Comparative Analysis of Different Nosql Databases

Dharmender Kumar[1], Vikrant Singh Bhardwaj[2], Dr. K. L. Bansal[3]
[1, 2, 3] Dept of Computer Science
[1, 2, 3] HPU Shimla

Abstract- *In this emerging world of technology, volume of data is increasing day by day. This increased amount of data is becoming complex to handle. Only RDBMS is not sufficient to control, retrieve and analyse the Big data. Introduction of structural, semi structural and unstructured data has challenged the flexibility, scalability, and processing capability of the relational RDBMS (Relational database management system). The new eras of frameworks demand horizontal scaling of the databases. But Relational databases were not able to provide the horizontal scalability. To handle this problem a new Data base management mechanism is designed which is NoSQL. This paper investigates the Need of NoSQL databases and how NoSQL databases have become an alternative to relational databases. In this paper Major differences between RDBMS and NoSQL are discussed. The four categories are compared on the basis of functional and non-functional requirements. Then one database is selected from each category for further analysis.*

Keywords- NoSQL, MongoDB, Unstructured data, CouchDB, Cassandra, Neo4j, Redis

I. INTRODUCTION

NOSQL was introduced by Carlo Strozzi in 1998. NoSQL, which stand for "not only SQL," is a substitute to traditional relational databases in which data is placed in tables and data schema is carefully designed ahead the database is built. NoSQL databases are especially useful for working with huge sets of distributed data. It is one of another type of data storage databases that is used to store large amount of data storage like data in facebook which keeps on increasing every day. Applications of NoSQL databases are taken different fields such as Real-Time Big Data, Content Management and Delivery, Mobile and Social Infrastructure, User Data Management, Data Hub, Social networks, Internet of Things etc.

NoSQL doesn’t follow the ACID properties. ACID basically stands for Atomicity, Consistency, Isolation and Durability. NOSQL depends upon a softer model named as the BASE model. BASE model contains the flexibility offered by NOSQL databases and similar approaches to the management and duration of unstructured data. BASE consists of three principles. Which are Basically Available, Soft state and Eventual Consistency.

A. Characteristics of NoSQL

- It supports simple and flexible non-relational data models.
- Huge volume of data stored by NoSQL and having more flexible structure.
- Use NoSQL without any inconsistency, in distributed environment so provide high availability.
- No discontinuation of any work, if any faults or failures exist in any machine.
- It allows data to store without fixed table schemas.
- NoSQL do not follow ACID transaction.
- Ability to scale data horizontally leading NoSQL databases to high performance over many commodity servers.

B. Features of NoSQL

The common features of NOSQL data stores are:

- Easy to use in conventional load-balanced clusters
- Data is persistent Scalability according to available memory.
- Allows schema migration
- Owns individual query system and do not use SQL
- Eventually consistent across the clusters

II. RELATED WORK

Many researches have been done in the NoSQL databases after knowing the importance of NoSQL databases.Sanobar Khan, Prof.Vanita Mane [1] proposed a method to integrate two types of databases, which are MySQL and MongoDB, by adding a middleware between application layer and database layer. The main problem of this DB model was that it does not support the encryption/decryption by default and is vulnerable to injection attack and has exposure to DOS attacks. Which could be overcome by using different cryptographic systems and analyse their time complexity [2]. Enqing Tang, Yushun Fan[3] compared five NoSQL databases. It was concluded that Cassandra and HBase, were 1.71 times and 1.86 times slower than Redis.The worst performance was presented by Couchbase, which was 2.49 times slower than Redis. HBase was 1.58 times faster than Cassandra. Mrs. Rupali M. Chopade, Mr. Nikhil S. Dhavase [4] did Performance Comparison of MongoDB and Couch Base for Image Dataset. It was concluded that for image dataset insertion, time required by MongoDBis more as compared to CouchBase and for image dataset retrieval case is exactly different. Akansha Goyal, ArunSwaminathan, et.al. [5] stated that there are two general approaches to migrate data from RDBMS to NoSQL. One is Direct Mapping and another is Intermediate Mapping.According to their experimentation, for 1 million records, the total validation time was 28.515 seconds only. Haleemunnisa Fatima, Prof. Kumud Wasnik[6] observed that SQL databases are relational and follow ACID properties. While NoSQL databases are schema less provide better performance and scalability and follow ACID properties. NewSQL databases retain SQL and ACID properties and include performance and scalability through the modern architecture. Neal Leavitt [7] observed thatNoSQL also have some cons like NoSQL databases don't work with SQL, they require manual query programming, which can be fast for simple tasks but time-consuming for others.

III. CLASSIFICATION OF NOSQL DATABASES

There are four general types of NoSQL databases where every database has its own properties. Four types of NoSQL databases are Key-value data stores, Column- oriented data stores, document data stores and graph data stores [8].

A. Key Value Data Stores

The key value databases name itself states that it is a combination of two things that is key and a value. It is one of the low profile (traditional) database systems. Key Value (KV) databases are mother of all the databases of NoSQL. Key is a unique identifier to a particular data entry. Key should not be repeated if one used that it is not duplicate in nature. Value is a kind of data that is pointed by a key [9].The most known key value stores are Raik, Dynamo, Voldemort, Redis, Tokyo Cabinet.

B. Document Data Store

The data is compressed as a document store quite similar to a key-value store and the data is in the form of collection of key value pairs, but the only difference is that the values stored (referred to as "documents") provide some structure and encoding of the managed data. This type of database store unstructured (text) or semi-structured (XML) documents which are usually hierarchal in nature. Here each document consists of a set of keys and values which are almost same as there in the Key Value Databases Most popular document databases are MongoDB, CouchDB,Terrastore, ThruDb.

C. Columnar oriented data stores

Columnar oriented data stores are column-oriented databases also known as column family databases because data tables are stored as sections of columns of data, rather than as rows of data. The column family contains the row key which consists of super columns. Super column is a column which contains other columns but not the super column. And a column is pair of name and value. There are two types of column-oriented databases which are as follow

Wide Column data stores are Based on Google's Big Table store. Data tables are stored as sections of columns of data, rather than as rows of data. These databases are used for processing of web, streaming of data and documents.

Column oriented databases are those databases in which all the values containing columns are put together. All values of a column serialize together in a Column oriented database and then the values of next column are serialized, and so on.

D. Graph Data Store

Graph databases are based on the graph theory. It is based on graph structure which is consists of nodes, edges and properties to store data. Graph databases use nodes to represent entities, edges to represent relationships and properties to represent attributes [8]. In general, we can say that graph usually consists of nodes, properties and edges. NoSQL Graph database consists of:

- Nodes represent entities
- Properties represent attributes
- Edges represent relationships

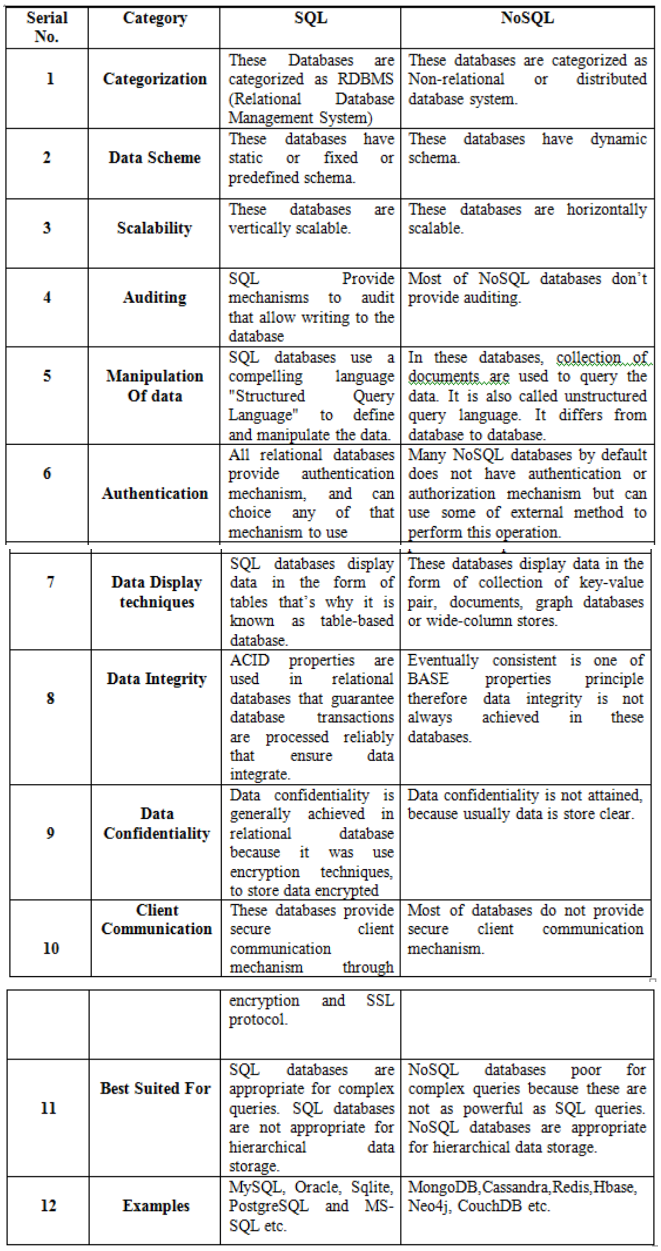
IV. RESULT AND ANALYSIS

This section includes the analysis of various databases and their results are compiled in the form of tables.

A. SQL vs. NOSQL

There are a lot of databases used today in the IT industry. Some of them are SQL databases, some are NoSQL databases. The conventional database is SQL database system that uses tabular relational model to represent data and their relationship between them. Some major differences between SQL and NoSQL are listed below in Table 1

Table 1: Comparison between SQL and NOSQL databases



Serial No.	Category	SQL	NoSQL
1	**Categorization**	These Databases are categorized as RDBMS (Relational Database Management System)	These databases are categorized as Non-relational or distributed database system.
2	**Data Scheme**	These databases have static or fixed or predefined schema.	These databases have dynamic schema.
3	**Scalability**	These databases are vertically scalable.	These databases are horizontally scalable.
4	**Auditing**	SQL Provide mechanisms to audit that allow writing to the database	Most of NoSQL databases don't provide auditing.
5	**Manipulation Of data**	SQL databases use a compelling language "Structured Query Language" to define and manipulate the data.	In these databases, collection of documents are used to query the data. It is also called unstructured query language. It differs from database to database.
6	**Authentication**	All relational databases provide authentication mechanism, and can choice any of that mechanism to use	Many NoSQL databases by default does not have authentication or authorization mechanism but can use some of external method to perform this operation.
7	**Data Display techniques**	SQL databases display data in the form of tables that's why it is known as table-based database.	These databases display data in the form of collection of key-value pair, documents, graph databases or wide-column stores.
8	**Data Integrity**	ACID properties are used in relational databases that guarantee database transactions are processed reliably that ensure data integrate.	Eventually consistent is one of BASE properties principle therefore data integrity is not always achieved in these databases.
9	**Data Confidentiality**	Data confidentiality is generally achieved in relational database because it was use encryption techniques, to store data encrypted	Data confidentiality is not attained, because usually data is store clear.
10	**Client Communication**	These databases provide secure client communication mechanism through encryption and SSL protocol.	Most of databases do not provide secure client communication mechanism.
11	**Best Suited For**	SQL databases are appropriate for complex queries. SQL databases are not appropriate for hierarchical data storage.	NoSQL databases poor for complex queries because these are not as powerful as SQL queries. NoSQL databases are appropriate for hierarchical data storage.
12	**Examples**	MySQL, Oracle, Sqlite, PostgreSQL and MS-SQL etc.	MongoDB,Cassandra,Redis,Hbase, Neo4j, CouchDB etc.

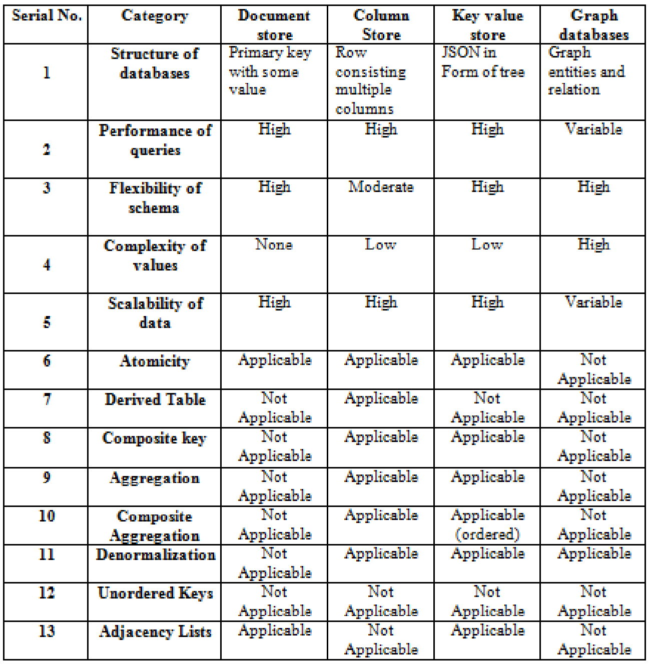
Table 1 compare the two most popular databases which are relational databases and NoSQL databases on the basis of some features and categories such as Categorization, Data Scheme, Scalability, Auditing, Manipulation Of data, Authentication, Data Display techniques, Data Integrity, Data Confidentiality, Client Communication etc. Table depicts that NoSQL databases are far better than Relational databases because NoSQL have some important features like these can scale data horizontally, NoSQL databases are schema less, these do not follow acid properties and many more. But in case of complex queries Relational databases beat NoSQL databases.

B. COMPARITIVE ANALYSIS OF NOSQL DATABASES

In this section four NoSQL databases are compared on the basis of their functional and non-functional requirements. And then Four different NoSQL databases and MySQL database are compared with each other.

- **Comparison of NoSQL databases on the basis of functional and non-functional features**

Table 2: Comparison of NoSQL databases on the basis of Functional and Non- functional features



Serial No.	Category	Document store	Column Store	Key value store	Graph databases
1	**Structure of databases**	Primary key with some value	Row consisting multiple columns	JSON in Form of tree	Graph entities and relation
2	**Performance of queries**	High	High	High	Variable
3	**Flexibility of schema**	High	Moderate	High	High
4	**Complexity of values**	None	Low	Low	High
5	**Scalability of data**	High	High	High	Variable
6	**Atomicity**	Applicable	Applicable	Applicable	Not Applicable
7	**Derived Table**	Not Applicable	Applicable	Not Applicable	Not Applicable
8	**Composite key**	Not Applicable	Applicable	Applicable	Not Applicable
9	**Aggregation**	Not Applicable	Applicable	Applicable	Not Applicable
10	**Composite Aggregation**	Not Applicable	Applicable	Applicable (ordered)	Not Applicable
11	**Denormalization**	Not Applicable	Applicable	Applicable	Applicable
12	**Unordered Keys**	Not Applicable	Not Applicable	Not Applicable	Not Applicable
13	**Adjacency Lists**	Applicable	Not Applicable	Applicable	Not Applicable

In Table 2 we compare Document store, Column Store, Key value store and graph database on the basis of their non-functional features such as, Structure of databases, Performance of queries, Flexibility of Schema, Complexity of Values, Scalability of data and functional features such as Atomicity, Derived Table, Composite key, Aggregation, Composite Aggregation, Denormalization, unordered keys and adjacency lists . Table 2 depicts that if data have to be represented in JSON format then Document store should preferred because this provide high performance and flexibility. And if we want to use denormalization, unordered key, composite key, composite aggregation operations on the database Document Store should not be used. Column store should be avoided if we want to use unordered keys, aggregation and group by, adjacency lists operations on the database. Key value store should not be preferred if we want to use composite key on the database. If we want to perform just denormalization then Graph store can be considered

- **Comparison between MongoDB, Cassandra, Redis, Neo4J and MySQL**

Table 3: Comparison between MongoDB, Cassandra, Redis, Neo4J and MySQL

Sr.No	Features	Document (MongoDB)	Wide Column Store(Cassandra)	KeyValue Store(Redis)	Graph Database (Neo4j)	SQL (MySQL)
1	**Website**	www.mongodb.org	cassandra.apache.org	redis.io	www.neo4j.com	www.mysql.com
2	**Developer**	MongoDB, Inc.	Apache Software Foundation	Salvatore Sanfilippo	Neo Technology	Oracle
3	**Initial Release**	2009	2008	2009	2007	1995
4	**Implementation Language**	C++	Java	C	Java, Scala	C and C++
5	**License**	Open source	Open source	Open Source	Open Source	Open Source
6	**Database model**	Document-based (Binary JSON) with structured object	Column-based table, Indexes over row-key and columns keys with multiple names or timestamps	Key-Value Store	Graph Database	Relational DBMS
7	**DB engine ranking**	Score 341.41 Rank #5	Score 119.09 Rank #10	Score 130.11 Rank #9	Score 40.90 Rank #22	Score 1226.40 Rank #2
8	**Server operating systems**	Linux OS X Solaris Windows	BSD Linux OS X Windows	Linux and OS X	Linux OS X Solaris Windows	
9	**Technical Documentation**	https://docs.mongodb.com/	http://cassandra.apache.org/doc/latest/	https://redis.io/documentation	https://neo4j.com/blog/technical-documentation-graph/	https://dev.mysql.com/doc/
10	**Description**	It is one of the most popular document store database.	It is one of the most known wide column store database. It is based on the concept of big table	It is an in-memory data structure store and an important key value store	It is a Open Source graph database	This one Widely used open source RDBMS
11	**DB**	Database	Key space	Database	Graph	Database
12	**Table**	Collection	Column Family	Hash set, List set, Sorted set and string	Label	Relation
13	**Supported Programming Languages**	Action Script, C, C#, C++, Clojure, Java, Java Script etc	C#, C++, Clojure, Erlang, Go, Haskell, Java, JavaScript, Perl, PHP, Python, Ruby, Scalaetc	ActionScript, C, C++, C#,Clojure, Common Lisp, D, Dart, Erlang, Go, Haskell,Java, JavaScript,Perl, PHP, Python, Ruby,Rust,Scala, Smalltalketc	.Net, Clojure, Elixir, Go, Groovy, Haskell, Java, JavaScript, Perl, PHP, Python, Ruby, Scalaetc	C# and VB are most commonly used
14	**Value**	Documents	Rows	Key value pair	Node and edges	Rows

15	**Read Operations**	Fast	Slow	Fast	Data dependent	Slow
16	**Write Operations**	Fast	Fast	Fast	Data Dependent	Slow
17	**Delete Operations**	Fast	Fast	Fast	Data dependent	Slow
18	**Data scheme**	No particular schema is followed.	Schema-Free	Schema-free	Schema-Free	Yes
19	**Predefined types**	Yes; Boolean, date, object_id, String, Integer, double	Yes; ASCII, int, blob, counter, decimal, double, list, map, set, text, timestamp, varchar	Partial; data types supported for value are strings, Bit arrays, hyper logs, hashes, lists, sets, sorted sets, and geospatial indexes	Yes; Boolean, byte, short, int long, float, double, char, string	Yes; int, float, double, date, time, bit, char, enum, binary, blob, Boolean
20	**Server Side Scripts**	JavaScripts	No	Lua	Yes	Yes
21	**Triggers**	No	Yes	No	Yes	Yes
22	**Partitioning methods**	Sharding with no individual point of failure	Sharding	Sharding	Partitioning should be avoided in Neo4j	Horizontal partitioning, sharding with MySQL Cluster or MySQL Fabric
23	**Foreign Keys**	Usually, not used, however equivalent operation with DBRef can be done	No	No	Yes	Yes
24	**Transaction Concepts**	Atomic operations can be performed within single document	Atomicity and Isolation are supported for single operations	Optimistic locking, atomic execution of command blocks and scripts	ACID	ACID
25	**User Concepts**	Access rights for users and roles	Access rights for users can be defined per object	Simple password – based access control.	Users, roles and permissions	Users with fine grained authorization concepts; no user groups or roles

26	**Type of Consistency concepts**	Eventual Consistency and Immediate Consistency	Eventual Consistency and Immediate Consistency	Eventual Consistency and Immediate Consistency	Causal and Eventual Consistency. configurable in Causal Cluster setup and Immediate Consistency in stand-alone mode	
27	**Durability**	Yes	Yes	Yes	Yes	

In table 3 one database have taken from each category of NoSQL databases that is MongoDB, Cassandra, Redis and Neo4j and a database from RDBMS which is MySQL. These are compared on the basis of some features like Read Operations, Write Operations, Delete Operations, Data scheme, Predefined types, Server-Side Scripts, Triggers, Partitioning methods, Foreign Keys, Transaction Concepts etc. Table 3 depicts that MongoDB stores values in form of document. Read operations are fast in MongoDB then Cassandra. MongoDB is more reliable compare to other databases. All NoSQL databases which are listed in table 3 are Open source and can be accessed easily. Read, write, and delete operations of in graph database are depend upon the type of data. On the other hand, Read, Write and delete operation are slower in MySQL then NoSQL databases. NoSQL databases are schema less while MySQL (SQL database) have a predefined schema. MongoDB has a no particular schema but usually contents of same documents as a convention have similar structures though it is not mandatory.

IV. CONCLUSION

The paper contains the comparison of many databases including relational and NoSQL databases on the basis of their features and common parameters. The aim of this paper is to provide an overview of NoSQL databases and how NoSQL databases dominate relational databases. NoSQL databases have schema less nature. This nature of NoSQL databases allows including fields without making any changes in the structure. To deal with the massive growth in structured, semi structured and unstructured data, NoSQL databases are very useful. These databases have ability to scale data horizontally which is missing in relational databases. This paper highlights some features, characteristics, advantages as well as disadvantages of NoSQL databases.

REFRENCES

[1] Sanobar Khan, Prof.Vanita Mane, “SQL Support over MongoDB using Metadata” International Journal of Scientific and Research Publications, Volume 3, Issue 10, 2013

[2] Jitender Kumar,VarshaGarg, ” SECURITY ANALYSIS OF UNSTRUCTURED DATA IN NOSQL MONGODB DATABASE”, Jaypee Institute of Information Technology. IEEE 2017

[3] Enqing Tang, Yushun Fan, “Performance Comparison between Five NoSQL Databases” Tsinghua National Laboratory for Information Science and Technology, Department of Automation Tsinghua University Beijing 100084, China, In IEEE, 7th International Conference on Cloud Computing and Big Data, 2016

[4] Mrs. Rupali M. Chopade,Mr. Nikhil S. Dhavase, ”MongoDB, CouchBase: Performance Comparison for Image Dataset”, Department of Information Technology MMCOE, Karvenagar Pune, India, in IEEE 2nd International Conference for Convergence in Technology 2017

[5] Akansha Goyal, Arun Swaminathan and Rasika Pande, “Cross Platform (RDBMS to NoSQL) Database Validation Tool using Bloom Filter”, Department of Computer Engineering and IT College of Engineering, Pune, Issue 8, 2016

[6] Haleemunnisa Fatima, Prof.Kumud Wasnik, “Comparison of SQL, NoSQL and NewSQL Databases for Internet of Things”, Computer Science and Technology UMIT, SNDT University Mumbai, India,IEEE, 2016

[7] Neal Leavitt " Will NoSQL Databases Live Up to Their Promise?" IEEE Computer Society IEEE, 2010

[8] Nishtha Jatana,Sahil Puri “A Survey and Comparison of Relational and Non-Relational Database” Department of Computer Science and Engineering, Maharaja Surajmal Institute of Technology, New Delhi, Volume 1, Issue Aug 2012

[9] Benymol Jose, Sajimon Abraham, “Exploring the Merits of NoSQL”, Research Scholar, School of computer Sciences,Mahatma Gandhi University, Kottayam, IEEE, July 2017

[10] Harpreet Kaur, JaspreetKaur, KamaljeetKaur “A Review Of Non Relational Databases, Their Types, Advantages AndDisadvantages”Sri Guru Granth Sahib World University,India, Volume 2, Feb 2013

[11] https://www.mongodb.com/scale/advantages-of-nosql

[12] Hossain Shahriar, Hisham M. Haddad, “Security Vulnerabilities of NoSQL and SQL Databases for MOOC Applications”, Department of Information Technology, Department of Computer Science Kennesaw State University, USA, Volume 8, Issue 01, March 2017