Design:

**Table of Contents**

1. What is a NoSQL Database?
2. Types of NoSQL Databases
   1. Document-Based Database
   2. Key-Value Database
   3. Wide Column Based Database
   4. Graph-Based Database
3. Different NoSQL Databases
   1. MongoDB
   2. Cassandra
   3. ElasticSearch
   4. Amazon DynamoDB
   5. HBase

**What is a NoSQL Database?**

***So what is a NoSQL database?***

You might have heard people saying that a NoSQL Database is any non-relational database that doesn’t have any relationship between the data. Well, that’s not completely true. They can also store the relationship between the data but in a different way.

We can say that “NoSQL” stands for “Not Only SQL”. Here, data is not split into multiple tables, as it allows all the data that is related in any way possible, in a single data structure. When you work with a huge amount of data, you don’t need to worry about the performance lags when you query a NoSQL database. No need to run the expensive joins! They are highly scalable and reliable and designed to work in a distributed environment.

**Types of NoSQL Databases**

Now that we know what a NoSQL database is, let’s explore the different types of NoSQL databases in this section.

**1. Document-Based NoSQL Databases**

Document-based databases store the data in JSON objects. Each document has key-value pairs like structures:

The document-based databases are easy for developers as the document directly maps to the objects as JSON is a very common data format used by web developers. They are very flexible and allow us to modify the structure at any time.

Some examples of document-based databases are MongoDB, Orient DB, and BaseX.

**2. Key-Value Databases**

As the name suggests, it stores the data as key-value pairs. Here, keys and values can be anything like strings, integers, or even complex objects. They are highly partitionable and are the best in horizontal scaling. They can be really useful in session oriented applications where we try to capture the behavior of the customer in a particular session.

Some of the examples are DynamoDB, Redis, and Aerospike.

**3. Wide Column-Based Databases**

This database stores the data in records similar to any relational database but it has the ability to store very large numbers of dynamic columns. It groups the columns logically into column families.

For example, in a relational database, you have multiple tables but in a wide-column based database, instead of having multiple tables, we have multiple column families.

Here is a good resource to learn more about column-based databases:

Popular examples of these types of databases are Cassandra and HBase.

**4. Graph-Based Databases**

They store the data in the form of nodes and edges. The node part of the database stores information about the main entities like people, places, products, etc., and the edges part stores the relationships between them. These work best when you need to find out the relationship or pattern among your data points like a social network, recommendation engines, etc.

Some of the examples are Neo4j, Amazon Neptune, etc.

Now, let’s have a look at some of the NoSQL databases and their features.

**List of the Different NoSQL Databases**

**1. MongoDB**

MongoDB is the most widely used document-based database. It stores the documents in JSON objects.

According to the website stackshare.io, more than 3400 companies are using MongoDB in their tech stack. Uber, Google, eBay, Nokia, Coinbase are some of them.

**When to use MongoDB?**

1. In case you are planning to integrate hundreds of different data sources, the document-based model of MongoDB will be a great fit as it will provide a single unified view of the data
2. When you are expecting a lot of reads and write operations from your application but you do not care much about some of the data being lost in the server crash
3. You can use it to store clickstream data and use it for the customer behavioral analysis

If you want to start with MongoDB, I highly recommend going through the below articles:

* [MongoDB in Python Tutorial for Beginners](https://www.analyticsvidhya.com/blog/2020/02/mongodb-in-python-tutorial-for-beginners-using-pymongo/?utm_source=blog&utm_medium=NoSQL_Databases)
* [Query a MongoDB Database using PyMongo!](https://www.analyticsvidhya.com/blog/2020/08/query-a-mongodb-database-using-pymongo/?utm_source=blog&utm_medium=NoSQL_Databases)

**2. Cassandra**

Cassandra is an open-source, distributed database system that was initially built by Facebook (and motivated by Google’s Big Table). It is widely available and quite scalable. It can handle petabytes of information and thousands of concurrent requests per second.

Again, according to stackshare.io, more than 400 companies are using Cassandra in their tech stack. Facebook, Instagram, Netflix, Spotify, Coursera are some of them.

**When to use Cassandra?**

1. When your use case requires more writing operations than reading ones
2. In situations where you need more availability than consistency. For example, you can use it for social network websites but cannot use it for banking purposes
3. You require less number of joins and aggregations in your queries to the database
4. Health trackers, weather data, tracking of orders, and time series data are some good use cases where you can use Cassandra databases

**3. ElasticSearch**

This is also an open-source, distributed NoSQL database system. It is highly scalable and consistent. You can also call it as an **Analytics Engine**. It can easily analyze, store, and search huge volumes of data.

If the full-text search is a part of your use case, ElasticSearch will be the best fit for your tech stack. It even allows search with fuzzy matching.

More than 3000 companies are using Elasticsearch in their tech stack, including Slack, Udemy, Medium, and Stackoverflow.

**When to use ElasticSearch?**

1. If your use case requires a full-text search, Elasticsearch will be the best fit
2. If your use case involves chatbots where these bots resolve most of the queries, such as when a person types something there are high chances of spelling mistakes. You can make use of the in-built fuzzy matching practices of the ElasticSearch
3. Also, ElasticSearch is useful in storing logs data and analyzing it

**4. Amazon DynamoDB**

It is a key-value pair based distributed database system created by Amazon and is highly scalable. But unfortunately, it is not open-source. It can easily handle 10 trillion requests per day so you can see why!

More than 700 companies are using DynamoDB in their tech stack including Snapchat, Lyft, and Samsung.

**When to use DynamoDB?**

* 1. In case you are looking for a database that can handle simple key-value queries but those queries are very large in number
  2. In case you are working with OLTP workload like online ticket booking or banking where the data needs to be highly consistent

**5. HBase**

It is also an open-source highly scalable distributive database system. HBase was written in JAVA and runs on top of the Hadoop Distributed File System (HDFS).

More than 70 companies are using Hbase in their tech stack, such as Hike, Pinterest, and HubSpot.

**When to use HBase?**

1. You should have at least petabytes of data to be processed. If your data volume is small, then you will not get the desired results
2. If your use case requires random and real-time access to the data, then HBase will be the appropriate option
3. If you want to easily store real-time messages for billions of people