Mertan I. Rasim

ICT & Software Engineering

Fontys University of Applied Sciences

November 2, 2022

- Main topic: What is the best way to store data in my project?
- Main question: What are the key factors, determining which technology would be the most suitable to use for storing data in my project?
- Sub questions:
    1. What are the main priorities of the project?
    2. What are the available methods to store data?
    3. What are the advantages and disadvantages of each technology?
    4. What use cases for my data is there?
- DOT Methods chosen:
    1. Question 1 & 4:
        - Explore user requirements – to directly define the priorities of the project
        - Domain modeling – to visualize the project
    1. Question 2 & 3:
        - Literature study – to understand how the technologies work
        - SWOT analysis – to research the advantages and disadvantages

Contents

What are the main priorities of the project?
The project's goal in question is to provide an easy interface for users to look up information about available bus lines. Simplicity, data accuracy, and average performance (at worst) are key priorities the project requires.

What are the available methods to store data?
Contemporary database technologies are divided into two categories: Relational databases and non-relational databases, also known as "NoSQL". The core difference between the two technologies is that in a relational database, users access data by stating what they want, but in a navigational database, data is accessed by defining the path to find the desired data.

What are the advantages and disadvantages of each technology?
NoSQL databases are often regarded as an upgrade to the standard relational model of database systems, although with few, but important caveats. According to research (Karim, Siddiqa, & Gani, FITEE.1500441, 2017), there are six key features that contrast their differences:

- Horizontal scaling, or in other words adding more nodes to the system.
- The ability to partition/distribute data over many servers.
- A simple call level interface or protocol (in contrast to a SQL binding).
- A weaker concurrency model than the ACID transactions of most relational (SQL) database systems.
- Efficient use of distributed indexes and RAM for data storage.
- The ability to dynamically add new attributes to data records.

(Karim, Siddiqa, & Gani, FITEE.1500441, 2017)

	Traditional database systems	**Big data storage systems**
Strengths	Support highly structured data stored and processed over an auxiliary server Vertical scalability with extendible processing on a server Specialized data manipulation languages Specialized schema	Support heterogeneous structured data Horizontal scalability with extendible commodity servers Support data-intensive applications Simultaneous accessibility Reliability and high availability High fault tolerance Eventual consistency
Weaknesses	Performance bottleneck Processing delays Increased deadlocks with growth of data Limited storage and processing capacity Co-relations which hinder scalability	No compliance with ACID due to scalability and performance

	Expensive join operations for multidimensional data	
Opportunities	Support complex queries Atomicity in complex transactions Built-in deployment support	Improved query response times Simplicity in storage structures Data-intensive
Threats	Extensive volume of data for storage with dynamic growth Frequently changing schema Complex data structures More concurrent access needs Frequent I/O needs Real-time processing needs Consistency of a large number of storage servers	Large number of small files Deployment may need community support

The weaknesses of non-relational systems are that they are relatively new and not as battle tested as their counterpart. The vast support for relational databases confirms this, as companies such as Microsoft, Amazon, & Meta use both SQL and NoSQL systems. Popular database engines such as Oracle, MySQL, MSSQL, & PostgreSQL hold the top positions of popularity. (DB-Engines, 2022) (Nance, Iype, Losser, & Harmon, 2013)

The advantages and disadvantages of SQL and NoSQL can be summarized in the following table:

	SQL	**NoSQL**
Advantages	ACID compliance Data accuracy Normalization Simplicity	Flexibility Scalability Performance Database administrators are not required Hardware failure handling in some engines Has evolved at a very high pace
Disadvantages	Scalability Flexibility Performance	Immature No standard query language Some NOSQL databases are not ACID compliant No standard interface Maintenance is difficult

(Nayak, Poriya, & Poojary, 2013)

What use cases for my data is there?

The project plan specifies the user stories and their acceptance criteria, which elaborate on what use cases to explore.

#	User story	Acceptance criteria
1	As a user, I would like to view the available bus lines in a city in a user-friendly manner to choose the one I want more information about.	I can see the available bus lines
		I can choose any bus line I would like to view
2	As a user, I would like to view the timetables of a bus line I chose to plan my schedule better.	I can see the bus stops and the time a bus will arrive
3	As a user, I would like to view what bus lines arrive on a bus stop I specified so I plan my schedule better.	I type the name of the bus stop and choose it
		I see the bus lines that stop there and the time when they will arrive
4	As a user, I would like to view visually on a map where the bus stops are located for convenience.	I can see the map of the city
		I can see the bus routes for every bus line or for a specific one
5	As a user, I would like to set my default route to show up when I view the website for convenience.	The set route shows up every time I open the website if I am logged in
6	As a user, I would like to know what the price and duration of a ticket for a is given bus line for convenience.	I can see the price of a ticket for a given bus line
		I can see the duration of the ticket
7	As a stakeholder, I would like users to be able to create accounts to save their preferences.	A user can create an account from a specific page
		A user can login to their account
8	As a stakeholder, I would like administrators to be able to create, edit and delete information.	An administrator can login to an account they have been given from the product owner.
		An administrator can CRUD data such as: bus lines, bus stops, routes, ticket information, and user accounts.
9	As a user, I would like to be able to save a route as a picture and have it sent to my email, for convenience.	I can choose the bus line and route, then choose the option for the route to be sent to my email as a picture.
10	As a user, I would like to be able to view my default route	I can see my default route when I visit the website.

Along with the C4 diagram, we can see that the model requires simple read/write operations regarding bus lines, routes, users, tickets, buses, and their coordinates. Fetching and storing information are not time-sensitive, except for saving coordinates of the active buses. The

benefit of NoSQL database would help for frequent reads and writes for that specific use-case.

Conclusion & Recommendation

Overall, both relational and non-relational databases have their specific use cases. Starting with project's main priorities, simplicity, data accuracy and average performance are within the project's priority scope. With the help of the SWOT analysis, the advantages and disadvantages were compared between relational and non-relational systems. Along with the analysis of user requirements, we can conclude that the project would benefit more from a relational database due to being simpler and ACID compliant. From that we can deduce that high performance and speed can be a second priority and focus more on data accuracy and simplicity. Additionally, relational databases have been tested for a longer period than the novel NoSQL technology, which would benefit the project security and consistency-wise.

The benefits of non-relational database would be applicable if the project was designed to handle "Big data" or be used by a larger group of users than, for example, the population of a city.

References

Cattell, R. (2011). Scalable SQL and NoSQL data stores. *ACM SIGMOD Record*, 16.

DB-Engines. (2022, October). *DB-Engines Ranking*. Retrieved from db-engines.com: https://db-engines.com/en/ranking

Karim, A., Gani, A., & Siddiqa, A. (2017). Big data storage technologies: a survey. *Frontiers of Information Technology & Electronic Engineering*, 31.

Karim, A., Siddiqa, A., & Gani, A. (2017, 09 22). *FITEE.1500441*. Retrieved from Springer: https://link.springer.com/article/10.1631/FITEE.1500441/tables/1

Nance, C., Iype, R., Losser, T., & Harmon, G. (2013). NoSQL VS RDBMS - Why there is room for both. *Association for Information Systems*, 7.

Nayak, A., Poriya, A., & Poojary, D. (2013). Type of NOSQL Databases and its Comparison with Relational databases. *International Journal of Applied Information Systems (IJAIS)*, 4.