Adam Elias
CS 300-ON
Fall 2023

Non-Relational Data Storage and Retrieval Systems

What does NoSQL mean?

NoSQL is a category of database that relies on a storage method that is different than the "traditional structures found in relational databases" (IBM, 2023). Put simply, it means "Not SQL," or "not only SQL." Whereas SQL relies on storing data in traditional tables, NoSQL databases do not. While SQL is one of the most common and popular approaches to database design, it does not fit every situation or need, and thus NoSQL may be necessary, appropriate, and more efficient. NoSQL represents a more recent and thus nontraditional database architecture that is "different from relational tables," allowing for query through "idiomatic language APIs, declarative structured query languages, and query-by-example languages" (Oracle, 2023). For those new to the field of databases, it may be interesting that NoSQL is framed as an alternative to a longstanding choice of SQL databases, but not as a specific alternative, as NoSQL can refer to a wide range of options and does not imply one particular approach; it is, in a sense, a generic term.

What types of problems were NoSQL databases designed to solve?

NoSQL databases were designed to solve problems in which data "doesn't fit neatly into the relational model" of databases. For example, in the modern world, a given application may pull in data from a wide variety of "disparate sources," such as social media platforms, smart devices and wearables, and widely decentralized sources/vendors/partners that maintain their data in closed or varying methods. These sources may not be easily compiled into a relational database, and thus a NoSQL database may provide a better fit and functionality. The "variety, velocity, and volume of data" involved in modern applications and processes may call for a database structure in which relational databases simply are not up to the task. (Amazon, 2023, "NoSQL")

What are three examples of NoSQL database products and how do each of them work?

Three examples of NoSQL database products include Couchbase Server, Amazon DynamoDB, and MongoDB. Couchbase defines its Server product as "a modern cloud-native, distributed database that fuses the strengths of relational databases such as SQL and ACID transactions with JSON flexibility and scale that defines NoSQL," and portrays its database product as one that "excels in high volume transactions" (Couchbase, 2023, "Server"). Amazon DynamoDB is "a fully managed, serverless, key-value NoSQL database designed to run high-

performance applications at any scale" (Amazon, 2023, "DynamoDB"). MongoDB's NoSQL product is called Atlas. Each of these are serverless cloud-based applications that provide NoSQL database capabilities. In each case, the company hosts data and database capabilities for its customers, allowing clients to deploy database solutions anywhere in the world, to scale up as needed without worrying about hardware and physical constraints, and to reduce complexity for aspects of client business.

What are the advantages and disadvantages of NoSQL databases?

The advantages of using a NoSQL database include its ability to "support large numbers of concurrent users (tens of thousands, perhaps millions), deliver highly responsive experiences to a globally distributed base of users, be always available – no downtime, handle semi- and unstructured data, and rapidly adapt to changing requirements with frequent updates and new features" (Couchbase, 2023, "NoSQL Databases"). Again, these are all scenarios in which the traditional SQL database may be impractical or insufficient. Couchbase identifies numerous global trends as fueling the need for NoSQL databases, including the increasing migration from a physical to a digital economy in which customers complete transactions online, the wide connectedness of devices to the Internet and each other, the massive amounts of data involved in economic processes, the cloud-based nature of business applications, and the increasingly mobile nature of apps and services. Traditional databases may be "too slow and too expensive," as well as "too rigid and complex" (Couchbase, 2023, "Server").

According to Couchbase (2023, "NoSQL Databases"), "Global 2000 enterprises are rapidly embracing NoSQL databases to power their mission-critical applications," and the company provides examples of Tesco, Marriott, Gannett, and GE as widely known clients who rely on the NoSQL database type for their businesses. As the complexity of the modern business landscape continues to increase, the usefulness of traditional SQL database structures may decline as more and more companies continue to invest in NoSQL capabilities. Further, the reality is that with SAAS applications as common and widely available as they are, even smaller companies and organizations can tap into the many and varied data streams that underscore the value of a NoSQL database approach. A company does not already have to have access to those kinds of data funnels; vendors and potential partners stand at the ready to sell access to them in ways that would be compatible with the implementation of NoSQL as through Couchbase, Amazon, or MongoDB.

At the same time, SQL is far from an outdated or obsolete technology, and continues to provide advantages over NoSQL. In terms of disadvantages, a "consistent complaint about NoSQL" is that—naturally—it does not

utilize SQL, which is "a mature technology" on which countless organizations and businesses have long based their database infrastructures (Foote, 2022). Another disadvantage is its "lack of standardization," as databases categorized as NoSQL may each use their "own unique schema," if there is a schema at all—or perhaps it is dynamic and changes with the use case (Foote, 2022). Because of the wide variety of architecture and protocols resulting from the lack of standardization, NoSQL can present challenges for analysts, as individual NoSQL scenarios may vary widely from case to case.

Another core disadvantage, according to Foote, is that NoSQL does not support the "ACID" properties, which stands for Atomicity, Consistency, Isolation, and Durability. While this may be changing as NoSQL solutions evolve, each of those terms represents an important property of SQL databases that positions that type of database as more "trustworthy" than NoSQL databases. Again, an advantage of SQL databases is that the technology has a history through which many engineers, scientists, researchers, thinkers, and analysts have come and gone, contributing their time and talents to the development of SQL, its surrounding principles and practices, and the world in which it has been used for many years. All this is to say, there is great demonstrated value in SQL, and the decision to move from SQL to NoSQL is not a simple choice.

What is a graph database?

According to Amazon, "A graph database is a systematic collection of data that emphasizes the relationships between the different data entities" (2023, "What is a graph database?"). The concept derives its name from mathematics, where a graph "contains a collection of nodes and edges," which are utilized in a graph database to portray data points and relationships. Each node "has properties or attributes that describe it. In some cases, edges have properties as well." There is no limit to the number of node types or relationships they can have with other nodes. Edges are used to portray "relationships between nodes," and can include examples such as "parent-child, actions, or ownership," among countless others. Further, such relationships can be between two individual nodes or among large groups, and any combination in between. "An edge always has a start node, end node, type, and direction." (Amazon, 2023, "What is a graph database?")

What types of problems were graph databases designed to solve?

Graph databases were designed to solve problems that SQL relational databases could not handle as well, such as needs arising from the rapidly expanding landscape of social media or other scenarios that involve "heavily interconnected data" (GeeksforGeeks, 2023). This type of database is highly appropriate for "applications that

require deep and complex queries, recommendation engines, and fraud detection systems," use cases that align with recent technologies such as—again—social media, but also eCommerce and the proliferation of online and mobile banking. Whereas traditional relational databases may not establish relationships between data, in graph databases, those relationships are "prioritized." (GeeksforGeeks, 2023)

What are two examples of graph database products and how do each work?

Two examples of graph database products include Neo4j and Microsoft Azure Cosmos DB. Neo4j describes its product as a "high-speed graph database with unbounded scale, security, and data integrity for mission-critical intelligent applications" (Neo4j, 2023). It "offers a cloud-ready architecture that scales with your data needs and minimizes infrastructure costs while maximizing performance across connected datasets," allowing to "scale out…data while taking advantage of infrastructure elasticity." It relies on a graph data model with nodes and relationships, with properties associated with each. Data is stored and accessible in a way that users can traverse relationships easily to access associated properties quickly. Neo4j uses a query language called Cypher, and also boasts ACID compliance, which supports its effectiveness in ensuring data integrity. Users are able to efficiently store and query data with highly complicated relationships, where understanding and leveraging the connections between and among data points is important. (Neo4j, 2023)

Similarly, Microsoft Azure Cosmos DB is "a globally distributed, multi-model database service [that] provides a flexible solution for building globally distributed applications with low latency and high availability" (Pesic, 2023). Since the database solution is multi-model, it is compatible with many different types of data models, including document, key-value, graph, column-family, and table models. With a global distribution footprint that allows low-latency worldwide and integration with other Microsoft Azure services, Cosmos DB provides a robust and powerful database service allowing for "streamlined application development and data processing," especially for companies and organizations that are already invested in the Microsoft family of products. (Pesic, 2023)

Both Neo4j and Azure Cosmos DB both work by providing database services that allow clients to store, organize, and understand data. Being that they are graph database solutions, these services are largely in the context of data that is connected to other data, where the relationships among the data points are important. The services provide the ability to store and organize data, search the data, scale up the system when more database capabilities are required, access the data from virtually anywhere, and more. The companies accomplish these services behind the scenes by using graph data modeling to store the data as nodes and relationships across company servers, likely

in a highly distributed way. The companies rely on a query language (e.g., Cypher for Neo4j) to search and query the data and engage in standard industry practices such as data replication, security, and compliance to ensure client data is kept securely and in a way that complies with laws and best practices.

What are the advantages and disadvantages of graph databases?

In relational databases, individual data points may be seen as the most important aspects of the database, and rightfully so, as data validity and integrity are essential to good data management. However, in graph databases, the relationships between and among data entities may be just as important, particularly for some of the use cases mentioned above—such as social media, in which a schema and structure may change or evolve over time—and a graph database provides that flexibility. The relationships among data points "often cannot be easily represented in a traditional relational database" (GeeksforGeeks, 2023). It may be argued that "relational database models become less optimal as the volume and depth of relationships increase…[Resulting in] data duplication and redundancy—multiple tables need processing to discover query results" (Amazon, 2023, "What is a graph database?"). Further, graph databases can be "shorter and more efficient at generating the same reports compared to relational databases," since they utilize linked nodes, which may require less time complexity than traversing the data in a relational database (Amazon, 2023, "What is a graph database?"). In summary, these advantages of graph database are in terms of flexibility, performance, and efficiency.

Disadvantages of graph databases must be understood before making an informed decision about the type of database that would be most appropriate for a given situation. Rund (2017) identifies several shortcomings. First, "graph databases are not as useful for operational cases because they are not efficient at processing high volumes of transactions and they are not good at handling queries that span the entire database." Further, a graph database is a powerful tool, but its core purpose is different than that of relational or NoSQL databases. Thus, it may be necessary to "combine a graph database with a relational or NoSQL database" in order to carry out critical business needs such as storing and retrieving data about customers or suppliers; "using a graph database alone is not a [master data management] solution." Additionally, graph databases do not enhance the quality of relational data; they "do not create better relationships. They simply provide speedy data retrieval for connected data." Thus, it is important that the source of the data be meaningful and accurate in order for the graph database to reflect that value. Finally, graph databases may allow for reduced complexity when searching for individual nodes, but when "mass analytics" are important and queries must be made "across all the relationships and records," these tasks may be

more difficult or time consuming when using a graph database. Rund sums up the most enticing utility of graph databases as this: "If you want to consume relationships at high speed, absolutely put those relationships in a graph." (Rund, 2017)

Conclusion

SQL, NoSQL, and graph databases are three highly valid, proven, and important concepts in the modern world of data processing, storage, and analytics. As use cases continue to rely on more and more data and processes at the business level and in analytics themselves become increasingly complex, these technologies and their related platforms, tools, vendors, and users will continue to grow and evolve. Because of the sheer size of its userbase, SQL is likely to persist for a long time to come, but the subsequent rise of NoSQL and graph databases shows that needs continue to be varied, and solutions must—and will—be responsive to those needs of the market.

References

Amazon (2023). DynamoDB. Retrieved from https://aws.amazon.com/dynamodb/.

Amazon (2023). NoSQL. Retrieved from https://aws.amazon.com/nosql/.

Amazon (2023). What is a graph database? Retrieved from https://aws.amazon.com/nosql/graph/.

Couchbase (2023). Server. Retrieved from https://www.couchbase.com/products/server/.

Couchbase (2023). NoSQL Databases: Why successful enterprises rely on NoSQL. Retrieved from https://www.couchbase.com/resources/why-nosql/.

Foote, K.D. (2022). NoSQL databases: Advantages and disadvantages. *Dataversity.* https://www.dataversity.net/nosql-databases-advantages-and-disadvantages.

Geeks for Geeks (2023). Introduction to graph database on NoSQL. Retrieved from https://www.geeksforgeeks.org/introduction-to-graph-database-on-nosql/#.

IBM (2023). What are NoSQL databases? Retrieved from https://www.ibm.com/topics/nosql-databases#:~:text=NoSQL%2C%20also%20referred%20to%20as,structures%20found%20in%20relational%20databases.

Neo4j (2023). Neo4j Graph Database. Retrieved from https://neo4j.com/product/neo4j-graph-database/.

Oracle (2023). What is NoSQL? Retrieved from https://www.oracle.com/database/nosql/what-is-nosql/.

Pesic, M. (2023). DB-Engines ranking: Top graph databases you should use. Retrieved from https://memgraph.com/blog/db-engines-ranking-top-graph-databases.

Rund, B. (2017). The good, the bad, and the hype about graph databases for MDM. *The Data Warehousing Institute.* Retrieved from https://tdwi.org/articles/2017/03/14/good-bad-and-hype-about-graph-databases-for-mdm.aspx.