



BRIEF STUDY ON SQL AND NOSQL

¹Mr.Amit A. Patel, ²Mr.Mahesh A. Prajapati, ³Mr.Gaurang J. Patel

¹Lecturer, ²Lecturer, ³Lecturer

¹Computer / I.T Department,

¹SWAMI SACHCHIDANAND POLYTECHNIC COLLEGE, SANKALCHAND PATEL UNIVERSITY, VISNAGAR, INDIA

Abstract: Industries require high-speed processing and industry after industry moves towards digitization as a result. The vast changes brought about by the Digital Economy offer the industries a new appearance. This paper's goal is to provide a review of the CAP in GDB and the ACID properties in RDBMS, and then explain why developers and businesses usually strive to provide straightforward yet incredibly effective solutions for certain issues related to speed, data flexibility, and previously unheard-of size. The issues encountered during the SQL to NoSQL transfer will be the next area of focus.

Keywords: data Migration, NoSQL, RDBMS, Big Data

I. INTRODUCTION

When the digital age whether at home, at work, or while on the go, an increasing number of individuals are using the internet. Everything from making payments for in-person purchases to making bookings online and even shopping for groceries online—the list is endless. This tackles the following technical issues: performance, scalability, and database upkeep. Companies have been grappling with data storage issues since the turn of the 20th century. Prior to this, there was no problem with digital data because relational databases were in use. Few businesses established alternative databases for market application in response to the big data dilemma. Concerning "one size fits all" data-stores, they have reservations. Storage issues are somewhat resolved. These queries were merely a result of the data storage issue. According to Jon Travis, the sheer volume of data being produced every day is causing individuals to consider alternative technologies. Jon Travis works at Spring Source as an engineer [1]. Therefore, researchers searched for a "magical database" that could both solve the big data problem and offer answers for the majority of big data requirements due to the enormous volume of data and request velocity.

Numerous non-relational structural solutions, such Big Tables from Google and Dynamo from Amazon, have been introduced to address the scalability issue. Google and Amazon released research papers in 2006 and 2007 outlining the necessity for a new feature-rich database that is highly scalable and agile. Additionally, there was a demand for such services or tools that could handle an ever-growing user and data base [2].

Big Data is Form with three Vs: Volume, Velocity, and Variety. Researchers define the term "variety" in. He claims that volume refers to the total amount of data and velocity to the rate at which this data is produced, while variety refers to the different data formats of the data being generated and stored. Thus, a solid foundation is required for every real-time big data architecture that is introduced. An architecture that is non-relational and known as NoSQL (Not Only SQL).

Carlo Strozzi coined the term "NoSQL" in a non-structural language, yet with a relational basis. A quick, portable, relational database management system with enormous scalability is provided by the use of NoSQL [4]. These days, the phrase "NoSQL" is frequently used to refer to something other than "SQL," such as "Not only SQL," and is thought of as the foundation of non-relational storage systems. Scalability and availability are two issues that are carefully addressed in the development of most noSQL systems.

II. LITERATURE REVIEW

Ted Codd was a computer scientist at IBM's San Jose Research Laboratory in 1972. He proposed a new method of organizing data for persistent data, which he called the "relational data model." Donald D. Chamberlin and Ray Boyce were members of this reputable IBM group. SEQUEL (Structured English QUery Language) is the moniker he gave to this query language. The following hesitant version, released in 1976–1977, was soon dubbed SEQUEL/2; however, licensing issues forced the name to be changed to SQL.

SQL is a well-developed query technology that is used by many developers in the era of mainframes and corporate applications. RDBMSs gained a lot of traction right away.

In fact, a tiny, unheard-of software business called "Relational Software" delivered the first commercial implementation in 1979. This obscure software firm is now well-known as Oracle [5].

In his chapter, Johannes Zollmann discussed the RDBMS's ACID features. He adds that there are significant assurances of consistency provided by ACID characteristics.

SQL is a well-developed query technology that is used by many developers in the era of mainframes and corporate applications. RDBMSs gained a lot of traction right away. In fact, a tiny, unheard-of software business called "Relational Software" delivered the first commercial implementation in 1979[5].

Even while relational databases benefit greatly from ACID, one disadvantage of this style of database is that it was designed to operate on a single server; hence the only way to increase capacity was to upgrade the server. In his presentation, researcher Antro Salminen points out that the only method to scale out an RDBMS is to increase its hardware processing capability [3]. RDBMSs are similarly limited in terms of physical storage [2]. In order for current generation web, mobile, and other apps to function at any scale, such a database was necessary. Since necessity is the mother of inventions; researchers introduced NoSQL at a time when big data was an issue for everyone.

NoSQL is separated into four categories: graphs, documents, columns, and key values. Every database offers benefits of its own. The choice of type is contingent upon the nature of the application and its specifications. NoSQL has several advantages, including replication, demoralization, and horizontal scaling, which contribute to its power.

NoSQL provides a property of simple graph traversal operation queries, which offers it an advantage over relational databases in terms of time. Because of NoSQL's horizontal Scaling feature, we may save a lot of information that is frequently needed in one table. Additionally, NoSQL may manage every table join at the application level. Data repetition is tolerated in NoSQL, which is another characteristic. This NoSQL feature facilitates faster and more efficient query execution. Nevertheless, NoSQL does not associate tables, therefore we must be extremely cautious to synchronize the data when updating the tables.

Numerous NoSQL systems claimed to be able to overcome any performance issues because to their horizontal scalability, as Markus Winand noted on his blog. However, scalability for the databases of the present generation is restricted to write operations and is achieved through an eventual consistency mechanism. Although write operations are slowed down by SQL databases' stringent consistency architecture, this does not always indicate poor performance.

Data migration between NoSQL and SQL, based on NoSQL database architecture. Relational database specialist Chris Bird noted the syntactical differences between NoSQL and SQL. He also discovered that learning NoSQL requires mental acrobatics, which makes migration challenging. As a result, the migration process depends on the NoSQL technology that is chosen for usage. A few researchers have attempted to bridge the RDB and GDB gaps. Any type of information can be represented as a graph in a graph database, which can easily adapt to changes in the data. Some examples of graph data query languages are Cypher Gremlin.

Users are limited in what they can access and how quickly they can view in certain noSQL databases since they do not offer range queries or joins. In October 2011, an Oracle white paper stated that selecting a new system is a strategic and exciting business activity that typically involves collaborating with suppliers, opportunities, and new technology. It has also been noted that data migration, which is nothing more than toying with risk, might result in significant, minor, or any hidden legacy issues if source and target are not sufficiently understood.

The target of a migration is typically believed to be aware of the current systems and to support the same structure. However, user adoption of NoSql query structure is more difficult to achieve. In his research, researcher Patricia Cavoto found that the complex structure of relational databases contributes to the complexity that arises while analyzing data in an RDB. Additionally, the graph model for data processing is more adaptable than relational models. According to certain researches, relational databases may demand very complex and costly operations in addition to complex join operations to obtain the necessary results in relational database management systems (RDBMS), which can lead to a decline in efficiency and performance[6].

Rethinking and redesigning data migration is crucial for nearly every organization. When migrating data from SQL to NoSQL, developers must consider how to represent their current model in the new database[7]. This implies that the mechanisms for storing and retrieving data are different in the two databases.

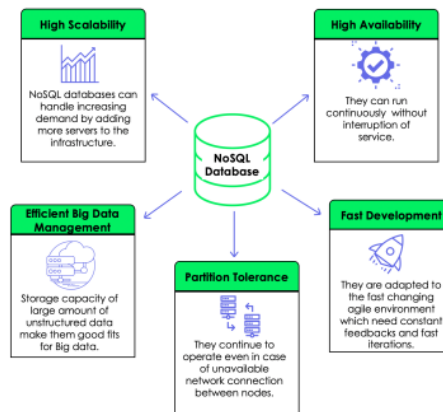


Fig.1 No SQL Database

III. Power of SQL

ACID compliance: SQL's finest feature, the ACID characteristic, lowers anomalies and safeguards the integrity of the database. **Security:** Data security is crucial while handling data in DBMSs. Regarding security, NoSQL databases lose their security because access control and permissions are granted in these systems, which implies they are less secure than SQL databases.

IV. Limitations of SQL

The accomplishment of scalability and flexibility poses a significant challenge for relational databases. Researchers have discovered that modifications to a single table can have an impact on the entire system [8]. It has been observed that a variety of factors affect how quickly a SQL query responds. Scalability and volume are two of them. These variables also have to do with how performance is dependent on data volume. Another drawback of SQL is its fixed set of columns, which makes it unsuitable for large data sets. As a result, non-relational databases are in high demand. It is noted by Leonardo Rocha and colleagues that Migration's Difficulties: Module Reconstruction

V. Conclusion

We can draw the following conclusions from the survey:

- Relational databases are the foundation of many large and medium-sized software systems now in use.
- There is a migration issue because of the differences in the syntactical and storage structures between SQL and NoSQL. SQL's performance is affected by the volume of data it stores. i.e., there is an inverse relationship between the performance and the data stored. This has the effect of making a query take longer. However, the most important topic when talking about dataset scalability is if we can increase this ratio by granting SQL of NoSQL the potential of horizontal scaling.

REFERENCES

- [1]. Computerworld "No to SQL" Anti-database movement gains steam. June 2009. – http://www.computerworld.com/s/article/9135086/No_to_SQL_Anti_database_movement_gains_steam.
- [2]. Couch DB - The Definitive Guide www.couchbase.com/-/binaries/content/assets/website/docs/whitepapers
- [3]. Antro Salminen "Introduction to NoSQL" NoSQL Seminar 2012 @ TUT
- [4]. C. Strozzi. Nosql relational database management system. http://www.strozzi.it/cgi-bin/CSA/tw7/1/en_US/NoSQL/HomePage, July 2012.
- [5]. Johannes Zollmann "NoSQL Databases" Chapter 1 Typeset August 20, 2012
- [6]. Oracal White paper "Successful Data Migration" October 11, oracal.com Copyright © 2011, Oracle and/or its affiliates.
- [7]. Markus Winand "Response Time, Throughput and Horizontal Scalability" blogpost - use-the-index <http://use-the-index.lukeA.Gupte.com> to database performance for developers.
- [8]. Hiring Headquarters "SQL vs. NoSQL Databases: What's the Difference" blog.upwork.com