Stephen Brom

10/29/2023

CS-300-ON Database Management System

Assignment 1: Investigate non-Relational Data Storage and Retrieval System

Non-Relational Data Storage and Retrieval Systems

"For my part, I know nothing with any certainty, but the sight of stars makes me dream." -Vincent Van Gogh. When I first started this journey into the world of computer science one of my first teachers said, "Computers are to computer science, as telescopes are to astronomy". I had no idea how true this statement was at the time. It is exciting to think how one could spend their whole life learning and investigating the tools developed in computer science. A life in computer science guarantees that one's investigations will never be completed. In this investigation we will take a deep look into tools such as NoSQL databases, graph databases and more. We will investigate into the history, and use of such tools.

One of the first aspects we want to investigate is What does NoSQL Mean? A NoSQL database is also called a "not only SQL" or a non-tabular database. This means unlike the in a relational database where it uses Columns and tables in a ruled based organization. NoSQL databases use non-related data structures to store and access data. They can do this in 4 formats according to Google Could website and reading to use a direct quote:

Document databases

Document databases, also called document-oriented databases or a document store, are used to store and query semi-structured data. Data is stored in a JSON-like document similar to the data objects that developers use in application code, making it easier to create and update applications without referencing a primary schema. Document databases are most used for blogging platforms, ecommerce and real-time analytics applications, and CMS systems.

Key-value databases

Key-value databases, also referred to as key-value stores, are the simplest type of NoSQL databases. Data is stored in a “key-value” structure, where a unique key is paired with a value, such as strings, numbers, Booleans, complex objects, and more. Data can be written or queried using the key, which is used to store or retrieve its associated value. Key-value stores are most commonly used for user preferences, shopping carts, and user profiles in web applications.

Wide-column stores

Wide-column stores, or column-oriented databases, store and read data in rows and are organized as a set of columns. While similar to the tabular format of relational databases, column names and formatting in wide-column stores can vary from row to row in a single table. They are optimal for analytics use cases, where you may need to query across specific columns in a database and aggregate the value of a given column quickly. Wide-column stores are most commonly used for catalogs, fraud detection, and recommendation engines.

Graph databases

Graph databases organize data as nodes, focusing on the relationships between data elements. These connections between nodes, referred to as edges, are stored as first-class elements, enabling richer representations of data relationships while offering more simplified storage and navigation. Graph databases are most commonly used in systems that map relationships, including social media platforms, reservation systems, and logistics applications.

So why did we develop these NoSQL databases? What issues do these solve? NoSQL databases are databases that do not use the relational model of data storage, which is based on tables, rows, and columns. NoSQL databases use different data structures, such as documents, key-value pairs, graphs, or wide columns, to store and access data. NoSQL databases have some advantages over relational databases, such as flexibility, scalability, speed, and ease of use for certain types of data and applications. Some of the problems that NoSQL databases were designed to solve are handling large amounts of unstructured or semi-structured data that are generated by web applications, social media and data of that sort. These data may have different formats, schemas, or levels of complexity that are not suitable for relational databases. Another Problem NoSQL databases solve is the ability to scaling out horizontally across multiple servers or nodes to increase the capacity and performance of the database. Relational databases usually scale up by upgrading the hardware of a single server, which is more costly and complex. No SQL databases also supporting agile and iterative development of web applications, as they allow developers to modify the data model without affecting the existing data. Relational databases require a predefined schema for the data, which may limit the flexibility and creativity of developers. Having developed these NoSQL database also help with storing and querying complex relationships between entities, such as social networks, recommendation systems, fraud detection, etc. Graph databases use a branch of mathematics known as graph theory to model and analyze these relationships using nodes and edges. NoSQL databases are not a replacement for relational databases, but rather a complement for different use cases.

Now that we have talked about what NoSQL databases are we can talk about a couple of real-world examples.

MongoDB: MongoDB is a document database that stores data as JSON-like documents. Each document can have a different structure and fields, which makes it flexible and easy to use. MongoDB supports dynamic queries, indexing, aggregation, and text search. MongoDB is suitable for web development, real-time analytics, and content management systems.

Redis: Redis is a key-value database that stores data as pairs of keys and values. The values can be strings, numbers, lists, sets, hashes, or other data structures. Redis supports various operations on the values, such as sorting, filtering, ranking, etc. Redis is also an in-memory database, which means it stores data in RAM for fast access. Redis is suitable for caching, session management, pub/sub messaging, and leaderboards.

Neo4j: Neo4j is a graph database that stores data as nodes and relationships. Nodes represent entities, such as people, places, or things, and relationships represent how they are connected. Neo4j supports powerful queries using a language called Cypher, which can traverse the graph and perform complex calculations. Neo4j is suitable for social networks, recommendation systems, fraud detection, and knowledge graphs.

More and More real-life examples are coming online every day. These real-world examples will have pros and cons. Having new examples will bring new advantages and disadvantages. We can look advantages and disadvantages overall.

So, what are the overall disadvantages and advantages of NoSQL databases? Sure, NoSQL databases, also known as non-relational databases, have both advantages and disadvantages. Lets look at some of the strongest advantages of NoSQL. First let’s talk about the scalability of NoSQL databases and how they are more scalable than traditional relational databases and can store a variety of formats. They were developed with the internet and cloud computing as support systems, making it possible to develop a scale-out architecture. They are also known to have a flexible data model with the schemas of several NoSQL databases that are quite flexible and can be controlled by the developers. Making it easier to modify the database for new forms of data. NoSQL have a high performance capabilities NoSQL databases can process large volumes of data at high speed. NoSQL can also support a bunch of different various data types, whether unstructured, structured, or semi-structured, can be stored in a NoSQL database unlike in a relational database. NoSQL are also very developer-friendly NoSQL is developer-friendly as it allows developers to control the structure of the data more easily. Even with all of these advantages, NoSQL also have some disadvantages. NoSQL is a less mature technology; NoSQL is a relatively new technology and may not have as vast a community as SQL. It might lack some features present in traditional SQL databases. NoSQL also have limited query capabilities when it comes to analytics and drawing business insights, NoSQL is not as efficient as SQL. There is also a lack of consistency with NoSQL they suffer from consistency when it comes to storing large amounts of data. Even with the advantages and disadvantages between SQL and NoSQL databases, choosing which ones depends on the specific requirements of your specific project.

Let us talk about what are graph databases. A graph database is a specialized, single-purpose platform used to create and manipulate data of an associative and contextual nature. The graph itself contains nodes, edges, and properties that come together to allow users to represent and store data in a way that relational databases can’t calculate. According to Oracle the main concept of a graph database system is a relationship. Relationships are defined as first-class citizens — this means everything you can do with all other elements can be done with a relationship. Data is related together in a graph to store a collection of nodes and edges, where the edges represent the relationship between nodes. Relationships allow data within the system to be linked together directly. Querying relationships in a graph database is fast since they’re stored in a way that doesn’t change. You may also visualize them, which makes them great for deriving insights for heavily interconnected data. Relational databases use data tables to structure information into rows and columns. Each column defines a specific attribute of the data entity, while the rows represent an individual data record. Since data tables have a fixed schema, users must define the relationships between different tables using primary and foreign keys. In contrast, a graph database structures data using a graph structure in which nodes, edges, and properties are used to represent data. Namely, nodes define objects, edges illustrate the relationships between nodes, and properties describe the attributes of the nodes and edges. Graph databases are designed to solve problems that require complex relationship and connection analysis. Here are some of the bigger issues that graph databases are designed to solve. They help with issues like, social network analysis they can efficiently analyze and map out relationships and interactions within a social network. They help with fraud detection; they can help uncover fraudulent activities by analyzing patterns and anomalies in transaction data. Graph databases are well-suited for building recommendation engines as they can effectively analyze user behavior and preferences. Data regulation and privacy can help ensure compliance with data regulation and privacy rules by mapping out data flows and accesses. AI and machine learning research is another issue graph databases can be used to represent complex, interconnected data sets, which is useful in AI and machine learning research. Matching problems graph databases can solve matching problems more efficiently, such as matching processors to workload or workers to jobs. Graph databases are used to iterative algorithms, they are also useful for running iterative algorithms such as PageRank, gradient descent, and other data mining and machine learning algorithms. When investigating graph databases, we should look at two examples in the real world. If we first look at Tiger Graph. Tiger Graph is an enterprise-scale graph database supporting real-time analytics. It is built around both local storage and computation, supports real-time graph updates, and works like a parallel computation engine. Tiger Graph sits across the top of your master data sources, connecting, cleansing, and unifying customer and product information. Its inbuilt analytics then query all of that connected data in real-time, providing the insights or triggers required to underpin a real-time cross-sell engine that operates across all channels. Another Real Life example is Neo4j is a graph database management system that uses nodes and relationships to represent and store data. Unlike traditional relational databases that use tables, Neo4j's structure allows for efficient querying and handling of complex interconnected data. It is designed for optimal management, storage, and traversal of nodes and relationships. Neo4j offers dedicated memory management and memory-efficient operations. It can be installed on-premises and deployed in various systems: Linux, macOS, Windows4. To extend Neo4j capabilities, you can install several optional tools such as plugins. As with any new tools we have advantages and disadvantages of using these. Some of the advantages are as follows. Graph databases use object-oriented thinking; Graph databases provide very clear, explicit semantics for each query you write. They allow flexibility, Graph databases are flexible and focus on quick data relationship insight. Graph databases are quick with complex queries. Systems with highly connected relationships leverage the power of graph databases, outperforming traditional relational databases. These Tools also have drawbacks. For graphing databases there is a big learning curve. There can be a steep learning curve as graph databases require a different mindset compared to traditional relational databases. Another really big disadvantage is that they have limited use cases. While they excel at handling complex relationships, they may not be the best choice for simple, tabular data. Like with NoSQL databases these are still newer tools in the toolbox in computer science. This being the case, there isn't as much established help or support.

In Conclusion of this investigation, non-relational data storage and retrieval systems, often referred to as NoSQL databases, have emerged as a critical component in the field of computer science, offering unique solutions to specific data storage and retrieval challenges. These systems come in various formats, including document databases, key-value databases, wide-column stores, and graph databases, each tailored to different use cases. NoSQL databases provide flexibility, scalability, speed, and ease of use, making them well-suited for handling unstructured or semi-structured data generated by modern web applications and social media platforms. The advantages of NoSQL databases include their scalability, flexible data models, high performance capabilities, and developer-friendly nature. They are particularly adept at handling diverse data types and accommodating agile development. However, NoSQL databases also have their disadvantages, such as being a relatively less mature technology compared to traditional SQL databases and having limited query capabilities for complex analytics. Graph databases, a specialized type of NoSQL database, focus on representing and storing complex relationships and connections between data elements. They excel in applications like social network analysis, fraud detection, recommendation engines, data regulation compliance, AI, and machine learning research. Examples like Tiger Graph and Neo4j demonstrate the power of graph databases in handling interconnected data efficiently. While graph databases offer clear advantages in managing complex relationships and providing quick insights, they come with a learning curve and may not be suitable for simpler, tabular data. Moreover, like other NoSQL databases, they are relatively newer tools, and support and resources may be less established. In summary, the world of non-relational data storage and retrieval systems, including NoSQL and graph databases, provides valuable options for addressing diverse data storage and analysis needs. Choosing between these technologies should depend on the specific requirements of your project, and understanding their strengths and limitations is crucial for making informed decisions in the field of computer science. As Vincent Van Gogh once said, "For my part, I know nothing with any certainty, but the sight of stars makes me dream," and in the realm of computer science, these innovative databases continue to inspire new possibilities and innovations.

References:

1) What Is NoSQL? NoSQL Databases Explained | MongoDB

2) What is NoSQL and how does it work | Google Cloud

3) NoSQL Basics: Features, Types, and Examples (spiceworks.com)

4) Challenges of NoSQL - GeeksforGeeks

5) NoSQL technologies are built to solve business problems, not just "wrangle big data" – O'Reilly (oreilly.com)

6) When to Use NoSQL Databases: NoSQL vs. SQL (serverwatch.com)

7) NoSQL Databases: How Does NoSQL Work + Example of Structures (couchbase.com)

8) NoSQL Databases: Advantages and Disadvantages - DATAVERSITY. https://www.dataversity.net/nosql-databases-advantages-and-disadvantages/.

9) NoSQL Advantages and Disadvantages - Blue Claw Database Developer Resource. https://blueclawdb.com/nosql/nosql-advantages-disadvantages/.

10) SQL vs. NoSQL: The Differences Explained + When to Use Each. https://www.coursera.org/articles/nosql-vs-sql.

11) SQL vs NoSQL: Differences, Advantages and Disadvantages - Webandcrafts Blog. https://webandcrafts.com/blog/sql-vs-nosql/.

12) What is a Graph Database? | Oracle

13) 17 Use Cases for Graph Databases and Graph Analytics (oracle.com)

14) (1) New Messages! (tigergraph.com)

15) Introduction - Operations Manual (neo4j.com)