Kendall Ladrillono

Assignment 1

CS300-ON

Database Management Systems

Non-Relational Data Storage and Retrieval Systems

Databases have a long-standing history, which began in the 1970s. Edgar F. Codd, a computer scientist, was the person who introduced a revolutionary idea which became the groundwork for relational databases. He proposed a method for storing and retrieving information in a relational database. The structure that he discussed consisted of tables that were linked together that would contain data and with each piece of data only stored one time. It was designed to answer any query, as long as that information was stored in the database. Software engineers and developers learned to communicate with the database using a special language called SQL. This language allows users to tell the database exactly what action they are targeting, and the database is able to respond. Relational databases grew in popularity throughout the next two decades. It was an efficient way to store and organize data for the requirements during that time. Unfortunately, it was not equipped to handle the arrival of the internet and because of this a new type of database was created to address this issue.

NoSQL developed as a response to the rise of the internet. The internet brought increasingly large amounts of data with it and the need for databases that were able to handle these volumes arose. Prior to NoSQL, relational database systems were the norm. During that time, the relational databases were capable of handling only small amounts of data very efficiently. Relational databases also require its data to be structured. With the internet, the need for processing many types of unstructured data and faster processing became a new challenge. It is applicable for many purposes such as big data, web applications, online transactions, gaming and social networking.

NoSQL is a non-relational database in which it does not store data with a defining structure of tables with fixed columns and rows. Instead, NoSQL can be broken down into four main types; these are document databases, key-value stores, column-family stores, and graph databases. NoSQL allows for dynamic schema which enables flexibility because a developer is not required to specify the type of data before storing. It is ideal for those who need to build something quickly or who may be in the early stages of creating an application because it handles unstructured data well. NoSQL is more tailored for specific-use applications while relational databases are more suited for general purpose applications.

There are many options when choosing a NoSQL database. MongoDB is one such database and it stores collections of documents. It is open-source therefore it is available to the public. MongoDB formats documents in BSON, which stands for binary encoded Javascript Object Notation or JSON. This allows for a wider range of data types. The collections contain documents, and the documents contain data in a structure of key-value pairs. Documents are assigned a specific key and the value can range from a selection of data types, another document, or an array of documents. MongoDB is multifaceted in which it accepts structured, semi-structured, and unstructured data. MongoDB allows a user to create more than one database and is optimal for developers who are searching for streamlined access to specific contents within documents as it provides case-by-case queries and indexing of data. The data is stored in RAM, which creates quick access and efficient searching. It balances out its database by sharding; in which it distributes portions of the database into smaller parts in order to maintain the large amounts of data. MongoDB also provides support for a multitude of programming languages.

Redis is another type of NoSQL database, and it is also open-source. It stands for Remote Directory Server and functions by creating key-value pairs and stores the data in memory rather than on a disk. It is a popular database choice due to its reliability, speed, and performance. When a user would like to access data, instead of sending out a query to an external database server and waiting for a response, Redis keeps the data within the server’s memory, which is physically much closer and easier to access. This way, the user can retrieve or send data much faster than data stored on disks. Redis also offers the ability to queue tasks and provides support for many different data types and data structures.

Apache Cassandra is a popular NoSQL database that is widely used by large companies. It is said to have no limit in regard to scalability and is extremely efficient. Cassandra is separated into many nodes and each node stores an amount of data. The nodes have the same capabilities and can be distributed and scaled horizontally as data grows but each node is responsible for its own portion of data. Nodes are grouped together in clusters. One key factor that differentiates Cassandra with other NoSQL databases is that data is copied on to multiple nodes within the cluster which ensures no single point of failure. Data is contained within a keyspace which allows users to have control over how their data is copied. Each keyspace has tables similar to relational databases but Cassandra does not require specific schema and supports unstructured data. The tables consist of columns, a primary key, and rows which store data into sections. Each row has its own partition key. The method in which Cassandra allows for separating data makes it ideal for scaling according to a user's needs.

NoSQL provides many advantages for developers. One characteristic that NoSQL allows is high scalability in which it can handle large amounts of data growth without significantly increasing the cost. As an application grows and develops, the reassurance that a database is reliable to handle an increase is beneficial. Another characteristic of NoSQL is its flexibility. As stated previously, a developer may be in the early stages of program development and may not have all the data types defined yet. NoSQL does not require a specified schema before it is able to store this data. Management of NoSQL is also easier than relational databases. It is more reliable and has much better performance when it comes to handling the demands of today's applications with constant exchange of large amounts of data.

Despite NoSQL having many advantages for use, it also has its disadvantages. Not all NoSQL share the same syntax so a developer must be aware to use the correct syntax when manipulating. The flexibility that NoSQL allows also comes with a downside. There is no specific requirement for data schema before storage and because of this, the data integrity and consistency may be lacking whereas in relational databases, they are embedded into their programming. This issue of consistency leads to another issue of data management. It can become more difficult to manage data where data types may not have the same structure and that data starts to grow quickly. Another caveat to the absence of a set schema requirement is that the developer must provide some sort of schema structure. In some relational databases this could be done by the database administrator. There are many options when choosing a NoSQL database and it may be a challenge for a user to choose which one is right for their needs. NoSQL databases have yet to be standardized so the user must weigh their options accordingly.

Graph databases are one form of NoSQL database storage types. It creates a map-like structure that organizes data into nodes and edges. The nodes represent a record of the data itself and the edges between nodes represent the relationships between the data. The graph stores these relationships and can effectively manage them. There are multiple different types of graphs. Undirected graphs are one such type of graph representations and in this model, the nodes and edges can be switched in any way. There are directed graphs in which the nodes and edges are not interchangeable, as they are with undirected graphs. This means that the relationship from node to node can only exist in one direction. Graphs with weight are another model type. This graph assigns a numerical value to the relationships between nodes and allows for mathematical computation. There are graphs with labels, in which labels are assigned to the relationships between nodes. It exists on social network platforms where a label such as sibling or aunt is a category that a user might want to use to define their relationship for specific “followers”. The last type of graph representation are property graphs. For this graph model, it is a type of graph with weight that can also have labels to allow for properties to designate to both nodes and edges.

Graph databases are optimal when the relationship between data is more important to store and control than the data itself. Graph databases are used to support and manage complex relationships between data. It is used for many applications. Some examples include social networks, fraud detection, scenario optimization, and search recommendations. Many of today’s social networks are essentially built on existing relationships and building relationships therefore the use of graph databases to store and manage them is optimal. These databases can determine patterns in data, and it makes possible the suggestions of new friends and pages that align with a specific user’s interests. Graph databases are also used as an underlying structure when using GPS. A GPS map itself is a representation of a graph and the cities and routes are the nodes and edges. The database can determine the most efficient route to take to get to a destination. Fraud detection is an important use for graph databases as well. If someone has stolen credit card information and attempts to use it, the graph database will notice this unsuspecting pattern change if the owner lives and uses their credit card in Oregon while the charge is being made in Florida. It will send an alert to the credit card company and the company will alert the owner. This can occur almost immediately, which shows how efficiently the database works.

Neo4j is a popular graph database and is implemented in the Java language. The graph model that it utilizes is a property graph. It stores nodes, which contain data, edges, which represent the relationship between nodes, and properties, which a user can pick to associate within a node. The nodes and edges store data in key-value pairs and these are the designated properties. Neo4j has its own query language called Cypher. Its databases handle queries by constantly updating the searched-for nodes and edges without updating the graph as a whole. This method allows for efficiency when performing a query. Data is stored on disks and the disks consist of linked lists. The properties are linked lists of records with each record possessing a key and value and points to the next property.

TigerGraph is another graph database service that has many features. It uses GSQL for its query language and is capable of handling large sets of data with complex relationships. Its main use is for data analytics and machine learning with relational data. TigerGraph’s database is distributed and contains underlying algorithms that can perform computations. A user’s data is automatically separated into several nodes and the platform keeps track of which nodes contain which portion of data. Data for nodes and edges in a graph has an equal and parallel unit of storage that is able to perform mathematical computations in real time. TigerGraph’s query performance is said to be extremely fast and efficient.

Advantages of graph databases include flexibility, performance, and efficiency. Its main purpose is to manage relational data. It is highly useful in today’s applications because relationships and patterns created over the internet grow increasingly fast in just one day. Graph databases use the relationships and various graph algorithms to find patterns, recommendations, paths, and new relationships. Since these databases place an importance on relationships, performing queries to determine connections can be highly efficient and can be completed quickly. Graphs are flexible and make it easy for a user to explore new connections that the database provides. Graphs are the closest natural representation of information that anyone can recognize. It is not largely affected by the amount of data but by the number of tangible relationships. There are many options for graph representation to choose from so a user can choose which form best suits their requirement.

There are some drawbacks to graph databases as well. Though these databases are built on storing and managing relationships, it is possible that they can make inferences of these relationships that may not be accurate or have relevance. This also may be an issue for sending queries. As data scales and the relationships become more defined, it can add to the complexity of these relationships. This may make it more difficult to manage. Another disadvantage is the difference in query language that a database uses, sometimes a database may have its own specific language that a user must be aware of before making their choice. Cost is another factor that can become a downside for usage. Depending on how complex or the capacity required for a user, the cost can certainly add up with need. It can also be difficult to find support when running into any problems. NoSQL databases are still considered new to the database realm so it may be difficult to find quick solutions.

Databases are an important part of the devices and applications we utilize throughout our day. We may not be aware of the underlying operations of our devices, but without databases, none of it would be possible. We live in a society that demands speed of functionality and is not aware of how much work it requires to create and manage the databases we interact with. With the development of NoSQL, the applications that millions of people use such as social media, streaming services, banking, all of it has been made possible, reliable, and quick. It is consistently improved and adapted to society’s changes and pace. Though it is still seen as a newer form of database, it handles the demands of modern-day expectations very well.

References:

Mullins, C. S., Vaughan, J., & Beal, B. (2021, April 8). *NoSQL (Not Only SQL database)*. Data Management. https://www.techtarget.com/searchdatamanagement/definition/NoSQL-Not-Only-SQL

MongoDB. (n.d.). *What is NoSQL? NoSQL databases explained*. https://www.mongodb.com/nosql-explained

Foote, K. D. (2018, June 13). *A Brief History of Non-Relational Databases - DATAVERSITY*. DATAVERSITY. https://www.dataversity.net/a-brief-history-of-non-relational-databases/

Database Security Best Practices. (n.d.). https://www.oracle.com/database/nosql/what-is-nosql/

Lacefield, J. (2018, August 29). The Evolution of NoSQL | Datastax. *DataStax*. https://www.datastax.com/blog/evolution-nosql

GeeksforGeeks. (2021, June 6). *What is MongoDB Working and Features*. https://www.geeksforgeeks.org/what-is-mongodb-working-and-features/

What is Redis Explained? | IBM. (n.d.). https://www.ibm.com/topics/redis

GeeksforGeeks. (2023, October 13). *Introduction to NoSQL*. https://www.geeksforgeeks.org/introduction-to-nosql/

4 Best & Most Popular NoSQL Databases - BairesDev Blog: Insights on software development & tech talent. (2023, May 9). BairesDev Blog: Insights on Software Development & Tech Talent. https://www.bairesdev.com/blog/nosql-databases/

What is a graph database? (n.d.). Amazon Web Services, Inc. https://aws.amazon.com/nosql/graph/#:~:text=Graph%20databases%20have%20advantages%20for,and%20quickly%20query%20these%20relationships.

6point6. (2023, September 5). *Use cases for graph databases - 6point6*. https://6point6.co.uk/insights/use-cases-for-graph-databases/

Jones-Gilardi, D. (2022, May 11). Introduction to Apache Cassandra - the "Lamborghini" of the NoSQL world. *DataStax*. https://www.datastax.com/blog/introduction-to-apache-cassandra-the-lamborghini-of-the-nosql-world

Neo4j: What a graph database is and what it is used for. (2020, December 30). BBVA API_Market. https://www.bbvaapimarket.com/en/api-world/neo4j-what-graph-database-and-what-it-used/

What is a Graph Database? (n.d.). Oracle. https://www.oracle.com/autonomous-database/what-is-graph-database/

Lee, V. (2018, May 2). *TigerGraph: The parallel graph database explained*. InfoWorld. https://www.infoworld.com/article/3269604/tigergraph-the-parallel-graph-database-explained.html