Couchbase #3

Which One is the One? Relational vs. Document Databases

Companies of all sizes rely heavily on <u>databases</u> to power their applications. But first, a *what*?

A database is a systematic collection of data that allows multiple users to maintain, update, and edit stored information. Think of it as a huge library *(the base)* with a huge collection of books *(the data)*.

When it comes to <u>picking the right database</u> for the back-end of your application, you end up at the fork in the road. And the big question is - *should I go for a* ***relational database or a document one***?

<u>http://gph.is/1WR5T0w</u>

<u>https://media.giphy.com/media/3orieKZtxf90bKHuww/giphy.gif</u>

Down the Memory Lane

Of the two, relational databases (RDBMS) are the old-timers. The first system sold as a SQL dates back to 1976. Needless to say, data has gone a long way since then: immense volumes of data and its increased complexity turned out above the pay grade of SQLs.

Therefore, fast forward to the early 21st century, NoSQL or document databases took precedence to offer a more flexible, scalable, and cost-friendly alternative.

So, what are the differences?

<u>https://gph.is/g/EGnoXd4</u>

https://media.giphy.com/media/jVZJsyEXEMPYBBJHLZ/giphy.gif

Data Structure

Relational databases are table-based. It means that in order to insert data into them, you have to define all the tables, columns, and fields with certain data types.

Document databases, on the other hand, don't store data as tables and columns. They store them as Jace on documents. It is a big plus for JS developers because they can work with Jason and JavaScript objects, basically full-stack. And when they get the data back from the database in a familiar format, there won't be a lot of transformation needed to be done .

Database query

Rumor has it that it is harder to query a document database compared to a relational one. However, it all boils down to a specific solution. For example, Couchbase database is based on a powerful nickel community, hence it helps you basically query your cheese on that.

At the same time, it is efficient and fast to query your data with a relational database. Also, SQL is a popular programming language that has been around for a long time. Therefore, queries can be run by business analysts or marketers, not necessarily developers.

To scale or not to scale

https://gph.is/g/EJ8pkDx

https://media.giphy.com/media/EyFH66Tv5MXbw6IXTP/giphy.gif

Achieving scalability and elasticity is a serious challenge for SQL databases. They just weren't meant for that. Relational databases are created to run on a single server, which means that to scale, you have to get a bigger, more complex, and more expensive hardware with larger processing power, memory, and storage.

However, they still allow for vertical scaling. It means you can ramp up the load on a single server by boosting things like CPU, RAM, or SSD.

On the other hand, document databases save you a pretty penny, since you can add capacity by scaling horizontally over affordable, commodity servers.

Development model

Document databases typically belong to the open-source community. The software is free, and most NoSQLs are free to use in commercial products. Relational databases are, for the most part, closed source with licensing fees baked into their product.

https://gph.is/g/EGNV9y4

https://media.giphy.com/media/dIBLtoII6axTGIGopo/giphy.gif

Migrate Data Into Your Database Painlessly

No matter which side you pick, data migration can sometimes be a pain. Let us introduce you to Couchbase. Couchbase is an award-winning distributed NoSQL cloud database. It empowers developers to build responsive and flexible cloud, mobile, and edge computing applications that scale effortlessly.

Have more questions? Head to the Hacker Noon Podcast to learn more about Document vs Relational Databases!

https://media.giphy.com/media/5UKlxhmcxiEoN2gHCq/giphy.gif

***

"Join our writing contest and get a chance to win cool prizes".

***

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**.