When To Pick A NoSQL Database?

In this lesson, we will discover when to choose a NoSQL Database over any other kind of database.

We'll cover the following

Handling A Large Number Of Read Write Operations

Look towards *NoSQL* databases when you need to scale fast. And when do you generally need to scale fast?

When there are a large number of read-write operations on your website & when dealing with a large amount of data, *NoSQL* databases fit best in these scenarios. Since they have the ability to add nodes on the fly, they can handle more concurrent traffic & big amount of data with minimal latency.

Flexibility With Data Modeling

The second cue is during the initial phases of development when you are not sure about the data model, the database design, things are expected to change at a rapid pace. *NoSQL* databases offer us more flexibility.

Eventual Consistency Over Strong Consistency

It's preferable to pick *NoSQL* databases when it's OK for us to give up on *Strong consistency* and when we do not require *transactions*.

A good example of this is a social networking website like Twitter. When a

tweet of a celebrity blows up and everyone is liking and re-tweeting it from around the world. Does it matter if the count of *likes* goes up or down a bit for a short while?

The celebrity would definitely not care if instead of the actual 5 million 500 *likes*, the system shows the *like* count as 5 million 250 for a short while.

When a large application is deployed on hundreds of servers spread across the globe, the geographically distributed nodes take some time to reach a global consensus.

Until they reach a consensus, the value of the entity is inconsistent. The value of the entity eventually gets consistent after a short while. This is what *Eventual Consistency* is.

Though the inconsistency does not mean that there is any sort of data loss. It just means that the data takes a short while to travel across the globe via the internet cables under the ocean to reach a global consensus and become consistent.

We experience this behaviour all the time. Especially on YouTube. Often you would see a video with 10 views and 15 likes. How is this even possible?

It's not. The actual views are already more than the likes. It's just the count of views is inconsistent and takes a short while to get updated. I will discuss *Eventual consistency* in more detail further down the course.

Running Data Analytics

NoSQL databases also fit best for *data analytics* use cases, where we have to deal with an influx of massive amounts of data.

There are dedicated databases for use cases like this such as *Time-Series databases*, *Wide-Column*, *Document Oriented* etc. I'll talk about each of them further down the course.

Right now, let's have an insight into the performance comparison of SQL and NoSQL tech.