



Technical Whitepaper:

# 5 Signs You've Outgrown Your Data Warehouse

Mentioning the words “migration” or “database refactoring” to a typical DBA is unlikely to help you make new friends. Most organizations are extremely averse to changing their data infrastructure, which is often assumed to be a long, arduous and expensive process. Well, It doesn’t have to be this way, since with the right tools you can build a scalable big data infrastructure on AWS in just a week or two; still, change is always scary.

## **Avoiding your next infrastructure crisis**

Nevertheless, when change is inevitable, waiting for the last minute is not the best idea. If your data is growing (and it probably is), you’re eventually going to hit a brick wall where what you’re currently doing no longer works. How do you know when to move your organization from databases to a

big data architecture? If you don't want to wait until everything officially breaks and the entire company is freaking out, you should be on the lookout for these 5 telltale signs.

## **1. Data retention is becoming an issue**

Think of this scenario: Your enterprise data warehouse has been storing historical transaction data from the past year, serving a few dozen users across multiple use cases - and everything is running smoothly. Then your organization decides to also use the same servers to analyze thousands of daily events generated by traffic logs, and suddenly everyone is having a bad time. At this stage DBAs find themselves in the frustrating position of telling the higher-ups in the organization that they will need to either pay more or work with less data, all while juggling conflicting demands from engineering, analyst and business teams.

Eventually you'll either have to pay through the nose, or rely on endless hacks, workarounds and trade-offs around the data you retain in your database - constantly optimizing and A/B testing to strike the perfect balance between retention and scale. You spend hours trying to fit a square peg in a round hole, only to come up with a solution that is kind-of-okay.

If your database is constantly at capacity, it might be time to consider a change of strategy.

## **2. You are in ETL hell**

Joins and aggregations are often necessary for ETLs. Many companies prefer to perform them using an SQL database since it's easier than writing code. The pain starts at scale; these operations are resource-intensive. They are eating up your database's memory, so query performance degrades for business users and they aren't happy about it.

When you're dealing with billions of records, even joining two tables several times a day can really slow your database down, while a spike in volume of data can often cause ETL jobs to fail and cause downtime. Switching to a Big Data mindset means moving joins and aggregations outside the database. This allows you to mitigate the risk and leave the database for business users to query using the SQL they already know.

### **3. You're constantly resizing and reconfiguring your DB cluster**

Databases are expensive, streaming data is piling up, and you are under constant pressure to keep costs under control without sacrificing performance or versatility. Since nobody is enthusiastic about giving you carte blanche authority to purchase more server space, you're generally going to go for a fine balancing act of trying to compress or otherwise

reduce the overall size of the data, while adding additional machines as needed.

Of course, this is easier said than done, and adding another machine is never simply a matter of pulling a switch. Pretty soon you're spending half of your time reconfiguring and load-testing servers - while the other half is spent lobbying for larger IT budgets. If this is the case, it's prime time to consider a simpler way to store raw data, while figuring out how to analyze it later - in other words, adopt a data lake approach.

#### **4. The business expects better latency than you can deliver**

No one likes waiting hours for queries to return results, and when it comes to perishable insights, you could actually lose the ability to act on the data if you're not seeing the results in real-time. However, when queries are performed in batch

processes, there are limits to the amount of information you can deliver in this fashion. Usually latency will be at least hours. Put simply, if the business is expecting you to deliver data or analysis at latencies that you find impossibly low, you might be due for an architectural overhaul.

## **5. Infrastructure has become your sole focus**

There's nothing wrong with some good old fashioned sysadmin work; but it also shouldn't be all your DBA team is doing. In fact, a healthy DBA or DevOps organization should be spending most of its time on developing new applications, rather than maintaining a never-ending string of patchwork solutions and workarounds just to maintain the existing infrastructure. If your current strategy with data is preventing this and making everyone miserable in the process, it might just be time to look at an a product alternative that supports your scale without the hassle of

constant optimization.

*Convinced that there's a problem but still hesitant to take the plunge? Upsolver is the cure for your procrastination - a zero-coding, zero-effort solution to build your big data architecture on-premise or in the Cloud. [Start your free trial](#) to discover how you can move from databases to a cloud data lake in literally days.*