THE UNIVERSITY OF HUDDERSFIELD
School of Computing and Engineering

CMI3509 Assignment I

Databases and its Application Domain

Name: OJIKUTU AISHAT OMOLARA
Student ID: U2293033
Date: December 24/03/2023

Abstract

In this research, three database systems, one relational and two NoSQL were used to analyze the strengths and weaknesses of each database using a real-life case study of system development projects. Overall, the choice between relational and non-relational databases depends on the specific needs of the organization and the type of data being stored and managed.

1 Introduction

DBMS is a software tool that facilitates the storage, management, and retrieval of digital data in a secure and consistent manner. Its design aims to ensure easy management and retrieval of data. Oracle (2023) The two primary categories of DBMS are relational and non-relational databases.

1.1Relational Database

IBM (2023) defines a relational database as a database that arranges data in tables with rows and columns. Relationships between tables are established through primary or foreign keys, and SQL queries are used by analysts to merge data from multiple tables and extract insights to enhance business performance, optimize workflows, and uncover new opportunities. Various data models are used to illustrate these relationships. Relational databases come in several types, such as PostgreSQL, MySQL, and others.

1.2 No relational Database

IBM (2023) explains that NoSQL databases are distinct from relational databases since they do not impose a strict schema for organizing data into tables. Instead, they organize data in several ways depending on the database type. The primary objective of non-relational databases is to tackle the flexibility and scalability challenges that relational models face when handling unstructured data formats like text, video, and images. In the realm of NoSQL databases, availability takes precedence over consistency. Thus, it is acceptable for different users to access different versions of data simultaneously. NoSQL databases come in various categories, such as Document database, Graph database, and others.

1.3 Data model

A model is a simplified representation of a complex system or concept.
In DBMS, a data model is a representation of data that describes how data is organized, stored, and retrieved in a database. A data model defines two things which are data structure and the operation to carry out on data structure.

2 Use cases

Applying PostgreSQL to manage regulatory data related to financial regulation (static data), MongoDB to manage Time series data related web traffic data and Neo4J to manage social network data.

2.1 Use Case I: Regulatory data (Financial Regulatory data) Using PostgreSQL

Firstly, a relational database system like PostgreSQL provides a structured and organized way of storing data, making it easier to maintain data integrity and consistency. Sharma, k (2021).

Financial regulation is a complex and highly regulated industry that requires secure and efficient management of substantial amounts of data. This structured approach is particularly important in the financial industry, where data accuracy and consistency are critical for compliance and regulatory reporting purposes. By using a relational database system like PostgreSQL, financial institutions can ensure that data is stored in a consistent and organized manner, reducing the risk of errors and inaccuracies.

PostgreSQL's ACID (Atomicity, Consistency, Isolation, and Durability) transactions guarantee that each transaction is processed as a single, isolated unit, ensuring that the database remains in a consistent state. Gulati, S. (2018),

This level of security and reliability is essential for regulatory data in the financial industry, where data breaches or errors can have profound consequences. PostgreSQL's scalability and performance make it an ideal choice for storing and managing substantial amounts of regulatory data. PostgreSQL's support for indexing and query optimization allows for efficient querying and processing of substantial amounts of data. Sharma.k,(2021)

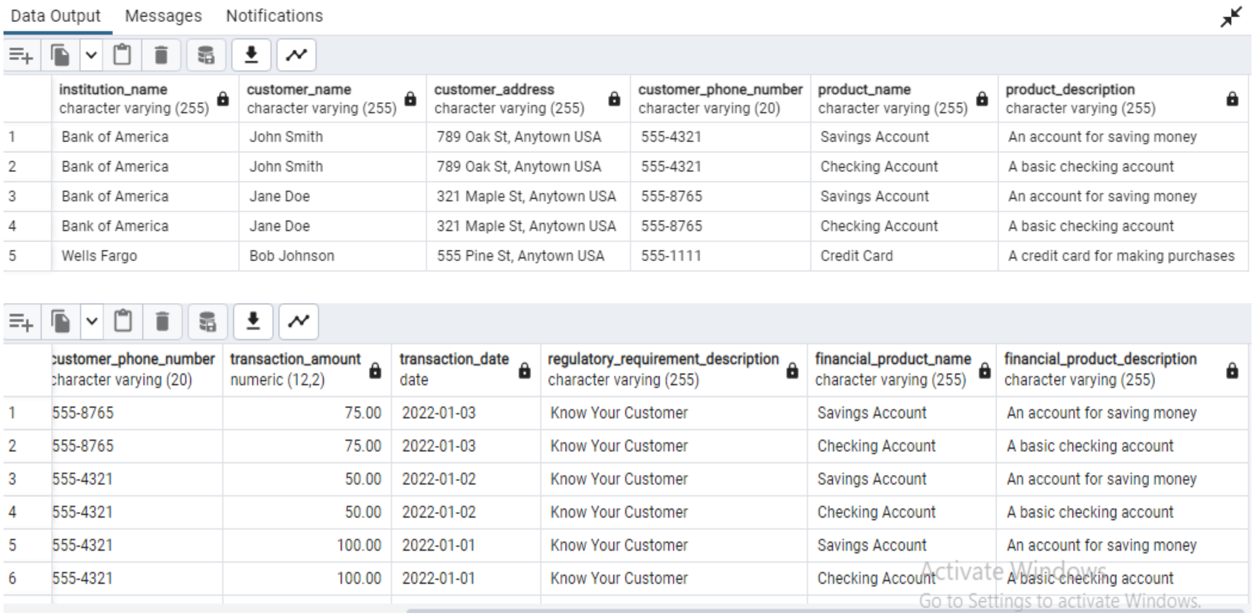
2.1.2 The Data Model of PostgreSQL

In PostgreSQL data modelling, a clear understanding of the requirements of the data and the relationships between the different entities needs to be generated. Identifying the entities involved in the regulatory data. Entities such as financial institutions, and customers. Defining the attributes for each entity. Attributes such as ID, address, date, and status. PostgreSQL (2023).

Determine the relationships between the entities. For example, a financial institution might have many customers, a customer might have many transactions, and a regulatory requirement might apply to many financial institutions.

Create a table for each entity in PostgreSQL, including columns for the identified attributes. Making sure to define primary and foreign keys as necessary to enforce the relationships between the tables.

In this example, a financial institution can have many customers and financial products. A customer can have many transactions. A regulatory requirement can apply to many financial institutions, and each financial institution must comply with multiple regulatory requirements.



	institution_name character varying (255)	customer_name character varying (255)	customer_address character varying (255)	customer_phone_number character varying (20)	product_name character varying (255)	product_description character varying (255)
1	Bank of America	John Smith	789 Oak St, Anytown USA	555-4321	Savings Account	An account for saving money
2	Bank of America	John Smith	789 Oak St, Anytown USA	555-4321	Checking Account	A basic checking account
3	Bank of America	Jane Doe	321 Maple St, Anytown USA	555-8765	Savings Account	An account for saving money
4	Bank of America	Jane Doe	321 Maple St, Anytown USA	555-8765	Checking Account	A basic checking account
5	Wells Fargo	Bob Johnson	555 Pine St, Anytown USA	555-1111	Credit Card	A credit card for making purchases

	customer_phone_number character varying (20)	transaction_amount numeric (12,2)	transaction_date date	regulatory_requirement_description character varying (255)	financial_product_name character varying (255)	financial_product_description character varying (255)
1	555-8765	75.00	2022-01-03	Know Your Customer	Savings Account	An account for saving money
2	555-8765	75.00	2022-01-03	Know Your Customer	Checking Account	A basic checking account
3	555-4321	50.00	2022-01-02	Know Your Customer	Savings Account	An account for saving money
4	555-4321	50.00	2022-01-02	Know Your Customer	Checking Account	A basic checking account
5	555-4321	100.00	2022-01-01	Know Your Customer	Savings Account	An account for saving money
6	555-4321	100.00	2022-01-01	Know Your Customer	Checking Account	A basic checking account

Overall, while PostgreSQL can be a strong choice for regulatory data management, it may not be the best for all use cases. However, like any technology, it has its limitations and weaknesses. Here are a few potential weaknesses of PostgreSQL for regulatory data: Limited scalability, complexity, Limited NoSQL capabilities, and Lack of native graph database support. It is important to consider the strengths and weaknesses of any database system before selecting it for a project.

2.2 Use Case II: Time series data (Web Traffic) using MongoDB.

MongoDB is a free and open-source database system that utilizes BSON for data storage and querying. It is a document-oriented database that offers high performance. One of the significant advantages of MongoDB is its dynamic schema, making it a flexible database system that enables easy integration of data, faster than traditional databases. (Zhao et al., 2013, p. 120).

Web traffic data is a type of time series data that can be used to track website activity over time, page views, unique visitors, and other metrics. As web traffic data is collected at regular intervals over time, it is well-suited to be stored in a time-series database, such as MongoDB, which is a NoSQL database system. MongoDB (2023).

To create a time series collection, it is important to use the createCollection command and provide distinct options, including the timeField option for the time series field, the metaField for metadata, the granularity, and the expireAfterSeconds field to automatically delete expired documents. MongoDB (2023).

MongoDB is an ideal solution for managing large volumes of data such as web traffic. MongoDB's scalability features, including replication, sharding, and replica sets, enable it to operate efficiently as data size and usage grow. MongoDB is also an excellent option for web traffic data because of its flexible schema design. It can be challenging to define a fixed data structure for analytics data upfront, as the most essential information to store may not be apparent. Packt (2018).
Flexible Data Model, High Write throughout, Scalability, Easy integration with Analytical tools, effective data processing, analysis and cost effectiveness are the reasons why MongoDB is a better choice for storing and analyzing web traffic data, MongoDB (2023).

2.2.1 Data Model of MongoDB

As per MongoDB's structure, it is a document-oriented database system that allows for multiple instances of databases. Each database instance can contain several collections that comprise different documents. These documents are constructed using a set of interconnected keys, and each document may have a distinct set of keys. Additionally, collections do not have a predefined schema, meaning that the keys of two documents in a collection may differ significantly. (Zhao et al., 2013, p. 120).

The MongoDB data model for web traffic data typically involves creating a separate collection to store each type of data, such as page views, user activity, or session data. Within each collection, data can be stored as individual documents, with each document representing a single event or activity. The data model can include fields for storing relevant information, such as user IDs, timestamps, page URLs, and other attributes. One common approach is to use nested documents to store related data, such as user information within a session document.

In summary, MongoDB is a well-suited database system for storing and analyzing web traffic data while a relational database system may be suitable for other use cases, MongoDB's strengths make it the better choice for handling the specific demands of web traffic data.

2.3 Use Case III: Social **Network data using Neo4j.**

Neo4j is a database management system that utilizes graph structures to store and manage data. It enables the creation, querying, and analysis of intricate, interrelated datasets in real-time. Graph databases represent data as nodes (or vertices) and relationships (or edges) between nodes. Each node may have properties, while each relationship may have a type and properties. Graph databases are highly beneficial for storing and retrieving data with intricate relationships, such as those found in social networks, recommendation engines, and fraud detection systems (Robinson, I. et al, 2013).

Neo4j supports complex data relationship, social networks have complex data relationships that are difficult to represent in a RDBS. For example, users can have multiple types of relationships with other users, such as friend, follower, or family member, each with its own set of properties. Representing these relationships in a relational database requires multiple tables and complex queries. Neo4j represents relationships as first-class citizens, making it easy to store and query complex data relationships. Robinson. I. et al (2013).

Neo4j also support performance, social networks require fast query response times to deliver a responsive user experience. Neo4j's graph database design is optimized for traversing complex graphs quickly, making it a faster option for social network applications.
Schema Design, Scalability, Data Integrity, and ease of development are also supported in Neo4j making it a better choice for social network project. Van Bruggen, R. (2014)

2.3.1 Data model of Neo4j.

The data model of Neo4j is based on a graph, which is a collection of nodes and relationships. Here, nodes represent entities, while relationships represent the connections between entities. Each node can have properties that define its attributes, and each relationship can have a type and properties that define its characteristics. Robinson. I. et al (2013). The main components of the data model in Neo4j are Nodes, Relationships, Labels, and Properties.

Overall, the graph-based data model of Neo4j provides a flexible and powerful way to represent complex, interconnected data sets. It allows for efficient querying and analysis of relationships between entities, making it a popular choice for use cases such as social networks, recommendation engines, and fraud detection systems. Robinson. I. et al (2013).

In summary, while relational databases have been traditionally used in system development, a no-relational database system like Neo4j is a better choice for social network projects due to its ability to handle complex data relationships, faster performance, schema flexibility, scalability, data integrity, and ease of development. Sadalage, P. et al. (2012).

3 Conclusion

In conclusion, the choice of database management system should be based on the specific requirements of the application and the characteristics of the data being stored and managed. With the right database management system in place, organizations can ensure the efficient, secure, and scalable management of their data.

4 References

Gulati, S. (2018). *The Minimalistic Guide to ACID Transactions*. https://shekhargulati.com/2018/08/25/the-minimalistic-guide-to-acid-transactions/

Li, L., Liu, Z., Ye, F., Zhang, P., & Lu, M. (2019). "Availability Enhancement of Riak TS Using Resource-Aware Mechanism",. *Mathematical Problems in Engineering, 2019*(2189125), 11. https://doi.org/ https://doi.org/10.1155/2019/2189125

Zhao, G., Huang, W., Liang, S., & Tang, Y. (2013). Modeling MongoDB with Relational Model. *2013 Fourth International Conference on Emerging Intelligent Data and Web Technologies,* 115-121. https://doi.org/10.1109/EIDWT.2013.25

Sharma, K. Ubuntu. (2021, January 19). What is PostgreSQL and why do enterprise developers and start-ups love it?. https://ubuntu.com/blog/what-is-postgresql#:~:text=The%20reliable%20database,trusted%20for%20its%20data%20integrity

Neo4j. (2023). *social media and social network graph*. https://neo4j.com/use-cases/social-network/

Oracle. (2023). *What is a Relational Database (RDBMS)?*. https://www.oracle.com/database/what-is-a-relational-database/

IBM. (2023). *What is a Relational Database (RDBMS)?*. https://www.ibm.com/topics/relational-databases

PostgreSQL. (2023). *PostgreSQL 15.2 Documentation*. https://www.postgresql.org/docs/current/

MongoDB. (2023). *MongoDB Time series data*. https://www.mongodb.com/features/mongodb-time-series-data

MongoDB. (2023). *Data modelling introduction* . https://www.mongodb.com/docs/manual/core/data-modeling-introduction/

Sadalage, P. J., Fowler, M. (2012). NoSQL Distilled: A Brief Guide to the Emerging World of Polyglot Persistence. Pearson Education.

Van Bruggen, R. (2014). Learning Neo4j. Packt Publishing.

Robinson, I., Webber, J., & Eifrem, E. (July 2013). *Graph Databases*. O'Reilly Media