5 Deadly Data Management Sins

Posted at 13:19h

FROM: <https://thomaslarock.com/2017/01/5-deadly-data-management-sins/>

**[](https://thomaslarock.com/wp-content/uploads/2017/01/5.jpg?x98864)**All too often, we data professionals are our own nemesis when it comes to handling data and data management. Many data professionals and system administrators fail to recognize that the danger in our own habits increases the risk that the business will fall short of its goals. The danger may not be as destructive as an all-out data breach, but we are often to blame for enabling our business end users to lust after BIG DATA, resulting in data hoarding leading to [**ROT (Redundant, Outdated, Trivial information)**](http://whatis.techtarget.com/definition/ROT-redundant-outdated-trivial-information).

So, while the world’s collective media shine a light on the [**never-ending list of security breaches**](http://www.idtheftcenter.org/Data-Breaches/data-breaches.html), I suggest that there are actually more common, and dare I say, even bigger threats that data professionals need to guard against. These threats are lust, gluttony, greed, slothfulness, and pride. Not all data professionals are guilty of every one of these sins; rather, the collection of individuals that comprise modern enterprise IT shops is culpable. Let’s walk through examples for each data management – or rather, data mismanagement and sin.

**Lust**: Every company on the planet is turning its gaze towards the mythical creature called [**big data**](https://hbr.org/2012/10/big-data-the-management-revolution).  Companies want to collect as much data as possible in the hope that they will find something of value. Unfortunately, many of these companies have [**no idea how to implement their big data strategy**](http://www.huffingtonpost.com/young-entrepreneur-council/12-major-mistake-companie_b_6849616.html). The lust after this data, and the money and power that comes with that data, is gaining mainstream acceptance.

**Gluttony**: This is data hoarding, pure and simple, and is something that has been happening since, well, forever. Nobody thinks about archiving data; they only think about how one day, far in the future, they may need it again and when they do, they want it available immediately. The end result is [**companies carrying useless data forward year after year**](http://www.informationweek.com/big-data/big-data-analytics/big-data-no-hoarding-allowed/d/d-id/1279154).

**Greed**: We lust for more data, and we are gluttons to consume and store as much as possible. This leads to greed in the form of needing bigger and faster hardware to process all the extra data that lust and gluttony have resulted in. How many times have we [**thrown hardware at the problem**](http://www.solarwinds.com/faster-database/)?

**Slothfulness**: As data accumulates, our systems get slower. Queries run longer. Server tape backups take longer than a day. There is so much data that you may decide to have your database backups become less frequent, or non-existent, or shift the task to the server team who decides that SAN snapshots are good enough (PRO TIP: [**they aren’t**](http://sqlmag.com/blog/should-i-be-using-san-snapshots-backup-solution)). Before you know it, your disaster recovery plan isn’t adequate for your business needs, but you don’t usually find out until it’s too late.

**Pride**: Along with hubris, pride serves to undermine everything you have built by instilling a false sense of security with regards to your data. As you collect more data, your security concerns should be growing as well. Every day that you don’t end up on the front page of the newspaper is just another day you got lucky.

These sins are committed by every shop, across every team. Each data professional has committed one or more of them. And you cannot mitigate them by telling management that your development teams are agile, or that you are downloading and installing DevOps as a way of making everything better. No, these threats, these deadly data sins, are only fixed through a culture shift usually brought about by experience through some type of disaster.

Don’t wait for disaster to strike. If you find examples of these sins in your shop, take action now. Here are five things you can do starting today:

Define objectives

The very first thing you need is to define your [**recovery point objective (RPO)**](http://whatis.techtarget.com/definition/recovery-point-objective-RPO) and your [**recovery time objective (RTO)**](http://whatis.techtarget.com/definition/recovery-time-objective-RTO). RPO is the point in time to which you can recover data as part of an overall business continuity plan. In other words, it is the acceptable amount of data loss. RTO is the amount of time it will take for you to recover data before the business is severely impacted. Taking log backups every 15 minutes may help satisfy your RPO objective, but if it takes you hours to recover a 5TB database then you are probably not going to be helping your business continuity plans.

Build recovery plan

Notice I didn’t say to build a backup plan. No, you start with your recovery plan and work towards a backup plan that meets the recovery goals (such as the RPO and RTO objectives). When building out a recovery plan you need to think about the architecture you have for disaster recovery (DR) and for high availability (HA).This is also where I will provide a very important piece of information that you everyone needs to know: **HA IS NOT THE SAME AS DR**. For the developers that might stumble upon this blog I would explain it like this: HA <> DR

Define data archiving plan

Chances are you don’t have any archiving strategy in place. I know because we are data hoarders by nature, and only now starting to [**realize the horrors of such things**](https://www.youtube.com/watch?t=2&v=GAXLHM-1Psk). Archiving data implies less data, and less data means faster query performance. One way to get this done is to consider partitioning. Partitioning [**requires some work**](https://msdn.microsoft.com/en-us/library/ms190787.aspx) on your end, and it will increase your administrative overhead. Your backup and recovery strategy must also change to reflect the use of more files and filegroups. Another method would be to consider the use of [**Stretch Database**](https://azure.microsoft.com/en-us/services/sql-server-stretch-database/).

Make data security a priority

Instead of security being an afterthought, make it a priority. Ask yourself “What if that piece of data got loose? What’s the worst that can happen?” By asking yourself such questions you help yourself to understand that every piece of data needs to be treated as if it was the most important piece of data. Guard your data as if the future of your company depended on its privacy remaining intact. People seemed surprised that data theft continues to happen. I think part of the reason is because most security systems are designed and focused on preventing hackers from breaking in that they don’t understand the dangers of [**allowing data to simply walk away**](http://blog.infoadvisors.com/index.php/2013/01/22/federal-department-bans-use-of-portable-devices-yaff/) on something like a USB stick.

Know when to build or buy

We’ve all had those experiences where throwing money at the problem does not solve the performance issue. This is the result of not knowing the root cause of the issue. You don’t want to be the one to spend six figures on new hardware to solve an issue with query locking and blocking. Even after ordering the new hardware it takes time before arrival, installation, and the issue resolved. Learn about the [**things you can do first, before buying that hardware**](https://thomaslarock.com/2015/10/improve-database-performance-without-changing-code/).

Summary

When it comes to data, no one is perfect. These days data is easy to come by, making data a cheap commodity. But the reality is that data is not cheap, it is the most critical asset your company owns. When it comes to data management there are many ways to do things wrong. With just a little bit of effort you can make things better for yourself, your company, and help set the example for everyone else.