

Summary of the selected topic: NoSQL

The advancements of modern day web applications has increasingly demanded databases to be more flexible, scalable, and consistent with high volumes of data. The argument of SQL vs NoSQL has brought emerging arguments over the internet, mainly issues such as relational versus non-relational, structured versus unstructured, uniformed versus non-uniformity, and scalability versus non-scalability. The creation of NoSQL databases surfaced as an important substitute to relational databases for greater benefits such as tolerance, availability, and scalability.

NoSQL, defined as “Not only SQL”. It's a type of non-traditional database that supports horizontal scalability, efficient storage, and retrieval of data with availability as the greatest purpose. The importance of NoSQL is primarily based on a non-strict schema approach, big data efficiency, and ability to handle failures independently.

NoSQL doesn't require strict schemas for database transactions. In traditional RDBMS, it requires schemas to be defined for its foundation. When users insert new data, change data types, or redefine the schema, it brings more difficulties on top of difficulties managing the database. As for NoSQL, defining a schema is optional. This allows users to store data of many structures within the same database, allowing greater flexibility, tolerance, and compliance.

NoSQL provides high scaling abilities. In traditional RDBMS, the general row/table/column approach limits the practice of RDBMS. As RDBMS increases in data storage, upgrading the hardware of the system is needed. However, NoSQL follows the distribution environment approach, the idea that increases in data storage are shared among different machines and servers. This allows users to store data across the scale of servers automatically, supporting high-level data distribution.

NoSQL handles server failures independently. NoSQL offers horizontal scalability, managing databases is as simple as adding nodes to its clusters. In cases of data distribution failures, the failure is independent from other servers. When one server fails, other servers are in working conditions, continuing in default nodes. NoSQL guarantees that a single point of failure doesn't affect other servers, supporting continuous data storage and consistent databases.

Popular NoSQL databases include MongoDB, Cassandra, Redis, Hypertable, CouchDB, and Oracle.

Gupta, A., Tyagi, S., Panwar, N., Sachdeva, S., & Saxena, U. (2017). *NoSQL databases: Critical analysis and comparison*. <https://doi.org/10.1109/ic3tsn.2017.8284494>

**Sethi, B., Mishra, S., & Patnaik, P. K. (2014). A Study of NoSQL Database. *International Journal of Engineering Research and Technology*, 3(4).
<https://www.ijert.org/research/a-study-of-nosql-database-IJERTV3IS041265.pdf>**