Available online at www.sciencedirect.com

ScienceDirect

Materials Today: Proceedings 24 (2020) 2036–2043

materialstoday: PROCEEDINGS

www.materialstoday.com/proceedings

I2CN-2K19

Performance analysis of NoSQL and relational databases with MongoDB and MySQL

Benymol Jose[a], Sajimon Abraham[b]

[a]Research Scholar, School of Computer Sciences,Mahatma Gandhi University, Kottayam

[b]School of Management & Business Studies, Mahatma Gandhi University, Kottayam

Abstract

Relational databases are not powerful, when we have to inquire with an extensive variety of gigantic information. Relational databases like MySQL are storing data in organized form. Document-based data stores like MongoDB, which is a type of NoSQL database can store huge volume of data which additionally have very powerful query engines and indexing features. This paper concentrates on the upsides of NoSQL databases over relational databases in the investigation of the big data by making a performance comparison of various queries and commands in both the systems using two different datasets of dissimilar sizes.

Keywords:Big Data; Relational Databases; NoSQL Databases; MySQL; MongoDB

1. Introduction

Big data alludes to information with enormous volume which is having exponential advancement in development. This data lands in different structures and with expanded speed. This sudden development in volume of information has presented new data storage, organization, execution and analysis methods. This caused the need of new models and query languages to deal with the enormous information produced in every second [1]. The huge volume of data made adds significantly to a substantially bigger assortment of data types. Storage and processing of this data with the traditional relational databases is difficult. NoSQL systems and Big Data Analytics hold huge guarantee for supporting the storage and processing issues with this data. This examination of large measures of varying data, enables organizations to comprehend their clients and helps better organization of business. Also, analysis of Big Data encourages improved conclusions, expanded perceivability and generally more noteworthy incentives in business [2].

The relational model has ruled the computer industry since the 1980s for the most part to store and retrieve data. After some time, relational database start losing its importance because of its dependence on a rigid schema which makes it hard to include new relationships between the entities [3]. Another critical reason of its failure is that as the accessible data is developing manifolds, it is getting complicated to work with relational model. This is because of the time it takes for joining a large number of tables [4].

Conventional relational databases utilize two-dimensional table to embody data, with firm reliability of database operations. It is a time-consuming procedure to read and write real time data, execute compound SQL queries, particularly when query is connected with multiple tables. In any case, for an extensive variety of huge data queries, multi-table query is not powerful. NoSQL is not like a conventional database management system. This framework utilizes a non-conventional structure of data storage. Normally, it doesn't bolster the join operation, while the query execution is productive. This paper endeavours to utilize this new database innovation to actualize high performance database operations, contrasting it with the traditional relational database systems utilizing MySQL. As the most famous freely downloadable database in the world, MySQL has lot of attractions as being small, fast, and low price. Here, NoSQL database framework is actualized with MongoDB, which is a document-based database system portrayed by huge data storage, at the same time with high and improved query performance.

MongoDB is a document-based NoSQL database created by MongoDB Inc, which is available as an open source. MongoDB systems use documents and collections rather than the tables utilized as a part of conventional databases. JSON (Java Script Object Notation), BSON (Binary JSON) based documents or sub documents are the major components of the collections in MongoDB. The capacity issues of traditional database systems related with managing immense volumes of unstructured data can be amended by utilizing NoSQL systems like MongoDB [5].

Here, a set of experiments are performed by executing different queries and commands. They are SELECT query with and without using aggregate functions, INSERT, and UPDATE commands. The SELECT query is executed for a simple case and also for a condition. All the experiments are performed using two datasets of different sizes. Performance graphs are drawn separately for the two datasets and the results show that MongoDB performs better in all the cases. The results can be a boost for companies to change the structure of their databases from conventional form to NoSQL.

The paper is composed as follows: section 2, depicts few studies related to this area and in section 3, a comparison of queries used in MySQL and MongoDB is covered. Section 4 gives an overview of the datasets which are being used for performing queries in both MongoDB and MySQL and the methodology used are discussed. In section 5, platform requirements and software's used are discussed. Also, the performance test is done by executing different queries and commands in both the databases using two different sized datasets. The time taken by them for different scale factors for both the datasets are noted in different tables and the corresponding graphs are also drawn. And finally, in section 6, the performance is evaluated and analyzed and the paper is concluded in section 7 by stating that the performance of MongoDB is increasingly when contrasted with that of MySQL for all types of operations.

2. Related Study

These days, there have been lot of discussions happening worldwide about the performance of SQL and NoSQL databases. But very few researches have occurred with the performance comparison of SQL and NoSQL databases with large datasets and simple experiments. Here, we are referring to few studies and works happened in this area. In a recent study, a method was projected to associate the properties of MySQL and MongoDB by adding a middleware between the layers of application and database. It consists of metadata which includes different kinds of packages [6].In another contribution, different database operations were performed in the SQL and NoSQL databases with the same dataset for an e-commerce system. And it is concluded that MongoDB does better for all operations except for few aggregate operations [7]. In another work, effort is made to utilize NoSQL database in place of the relational database. It is associated to a conventional information management organization, compared the two database technologies, provided the key code of NoSQL execution, and lastly recorded the performance comparison of the two schemes [8].

Another research is endeavoring to assess the execution speed of five different NoSQL databases (Redis, MongoDB, Couchbase, Cassandra, HBase) with an evaluating device called - YCSB (Yahoo! Cloud Serving Benchmark) [9]. Another exploration is centered around OracleDB as a relational DBMS and MongoDB as a nonrelational DBMS to discover the distinctions in their performances, keeping in mind the end goal to demonstrate which one performs better in information recovery by executing few queries [10].

3. Comparison of Queries

Diverse sorts of NoSQL databases are available and they are key value databases, column value databases, document-oriented databases and graph databases [11]. MongoDB is a case of document-oriented datastore. Information is put away in documents and collections instead of the rows and tables in conventional relational databases. It is hard to store substantial volumes of unstructured data with the conventional relational database management systems.

The main property of MongoDB is auto sharding, in which additional replica server nodes are brought to the system. It is a database which works fastly and gives indexes on the primary attributes as well as on the secondary attributes [12]. Different collections can be compared utilizing technologies such as aggregation framework, Map Reduce and Hadoop systems [13].Document databases, for example MongoDB utilize JSON documents instead of the records in a conventional database [14].

Relational Databases is the reason for Structured Query Language. All modern database systems like MS SQLServer, IBM DB2, Oracle, MySQL, and Microsoft Access work based on the principle of relational databases. The data in an RDBMS is put away in database objects which are named as tables. This table is essentially a gathering of related data entries and it consists of various columns and rows.The table below,
To do this demonstration, MongoDB, which is a document-based NoSQL database and MySQL, which is a conventional relational database are considered. And the operations which are listed here with these two databases are CREATE, SELECT, INSERT, DELETE and IMPORT.

Getting the measure of data required for execution needs of NoSQL database frameworks is difficult since solid outcomes require records of around millions and such big numbers are only controlled by enormous organizations, for example, Google, CERN, Facebook and so forth.

This is the reason of the requirement of explores in this area that implements tests to check the performance of these systems.

4. Methodologyand Data Sets Used

Here different queries and commands are executed for simple select, select with a condition, insert, and update. The comparison between the two databases is done by executing a series of queries and commands with two different datasets of different scales. The performance of these operations is evaluated and analysed by drawing graphs.
The datasets used were downloaded from the URLS https://data.cityofchicago.org/browse?category=Buildings and https://www.kaggle.com/safecast/safecast/data. The first one includes the particulars of licenses given by the Buildings Department in the Chicago City from 2006 to the current. The dataset for every year covers in excess of 65,000 records/rows of data and cannot be seen in complete in Microsoft Excel. Furthermore, the dataset taken for the experiment and analysis consists of around 1.5 lakhs of records. It is denoted as D1.

The second data set considered consists of radiation and environmental data from all over the world which begun in response to the nuclear disaster in Japan in March, 2011. It consists of around 10 lakhs 50 thousand records of 1GB size. To denote it in experiments the name D2 is used. In order to avoid complication, only queries with one dataset, D1 are shown in the following demonstrations.

5. Performance Tests and Results

All the tests are done using MongoDB Studio3T 3.4.5 and MySQL Workbench 6.3 for different data manipulation commands as select, update and insert. The experiments are performed in a machine running with Intel(R) Core (TM) i7-7500U CPU @ 2.70GHz, with Windows10, 64-bit operating system. The machine has 8GB of physical memory and 256GB of SSD hard disk space. The MySQL Workbench 6.3 and MongoDB Studio 3T 3.4.5 are the respective software's used to test the data.The datafiles are downloaded in CSV formatthrough the export menu option in Studio3Tand then migrated to MySQL Workbench with the import option available there.

These tests are performed using the two datasets D1 and D2. The example operations shown here are for dataset D1, but all the operations are executed separately for both the datasets. And the time taken for executing the operations with both the datasets are noted in the corresponding tables. The graphs are drawn separately for the two datasets.

5.1. Select Query

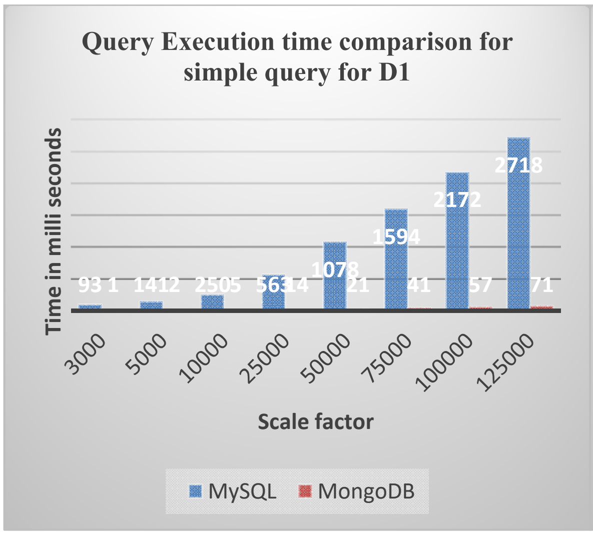
The select query is used for retrieving records from a table. The following queries are used to check the performance of the two databases in case of select.

5.1.1.Simple Select Query

Here the test is done by performing a simple query which includes only a SELECT statement in MySQL and the equivalent in MongoDB. A sample query executed for both the databases with dataset D1 for 25000 rows is shown below.

MySQL: select *from buildingpermit LIMIT 0,25000;
Studio3T:db.buidingpermit.find().limit(25000).explain("executionStats")

The commands are executed for different scale factors as 3000, 5000, 10000, 25000, 50000, 75000, 100000 and 125000 with the two datasets. The commands and its syntax used in both cases are different. To, draw graphs for comparing their performance, both are represented in milliseconds. The graphs are drawn with the different scale factors on X axis and time in milliseconds on Y axis and are shown in Fig. 1 and Fig. 2.



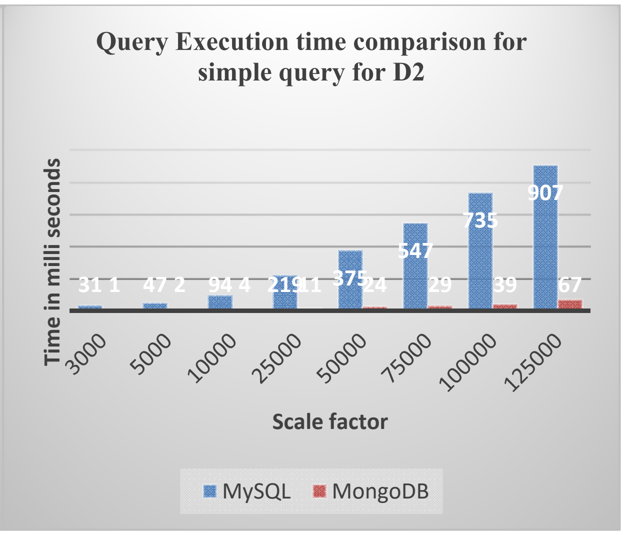


Fig. 1. Performance Comparison for the Simple Query for D1**Fig.2.** Performance Comparison for the Simple Query for D2

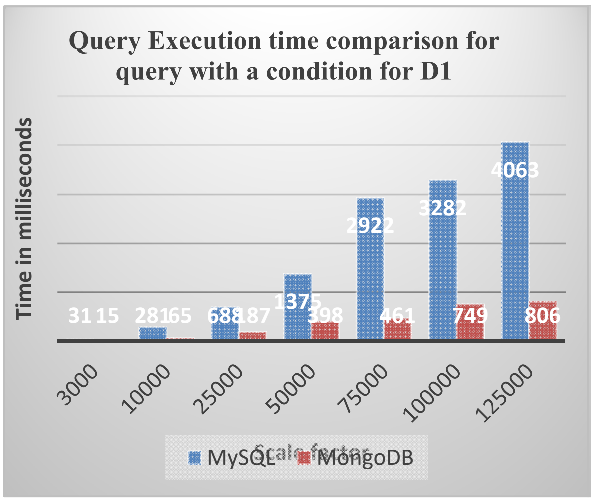
From the graphs, it is evident that MongoDB shows higher performance for the simple query operation with all the differentscale factors for both the datasets D1 andD2.

5.1.2.Select Query with a Condition

Here, the select command is executed with a given condition. In this case also the query is executed by varying the number of records. A sample query in case of a simple query with a condition executed for both the databases with dataset D1 for 25000 rows is shown below.

```
MySQL: select *from buildingpermit where street_number>1000 LIMIT 0,25000;
Studio3T:db.buidingpermit.find({"STREET_NUMBER":{$gt:NumberInt(1000)}}).limit(25000).explain("executionStats");
```

The execution time taken by the query with both the datasets D1 and D2 and the time taken by both MySQL Workbench and Studio 3T is noted for various scale factors 3000, 10000, 25000, 50000, 75000, 100000 and 125000. The corresponding graphs for D1 and D2 with scale factors on X axis and time in milliseconds on Y axis are shown in figures, Fig. 3 and Fig. 4.



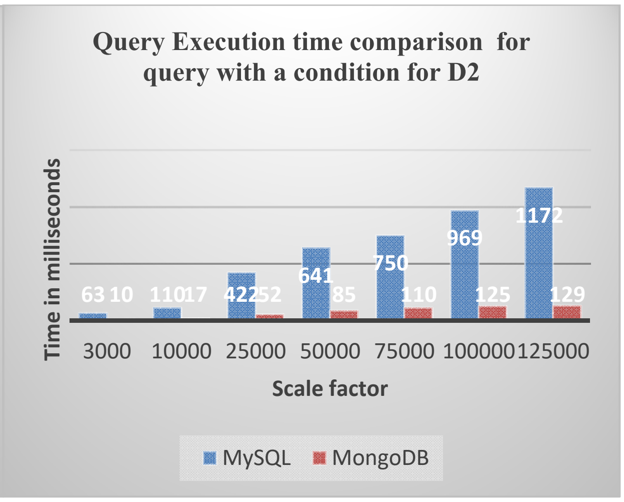


Fig. 3. Performance comparison for query with a condition for D1**Fig. 4.** Performance comparison for query with a condition for D2

From the graphs, it is evident that MongoDB shows higher performance for the simple query with a given condition operation for all the different scale factors with datasets D1 and D2.

5.2.UPDATE Command

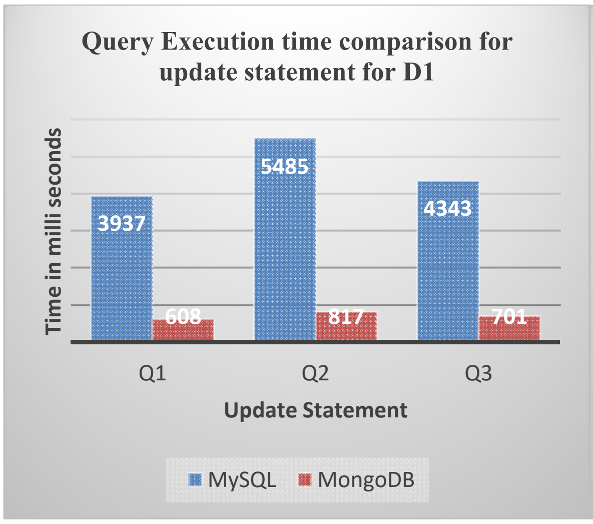
The update statements are executed using both MongoDB studio 3T and MySQL Workbench for the two datasets D1 and D2. Theupdate statements are executed for the following three cases and are labelled and shown below as Q1, Q2 and Q3.The corresponding queries are shown in the following tables Table I, Table II and Table III.

- **Q1:** To update the contractor_1_type to Architect where street_number<1000
- **Q2:** To update the contractor_1_type to Civil where street_number>1000 and street_number<5000
- **Q3**: To update the contractor_1_type to architect where street_number>5000.

Table 1.UPDATE statement for Q1

Query in MySQL	Query in Studio 3T
UPDATE buildingpermit SET CONTRACTOR_1_TYPE = 'ARCHITECT' WHERE STREET_NUMBER < 1000;	db.buildingpermit.explain("executionStats").update({"STREET_NUMBER":{$lt:1000}},{$set:{"CONTRACTOR_1_TYPE":"ARCHITECT"}},{multi :true})

The execution time taken by these three different update statements are noted for the two databases with the datasets D1 and D2. Based on this time, performance graphs are drawn separately for the two datasets D1 and D2 with the different update statements Q1, Q2 and Q3 on the X axis and the time taken for the execution on Y axis with and are shown in figures, Fig. 5 and Fig.6



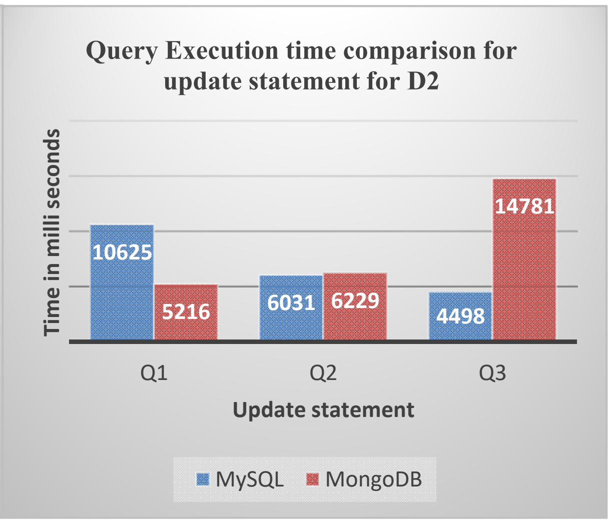


Fig. 5. Performance Comparison forthe update statements for D1**Fig.6.** Performance Comparison for the update statements for D2.

TABLE 2. UPDATE Statement for Q2

Query in MySQL	Query in Studio 3T
UPDATE buildingpermit SET CONTRACTOR_1_TYPE = 'CIVIL' WHERE STREET_NUMBER > 1000 AND STREET_NUMBER < 5000;	db.buildingpermit.explain("executionStats").update({ $and:[{"STREET_NUMBER":{$gt:1000}},{"STREET_NUMBER":{$lt:5000}}]},{$set:{"CONTRACTOR_1_TYPE":"CIVIL"}},{multi :true})

TABLE 3.UPDATE statement for Q3

Query in MySQL	Query in Studio 3T
UPDATE buildingpermit SET CONTRACTOR_1_TYPE = 'ARCHITECT' WHERE STREET_NUMBER > 1000 AND STREET_NUMBER < 5000;	db.buildingpermit.update({"STREET_NUMBER":{$gt :5000}},{$set:{"CONTRACTOR_1_TYPE":"ARCHITECT"}},{multi :true})

From the performance graphs it is clear that MongoDB shows better performance than MySQL for all the three update statements considered. In this case also the time taken by both MySQL and MongoDB increases as the size of the dataset increases. There is a proportionate increasein time taken in case of MySQL and MongoDB when dataset D2 is used. Anyway, MongoDB performs well in all cases by taking less time for execution

5.3. INSERT Command

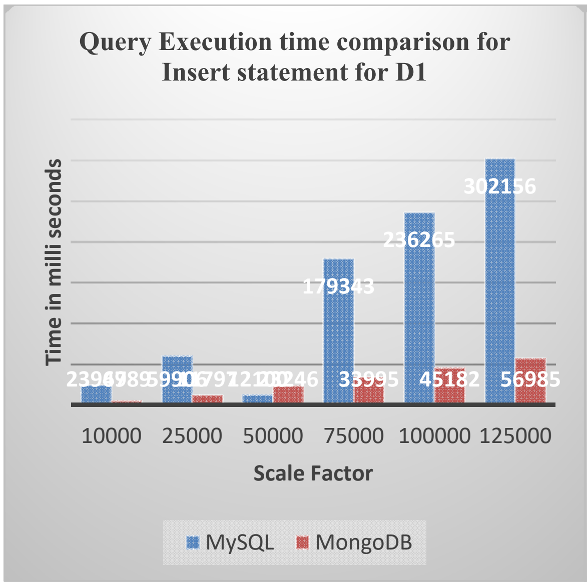
For executing the INSERT statement both the datasets D1 and D2 are used and the statements are executed with six scale factors of 10000, 25000, 50000, 75000, 100000 and 125000.

The corresponding evaluation graphs are drawn with different scale factors on X axis and execution time in milliseconds on Y axis and are shown in Fig. 7 and Fig. 8 for D1 and D2. The, outcomes demonstrate that MongoDB does INSERT tasks with higher speed in every one of the six scales, and with the expansion in quantity of records this predominance would be more prevailing.

6. Analysis and Evaluations

The performance of MongoDB Studio 3T is compared with MySQL Workbench by executing queries in four different cases. Two datasets of comparatively large number of records are taken and the operation is performed by increasing the scale factor of records being processed in successive steps. Graphs are plotted based on the query execution time and are shown in the corresponding figures.

On analyzing and comparing, it is seen that, the execution of MongoDB is progressively when compared in relation to that of MySQL. When the number of records looked at is small or big, there is much distinction in the execution time taken for the activities to finish for both MongoDB and MySQL databases.



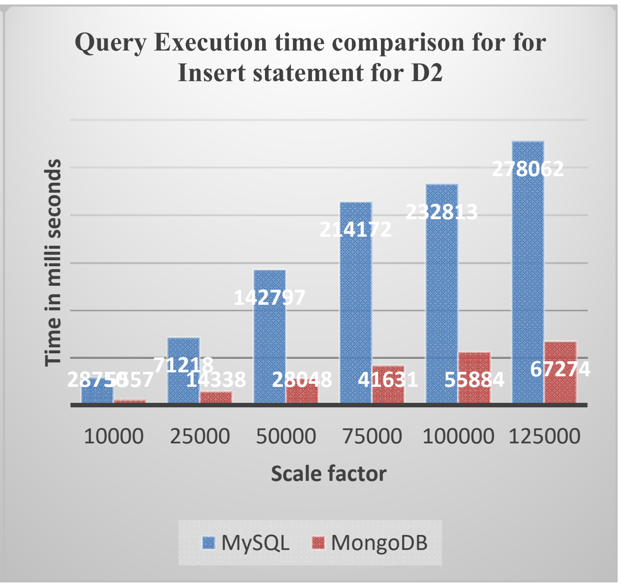


Fig. 7. Performance Comparison in case of insert statement with D1**Fig. 8.** Performance Comparison in case of insert statements with D2.

Be that as it may, MongoDB demonstrates great improvement by taking less time for the completion of queries compared to MySQL. NoSQL databases show good performance and scalability for most operations over huge datasets. In this paper, the experiments are done with two different datasets for different workloads to find the contrasts in execution time which is there with relational and NoSQL databases. It was performed by executing queries and different commands with MySQL workbench 6.3 and MongoDB studio3T 3.4.5. The differences in execution time is shown using the graphs, and from the graphs the distinction in the processing time taken by the two databases including in case of aggregation operation is clear. In all the cases considered here, MongoDB performs well by taking less time and thus can be favored for its performance.

7. Conclusion

This paper clarifies the contrasts in execution time which is there with relational and nonrelational databases. It was performed by executing queries using different types of commands with MySQL workbench and MongoDB studio3T. The differences in execution time is noted and shown using the corresponding graphs. In all the cases NoSQL databases performs well than relational databases. With all the analysis done here, it is seen that, the performance of MongoDB is increases when contrasted with that of MySQL. It is a motive for the commercial business organizations to shift from conventional database systems to NoSQL databases in managing todays big data. This research can be further improved by using several different types of queries with a higher number of records for different types of NoSQL databases.

References

[1] Yesha Mehta, Sanjay Buch, “Big Data Mining and Semantic Technologies: Challenges and Opportunities”, Int. J. on Recent and Innovation Trends in Computing and Communications,2015, 3(7), 4907-4913.
[2] Lidong Wang, Cheryl Ann Alexander, “Big Data Driven Supply Chain Management and Business Administration”, American Journal of Economics and Business Administration, 2015, 7(2), 60-67.
[3] Ramez Elmasri, Shamkant B. Navathe, “Fundamentals of Database Systems”, Pearson (eds.), India, 2007, pp. 621-622.
[4] Dipina Damodaran B, Shirin Salim, Surekha Marium Varghese, “Performance Evaluation of MySQL and MongoDB databases”, International Journal of Cybernetics and Informatics, 2016, 5(2), 387-394.
[5] Guoxi Wang, and Jianfeng Tang, “The NoSQL Principles and Basic Application of Cassandra Model”, In the proceeding of the International Conference on Computer Science and Service Systems, August 2012, pp.1332-1335.
[6] Sanobar Khan, Vanita Mane, “SQL support over MongoDB using metadata”, International Journal of Scientific and Research publications, 2013, 3(10), 1-5.
[7] Seyyed Hamid Aboutorabi, Mehdi Rezapour, Milad Moradi Nasser Ghadiri, “Performance evaluation of SQL and MongoDB databases for big e-commerce data”, International Symposium on Computer Science and Software Engineering , August 2015, pp.72-78.
[8] Zhu Wei-ping, Li Ming-xin, Chen Huan, “Using MongoDB to implement textbook management system instead of MySQL”, IEEE 3rd International Conference on Communication Software and Networks, May 2011, pp.303-305.
[9]Enqing Tang, Yushun Fan, “Performance Comparison between Five NoSQL Databases”, In Proc. the 7th International Conference on Cloud Computing and Big Data, November 2016, pp. 105-109.
[10] Azhi Faraj, Bilal Rashid, Twana Shareef, “Comparative Study of Relational and Nonrelational Database Performances Using Oracle and MongoDB Systems”, International Journal of Computer Engineering and Technology, 2014, 5(11): 11-22.
[11] http://Nosqldatabase.org
[12]Benymol Jose, Sajimon Abraham, “Exploring the Merits of NoSQL: A Study Based on MongoDB”,IEEE International Conference on networks&Advances in Computational Technologies 2017,pp. 266-271.
[13] Pramod J. Sadalage, Martin Fowler, “NoSQL distilled: a brief guide to the emerging world of polyglot persistence”,Pearson Education, India, 2012.
[14] http://docs.mongodb.com/manual/introduction