

Ghost in Your Machine

Preview text: **It's time to break free**

What's more frightening than Halloween? It's definitely: Data migration.

Data migration processes can be terrifying, and rightfully so.

Whether you're a company owner, manager, or developer, moving from an **SQL** database to a **NoSQL** database (or vice versa) is quite the undertaking.

That being considered, modern technology is dictating its terms. Thus, the fear of data migration is simply not a good enough reason for holding on to something which you've outgrown.



<https://media.giphy.com/media/RokPIX3C71piryApkp/giphy.gif>

Is Ditching SQL for NoSQL Worth it?

Usually, you don't decide to dump your **relational DB** in the spur of the moment. In most cases, your application and use cases naturally bring you to this point.

The need for greater flexibility — both in the scaling model and the data model— usually serves as motivation for this changeover.

Therefore, it's not a matter of worthiness, it's more about keeping up with current needs.

RDBMS are not Enough

Typical web apps are built on a three-tier architecture that includes data, logic, and presentation.

If you want to scale the application, you simply add commodity web servers to support more users. However, that's where there is a catch.

Relational databases don't have a flexible data model. This means that you won't get away with using commodity hardware in your scaling initiative.

Instead, you'll have to draw on immensely complex, big servers that guzzle up monstrous fees. Seems pretty bleak. Doesn't it?

It's Time

Below is some literature for how you decide on whether to free yourself from the shackles (or not):

1. Identify the key requirements for your application:

You will hit the jackpot with your NoSQL transitioning when:

- A. You need rapid app development due to the shifting market or data needs.
- B. You need scalability under notable user demand.
- C. You need constant performance with low response time and high throughput.
- D. You need operational reliability to handle failures.



<https://media.giphy.com/media/l2JdY5cEtIOdiRh3G/giphy.gif>

2. Study the various types of NoSQL offerings:

Document databases are not created equal.

In some cases, Couchbase might be the database you use to build responsive and flexible cloud, mobile, and **edge computing** applications that scale with ease.

3. Make the tough call:

The main difficulty is in understanding the vastness of NoSQL databases.

While each record in a SQL database complies with a **schema**, NoSQL DBs can have each document be composed with a unique structure.

Don't worry, you don't need any extra management tools to handle those changes to document schemas.

Besides, some NoSQL databases have added *more structural features* to bring the familiar tables and schema format to a NoSQL database.

Translation difficulties may include:

- Denormalization (repetition of data is acceptable in NoSQL).
- Sacrificing **ACID** to achieve scalability.

- No relationship between the tables (although some of the NoSQL DBs allow relationships between entities).

There are a lot of different ways to ETL aka transform data from a SQL database to a NoSQL database, hence you'll face some dilemmas when deciding on data import as well.

Don't Leave me

With all that *rambling on* about how difficult it is to migrate your data, consider one important point.

There may be many situations where you want to leave behind your SQL server and go to a new one.

However, there are still lots of use cases where a migration doesn't mean you abandon the old database. You can use multiple database systems to separate data belonging to different applications.

Migration is not the only option. You can abandon one or the other, or build a bridge between the two.

This is the part when you break free (from the dilemma).



<https://media.giphy.com/media/6pFZbo0YRP27e/giphy.gif>

Join us today to thank our newsletter sponsor, Couchbase. In an increasingly virtual world, companies operating on cloud services need to be able to scale fast

without any hiccups. If you are considering moving your distributed and elastic data sets over to a NoSQL database, Couchbase has the right environment for you. Couchbase is the architect's choice for NoSQL. Couchbase is an award-winning, distributed cloud database that offers reliable services critical for business applications.

Got a tech story to share with our readers? Everything you've ever wanted to know about how to get published on *Hacker Noon* - **[get it here](#)**