Ben Albright

A NoSQL database is a database design approach that allows for the storage and querying of data that is different than what is done in traditional SQL database. NoSQL databases keep data within one data structure, like a JSON document. (https://www.ibm.com/topics/nosql-databases)

This kind of setup allows for rapid scalability to manage large, usually unstructured data sets. Companies have begun using NoSQL for these reasons as well as high performance speeds and ease of use. (https://www.ibm.com/topics/nosql-databases)

NoSQL is also a kind of distributed database, which means that information can be copied from the database and stored elsewhere, either locally or remotely. This functions as an important backup plan, because if something happens and some of the data goes down, there rest of it would still be accessible and functional. (https://www.ibm.com/topics/nosql-databases)

Cassandra is a NoSQL distributed database. This means that it can run on multiple machines at the same time while appearing as a single database to users. This also means that it can run on multiple nodes, with each node representing an instance of Cassandra. (https://cassandra.apache.org/_/cassandra-basics.html)

These nodes can communicate with each other through peer-to-peer communication. The database has a masterless architecture, which means that any node in the database has just as much functionality as any other node. (https://cassandra.apache.org/_/cassandra-basics.html)

Because Cassandra is based on nodes, it scales horizontally by adding more nodes as needed with no downtime. (https://cassandra.apache.org/_/cassandra-basics.html)

MongoDB is an open-source NoSQL database that uses flexible documents instead of rows and tables to store and process data. It provides elastic data storage which allows users to store and query various data types. It is highly scalable and allows for cross-platform applications and services. (https://www.ibm.com/topics/mongodb)

MongoDB's allows for the storing of back-end application data on Apple and Android devices, as well as the cloud. It also shares the processing of large amounts of data across multiple virtual machines, which eases the workload between them. (https://www.ibm.com/topics/mongodb)

This is a form of horizontal scaling called sharding, and it avoids the issues of vertical scaling while expanding overall data storage capacity. (https://www.ibm.com/topics/mongodb)

The database uses query language like that of SQL databases and ad hoc queries that do not require predefined schemas, making it easy to use for novice developers. MongoDB also supports multiple popular programming languages, such as Python, PHP, Ruby, Node.js, C++, Scala, and JavaScript. https://www.ibm.com/topics/mongodb)

HBase is a column-oriented NoSQL database that runs on top of the Hadoop Distributed File System. It is primarily designed for real-time data processing and random read/write access to large volumes of data. Applications are written in Java and do not support a structured query language like SQL. (https://www.ibm.com/topics/hbase)

The database is designed to scale linearly and, much like a traditional database, is made up of tables with rows and columns. Each table has an element which acts as its primary key, which allows for access to that given table. (https://www.ibm.com/topics/hbase)

NoSQL databases provide a way to store data from different sources with different formats with little to no need for transformations when the data is being stored or retrieved. (https://www.dataversity.net/nosql-databases-advantages-and-disadvantages/)

These databases use the internet and cloud as support systems, which make it possible to spread out the data storage and computing processes over multiple computers. Capacity is increased by adding more computers. (https://www.dataversity.net/nosql-databases-advantages-and-disadvantages/)

NoSQL databases offer flexibility for developers, making it easy to modify the database for new forms of data. They all offer the same basic properties, which include supporting easy updates to schemas, processing large volumes of data at high speed, scalable storage, and storing structured, semi-structured and unstructured data. These are all advantages not offered by relational databases. (https://www.dataversity.net/nosql-databases-advantages-and-disadvantages/)

These databases are also designed to be very developer friendly. The use and development of NoSQL databases has been mainly focused on developers who found they could easily develop and modify specific applications using NoSQL. (https://www.dataversity.net/nosql-databases-advantages-and-disadvantages/)

The databases use JSON to turn the data into something resembling code, which allows the developers to control the structure of that data more easily. This data can also be stored in its native format, which means that developers are not required to reformat that data to store it. (https://www.dataversity.net/nosql-databases-advantages-and-disadvantages/)

This also means that not as many applications need to be developed or purchased when launching a new database. (https://www.dataversity.net/nosql-databases-advantages-and-disadvantages/)

NoSQL databases are not without their shortcomings however, and the lack of a standardized structure can make the data untrustworthy or difficult to organize. Part of the difficulty of organization comes from the issue that NoSQL does not come with SQL. (https://www.dataversity.net/nosql-databases-advantages-and-disadvantages/)

SQL is designed to organize data and has been used for a long time, while NoSQL is newer and was designed for researching large amounts of disorganized, stored data that has been collected from multiple sources. (https://www.dataversity.net/nosql-databases-advantages-and-disadvantages/)

Since each NoSQL uses its own unique scheme, or no schema at all, it can be an issue for developers to decide which database would be the most appropriate, as they would all have their own strengths and weaknesses that would need to be considered beforehand. (https://www.dataversity.net/nosql-databases-advantages-and-disadvantages/)

NoSQL databases also lack atomicity. Atomicity is an all-or-nothing approach to a process, where the changes will only occur if all operations of that process are completed successfully. If data was being moved from one place to another, but failed partway through, nothing would change. NoSQL databases lack this kind of protection. (https://www.dataversity.net/nosql-databases-advantages-and-disadvantages/)

A lack of consistency is another issue. Consistency checks that any information to be transferred is valid both before and after that process is completed. In a case where money is being transferred from one account to another, consistency would ensure that the balance of both accounts and the amount of money being transferred would be correct at the beginning and end of that transaction. (https://www.dataversity.net/nosql-databases-advantages-and-disadvantages/)

NoSQL's lack of consistency means that there could be errors when data is transferred in this way. (https://www.dataversity.net/nosql-databases-advantages-and-disadvantages/)

NoSQL also lacks isolation, which is what dictates when and how changes that have been made to an operation become visible to parallel operations. It also allows for multiple processes to happen at the same time without any of them degrading or conflicting with one another. (https://www.dataversity.net/nosql-databases-advantages-and-disadvantages/)

Another disadvantage that makes NoSQL less trustworthy than SQL is a lack of durability. Durability ensures that data would be safe in the event of a database crash or other catastrophic event. It is normally backed up by keeping information stored on a safe storage medium. (https://www.dataversity.net/nosql-databases-advantages-and-disadvantages/)

Despite its shortcomings, NoSQL is still a beneficial resource in appropriate situations. While they may lack consistency, this also means that if one fits a developer's needs particularly well, that it could be of much more use than a typical SQL database. (https://www.dataversity.net/nosql-databases-advantages-and-disadvantages/)

They are still good at handling large amounts of data for research and development and should be used in such a way that is related to the needs of the organization or project. (https://www.dataversity.net/nosql-databases-advantages-and-disadvantages/)

A graph database is the systematic collection of data with an emphasis on relationships between different data entities. These databases are well-suited for modeling real world scenarios due to their efficient performance and flexibility. (https://aws.amazon.com/nosql/graph/)

Graph databases represent their data sets using nodes, edges, and properties. The nodes are instances or entities of data which are representative of any object to be tracked. (https://phoenixnap.com/kb/graph-database)

Edges are the representations of the relationships between the nodes. They can be either unidirectional or bidirectional depending on the kind of relationship. (https://phoenixnap.com/kb/graph-database)

Properties are the descriptive information associated with nodes. Edges can have properties as well. (https://phoenixnap.com/kb/graph-database)

These graph databases have a myriad of uses, including sophisticated forms of fraud prevention. Graph databases can process financial transactions in almost real time, which, with fast graph queries, can detect that a potential purchaser is using the same email address and credit card as in a known fraud case. (https://aws.amazon.com/nosql/graph/)

They can also help detect fraud by using relationship patterns, like seeing that multiple people are associated with the same email address or sharing the same IP address while being in different physical locations. (https://aws.amazon.com/nosql/graph/)

Graph databases are also used for routing. Information travels through the network by finding optimal paths, not unlike how a GPS navigates to a place using the shortest path from start to finish. (https://phoenixnap.com/kb/graph-database)

Graph databases can also be used in applications to provide recommendations to users. Graphs can be used to store relationships between various categories, such as interests, purchase history or friends. This allows programs to make recommendations to users based on what other users have done in the past, like products or services. (https://aws.amazon.com/nosql/graph/)

There are several advantages to using a graph database as opposed to a relational database, including flexibility, performance, and efficiency. (https://aws.amazon.com/nosql/graph/)

The flexibility of graph databases comes from the schema and structure of those models can be changed with those applications. Data analysts can add or modify existing graph structures without impacting existing functions. (https://aws.amazon.com/nosql/graph/)

Graph databases are also good at maintaining performance speeds even when the graph data volume increases. They do not have to deal with the data duplication and multiple tables needed to process and discover query results that relational databases must deal with. (https://aws.amazon.com/nosql/graph/)

Compared to relational databases, graph databases are much more efficient at generating the same results because they take advantage of linked nodes. (https://aws.amazon.com/nosql/graph/)

Traversal between nodes is very fast, because the relationship between nodes are persisted in the database instead of being calculated at query time. (https://aws.amazon.com/nosql/graph/)

JanusGraph is a graph database that is designed to support the processing of graphs so large that they require more storage and computational capacity than a single computer can provide. It provides support for very large graphs, and the graphs scale with the number of machines in the cluster. (https://docs.janusgraph.org/)

These databases are also able to handle multiple concurrent transactions and operational graph processing and answer complex traversal queries on huge graphs within milliseconds. It also provides efficient use of storage and speed of access through optimized disk representation. (https://docs.janusgraph.org/)

Unlike other NoSQL databases, JanusGraph supports ACID transactions. ACID stands for atomic, consistent, isolated, and durable. It can also handle thousands of concurrent users. There are also multiple options for storing the graph data, including Cassandra and HBase. (https://phoenixnap.com/kb/graph-database)

AllegroGraph is a horizontally scalable database technology that extracts decision insights and predictive analytics from complex, distributed data. (https://allegrograph.com/innovation-technology/)

AllegroGraph uses a combination of semantic graph and document technologies to process data with contextual and conceptual intelligence.

Graph databases are not perfect though and have several general shortcomings. There is no standardized query language across all the different graph databases. The languages vary based on what platform is used, and users may have to learn a new language to use their database. (https://phoenixnap.com/kb/graph-database)

They are also not designed for transactional based systems. The userbase for graph databases is also small compared to other database systems, so it may be more difficult to find support when running into a problem. (https://phoenixnap.com/kb/graph-database)

While not perfect, NoSQL databases have risen in popularity over the 15-20 years and continue to work towards solving the issues that relational databases have. Each NoSQL database has its own pros and cons that users need to understand before choosing one, and deciding whether that would be better than using a traditional SQL

database. Regardless of what ends up being chosen, it is clear that NoSQL databases are unique in their functionality and can be highly beneficial when used in the right situations.

Bibliography

https://www.ibm.com/topics/nosql-databases

https://www.dataversity.net/nosql-databases-advantages-and-disadvantages/

https://cassandra.apache.org/_/cassandra-basics.html

https://www.ibm.com/topics/mongodb

https://www.ibm.com/topics/hbase

https://aws.amazon.com/nosql/graph/

https://docs.janusgraph.org/

https://allegrograph.com/innovation-technology/

https://phoenixnap.com/kb/graph-database