



NoSQL Databases - Introduction

In this lesson, we will get an insight into NoSQL databases and how they are different from Relational databases.

We'll cover the following



- What is a NoSQL database?
- How is a NoSQL database different from a relational database?
- Scalability
- Clustering

What is a NoSQL database?#

In this lesson, you will get an insight into NoSQL databases and how they are different from relational databases. As the name implies, *NoSQL* databases have no *SQL*, and they are more like *JSON-based* databases built for Web 2.0

They are built for high-frequency read and writes, typically required in social applications like Twitter, LIVE real-time sports apps, online massive multiplayer games, etc.

How is a NoSQL database different from a relational

database?#



Now, one obvious question that would pop up in our minds is:

Why the need for NoSQL databases when relational databases were doing fine, were battle-tested, were well adopted by the industry, and had no major persistence issues?

Scalability#

Well, one big limitation with *SQL* based relational databases is *scalability*. Scaling relational databases is not trivial. They have to be *Sharded* or *Replicated* to make them run smoothly on a cluster. In short, this requires careful thought and human intervention.

On the contrary, *NoSQL* databases have the ability to add new server nodes on the fly and continue the work without any human intervention, just with a snap of your fingers.

Today's websites need fast read-writes. There are millions, if not billions of users connected with each other on social networks.

A massive amount of data is generated every microsecond, and we needed an infrastructure designed to manage this exponential growth.

Clustering#

NoSQL databases are designed to run intelligently on clusters. When I say intelligently, I mean with minimal human intervention.

Today, the server nodes even have *self-healing* capabilities. This is pretty smooth. The infrastructure is intelligent enough to self-recover from faults.



However, all this innovation does not mean old-school relational databases aren't good enough, and we don't need them anymore.

Relational databases still work like a charm and are still in demand. They have a specific use case. We have already gone through this in *When to pick a relational database lesson*

(<https://www.educative.io/collection/page/6064040858091520/6411938009448448/6652931912761344>). Remember? 😊

Also, *NoSQL* databases had to sacrifice *Strong consistency*, *ACID Transactions*, and much more to scale horizontally over a cluster and across the data centers.

The data with *NoSQL* databases is more *eventually consistent* as opposed to being *strongly consistent*.

So, this obviously means *NoSQL* databases aren't the silver bullet. This is completely fine, we don't need silver bullets. We aren't hunting werewolves. We are up to a much harder task connecting the world online.

I'll talk about the underlying design of *NoSQL* databases in much more detail and why they have to sacrifice *strong consistency* and *transactions* in the upcoming lessons.

For now, let's focus on some of the features of *NoSQL* databases.

[← Back](#)[Next →](#)

When should you pick a relational dat...

Features of NoSQL Databases



Mark as Completed



Report an Issue

