Background

Patient A got in touch because they were having performance pain with $VENDOR’s applications. Patient A wasn’t sure if the problem was hardware, their configuration, or something in $VENDOR’s code.

$VENDOR wasn’t sure either, but they were pretty sure it wasn’t their code.

Patient A also wanted help with maintenance and best practices.

Check out what happened next!

BRENT OZAR UNLIMITED®

Patient A
SQL Critical Care®

AKA: “Why is this thing so slow?

Executive Summary

You filled in the blank

“Our # 1 pain point is performance”

Your performance prescription

Hardware isn't the problem.

You can make $VENDOR's application faster with indexes in the short term.

- Add non-clustered indexes to speed up jobs and queries during the day.
- Disable unneeded non-clustered indexes on high insert tables.

But that won't fix everything!

Long term, $VENDOR needs to make schema changes to improve query performance.

You can speed up $VENDOR's index maintenance right away by tweaking some settings in the job.

Why can’t you change it all now?

Ask $VENDOR if you can create and disable indexes

- You need to make sure it’s OK to change this, so you don’t break future upgrades
- If they’re concerned about this, it may be OK with them if you maintain a list of all your changes and revert them prior to a planned upgrade

You shouldn’t change the schema on their tables to fix the computed column problem

- This will require testing and careful consideration by $VENDOR as it impacts thousands of columns
- Escalate the issue to them as soon as possible
- This impacts query optimization for all their customers

Let’s dig in!

Speeding up Performance

1. Indexes and UndisclosedToolName processing
2. Type conversions causing your query optimization problem
3. Tuning $VENDOR’s index maintenance

1. Indexes and UndisclosedToolName processing

UndisclosedToolName processing was slow this week

When we looked at the SQL Server, we saw:

- Lots of small inserts
- Duration was usually under 1 second, sometimes under two seconds

UndisclosedToolName processing isn’t suffering from long running commands, it’s lots of commands that are actually very fast

We verified this using SQL Server’s execution plan cache and other dynamic management views

Indexes can relieve some pain

By indexing for the frequent commands, you can

- Reduce IO for them
- Reduce CPU for them
- Reduce duration

And that speeds them up!

You just need to sync up with $VENDOR to make sure that changing this doesn’t violate your support contract

REDACTED
(9 slides of specific index tuning recommendations)

2. Type conversions causing your query optimization problem

This impacts the $VENDOR's SecretName tool – and more

All the queries we saw running from that tool had common problems

In each case, the SQL Server optimizer was warning that it couldn't generate a good execution plan

We saw this in other queries too, but it seemed to be especially bad for the SecretName queries, which are a big pain point

Redacted - Screenshots of type conversion error with query specifics

Computed columns are at the heart of this

System functions in computed columns are converting types and causing problems

Queries are using these columns in their joins and where clauses

The computed columns are NOT persisted

- That means they have to be evaluated each time

This is:

1. Forcing scans instead of seeks
2. Causing bad query optimization - extra operators to filter data and higher CPU usage

Let’s reproduce the issue

REDACTED

(customized training resources showing an example of the problem and the fix)

Putting this all together

Ideally, computed columns wouldn’t be used like this

However, that would be a huge change

Request $VENDOR change the computed columns to be *persisted*

- Warning: adding a computed column that is persisted may get messed up if there’s another copy of the column that is NOT persisted

The computed columns can also be indexed, but that isn’t needed for every computed column

Get a list of all computed columns at any time

```
/* When you get a fix for this, you can use
this to verify the columns are now persisted
*/

SELECT
    OBJECT_NAME(object_id) as TableName,
    name as column_name,
    definition,
    is_persisted
FROM sys.computed_columns;
GO
```

3. Tuning $VENDOR’s index maintenance

Index maintenance is killing you

It’s supposed to be a good thing

Instead, it’s running for 12 hours some nights

It inflates your log backups

It overlaps with your full backups and drags down your disk performance

And it causes massive blocking on tiny 8MB tables

Two parts to the change

Reschedule the job

Change the job settings

Remember: This job is distributed by your vendor, so you need to make sure they're aware of the changes and won't reset the job whenever an upgrade rolls out

Part 1: skip the small indexes

[Redacted Index information and size]

- Online rebuild took 98 minutes one night, 72 minutes the prior night

Online rebuilds in SQL Server 2012 require an exclusive lock at the end

Nightly jobs are having a lock war with the index maintenance

But why rebuild an index this small anyway?

Raise the page count

The job uses Ola Hallengren’s index maintenance

The default minimum @PageCountLevel is 1000

In an availability group like yours, that’s too aggressive

Recommendation for your job:

- Set @PageCountLevel=3000

Hey, now it won’t even touch that index it was blocking on!

Other parameters to tweak

REDACTED
(specific index maintenance customized for the client’s environment)

Solving the scheduling problem

REDACTED
(specific scheduling instructions customized for the client’s environment)

Your Action Plan

Your Action Plan is in a separate FILE

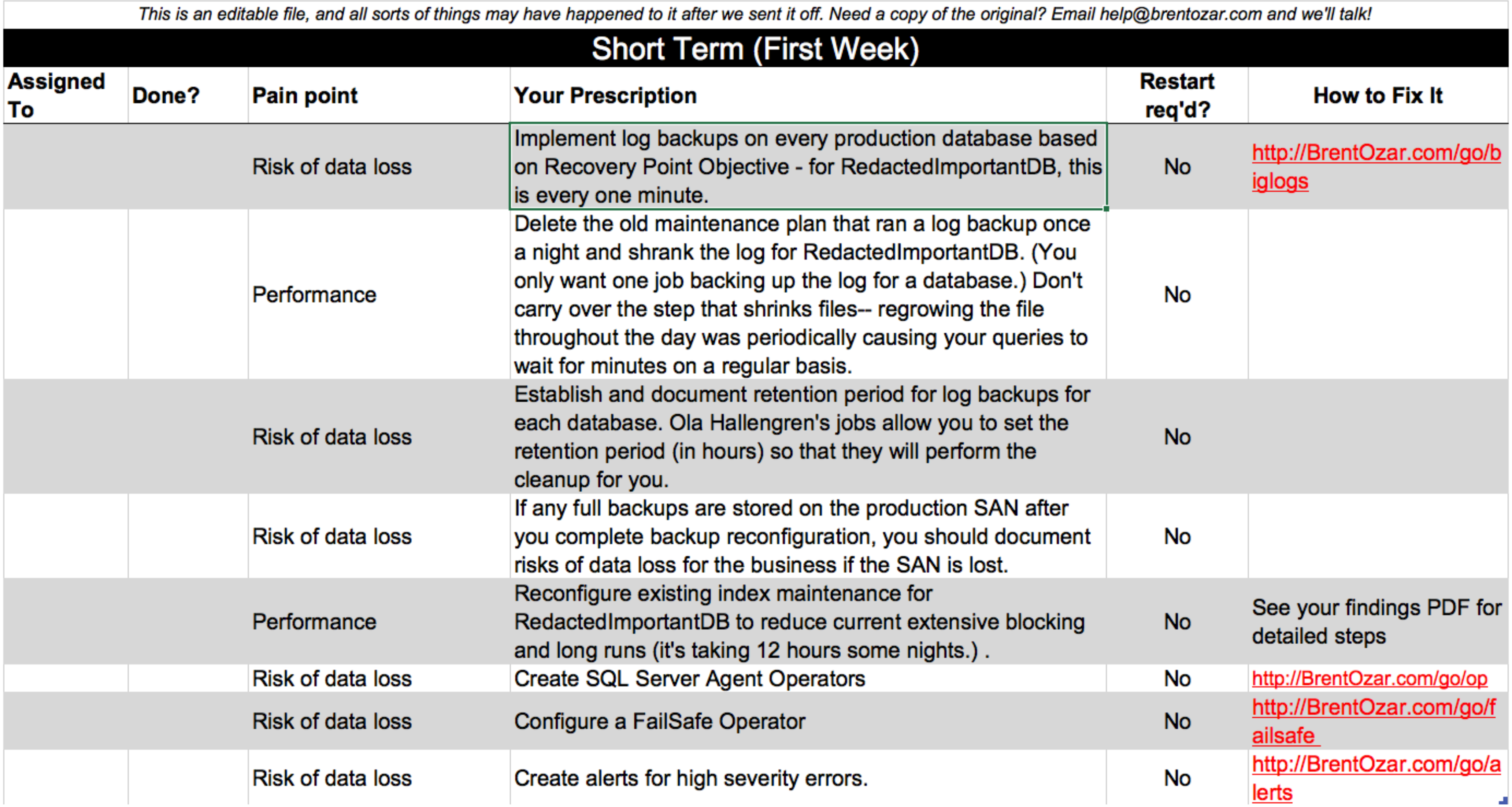
It’s in Excel so you can assign folks and edit it

We’re going to show screenshots of it here

If you’re reading this later and don’t have a copy, just email us at help@BrentOzar.com

(Don’t worry, we’ll make sure it’s really OK to send it to anyone who asks for it.)

The action plan doesn’t include AG architecture specifics, since you have to make the 1/2/3 choice first in order to build a project plan for that move.



This is an editable file, and all sorts of things may have happened to it after we sent it off. Need a copy of the original? Email help@brentozar.com and we'll talk!

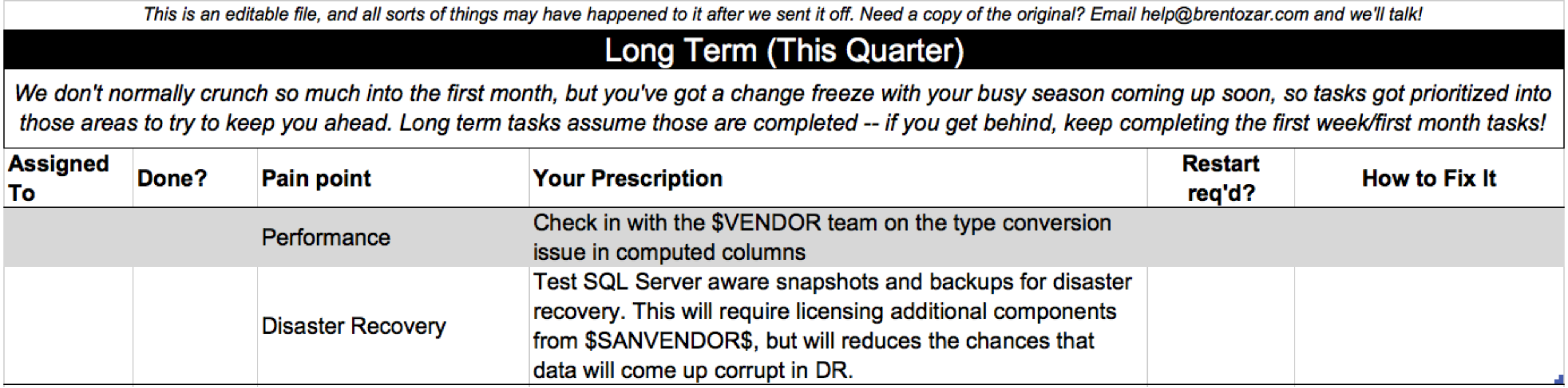
Short Term (First Week)

Assigned To	Done?	Pain point	Your Prescription	Restart req'd?	How to Fix It
		Risk of data loss	Implement log backups on every production database based on Recovery Point Objective - for RedactedImportantDB, this is every one minute.	No	http://BrentOzar.com/go/biglogs
		Performance	Delete the old maintenance plan that ran a log backup once a night and shrank the log for RedactedImportantDB. (You only want one job backing up the log for a database.) Don't carry over the step that shrinks files-- regrowing the file throughout the day was periodically causing your queries to wait for minutes on a regular basis.	No	
		Risk of data loss	Establish and document retention period for log backups for each database. Ola Hallengren's jobs allow you to set the retention period (in hours) so that they will perform the cleanup for you.	No	
		Risk of data loss	If any full backups are stored on the production SAN after you complete backup reconfiguration, you should document risks of data loss for the business if the SAN is lost.	No	
		Performance	Reconfigure existing index maintenance for RedactedImportantDB to reduce current extensive blocking and long runs (it's taking 12 hours some nights.) .	No	See your findings PDF for detailed steps
		Risk of data loss	Create SQL Server Agent Operators	No	http://BrentOzar.com/go/op
		Risk of data loss	Configure a FailSafe Operator	No	http://BrentOzar.com/go/failsafe
		Risk of data loss	Create alerts for high severity errors.	No	http://BrentOzar.com/go/alerts

This is an editable file, and all sorts of things may have happened to it after we sent it off. Need a copy of the original? Email help@brentozar.com and we'll talk!

Medium Term (This Month)

Assigned To	Done?	Pain point	Your Prescription	Restart req'd?	How to Fix It
		Performance	Fix the top five missing indexes on your SQL Server instance. Work with $VENDOR to test the new indexes in development. Safe off rollback scripts. Deploy to production.	No	
		Performance	Configure and enable the sp_delete_backuphistory job (from Ola Hallengren) to clean up backup history. If you never trim msdb's backup history, it will grow to epic proportions. It will slow down your backups. Avoid the tragedy.	No	
		Configuration and Management	Shrink and remove the second log file for the RedactedImportantDB database (there are two on the L drive)	No	
		Responding during a crisis.	Enable and practice using the Remote DAC. Practice connecting with this, seeing what is running, and documenting the waits. (In some kinds of extreme performance problems, a monitoring tool won't be able to observe activity, but you'll be able to connect with the DAC.)	No	http://Brentozar.com/go/dac
		Availability	Implement CHECKDB jobs. Check all the small databases nightly, check RedactedImportantDB once a week. Make sure CHECKDB jobs do not overlap with full backups or index maintenance.		
		Performance	Disable power savings in the BIOS of your production server	Yes (but can move the VM to other hosts)	http://brentozar.com/go/dellpower
		Performance	Set data files to grow in 256MB units for all databases. Set log files to grow at 128MB units.	No	http://www.brentozar.com/blitz/blitz-result-percent-growth-use/



This is an editable file, and all sorts of things may have happened to it after we sent it off. Need a copy of the original? Email help@brentozar.com and we'll talk!

Long Term (This Quarter)

We don't normally crunch so much into the first month, but you've got a change freeze with your busy season coming up soon, so tasks got prioritized into those areas to try to keep you ahead. Long term tasks assume those are completed -- if you get behind, keep completing the first week/first month tasks!

Assigned To	Done?	Pain point	Your Prescription	Restart req'd?	How to Fix It
		Performance	Check in with the $VENDOR team on the type conversion issue in computed columns		
		Disaster Recovery	Test SQL Server aware snapshots and backups for disaster recovery. This will require licensing additional components from $SANVENDOR$, but will reduces the chances that data will come up corrupt in DR.		

SPECIALIST'S PRESCRIPTION FOR RELIEF

During our SQL Critical Care® together this week, we came up with a set of recommended changes to your server.

We've compiled them into a handy prescription we're handing over in editable files that are friendly for markup and copy-paste.

SCRIPTS AND COMMANDS MAY NEED CUSTOMIZATION.
THEY WILL NEED TESTING.
THEY MAY CAUSE ITCHING AND DISCOMFORT.
EDITABLE FILES MAY BE EDITED AFTER WE SEND THEM TO YOU.

Read through all recommended steps carefully, understand what we're suggesting, and test them.

Ideally, measure your server to know that these changes helped, and consider backing them out if they don't help.

If time has passed and you want a fresh copy of those editable files, email help@brentozar.com and we'll ask you for a secret password.

We believe this is how to solve your pain point.

Which task will be the most difficult for you?