**NoSQL VS Relational Database**

In 1960, computerized database was started, when the computers came into mainstream because it was more cost-effective choice for private enterprises. Two well known data models of that time were called: a network model CODASYL (Conference/Committee on Data Systems Languages) and a hierarchical model IMS (Information Management System). The SABRE program used by IBM to assist U.S. Airlines in the management of their reservations data was a database program and became a commercial success.

In 1970, Edgar F. Codd was a mathematician working in the IBM Research Lab. He wrote a paper titled "The Relational Model for Large Shared Data Banks," in which the world was introduced with the relational data systems we know of today. Codd’s idea changed the way people think about databases. In his model, the database structure, or logical framework, is removed from physical information storage, and this has become the common concept for database systems. Codd developed the concept of relational databases that derived influence from the mathematical relational calculus. He employed relational calculus as the database pattern. It's quick to introduce the in-memory. Nonetheless, it is extremely difficult to have a relational database running with limited quantities of memory and being required to store much of the data on the disk. Relation database was easy to implement on system if we don’t care much about storage issues and efficiency. ‘*But why relational database was still been dominating?’* Many people misinterpreted it and assumed that the relational database simply represented relationships, rather than realizing that the relational database was data conveyed as a series of tables, with named columns and rows of data, which contained almost no true relationship information, but could be used to convey any "relationship," whether useful or not. It was a catchy name. Everyone thought, "Yeah, companies are focused on relationships, and the database will be able to articulate them." People did not understand the difference between the conception of causal relationships and the theoretical nature of quantitative relationships. Everyone needs to be familiar with the newest stuff, and the word "relational" was bandied around by people who just didn't realize what it really meant. IBM was carrying the relational bandwagon because it was developed by one their researcher. So, if IBM released any product in the market people used to bought without any second thought because at that time there was no other legit company who were selling products at that high demand rates. Even before releasing of the relational database people were talking about it. The idea of pre-booking and separately selling of the product made the buzz about relational database on another higher level.

NoSQL is generally referred to as "Not only SQL." For NoSQL, unstructured, schema-less data can be processed in several collections and nodes and does not need fixed table schemas, allows minimal link queries, and scales horizontally. NoSQL database architecture stresses non-relative data access. In other words, rather than storing data in a rigidly organized table form, NoSQL uses any variety of methods — or, in some cases, a mixture of methods — to store data in a manner that is transparent like relational databases. NoSQL solves the many problem which relational database is not equipped to handle. Features such as Schema-less, relational databases have a fixed structure. Users need to go through several iterations to model data. Modifying the database structure NoSQL overcomes this constraint by having a robust structure. These repositories outline the collection of data and the internal function of the users. They provide support for the management of user-defined data structures. Flexible schema shortens the time of growth. You don't have to go through a number of iterations in data analysis & architecture. Developers can store and access whatever they want. The only downside to the schema-less architecture is that it raises the risk because there is a loss of regulation. NoSQL databases are either key-value pairs, document-based, graph databases or wide-column stores and SQL database are table-based. Maintaining high-end RDBMS systems is costly and requires skilled personnel to handle the database but less maintenance is needed for NoSQL databases. It supports other features such as automated maintenance, smoother data delivery and simplified system structures, reducing the need for NoSQL management and tuning. The NoSQL applications are cheap and open source. Implementation of the NoSQL database is simple and only requires inexpensive servers to handle the volatile data and operation whereas the RDBMS applications are complex and require massive servers and storage facilities. And in the case of NoSQL the cost of storing and manipulating data per gigabyte can be several times smaller than the cost of RDBMS. NoSQL database is schema less and data can be stored without any predefined structure in a NoSQL database. Therefore, the configuration or data model can be changed at any time, without the interruption of application and change control is a big hassle in SQL. NoSQL data output performance gets increase due to supported caching in system memory and with the relational database there should a separate infrastructure to avail this feature. NoSQL databases are horizontally scalable, which ensures they can accommodate additional traffic by merely adding more servers to the network.

NoSQL databases have the potential to become larger and more efficient, making them the chosen alternative for massive or continuously changing data sets. Since the exponential growth in the volume of unstructured data, it has become impossible to store data on a single computer. Relational Databases require sophisticated equipment to support loads without losing performance. Therefore, to scale it was important to build a program that can store data on a cluster of computers and recover it efficiently. NoSQL databases are denormalized. Query speed- Speed increases dramatically, as only a lookup on the main attribute is needed and several tables do not need to be joined and Storage & Retrieval tables — just save and get a single document. In a rational argument NoSQL is better than relational databases but it does not mean it has to be a perfect replacement for relational database. It mainly depends on the system requirement to choose NoSQL or relational database.

MongoDB is a DBMS framework (DBMS), which uses a document-based architecture for databases. C++ is compiled in MongoDB. Specific data formats provided by MongoDB. MongoDB contains data with its own binary storage structures in flat files. Data storage is thus incredibly lightweight and effective, suitable for large volumes of data. In JSON-like records, the MongoDB stores data, making the database incredibly lightweight and scalable. MongoDB is a concept of a database oriented to text. Each MongoDB database contains collections and contains documents in turn. Each document may be different and depends on the number of fields that vary. Each document's model will differ from one another in size and content. The features of the data model allow you to store structured arrays and complex in a hierarchical relation. A MapReduce is a programming model consisting of two procedures: Map) (and Reduce). (Map) (will search and sort and Reduce) (will do a summary operation. MongoDB will run over many servers. In case of hardware failure, the data are duplicated to hold the machine up and to preserve its operating state. MongoDB is a database which has no schema.

In MongoDB architecture – one set includes various papers. The dataset is schema-less, and you can compare specific material, fields, and size in the same dataset to another text. NoSQL apps match with MongoDB offering. Thus, NoSQL cannot replace MongoDB or MongoDB cannot replace NoSQL. MongoDB is a schema-free database that is simpler than traditional database tables. This has no structure, meaning even in the same set it may have several different fields, content, and size than another text. MongoDB is a schema-less database that is simpler than conventional database tables. Is written in the C++ language. It has no schema because in the same set, it may have several different fields, content and size than another text. MongoDB is a high-performance open-source database. MongoDB is a database with high availability and scalability. It facilitates quicker query response due to features such as indexing and replication. Indexing is very critical for optimizing the efficiency of search queries. MongoDB uses dataset indexing to improve database efficiency and search speed. MongoDB indexing boosts efficiency for quicker search demand. The MongoDB database can be used for indexing using primary and secondary indices. MongoDB can be used as a file system with load balancing and data replication features over several file storage devices. The replication function is the distribution of multiple nodes of data. It may have primary nodes and secondary nodes to duplicate the results. Data replication is performed using the master-slave model. MongoDB has a replication mechanism by spreading data across several devices. This method distributes data to several physical partitions called shards. Thanks to the sharding of the MongoDB automated load balancing operation. We use sharding in situations where we need to work on very large datasets.

Consistency: ensures that after the upgrade process is over, everybody will assume that the most recent iteration of the data in the database and the program is a stable system. However, a program in which modified data cannot be accessed at once by all users is understood to be fundamentally reliable. Availability: is achieved if the device is still in continuous operation.

Availability is accomplished by spreading the database as a cluster of nodes, replicating or partitioning data across many nodes. In this case, if one node fails, the other nodes will continue to operate. Partition Tolerance: a storage system that can work even though one of the nodes fails or is unavailable.

These are various types of NoSQL database management system available in the market which are shown below:

* Key-value stores with attribute name (or "key") together with its value. Examples include Riak, Voldemort, and Redis.
* Wide-column stores data from columns together instead of rows which are designed for queries over large datasets. Example: Cassandra and HBase
* Document databases pair each key with a complex document-structure. Documents can include several key-value pairs, key-array pairs, or even nested documents. Example: MongoDB
* Graph databases are used to store information about networks. Examples: Neo4J and HyperGraphDB.

Redis: It is used as a session cache one of the most obvious instances of usage for Redis. In comparison, the benefits that Redis gives durability over other session stores such as Memcached. Although holding a cache is not usually important in terms of accuracy, wouldn't most users just appreciate if all their cart sessions were gone? Fortunately, with the Redis momentum through the years, papers on the correct use of Redis for cache are very easy to locate. Even Magento has a plugin for Redis.

Cassandra: Cassandra is easy to set up and maintenance is done automatically. Even when it is expanded or when a node is attached, the portal is very simple. Cassandra also maintains the data is re-synchronized, synchronized or transmitted. The platform is well-known for delivering high-speed random read written information as opposed to other NoSQL systems as it is capable of column storage and of spreading modular architecture. It is only appropriate when secondary index criteria are that, meaning they are fully DE standardized.

Neo4J: Machine learning technology is more available to businesses today than ever before. Open source artificial learning systems and professional deep learning applications provide developers and data scientists with the resources and skills of use cases such as intelligent image recognition, voice recognition, and smart chatbots of benefit from machine learning / deep learning. Since graph-structured data essentially outstands in representing the relationships between data points, it is used to drive a commonly used machine learning application, namely recommendation engines.

Concluding this analogy, we found that in some places each form of database system appears to be efficient, relational database systems are still being utilized because there are certain usage cases where they are successful. Yet NoSQL has been a wildcard in this battle, it has developed rapidly because it has established itself on the basis of the drawbacks of the earlier one it has succeeded in removing the identified problems and has succeeded in offering solutions to the various needs of the industry.

References:

1. Burtica Practical application and evaluation of no-SQL
2. Rautmare MySQL and NoSQL db
3. [History of Relational Database](https://www.quickbase.com/articles/timeline-of-database-history)
4. [Redis Use case](https://www.objectrocket.com/blog/how-to/top-5-redis-use-cases/)
5. [Cassandra use case](https://www.edureka.co/blog/cassandra-use-cases/)
6. [Neo4j use case](https://neo4j.com/blog/5-noteworthy-use-cases-graph-technology-analytics/)