NoSQL Databases:

* Designed for low-latency analytics over semi-structured data
* Variety of data models
  + Key-value:
    - Uses individual or combination of keys to retrieve associated values
  + Document:
    - Uses flexible documents to store data in field-value pairs
  + Graph:
    - Data management designed to handle very large sets of structured, semi-structured or unstructured data
  + Optimized for performance and scale
* Horizontal Scalability (NoSQL) vs Vertical Scalability (SQL)
  + Vertical - “Scaling up”
    - Adding more computing power to existing instances/nodes
    - If your server requires more processing power, vertical scaling would mean upgrading the CPUs
    - Vertical scaling may also describe replacing a server entirely or moving a server’s workload to an upgraded one.
  + Horizontal - “Scaling out”
    - Adding additional nodes or machines to your infrastructure to cope with new demands.
    - If you are hosting an application on a server and find that it no longer has the capacity or capabilities to handle traffic, adding a server may be your solution.
    - It is quite similar to delegating workload among several employees instead of one.
  + NoSQL excels in low-latency data processing that goes beyond a single instance
* Scaling
  + NoSQL databases are partitionable, meaning that access patters are able to scale out by using distributed architecture to increase throughput.
    - TL;DR: Consistent performance due to horizontal scalability
* Performance
  + Depends on hardware, network latency, and calling application
* APIs
  + Object-based APIs allow the ease-of-access when it comes to storing and retrieving data.

<https://dynobase.dev/dynamodb-advantages-and-disadvantages/>

https://aws.amazon.com/nosql/