Exploring the Merits of NoSQL: A Study Based on MongoDB

Benymol Jose
Research Scholar, School of computer Sciences,
Mahatma Gandhi University, Kottayam
benymol.jose@mariancollege.org

Sajimon Abraham
School of Management & Business Studies,
Mahatma Gandhi University, Kottayam
sajimabraham@rediffmail.com

***Abstract*—**Introduction of structural, semi-structural and unstructured data has tested and challenged the scalability, flexibility and processing ability of the conventional relational database management systems (RDBMS). The new eras of frameworks demand horizontal scaling of the databases. Data with various formats including unstructured data has to be stored and processed in the databases. NoSQL methodologies are answers to these issues. The conventional approaches used in relational and object oriented DBMS cannot adopt the flexible scaling required for today's systems. Likewise, Big Databases have come into existence. But systems with advanced facilities have not emerged, which can support such databases. These enormous databases could not be stored in one physical system. It is recommended to have a distributed framework for this. With a specific goal to enhance the unstructured data processing, a NoSQL framework can be utilized. A document based NoSQL database MongoDB which commonly uses JSON data is discussed to explore the merits of NoSQL databases.

Keywords—RDBMS, Big Data, NoSQL, MongoDB, JSON Data

I. INTRODUCTION

Big data represents huge volume or exponential growth in data. This data comes in different forms and with high speed. The rise of big data is changing the current data storage, administration, processing and analytical methods and leads to the new architecture designs to handle big data applications [1]. Today the discussions of Big Data have turned into an interesting issue over several business ranges. Identifying an efficient storage mechanism is the main concern with Big Data. It should be capable of dealing with the gigantic volume and assortment of data proficiently. Because of the unpredictable nature of the Big Data, which normally contains huge data volume, proficient querying strategies are used for information examination purposes [2]. RDBMS systems can only store schema based data in certain predefined forms. But, the data being produced nowadays have no predefined schema. A prominent portion of the generated digital content has no specific structure and the data comes in several formats such as audio, video, text and so on. NoSQL data bases have been proposed as a solution to the problem of storing the unstructured data. NoSQL gives the flexibility to various sorts of data, structured, semi structured or even unstructured [3]. The important aspects identified as connected to the reliability of the database transactions are lost by including scalability and execution enhancements. In place of relational database management systems, a wide class of NoSQL databases is commonly used nowadays. They are designed as not depending on tables, and also not using SQL queries for data transactions. These systems are streamlined for retrieval and appending operations and provide advantage with storage requirements. There are improvements in scalability and performance of NoSQL systems. It is compensated with the reduction in runtime flexibility. Non-relational database management systems are used mainly to manage huge volume of data. When the data's temperament does not require a relational model, NoSQL databases are used [4].

Different types of NoSQL databases are available and one important one among them is the Document based NoSQL databases. Document databases are for the most part useful for content management systems, blogging platforms, real time analytics, and web based business applications and so on. The main aim of this paper is to show how the data is spoken to and queried in MongoDB which is an example of document databases. The paper is presented as follows: section 2, describes few related studies and section 3, contains the classification of NoSQL databases and some of its advantages are discussed. Section 4, gives the overview of MongoDB and its features and also compares it to the relational databases. In section 5, the JSON representation of collection in MongoDB and screen shots of the execution of queries in MongoDB are included in section 6. Finally, the paper is concluded with the future scope of study and investigation.

II. RELATED WORK

Because of the need of using NoSQL databases to include data diversity a substantial work is done in this area. In one of the recent studies a new method was proposed to integrate MySQL and MongoDB by adding a middleware between application and database layers [5]. The middleware comprises of metadata which contains diverse types of packages. In another work three popular NoSQL databases were looked at and compared namely, Cassandra, MongoDB, and CouchDB [9]. The study revealed that each was developed with its own advantages and disadvantages. In another research work a proposal is made for a data mining framework using the Kalman Filter algorithm to determine term association between topics in the data source. The flexibility of the framework is the ability to perform data mining tasks from the document source directly via HTTP without any copying or formatting of the existing JSON data. It supports topic extraction, term organization, term classification and term clustering [3]. As continuation to it another research focused on topics and terms mining, based on clustering, in document-based NoSQL. This is achieved by adapting the architectural design of an analytics-as-a-service framework and the proposal of the Viterbi algorithm to enhance the accuracy of the terms classification in the

978-1-5090-6590-5/17/$31.00 ©2017 IEEE

system. [6]. Again, another research work proposed a data analytics framework that enables knowledge discovery through information retrieval and filtering from document based NoSQL focusing on terms and topic mining. This architecture is built and tested algorithmically. It is done using two methodologies namely the Inference based Apriori and the Baum-Welch algorithm [7].

III. NoSQL DATABASES

A. Classification of NoSQL Databases

The different types of NoSQL are:

1) Key-Value databases

These are the commonly used NoSQL data bases. This type of databases allows the user to do operations on the database based on a key-value pair. As key-value stores depend mostly on primary-key access, they have extraordinary performance compared with other NoSQL databases and can be easily scaled. Riak, Redis, Memcached and Couchbase are some of the commonly used key value databases.

2) Document databases

Documents are the fundamental idea in document databases. Document databases stores and recovers XML, JSON, BSON documents. Document databases store all data related with a given object as a single instance in the database. Because of this document stores are used extensively for web application programming. They provide for embedded documents, which are self-describing, hierarchical tree structures that often comprise of maps, collections, and scalar values. Document databases like MongoDB has very powerful and rich query language and constructs, with which a simpler move from relational databases is possible. Some of the well-known document databases are MongoDB, CouchDB, Terrastore, OrientDB and RavenDB.

3) Column family stores

These databases store data in column families as rows. These rows have numerous columns related with the key of that row. These are collections of data that is related and accessed together. Each column family can be contrasted to a container of rows in an RDBMS table. In RDBMS systems, the key is used to recognize each row. Here each row comprises of numerous columns. Cassandra is one of the prominent column-family databases and other important column value databases are HBase, Hypertable and Amazon DynamoDB.

4) Graph Databases

Graph databases facilitate the storage of entities and connections or relationships between them. Another name for entities is nodes, which have properties associated with it. Occurrence of an object in the application is noted as a node. The lines connecting the nodes show the relations and are known as edges, which can have properties. Edges show the directional significance and nodes are organized based on these relationships. With this arrangement, the interesting patterns between the nodes can be identified. Also Graph databases allow the interpretation of the data in different ways based on relationships even though it is having a single instance of storage. Neo4J, InfiniteGraph, OrientDB, or FlockDB are some of the popular graph databases commonly used. [8].

B. Advantages of NoSQL

1) Elastic scaling

Often database administrators are buying bigger servers to meet scaling up as the volume of data increased. Usage of multiple hosts was not possible to allocate the big load generated. The financial focal points of scaling out on commodity hardware are becoming more strong and powerful, as transaction speed and rates and also requirements of storage increase. The same problem occurs when databases move into the cloud or onto virtualized situations. RDBMS cannot be scale out easily on commodity clusters. NoSQL databases can be used in these situations which grow easily to exploit the advantage of new nodes. Equipment with low costs is used for building NoSQL databases.

2) Big data

The volume of data managed today has increased lot. It is as a consequence of increased number of transactions in the current scenario. Even though the RDBMS capacity has been increased to incorporate these new requirements, the practical management of the system is becoming difficult. Today only the NoSQL systems can manage the huge volumes of big data which is generated very fast

3) No DBAs

In spite of the numerous reasonability changes guaranteed by RDBMS sellers throughout the years, top of the line RDBMS frameworks can be kept up just with the help of costly, exceptionally prepared DBAs. DBAs are personally required in the outline, establishment, and continuous tuning of top of the line RDBMS frameworks. NoSQL databases are for the most part composed from the beginning to require less administration, programmed repair, information dissemination, and more straightforward information models prompting lower organization and tuning necessities.

4) Economics

NoSQL databases normally use many servers to store the huge data using different techniques. RDBMS often use costly servers for this purpose. This increases the cost involved tremendously in the case of RDBMS systems whereas NoSQL can be managed with normal costs.

5) Flexible data models

Administrating change is a big problem for substantial generation RDBMS. Indeed, even minor changes to the information model of a RDBMS must be precisely overseen and may require downtime or decreased administration levels. Key value stores and document databases are used to enable the application to store data of

any structure it needs in an information component. Indeed, even the all the more inflexibly characterized BigTable based NoSQL databases (Cassandra, HBase) normally enable new sections to be made without an excessive amount of complain. The application changes and database mapping changes are easily possible in the case of NoSQL databases [10].

IV. MongoDB

MongoDB is an example of document oriented database. It is difficult to store large volumes of unstructured data using the conventional relational database management systems. The important storage components in document databases like MongoDB are collections, instead of tables in the case relational databases. These collections in MongoDB consist of similar or different JSON, BSON based documents or sub documents. Those documents that share some form of similarity in structure are organized as collections. It can be made whenever needed, without any predefinitions. Instances within instances or documents with in documents, even lists or arrays of documents are possible with MongoDB. A document in MongoDB can be of any fundamental data type, such as dates, arrays, numbers, strings, or a sub-document. MongoDB has unique multiple storage engines within a single deployment. This arrangement is useful for moving data between storage engine technologies. It is done utilizing local replication. The following figure Fig.1 shows the adaptable storage architecture of MongoDB. MongoDB 3.2 has four productive storage engines as shown in Fig. 1, and they all are available within a single replica set of MongoDB. The WiredTiger engine in MongoDB has concurrency control and native compression properties. Also, it has the best storage and performance efficiency.

The concurrency control and native compression with best storage and performance efficiency are achieved through the default Wired Tiger storage engine. MongoDB permits both the blends of in memory motor for ultralow latency operations with a disk based motor for altogether existence. It permits to assemble large scale, very accessible, robust frameworks and empowers diverse sensors and applications to store their data in a schema adaptable way. There is no database blockage as we experience during alter table commands in RDBMS, which is used to change the schema. However, in uncommon cases, such as during the write-intensive scenarios in master-slave nature of MongoDB there may be blockage at the document level or bottleneck to the system if sharding is not used. Sharding is a type of database partitioning that separates very large databases into smaller, speedier, more effortlessly managed parts called shards. MongoDB empowers horizontal scalability because table joins are not essential as they are in conventional RDBMS.

MongoDB has the property of auto sharding in which more replica server nodes are added to the system. It is a very fast database and gives indexes not only on the primary attributes but also on the secondary attributes. This facility is also, available even within the sub-documents. And the comparison of various collections is performed using technologies such as aggregation framework, Map Reduce and Hadoop systems.

Fig 1. Flexible storage architecture, optimizing MongoDB for unique application demands.

Some important features of MongoDB are:

1) *Flexibility:* MongoDB stores data in document format using JSON. It is schema less and maps to local programming language types.

2) *Rich Query language:* Dynamic queries, sorting, secondary indexes, rich updates, easy aggregation are few RDBMS features available in MongoDB. Also, it provides adaptability and scalability.

3) *Sharding:* This allows linear scaling of the cluster. It is made possible by adding more machines. Because of sharding, the efficiency can be maintained even if there is an unexpected increment of load in the web.

4) *Ease of Use:* MongoDB is a freely available document database which is easy to install. Also, using, maintaining and configuring it are also very simple.

5) *High Performance:* It provides faster query processing. It is achieved by supporting embedded documents and indexing. It supports speed by reducing I/O actions on database systems.

6) *High availability:* MongoDB allows replica set. It is the gathering of servers that keeps up same dataset.

7) *Support for Multiple Storage Engines:* It uses Wired Tiger storage engine which has multiple storage engines. It has additional support for pluggable storage engine API that permits an outsider to develop storage engine for MongoDB [11].

In the following table, Table I an examination between the ideas utilized as part of ORACLE and MongoDB is made [12].

TABLE I: A comparison of concepts used in ORACLE and MongoDB

ORACLE	MongoDB
Database Instance	MongoDB instance
Schema	Database
Table	Collection
Row	Document
Row_id	-id
Join	DBRef

And the table, Table II looks at the basic queries used in a relational database and in MongoDB [13].

V. Data Storage in MongoDB in JSON format

Java Script Object Notation (JSON) is based on a subset of the Java Script programming language. It is a light weight data interchange format. It is simple for humans to read and compose. It is simple for machines to parse and produce. It is data in light of a subset of the Java Script programming language. JSON supports all the fundamental data types, numbers, strings, and Boolean values, as well as arrays and hashes. Document databases such as MongoDB use JSON ore records in a conventional database. Here is a document in order to store records, just as tables and rows example of JSON document:

```
{“first_name”: “John”, “last_name”: “Smith”,
“age”: 25, “address”:
    {“st_address”: “21 2nd street”,
        “city”: “New York”,
        “State”: “NY”,
        “Postal”: “10021”
    },
“phonenumber”: [
        {“type”: “home”, “number”:
        “212555-1234”},
        {“type”: “fax”, “number”:
        “646555-4567”}],
“gender”: {“type”: “male”}
}
```

The above example demonstrates a possible JSON representation depicting a person**.** The JSON format presents data of a document as a name, value pair. All related data of the user is put away into one document. In this way, many related documents are put away together into a collection.

Each document may contain more sub documents known as embedded documents.

TABLE II: Queries Used in Two Different Databases

Query	Relational Database	MongoDB Database
Create Command	CREATE TABLE table_name (col_name1 datatype, Col_name2 datatype)	No Schema
Insert Command	INSERT INTO table_name (col_name1,col_name2) VALUES(value 1,value2)	db.collection-name.insert ({name1:value1,name2 :value2})
Delete Command	DELETE FROM table_name WHERE (Condition)	Db.collection_name.remove ({condition})
Select command	SELECT column_name FROM table_name	Db.collection_name({} ,{condition})
Import command	BULK INSERT table_name From filename WITH {FIELDTERMINATOR=’,’, ROWTERMINATOR=’\n’} GO	mongoimport --dbdatabase_name –collection collection_name –type csv --file ‘file_name”

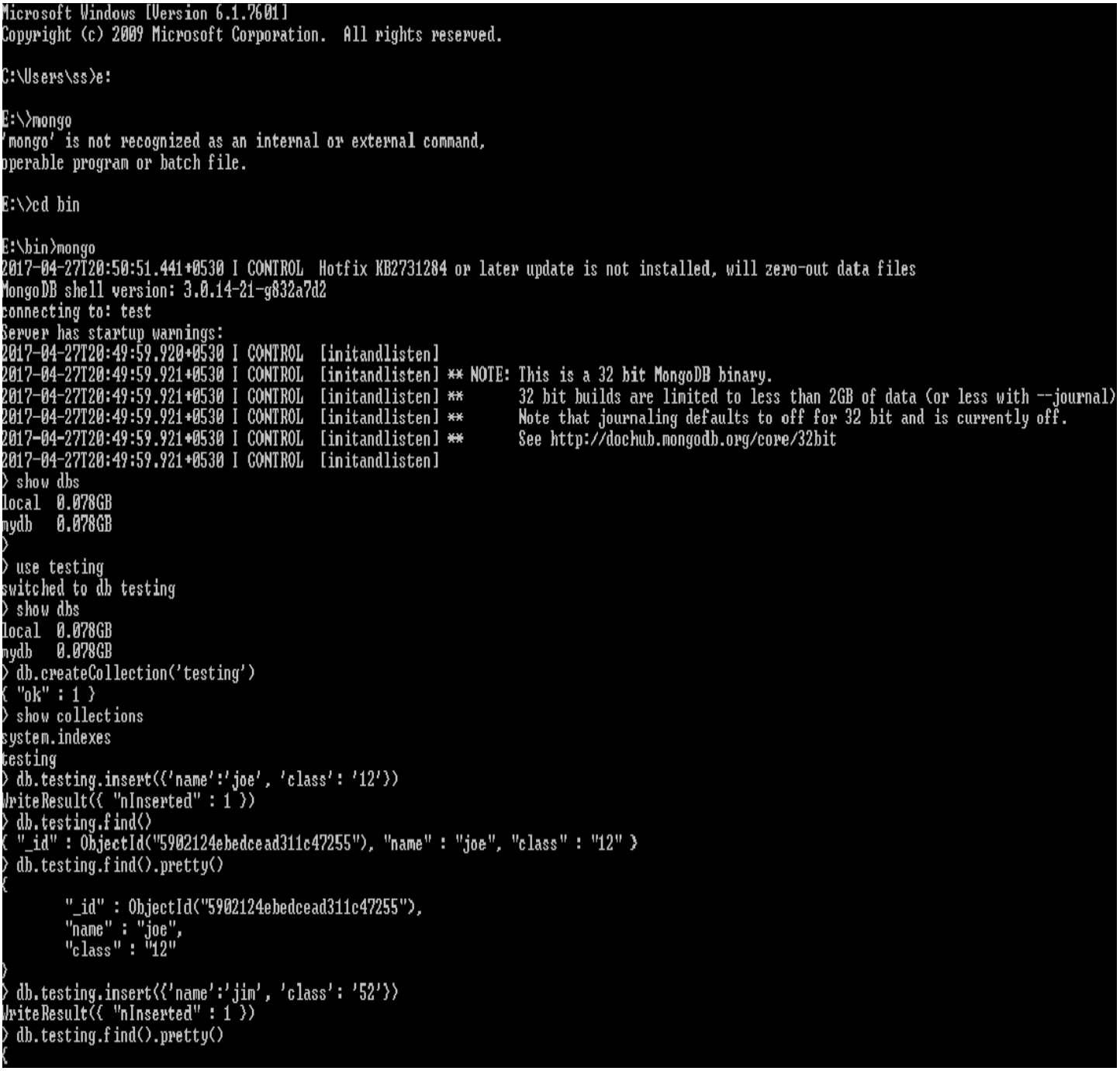

Fig. 2 Sample Screen Shots

VI. Screen Shots: Execution of Queries in MongoDB

The execution of the queries in MongoDB for creating a database, inserting a document, searching a document, deleting and updating documents are shown in the following figures Fig. 2 and Fig. 3. The different commands used are:

- use testing
- db.createCollection(‘testing’)
- show.collections.
- db.testing.insert({‘name’: ‘jim’,’class’:’2’})
- db.testing.find ()
- db.testing.find.pretty ()
- db.testing.update({‘name’:’jim’},{$set:{‘class’:’50’}})
- db.testing.remove

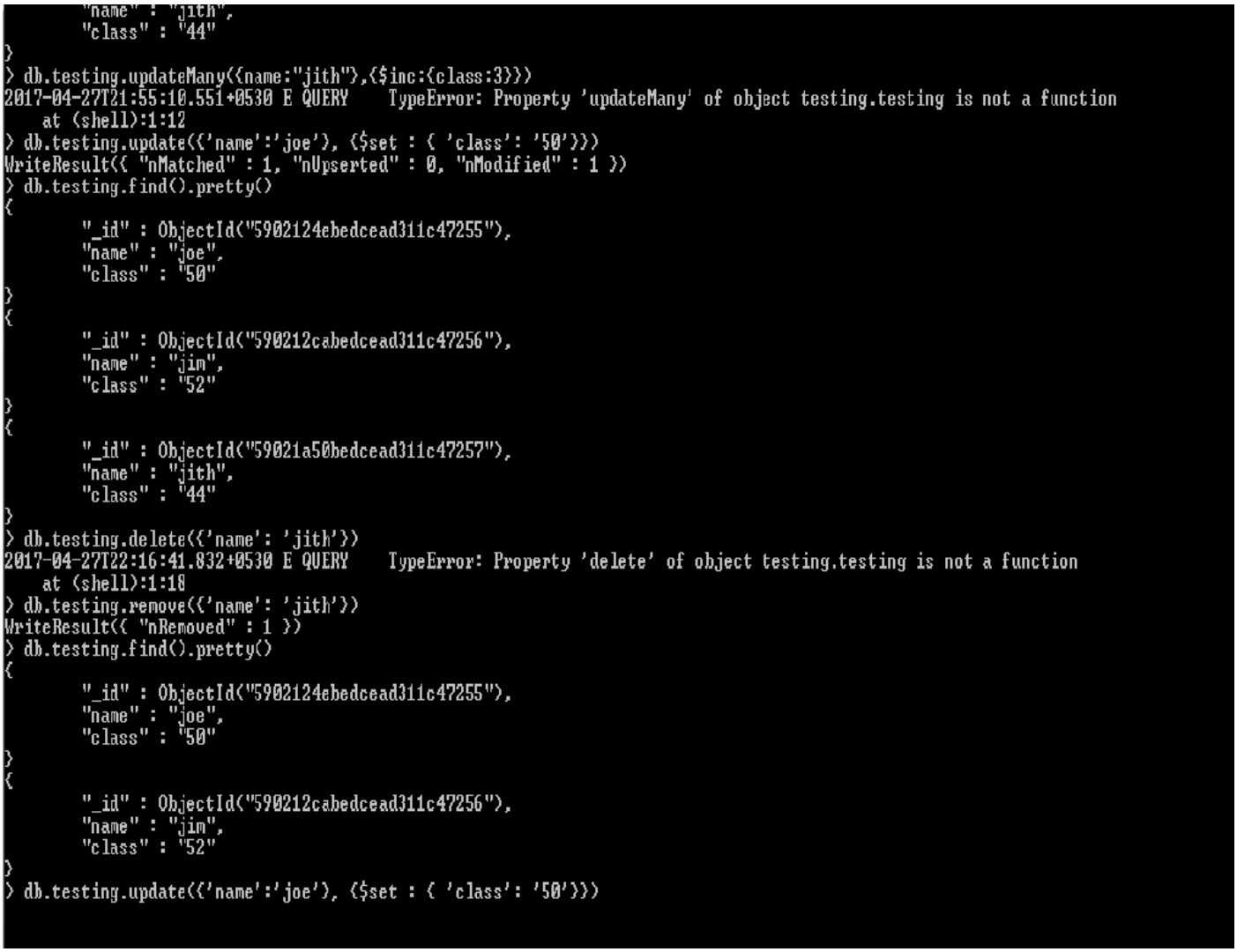

Fig. 3. Sample Screen Shots

CONCLUSION

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. This paper highlights some features of document databases particularly MongoDB. The data representation of JSON data in MongoDB is talked about and also the screen shots of the execution of few queries in MongoDB are included. As a future scope of this study, the query performance of MongoDB can be compared with CouchDB which is another document based NoSQL database. Another possibility is to do performance comparison between a document database and a traditional database management system. It can be performed considering the cases of MySQL and MongoDB.

REFEENCES

[1] Yesha Mehta, and Sanjay Buch, "Big Data Mining and semantic technologies: challenges and opportunities", Int. J. on Recent and Innovation Trends in Computing and Communications(ISSN 2321-8169) Vol.3,No 7,2015.

[2] Rodrigues M., Santos M.Y., Bernardino J., "Describing and comparing Big Data advances in information systems and technologies". WorldCIST 2017 Advances in Intelligent Systems and Computing, Vol 569. Springer, Cham,2017

[3] Lomotey ,RK and Deters, R , "Data Mining from Document – Append NoSQL", Int.J.of Services Computing(ISSN 2330-4472),Vol.2,No.2,2014.

[4] Nadeem Qaisar Mehmood,Rosario Culmone and Leonardo Mostarda,"modeling temporal aspects of sensor data for mongoDB NoSQL database",Journal of Big Data,Springer,2017

[5] Sanobar Khan, Prof.Vanita Mane, "SQL support over MongoDB using metadata". International Journal of Scientific and Research publications, Vol 3,Issue 10,2013

[6] Lomotey, RK and Deters, R "Terms Mining in Document-Based NoSQL: Response to Unstructured Data", IEEE Int. Conference on Big Data, ISSN: 2379-7703, 2014

[7] Lomotey, RK and Deters, R, "Terms analytics service for CouchDB: a document –based NoSQL", Int. J. Big Data Intelligence, Vol. 2, No 1, 2015.

[8] http://Nosqldatabase.org

[9] Christina Bazar, Cosmin Sebastian IOSIF, "The transition from RDBMS to NoSQL. A comparative analysis of three popular non-relational solutions: Cassandra, MongoDB and Couch base", Database systems Journal, Volume 5, Issue 2, 2014

[10] Nancy Bhardwah, Gurvinder Kaur, "Greedy based privacy preserving in Data Mining using NoSQL", International Journal of Engineering Development and Research, Vol5, Issue1, ISSN: 2321-9939, 2017

[11] Divya Chauhan, KL Bansal , "Using the Advantages of NoSQL: A case study on MongoDB", Int. J. on Recent and Innovation Trends in Computing and Communication 5, Issue 2, ISSN:232/-8169

[12] Sadalage, P. J., & Fowler, M. (2012). *NoSQL distilled: a brief guide to the emerging world of polyglot persistence*. Pearson Education.

[13] http://docs.mongodb.com/manual/introduction