BVG Trust

*J. Sci. Trans. Environ. Technov.*, 2016, 10(1) :36-41

Scientific Transactions in Environment and Technovation

https://doi.org/10.56343/STET.116.010.001.009
http://stetjournals.com

# Big Data Management Using NOSQL

**V.AlameluMangaiyarkarasi* and M.V.Srinath[2]**

Department of Master of Computer Applications, Sengamala Thayaar Educational Trust Women's college, Sundarakkottai, Mannargudi-614016, Tamilnadu, India.

## Abstract

**Objectives:** Today's world of knowledge processing data storage is very important for every organization. Data retrieval and data migration are also more important and it has to be more flexible to all users; here we use NOSQL to manage the large volume of data. NOSQL means not only SQL but also NOSQL . It is not a RDBMS concept so it is called Non-Relational. This article deals with big data management in social networks using NOSQL databases. Performance, merits and demerits of SQL and NOSQL, categories of NOSQL databases, comparative analysis of NOSQL databases are discussed. NOSQL based databases are available in more numbers. This article also deals with DynamoDB, Cassandra, Hbase ,SAP HANA, MongoDB Cough DB, Polyglot, and Neo4,ansd also about the insertion, and join times are better than relational databases..

**Keywords:** NOSQL, SQL, DynamoDB, Cassandra, Hbase, SAP HANA, MongoDB Cough DB

## INTRODUCTION

Generally large volume of data are created, transferred, stored by social networking websites, and this aspect is growing day by day. These data are called big data; which contain any type data like photos, videos, messages, etc. Instead of RDBMS it is widely used in many applications to store and retrieve data.It is working well for limited amount of data.But it is inefficient to handle large volume of data. Now a -days most of the leading social networking companies such as Google, Facebook, Twitter, and Amazon are using NOSQL to handle big data. The main objective of NOSQL is to handle any unstructured data like documents, email, and multimedia. NOSQL is a non-relational database management system. In this environment, files and documents are distributed to many servers. It shares data sets with a flexible schema. Data processing is also performed in parallel manner. NOSQL system provides horizontal scaling across a large of servers like tens, hundreds and thousands of servers.This databases usually interact with the UNIX operating system (Vatika Sharma, MeenuDave,2012).

## TYPES OF NOSQL DATABASE

NOSQL databases are designed to handle big data so as to implement the methods to improve the performance of storing and retrieving data (Mohamed et al., 2014). There are four types of NOSQL database

1. Key values stores
2. Graph stores
3. Column stores
4. Document stores

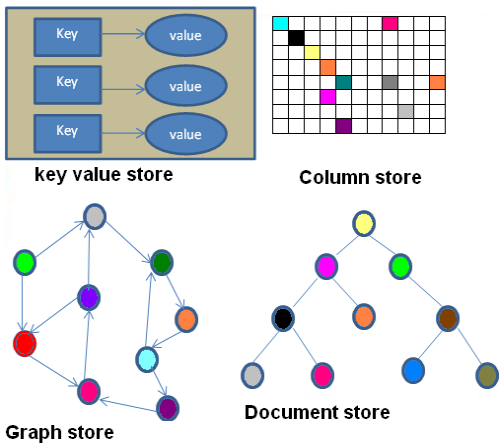


**Fig 1.**NOSQL data base structures

Figure1. Represents the types of NOSQL data base structures,such as key value store,column store,graph store,and document store. Each structures are formatted differently.

### Key value stores

It acts like a hash table, a key-value pair store in a file system (Veronika Abramova et al., 2014). Often there is no occurrence of update. In that file system path acts as the key and content acts as the file. The process of finding value associated with key is called lookup and the relationship between the key and its values is called

*Corresponding Author :
email: *vsalamu@gmail.com*



P - ISSN 0973 - 9157
E - ISSN 2393 - 9249

July to September 2016

mapping. Here the data are represented about student mark details, normally this data can represent in SQL as

**Table1.a)** Student Mark Table

| Reg.no | Student name | Mark1 | Mark2 | Total | Result |
|---|---|---|---|---|---|
| 132001 | Ananya.S | 80 | 80 | 160 | Pass |
| 132003 | Yamini.J | 80 | 30 | 110 | fail |

**Table 1.b)** Student Personal Information

| Reg. no | Father name | Mother name | Address | Laststudied college | Percentage |
|---|---|---|---|---|---|
| 132001 | Nathan | Alamelu | Madukkur | STET | 80% |
| 132003 | Kannan | Radha | Salem | STET | 78% |

Table.1. a and b represent the format which takes more space to store data, it is fully structured and it is not easy to add new field in the middle, but can be added only at the end. It affects total system and it takes more time to update all records. We cannot view all the data at the single window, so we have to create more than one table for the one concept. We can add keys to link one table to another table like primary key and foreign key. But in NOSQL we use key to represent particular data and it is not structured, and attributes can vary from one record to another.It takes less space to store data. It is easy to retrieve data from the database.

**Table.2.** Key-value pair stores in NOSQL database

| Student | |
|---|---|
| **Key** | **Attributes** |
| 1 | Reg No.: 132001<br>Student Name : Ananya. S<br>Mark1:80<br>Mark2:80<br>Total:160<br>Result: Pass |
| 2 | Reg No.: 132003<br>Student Name : Yamini. J<br>Mark1:80<br>Mark2:30<br>Total:110<br>Result: Fail |

Table 2 represents the key value pair NOSQL Data base here key are 1 and 2 and the corresponding values are related to student's details such as Regno, student name, mark1, mark2, total and result. These data are based on the keys.

**Graph stores**

Graph stores excel to interconnect between the nodes, both nodes and edges, and also store key-value pair.Graph databases are useful when one is more interested in relationships between data than in the data itself. For example, in representing and traversing social networks, generating recommendations Graph databases are Neo4J, infogrid, sones graphDB, allegro graph and infinite graph.

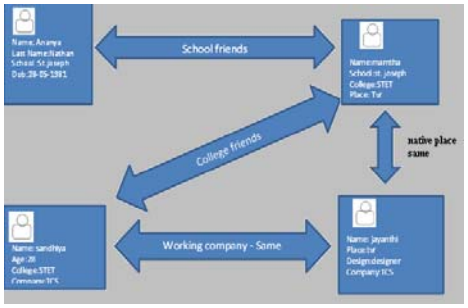


**Figure 2.** Graph NOSQL way to link

Figure.2.represents relationship in social media Face book, we can link friends through the profile information, and through this media we can meet our old school friends, college friends and others.

**Column stores**

Data are stored in column oriented databases in the form of whole column rather than a row. It contains serialized all the values of a particular column together on disk, which makes fast retrieval of integrated data on a particular column. This decreases the disk access compared to relational table, which consists of column and rows with uniform size fields for every record (Supriya et al.,2015).

**Table 3.** Column store NOSQL database

| Column families: customer | Column families: orders |
|---|---|
| RowID: 1001<br>Column : Name<br>First Name: Ananya<br>Last Name : Nathan<br>Column ; Address<br>Street : West<br>City : Thanjavur<br>Pin: 614903<br>Column : Order<br>Order ; ord001<br>Total cost : Rs. 5000.00 | RowID : 2010<br>Column: Order<br>OrderID : ord001<br>Date : 28-01-2016<br>Column : items<br>Item Code1 : IT010<br>Iterm code2 : IT012<br>Column : amounts<br>Discount : Rs. 200.00<br>Amount : Rs. 2800.00 |
| RowID : 1002<br>Column : Name<br>First Name : Suresh<br>Last Name : Kannan<br>Column : Address<br>Street : Main Road<br>City : Trichy<br>Pin : 610003<br>Column : order<br>order : ord002<br>Total Cost : Rs. 10000.00 | RowID: 2011<br>Column : order<br>OrderID : ord001<br>Date : 28-01-2016<br>Column : items<br>Item code1: IT018<br>Column : amounts |

Table.3. represents customer information and order information, each column contains Row ID and any one of the attribute connects two or more columns, here the person RowID: 1 purchase more number of items through the order number ord001.

**Document stores**

This database is suitable to store and manage big data like text documents, email and XML documents. This type of database stores structured or semi structured documents which are usually hierarchical in nature. This is good for storing semi structured data also. Examples: MongoDB and CoughDB. It can also maintain collection of complex documents with arbitrary nested data formats and varying record format.

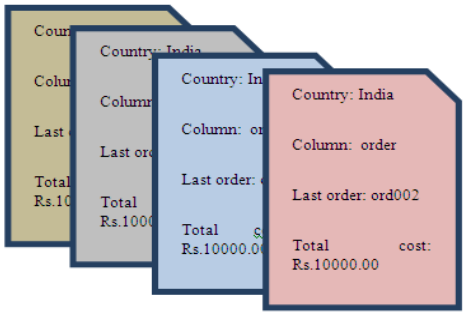


**Figure 3.** Document database in NOSQL

Fig3 represents the document store NOSQL database structure, here each data is considered as a document, we can retrieve these data through MongoDB and CoughDB database using queries.

**BENEFITS OF NOSQL**

NOSQL databases are more scalable and provide superior performance. NOSQL provide more benefits to users, which include

- Dynamic schemas
- Auto-Sharding
- Replication
- Integrated caching

**Dynamic schemas**

In SQL database it is essential to know as to what is to be stored, in advance. This concept not fit into agile methodology, the schema of your database need to change often. When it is needed to add column into database, then the entire databases are to be migrated into the new schema, which is very difficult for large databases. But in NOSQL database are built to allow the insertion of data without predefined schema. It makes easy to change the applications in real time. NOSQL databases allow validation rules to be applied within the databases, allowing users to enforce governance across data while maintaining the agility benefits of dynamic schema. In Dynamic Sharding, an external locator service determines the location of entries. It can be implemented in multiple ways. When the cardinality of partition keys is relatively low, then the locator can be assigned per individual key. Otherwise a single locator can address a range of partition keys[10].

**Auto-Sharding**

Sharding is a type of database partitioning that separates very large databases into smaller, faster, more easily managed parts called data shards. Auto Sharding means NOSQL databases natively and automatically spread across an arbitrary number of servers, without requiring the applications to be aware of the composition of the server pool. Data and query load are automatically balanced across servers and when a server goes down, it can be quickly and transparently replaced with no application disruption.

**Replication**

Data replication is very important part of keeping the database up and serving queries. Like many SQL database authors, we decided to keep R=3 copies of each piece of data in the database, not use RAID (Redundant Array of Independent Disk) to improve reliability. The key goal we were shooting for was a database which degrades gracefully when there are many small failures over time without needing human intervention.

**Integrated caching**

Many NOSQL database technologies have excellent integrated caching capabilities, keeping frequently-used data in system memory as much as possible and removing the need for a separate caching layer. Some NOSQL databases also offer fully managed, integrated in memory database management layer for workloads demanding the highest throughput and lowest latency.

**NOSQL DATABASES**

**Dynamo DB**

Developedby: amazon.com

Released year: 2012

Data base type: key value store

Operating system: cross platform

DynamoDB is a cloud based NOSQL database offered by Amazon. In this database we can choose the level of throughput desired. It is c completely managed by the amazon.com. The cost involved in the operation is low.

**Cassandra**

Developed by: Apache software foundation

Released year: 2008

Data base type: key value store

Operating system: cross platform

Apache Cassandra is a massively scalable open source non-relational database that offers continuous availability, linear scale performance, operational simplicity and easy data distribution across multiple data centers and cloud availability zones. Cassandra was originally developed at Facebook, was open sourced in 2008 and become a top level apache project in 2010.

**HBase**

Developed by: Apache software foundation

Released year: 2016

Data base type: Column based

Operating system: cross platform

HBase is an open source, non-relational, distributed databases model developed after Google'sBig Table and written in Java. HBase features Compression, in-memory operation and bloom filters on a per-column basis as outlined in the original big table paper. HBase is now serving for several data driven websites including Facebook's messaging platform.

**SAP HANA**

Developed by: SAP SE

Released year: 2016(revised)

Data base type: Column based

Operating system: cross platform

SAP High-Performance Analytic Appliance is an in-memory column oriented RDBMS System and SAP HANA can be deployed on-premises as an appliances from a certified hardware vendor or on certified hardware with Tailored Data center Integration (TDI) .

HANA is also available in the cloud as a database as a service on Amazon web services, Microsoft azure, or the SAP HANA cloud platform.

**MongoDB**

Developed by: MongoDB Inc.

Released year: 2009

Data base type: Document based

Operating system: cross platform

MongoDB is a document-oriented database, which has been adopted for usage by multiple large vendors such as EBay, Foursquare, LinkedIn and others Queries and

**Table 4.** Difference between SQL and NOSQL

| Category | SQL (Structured Query Language) | NOSQL (Not Only SQL) |
|---|---|---|
| Types | Based on one type of database | Based on different type of data bases, key-value, documnet storebas, column store, graph store |
| Development | Developed in 1970 | Developed in 2000 |
| Examples | MYSQL, Microsoft SQL Server | MongoDB, Cassandra, Hbase, Neo4j |
| Storage model | Individual records are stored in rows in table, column contain specific set of data e.g. name, reg.no etc., | Varies based on database type In key value, only two columns key and value, In column based BLOB model, In document model based on XML, JSON model. |
| Schemes | Structures and data type fixed in advance | Fields are adding dynamically. |
| Scaling | Vertical scaling, meaning a single server must be made increasingly powerful in order to deal with increased demand. It is possible to spread SQL databases over many servers, but significant additional engineering is generally required, and core relational features such as JOINs, referential integrity | Horizontal scaling, The database automatically spreads data across servers As necessary. |
| Data manipulation | Use specific language like select, update, insert statements e.g. Select * from employee where salary>=10000; | Use object oriented API |
| Consistency | Strong consistency | Depend on product, some product in strong consistency e.g. MongoDB, some provide eventual consistency e.g. Cassandra |

Data are represented in JSON format, which is better than SQL in terms of security because it is more "well defined", very simple to encode/decode and also has good native implementations in every programming language.

MongoDB duplication is organized using a primary-secondary server configuration whereby one server is primary and all others are secondary. The primary server or primary replica switches all write operations and records them in a distinct collection where the secondary's read and apply them. Secondary replica severs can also read the operations from another secondary and thus limiting the amount of load on the primary server.

**CoughDB**

Developed by: Apache software foundation

Released year: 2008

Data base type: Document based

Operating system: cross platform

CouchDB is an apache project developed in 2008 Erlang. This data model is richer. A document has field values that can be scalar (text, numeric, or Boolean) or compound (a document or list). Queries are done with what CouchDB calls "views", which are defined with JavaScript to specify field constraints. The indexes are B-trees, so the results of queries can be ordered or value ranges. Queries can be distributed in parallel over multiple nodes using a map reduce mechanism. However, CouchDB's view mechanism puts more burdens on programmers than a declarative query language.

**Neo4j**

Developed by: Neo technology

Released year: 2007

Data base type: graph database

Operating system: cross platform

Neo4j is one of the popular graph databases and CQL stand for cipher query language. It is written by using Java language. CQL is a query language for graph databases. It is a declarative pattern matching language.

**Polyglot**

Polyglot persistence will occur over the enterprise as different applications use different data storage technologies. It will occur within a single application as different parts of an applications data store have different access characteristics (Martin Fowler and Pramod Sadalage, 2012).

Table 4 represents the difference between SQL and NOSQL the differences are about types, development, storage model, schemas, scaling, data manipulation and consistency. After Comparison NOSQL performances are better than SQL.

## COMPARATIVE ANALYSIS OF SQL AND NOSQL DATABASE MONGODB

There are three different type of data set such as small, medium, and large (Rajat Aghi *et al.*,2015). The specifications of the data sets are,

1. Small data sets contains 10 rows and 2 columns

2. Medium data sets contains 400 rows and 35 columns

3. Large data set contains 2000 rows and 20 columns.

**Table 5.** The insertion and join time of various data into two databases SQL and MONGODB

| Data Size | Insertion time in seconds | | Join time in seconds | |
|---|---|---|---|---|
| | SQL | MongoDB | SQL | MongoDB |
| Small | 0.000668 | 0.000311 | 0.23943 | 0 |
| Medium | 0.000613 | 0.000235 | 10.3739 | 0 |
| Large | 0.000613 | 0.000243 | 27.24618 | 0 |

Table 5 represetns the timing details of insertion of data in SQL and MongDB in NOSQL. The data ranges are given in three varities like small ,medium and large and query join time also represented here for both SQL and mongoDB in NOSQL.

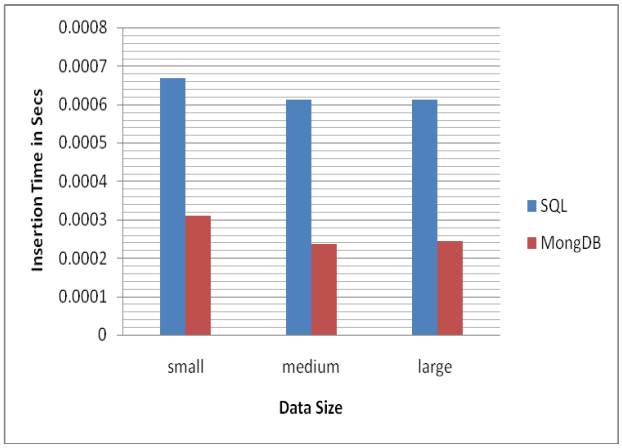


**Figure 4.** Insertion time of SQL and MONGODB databases

Figure 4 represents the graphical representation of insertion time of SQL and MongoDB in NOSQL. According to the graph the MongoDB data base insertion time is smaller than SQL in all three levels such as small, medium and large.

P - ISSN 0973 - 9157
E - ISSN 2393 - 9249

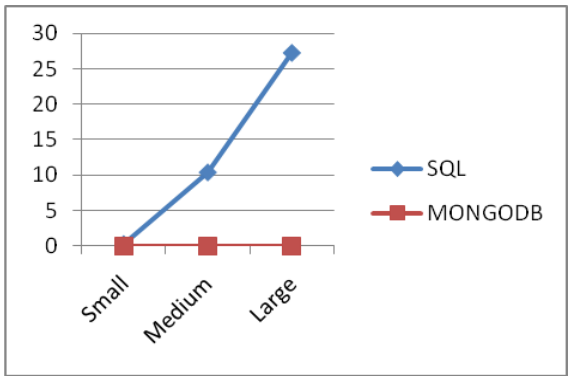


**Figure 5.** Join time of SQL and MONGODB databases

Fig 5 represents the join time of SQL and MongoDB in NOSQL, here compare NOSQL with SQL, NOSQL have no join time to retrieve and store the data in data database. According to the graph the SQL take more time than MongoDB in NOSQL.

## CONCLUSION

For large computational and storage requirements of applications such as big data analytics, business intelligence and social networking over Peta byte data sets have pushed SQL like centralized database to their limits. NOSQL data bases perform well in scalability for simple operations over large volume of datasets. NOSQL systems are distributed, non-relational databases designed for large-scale data storage and for massively-parallel data processing across a large number of commodity servers. NOSQL has the advantage of horizontal scaling, but for difficult SQL requests, it cannot support them very well. NOSQL is a good choice.

## REFERENCES

Martin Fowler, Pramod Sadalage 2012. Polyglot Persistence. http://martinfowler.com/articles/nosql-intro-original.pdf February 8,

Mohamed, A.Mohamed, obey G., Altrafi Mohamed, Ismail, O. 2014.Relational vs. NoSQL Databases; *Int. J. Computer and Information Tech.*3 (3) :598-601.

Rajat Aghi, Sumeet Mehta, Rahul Chauhan, Siddhant Chaudhary and Navdeep Bohra. comprehensive comparison of SQL and MongoDB databases. *Int. Journal of Scientific and Research Publications*, 5(2), 2015 Feb; 2250-3153 ;P.1-3.

Supriya S. Pore and Swalaya B. Pawar. 2015. Comparative Study of SQL & NoSQL Databases. *Int. J. Adv. Res. Comp. Eng. & Tech.* (IJARCET). 2015 May; 4(5); 1747-1753.

Vatika Sharma, MeenuDave . 2012. SQL and NOSQL Databases. *Int. J. Adv. Computer Science and Software Eng.*, 2(8):20-27.

VeronikaAbramova, Jorge Bernardino, Pedro Furtado. 2014 Which NoSQL Database? A Performance Overview. *Openn Access Open J. Databases* (OJDB) 1( 2):17-24.

MongoDB: http://www.mongodb.org/

DynamoDB, http://aws.amazon.com/dynamodb/

CouchDB: http://couchdb.apache.org

Cassandra: http://cassandra.apache.org/

Hbase: http://hbase.apache.org/

https://medium.com/@jeeyoungk/how-sharding-works-b4dec46b3f6.

http://searchcloudcomputing.techtarget.com/definition/sharding

https://www.mongodb.com/nosql-explained

http://www.tutorialspoint.com/neo4j/http://www.tutorialspoint.com/neo4j/neo4j_cql_introduction.htm

http://highscalability.com/blog/2012/7/9/data-replication-in-nosql-databases.html