

Polyglot Persistence

This lesson explains the reasons why NoSQL is chosen by engineers-- it touches on the concept of Polyglot Persistence.

WE'LL COVER THE FOLLOWING



- Why Choose NoSQL? *
- Minimizing the Impedance Mismatch
- Embracing large scale data
- Conclusion

Why Choose NoSQL?

There are two main reasons why engineers choose NoSQL databases for their problems:

- Minimizing the Impedance Mismatch #

Since a lot of effort is spent on mapping in-memory to in-database, saving that effort improves developer productivity.

Sometimes, a NoSQL database has a data model that better suits the needs of our application, and thus simplifies the interaction of the application code with the database. This way, we have less code to develop and maintain.

For example, in *MEAN stack* (M is for Mongo DB) the whole stack uses JSON objects, and the interaction of application code and database is minimal.

- Embracing large scale data #

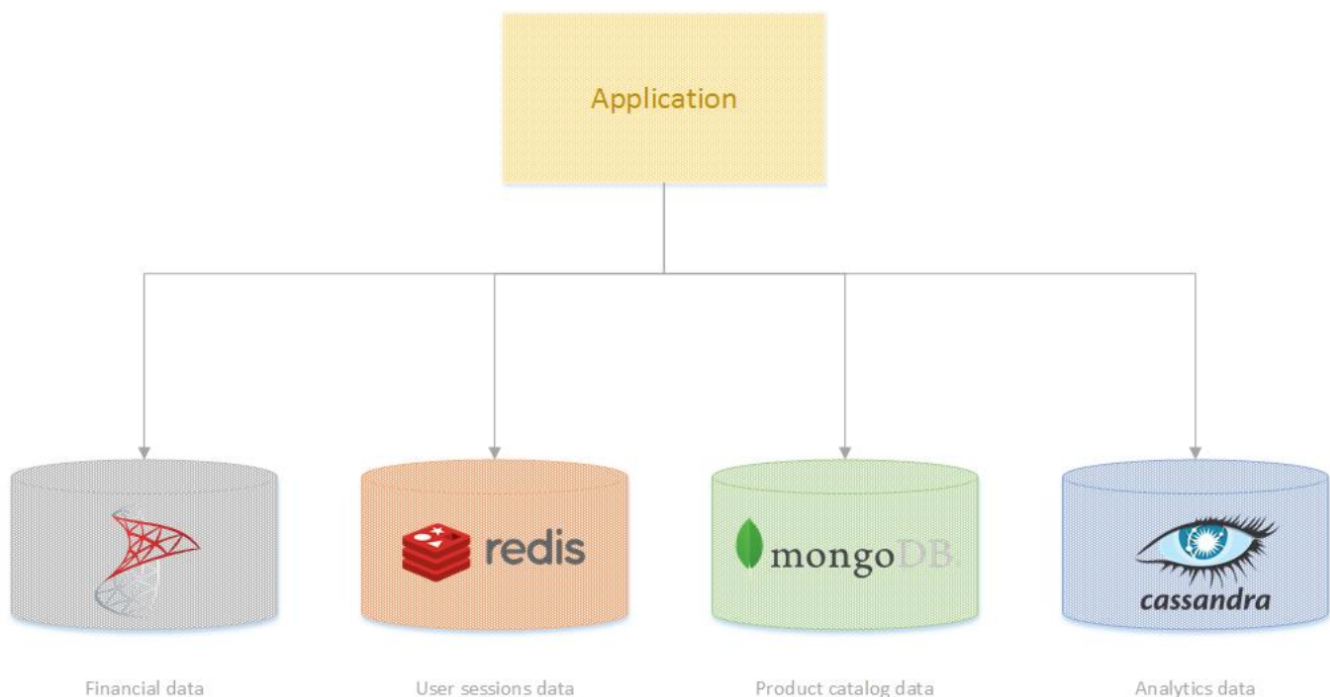
Today it is very time-consuming to store a large amount of data in the relational databases. Businesses have a need to capture and process a lot

of data more quickly because many NoSQL databases are designed to run on clusters, which are a better fit for this kind of problem.

Large scale clusters give us the ability to store larger data sets and process large amounts of analytic data. Also, NoSQL databases have different data models that may be better for processing huge amounts of data.

Does this mean that relational databases are dead?

No, not at all. The relational data model is still the best choice for a large number of problems out there. Additionally, relational databases have been here for decades, which means that there are many tools for them and people are familiar with them, unlike the fairly new concept of NoSQL database.



The only difference is the way that we should perceive relational databases--they are no longer the only option for data storage.

Now, we need to understand the nature of the data and use different data stores in different situations; Martin Flower calls this **Polyglot Persistence** – [NoSQL Distilled: A Brief Guide to the Emerging World of Polyglot Persistence](#).

This way of looking at data storage will lead us to solutions with multiple databases, each of which will be used for a different purpose. For example, we can use:

- *SQL* for a financial part of the application

- *MongoDB* for products catalog
- *Cassandra* for large-scale analytics

Conclusion

Polyglot Persistence has opened a new door in application development, never before have we had so many options and possibilities.

It has also raised a lot of questions. One of them being which database should be used in what situations.

This approach creates a lot of complexity, too, since there is no simple mechanism for maintaining data consistency. Maybe multi-model databases will fill this gap. Either way, it has not been this exciting in the database world since the mid-80s-- what a time to be alive!

You've learned about NoSQL and all of its types so far. In this course, you will be learning about MongoDB. In the next chapter, we will learn its basics.

Next is a short quiz for you to take. Best of luck!