|  |  |
| --- | --- |
|  | Implications of Selecting NoSQL vs SQL Database |
|  |  |
|  |  |

Natalia Palej  
A00279259  
Software Design with Artificial Intelligence for Cloud Computing  
Year 4

# Introduction

In today’s world, the amount of data we generate is unbelievably huge. This was one of the reasons why the way we store it has evolved. For a very long time, relational databases (SQL) have been the go-to solution. It offered structure, reliability and well-established approach to organizing data. However, with current explosion of data and demands for scalability, speed and flexibility – NoSQL databases have become more popular.

This essay explores implications of choosing NoSQL Database over a traditional Relational one (SQL). I will go through advantages and challenges of both approaches and see how they compare in real-world scenarios. I will not answer questions like “which one is better?”, as ultimately the question is which one fits the specific needs of a given project or business.

# Background

# Relational (SQL) Approaches

# NoSQL Approaches

# Comparing SQL vs NoSQL

# Conclusion

# References

[1] Burtica, R.; Mocanu, E.M.; Andreica, M.I.; Tapus, N., "Practical application and evaluation of no-SQL databases in Cloud Computing," Systems Conference (SysCon), 2012 IEEE International , vol., no., pp.1,6, 19-22 March 2012

[2] Boicea, A.; Radulescu, F.; Agapin, L.I., "MongoDB vs Oracle -- Database Comparison," Emerging Intelligent Data and Web Technologies (EIDWT), 2012 Third International Conference on , vol., no., pp.330,335, 19-21 Sept. 2012

Relational Database is a technology that allow users to organize their data using a structured, well-defined model. Many relational database systems have the option of using Structured Query Language (SQL) to query and maintain the database. On the other hand, NoSQL are often faster and can easily handle massive amount of data. It integrates particularly well with applications that support real-time analytics, website personalization, mobile apps. But the question is, what are the implications of selecting NoSQL vs SQL? Should we even make a choice which database type is better?

Relational Database Model, in fact, is called the RDBMS database (Relational Database Management System). Indeed, the name SQL is a simplification because databases of this type use SQL (Structured Query Language). Most IT fanatics have started their database journey with it. SQL has certain structure that must be followed hence why user should get familiar with schema model to determine the pattern of the data. General use of RBDMS is to execute CRUD functionality (Create, Read, Update, Delete). Primary Key must be provided to uniquely identify the rows of the tables and these can be joined together by introducing Foreign Key. For instance, an application could create a new user and insert it into database. It could read existing users, work with the data, update or delete the information. Its main feature and advantage is manipulating data. RDBMS is the most widely used database technology. Its data files are well-structured, kept in a relational manner with maximum efficiency when fetching and are excellent for complex queries. It has proven reliability in the past 50+ years, it is very easy to learn and simple to use, has great compatibility, portability, high speed and enormous dataset processing. As we can notice technologies are changing but SQL does not fade away which definitely makes it so dominant. We cannot forget that at the same time it can be very restrictive due to its predefined setup.

What is NoSQL? NoSQL Database (Not Only SQL) can create the impression of something negative. The acronym was used to define non-relational databases. It is a database type that accumulates information in JSON format instead of tables. NoSQL term refers to an unidentified set of databases that are open source and mostly not using SQL language. Many people wonder, why was NoSQL created and why do we need it if SQL works perfectly up to date. One of the main reasons is scalability. NoSQL has ability to effectively cope with a fluctuating load. Horizontal scaling is more flexible than vertical, which means there is no need to replace servers. High prices and difficulty in determining the required licenses in relational database are off putting. NoSQL offers free and open source software which encourages users. Flexibility - appropriate databases to the relevant needs. Many clustered servers allow availability, even if one of the servers fails. Therefore, NoSQL databases are built to be flexible, scalable, and capable of rapidly responding to the data management demands of modern businesses with high availability. The mentioned above made NoSQL evolve so fast and become popular within IT world.

# Comparing Relational and NoSQL

Relational databases depend on tables, columns, rows, schemas to arrange and retrieve data. However, NoSQL do not rely on that layout, it uses flexible data models. NoSQL databases have been adopted by recognised enterprises whereas RDBMS failed to meet the performance, scalability and flexibility. Most SQL databases are upscaled (vertically) which means you can add components like RAM, CPU etc. On the other hand, NoSQL databases are designed for horizontal scaling which means adding more machines. Relational databases are not designed to run in clusters. However, the division of these is possible but is classified as "unnatural activity".

There are four categories of NoSQL database types which have been formed to support specific needs and use cases. In Key-Value stores model, the data is represented as collections of key-value pairs and each key can appear only once. The application has complete control over what is stored in the value, making this the most flexible NoSQL model.

Wide Column stores is a two-dimensional map where each key has one or more key-value pairs. It is possible to store and use large data. The column family store data in tables similar to RDBMS but names and formats of columns can vary from row to row across the table.

Document Databases store documents in one of the standard formats (XML, YAML, JSON, BSON) and consist of semi-structured data. A value is a single document that stores all data related to a specific key. Joins or relations are not supported.

Graph Databases type is intended for data whose relationships are well presented as a graph and where number of relations is finite (e.g. road maps, public transport connections). The graph is based on nodes (entities), edges (relations), properties and graph theory.

We can distinguish two different transaction models for databases. ACID is an acronym for the four relational database properties which stands for Atomicity, Consistency, Isolation and Durability. The above ensures accuracy, completeness, and data integrity. Atomicity states the indivisibility of transaction, which means transaction cannot be left partially completed - each transaction will either be carried out in full or not at all. Consistency means that after the transaction is completed, the system will be consistent and integrity rules will not be violated. Isolation means that if one or more transactions are being executed at the same time and in parallel, all the transactions will be carried out and executed as if it is the only one in the system. Finally, Durability means that the system is able to boot up and provide consistent, intact, up-to-date recorded and committed transaction data, for example after a sudden power failure or restart.

The BASE describes properties of NoSQL Databases. It stands for Basically Available, Soft state, Eventual consistency. Breaking it down further, Basically Available indicated that the system offers availability most of the time, even if partial layer fails. Soft state means that the system does not have to be write-consistent or mutually consistent and the state of the system could change over time. Eventual consistency indicates that the system will eventually become consistent, given that the system stops receiving input. NoSQL databases give up the A, C and/or D requirements, and in return they improve scalability. In short, the ACID model provides a consistent system where BASE is all about high availability.

# Conclusion

Instead of looking at NoSQL database as an alternative technology or competition for Relational Database Model we should open up our minds and identify how well they complement each other. When combined, they are not only outstanding but also very powerful. It is worth mentioning, that we have been using both of them in parallel for years. NoSQL key-value databases are used as a cache layer for relational SQL databases. To give a clear example, Redis and Memcached which are using the possibility of smooth horizontal scaling and high performance, support data processing in various types of applications which require real-time processing and high availability. Let’s go back to our question, which database should be choose and which one is better? The answer is only one: it depends. NoSQL database works faster and is horizontally scalable, which allows you to work with big data. Relational Database provides greater data integrity (schema) and relationships. Depending on the project, you may need either of them and neither of them is better than the other.

References