SQL "+" NoSQL, A Merger with Great Capabilities

Wumi Ajayi[1]
[1] Computer Science Department, School of Computing and Engineering Sciences, Babcock University, Ilisan Remo, Ogun state, Nigeria.

Abstract

Analysis of SQL and NoSQL databases is presented here with an intention of prescribing how they can work together for better capabilities. The work begins with short background on the basic terms (data, database and database management system) used in the database paradigm. Next a review of some closely related work is presented; followed by an analysis of the two databases with special attention on query structure, standards and characteristics.
Then special features that could make then work together were presented before concluding on recommendation. To aid smooth reading and understanding, the work is presented in sections.

Keywords*: Databases, NoSQL, SQL, Database Integration, MongoDB, MySQL*

I. INTRODUCTION

A database is "an integrated collection of logically related records or files consolidated into a common pool that provides data for one or more multiple uses" [7]. While a Database Management System (also called the DBMS) is a collection of interrelated data and a set of program to access those data. Generally, the DBMS can be viewed as a system designed to handle the storage and retrieval of large body of information stored in the database [9], [13].
Data that exist in databases are organized based on a data model. According to [13] book on database system concepts, there are a number of different data models. An important point to note is that development process may differ depending on the environment being modeled or considered.
Prior to the full development and implementation of DBMS, a major way of keeping information on a computer is by storing it in operating system files [13]. Example of such system is the file processing system. The system has a number of application programs that manipulate the files. For instance, in a banking environment this may include programs to:

(i) Add new customers, customer managers, and accounts
(ii) Put customers up for loan as they apply and merit it
(iii) Authorizes loan to customers, compute interest and generate balance and statements

These application programs are written to meet the needs of the bank and new ones can be written if there are needs for them. Invariably, as this goes on the system then acquires more files and more application programs.
As stated earlier, to get information from a database as shown in figure 1, In SQL a set of program called queries would be needed to access the data.
Based on [13] there are many database query languages in use; and the most widely used is the SQL. Also the work of [11] and [8] on "A Comparison of SQL and NoSQL Databases" established that other databases known as NoSQL could do well (or even better) where SQL is being implemented.
To this end, in this work, a deeper look is taken into some basic principles of these " DB Rivals" to establish the power of both and how they can be combined with the together for stronger capabilities. In an attempt for the merger, some basic operation of both query languages such as their data definition, basic query structure, standards and characteristics will be considered.
For clarity and to reach a logical conclusion on the subject, the work is structured into different sections and subtopic which shall be examined in turns.

A. Aim and objectives

The aim of this work is to explore the possibilities of merging the SQL and NoSQL for stronger capabilities in databases. The specific objectives are to

a) analyze basic standard, characteristics and operations of SQL and the NoSQL

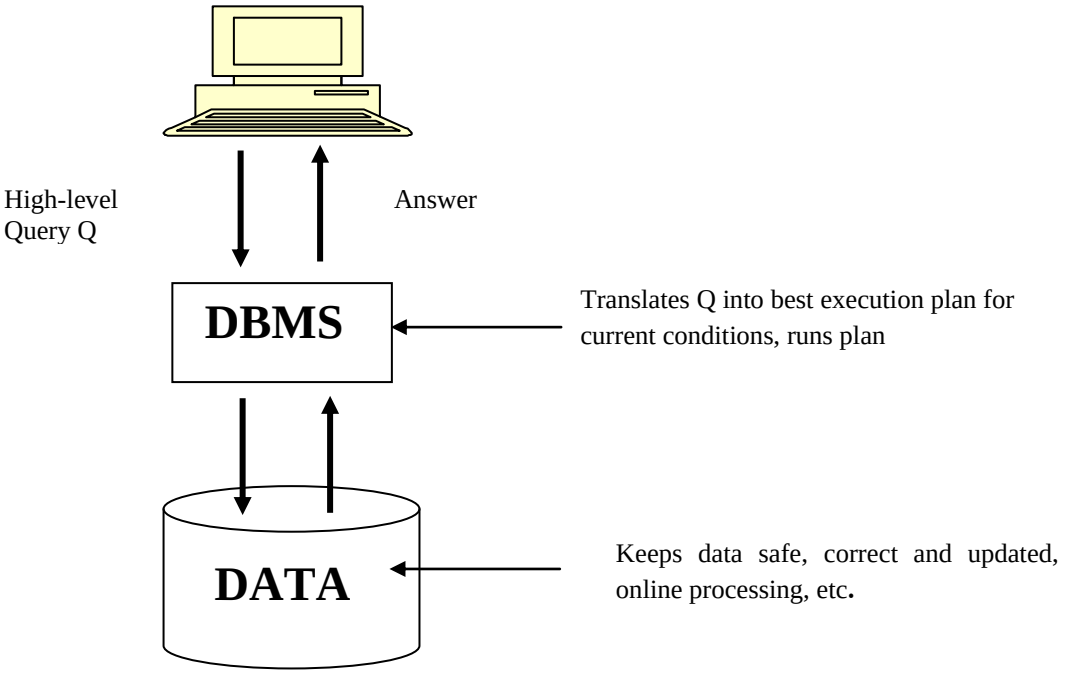

Figure 1 showing link between of DBMS and Database.
Source: [3]

b) establish the strength of both SQL and NoSQL in database models operation

c) Identify reasons for moving from SQL and NoSQL and how they can be combined with together for stronger capabilities

B. Methodology

As stated in the aim, the center of this work is to look into a possibility of merging the SQL and the NoSQL. To arrive at a sensible conclusion at end of this work, several literatures were consulted and analysed for facts on both query languages. Again, some closely related works in this paradigm were analysed to see existing areas where they have been used separately and concurrently.

It should be noted that no data on the usage of SQL and NoSQL is collected. All analysis (and discussions) done in this work are based on data from existing work.

II. LITERATURE REVIEW

In this segment, an analysis of the two query languages (SQL and NoSQL) with particular reference to how data is defined, basic query structure, standards, and modification of the database is presented. Next, we look at some closely related works already done in this paradigm to see if there exist some gaps in research.

A. Essential reviews

The work of [8] on "A comparison of SQL and NoSQL Databases" presented an overview of both SQL and NoSQL including the definitions, characteristics, products and projects, the ACID properties of SQL and the corresponding BASE transaction in the NoSQL. The work is a very good insight on fundamental topics for understanding both SQL and NoSQL. However, author's analysis only dwells on comparison using some basic characteristics, definition and usage. It didn't provide any form of combined implementation or merger of both or any area of implementation.

In similar view by [10] on "Comparative analysis of NoSQL (Mongo DB) with MySQL Database", the rising of non-relational databases (also called the NoSQL) is seeing as a result of new requirements with corresponding need in handling larger volume of data. The work began with an overview of the MongoDB, its architecture which comprises of document data representation and dynamic schema, query model and auto sharing. Authors' aim is focused on using one of the NoSQL – the MongoDB in place of a flavor of the traditional SQL (MYSQL). Ease of use and timing performance are the two identified reasons why MongoDB is always chosen before MySQL. The latter part of the work present a comparison between the duo (i.e. MongoDB and MySQL) based on terms and concept, based on

queries and based on query execution speed and performance.

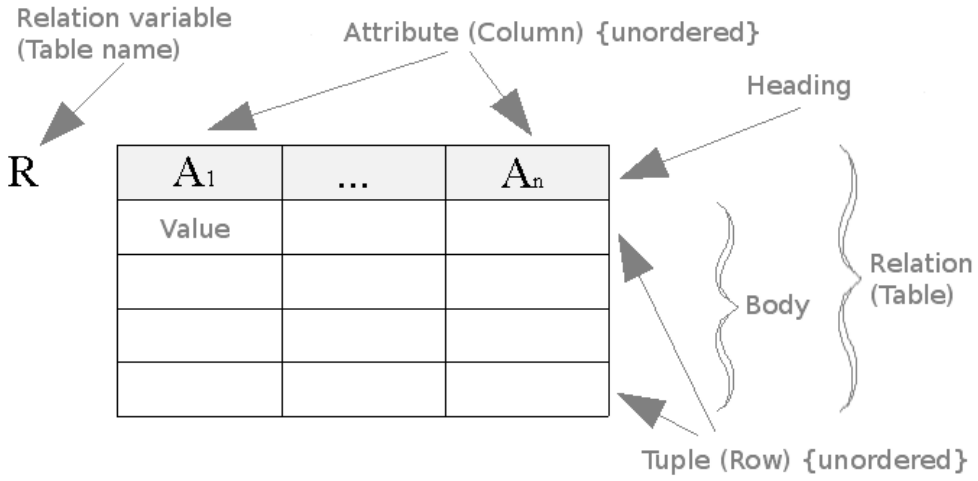

Figure 2: showing Tuples, attributes and Relation variable
Source: (Wikipedia in [2])

However, against expectations, the basis for comparison seems too narrow as no analysis or attempt on trying to make them work together is presented; except brief mention of it in the conclusion. Reference [6] in a work titled "From Only-SQL to NoSQL to YeSQL Solving the data scaling challenge without a complete rewrite" emphasize drivers for adopting alternatives to the conventional SQL database. As supported by [10], the author reasoned that a number of application developments cannot be addressed with the traditional database method due to continuous increase in the volume of data being modeled. The cost of development and implementation is another major issue to consider when using SQL. [6] Further addressed the alternatives available to the use of SQL alone. Some of these methods proposed include: "MongoDB, Microsoft's Azure Table, Amazon's SimpleDB and Google's App Engine Megastore".

All analysed works above seems to support that the trend of usage or adoption is tending towards the NoSQL [6], [10].

Nevertheless, we need not jump into conclusion yet; at least not until we have examined some essential details and basic characteristics of both separately. Hence, in what follows, based on the gaps observed during the analysis of related works, we try to examine both SQL and NoSQL separately and then propositions will be made on how they can work together on the same task.

B. SQL or NoSQL

The SQL

Reference [3] on "fundamentals of Database System" explains SQL (structured Query Language) as a common language used for diverse databases. The SQL contains statements for data definitions, queries and updates both for data definition language (DDL) and data manipulation language (DML). SQL language has several parts [13]. These parts include "the Data-definition language, Data-manipulation language, Integrity, View definition, Embedded SQL and dynamic SQL, Authorization". The structure of SQL query is very different compared to that of a typical NoSQL. A good way to explain this is through the query and data structure.

SQL Query and data Structure

A typical SQL statement consist of the following

"SELECT $A_1, A_2, A_3, \ldots\ldots, A_n$
FROM $r_1, r_2, r_3, \ldots.., r_m$
WHERE P"

From the query above and as depicted in figure 2. A_i represents an attributes, R_i represents a relation and P is a predicate. The data in SQL is stored in a table with a predefined structure which can then be queried using any of the fields.

A. SQL : Basic Characteristic

Based on [8] the basic characteristics of SQL include the following:

I. Data is stored in columns (with attributes header), turples and in a relation (also called tables)
II. Relationships are represented by data
III. Data is manipulated through the use of data manipulation language. For instance the use

of select, insert, update and delete statements, data aggregation, functions and procedures etc

IV. Data Definition Language is another major characteristic fully exhibited in SQL. They include definition of schema at the start of the design, the create table command, security and access control, alter and drop and so on.

V. **Transactions – ACID Properties**
All transactions carried out in SQL must conform to the ACID properties [3]. Succinctly, the ACID property means:
Atomicity: Either all operations of the transaction are properly reflected in the database or none are.
Consistency: Execution of a transaction in isolation preserves the consistency of the database.
Isolation: Although multiple transactions may execute concurrently, each transaction must be unaware of other concurrently executing transactions.
Durability: After a transaction completes successfully, the changes it has made to the database persist, even if there are system failures.

VI. Abstraction from physical layer

B. The SQL standard

As presented in both [3],[13], the SQL standard supports diverse of built-in types. These include:
"Char (*n*): A fixed-length character string, the length of n can be specified by the user.
varchar(*n*): A variable-length character string with user-specified maximum length *n*. The full form, **character varying**, is equivalent.
int: An integer (a finite subset of the integers that is machine dependent). The full form, **integer**, is equivalent.
smallint: A small integer (a machine-dependent subset of the integer type).
numeric(*p*, *d*) :A fixed-point number with user-specified precision. The number consists of *p* digits (plus a sign), and *d* of the *p* digits are to the right of the decimal point. Thus, **numeric** (3,1) allows 44.5 to be stored exactly, but neither 444.5 or 0.32 can be stored exactly in a field of this type.
real, double precision: Floating-point and double-precision floating-point numbers with machine-dependent precision.
float(*n*): A floating-point number, with precision of at least *n* digits"

C. Notable Pro and cons of Using the SQL

As identified from literature, amongst others the following are the Pros & Cons of using the SQL.
Pros:

a) The use of the SQL comes with very good flexibility
b) It is universal (Oracle, Access, Paradox, etc)
c) Has an averagely few commands to Learn

Cons:

a) It requires detailed knowledge of the structure of the database
b) likelihood of providing misleading results is high, since it can produce duplicate result

NoSQL

The NoSQL stands for "not only SQL" [11]. According to this author, the NoSQL is "a set of theory, ideas, technologies, and software which has to do with big data, high horizontal scalability, sparse un/semi-structured data and massive parallel processing".
Generally, the name NoSQL is a term used normally for all databases that does not conform to the principles and approach of a traditional RDBMS. Based on Couchbase's technical report on "Why NoSQL?" and [6] the need for NoSQL arise from:

a) Increase in the numbers of application user which are going online.
b) The Internet is now connecting everything through the concept of Internet of things
c) The incessant increase in data volumes (known as Big Data).
d) Most Applications now use more of cloud resources
e) Advent of mobile technology.

Contrary to the SQL and other RDBMS, architecture, queries etc are done a little differently while using the NoSQL. As a matter of fact, different applications, paradigm, approaches require different NoSQL solutions.

Base on [4] -"NoSQL Database Architectural Comparison" and [11] there are many variants of NoSQL around today.
These amongst others include: Riak, Cassandra, Couchbase, MongoDB and the HBase. In what follows, the basic concepts of the NoSQL databases is analysed. But just before then, figure 3 shows the "NoSQL Toolbox which connects the techniques of NoSQL databases with the desired functional & non-functional system properties they support".

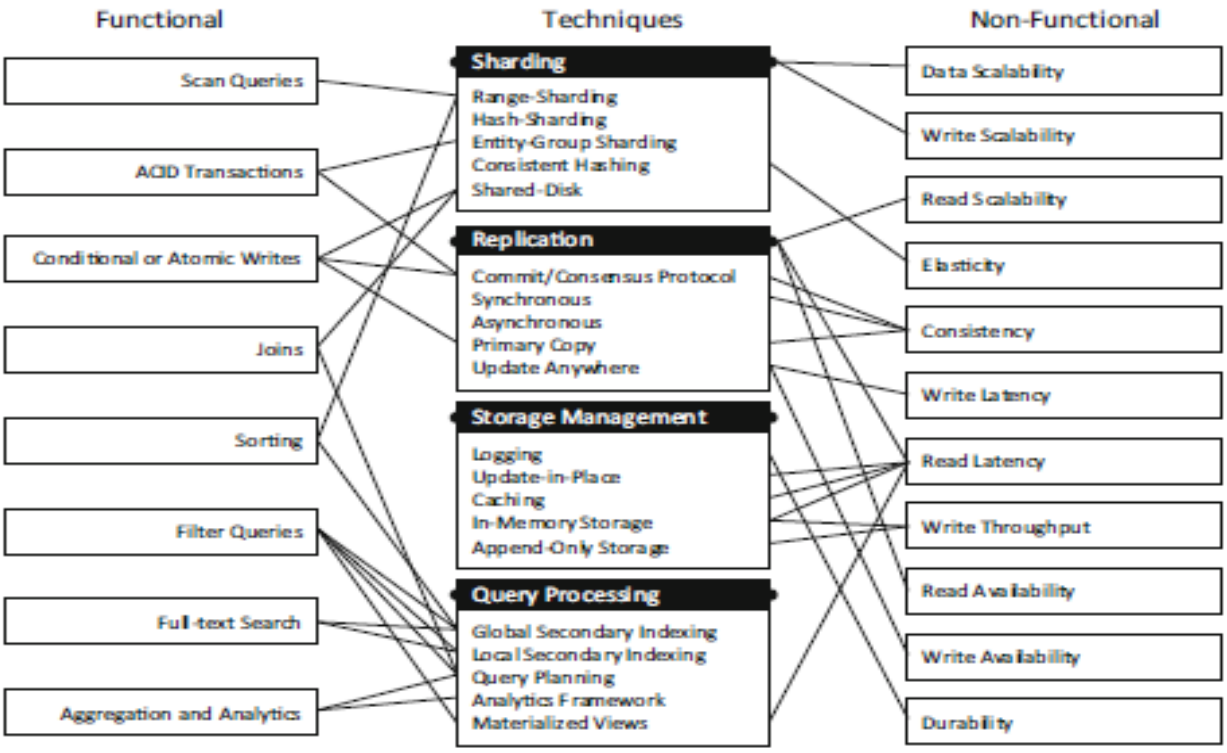

Figure 3 "The NoSQL Toolbox: showing how it connects the techniques of NoSQL databases with the desired functional & non-functional system properties they support"

Source: [5]

A. NoSQL Data Structure and Architecture

All NoSQL do not share the same structure, different databases have different data models, which mean Riak, Cassandra, Couchbase, MongoDB and other variants of NoSQL all have different data structure and maintain their differences in architectural formation too. For instance, in terms of architecture, the MongoDB architecture is hierarchical in nature (see figure 4) while that Cassandra is ring topology.

B. Characteristics of NoSQL

Amongst others, the following are some of the main distinguishing characteristics of NoSQL.

a) Capable of handling large data volumes e.g. Google's "big data"
b) Scalable replication and distribution .e.g. potentially thousands of machines, potentially distributed around the world
c) The queries must return answers as quick as possible. Although the answers may not be a distinct one
d) NoSQL comprises of mostly queries with few updates.
e) Allows asynchronous Inserts & Updates in the database.
f) Schema-less
g) BASE (which stands for Basically Available Soft state Eventually consistent) replaces the ACID transaction properties of the SQL
h) CAP Theorem – Consistency, Availability and Partition tolerance- which stipulates what a distributed system can support.
i) Open source development

C. **Summary:** SQL has ACID transaction properties while NoSQL has BASE characteristics.

III. WHAT THEN? SQL OR NOSQL

After a critical assessment of the related works on the subject being considered here vis-à-vis the separate analysis on SQL and NoSQL; recommendations on whether to use SQL or to accept the use of the NoSQL depends on the choice of the designer and the type of application or enterprise.
Though the proponents of the NoSQL advocates the use of a variant of it (say the MongoDB) because it is the trend and possibly due to its capacity to process huge amount of data compared to the RDBMSs.

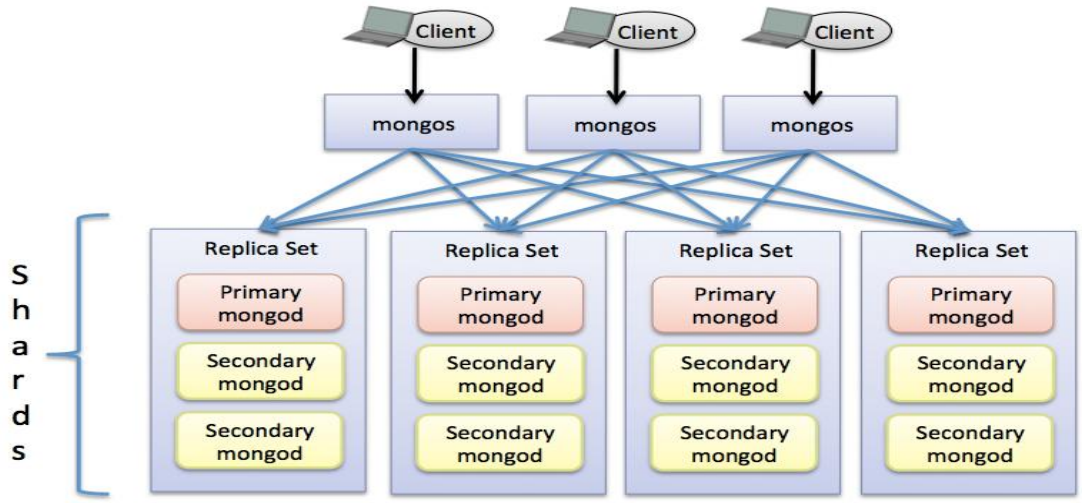

Figure 4 : showing MongoDB's Hierarchical topology

On the other hand, one of the main strengths of the SQL is the capacity to scale vertically (which also comes with some limitations).
SQL proponent may count on this and several implementation experiences over the years amongst others as basis for their support.
Again, SQL databases have predefined schema with standard definition and interface languages, tight consistency along with well defined semantics; while the NoSQL Database has no predefined Schema, it is done per-product definition and interface language. Above all, timing and quick result is very important in NoSQL databases. This means getting an answer quickly is more important than getting a correct answer. A question yet unanswered is: which of the databases is better?

Table i
Acid and base comparative representation.
Source: [2]

ACID(relational)	BASE (NoSQL)
Strong consistency	Weak consistency
Isolation	Last write wins
Transaction	Program managed
Robust database	Simple database
Simple Code (SQL)	Complex code
Available and consistent	Available and partition-tolerant
Scale-up(limited)	Scale-out(limited)
Shared (disk, memory, proc etc)	Nothing shared (parallelizable)

A. Is "SQL + NoSQL" then Possible? :

In their concluding remarks, [5] stated that: "Choosing a database system always means to choose one set of desirable properties over another". Generally, a cogent point to note is that there are two sides to both SQL and NOSQL databases; they both have advantages and disadvantages. If SQL is set aside (for a while), even amongst the NoSQL database flavors there are differences either in the features or the way operations are carried out. For instance, in the areas of how data is stored and accessed [5]; data storage in NoSQL variants can be either "keyvalue store, document store or wide-column store". Table II shows a summary of feature comparison of MongoDB and CouchDB. Without doubt, the NoSQL is as strong as the SQL. However,, a major concern for experts (or designers) is how to balance the "power tussle" in the usage. Deductions from Couchbase's report [5] and some other literatures mentioned earlier affirms that NoSQL has gained ground as the preferred database technology to power major applications such as Internet of things (IoT), mobile and web applications, whereas the SQL is a full-grown query technology which has been used by many around the world and supported by virtually all programming languages. Again, the NoSQL (e.g. MongoDB) has the advantage of horizontal expansion, but for complex SQL requests the NoSQL may not be at optimum; therefore the SQL may be applied.

Table ii

Feature comparison of mongodb & couchdb as presented by [12]

Model / Features	MongoDB (Year: 2007)	CouchDB (Year: 2005)
Relational nature	No	No
Developer	10gen	IBM (Damien Katz)
Written in	C++	Erlang
Query language	MongoDB ad-hoc query language	JavaScript
SQL nature	No	No
High availability	Yes	Yes
High scalability	Yes	Yes
Single point of failure	No	No
Open source	Yes	Yes
Versioning	Yes (different database for audit trails, not native)	Yes (MVCC)
Indexing	Advance and wide variety (includes spatial indexing)	Basic associated map and reduce functions
Data Processing nature	Batch processing and event streaming	Batch processing

B. Propositions

Given these identified strength and weakness on both sides plus others already identified factors from the analysis of the databases separately, it is therefore proposed that database designers should endeavor to do away with the contentions (or/ and debates) on which is stronger or better between the databases. Instead, more prominence and focus should be given to *how to use the two databases concurrently*. It will be of immense benefit (both to the designers and the industry) if database designers can develop their individual abilities to the "utmost or professional level" at which they are able to balance their skills in making the features, tools, capabilities and everything attainable in the usage of both databases available during application or system deployment and use them appropriately to their advantages.

REFERENCES

[1] Aslett, M. (2011). Retrieved Dec 28, 2017, from https://cs.brown.edu/courses/cs227/archives/2012/papers/newsql/aslett-newsql.pdf.

[2] Birgen, C. (2014). Advance Process Simulation, SQL Vs Nosql. University Of Science And Technology. Retrieved Dec 2017, from: http://folk.ntnu.no/preisig/HAP_Specials/AdvancedSimulation_files/2014/AdvSim-2014__Birgen_Cansu_Databases.pdf

[3] Elmasri, R. and Navathe, S. (2011). Fundamentals of Database Systems (6 ed.). (Addison-Wesley, Ed.) Pearson.

[4] FIXSTARS. (2017). NoSQL Database Architectural Comparison. (revision 1.00).

[5] Gessert, F., Wingerath, W., Friedrich, S., and Ritter, N. (2016). NoSQL database systems: a survey and decision guidance. Springer.

[6] GIGASPACES (2011) NoSQL, NewSQL and Beyond: The answer to SPRAINed relational databases, retrieved from https://blogs.the451group.com/information_management/2011/04/15/nosql-newsql-and-beyond/

[7] Halvorsen, H.P. (2016). Software Project management, retrieved from https://home.usn.no/hansha/documents/software/software_development/topics/resources/Software%20Project%20Management%20Overview.pdf

[8] Hare, K. (2011). A Comparison of SQL and NoSQL Databases. JCC Consulting Inc. ISO/IEC JTC1 SC32 WG3

[9] Hoffer, J.A., Prescott, M.B. and Topi, H. (2009). Modern Database Management (9 ed.). US: Prentice Hall. Retrieved from http://fkee.uthm.edu.my/helmy1299/class../dec2213/Present_Bab_8.pdf

[10] Kumar, L., Rajawat, S., and Joshi, K. (2015). Comparative analysis of NoSQL (MongoDB) with MySQL Database. International Journal of Modern Trends in Engineering and research IJMTER, 120-127.

[11] Pozzani, G. (2013). Introduction to NoSQL. Retrieved from http://profs.sci.univr.it/~pozzani/attachments/nosql%20-%2001%20 %20introduction.pdf

[12] Sharma, S. (n.d.). An Extended Classification and Comparison of NoSQL Big Data Models. Retrieved from https://arxiv.org/ftp/arxiv/papers/1509/1509.08035.pdf

[13] Silberschatz, A., Korth, H.F. and Sudarshan, S. (2011). Database System Concepts (6 ed.). India: Mcgraw-hill.