Foglight Skills 101 Episode 4

Mastering SQL Server Performance with Quest Foglight

Janis Griffin Senior Systems Consultant







Who Am I?

Current – 30+ Years in Oracle®, DB2®, ASE, SQL Server®, MySQL®, PostgreSQL

DBA and Developer

- Specialize in Performance Tuning
- Customers Common Question: How do I tune it?





Janis.Griffin@quest.com

Twitter® - @DoBoutAnything



Today's Focus

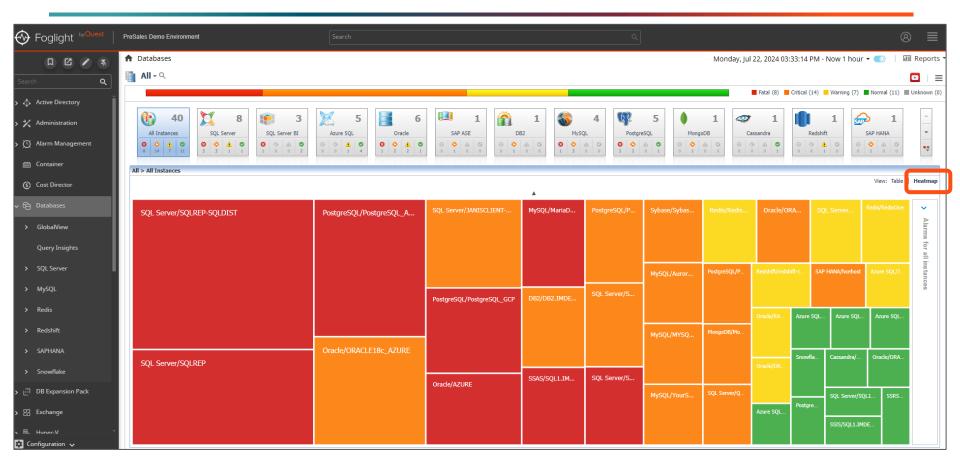
Utilize Query Insights

Monitor & evaluate adaptive baselines & change tracking

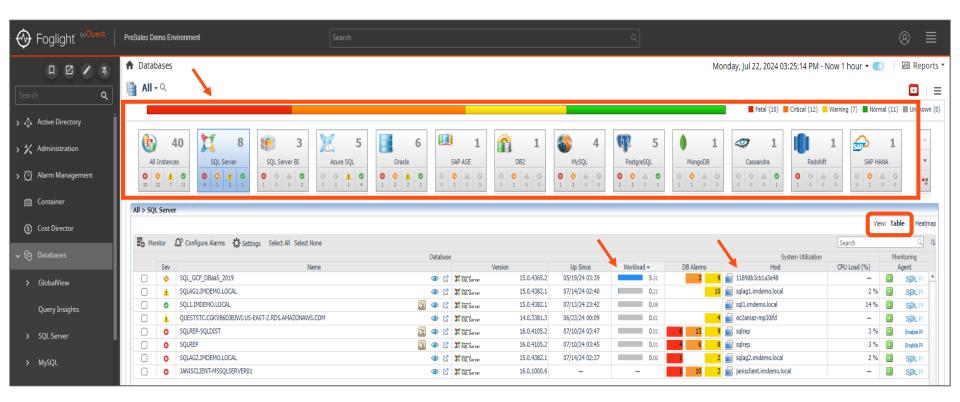
liest

- Including historical examination & decision making
- Review key metrics & query execution plans
- Use wait types to quickly identify bottlenecks
 - To gain insights on the best tuning approach
- Consider tuning with SQL Optimizer

Foglight Shows Company Level Database Pain Quest



Global Company or Database Specific Views



Ouest

Quick View of Workloads / Baselines

Databases							Mo	onday, Jul 22, 2024 0	3:30:14 PM - Now 1 hour 🔻		🖻 Rep
All - Q											
								Fatal (9)	Critical (14) Warning (6)	Normal (11)	Unknov
0 0 A 0 0	8 SQL Server 3 1 1 3 1 1		6 0racle 0 ◆ ▲ ● 0 1 2 2 1 0 ↓ 1	A 0	8 4 8 8	4 hysQL 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0	5 MongoDB	Cassandra © 0 0 0 1 0 0 0 1	8 4 4 9 8 4		
A 1										View: Table	
Monitor 🏛 Configure Ala	arms 🔅 Settings Select	All Select None	Dat	base					Search System Utilization	Monito	Q
Sev		Name	Data	ibase	Version	Up Since	Workload	DB Alarms 👻	Host	Age	
SQLREP-SQLDI	IST		🛃 👁 I	Z 🔀 SQL Server	16.0.4105		0.01	6 13 9 💼 s	qlrep		Enable Pl
3 SQLREP			3 4 1	SQL Server	16.0.4105	2 07/10/24 03:45	0.01	4 6 8 📄 s	qlrep		Enable P
SQLAG2.IMDEN	MO.LOCAL		٩	Z SQL Server	15.0.4382	1 07/14/24 02:37	0.00	1 2 💼 s	qlag2.imdemo.local		SØL P
JANISCLIENT-M	MSSOLSERVER01		۵	7 Marine Carner	16.0.1000	6 07/22/24 11:24	0.00	10 2 📻 ja	nisclient.imdemo.local		soi p
SQL_GCP_DBa	_			SQL Server			5.68		189db3cb1a3e48		SØL PI
	KVB6DJBJWI.US-EAST-2.RDS.	MAZONAWS.COM		SQL Server			0.00		c2amaz-mp30ifd		SØL PI
				SQL Server		1.1	0.00		glag1.imdemo.local		SØL P
SQL1.IMDEMO.	LOCAL		2 (D)	SQL Server	15.0.4382	1 07/13/24 23:42	0.03	iii s	gl1.imdemo.local		SØL PI
A SQLAGI.IMDEM SQL1.IMDEMO. SQL_GCP_DBaa5_2019				3 K SQL Server		1.1	0.00		Jag1.imdemo.local	_	
General	HADR	System Utilization		5	torage	Workload					
DB type: 🔀 Sql Server	Log shipping:	CPU Load (%)		n/a 🗵	Total Used Space Percent				<u>^</u>	1	1
Version: 15.0.4365.2 Up since: 05/19/24 03:39	Mirroring: Replication:	•		100		\sim	\sim	\sim			6 se
Up since: 05/19/24 03:39 Host: 📄 1189db3cb1a3		 Memory (%) 		n/a 🛛	File Groups: 7					-	4 ⁰ nds
OS Cluster: 💿	Cluster Aware:	0.00		100	Files:				•	+	25
Alarms	Performance				17	15:30 15:35 15:40	15:45 15:50 1	5:55 16:00 16:05	16:10 16:15 16:20	16:25	+0
		Disk (% Busy)		n/a 🖉	15 GB						
	Active sessions:	1 0.00		100	24 %			🛆 🗔 🖉			

Quest

Get Query Insights over All or Selected Domains Quest

Foglight by Ouest	PreSales Demo Environment	Search				Q		
	A Query Insights							Monday, Jul 22, 2024 03:44:15 PM - Now 1 hour 🔻 🥑
Search Q	Query Insights * Display: Top 50 Queries + by Im	npact						Statement Details
Active Directory Administration	SQL Server Impact Response Time Elapse	ed Time 🗸 Executi	ons 🗸	Search text Q	Apply 🔿 Reset			Name Query ID UPDATE [dbo].[BAN 0x0bd0a4d0d5e9
> (Alarm Management	Query i ■ Impact on target	Impact 🕴	Elapsed Time 🧯	Executions	Response Time 🧃	Domain i	ि Target ा	K] SET [BANK_NAM 4049 SQL Text Copy
Container	ରି UPDATE [dbo].[BANK] SET [BANK_NAME] = LTRIM(RTR	RIM(BAN 46.53 %	1.91 hr	40	2.87 min	SQL Server	SQL_GCP_D	UPDATE [dbo].[BANK] SET [BANK_NAME] =
③ Cost Director	बि SELECT LAST_NAME, STREET INTO #tmp_sales FROM E	EMPLOYE 18.48 %	4.38 min	1in: 28.5	4 min	SQL Server	SQLAG1.IM	LTRIM(RTRIM(BANK_NAME)) FROM BANK WHERE [BANK_CODE] = '** Removed by Foglight **'
ר 🗘 Databases	ରି OPEN getsumamount;	14.39 %	3.39 min		5 min	SQL Server	SQLAG1.IM	rogiight **
> GlobalView	ត្តិ ROLLBACK TRAN;	9.01 %	5.7 min	lax: 28.9		SQL Server	SQLAG1.IM	Investigate
Query Insights	🛱 INSERT IN TO #GETSUMAMOUNT(CUSTOMER_NAME,	ORDER_ 8.76 %	4.38 min			SQL Server	SQLAG1.IM	Elapsed Time
> SQL Server	बि UPDATE [BB] set [B] = @1	8.43 %	20.85 min	242	5.1 sec	SQL Server	SQL_GCP_D	2.0M
> MySQL	बि UPDATE [AA] set [A] = @1	7.19 %	17.76 min	176	6 sec	SQL Server	SQL_GCP_D	الله الله الله الله الله الله الله الله
> Redis	鼠 INSERT INTO [dbo].[ORDL1] SELECT * FROM [dbo].[OF	RDER_LII 6.23 %	18.05 min	71	15.9 sec	SQL Server	SQL_GCP_D	1.0M • Elapsed Time: 28.55 min
> Redshift	ର INSERT INTO [dbo].[ORDL2] SELECT * FROM [dbo].[OF	RDER_LII 5.7 %	19.69 min	57	20.9 sec	SQL Server	SQL_GCP_D	
> SAPHANA	INSERT INTO [dbo].[ORDL3] SELECT * FROM [dbo].[OP	RDER_LII 5.66 %	19.32 min	57	20.5 sec	SQL Server	SQL_GCP_D	0 15:45 16:00 16:15
> Snowflake	🗟 DELETE FROM [dbo].[ORDL1]	5.4 %	12.64 min	70	11.1 sec	SQL Server	SQL_GCP_D	Executions i
> ⊡ DB Expansion Pack	鼠 FETCH getsumamount INTO @CUSTNME, @ORDERNB	R, @PRC 5.35 %	2.56 min	98	1.5 sec	SQL Server	SQLAG1.IM	0 July 22, 2024 3:45 PM 15: • Eventitions: 10 00 16:15
> 82 Exchange	뤅 DELETE FROM [dbo].[ORDL3]	5.09 %	11.85 min	56	12.6 sec	SQL Server	SQL_GCP_D	U July 22, 2024 3:45 PM 15: • Executions: 10 00 16:15

Query Insights – Investigate SQL PI

🚱 Foglight ^{: by} Quest	PreSales Demo Environment						⊗ ≡
	✿ Query Insights > SQL PI				Monday, Jul 22, 2024 03:59:08 PM - No	ow 1 hour 👻 🥑) 🛛 🖉 Reports 🔻
Search Q	♦SQL_GCP_DBaaS_2019 - X Overview	SQL PI Memory - Activity - Databases - Services - HADR - Logs - Configuration	User-defined -				¢ ¢ ≡
> 🔶 Active Directory	Workload CPU I/O Memory Network	Lock Latch Log CLR Remote Provider XTP Other					Powered by SOL PI
> 🗙 Administration		story Advanced Analytics					View as PDF
> 🕚 Alarm Management	SQL Statements QUPDATE [dbo].[BANK] SET [BANK.	mension Filter: Instance View + SQL Statements + UPDATE [dbo].[BANK]					Top Wait Events
Container		E [dbo].[BANK] SET [BANK_NAME] = LTRIM(RTRIM(BANK_NAME)) FROM BANK WHERE [BANK_CODE] = "** Remov It ^{#**}	^{red} Top Wait Events				:
③ Cost Director	INSERT INTO [dbo].[ORDL1] SELEC		Resource: All Wa	ait Events 💉			
Ŭ.	INSERT INTO [dbo].[ORDL3] SELEC INSERT INTO [dbo].[ORDL2] SELEC		Category	Event Name	% of Total		Wait Time 👻 🗄
🗸 🔁 Databases	DELETE FROM [dbo].[ORDL3]		Lock Wait	LCK_M_U		63.14	9,443.89
	ELETE FROM [dbo].[ORDL2]	16:00 16:02 16:04 16:06 16:08 16:10 16:12 18:14 18:16 16:18 16:20 16:22 16:24 16:26 18:28 18:30 1	6:3 CPU Wait	SOS_SCHEDULER_YIELD		3.02	452.44
> GlobalView	DELETE FROM [dbo].[ORDL1]	Overview Blocking History Activity Highlights	Memory Wait	MEMORY_ALLOCATION_EXT		1.52	227.84
Query Insights	Sol SELECT [customer_id], Sum(amount Sol SELECT TOP (@RowLimit) CONVERT	Overview blocking history Activity highlights	Log Wait	LOGMGR_FLUSH		0.61	90.55
() <u>-</u>		tive Time	Memory Wait	RESERVED_MEMORY_ALLOCATIO	N_EXT	0.39	58.87
> SQL Server	E SOL FETCH NEXT FROM d_cursor INTO (Other Wait	CXPACKET	-	0.10	14.79
	Select Distinct A2.request_session_i		Network Wait	ASYNC_NETWORK_IO		0.04	6.43
> MySQL	SELECT TOP (@P0) sql_handle, stat		The event where	ASTRO_RETWORK(_10		0.01	0.15
	FETCH NEXT FROM d_cursor INTO (SOL Select Top 1 @DBID = B.[dbid], @Q	16:00 16:02 16:04 16:06 16:08 16:10 16:12 16:14 16:16 16:18 16:20 16:22 16:24 16:26 16:28 16:30 16		9.40 18.47 18.44 18.48 18.40 18.50 18.57 18.	64 18-59 18-59		
> Redis	insert into #temp_trace select top 1		.32 10.34 10.30 10.38 10	3.40 10.42 10.44 10.40 10.48 10.50 10.52 105	54 10.50 10.58		
> Redshift	Select @ord_id,@dte_ord,@stat_des	forkload related Metrics					
		🛛 🛪 Select Metric 🛛 D View SQL Text 🚑 Analyze Plan 🗄 Tune SQL 😳 Compare				Search	् 🕫
> SAPHANA	Sol select @ord_id,@dte_ord,@stat_des	Active Time	14	Resource forkload	Total		14,958.01
	- SUL	Average SQL Response Time		forkload			14,958.01
> Snowflake		Batches Rate		forkload			11.66
-		CPU Usage		PU			4,936.06
DB Expansion Pack	Databases	Deadlocks	Lo	ock			19.33
> 🔀 Exchange		Executions	W	forkload			1,564.00
Co exchange		.ogical Reads	1/				1,387,327,112.00
、 BL Humer-V ×	Client Machines	Nait Time Percent	W	/orkload			67.00
🔁 Configuration 🗸	- SI CONCERCINOS						-

SQL PI – Analyze Plan



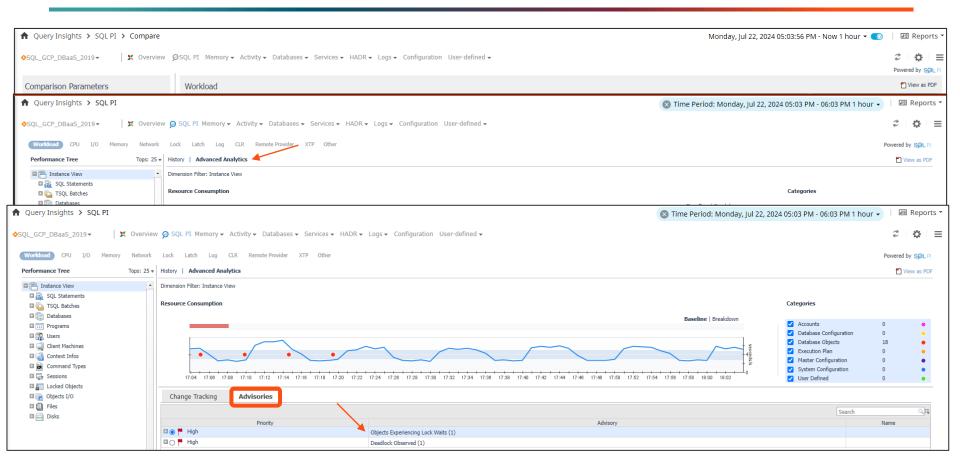
\odot	· Foglight ^{- t}	oyQuest	PreSal	les Demo En	vironment				Search						٩					8	≡
		/ *	A 0	Query Insig	ghts > SQ)L PI										Mond	day, Jul 22, 2024	03:59:08 PM - Nov	v 1 hour 👻 🧐	🔊 Rep	ports 🔻
Sear		٩	♦ SQI	GCP_DBa	aS_2019▼	X 0	verview 😡 SQL	L PI Me	emory 🗸 Activi	ity 👻 Da	atabases 👻 Service	s ▼ HADR ▼	Logs - Configuration	User-defined 🗸						$z \mid \phi$	
> 	Active Directory	Í		Vorkload	CPU I/O	Memory Ne	twork Lock	Latch	Log CLR	Remote	Provider XTP Oth	ier								Powered by SG	∂L PI
> X	Administration			rformance T		Tops: 3														View as	s PDF
> ©) Alarm Managen	nent		Instance	tatements	ANK] SET [BANK_			-	Statement	ts ▶ UPDATE [dbo].[BAI	NK]								Top Wait Ev	vents
) Container	🔒 Qu	ery Insig	ghts > So	QL PI											Monday	/, Jul 22, 2024 04	:34:09 PM - Now	1 hour 👻 🧑	💷 Rep	orts •
3	Cost Director	Execution I	Plan																	Powered by	□> y S <mark>ØL</mark> F
	Databases GlobalView	3/3/24 Date	I, 6:23 PI	1 👻 Cach Type	e Collected	0x06000600 Plan Handle	96ff801d10d05	02a100	00000010000	•								Compare Plans	Generate Plan	top SSI	en in MS
	Query Insights	Statem	ent						Plan Analysis												
>	SQL Server	UPDATE BANK W	[dbo].[BAI HERE [BAN	NK] SET [BAN NK_CODE] = '	IK_NAME] = L ** Removed b	TRIM(RTRIM(BAI by Foglight **'	IK_NAME)) FROM		Total cost: 0.0	165715 '	Total I/O cost: 0.0162	2500 Total Cl	PU cost: 0.0003215								
	- MySQL								Plan Detai	ils	Operator Analysis	Objec	t Analysis								
Ĺ	MySQL										Name		Database		Туре			Associated O	Search		्रा
>	Redis								dbo.BANK		Name		sales	Table	туре		Table Update, RID Lo		perators		
>	Redshift								dbo.BANK.PK_	BANK			sales	Index (NonClustered)			Index Seek				
>	SAPHANA																				
>	Snowflake	4					Þ														
÷.	DB Expansion Pa	ack																			-
× 83	Exchange	SQL Tex	t																		2
、 L	Hyper-V	SET FROM	BANK	ME] = LTRIM	I (RTRIM (BA Removed by F	NNK_NAME))															

SQL PI - Compare



Query Insignts > SQL PI		Monday, Jul 22, 2024	04:54:35 PM - Now 1 hour 🔻 🗾) 💷 Repor
♦SQL_GCP_DBaaS_2019▼ X Overv	view 🗩 SQL PI Memory 🗸 Activity 🗸 Databases 🗸 Services 🗸 HADR 🖌 Logs 🗸 Configuration User-defined 🗸			2 ¢
Workload CPU I/O Memory Networ	ork Lock Latch Log CLR Remote Provider XTP Other			Powered by SOL
Performance Tree Tops: 25 v	History Advanced Analytics			🗂 View as PD
■ 📇 Instance View	Dimension Filter: Instance View + SQL Statements + UPDATE [dbo].[BANK]			
SQL Statements				
💷 🔂 UPDATE [dbo].[BANK] SET [BANK_N	Resource Consumption			Top Wait Event
	PDATE [dbo].[BANK] SET [BANK_NAME] = LTRIM(RTRIM(BANK_NAME)) FROM BANK WHERE [BANK_CODE] = '** Removed by		Resource Breakdown	
DPDATE [AA] set [A] = @1	glight ***	т4		
🖬 🛐 INSERT INTO [dbo].[ORDL1] SELEC		- s	Lock Wait	100.00%
🖾 🙀 INSERT INTO [dbo].[ORDL3] SELEC		29		
🖬 🚮 INSERT INTO [dbo].[ORDL2] SELEC		- S		
DELETE FROM [dbo].[ORDL3]	16:56 16:59 17:00 17:02 17:04 17:08 17:08 17:10 17:12 17:14 17:16 17:18 17:20 17:22 17:24 17:26 17:28 17:30 17:32 17:34 17:38 17:39 17:40	17.42 17.44 17.48 17.49 17.50 17.50 17.54		
DELETE FROM [dbo].[ORDL1]	10:00 10:00 1/302 שנים שנים שנים שנים אינים 12:00 10:00 ועבים שנים סובים	17342 17344 17340 17346 17300 17302 17304		
DELETE FROM [dbo].[ORDL2]				
SELECT [customer_id], Sum(amount)	Overview Blocking History Activity Highlights			
Insert Into #ResourceDB Select Dist				
SELECT TOP (@RowLimit) CONVERT	Active Time			
Select Distinct A2.request_session_i			Sum of all the active waits and cpu usa	
sign insert into #temp_trace select top 1		2%	session total activity within the current i	interval.
Select Top 1 @DBID = B.[dbid], @C		- Cond		
sign insert into #temp_trace select top 1		5 m		
🗉 👸 select top 100 DatabaseName, Filen	16:56 16:58 17:00 17:02 17:04 17:06 17:08 17:10 17:12 17:14 17:16 17:18 17:20 17:22 17:24 17:26 17:28 17:30 17:32 17:34 17:38 17:38 17:40 1	7:42 17:44 17:46 17:48 17:50 17:52 17:54		
🖬 🙀 (@P0 bigint,@P1 int,@P2 int,@P3 m	Workload related Metrics			
🖬 🙀 xp_readerrorlog				
sql xp_instance_regread	🐵• Select Metric 🛛 View SQL Text 🚠 Analyze Plan 🗮 Tune SQL 🛃 Compare		Search	् म्
🗉 🙀 SELECT top (@P0) t2.spid, t2.login_	Metric 🔺	Resource	Total	
🖾 🙀 Select @DBName As DBName , DB_	Average SQL Response Time	Workload		175.22
DROP TABLE #QS_sysfiles	CPU Usage	CPU		< 0.01
select create_time from sys.dm_xe_	Executions	Workload		39.00
🖬 🖕 TSQL Batches	Lock Update	Lock		6,834.83
Databases	Lock Wait	Lock		6,834.83
Programs	Logical Reads	I/O		135.00
🖽 🞇 Users	Row count	Workload		13.00
Client Machines	Wait Time Percent	Workload	-	100.00
Context Infos				

SQL PI – Advanced Analytics



Advisories

Name Objects Experiencing Lock Waits Description Object lock contention was observed during the analyzed period, taking 62.22% of response time. Figure 1 illustrates the total wait for lock for the monitored SOL Server instance as compared to the overall active time of the instance 16:10 16:15 16:20 18.25 16:30 16.35 16-40 16.45 16.50 16.55 17:00 17:05

Table 1 shows the list of top objects that experienced Lock Contention.

Locked Objects	Lock Wait Time (seconds)	% of Overall Response Time 😔
sales.dbo.BANK.PK_BANK [INDEX]	7,026.65	75.43
sales.dbo.BB [TABLE]	1,251.99	13.44
sales.dbo.AA [TABLE]	1,104.32	11.85

Click on the Locked Object name to find out more about the identity of the blockers of that object, the amount of sessions involved, and the identity of the locked sessions.

Key Metrics

Objects Experiencing Lock Waits

Table 2 displays the trend of related key metrics, recorded during the analyzed period.

Metric	Average	Min	Max :=
Blocked Lock Requests	16.91	6.00	33.00
Table Lock Escalations	10.70	0.00	40.00
Lock Timeouts	23.78	6.00	56.00
Average Lock Wait Time (ms)	12,248,568,325.27	12,244,140,552.00	12,253,095,061.00

Recommentation

Review the application logic and consider techniques to reduce contention caused by concurrent updates to data. Some of the main causes of table locking include the following:

- · Contention for a specific row in the database. The application design may require that many processes update or lock the same row in the database. For example, when primary keys are generated using a sequence table.
- · Table locks caused by non-indexed foreign keys, or ineffective / missing indexes, or out-of-date statistics. When a non-indexed foreign key is updated, the parent table could be subjected to a table lock until the transaction is complete. If the indexes defined are ineffective or missing, the SQL optimizer chooses non-optimal query plans. Nonoptimal plans can also be chosen when a qualifying index exists, if statistics are out of date.
- Long-running transactions. Individual transact-SQL statements may perform well, but maintaining transactions for extended operations can dramatically increase the amount of contention within a database. Investigate transaction control logic to determine where operations can be broken up into smaller, component transactions. In addition, review the use of transaction isolation levels in application code to determine whether a less restrictive isolation level can be used.

Next steps

- Review Blocking History for a detailed report of all sessions involved in the blocking scenario during the period described
- Review Activity Highlights to find out top contributors to the lock wait.



Blocking History

Databases > Overview > SQL PI							🗴 Time Period: Wednesday, Jul 24, 2024 06:08 PM - 07:0	08 PM 1 hour 👻 🗌 💷 Repo
QL_GCP_DBaaS_2019▼ 🛛 💢 Overview 🨡	SQL PI Memory - Activity - D	Databases → Services → HAD	R ▼ Logs ▼ Configu	uration User-defined -				\$ \$
Workdoad CPU I/O Memory Network Low	ock Latch Log CLR Remote	e Provider XTP Other						Powered by SOL
Performance Tree Tops: 25	5 - History Advanced Analytics							🗂 View as P
E 🔚 Instance View	Dimension Filter: Instance View							
SQL Statements								
🖾 🖕 TSQL Batches	Resource Consumption							Top Wait Even
Catabases							Baseline Breakdown Resource Break	down
Programs	There Is No Data To Display							These To Ma Date To
								There Is No Data To Display
Client Machines								
Context Infos								/
Command Types Sessions								
Locked Objects								
a sales.dbo.BANK.PK_BANK [INDEX]	Overview Blocking H	listory Activity Highlights						
sales.dbo.BB [TABLE]	Diocking	nscory Activity Highlights						
ales.dbo.AA [TABLE]	View Full Text							Search Q
Objects I/O					Lock			
E E Files	Event Start	SPID	Blocked By	Resource	Resource State Type	s Duration	Program	User 🗡
Disks	7/24/24, 6:39 PM 84 [2]	024-07-24 18:36:02.61]				ng 173.22	SQLAgent - TSQL JobStep (Job 0xCC31233C4311984AA5D2BF9F0D924748	: Step 1) sqlserver 🦕 BEGIN TRAN
	O 7/24/24, 6:39 PM 67 [2	024-07-24 18:39:01.593] 84 [2024-	07-24 18:36:02.61] sale	es.dbo.BANK.PK_BANK [INDEX]	KEY Block	ed 173.22	SQLAgent - TSQL JobStep (Job 0x2BE2C65EE58F9A4FB5D10B3B9C206BEF	: Step 1) sqlserver 🚮 UPDATE [dbo]
	7/24/24, 6:21 PM 85 [2]	024-07-24 18:18:02.443]			Block	ng 173.03	SQLAgent - TSQL JobStep (Job 0xCC31233C4311984AA5D2BF9F0D924748	: Step 1) sqlserver 🦕 BEGIN TRAN
	O 7/24/24, 6:21 PM 69 [20	024-07-24 18:21:01.873] 85 [2024-	07-24 18:18:02.443] sale	es.dbo.BANK.PK_BANK [INDEX]	KEY Block	ed 173.03	SQLAgent - TSQL JobStep (Job 0x2BE2C65EE58F9A4FB5D10B3B9C206BEF	: Step 1) sqlserver 🚮 UPDATE [dbo]
	7/24/24, 7:06 PM 81 [20]	024-07-24 19:03:03.11]			Block	ng 172.77	SQLAgent - TSQL JobStep (Job 0xCC31233C4311984AA5D2BF9F0D924748	: Step 1) sqlserver 🦾 BEGIN TRAN
	O 7/24/24, 7:06 PM 74 [20	024-07-24 19:06:02.147] 81 [2024-	07-24 19:03:03.11] sale	es.dbo.BANK.PK_BANK [INDEX]	KEY Block	ed 172.77	SQLAgent - TSQL JobStep (Job 0x2BE2C65EE58F9A4FB5D10B3B9C206BEF	: Step 1) sqlserver 🚮 UPDATE [dbo
	■ 7/24/24, 6:42 PM 67 [2	024-07-24 18:39:01.593]			Block	ng 172.45	SQLAgent - TSQL JobStep (Job 0x2BE2C65EE58F9A4FB5D10B3B9C206BEF	: Step 1) sqlserver 🦕 BEGIN TRAN
		024-07-24 18:42:02.487] 67 [2024-	07-24 18:39:01.593] sale	es.dbo.BANK.PK_BANK [INDEX]	KEY Block	ed 172.45	SQLAgent - TSQL JobStep (Job 0x45028A985BFE674999EDC6294A79162C :	: Step 1) sqlserver 🚮 UPDATE [dbo
	C 7/24/24, 6:48 PM 81 [2	024-07-24 18:45:02.64]			Block		SQLAgent - TSQL JobStep (Job 0xCC31233C4311984AA5D2BF9F0D924748	odr
	O 7/24/24, 6:48 PM 78 [2	024-07-24 18:48:02.477] 81 [2024-	07-24 18:45:02.64] sale	es.dbo.BANK.PK_BANK [INDEX]	KEY Block	ed 172.37	SQLAgent - TSQL JobStep (Job 0x2BE2C65EE58F9A4FB5D10B3B9C206BEF	: Step 1) sqlserver 🚮 UPDATE [dbo]
	■ 7/24/24, 6:18 PM 69 [2	024-07-24 18:15:02.787]			Block	ng 172.36	SQLAgent - TSQL JobStep (Job 0x45028A985BFE674999EDC6294A79162C :	: Step 1) sqlserver 🦕 BEGIN TRAN
	O 7/24/24 6:18 PM 85 [2	024-07-24 18:18:02.443] 69 [2024-	07-24 18:15:02.787] sale	es.dbo.BANK.PK_BANK [INDEX]	KEY Block	ed 172.36	SQLAgent - TSQL JobStep (Job 0xCC31233C4311984AA5D2BF9F0D924748	: Step 1) sqlserver 🚮 UPDATE [dbo]
			-	-			1	a der
	□ 7/24/24, 6:12 PM 72 [2	024-07-24 18:09:02.81]			Block	ng 172.35	SQLAgent - TSQL JobStep (Job 0xCC31233C4311984AA5D2BF9F0D924748	: Step 1) sqlserver 🤄 BEGIN TRAN
	C 7/24/24, 6:12 PM 72 [2	024-07-24 18:09:02.81] 024-07-24 18:12:02.147] 72 [2024-	07-24 18:09:02.81] sale	es.dbo.BANK.PK_BANK [INDEX]			SQLAgent - TSQL JobStep (Job 0xCC31233C4311984AA5D2BF9F0D924748 SQLAgent - TSQL JobStep (Job 0x2BE2C65EE58F9A4FB5D10B3B9C206BEF	

Activity Highlights



atabases > Overview > SQL PI		🛞 Time Period: Wednesday, Jul 24, 2	2024 06:08 PM - 07:08 PM 1 hour	👻 💷 Rep
GCP DBaaS 2019▼ X Over	rview 🧔 SQL PI Memory 🗸 Activity 🗸 Databases 🗸 Services 👻 HADR 👻 Logs 👻 Configuration User-defined 👻			\$ \$
orkload CPU I/O Memory Networ	rk Lock Latch Log CLR Remote Provider XTP Other			Powered by Sp
formance Tree	Tops: 25 • History Advanced Analytics			🔁 View as
Page 1 Instance View	Dimension Filter: Instance View			
SQL Statements				
🛛 🦕 TSQL Batches	Resource Consumption			Top Wait E
Databases		Baseline Breakdown	Resource Breakdown	
Programs		+6	Lock Upo	late 49.75%
I 🙀 Users I 🗐 Client Machines		+ se 400	Non-Loc	
Context Infos		+ onds	- Non-Edd	CAC 30.2576
Command Types				
Sessions	18:09 18:12 18:15 18:18 18:21 18:24 18:27 18:30 18:33 18:36 18:39 18:42 18:45 18:48 18:	51 18:54 18:57 19:00 19:03 19:06		
Locked Objects	Overview Blocking History Activity Highlights The Instance consumed 5.762 seconds waiting for Lock. 49.75% of its db-time spent waiting for Lock.	ock Update		C
-	The Instance consumed 5,762 seconds waiting for Lock. 49.75% of its db-time spent waiting for Lock			
-	The Instance consumed 5,762 seconds waiting for Lock. 49.75% of its db-time spent waiting for Loc Highlights			Wait Event
-	The Instance consumed 5,762 seconds waiting for Lock. 49.75% of its db-time spent waiting for Lo Highlights SQL Statement UPDATE [dbo].[BANK] SET [BANK_NAME] = LTRJM(R is attributed to 60.24% of the instance Lock Wait		3,470.61 LCK_M_U	Wait Event
-	The Instance consumed 5,762 seconds waiting for Lock. 49.75% of its db-time spent waiting for Lo Highlights SQL Statement UPDATE [dbo].[BANK] SET [BANK_NAME] = LTRIM(R is attributed to 60.24% of the instance Lock Wait SQL Statement UPDATE [B6] set [B] = @1 is attributed to 2241% of the instance Lock Wait		3,470.61 LCK_M_U 1,291.37 LCK_M_U	Wait Event 100 ' 100 '
-	The Instance consumed 5,762 seconds waiting for Lock. 49.75% of its db-time spent waiting for Lo Highlights SQL Statement UPDATE [dbo].[BANK] SET [BANK_NAME] = LTRIM(R is attributed to 60.24% of the instance Lock Wait SQL Statement UPDATE [BB] = @1 is attributed to 22.41% of the instance Lock Wait SQL Statement UPDATE [AA] set [A] = @1 is attributed to 19.01% of the instance Lock Wait		3,470.61 LCK_M_U 1,291.37 LCK_M_U 1,095.24 LCK_M_U	Wait Event 100 100 100
-	The Instance consumed 5,762 seconds waiting for Lock. 49.75% of its db-time spent waiting for Lo Highlights SQL Statement UPDATE [dbb].[BANK] SET [BANK_NAME] = LTRIM(R is attributed to 60.24% of the instance Lock Wait SQL Statement UPDATE [db] = @1 is attributed to 22.41% of the instance Lock Wait SQL Statement UPDATE [AA] set [A] = @1 is attributed to 19.01% of the instance Lock Wait TSQL Batch BEGIN TRAN UPDATE [dbb].[BANK] SET [BANK_NAME is attributed to 39.27% of the instance Lock Wait		3,470.61 LGK_M_U 1,291.37 LGK_M_U 1,095.24 LGK_M_U 2,262.37 LGK_M_U	Wait Event 100 100 100 100
-	The Instance consumed 5,762 seconds waiting for Lock. 49.75% of its db-time spent waiting for Loc Highlights SQL Statement UPDATE [dbo].[BANK] SET [BANK_NAME] = LTRIM(R is attributed to 60.4% of the instance Lock Wait SQL Statement UPDATE [BB] = @1 is attributed to 22.41% of the instance Lock Wait SQL Statement UPDATE [AA] set [A] = @1 is attributed to 19.01% of the instance Lock Wait TSQL Batch BEGIN TRAN UPDATE [dbo].[BANK] SET [BANK_NAME is attributed to 39.27% of the instance Lock Wait TSQL Batch (@1 int)UPDATE [BB] set [B] = @1 is attributed to 22.41% of the instance Lock Wait		3,470.61 LCK_M_U 1,291.37 LCK_M_U 1,095.24 LCK_M_U 2,262.37 LCK_M_U 1,291.37 LCK_M_U	Wait Event 100 ¹ 100 ¹ 100 ¹ 100 ¹ 100 ¹
-	The Instance consumed 5,762 seconds waiting for Lock. 49.75% of its db-time spent waiting for Loc Highlights SQL Statement UPDATE [db0].[BANK] SET [BANK_NAME] = LTRIM(R., is attributed to 60.24% of the instance Lock Wait SQL Statement UPDATE [BB] set [B] = @1 is attributed to 22.41% of the instance Lock Wait SQL Statement UPDATE [db1].[BANK] SET [BANK_NAME is attributed to 39.27% of the instance Lock Wait TSQL Batch BEGIN TRAN UPDATE [db1].[BANK] SET [BANK_NAME is attributed to 22.41% of the instance Lock Wait TSQL Batch BEGIN TRAN UPDATE [db1].[BANK] SET [BANK_NAME is attributed to 20.97% of the instance Lock Wait		3,470,61 LCK_M_U 1,291.37 LCK_M_U 1,095.34 LCK_M_U 2,262.37 LCK_M_U 1,291.37 LCK_M_U 1,291.37 LCK_M_U	Wait Event 100 100 100 100 100 100
-	The Instance consumed 5,762 seconds waiting for Lock. 49.75% of its db-time spent waiting for Loc Highlights SQL Statement UPDATE [dbo].[BANK] SET [BANK_NAME] = LTRIM(R is attributed to 60.24% of the instance Lock Wait SQL Statement UPDATE [BB] set [B] = @1 is attributed to 22.41% of the instance Lock Wait SQL Statement UPDATE [AA] set [A] = @1 is attributed to 190.1% of the instance Lock Wait TSQL Batch BEGIN TRAN UPDATE [dbo].[BANK] SET [BANK_NAME is attributed to 39.27% of the instance Lock Wait TSQL Batch @61 in/JUPDATE [dbo].[BANK] SET [BANK_NAME is attributed to 20.97% of the instance Lock Wait TSQL Batch @61 in/JUPDATE [dbo].[BANK] SET [BANK_NAME is attributed to 20.97% of the instance Lock Wait		3,470,61 LCK_M_U 1,291.37 LCK_M_U 1,095.24 LCK_M_U 2,262.37 LCK_M_U 1,291.37 LCK_M_U 1,208.34 LCK_M_U 11,449.77 LCK_M_U	Wait Event 100 100 100 100 100 51
-	The Instance consumed 5,762 seconds waiting for Lock. 49.75% of its db-time spent waiting for Loc Highlights SQL Statement UPDATE [dbo].[BANK] SET [BANK_NAME] = LTRIM(R is attributed to 60.24% of the instance Lock Wait SQL Statement UPDATE [dB] = @1 is attributed to 22.41% of the instance Lock Wait SQL Statement UPDATE [AA] set [A] = @1 is attributed to 19.01% of the instance Lock Wait TSQL Batch BEGIN TRAN UPDATE [dbo].[BANK] SET [BANK_NAME is attributed to 39.27% of the instance Lock Wait TSQL Batch @6 int/UPDATE [dbo].[BANK] SET [BANK_NAME is attributed to 39.27% of the instance Lock Wait TSQL Batch @6 int/UPDATE [dbo].[BANK] SET [BANK_NAME is attributed to 20.97% of the instance Lock Wait TSQL Batch BEGIN TRAN UPDATE [dbo].[BANK] SET [BANK_NAME is attributed to 20.97% of the instance Lock Wait Database sales is attributed to 10.16.% of the instance Lock Wait Program SQLAgent - TSQL JobStep (Job 0x2BE2C65EE58F9A is attributed to 20.97% of the instance Lock Wait		3,470,61 LCK_M_U 1,291.37 LCK_M_U 1,095.34 LCK_M_U 2,262.37 LCK_M_U 1,291.37 LCK_M_U 1,208.24 LCK_M_U 11,449.77 LCK_M_U 1,208.24 LCK_M_U	Wait Event 100 4 100 4 100 4 100 4 100 4 100 4 100 4 100 4
-	The Instance consumed 5,762 seconds waiting for Lock. 49.75% of its db-time spent waiting for Loc Highlights SQL Statement UPDATE [dbo].[BANK] SET [BANK_NAME] = LTRIM(R is attributed to 60.24% of the instance Lock Wait SQL Statement UPDATE [db] = @1 is attributed to 22.41% of the instance Lock Wait SQL Statement UPDATE [db] = @1 is attributed to 19.01% of the instance Lock Wait TSQL Batch BEGIN TRAN UPDATE [dbo].[BANK] SET [BANK_NAME is attributed to 39.27% of the instance Lock Wait TSQL Batch @1 int]UPDATE [dbo].[BANK] SET [BANK_NAME is attributed to 29.37% of the instance Lock Wait TSQL Batch @61 int]UPDATE [dbo].[BANK] SET [BANK_NAME is attributed to 20.37% of the instance Lock Wait TSQL Batch BEGIN TRAN UPDATE [dbo].[BANK] SET [BANK_NAME is attributed to 20.37% of the instance Lock Wait Database sales is attributed to 100.68% of the instance Lock Wait Program SQLAgent - TSQL JobStep (Job 0xCC31233C431198 is attributed to 20.88% of the instance Lock Wait		3,470,61 LCK_M_U 1,291.37 LCK_M_U 2,262.37 LCK_M_U 1,291.37 LCK_M_U 1,291.37 LCK_M_U 1,208.24 LCK_M_U 11,449.77 LCK_M_U 1,208.24 LCK_M_U 1,208.24 LCK_M_U	100 9 100 9 100 9 100 9 100 9 51 9 100 9 100 9
	The Instance consumed 5,762 seconds waiting for Lock. 49.75% of its db-time spent waiting for Loc Highlights SQL Statement UPDATE [dbo].[BANK] SET [BANK_NAME] = LTRIM(R is attributed to 60,2% of the instance Lock Wait SQL Statement UPDATE [db] = @1 is attributed to 22.41% of the instance Lock Wait SQL Statement UPDATE [db] = @1 is attributed to 19.01% of the instance Lock Wait TSQL Batch BEGIN TRAN UPDATE [db] [BANK] SET [BANK_NAME is attributed to 39.27% of the instance Lock Wait TSQL Batch BEGIN TRAN UPDATE [db] [BANK] SET [BANK_NAME is attributed to 20.97% of the instance Lock Wait TSQL Batch BEGIN TRAN UPDATE [db] [BANK] SET [BANK_NAME is attributed to 20.97% of the instance Lock Wait TSQL Batch BEGIN TRAN UPDATE [db].[BANK] SET [BANK_NAME is attributed to 20.97% of the instance Lock Wait Program SQLAgent - TSQL JobStep (Job 0x282C65EE5BF9A is attributed to 20.97% of the instance Lock Wait Program SQLAgent - TSQL JobStep (Job 0x198CA7AC67AF5E is attributed to 19.01% of the instance Lock Wait	Resour	3,470,61 LCK_M_U 1,291.37 LCK_M_U 1,095.24 LCK_M_U 2,262.37 LCK_M_U 1,291.37 LCK_M_U 1,208.24 LCK_M_U 1,208.24 LCK_M_U 1,208.24 LCK_M_U 1,208.24 LCK_M_U 1,208.24 LCK_M_U	Wait Event 100 f 100 f 100 f 100 f 100 f 100 f 100 f 100 f 100 f 100 f
-	The Instance consumed 5,762 seconds waiting for Lock. 49.75% of its db-time spent waiting for Loc Highlights SQL Statement UPDATE [dbo].[BANK] SET [BANK_NAME] = LTRIM(R is attributed to 60.24% of the instance Lock Wait SQL Statement UPDATE [db] = @1 is attributed to 22.41% of the instance Lock Wait SQL Statement UPDATE [db] = @1 is attributed to 19.01% of the instance Lock Wait TSQL Batch BEGIN TRAN UPDATE [dbo].[BANK] SET [BANK_NAME is attributed to 39.27% of the instance Lock Wait TSQL Batch @1 int]UPDATE [dbo].[BANK] SET [BANK_NAME is attributed to 29.37% of the instance Lock Wait TSQL Batch @61 int]UPDATE [dbo].[BANK] SET [BANK_NAME is attributed to 20.37% of the instance Lock Wait TSQL Batch BEGIN TRAN UPDATE [dbo].[BANK] SET [BANK_NAME is attributed to 20.37% of the instance Lock Wait Database sales is attributed to 100.68% of the instance Lock Wait Program SQLAgent - TSQL JobStep (Job 0xCC31233C431198 is attributed to 20.88% of the instance Lock Wait		3,470,61 LCK_M_U 1,291.37 LCK_M_U 2,262.37 LCK_M_U 1,291.37 LCK_M_U 1,291.37 LCK_M_U 1,208.24 LCK_M_U 11,449.77 LCK_M_U 1,208.24 LCK_M_U 1,208.24 LCK_M_U	Wait Event 100 f 100 f 100 f 100 f 100 f 100 f 100 f 100 f 100 f 100 f
-	The Instance consumed 5,762 seconds waiting for Lock. 49.75% of its db-time spent waiting for Loc Highlights SQL Statement UPDATE [dbo].[BANK] SET [BANK_NAME] = LTRIM(R is attributed to 60,2% of the instance Lock Wait SQL Statement UPDATE [db] = @1 is attributed to 22.41% of the instance Lock Wait SQL Statement UPDATE [db] = @1 is attributed to 19.01% of the instance Lock Wait TSQL Batch BEGIN TRAN UPDATE [db] [BANK] SET [BANK_NAME is attributed to 39.27% of the instance Lock Wait TSQL Batch BEGIN TRAN UPDATE [db] [BANK] SET [BANK_NAME is attributed to 20.97% of the instance Lock Wait TSQL Batch BEGIN TRAN UPDATE [db] [BANK] SET [BANK_NAME is attributed to 20.97% of the instance Lock Wait TSQL Batch BEGIN TRAN UPDATE [db].[BANK] SET [BANK_NAME is attributed to 20.97% of the instance Lock Wait Program SQLAgent - TSQL JobStep (Job 0x282C65EE5BF9A is attributed to 20.97% of the instance Lock Wait Program SQLAgent - TSQL JobStep (Job 0x198CA7AC67AF5E is attributed to 19.01% of the instance Lock Wait	Resour	3,470,61 LCK_M_U 1,291.37 LCK_M_U 1,095.24 LCK_M_U 2,262.37 LCK_M_U 1,291.37 LCK_M_U 1,208.24 LCK_M_U 1,208.24 LCK_M_U 1,208.24 LCK_M_U 1,208.24 LCK_M_U 1,208.24 LCK_M_U	Wait Event 100 % 100 % 100 % 100 % 100 % 100 % 51 % 100 %

SQL PI – Tune SQL



	Query Insignts > SQL PI		Monday, Jul 22, 2024	04:54:35 PM - Now 1 hour 🔻 🧐	ич керо
Out 10 Number Network Lok Lukh Lug CL Remoter XIP Oler Network YIP Oler					-
Performance Tree Top 12 * Winter Letter View SQL Statements + UPGATE [doi:]GAM2 Top Control View SQL Stateme	SQL_GCP_DBaaS_2019▼ X Oven	view 🧭 SQL PI Memory 👻 Activity 👻 Databases 👻 Services 👻 HADR 👻 Logs 👻 Configuration User-defined 👻			$\varphi = \mathbf{Q}$
Instance Vere Solutions Plane: Instance Vere SQL Statements • VPOATE (doi:]084/0] - Solutions Plane: Instance Vere SQL Statements • VPOATE (doi:]084/0] - Solutions Plane: Instance Vere SQL Statements • VPOATE (doi:]084/0] - Solutions Plane: Instance Vere SQL Statements • VPOATE (doi:]084/0] - Solutions Plane: Instance Vere SQL Statements • VPOATE (doi:]084/0] - Solutions Plane: Instance Vere SQL Statements • VPOATE (doi:]084/0] - Solutions Plane: Instance Vere SQL Statements • VPOATE (doi:]084/0] - Solutions Plane: Instance Vere SQL Statements • VPOATE (doi:]084/0] - Solutions Plane: Instance Vere SQL Statements • VPOATE (doi:]084/0] - Solutions Plane: Instance Vere SQL Statements • VPOATE (doi:]084/0] - Solutions Plane: Instance Vere SQL Statements • VPOATE (doi:]084/0] - Solutions Plane: Instance Vere SQL Statements • VPOATE (doi:]084/0] - Solutions Plane: Instance Vere SQL Statements • VPOATE (doi:]084/0] - Solutions Plane: Instance Vere SQL Statements • VPOATE (doi:]084/0] - Solutions Plane: Instance Vere SQL Statements • VPOATE (doi:]084/0] - Solutions Plane: Instance Vere SQL Statements • VPOATE (doi:]084/0] - Solutions Plane: Instance Vere SQL Statements • VPOATE (doi:]084/0] - Solutions Plane: Instance Vere SQL Statements • VPOATE (doi:]084/0] - Solutions Plane: Instance Vere SQL Statements • VPOATE (doi:]084/0] - Solutions Plane: Instance Vere SQL Statements • VPOATE (doi:]084/0] - Solutions Plane: Instance Vere SQL Statements • VPOATE (doi:]084/0] - Solutions Plane: Instance Vere SQL Statements • VPOATE (doi:]084/0] - Solutions Plane: Instance Vere SQL Statements • VPOATE (doi:]084/0] - Solutions Plane: Instance Vere SQL Statements • VPOATE (doi:]084/0] - Solutions Plane: Instance Vere SQL Statements • VPOATE (doi:]084/0] - Solutions Plane: Instance Vere SQL Statements • VPOATE (doi:]084/0] - Solutions Plane: Instance Vere SQL Statements • VPOATE (doi:]084/0] - Solutions Plane: Instance Vere SQL Statements • VPOATE (doi:]084/0] - Solutions Plane: Instance Vere SQL State	Workload CPU I/O Memory Netwo	ork Lock Latch Log CLR Remote Provider XTP Other			Powered by SOL
Big Op Statement Description Description <thdescription<< td=""><td>Performance Tree Tops: 25 w</td><td>History Advanced Analytics</td><td></td><td></td><td>🖺 View as Pl</td></thdescription<<>	Performance Tree Tops: 25 w	History Advanced Analytics			🖺 View as Pl
Tig upport (abu](set (B) ≤ (B) ≤ (B) Kence (damage) Tig upport (a) upport (b) (abu)(set (B) ≤ (B) ≤ (B) (B) ≤ (Instance View	Dimension Filter: Instance View + SQL Statements + UPDATE [dbo].[BANK]			
Lindext [B] widt [B] eight [B] ei					
Builder Bradition Bester trop (das) (dott) 3 state Bester trop (das) (das) (das) Bester trop (das) (das)	out-	Resource Consumption			Top Wait Even
In dense (priv 2- priv 2- or) In dense (priv2- priv 2- or) In dense (priv 2- priv2- or) In dense (pri	and the second s	PDATE [dbo].[BANK] SET [BANK_NAME] = LTRIM(RTRIM(BANK_NAME)) FROM BANK WHERE [BANK_CODE] = '** Removed by niloht **'		Resource Breakdown	
 IN INSERT INTO (dob) (100.2) SECE INSERT INTO (dob) (100.2) SECE<td>III SOL OLDALE [AN] SCE[A] = GET</td><td></td><td>T⁴</td><td>Lock Wait</td><td>100 00%</td>	III SOL OLDALE [AN] SCE[A] = GET		T ⁴	Lock Wait	100 00%
 Insert IPTO (de)[ORD_1] DeLTE FROM (de)[ORD_1] DeLTE FROM (de)[ORD_1] DeltTE FROM (de)[ORD_1] DeltTE FROM (de)[ORD_1] DeltTE FROM (de)[ORD_1] Steft TO' (PROM_m) S	adr.		sec	- LUCK Wait	100.0078
DELETE FROM [dob] (DODL] DELETE FROM [dob] (DODL] SELCT (astome, july, sum, anound sum of all the active walts and cpu usage, equal to the sector top 100 based starts, Files Sector top	odr		2 ² nds		
Delete FROM (do)[ORDL1] Delete FROM (do)[ORDL2] Select FROM (do)[ORDL2] Select Trade 17.20 17.20 17.40 17.10 17.10 17.10 17.10 17.10 17.20					
 DeLETE FROM (dbo](ORL2] StLECT (oxformer_Jd, Sumiknown StLECT TOP (@RowLmk) COWER StLECT TOP (BROWLmk) COWER StLECT TOP (BROWLKK) COWER	adr	16:56 16:58 17:00 17:02 17:04 17:06 17:08 17:10 17:12 17:14 17:16 17:18 17:20 17:22 17:24 17:26 17:28 17:30 17:32 17:34 17:36 17:38 17:40	17:42 17:44 17:48 17:48 17:50 17:52 17:54		
SELECT (sustainer, jd), Sum(annur, jd) Market Into #Resource@ Select Did: Select Did: Activity Highlights Market Into #Resource@ Select Did:					
 Insert Into #ResourceDB Select Drag Select Tro (P (Revolumi) CONVERS Select Tro (P (Revolumi) CONVERS Select Top (P (P (N (m)) CONVERS Select Crabel Imperson Select Cra	ode	Oversiden Statistics United Statistics			
 SELECT TOP (@RowLimit) CONVER Select Dop (A 2/request, session, Dial A 2/request, session, Di	adr.	Overview Biocking History Activity Highlights			
 Select Distint A2.request_session, insert into #temrace select too 1 Select Too 1 QOBD = B.(dwl), @ Select Too 1 QOBD = B.(dwl), ago Select Too 1 (200 1 7.20 1	out.				
 in the thin of #temp_trace select top 1 is bet top 1 (0BU) = b[(bid]), (0C) in the tin of #temp_trace select top 1 is bet top 100 baseses is bet toreate time from sys. dm, yea is there toreate time from sys. dm, yea is bet toreate time from sys. dm, yea is the state toreate time from sys. dm, yea is the state toreate time from sys. dm, yea is the state toreate time from sys. dm, yea is the state toreate time from sys. dm, yea is the state toreate time from sys. dm, yea is the state toreate time from sys. dm, yea is the state toreate time from sys. dm, yea is the state toreate time from sys. dm, yea is the state toreate time from sys. dm, yea is the state time from sys. dm, yea is the state toreate time from sys. dm, yea is the state toreate time from sys. dm, yea is the state toreate time from sys. dm, yea is the state time from sys. dm, yea is the state toreate time from sys. dm, yea is the state toreate time from sys. dm, yea is the state toreate time from sys. dm, yea is the state time from sys. dm, yea is t	udr.	Active Time			
 a field that the steep trace select top 1 b field 10 DatabaseName, Field c field 10 DatabaseName, Field 10 DatabaseName, Field c field 10 DatabaseName, Field 10 D	adr.			Sum of all the active waits and cpu usage session total activity within the current in	e, equal to the terval.
Image: Insert into #temp_trace select top 10 Image: Insert into #temp_trace select top 11 Image: Insert into #temp_trace sel			1 ² is		
 Big select top 100 DatabaseName, Files Big top Digint, QP1 int, QP2 int, QP3 int, QP3 int, QP1 int, QP2 int, QP3 int, QP1 int, QP	odr		+ and		
Image: Imag	adr.				
Image: Spice derivation of geno (score) Workload related Metrics Image: Spice derivation of geno (score) Search Search Search Search Search Image: Spice score Search		16:56 16:58 17:30 17:32 17:34 17:36 17:38 17:10 17:12 17:14 17:16 17:18 17:20 17:22 17:24 17:26 17:28 17:30 17:32 17:34 17:36 17:38 17:40	17:42 17:44 17:46 17:48 17:50 17:52 17:54		
Image: Spinstance_regread Image: Spin	odr	Workload related Metrics			
Bit SELECT top (@P0) t2.spid, t2.login_ Metric A Metric A Resource Total I Bit Select @DBName As DBName , DB_ Average SQL Response Time Workbad 17.52 Image: Select GDBName As DBName , DB_ Average SQL Response Time Image: Select GDBName As DBName , DB_ Image: Select GDBName , DB_ Image: Select G		Ox Select Metric		Search	0 =
Big Selet @DBName As DBName , DB_ Average SQL Response Time Workload 175.2 Big DROP TABLE #QS_sysfles CPU Usage CPU <<<			Deserves		·•
Image: DROP TABLE #QS_sysfles CPU Usage Color Image: DROP TABLE #QS_sysfles Executions Executions Image: DROP TABLE #QS_sysfles Executions Executions Image: DROP TABLE #QS_sysfles Executions Executions Image: DROP TABLE #QS_sysfles Executions Workload 39.00 Image: DROP TABLE #QS_sysfles Lock Update Lock 6.834.83 6.834.83 Image: Drop Table #QS_sysfles Lock Vipdate Lock 6.834.83	adr.			Total	175.22
Image: Select create time from sys.dm_xe_ Excutions Excutions Workload 39.00 Image: SQL Batches Lock Update Lock Update Lock 0 6,834.83 Image: Databases Lock Wait Lock Vait 6,834.83 6,834.83 Image: Databases Lock Wait Lock Vait 100.00 100.00 Image: Databases Wait Time Percent Workload 100.00 100.00					
Image: Solution of the solution	July 1				
Image: Databases Lock Wait Lock 6,834.83 Image: Programs Lock Content Lock 100 Image: Users Row count Workload 130.00 Image: Users Wait Time Percent Workload 100.00	adr.				
Image: Programs Logical Reads //o 135.00 Image: Programs Logical Reads //o 135.00 Image: Programs Row count Workload 130.00 Image: Programs Wait Time Percent Workload 100.00					· · ·
Image: Second and Second an	<u> </u>				
Gient Machines Wait Time Percent Workload					
Woll Nodu Woll Nodu				<u> </u>	
	Context Infos	Wait Time Percent	Workload		100.00

SQL Optimizer



Quest SQL Optimizer for SQL Server 10.1.2														-	o ×
Optimize SQL Optimize Indexes Find SQL Scan SQL	Manage Plan Guid	les										Search Toad	World	Connection	ptions Help
🖥 🔹 🍃 🔓 SQL Rewrite 1 🗙 🛛 🛨															
SQL Rewrite SQL Details SQL Compare Report									4	JANISCLIENT\MS	SQLSERVER01 (IMDEM	10\janis) DOT	 <defau< li=""> </defau<>	ilt> 💌 🐰 C	istom 🥼
🏕 🛅 • 🖄 🕂 🕴 • • 👭 🛤 • 🖻 • 🔗 🖑 🗸															
Iternative Details 🔹 🔻 🖡	Schema Information	n													- 9
🗋 🗁 Alt4 🔹 🔍 🔍 😥	Summary														• 18
Optimize for Cursor Settings 🔚 Temp Tables [Empty]	Object Used in SQL														
UPDATE [dbo].[BANK]	DOT.dbo.bank	k													
SET [BANK_NAME] = LTRIM(RTRIM(BANK_NAME)) FROM BANK															
WHERE [BANK CODE] + '' >= '** Removed by Foglight **'															
AND [BANK_CODE] <= '** Removed by Foglight **'															
	Summary Details													Ē	Retrieve all
	Summary Details	e Owner Type	Created Datetime	Cardinality	Pages										Retrieve all
	Table Name Table	e Owner Type user table	Created Datetime e 7/23/2024 2:52:37 Pf		Pages 73996										Retrieve all
	Table Name Table														Retrieve all
	Table Name Table													٢	Retrieve all
	Table Name Table	user table	e 7/23/2024 2:52:37 Pt	M 7764288	73996									٢	Retrieve all
•	Table Name Table	user table	e 7/23/2024 2:52:37 Pt	M 7764288	73996	iables								E	Retrieve all
<	Table Name Table	user table	e 7/23/2024 2:52:37 Pt	M 7764288	73996	iables									Retrieve all
liternatives	Table Name Table Image: bank dbo Execution Plan SQL 1	user table	e 7/23/2024 2:52:37 Pt	M 7764288	73996 ttings and Bind Va	iables								Ē	Retrieve all
O faster found (19 not tested)	Table Name Table Image: bank dbo Execution Plan SQL 1	user table	e 7/23/2024 2:52:37 Pf	M 7764288	73996 ttings and Bind Va	ables									Retrieve all
y 0 faster found (19 not tested)	Table Name Table Image: bank dbo Execution Plan SQL 1	user table	e 7/23/2024 2:52:37 Pf	M 7764288	73996 ttings and Bind Va	iables									Retrieve all
y 0 faster found (19 not tested)	Table Name Table Image: bank dbo Execution Plan SQL 1	Information Execution	e 7/23/2024 2:52:37 Pf	M 7764288	73996 ttings and Bind Va	tables									Retrieve all
Iternatives O faster found (19 not tested) how faster alternatives only General	Table Name Table Image: bank dbo Execution Plan SQL 1	user table	e 7/23/2024 2:52:37 Pf I Statistics Schema Info P Index Lucer Altr cution Statistics	M 7764288	73996 ttings and Bind Va Run - All		Logical Reads PP	nysical Reads	Scans	Read-Ahread Reads	Comple Elapsed Time	Comple CPU Time	Buffers Received		Retrieve all
Iternatives Iternatives <td>Table Name Table Table Name Table Execution Plan SQL 1</td> <td>user table</td> <td>e 7/23/2024 2:52:37 Pf I Statistics Schema Info P Index Lucer Altr cution Statistics</td> <td>M 7764288</td> <td>73996 ttings and Bind Va Run - All Response Time</td> <td></td> <td>Logical Reads PH</td> <td>nysical Reads 0</td> <td></td> <td>Read-Ahead Reads</td> <td></td> <td></td> <td>Buffers Received</td> <td>Bytes Received</td> <td>····· 1</td>	Table Name Table Table Name Table Execution Plan SQL 1	user table	e 7/23/2024 2:52:37 Pf I Statistics Schema Info P Index Lucer Altr cution Statistics	M 7764288	73996 ttings and Bind Va Run - All Response Time		Logical Reads PH	nysical Reads 0		Read-Ahead Reads			Buffers Received	Bytes Received	····· 1
Iternatives O faster found (19 not tested) show faster atematives only General ⁽¹⁾ Scenario Name ⁽²⁾ Cinginal 0.0165715 2 0	Table Name Table Table Name Table Execution Plan SQL I	Information Execution pence Level for Index T Exec Test Run Label Exec	e 7/23/2024 2:52:37 Pf n Statistics Schema Info P Index Lucer Altr cution Statistics cution Elapsed Time To	M 7764288 formation Cursor Se ternative RM Test R otal Elapsed Time F 00:00:00.000	73996 ttings and Bind Va Run - All Response Time	Execution CPU Time			0		00:00:00.000	00:00:00.000	4	Bytes Received	••••••••••••••••••••••••••••••••••••••
Alternatives Internatives Inster found (19 not tested) Show faster alternatives only General If Scenario Name Plan C Executions Record Count Status If Criginal 0.0165715 2 Alt4 0.0165722 1	Table Name Table Table Name Table Execution Plan SQL I Change Intellige * Execution Elaosed Time 00:00:00.000	Information Execution ence Level for Index Exec Test Run Label Exec 12:41:12 PM	e 7/23/2024 2:52:37 Pf I Statistics Schema Infe P Index Lise Alte cution Statistics cution Elapsed Time 00:00:00.000	M 7764288 formation Cursor Se ternative RM Test R otal Elapsed Time F 00:00:00.000	73996 ttings and Bind Var Run - All Response Time 00:00:00.001	Execution CPU Time 00:00:00.000	3 3	0	0	0	00:00:00.000	0 00:00:00.000	4	Bytes Received 4 18,5 4 22,)1 ▲ 107 ■
Iternatives Vertice 0 faster found (19 not tested) how faster alternatives only Status General Image: Scenario Name Plan C. • Executions Record Count Status Image: Scenario Name Plan C. • Executions Record Count Status Image: Scenario Name Status Image: Scenario Name Image: Scenario Name Image: Scenario Name Status Image: Scenario Name Image: Scenario Nam Image: Scenario Name <	Table Name Table Table Name Table Execution Plan SQL I	Information Execution ence Level for Index Test Run Label 12:41:12 PM 12:41:12 PM	e 7/23/2024 2:52:37 Pf i Statistics Schema Info i Index Lise Alta cution Statistics cution Elapsed Time Tc 00:00:00.000 00:00:00.000	M 7764288	73996 ttings and Bind Va Run - All Response Time 00:00:00.001 00:00:00.000	Execution CPU Time 00:00:00.000 00:00:00.000	3 3	0	0 1 0	0	00:00:00.000 00:00:00.000 00:00:00.000	0 00:00:00.000 0 00:00:00.000 0 00:00:00.000	4	Bytes Received 4 18,5 4 22,	01 × 4 07 Ξ 95

SQL Optimizer



timize SOL	0.11.1.1.1.1	E 1 COL	C	 Manager Dial C 															tion Options
			Scan SQI	L Manage Plan G	ulaes											Search Toad	wond	Connect	
😂 🖬 🖆 Si	QL Rewrite 1 ×	+																	
Rewrite	SQL Details 🛛 👧 Co	ompare 🔄 🖬 Repor	t										۹	JANISCLIENT\MS	QLSERVER01 (IMDEM	O\janis) DOT	Oefau	ult> 💌	Lift Custom
atives																			
neral						Execution Sta													
		utions Record Coun	t Status	Execution Elapsed Tim						Execution CPU Time				lead-Ahead Reads	Compile Elapsed Time	Compile CPU Time			
🔮 Original	0.0165715	2	0	00:00:00.000	12:41:12 PM	00:00:00		00:00:00.000	00:00:00.00		3	0	0	0	00:00:00.000	00:00:00.000	4		8,501
Alt4	0.0165722	2	0	00:00:00.000	12:41:12 PM	00:00:00		00:00:00.000	00:00:00.00		3	0	1	0	00:00:00.000	00:00:00.000	4		22,107
Alt3	111.9610	2	0	00:00:02.094	12:41:12 PM	00:00:02		00:00:02.094	00:00:02.04		73,996	0	5	0	00:00:00.000	00:00:00.000	8		50,349
Alt2	114.7170	2	0	00:00:00.695	12:41:12 PM	00:00:00		00:00:00.695	00:00:00.65		73,996	0	5	0	00:00:00.000	00:00:00.000	4		19,752
Alt1	24.63000	2	0	00:00:00.335	12:41:12 PM	00:00:00	0.335	00:00:00.335	00:00:00.32	6 00:00:01.313	24,212	0	5	0	00:00:00.000	00:00:00.000	6		33,459
rison 🦯																			
ayout: SoL and P	Plan (Left-Right)	🥸 Swap Panes 🛛 🚖	Maximize																
inal								× (4)		Alt4									- (() (
E [dbo].[BA	ANKI									DATE [dbo].[BAN	IKI								
		RIM (BANK_NAME))								-	TRIM (BANK_N	NAME))						
ET [BANK_NAM OM BANK	ME] = LTRIM(RT	-							UE	DATE [dbo].[BAN SET [BANK_NAME FROM BANK] = LTRIM(P	_							
ET [BANK_NAM OM BANK	ME] = LTRIM(RT	RIM(BANK_NAME) ved by Fogligh							UE	PDATE [dbo].[BAN SET [BANK_NAME FROM BANK HERE [BANK_CODE] = LTRIM(F	** Removed	by Fog						
ET [BANK_NAM	ME] = LTRIM(RT	-							UE	DATE [dbo].[BAN SET [BANK_NAME FROM BANK] = LTRIM(F	** Removed	by Fog						
T [BANK_NAM DM BANK RE [BANK_COE	ME] = LTRIM(RT	-							UE	DATE [dbo].[BAN SET [BANK_NAME FROM BANK HERE [BANK_CODE AND [BANK_CODE] = LTRIM(F	** Removed	by Fog						
ET [BANK_NAM DM BANK RE [BANK_COL Stual Plan Estin	ME] = LTRIM (RT DE] = ¹ ** Remo	ved by Fogligh				Act rows	Estrows A			PDATE [dbo].[BAN SET [BANK_NAME FROM BANK IHERE [BANK_CODE AND [BANK_CODE AND [BANK_CODE	[] = LTRIM(F [] + '' >= ' [] <= '** Re ated Plan	** Removed moved by Fo	by Fog			Act rows	Est rows A,	/E Row	Est cost
ET [BANK_NAM DM BANK RE [BANK_COL ctual Plan Estim	ME] = LTRIM (RT DE] = *** Remo imated Plan size: 9; Est executions	ved by Fogligh				Act rows		/E Row Est cos	UE M st Pla	PDATE [dbo].[BAN SET [BANK_NAME FROM BANK HHERE [BANK_CODE AND [BANK_CODE AND [BANK_CODE Actual Plan Estimation	[] = LTRIM(F [] + '' >= ' [] <= '** Re	** Removed moved by Fo	by Fog			Act rows	Est rows A, 0 1.00	/E Row -100.0%	Est cost
ET (BANK_NAM DM BANK RE (BANK_COD Ctual Plan Estim Est avg row s 4 - Computer Street	ME] = LTRIM (RT DE] = *** Remo imated Plan size: 9; Est executions ute Scalar Vsi: 0.00; Est subtree	ved by Fogligh : 1; cost: 0.006570; Est su	10 **'	9.65;		Actrows	Est rows A, 1.00		st Pla	DATE [dbo], [BAN SET [BANK_NAME FROM BANK HERE [BANK_CODE AND [BANK_CODE AND [BANK_CODE Actual Plan Estim In 4 - 1 and 1	<pre>[] = LTRIM (F [] + '' >= ' [] <= '** Re ated Plan my size. TT, LSV EA ted Loops / Inn executions: 1;</pre>	** Removed moved by Fo	by Fog glight	***		Act rows			Est cost
ET [BANK_NAM M BANK RE [BANK_COE tual Plan Estin Est avg row s 4	ME] = LTRIM (RT DE] = '** Remo imated Plan size: 9; Est executions ute Scalar t%: 0.00; Est subtree cost: 0.000000; Est G	ved by Fogligh : 1; cost: 0.006570; Est su V1 cost: 0.00000;	10 **'	9.65;		Actrows		/E Row Est cos		DATE [dbo].[BAN SET [BANK_NAME FROM BANK HERE [BANK_CODE AND [BANK_CODE AND [BANK_CODE Actual Plan Estim n 4 	<pre>[] = LTRIM (F [] + '' >= ' [] <= '** Re ated Plan my size. TT, LSV EA ted Loops / Inn executions: 1;</pre>	** Removed moved by Fo	by Fog. oglight		·	Act rows			Est cost
ET [BANK_NAM DM BANK RE [BANK_COL Ctual Plan Estron Est avg row 1 Est avg row 1 Est (Jo Est avg	ME] = LTRIM (RT DE] = *** Remo imated Plan size: 9; Est executions ute Scalar Vsi: 0.00; Est subtree	ved by Fogligh : 1; cost: 0.006570; Est su Vucost: 0.00000; utions: 1;	10 **'	9.65;		Actrows		/E Row Est cos		DATE [dbo].[BAN SET [BANK_NAME FROM BANK HERE [BANK_CODE AND [BANK_CODE AND [BANK_CODE Actual Plan Estim 4 4 2 Actual Plan Estim Actual Plan Estim	:] = LTRIM (F ::] + '> :] <=	** Removed by Fo moved by Fo erutions. 1, er Join ubtree cost: 0.000 jst CPU cost: 0.1 st executions: 1;	by Fog. oglight	***		Act rows	0 1.00		Est cost 0.000004 1
ET [BANK_NAM BANK RE [BANK_COI Ctual Plan Estimation Est avg row :: Est cost Est cost Est row Est row Est row Est row Est row	ME] = LTRIM (RT DE] = '** Remo mated Plan size: 9; Est executions ute Scalar t %: 0.00; Est subtree costs: 0.00000; Est C row size: 4% Est exec ested Loops / Inner t executions: 1;	ved by Foglig! : 1; cost: 0.006570; Est su PU cost: 0.00000; utions: 1; Join	ut ***			Act rows 0	1.00	/E Row Est cos 0.0000	st Pla	DATE [dbo].[BAN SET [BANK_NAME FROM BANK HERE [BANK_CODE AND [BANK_CODE AND [BANK_CODE Actual Plan Estim 4 4 2 Actual Plan Estim Actual Plan Estim	:] = LTRIM (F :] <= '** Re	** Removed moved by Fo custoris: 1, er Join ubtree cost: 0.00 j Est CPU cost: 0.1 st executions: 1;	by Fog: oglight 5571; Ests 000004;	***		Act rows			Est cost
ET [BANK_NAM M BANK RE [BANK_COL Lual Plan Estimation Est avg row 1 Est avg row 1 Est col Est avg row 1 Est tool Est tool Est tool Est avg 3 = - Compu Est col Est avg row 1 Est tool Est avg row 1 Est tool Est avg row 1 Est tool Est avg row 1 Est avg row 1 Est tool Est avg row 1 Est tool Est avg row 1 Est avg	ME] = LTRIM (RT mated Plan size: 9: Est executions uté Scalar t %: 0.00; Est subtree cost: 0.00000; Est G t %: 0.00; Est subtree cost: 0.00000; Est G t %: 0.00; Est subtree cost: 0.00000; Est Subtree t %: 0.00; f		it ***			Act rows 0	1.00	/E Row Est cos 0.0000		DATE [dbo].[BAN SET [BANK_NAME FROM BANK HERE [BANK_CODE AND [BANK_CODE AND [BANK_CODE Actual Plan Estim 4 4 2 Actual Plan Estim Actual Plan Estim	:] = LTRIM (F :] : '*** Re sted Plan	** Removed moved by Fo economics 1, er Join ubtree cost: 0.00 j Est CPU cost: 0.0 st executions: 1; Est subtree cost: Est subtree cost:	by Fog: bglight 5571; Est st 5000004; 0.003284; j t: 0.00000	ubtree cost %: 39.65		Actrows	0 1.00		Est cost 0.000004
ET (BANK_NAM M BANK RE (BANK_COL Ctual Plan Estr Ctual Plan Estr Status Compute Est avg row : Est cost Est cost Est avg row : Est avg row : Es	ME] = LTRIH (RT DE] = '** Remo mated Plan size: 9; Est executions size: 9; Est executions (************************************	ved by Fogligt : 1; cost: 0.006570; Est su 20 cost: 0.00007; utbors: 1; 30in btree cost: 0.006570; E st: CPU cost: 0.0006570; Est CPU cost: 0.000670; Est Cost: 0.000670; Est CPU cost: 0.000670; Est Cost: 0.000670; Est Cost: 0.000670; Es	it **' bbree cost %: 39 Est subtree cost %			Act rows 0	1.00	VE Row Est cos 0.0000 -100.0% 0.0000		DATE [dbo].[BAN SET [BANK NANE FROM BANK HERE [BANK_CODE AND [BANK_CODE AND [BANK_CODE AND [BANK_CODE AND [BANK_CODE Actual Plan Estim actual Plan Estim Actual Plan Estim Actual Plan Estim Comparison actual actual Estim Set (State State Sta	I = LTRIM (P I + ' > > ' ' I - ' ** Re ated Plan - ' ** Re m size: Tr, Lak EA - ' ** Re m size: No.03; Et I - ' ** Re ost % 0.03; Cost 0.00500 / O cost: 0.00000 / Costu 0.0000 / Costu 0.0000 / Est // O cost: 0.00000 / Est // O cost: 0.00000 / Est // O cost: 0.000 / Est // O cost: 0.00000 / D cost 0.00000 / O cost: 0.0000 / O cost: 0.000 /	** Removed moved by Fo economics 4, er Join ubtree cost: 0.000 (Est CPU cost: 0.1 St Est CPU cost: 2000); Est cost: 2000); Es	by Fog: bglight 5571; Est si 000004; 0.003284; ji t: 0.00000 : 1;	ubtree cost %: 39.65		Act rows	0 1.00	-100.0%	Est cost 0.000004 '
ET (BANK_NAM M BANK RE (BANK_COL Ctual Plan Estr Ctual Plan Estr Status Compute Est avg row : Est cost Est cost Est avg row : Est avg row : Es	ME] = LTRIH (RT DE] = '** Remo mated Plan size: 9; Est executions size: 9; Est executions (************************************		it **' bbree cost %: 39 Est subtree cost %			Actrows 0	1.00	/E Row Est cos 0.0000		DATE [dbo].[BAN SET [BANK NANE FROM BANK HERE [BANK_CODE AND [BANK_CODE AND [BANK_CODE AND [BANK_CODE AND [BANK_CODE Actual Plan Estim actual Plan Estim Actual Plan Estim Actual Plan Estim Comparison actual actual Estim Set (State State Sta	I = LTRIM (P I + ' > > ' I + ' > > ' I + ' > > ' I - ' > ' > ' Read ated Plan I'' >	** Removed moved by Fo moved by Fo person (000):5: 4, er Join ubtree cost: 0.00 (5: CPU cost: 0.1 st executions: 1; f Est subtree cost: 2000; Est CPU cost: 15: Est executions (1007); [db0],[b1 (1): 1):	by Fog: glight 5571; Est si 000004; 0.003284; tt 0.00000 : 1; ank].[pk_	btree cost %: 39.65 st subtree cost %: [;] bank2]	19.81;	Act rows	1.00	-100.0%	Est cost 0.000004 * 0.000001
ET (BANK_NAM M BANK RE (BANK_COL ctual Plan Estr test avg row : 4	ME] = LTRIH(RT mated Plan size: 9; Est executions ute Scalar Ys(1.00; Est subtree cost: 0.00000; Est Class Ys(1.00; Est subtree row size: 44; Est exec ested Loops / Inmer texecutions: 1) t cost %: 0.03; Est adu (1)(c cost: 0.00000; E) t cost %: 0.03; Est adu (1)(c cost: 0.00000; E) Est cost %: 19.81; Est cost %: 19.81;	ved by Foglig: : 1; cost: 0.000570; Est su 20 cost: 0.00000; ubons: 1; Join bree cost: 0.006570; fo secutions: 1; [/dbo]/bank]/pk_] Est subtree cost: 0.003	bbree cost %: 35 St subtree cost % 3; bank2]	%: 39.65;		Actrows 0	1.00	VE Row Est cos 0.0000 -100.0% 0.0000		DATE [dbo].[BAN SET [BANK NANE FROM BANK HERE [BANK_CODE AND [BANK_CODE AND [BANK_CODE AND [BANK_CODE AND [BANK_CODE Actual Plan Estim actual Plan Estim Actual Plan Estim Actual Plan Estim Comparison actual actual Estim Set (State State Sta	I = LTRIM (F I + ' > > ' I + ' > > ' I - ' > * * Re atted Plan - ' > * * Re mr Size: Tr, Lot EA - bet Loops / Innexecutions: I: - oost %: 0.03; Est : 0.000; Dest / 0.00; Est	** Removed moved by Fo er Join ubtree cost: 0.00 (5 st CPU cost: 0.1) (5 st executions: 1; est subtree cost: 0000; [5 st CPU cost (5); [51	by Fog: oglight 5571; Est st 000004; 1; ank].[pk_ cost: 0.000	bbree cost %: 39.65 Est subtree cost %: [:] bank2] 2283; Est subtree cost	19.81;	Act rows	1.00	-100.0%	Est cost 0.000004 4 0.000001
ET [BANK_NAM M BANK RE [BANK_COD ctual Plan Estr test avg row : 4	ME] = LTRIH(RT mated Plan size: 9; Est executions ute Scalar Ys(1.00; Est subtree cost: 0.00000; Est Class Ys(1.00; Est subtree row size: 44; Est exec ested Loops / Inmer texecutions: 1) t cost %: 0.03; Est adu (1)(c cost: 0.00000; E) t cost %: 0.03; Est adu (1)(c cost: 0.00000; E) Est cost %: 19.81; Est cost %: 19.81;	ved by Foglig: : 1; cost: 0.006570; Est su VL cost: 0.00000; utons: 1; Join three cost: 0.006570; Est CPU cost: 0.0000570; Est CPU cost: 0.000570; Est CPU cost: 0	bbree cost %: 35 St subtree cost % 3; bank2]	%: 39.65;		Actrows 0	1.00	VE Row Est cos 0.0000 -100.0% 0.0000		DATE [dbo].[BAN SET [BANK NANK FROM BANK HERE [BANK_CODE AND [BANK_CODE CACUAL Plan (BANK_CODE CACUAL Plan (BANK_CODE CACUAL Plan (CACUAL Plan) CACUAL PLAN (CACUAL PLAN (CACUAL PLAN) CACUAL PLAN (CACU	I = LTRIM (P I = LTRIM (P I = I = ITRIM (P I = ITRIM (P <t< td=""><td>** Removed moved by Fo er Jon ubtree cost: 0.00 ; Est CPU cost: 0.1 5t executions: 1; Est subtree cost: 15; Est executions 15; Est executions 15; Est executions 15; Est executions 19; 81; Est subtree 0.003125; Est CPU 19; 81; Est subtree</td><td>by Fog: bglight 5571; Est si 000004; 1: ank].[pk_ cost: 0.00; U cost: 0.00</td><td>bbree cost %: 39.65 Est subtree cost %: [:] bank2] 2283; Est subtree cost</td><td>19.81;</td><td></td><td>0 1.00 1.00 1.00</td><td>-100.0%</td><td>Est cost 0.000004 4 0.000001 0.003283</td></t<>	** Removed moved by Fo er Jon ubtree cost: 0.00 ; Est CPU cost: 0.1 5t executions: 1; Est subtree cost: 15; Est executions 15; Est executions 15; Est executions 15; Est executions 19; 81; Est subtree 0.003125; Est CPU 19; 81; Est subtree	by Fog: bglight 5571; Est si 000004; 1: ank].[pk_ cost: 0.00; U cost: 0.00	bbree cost %: 39.65 Est subtree cost %: [:] bank2] 2283; Est subtree cost	19.81;		0 1.00 1.00 1.00	-100.0%	Est cost 0.000004 4 0.000001 0.003283
ET (BANK, NAM OM BANK RR (BANK_COL Ctual Plan Estr Est avg row : 4 - Compute Est avg row : 4 - Est avg row : 5 st cost Est avg 3 - Charles Est avg 3 - Charles Est avg row : 5 st cost Est avg row : 5 st cost Est avg row : 1 - Compute Est avg row : 1 - C	ME] = LTRIH (RT DE] = '** Remo mated Plan size: 9; Est executions size: 9; Est executions size: 9; Est executions size: 9; Est executions: 1; two size: 44; Est exec sted loops / Inner t executions: 1; loop (State Seek [DOT Act executions: 1; Est cot %: 10.30; Est 1/O cost %: 10.30	ved by Foglig: : 1; cost: 0.006570; Est su U cost: 0.00000; ubors: 1; Join btre cost: 0.006570; E St CPU cost: 0.00 (Jdbo) [Jbank] (pk] Est subtree cost: 0.003 (25; Est CPU cost: 0.00 (25; Est CPU cost: 0.00	bbree cost %: 35 St subtree cost % 3; bank2]	%: 39.65;		Act rows 0	1.00 1.00 1.00	/E Row Est cos 0.0000 -100.0% 0.0000 -100.0% 0.0032 -100.0% 0.0032	U B B C C C C C C C C C C C C C C C C C	DATE [dbo].[BAN SET [BANK NANK FROM BANK HERE [BANK_CODE AND [BANK_CODE CACUAL Plan (BANK_CODE CACUAL Plan (BANK_CODE CACUAL Plan (CACUAL Plan) CACUAL PLAN (CACUAL PLAN (CACUAL PLAN) CACUAL PLAN (CACU	I) = LTRIM (F) + · · > · · · · Re ated Plan made: - Tr, Lot EA ted Loops / Inn executions: 1; 0 cost %: 0.03; Est s /0 cost: 0.000000 Est /0 cost: 0.00 Est cost %: 0.03; Est s Compute Scalas Est cost %: 0.03; Est s Loc st %: 0.00; Est /0 cost: 0.00 Est cost %: 0.00; Est /0 cost: 0.00 Est cost %: 0.00; Est /0 cost: 0.00 Est cost %: 0.00; Est /0 cost: 0.00 Est avg row size: Bet J/O cost: 0.00 Est avg row size: Bet J/O cost: 0.00 Est avg row size: Bet J/O cost: 0.00	** Removed moved by F(er Join ubtree cost: 0.00(j Est CPU cost: 0.1) Est subtree cost: 3000; j Est CPU cos Est subtree cost: 3000; j Est CPU cos 1; j Est executions: 1; j Est executions 1; j Est execution	by Fog: bglight 5571; Est si 000004; 1: ank].[pk_ cost: 0.00; U cost: 0.00	bbree cost %: 39.65 Est subtree cost %: [:] bank2] 2283; Est subtree cost	19.81;		1.00	-100.0%	Est cost 0.000004 * 0.000001
ICM BANK IRE [BANK_COL Actual Plan Est avg row 4 - Compu- Est avg row 5 trob Est avg row 6 to the 1 - 1 2 - 5	ME] = LTRIH (RT matted Plan size: 9; Est executions ute Scalar Vision 00000; Est C Stabubre cost: 0.00000; Est C Stabubre cost: 0.00000; Est C Stabubre cost: 0.00000; Est C Stabubre cost: 0.00000; Est C Stabubre (1) C cost: 0.00000; Est cost Vision 19; Est cost Vision 19; Est cost Vision 19; Est Vig Cost: 0.0000; Est avg row size: 41; Est Vig Cost: 0.0000; Est avg row size: 19;	ved by Foglig: : 1; cost: 0.006570; Est su V cost: 0.000570; Est su V cost: 0.00000; utors: 1; Join types cost: 0.0005570; Sit CPU cost: 0.00005 Sit CPU cost: 0.00005 Sit CPU cost: 0.0000 Sit CPU cost: 0.0000 Sit CPU cost: 0.0000 Sit Sit CPU cost: 0.0000 Sit CPU cost: 0.000	bbree cost %: 35 St subtree cost % 3; bank2]	%: 39.65;		0	1.00 1.00 1.00	/E Row Est cos 0.0000 -100.0% 0.0000 -100.0% 0.0032 -100.0% 0.0032	U st st () () () () () () () () () ()	DATE [dbo].[BAN SET [BANK_NANK FROM BANK HERE [BANK_CODE AND [BANK_CODE CACUAL Plan CACUAL	t] = LTRIM (F t] + + + + + + + + + + + + + + + + + + +	** Removed by Fr moved by Fr er Join ubtre cost: 0.00 jest CPU cost: 0.1 jest cPU cost: 0	by Fog: bglight 5571; Est st 0000004; it: 0.000000 : 1; ank].[pk_ cost: 0.00; U cost: 0.00; U cost: 0.00; U cost: 1;]	bbree cost %: 39.65 Est subtree cost %: [:] bank2] 2283; Est subtree cost	19.81; ;t %: 19.81;		0 1.00 1.00 1.00	-100.0%	Est cost 0.000004 4 0.000001 0.003283

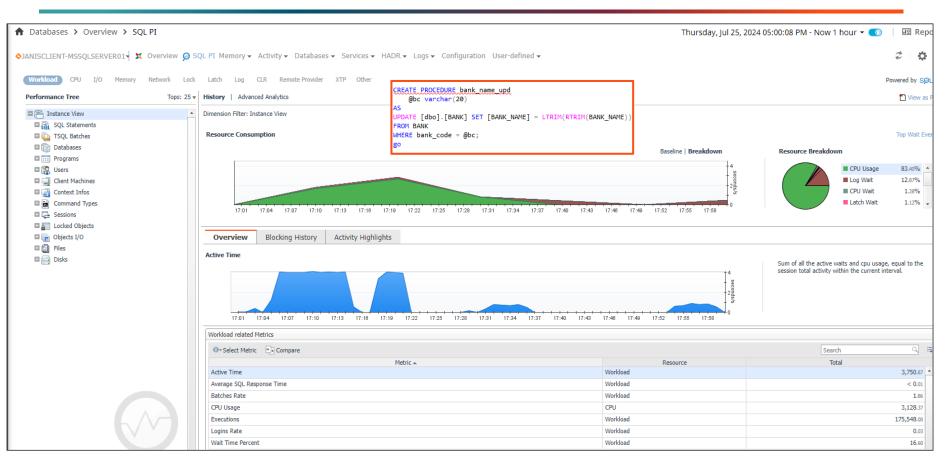
Change Heap to Clustered Index

			9 CF	[bank_name] [address] [bo].[bank]([varchar](2 [varchar](50) [varchar](50) [rchar](2) NU	0) NULL, NULL,				
7/24/24, 8:09 PM Cache Collected 0x0600070086d9220d20db6103 Date Type Plan Handle	36b010000010000			[zip] [varc CONSTRAINT [PK [bank_code])	har](10) NUU BANK] PRIM		TERED	Compare Plans	Generate Plan	Open in SSMS
Statement	Plan Analysis		_)(N [PRIMARY]						
UPDATE [dbo].[BANK] SET [BANK_NAME] = LTRIM(RTRIM(BANK_NAME)) FROM BANK WHERE [BANK_CODE] = "** Removed by Foglight **"	Total cost: 0.0165715 Total I/O co	st: 0.0162500 Total CPU cost:	: 0.0003215 ir	isert into ban	k select * f	from <u>bank_old</u>	;			
	Plan Details Operator	Analysis Object Analysis	5							
									Search	् म्
	Operator	Object	Operator Cost	Subtree Cost	I/O Cost	CPU Cost	Rows		Arguments	
Before	Table Update (Update)	dbo.bank	60.35 %	0.0165715	0.0100000	0.0000010		1 [DOT].[dbo].[bank].[b	oank_name] = [Expr1003	1 🔺
	Compute Scalar		0.00 %	0.0065705	0.0000000	0.0000001		1		
	Nested Loops (Inner Join)		0.03 %	0.0065704	0.0000000	0.0000042		1		
	Index Seek	dbo.bank.pk_bank2	19.81 %	0.0032831	0.0031250	0.0001581		1		
	RID Lookup	dbo.bank	19.81 %	0.0032831	0.0031250	0.0001581		1		
7/24/24, 8:04 PM + Cache Collected 0x060007004ad3961f50d9da44 Date Type Plan Handle	6b010000010000							Compare Plans	Generate Plan	Open in SSMS

Quest

Statement	Plan Ana	ysis								
UPDATE [dbo].[BANK] SET [BANK_NAME] = LTRIM(RTRIM(BANK_NAME)) FROM BANK WHERE [BANK_CODE] = '** Removed by Foglight **'	Total cos	: 0.0132842 Total I/O cost: 0.0	100000 Total CPU cost: 0	.0000010						
	Plan	Details Operator Analys	is Object Analysis							
									Search	् ।
After		Operator	Object	Operator Cost	Subtree Cost	I/O Cost	CPU Cost	Rows	Arguments	
7 (110)	Cluster	ed Index Update (Update)	dbo.bank.PKBANK	100.00 %	0.0132842	0.0100000	0.000010		1 [DOT].[dbo].[bank].[bank_name] = [Expr1002]	

Create Reusable Stored Procedure



inct

Comparison of Changes = Improvement

- Heap to Clustered Index (on bank_code)
- Change query from passing literals to parameters
 - With cached stored procedure



Another SQL Optimizer Example

Who registered yesterday for SQL Tuning?

SELECT s.fname, s.Iname, r.signup_date FROM student s

INNER JOIN registration r ON s.student_id = r.student_id

INNER JOIN class c ON r.class_id = c.class_id

WHERE c.name = 'SQL TUNING'

AND r.signup_date BETWEEN :beg_date AND :beg_date AND r.cancelled = 'N'

Execution Stats –3,050,570 Logical Reads ⁴ Execution Time – Approx. 2 seconds Wait Type – ASYNC_NETWORK_IO

10 Minute Time slice

Overview Blocking H	History Activ	ity Highlights							
Average SQL Respor	nse Time								
d				i	i				1.0 0.0
	16:11	16:12	16:13	16:14	16:15	16:16	16:17	16:18	18:19
Workload related Metri		_			SQL Te:	dt.			
		t 💼 Analyze P			-				
Active Time	letric 🔺		To	cai 901.79		student	s.lname, r. s	signup_date	
Average SQL Respons	se Time			1.73			IN registrat	ion r r.student id	
CPU Usage				168.68		INNER JO	IN class c		
CPU Wait				26.93	WHERE		class_id = c me = 'SOL TU		
Executions				523.00	RITEIX	AND r.si	gnup_date BE	TWEEN @beg_dat	e AND @beg_date + 1
Logical Reads Wait Time Percent				3,050,570.00 81.30		AND r.ca	ncelled = 'N		
Top Wait Events									
Resource: All Wa	ait Events	¥							
Category			Event N	ame		9	% of Total W	/ait Time	Wait Time 👻
Network Wait	ASYNC	_NETWORI	K_IO					77.85	702.04
CPU Wait	SOS_S	CHEDULER	_YIELD					2.99	26.93
I/O Wait	PAGEI	OLATCH_SH	H					1.19	10.77
Memory Wait	MEMOR	RY_ALLOCA	ATION_EXT					0.36	3.21
Memory Wait	RESER	VED_MEMO	RY_ALLOC	ATION_EXT				0.07	0.59

liest

Sql Optimizer Indexes .011 to .001

Optimize SQL Optimize Indexes Find SQL Scan SQL Manag	je Plan Guides Community				Search Toad Work	≠ ▼ Conr	ection Options Help
🛅 🔻 😹 📘 🕞 Class without idx 🖉 Class_1 🗙 🕞 sales_without_index	🕞 sales_1 🕞 dot_without_idx 🕞 DOT_1	+					
SQL Rewrite SQL Details 🎭 Compare 🖬 Report					🚯 10.0.0.10 (sa) 🛛 test	Oefault	> 🔳 🎽 2 🏄 2
🏄 🔚 + 🍇 V Y + VY 🛤 + 🖻 + 🛷 🐰 🗸							-
Alternative Details Alternative Details Coptinue for Cursor Settings Temp Tables [Empty] SELECT s.fname, r.signup_date FROM student s INNER JOIN registration r ON s.student_dd = r.student_id INNER JOIN registration r ON r.class_id WHERE c.name = @cl_name AND r.cancelled = 'N' From State and @beg_date + 1 AND r.cancelled = 'N' From State and Beg_date + 1 Concentration and the set of the set							- 4
Alternatives Alternatives Alternatives Alternatives General General General	💡 Index : 🍇 User Alternative - 👫 Test Run - Al						•••••••••••••••••••••••••••••••••••••••
🧋 📑 Scenario Name Plan Cost Executions Record Count Status 🛛 🕌 🔤	cution Elapsed Time Test Run Label Execution Elapsed Time	Total Elapsed Time Response Time Execution		Scans Read-Ahead Reads Compile	Elapsed Time Compile CPU Time E	uffers Received B	vtes Received
	00:00.011 1:35:19 PM 00:00:00.011		00:00:00.016 477 0		00:00:00.000 00:00:00.000	7	35,147
	00:00.006 1:35:19 PM 00:00:00:00.006		00:00:00.000 97 C		00:00:00.000 00:00:00.000	7	34,311
	00:00.001 1:35:19 PM 00:00:00.001		0:00:00.000 61 0		00:00:00.000 00:00:00.000	6	30,111
	00:00.010 1:35:19 PM 00:00:00.010		00:00:00.000 425 C		00:00:00.000 00:00:00.000	7	35,489
	00:00.001 1:35:19 PM 00:00:00.001		00:00:00.000 121 C		00:00:00.000 00:00:00.000 00:00:00.000	7	40,069
	00:00.004 1:35:19 PM 00:00:00.004 00:00.001 1:35:19 PM 00:00:00.001		00:00:00.000 84 C		00:00:00:000 00:00:00.000	6	34,391 30,185
	00:00:001 1:35:19 PM 00:00:00101 00:00:001 1:35:19 PM 00:00:00:001		00:00:00:000 108 C		00:00:00.000 00:00:00.000	7	40,147

Quest

Sql Optimizer Indexes .011 to .001

Optimize SQL Optimize Indexes F	ind SQL Scan SQL Manage Pl	lan Guides Community							Search Toad World	Connection	n Options Help
🛅 👻 🚔 🔲 🛛 🙀 Class without idx 🖉 🕼 class	5_1 × 🔯 sales_without_index 🕻	🕽 sales_1 🛛 🞲 dot_without_i	idx 💭 DOT_1 +								
SQL Rewrite SQL Details 👧 Compare	Report							6 , 10.0	.0.10 (sa) test	Oefault>	🔳 🧏 2 🍰 2
🎒 🚡 · 🍇 🕴 💡 · 🙌 🛲 · 🖻,	- de da ✓										
Alternative Details											
🖁 💭 🗁 🍸 Index2											• • • • •
🖥 🔲 Optimize for Cursor Settings 📰 Temp Tables [E	impty]										
SQL Text Virtual Indexes Index Script	• Next color control to be active and the	Forth For									
🖉 🕂 🗙 Rename 🗕 Clear 🔳 🗐	A Please make connection to get an updated	I Cable list.									
PT_SX_IDX_4D2AC5B21ADD693 👄	Tables / Columns				Indexed	Iolumns					
Figure 3 registration Class_id, cancelled, signup_date	registration				 class_id cancelled 	Ascending Ascending	*				1
Space class_id, cancelled, signup_date Space class_id, cancelled, signup_date	Connect to view column					ate Ascending					+
SQL Information											
					Include C	olumns					
Exe					>						
Execution Plan					-						
Cursor	Index Type UNIQUE CLUSTERED										
SOF SE											
Setting	Advanced Options										
Alternatives		N									····· 🕈 🗸
A Not optimized Y Inde (Click Optimize to start)	ex2	Index 🖄 User Alternative 👫 Te	est Run - All								
Show faster alternatives only	"										
General		Exe	ecution Statistics								
Scenario Name Plan Cost Executions				Elapsed Time Response Time Exe							
Alt32 0.7938320 2 Alt33 0.6281010 2		0.019 1:35:19 PM 0.011 1:35:19 PM	00:00:00.019	00:00:00.019 00:00:00.021 00:00:00:013	00:00:00.031	584	0 3	0 00:00:00.000	00:00:00.000	7	38,041 *
Alt34 0.8185790 2			00:00:00.026	00:00:00.026 00:00:00.028	00:00:00.031	579	0 3	0 00:00:00.000	00:00:00.000	8	43,693
Index1 0.2182100 2	20 00:00:00	0.006 1:35:19 PM	00:00:00.006	00:00:00.006 00:00:00.008	00:00:00.000	97	0 2	0 00:00:00.000	00:00:00.000	7	34,311
	20 00:00:00	1:35:19 PM	00:00:00.001	00:00:00.001 00:00:00.004	00:00:00.000	61	0 3	0 00:00:00.000	00:00:00.000	6	30,111
V Index3 0.5868010 2			00:00:00.010	00:00:00.010 00:00:00.013	00:00:00.000	425	0 2	0 00:00:00.000	00:00:00.000	7	35,489
Index4 0.6401440 2 Index5 0.0956162 2			00:00:00.012	00:00:00.012 00:00:00.015 00:00:00:00 001 00:00:004	00:00:00.016	464	0 2	0 00:00:00.000	00:00:00.000	7	35,065 40,069
Index6 0.2077420 2			00:00:00.001	00:00:00.001 00:00:00.004	00:00:00.000	84	0 2	0 00:00:00.000	00:00:00.000	7	34,391
Index7 0.0453280 2			00:00:00.001	00:00:00.001 00:00:00.003	00:00:00.000	48	0 3	0 00:00:00.000	00:00:00.000	6	30,185
Vindex8 0.5763330 2		0.011 1:35:19 PM	00:00:00.011	00:00:00.011 00:00:00.014	00:00:00.016	412	0 2	0 00:00:00.000	00:00:00.000	7	35,565
Index9 0.0851484 2	20 00:00:00	0.001 1:35:19 PM	00:00:00.001	00:00:00.001 00:00:00.004	00:00:00.000	108	0 3	0 00:00:00.000	00:00:00.000	7	40,147 *

Quest

Improvement?



• 4x faster, 3x throughput

Workload				🔁 View
Max: 0.29 Avg: 0.29 Min: 0.29	Max: 0.34 Avg: 0.34 Min: 0.34		0.29 % 0.34 %	
Max: 0.29 Avg: 0.29 Min: 0.29 Comparison	Max: 0.34 Avg: 0.34 Min: 0.34	[2	■0.29 % ■0.34 % Total Changes	
	Max: 0.34 Avg: 0.34 Min: 0.34	2	Total Changes	0
Comparison	Max: 0.34 Avg: 0.34 Min: 0.34		Total Changes	0
Comparison Statistics	•		Total Changes Accounts Database Configuration	0
Comparison Statistics Active Time (seconds) Average SQL Response Time (seconds) CPU Usage (seconds)	208	305 5.2 8.3	Accounts Database Configuration Database Objects	0 0
Comparison Statistics Active Time (seconds) Average SQL Response Time (seconds)	208 1.3	305 5.2	Accounts Database Configuration Database Objects Execution Plan	0 0 0

Summary

Quesť

1. Focus on the queries that impact performance the most

– Query Insights

2. Review baselines & changes

Utilize compare & change tracking features

3. Review key metrics & query execution plans

Response time, executions, logical_reads & row_counts

4. Use wait types to identify bottlenecks

- They give clues to best tuning approach
- 5. Determine if it's a poorly written query or database design flaw

6. Consider tuning with SQL Optimizer

More Performance Tuning Help...

Quest

					Resources	Blogs Forums	₩	م ک
Quest	Products	Solutions	Support & Services	Partners	About	Free Trials	Requ	est Pricing

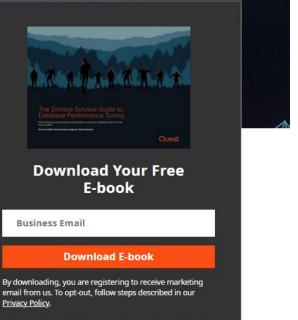
The Zombie Survival Guide to Database Performance Tuning

You know that feeling of banging your head against a SQL query or an entire database for minutes – then hours – and watching your day vanish and feeling your brain turn to mush?

Nobody wants you to be a database zombie.

That's why we've pulled together a few decades of our experience in improving database performance to bring you this guide. You'll find six steps on database tuning, with plenty for both newbies and old hands to latch onto and learn from. We've also included a section with variations for Microsoft SQL Server, Oracle Database, MySQL and PostgreSQL.

Zombie Survival Guide



reCAPTCHA protects this site. See Google's <u>Privacy Policy</u> and Terms of Use.

Thank You





Looking Forward to Seeing You There Register Today!

> Foglight 101 Series: Episode 5 Exploring REST API Functionality

> > Clay Jackson

May 21st

