Comparative Study of SQL & NoSQL Databases

Supriya S. Pore[1], Swalaya B. Pawar[2]

[1]*Master of Computer Application, IMCOST, Mumbai University, Mumbai, India*.
[2] *Master of Computer Application, IMCOST, Mumbai University, Mumbai, India*.

Abstract— ***SQL (Structured Query Language) is a standard query language for relational database management system. Different types of RDBMS such as MS Access, Oracle, MySQL, Informix, SQL Server, Sybase uses SQL as their standard database query language. NoSQL is the Not Only SQL is a collection of non-relational data storage systems. All NOSQL offerings relax one or more of the ACID properties. Today SQL databases become an integral part of IT infrastructure of any organization. For example, MySQL is a Relational Database based SQL implementation for the web, now uses in very large-scale websites such as Facebook etc. The main focus of NoSQL databases is more on BASE (Basically, Available, Soft state Eventually Consistent) than RDBMS ACID (Atomicity, Consistency, Isolation, and Durability) properties. Some of the NoSQL databases provide advanced features namely Sharding, which takes Database Partitioning to a new level in the form of horizontal scalability and availability. Some of the NOSQL databases are Cassandra, CouchDB, Hadoop & Hbase, MongoDB, StupidDB. The aim of this research paper is to evaluate and compare these two SQL & NOSQL databases and tries to answer which of these is better in terms of its performance and scalability.***

Index Terms— ***RDBMS, NoSQL, ACID properties, MongoDB, Document Stores, Key-value stores, Column-Oriented Databases, Graph Databases, Object-Oriented Databases.***

I. INTRODUCTION

Today, the rapid growth of computer and internet causes an issue of efficient storage and retrieval of data. Extremely large amounts of online transactions and experimentation result in massive amounts of data which require well organized storage solutions. Databases play an important role in order to satisfy this need of storing and retrieving data in an organized manner.

Two most commonly used database types are SQL and NoSQL databases. The fundamental concept of SQL database is that, the SQL database is nothing but Table based i.e. Relational Database. The important concepts involved in relational databases were laid out by Edger Codd in order to overcome the disadvantages of the previous linked lists implementations in databases. NoSQL databases started gaining popularity in the 2000's when companies began investing and researching more into distributed databases.

A SQL database is a relational database which is strictly based on relation (tables) to store data. A relation (table) in a relational database is divided into set of rows and columns. Each column represents a field and each row represents a record. Tables can be related and linked with each other with the use of foreign keys or common columns. On an abstract level tables represent entities, like Employees, or Department etc. This abstraction is helpful when the designing the database schema as real world objects need to be mapped to the database in addition with the relation between them. However, relational databases having the variety of limitations due to constant growth of stored and analysed data, e.g. the restrictions on scalability and storage, and efficiently losing of query as the volume of data is very large, and the storing and managing of larger databases become challenging.

Google, Amazon, Facebook, and LinkedIn are among the first companies to discover the serious limitations of SQL database technology for supporting big data and big user's requirements. So the new database is formed in order to overcome these limitations which are known as NoSQL. One of the most important features of NoSQL databases is that they have no fixed schematic

ISSN: 2278 – 1323

structure, records can have different fields as per the requirements, and this is called as a dynamic schema. NoSQL databases provide latest feature called as sharding which takes Database Partitioning to a newer level as in the form of horizontal scalability and availability. The drawbacks of NoSQL databases are lack of RDBMS support for end user querying, limited integrity constraints like foreign key at the structure level and limited support for transaction processing.

II. AXIOMATICS OF SQL & NoSQL DATABASES

A.Axiomatics of SQL Databases:

1. ACID Properties:-

For the reliability of transactions SQL databases support ACID (Atomicity, Consistency, Isolation, and Durability) properties. Each of the ACID property is explained in the context of databases as follows:-

- **Atomicity**: When there are updations to the database, it should be either commit or abort.

- **Consistency**: The integrity of the database must be maintained by all transactions. The database must not be in inconsistent state, once the transaction gets completed.

- **Isolation**: The transaction should be isolated so there is no conflict between concurrent transactions.

- **Durability**: If the system or failure of storage media occurs then system must recover the committed transactions. The updations to the database should never be lost.

2. Normalization of schema:-

Normalization involves 3 basic forms which are as follows:

- **First Normal Form (1NF):** Groups of repeated data are eliminated and new tables are created for each group of related data where each table is identified by a primary key.

- **Second Normal Form (2NF):** For multiple records with same set of values then such records are move to the new table and these two tables are link with a foreign key.

- **Third Normal Form (3NF):** Fields which are not dependent on primary key of a table should be removed and place into another table if necessary.

It is also important for a database schema to satisfy the 2NF to first satisfy 1NF and the same applies for 3NF correspondingly.

3. Vertical Scalability:-

SQL databases support vertical scalability. They are scaled by increasing the horse-power of the hardware.

B. Axiomatics of NoSQL Databases:

1. BASE (Basically, Available, Soft state Eventually Consistent) Transactions:-

BASE is an opposite of ACID. The state that gets after transaction consistency is soft state and not a solid state. The BASE focuses on permanent availability. E.g. Online Railway Reservation.

2. CAP Theorem:-

CAP Theorem involves following three properties: Consistency, Availability, and Partition -tolerance. It is proposed by Eric Brewer.

- **Consistency:** The data present on all machines must be same in all updations to be made on all machines frequently i.e. consistent data.

- **Availability:** Data should be available permanently and not temporarily i.e. it should be accessible all the time i.e. availability.

- **Partition-tolerance:** In case of machine failure or any faults occurred in the machines database should work properly without taking any halt i.e. partition tolerance.

3. No schema required: - Data can be inserted in a NoSQL database without pre defining database schema. The structure of the data being inserted can be modified at any time, without application interruption.

4. Auto elasticity: - Without requirement of application assistance**,** NoSQL automatically distribute your data onto multiple servers.

5. Integrated caching: - NoSQL database cache data in system memory, in order to increase data through and increase the performance advance.

III. TYPES OF SQL & NoSQL DATABASES

A.SQL Databases:

1. MySQL Community Edition:-

MySQL Database is commonly used, popular open-source database. It is normally been used with apache and PHP, although it can be also used with server side java scripting technique using Node js. The following are the few MySQL benefits and strengths:

- **Replication**: By replicating MySQL database across multiple hosts and servers the work load can be reduced heavily. This increases increasing the scalability and availability of business application.

- **Sharding:** MySQL Sharding operating system is beneficial when there is large number of write operations needs to be performed in a high traffic website. By performing sharding of MySQL servers, the application is divided into multiple servers partitioning the database into small chunks. Servers having comparatively low cost can be deployed for this reason.

- **Memcaching of a NoSQL API to MySQL**: Memcaching can be used to improve the performance of the data retrieval operations which gives an advantage of NoSQL API to MySQL server.

- **Maturity**: This database has been used for a long time and wide range of community input and testing has gone into this database making it very stable.

- **Wide range of Platforms and Languages**: MySQL database is available for all major operating systems like Windows, Linux, Mac, BSD and Solaris. MySQL is also capacity to connect with various programming languages such as Node.js, Ruby, C, C++, C#, Java, PHP etc.

- **Cost effective**: MySQL is available free of cost and open source database, hence it can be widely used in most of the database applications.

2. MS-SQL Server Express Edition:

MS-SQL is a product of Microsoft having good reliability, scalability, stability features. It is powerful and user friendly database. The following are the few MS-SQL benefits and strengths:

- **Integrated Development Environment (IDE) support:** For efficient development and in order to increase developer's productivity, there are various tools available such as Microsoft visual studio, SQL Server Management Studio and Visual Developer tools.

- **Disaster Recovery mechanism**: By providing database mirroring mechanism, fail over clustering technique MS-SQL provides efficient disaster recovery mechanism.

- **Cloud back-up support:** Microsoft supports Cloud back-up by providing cloud storage while performing cloud-back of MS-SQL database.

3. Oracle 11g Express Edition Database:

This database is for development, deployment, and to distribute. It is having simpler administration and is fast to download. The following are the few Oracle benefits and strengths:

- **Easily Upgrade**: Oracle database can be easily upgraded to the new and advanced versions.

- **Vast platform support**: It supports multiple operating systems such as Linux and Windows.

- **Scalability**: Scalability of Oracle database is not cost effective. It also provides the facility which is easily manageable and productive also secure and reliable.

B. NoSQL Databases:

ISSN: 2278 – 1323

1. Key-Value Store Databases:-

A Key-Value Store database is one of the types of NoSQL databases which are having easy and user-friendly Application Programming Interface (API). There is no fixed schema (Schema less) for KV databases. As the name suggest, the KV data store consists of two parts, a string which shows key and the actual data which shows value. The KV databases are like hash tables where the keys are used as indexes. It is found that KV databases are comparatively faster than RDBMS. KV databases can be found in Online Shopping Websites etc.
Example of Key- Value Store Databases:-

- **Amazon DynamoDB**

One of the newly released and fully managed KV database is Amazon DynamoDB which is offered by Amazon and implemented using Amazon's Dynamo Model. It is design especially for internet scale applications. It provides high reliable, fast and cost-effective service of NoSQL database. It stores data on Solid State Drives. So it provides fast access to data. Instead of storing data on traditional hard drives, it stores data on solid state drives, so the data access is faster. In this database predictable latencies are offered at any scale. It provides synchronous data replication across number of AWS Availability Zones in an AWS region. Thus it provides high availability and data durability also in complex failure conditions.

- **Riak**

Basho technologies developed RIAK by using C, Java script, and Erlang. Riak uses principles from Amazon's Dynamo Paper for its implementation. Different components of Riak include Riak clients, Riak search, Web machine, Riak KV, Riak core, Riak SNMP/JMX, Riak Pipe etc. Riak is open source, distributed and offers partition tolerance as well as persistence. Riak can be used for collecting and checking Point of Sales i.e. POS data, maintaining user's personal information on social networking sites etc.

2. Column-Oriented Databases:-

Data stored in Column-Oriented databases in the form of whole column rather than a row. This decreases the disk access compared to relational table, which consists of column and rows with uniform sized fields for every record.
Example of Column-Oriented Databases:-

- **HBase**

HBase is developed after Google's BigTable as well as is open source, distributed and non-relational database written in Java. HBase provides BigTable-like capability for Hadoop as it is modeled as a part of Apache Software Foundation's Apache Hadoop Project. It is also run on top of Hadoop Distributed File system i.e. HDFS. HBase features include in-memory operation, compression and Bloom filters on a per column basis. Several data-driven websites like Facebook's Messaging are now serving by HBase.

- **Cassandra**

Like HBase, Cassandra was also developed using Java. It was formed by Apache Software Foundations and released in 2008.It involves the concepts of key-value stores and column-store database as it is based on Amazon's Dynamo Model and Google's Big table. Some of the features of Cassandra are dynamic schema, partition tolerance, high availability, persistence, high scalability etc. Applications that use Cassandra are banking and finance, social networking websites and real time data analytics etc. Cassandra is also used by Adobe, Twitter, eBay etc. The drawback of Cassandra is that read operations are comparatively slower than write operations.

3. Document Store Databases:-

Document Store Databases stores the data in the form of documents. Document Store databases are schema less, so they are much more flexible compared to the records in relational databases. Format of documents are PDF, XML, JSON etc. Each document in document store databases is addressed by unique key for representation of that document. It is used in such applications where data need to be store as documents that having some special characteristics.
Examples of Document Store databases:-

- **MongoDB**

MongoDB is proposed by 10gen company in order to handle growing data storage needs. MongoDB is open source NoSQL document store database which is written in C++. MongoDB uses JavaScript as its query language. Data is stored in MongoDB in the form of collections. Each collection consists of documents. MongoDB stores the documents in BSON format which is binary form of JSON. BSON supports different data types such as integer, float, string, Boolean, date etc. MongoDB is schema-less as it is having document structure. To distribute the collections across multiple nodes MongoDB offers sharding technique. MongoDB automatically redistribute the data across the nodes, thus the load is balanced and equally distributed over the nodes. MongoDB allows Master-Slave replication

technique. Here the Slaves are the nodes that contain the copies of Master nodes and it is used for backup process and read operations.

- **CouchDB**

CouchDB was proposed by Apache Software Foundation and developed using C++. It was released in 2005. For storage of data, it uses JSON documents as well as to create and update database documents, CouchDB provides RESTful HTTP API. For administration process it provides built in web application known as FULTON and uses JavaScript as a query language. CouchDB provides concurrent access to users by implementing Multi- Version Concurrency Control i.e. MVCC. It is fault tolerant, persistent, and highly available as well as having efficient replication and synchronization capabilities. It is helpful in applications where data changes occasionally on which pre-defined queries should be used. CouchDB is being used for CMS system, Customer Relationship Management (CRM), Facebook apps like Horoscope etc. Few Disadvantages of CouchDB are no support for ad-hoc queries, temporary views in CouchDB are very slow on large datasets etc.

4. Graph Databases:

In Graph Databases, data is store in the form of a graph. Graph contains nodes which act as objects and edges that represent the relationship between the objects. Graph is also having properties which are related to nodes. Every node in a graph database consists of a direct pointer that point to the adjacent node, this technique is known as index free adjacency. By using this technique millions of records can be traversed. Graph databases don't have pre-defined schema and focus on connection between the data. They allow efficient storage of semi-structured data. Graph databases are faster than relational databases and easily scalable as queries are expressed as traversals. Graph databases offers rollback support and are ACID compliant. Graph databases used in number of applications like content management, social networking sites, bioinformatics, cloud management etc.
Example of Graph Databases:-

- **Neo4j**

Neo Technology was released in 2007 and developed using Java. MongoDB was developed using Neo Technology. It provides flexible network structure. Neo4j is object oriented, high performance graph database. Neo4j uses property graph data model which composed of nodes and relationship with their properties. Neo4j is highly available, ACID compliant, scalable and reliable graph database. It provides REST interface and Java API. It is also possible to embedded into JAR files. Query language used by Neo4j is CYPHER. Neo4j helpful in applications which are having complexity in relationships such as recommendation engines, social networking sites etc. It is not possible to use Sharding in Neo4j. If relationships do not exist among the data then Neo4j should be avoided to use. Neo4j used by many companies such as Adobe, Lufthansa, Cisco, Mozilla, Accenture etc.

5. Object Oriented Databases

In Object Oriented databases the data to be stored is represented as an object. Object Oriented database is a combination of object oriented programming and database characteristics. Object oriented data store offers OOP characteristics such as abstraction, encapsulation, polymorphism etc. In this database the class, objects, and class attributes are compared to table, tuple and columns in RDBMS respectively. Each object consists of object identifier for uniquely representing that object. Objects are possible to directly retrieved using pointers; hence data access is comparatively faster. Object oriented databases uses agile as a software development methodology.
Example of Object Oriented Database:-

- **db4o**

In 2000, the Carl Rosenberger was started db4o and this product was initially shipped in 2001. It was commercially launched as Db4jobjects Inc. in 2004 and then it was acquired by Versant Corporation in 2008. Java and C# are used for the development of db4o. It offers a GUI known as Object Manager Enterprise for various reasons like browsing databases, administrative functions, database connection, building queries. It also offers NQ (Native Queries) so that the users can use common object oriented programming languages like Java, C# etc. and not query languages like SQL. Db4o allows the user to store an object in a single query. It also offers db4o Replication System that allows synchronizing relational backend with db4o. The disadvantage of db4o is that there is no built in support to export or import data from JSON, XML or text files provided by other data stores. There is no support for referential integrity and OLAP tools. Db4o are used by the companies such as IBM, BMW, Intel, Seagate etc.

IV. COMPARISON BETWEEN SQL & NoSQL DATABASES

- SQL databases are the synonym for Relational Databases (RDBMS); whereas NoSQL databases are primarily called as non-relational or distributed database.

ISSN: 2278 – 1323

- **Development History:** SQL Databases developed in 1970s to deal with first wave of data storage applications; whereas NoSQL databases developed in 2000s to deal with limitations of SQL databases, particularly concerning scale, replication and unstructured data storage.

- **Types:** SQL Databases are table based databases (One type [SQL database] with minor variations.). NoSQL databases includes key-value databases, document store databases, wide-columnar stores, and graph databases.

- **Schemas and Flexibility:** In SQL databases each record conforms to fixed schematic representation, meaning the columns must be decided and locked before data entry and each row must contain data for each column. It involves altering the whole database and going offline. In NoSQL schemas are dynamic. Information can be added on the fly, and each 'row' doesn't have to contain data for each 'column'.

- **Scalability:** SQL Databases are basically vertically scalable; it means a bigger server, which can get very expensive. RDBMS can be implemented across multiple servers, but this one of the difficult processes and time consuming approach. NoSQL databases are horizontally scalable, meaning across servers. These multiple servers can be cheap commodity hardware or cloud instances, making it a lot more cost-effective than vertical scaling. Many NoSQL databases also distribute data across servers automatically.

- **ACID Compliancy:** The vast majority of SQL databases are ACID compliant. Varies between technologies, but many NoSQL solutions sacrifice ACID compliancy for performance and scalability.

- **Query Language:** SQL databases use SQL (Structured Query Language) for defining and manipulating the data. In NoSQL database, queries are focused on collection of documents and not on tables. Sometimes it is also called as UnQL (Unstructured Query Language). The syntax of using UnQL varies as the database changes i.e. different syntax for different databases.

- **SQL database examples**: MySQL, MS-SQL Oracle, and SQLite,. **NoSQL database examples**: MongoDB, CouchDb, BigTable, Redis, Cassandra, Hbase and Neo4j.

- **For complex queries**: SQL databases are good fit for the complex queries; whereas NoSQL databases are not good fit for complex queries. On a high-level, NoSQL does not provide standard interfaces to perform complex queries, and the queries in NoSQL are not as powerful as SQL query language.

- **For the type of data to be stored**: SQL databases are not much good fit for hierarchical data storage. NoSQL database are comparatively better for the hierarchical data storage as it follows the key-value pair way of storing data content which is similar to JSON data. NoSQL database are mostly preferred for large data set (i.e. for big data). Hbase is one of the examples for this.

- **For high transactional based application**: SQL databases are more useful for heavy transactional type applications, as it is more stable and provides the atomicity as well as integrity of the data. While it is possible to use NoSQL for transactions purpose but it is still not comparable and usage effective in high load and for complex transactional applications.

- **For support**: Excellent support is provided for all SQL databases from their vendors. There are also good, effective and independent consultations that can help you with SQL database for very large scale deployments. For some of the NoSQL databases you still have to dependent on community support, and there are only few outside experts are available for you to setup and deploy your large scale NoSQL deployments.

- **Data Manipulation:** Data manipulation activities in SQL databases is done by using Selection, Insertion, and Updation statements, e.g. SELECT columnname FROM tablename WHERE condition… In NoSQL databases data manipulation is done using different object-oriented APIs.

V.CONCLUSION

The main aim of this research paper is to evaluate the basics of SQL and NoSQL databases and the comparative analysis of these two databases. This paper also describes the Axiomatics of SQL and NoSQL databases. ACID property is not used in the NoSQL databases because of data consistency. This paper also describes examples of SQL databases and types of NoSQL databases on the basis of CAP Theorem. Databases are horizontally scalable in case of NoSQL databases and vertically scalable in case of SQL databases. As the two databases (SQL and NoSQL) behave differently according to the type of queries used

ISSN: 2278 – 1323

the choice of which database to use lies on the type of application the system will be using. Performance of both the databases is depending on the database size and the type of queries which will be performed by the applications.

VI.SUGGESTIONS

- . Standardization of queries in NoSQL is still not implemented. There should be some specific compiler or middleware needs to be implemented in coming future. So that using NoSQL databases becomes easier.
- NOSQL databases are most widely used in online applications. It is still no much used in standalone database applications. There is some organized way needs to be implemented for NOSQL databases so that it can completely ready to replaced RDBMS.
- NOSQL databases are not good fit for complex queries. There is some organized way needs to be implemented for NOSQL databases so that it can able to deal with complex queries.
- NOSQL databases allow columns to be NULL. This may lead to loss of important data. So, there is some specific mechanism which emphasis for all columns must contain data for each row.
- There is nothing like database developer for NoSQL databases. Front end application developer needs to build back end connectivity with NOSQL (e.g. MongoDB). As a simple query in NoSQL requires complex programming experts, there needs to be simplicity in NoSQL queries so that database developed by database developer can be easily connected with BI logic developed by application developer. This results in faster application creation.

REFERENCES

[1]. International Journal of Advanced Research in Computer Science and Software Engineering, 'SQL and NoSQL Databases' by Vatika Sharma, Meenu Dave.

[2]. International Journal of Applied Information Systems (IJAIS), 'Types of NOSQL Databases and its Comparison with Relational Databases' by Ameya Nayak, Anil Poriya, Dikshay Poojary.

[3]. International Journal of Science & Engineering Research, 'Modeling and Querying Data in MongoDB' by Rupali Arora, Rinkle Rani Aggarwal

[4]. SQL vs NoSQL Databases Differences Explained with few Examples DB by LUKE P. ISSAC on JANUARY 14,2014.
http://www.thegeekstuff.com/2014/01/sql-vs-nosql-db/

[5]. 'RDBMS vs NoSQL: Performance and Scaling Comparison' by Christoforos Hadjigeorgiou

http://static.ph.ed.ac.uk/dissertations/hpc-msc/2012-2013/RDBMS%20vs%20NoSQL%20-%20Performance%20and%20Scaling%20Comparison.pdf

[6].
http://www.thewindowsclub.com/difference-sql-nosql-comparision

[7]https://www.digitalocean.com/community/tutorials/understanding-sql-and-nosql-databases-and-different-database-models

Author's Profile

Supriya Pore is currently studying in Mumbai University at IMCOST, Thane since 2013, currently pursuing MCA with excellent academics. She is having interest in Studying New Technologies.

Swalaya Pawar is currently studying in Mumbai University at IMCOST, Thane since 2013, currently pursuing MCA with excellent academics. She is having interest in Studying New Technologies.

ISSN: 2278 – 1323