CENTRO UNIVERSITÁRIO DE BRASÍLIA

Faculty of Technology and Applied Sciences (FATECS)
COURSE: COMPUTER SCIENCE
Subject: Database II

BRUNO CÉSAR GONÇALVES MEIRELES
FELIPE RODRIGUES QUEIROZ
GABRIEL DE MEDEIROS LIMA

NoSQL DATABASES

Brasília/DF, April and May of 2024

INTRODUCTION

This academic essay aims to present and analyze what are NoSQL Databases and their various types, as well as their advantages and disadvantages. NoSQL Databases (Not Only SQL) are non-relational databases, meaning they differ from traditional relational databases that support SQL, as they do not use a fixed table schema. Thus, NoSQL databases can store and process unstructured or semi-structured data of various types on a large scale, offering greater flexibility and scalability in handling data.

In this context, this essay uses references from major companies in the field worldwide, as well as study materials like books and activities we, as students and authors of this essay, have access to. This topic is significant for reflection, as it has been gaining relevance and prominence in the tech world over the years, providing viable alternatives for solving problems or developing projects.

DEVELOPMENT

1.1 Introduction

This chapter focuses on the main categories of existing NoSQL databases and their characteristics, along with the advantages and disadvantages of each. Finally, a technical comparison between MongoDB and OracleDB will be presented, highlighting architectural details, benefits, limitations, and cost-effectiveness of each selected structure.

1.2 Categories of NoSQL Databases and Their Characteristics

NoSQL Databases have increasingly gained popularity in the market, offering freedom, flexibility, and efficiency for developers managing projects worldwide. However, there are four main categories of NoSQL databases: Document Databases, Key-Value Databases, Column-Oriented Databases and Graph-Oriented Databases. Each one of these categories will be described below.

1.2.1 Document Databases

In this type of database, data is stored in documents using the JSON format, providing significant flexibility and ease of evolution. Additionally, it reduces the number of conversions required when using this data in applications, as the data is encapsulated in a single document, resulting in faster queries.

Document-oriented databases are often used for content management and catalog information storage, with MongoDB being the most renowned example of this category.

1.2.2 Key-Value Databases

This type of database consists in storing data in sets of key-value pairs, where the key acts as a unique identifier for the associated value. Both the key and the value can take various forms. This category is primarily characterized by high-speed data access and horizontal scalability.

It is often employed in session management for web applications and caching storage, with Amazon DynamoDB being the most recognized option in the market.

1.2.3 Graph-Oriented Databases

Graph-oriented databases use graph theory for data storage and processing. This model is recommended for querying highly interconnected data and is characterized by its efficiency and flexibility. Performance significantly improves when handling large datasets and connections.

These graph databases are commonly applied in contexts such as social networks and recommendation engines, with Neo4j being the most popular option in this category.

1.2.4 Column-Oriented Databases

Column-oriented databases primarily utilize block iteration. This means the database does not work line-by-line to return a result. Instead, it uses only the elements from the desired column with a single CPU instruction.

Generally, Column-oriented databases are optimized for data reading due to their block iteration, as opposed to row-by-row processes. The most famous example in this category is Apache Cassandra.

1.3 Advantages and disadvantages of each NoSQL Database Category

1.3.1 Document Databases

Document databases facilitate development by allowing users to create and update documents directly from the code without affecting other documents within the same collection, due to their "schema-free" nature.

1.3.2 Key-Value Databases

Key-value databases are particularly advantageous for Big Data projects with enormous amounts of data, characterized by speed and simplicity in accessing the data. However, this simplicity also highlights a significant limitation: the inability to perform complex queries.

1.3.3 Graph Databases

Graph-oriented databases are highly advantageous and useful for complex queries to retrieve specific data. Their performance is particularly high, offering shorter and faster paths, especially with large graphs. However, these databases are not recommended for data that can be retrieved more simply or that are not highly interconnected.

1.3.4 Column-Oriented Databases

Column-oriented databases excel in data compression capabilities. In this regard, they surpass any other type of database storing records in rows, as columnar databases rely on block-based reading, favoring compression.

1.4 Comparative: MongoDB e OracleDB

To provide a better illustration, a comparison between MongoDB and OracleDB will be presented, showcasing the differences between NoSQL and SQL databases, respectively.

1.4.1 Architectural Differences

The main architectural difference lies in data storage. OracleDB uses tables to store data, which is a more traditional concept, while MongoDB utilizes collections to store data.

1.4.2 Cost-Effectiveness

MongoDB is open-source, requiring no initial licensing costs. However, there are costs associated with hardware infrastructure and the team needed to maintain the

system. On the other hand, OracleDB has more significant costs, including software licensing and hardware requirements.

1.4.3 Usage and Management

MongoDB is known for its ease of use and management due to its flexible architecture and lack of rigid schemas. OracleDB, however, may demand greater expertise for a more efficient management, especially when handling large datasets.

1.4.4 Capabilities and Limitations

In terms of security, OracleDB is known for its robust measures, such as access control and cryptography. MongoDB, while improving its security, primarily relies on data authentication and authorization features. Regarding data consistency and availability, Oracle prioritizes consistency, ensuring cohesive transactions and queries. MongoDB, however, favors availability, allowing data access even during temporary unavailability.

CONCLUSION

This study presented and analyzed the characteristics of NoSQL databases, including a comparative analysis to further clarify the described aspects. Based on this study, we conclude that, although NoSQL databases are not the most widely used in the market, they offer significant benefits and are likely to grow further in the tech field.

When determining which database is superior between MongoDB and OracleDB, we do not claim one is better than the other. Instead, each is most effective when applied to the appropriate context, addressing specific needs.

By the end of this work, through an in-depth exploration of the NoSQL theme, we enhanced our understanding of both relational and non-relational databases. Additionally, we refined our database skills and acquired new ones, particularly through the use of MongoDB.

REFERENCES

AWS AMAZON. **Amazon Web Services**, 2023. O que é NoSQL?. Available in: https://aws.amazon.com/pt/nosql/ . Accessed on: 23/04/2024.

MICROSOFT, **Azure Microsoft**, 2024. Banco de Dados NoSQL. Available in: https://azure.microsoft.com/pt-br/resources/cloud-computing-dictionary/what-is-nosql-database . Accessed on: 23/04/2024.

IBM, **IBM**, 2023. O que são bancos de dados NoSQL?. Available in: https://www.ibm.com/br-pt/topics/nosql-databases . Accessed on: 24/04/2024.

ABRANTES, Andressa. Banco de Dados NoSQL: Um guia para iniciantes em Banco de Dados Não Relacional. **DIO**, 2023. Available in: https://www.dio.me/articles/banco-de-dados-nosql-um-guia-para-iniciantes-em-banco-de-dados-nao-relacional . Accessed on: 23/04/2024.

Erick. Banco de Dados NoSQL e sua utilização para big data. **Diário da Ciência de Dados**, 2023. Available in: https://diariodacienciadedados.com.br/banco-de-dados-nosql/ . Accessed on: 26/04/2024.

MICROSOFT, **Azure Microsoft**, 2024. Escalar verticalmente versus escalar horizontalmente. Available in: https://azure.microsoft.com/pt-br/resources/cloud-computing-dictionary/scaling-out-vs-scaling-up. Accessed on: 25/04/2024.

WURTHMANN, Felipe. Introdução à Bancos de Dados NoSQL Orientado a Colunas. **Medium**, 2020. Available in: https://medium.com/@fellipe.wurthmann/introdu%C3%A7%C3%A3o-a-bancos-de-dados-nosql-orientado-a-colunas-1f47dd52b8c2. Accessed on: 29/04/2024.

REFERENCES

BARROSO, Isaias. Banco de dados orientado a colunas. **Wordpress**, 2012. Available in: https://isaiasbarroso.wordpress.com/2012/06/20/banco-de-dados-orientado-a-colunas/ . Accessed on: 30/04/2024.

ByteScout Team of Writers, **ByteScout**, 2024. MongoDB vs Oracle Detailed Comparison. Available in: https://bytescout.com/blog/mongodb-vs-oracle.html#1 . Accessed on: 28/04/2024.

BD-ENGINES, **DB-Engines**, 2024. System Properties Comparison MongoDB vs. Oracle. Available in: https://db-engines.com/en/system/MongoDB%3BOracle. Accessed on: 29/04/2024.

UA - **Analisar e compreender a utilização do Banco de Dados NoSQL** – 11503. Accessed on: 26/04/2024.