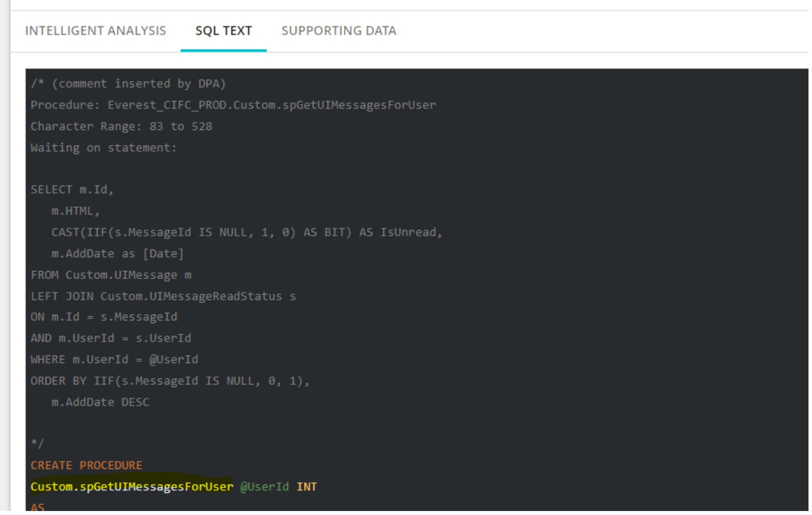
2:00 PM

End

01 Mar 2023

3:00 PM

8. If we want to see the SQL text, we just go ahead and hit the button SQL TEXT:



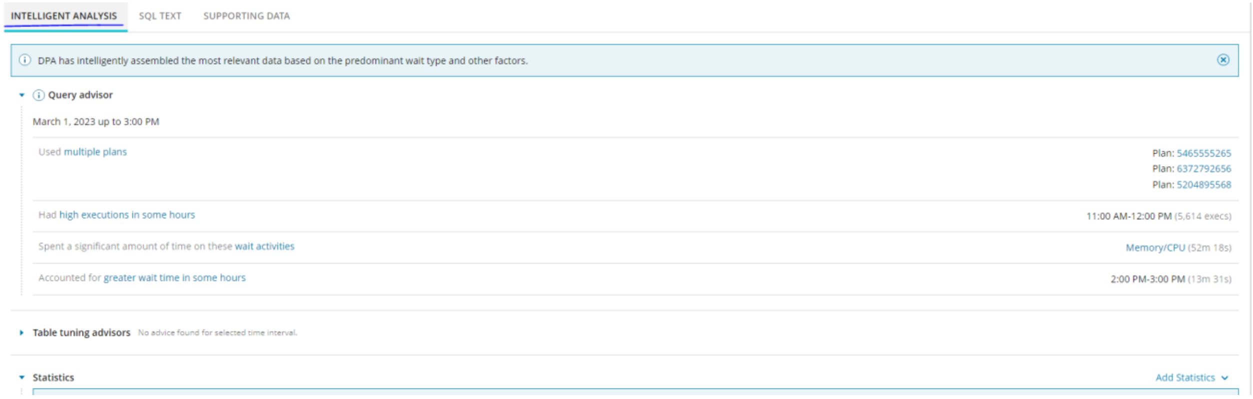
```
/* (comment inserted by DPA)
Procedure: Everest_CIFC_PROD.Custom.spGetUIMessagesForUser
Character Range: 83 to 528
Waiting on statement:

SELECT m.Id,
   m.HTML,
   CAST(IIF(s.MessageId IS NULL, 1, 0) AS BIT) AS IsUnread,
   m.AddDate as [Date]
FROM Custom.UIMessage m
LEFT JOIN Custom.UIMessageReadStatus s
ON m.Id = s.MessageId
AND m.UserId = s.UserId
WHERE m.UserId = @UserId
ORDER BY IIF(s.MessageId IS NULL, 0, 1),
   m.AddDate DESC

*/
CREATE PROCEDURE
Custom.spGetUIMessagesForUser @UserId INT
AS
```

We will se which part exactly of the statement was the waiter inside the comments inserted by DPA, and below we see the entire name of the object, in this case it was a stored procedure.

9. Going back to the main tabs, if we scroll down, we will see some recommendations in the **“Intelligent analysis”** tab:



DPA has intelligently assembled the most relevant data based on the predominant wait type and other factors.

Query advisor

March 1, 2023 up to 3:00 PM

Used multiple plans — Plan: 5465555265, Plan: 6372792656, Plan: 5204895568

Had high executions in some hours — 11:00 AM-12:00 PM (5,614 execs)

Spent a significant amount of time on these wait activities — Memory/CPU (52m 18s)

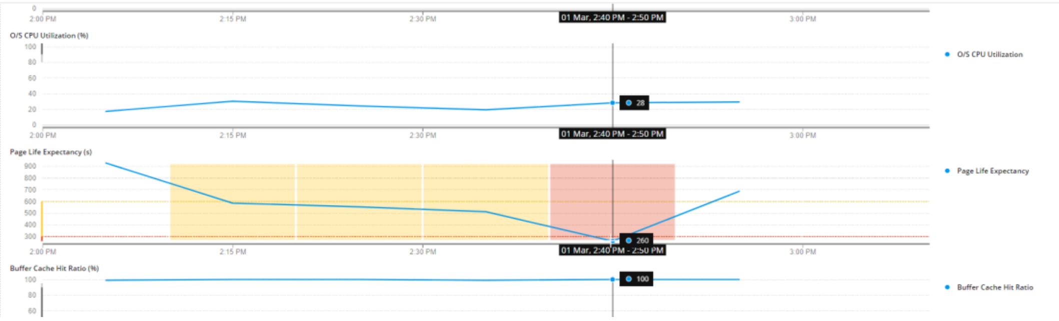
Accounted for greater wait time in some hours — 2:00 PM-3:00 PM (13m 31s)

Table tuning advisors No advice found for selected time interval.

Statistics

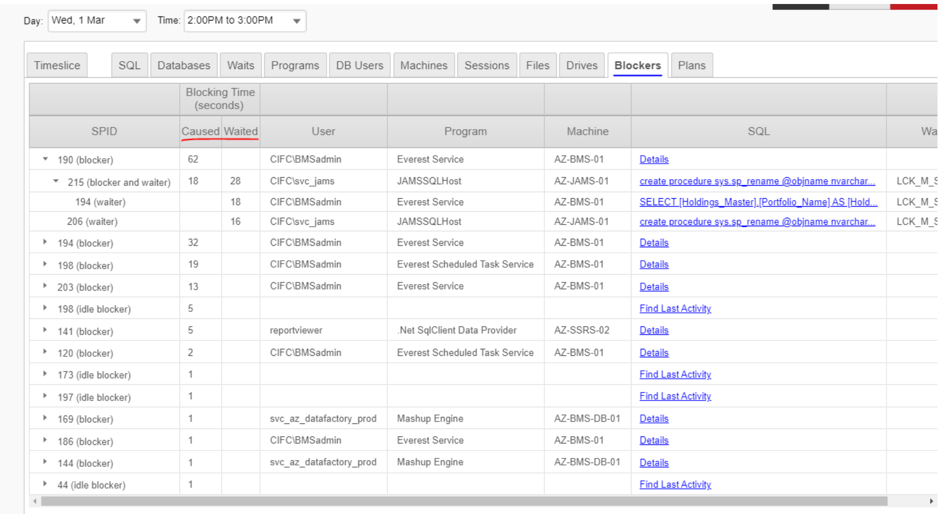
- In this case this query has 3 different plans, which can indicate recompilations of the same query (something for analyze in a deeper way). Also, we want to analyze each one of the query plans to see which is the best.
- Table tunning advisor has nothing for this query but for other we can find some recommendations as well

10. If we continue, we will find some charts about resource statistics in that range of time:



Here we see an alert regarding Page Life Expectancy at 2:40 the value was of 200. It may indicate a memory pressure that could be caused for expensive queries or disk IO stress.

11. If we go back to the top SQL statements (Step 5) now we can explore the different option in the tabs. Let’s start with **“Blockers”** tab:

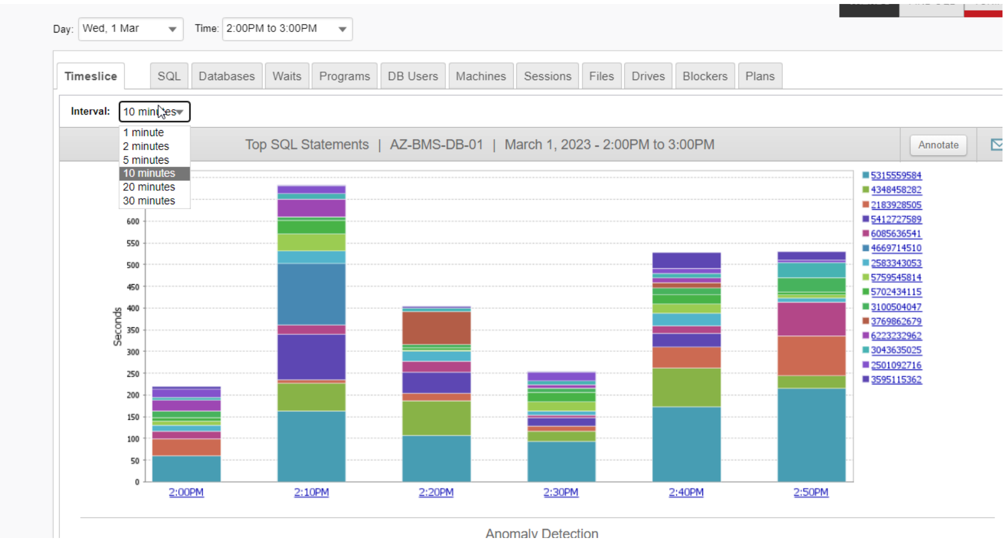


Day: Wed, 1 Mar Time: 2:00PM to 3:00PM

	Blocking Time (seconds)						
SPID	Caused	Waited	User	Program	Machine	SQL	Wa
190 (blocker)	62		CIFC\BMSadmin	Everest Service	AZ-BMS-01	Details	
215 (blocker and waiter)	18	28	CIFC\svc_jams	JAMSSQLHost	AZ-JAMS-01	create procedure sys.sp_rename @objname nvarchar...	LCK_M_S
194 (waiter)		18	CIFC\BMSadmin	Everest Service	AZ-BMS-01	SELECT [Holdings_Master].[Portfolio_Name] AS [Hold...	LCK_M_S
206 (waiter)		16	CIFC\svc_jams	JAMSSQLHost	AZ-JAMS-01	create procedure sys.sp_rename @objname nvarchar...	LCK_M_S
194 (blocker)	32		CIFC\BMSadmin	Everest Service	AZ-BMS-01	Details	
198 (blocker)	19		CIFC\BMSadmin	Everest Scheduled Task Service	AZ-BMS-01	Details	
203 (blocker)	13		CIFC\BMSadmin	Everest Service	AZ-BMS-01	Details	
198 (idle blocker)	5					Find Last Activity	
141 (blocker)	5		reportviewer	.Net SqlClient Data Provider	AZ-SSRS-02	Details	
120 (blocker)	2		CIFC\BMSadmin	Everest Scheduled Task Service	AZ-BMS-01	Details	
173 (idle blocker)	1					Find Last Activity	
197 (idle blocker)	1					Find Last Activity	
169 (blocker)	1		svc_az_datafactory_prod	Mashup Engine	AZ-BMS-DB-01	Details	
186 (blocker)	1		CIFC\BMSadmin	Everest Service	AZ-BMS-01	Details	
144 (blocker)	1		svc_az_datafactory_prod	Mashup Engine	AZ-BMS-DB-01	Details	
44 (idle blocker)	1					Find Last Activity	

This allows us to see which sql statements were blocking or blocked processes in that time. If we click in the details hyperlink it will show us the whole sql statement. Also notice the “blocking time” that it spent as blocker and blocking query.

12. Now if we go through **“TimeSlice”** tab, we can choose the time more granularly:

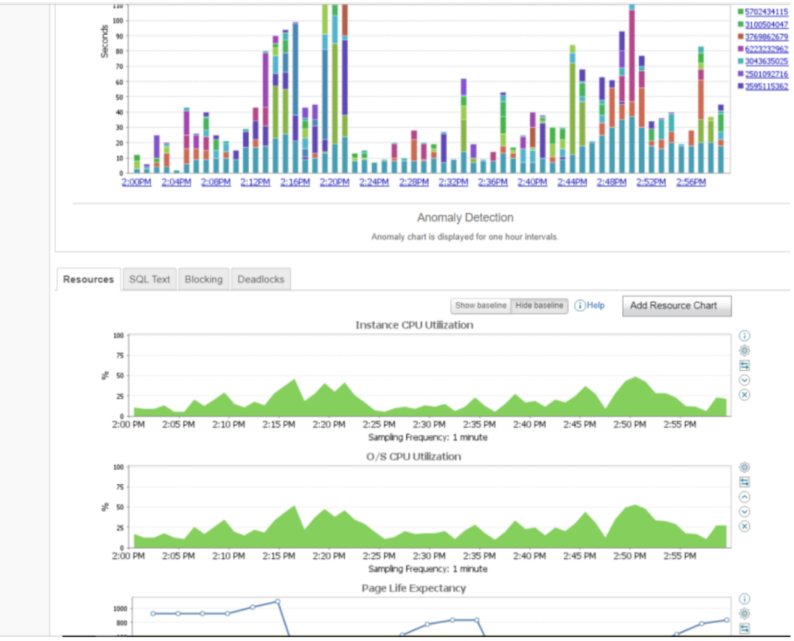


Day: Wed, 1 Mar Time: 2:00PM to 3:00PM

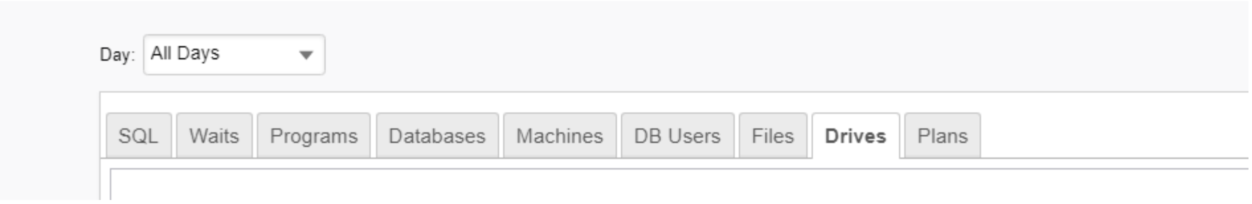
Interval: 10 minutes

This is useful if we want to see what happened in a range of 5 minutes for example.

13. Once I choose my filter, I can see the resource utilization in that period:



14. There are more tabs that I can use if necessary:



"Waits": List the top wait types that my queries are going through by time.

"Programs": List the applications that are being used in my queries by time. Example: Jams, .Net, Everest services, etc.

"Databases": List in order the databases that consumes the most resources.

"Machines": List the source servers from which my queries are being executed.

"DB users": List in order the logins that executed the queries by time.

"Files": List in order the usage of files .mdf, .ndf and .ldf of the databases by time.

"Plans": List in order the most expensive execution plans by time.