CHALLENGES IN NON-RELATIONAL DATABASE DEVELOPMENT

Knysh Ludmyla

English teacher

Shnurok Vladyslav

Student

Ivano-Frankivsk National Technical University of Oil and Gas

Ukraine, Ivano-Frankivsk

vladyslav.shnurok-ip232@nung.edu.ua

INTRODUCTION. The rapid advancement of information technologies and the exponential growth in data volumes have necessitated new approaches to data storage and processing. As organizations generate vast amounts of data through social media, IoT devices, e-commerce, and other digital interactions, the limitations of traditional relational databases (RDBs) become increasingly apparent.

While RDBs have been reliable for managing structured data, they often struggle to meet the scalability and flexibility requirements of modern applications. Their fixed schema necessitates predefined data structures, making it difficult to accommodate the dynamic nature of today's data, which can include unstructured and semi-structured formats.

In contrast, non-relational databases, or NoSQL databases, have emerged as flexible alternatives that address these limitations. NoSQL databases are designed to handle diverse data types and support rapid scalability.

This report examines the challenges developers encounter when creating and implementing non-relational databases, focusing on key aspects such as data consistency, high availability, distributed data management, and optimal architectural choices. By analyzing these challenges, we aim to clarify when non-relational databases are the most suitable option compared to traditional relational systems, ultimately providing insights for effective data management strategies.

AIM. The aim of this report is to investigate and outline the challenges encountered in the development and implementation of non-relational databases.

MATERIALS AND METHODS. Modern demands for storing and processing large data volumes are driving the development of new approaches and technologies. Traditional relational databases (RDBs) remain important due to their structured design, SQL query capabilities, and robust support for transactions. However, with the rise of Big Data and new business requirements, certain limitations are revealed by relational systems. For instance, RDBs face difficulties with horizontal scaling, which can reduce their efficiency in large-scale systems. Moreover, the rigid structure of relational data requires predefined formats, which may not suit applications with frequently changing structures.

RESULTS AND DISCUSSION. During the development of non-relational databases, several challenges arise. One of these is the necessity to balance consistency, availability, and partition tolerance, as outlined by the CAP theorem. Maintaining consistency is particularly challenging, as non-relational databases often follow the BASE (Basic Availability, Soft-state, Eventually Consistent) principle instead of ACID(Atomicity, Consistency, Isolation, Durability), which creates additional difficulties in upholding data integrity. Scaling and data management also require significant resources: while non-relational databases scale well, distributed data management demands substantial effort. Security is another key concern, as limited transactional security in non-relational databases necessitates additional protective measures.

Despite their advantages in flexibility, scalability, and handling diverse data types, the development of non-relational systems presents ongoing challenges. Specifically, choices must be made between consistency, availability, and partition tolerance (the CAP theorem), while also ensuring consistency and security, since non-relational databases often rely on BASE rather than ACID. Although non-relational databases are ideal for high-load, modern applications, the choice between them and relational databases depends on the task.

CONCLUSIONS. The choice between relational and non-relational databases depends on the project's specific needs: relational databases remain essential for complex transactions, while non-relational databases suit adaptable, scalable systems where fast data processing and flexibility are essential. Understanding each model's characteristics enables better decisions in database architecture, ensuring system efficiency.

REFERENCES

1. GeeksforGeeks. Non-Relational databases and their types - geeksforgeeks. *GeeksforGeeks*. URL: https://www.geeksforgeeks.org/non-relational-databases-and-their-types/ (date of access: 03.11.2024).
2. Non-Relational database vs relational: 10 key differences. *Atlan | Third-Gen Data Catalog*. URL: https://atlan.com/non-relational-database-vs-relational/ (date of access: 03.11.2024).
3. Relational vs. non-relational databases - datasciencecentral.com. *Data Science Central*. URL: https://www.datasciencecentral.com/decoding-different-types-of-databases-a-comparison/ (date of Access: 03.11.2024).
4. Relational vs nonrelational databases - difference between types of databases - AWS. *Amazon Web Services, Inc.* URL: https://aws.amazon.com/compare/the-difference-between-relational-and-non-relational-databases/ (date of access: 03.11.2024).
5. What's the difference? Relational vs non-relational databases. *insightsoftware*. URL: https://insightsoftware.com/blog/whats-the-difference-relational-vs-non-relational-databases/ (date of access: 03.11.2024).