Peertechz

Engineering Group

Trends in Computer Science and Information Technology

DOI http://dx.doi.org/10.17352/tcsit.000002

DOI CC By

Mukhtar Ahmad[1] and Zeeshan Siddiqui[2]*

[1]Shaheed Zulfiqar Ali Bhutto Institute of Science and Technology, Karachi, Pakistan
[2]College of Computer and Information Sciences, King Saud University, Riyadh, Kingdom of Saudi Arabia

Dates: Received: 12 December, 2016; **Accepted:** 29 December, 2016; **Published:** 30 December, 2016

***Corresponding author:** Zeeshan Siddiqui, Faculty Member, King Saud University, Kingdom of Saudi Arabia. Tel: 966114696251, Email: zeeshan.siddiqui@ksu.edu.sa.

Keywords: SQL to NoSQL Transformation; Schema Mapping; SQL Server; HBase; Big Data Analytics; Production Automation

https://www.peertechz.com

Research Article

Leveraging Data Analytics by Transforming Relational Database Schema in to Big Data

Abstract

The growth of data and its efficient handling is becoming more popular trend in recent years bringing new challenges to explore new avenues. Data analytics can be done more efficiently with the availability of distributed architecture of "Not Only SQL" NoSQL databases. Technological advancements around us are changing very rapidly and major shift is being carried out, a shift from relational to non-relational world. More precisely we are talking about the shift from traditional relational database models to non-relational database models. When moving from relational to non-relational models, database administrators face common issues due to the fact that NoSQL is a No-Schema database. Logical mapping of the schema from relational to non-relational models is complex and it is not a standard process. The purpose of conducting this research is to propose a mechanism by which the schema of a relational database management system and its data can be transformed into big data by following a set of standardize rules. This model can be very useful for relational database administrators by enabling them to focus on logical modeling instead of procedural writing for every SQL to NoSQL transition. In this paper, we studied both models and focus our research to present a set of rules and framework that can be used to apply transformation operation in a seamlessly manageable way.

Abbreviation

HDFS: Hadoop Distributed File System; SQL: Structured Query Language; NoSQL: Not Only Structured Query Language; RDBMS: Relational Database Management Systems; ACID: Atomicity, Consistency, Isolation, Durability; BASE: Basically-available, Soft-state, Eventual Consistent; CAP: Consistency, Availability, Partition Tolerance; HUSH: HBase URL Shortner

Introduction

In contrast with traditional relational databases, NoSQL database has four points of interest, that are; it is easily extendable, high in performance, provide flexible data model and high availability [1].

Previous research conducted on this topic is very generalized and does not target any specific Relational Database Management System (RDBMS) product or a practical data collection from any real-time system [2-10].

Studies have shown that this area needs standard set of rules which can help database administrators to carry out transformation task seamlessly [11].

This research is therefore unique in a way that it not only proposes a set of rules and model but also targets transformation of specific software products from Microsoft SQL Server to Hadoop H-Base based on a real world case study.

The term and idea of NoSQL was first used in 1998 by Carlo Strozzi to refer to an open source database that does not depend on SQL interface [12].

Structured Query Language (SQL) is a programming language used for storing and managing data in RDBMS. When we talk about massive information the big data domain is primarily dependent on NoSQL programming model [1]. NoSQL has been observed as an alternative to the customary RDBMS models, While SQL being utilized by industry leader database vendors such as Oracle, DB2, MS-SQL etc. [13], On the other hand there are various NoSQL models and products such as Hadoop, Hbase, MongoDB, Cassandra, Tarantool and Apache Spark [14].

To validate our framework, we implement our framework on a real world application that uses relational database for online processing and utilizes the large amount of data produced from RDBMS which further transformed into Big Data to take benefits of it. Specifically, when the data needs to be analyzed and required for taking business decisions.

Citation: Ahmad M, Siddiqui Z (2016) Leveraging Data Analytics by Transforming Relational Database Schema in to Big Data. Trends Comput Sci Inf Technol 1(1): 012-017. DOI: http://dx.doi.org/10.17352/tcsit.000002

NoSQL empowers application management and minimize the necessity of application modification or change in database schema. Additionally, with the expansion of data, NoSQL databases have better and simpler horizontal scalability. Those databases are equipped for taking advantages of the new clusters and nodes transparently, without requiring involvement from database administrators or manual distribution of data across different nodes [12].

RDBMS is being used for decades and still proved to be a viable solution for many use-cases. However, for modern applications, flexibility is a mandatory requirement in scaling models and data models. Due to the continuous growth in data, complex server resources are being regularly added to serve more users.

In this research, the term "Hadoop" refers to the Hadoop Distributed File System (HDFS) and in addition, refer to other open source software products and technologies developed by Apache Software Foundation and various other open source software vendors [15].

SQL Databases referred to as RDBMS (Relational Database Management Systems) is most widely used and proven approach for database solutions. In RDBMS data storage is done in a structured pattern using the tables and the relation in between them. Although the capacity of SQL allows managing huge amount of data, technically it does not provide an optimal solution to existing Big Data requirements, which requires speedy response, vast scalability and high availability [16].

With the improvement of distributed computing and cloud framework, more applications are migrating to a cloud environment in order to utilize its computing power and influence the benefits of scalability. In the initial era of distributed computing and non-relational database models, Google and Amazon were the first to propose new alternatives to data management. The lack of commercially available alternatives at that time leads to the popularity of their frameworks. The non-relational database technologies proposed are now fulfilling majority of the needs of modern software systems.

Apache HBase is such a system. It mainly utilizes to distribute open source database. Google Big Table [17], is used for the modeling purpose and provide random access to large amounts of data. Apache HBase is becoming an increasingly popular choice of database applications these days [18].

CAP Theorem

When talking about and comparing relational and non-relational database, there are two terms mainly discussed [Figure 1].

1. ACID Properties (Relational Databases)
2. BASE Properties (Non-Relational Databases)

ACID stands for Atomicity, Consistency, Isolation and Durability. This is the core principle behind the RDBMS databases and mainly used for transaction. A transaction can be defined as a logical operation on the data. ACID properties are key to ensure the integrity of the data [19].

BASE is the acronym for Basically Available, Stable state, eventually consistent. It emphasizes on Availability and Scalability using Partition tolerance, Simple and fast, good choice to use where data availability and speed is highly concern. However, it results in weak consistency.

Uniqueness of this Study

There are less number of studies conducted for database schema conversion between relational and non-relational databases. There are even lesser examples into an automated conversion model to get you started with such scenarios [20]. This is a common scenario faced by every database administrator or developer during the transformation from relational to non-relational model [21]. This has leaded us to investigate a more systematic approach to migrate. Migration can be challenging and trivial in terms of two things; Schema Translation and Data Translation

Simple export and import between identical data stores does not solve the problem, as you need to know exactly what you are importing and why.

This paper is further organized as follows: Section III comprises of the literature review in which related studies were carried out. Section IV highlights the proposed approach. Section V contains the experimental details and evaluation of the study followed by the conclusion and future work in section VI.

Literature Review

For RDBMS, the architecture that is most commonly followed is a client-server based architecture in which servers are equipped to handle database application tasks. This architecture is proved to be best and it is highly optimized. However, as the data grows, the RDBMS cannot provide best results, specifically for the read-operations. Log-based and

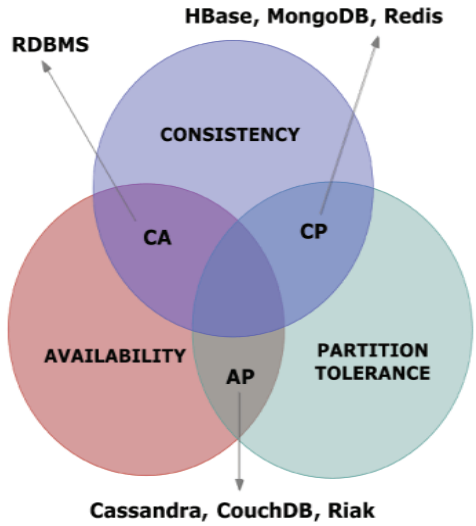

Figure 1: CAP THEOREM AND NOSQL DATABASES.

Citation: Ahmad M, Siddiqui Z (2016) Leveraging Data Analytics by Transforming Relational Database Schema in to Big Data. Trends Comput Sci Inf Technol 1(1): 012-017. DOI: http://dx.doi.org/10.17352/tcsit.000002

fast growing databases requires other type of database which is capable enough to support fast read-operations. NoSQL is the answer for such situations. NoSQL databases are proven to be best suited in these scenarios and provide ample control over the scalability and high availability altogether with distributed architecture by distributing the data over various nodes [22].

We studied various NoSQL databases and based on it HBase is found to be best suited for the read based operations [12].

Considering the HBase table schema design, there are basically two approaches. First one is the "Tall Narrow" design which is feasible for the data sets where table has large number of rows but less number of columns. The second approach is "Flat Wide" design and can be adopted where the table has large number of columns but less number of rows [11].

Another approach, while designing schema architecture for the Relational database, is based on a log file to store operations, configurations, modifications and query processes. The log files are usually used to monitor the database and track database wide operations. Additionally, the logged operations can be identified by the queries accessing the database. Therefore, by analyzing the log file, we can track the tables which are frequently accessed during the query processing [23].

The HBase hush database

We studied one of the database called HUSH, which provides both SQL and NoSQL version schemas [24]. HUSH is used to set up a very specific set of tables, which contains an equal and specific set of data. This functionality assists in order to easily understand what is given for transformation and how specific operation is being performed. Therefore, evaluating Hush database facilitates our study to clearly understand the schema models of both SQL and NoSQL databases.

General guidelines

RAD (Rapid application development): It is a market and data requirement for the applications targeting Big Data model so that the transformation can be performed in a fast and efficient manner [11].

Scalability: Is the user-demand to meet with the constantly growing throughput of the data and to access it [11].

Consistent Performance: Low response time is the key of success when handling with large amount of data and is vital for its growth [11].

Operational reliability: Built-in High Availability [11].

Problem statement and motivation

Very few examples of database schema conversion between relational and non-relational databases currently exist in literature and there are even fewer examples of an automated conversion tool or even anything that gets you started [20]. The topic of big data is very vast and much research has already been made, but there are avenues that need to be explored in conjunction with current ongoing research.

This research is unique in two different ways:

It has targeted a specific RDBMS platforms transformation, i.e. Microsoft SQL Server

It has targeted a real-time data of a specific business domain that operates automated machines. These machines generate very huge amount of data during the production life cycle.

After performing literature survey we found that there is no such study and experiment conducted on specific industry such as, a textile industry. The data is collected and later being analysed for business related decisions by performing data analytics. The size of the data can be measured by the actual machine-units running on 24 hours' basis (unless any walkout occurs due to any reason like machine faults or electricity outrage).

Database growth

Table 1 explains the growth of database tables during a single transaction per day and listed only huge tables. When we calculate this growth over the period of time it produces very large amount of Tera-byte data. This data requires to be analyzed for read operations in order to get in-depth knowledge and data analysis. Our practical use-case explains that the data requires only read operations once the transaction is being processed from RDBMS system. Therefore, based on the study, we choose HBase as our Non-Relational model and mapped the large tables SQL schema to NoSQL schema using our proposed approach.

Proposed approach

Migration can be insignificant given that experts tends to follow principles. The aim of this study is to simplify and standardize the transformation process. In further lines, we have demonstrated the use of rules and perform some experiments that can be evaluated by the NoSQL community for validation.

The case study and experiments performed on a data set is being generated from 04 out of 400 automated machines that generates approximately 0.1 Million units produced per day. Each unit can further be categorized on an average of 100 sub units of the main unit. This sums up on each day almost 0.1 X 100 records are being generated from one production environment. Such a huge data must be scaled by utilizing the Big Data power of parallelism and scalability. There is a major breakthrough in generating decision based alerts on it.

Table 1: Dataset: Database Tables.

S.No.	Table Name	Row Count	Size (MB)
1	pp.production	1,728,460	81
2	pp.packing	1,276,537	176
3	pp.pieces	1,037,953	160
4	pp.auditlog	780,041	50

Citation: Ahmad M, Siddiqui Z (2016) Leveraging Data Analytics by Transforming Relational Database Schema in to Big Data. Trends Comput Sci Inf Technol 1(1): 012-017. DOI: http://dx.doi.org/10.17352/tcsit.000002

There are two primary objectives of standardized schema transformation:

- Aim to minimize the work of database administrators, and
- Set transformation standards in order to achieve seamless transition from SQL-Server tables to Hbase tables.

SQL Server Table Schema and Metadata

We start by standardizing the SQL Server table metadata in order to achieve the transformation process. The table metadata consists of several fields namely (column name, data type, default value, computed value, description of column)

- Our model is based on the metadata information stored in the field 'Description'.
- In any SQL Server table, we set the 'Description' column to hold the name which needs to be mapped to the Hbase column.
- Our schema mapping program looks for the fields with metadata column information and pick only those fields for the mapping.
- Figure 2 shows how the transformation takes place by setting the metadata.

HBase Table Schema

From a logical perspective, HBase model is comprises of four major components [25].

- A Table is a collection of column families
- Colum Family is a logical and physical grouping of columns and refers to as basic storage unit.
- Columns as compared to traditional RDBMS model and are very different in HBase. Columns exist only when data is inserted and can have multiple versions based on the timestamp auto generated at the time of data insertion. Therefore, for a single row key there can be multiple versions based on the timestamp and only last available version is queried until or unless another version is queried explicitly. Every row in a table can have different set of columns identified by its row-key.
- Row-Key is an implicit primary key to uniquely identify the record. It is an ordered key and therefore provide efficient query processing (Figure 3).

Relationship type: One-to-One

For the conversion of one-to-one relationship, there are two possible conceivable approaches to translate it into HBase table.

- Single HTable

The first would be to combine the two tables together into one HTable. The subsequent HTable could have two originations.1) Single Column Family.2) Multiple or Two Columns families.

- Two or Multiple HTables

The second conceivable configuration is to make two HTables where each HTable contains one column family and every column family contains all SQL columns. At last, embed the row key of each HTable into both HTables.

Relationship type: One-to-Many

In the transformation of a one-to-many relationship, there are two conceivable approaches to transform it into HBase:

- Single HTable

The first would be to combine the two tables together into one HTable. The subsequent HTable could have two originations.1) Single Column Family.2) Multiple or Two Columns families.

- Two or Multiple HTables

Each SQL table is added to two separate HTables with a single column family. The primary HTable contains the actual relationship and a second column family is included that holds the referenced column Keys from the second HTable.

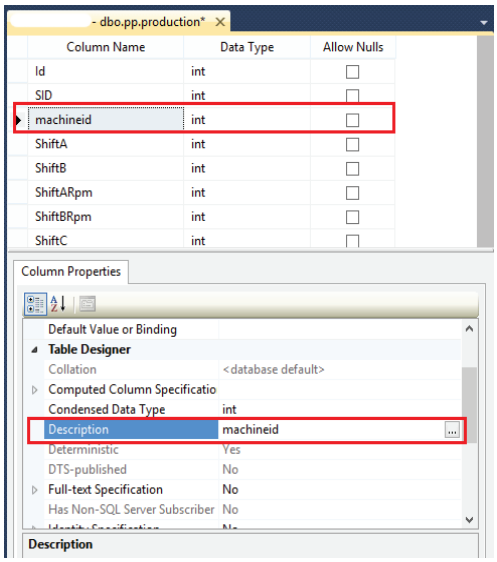

Figure 2: SQL Server column metadata settings.

```
Table{
"Row-Key1" : {   //Row key
  "ColumnFamily1" : {  //Column Family
    "Column-Key1":  { value1 }//Column Quallifier
    "Column-Key2":  { value2 }
     ...
     }
  }
}
```

Figure 3: SQL Server column metadata settings.

Citation: Ahmad M, Siddiqui Z (2016) Leveraging Data Analytics by Transforming Relational Database Schema in to Big Data. Trends Comput Sci Inf Technol 1(1): 012-017. DOI: http://dx.doi.org/10.17352/tcsit.000002

Experiment

An experiment has been carried out on a dataset listed in Table 2 and the process is as follows:

- Step I: SQL Server table columns were marked by adding metadata information 'Description'. In a single table only those columns which need to be imported and mapped to HBase table are defined in 'Description' metadata.
- Step II: Our script checks for the metadata information based on a SQL query run on the information metadata of a SQL Server database. That script picks only those columns and generates a list.
- Step III: Explained in Table 4, based on the metadata information, we generate the CREATE TABLE HBASE scripts for those SQL server table. This helps developers and administrators to have pre-generated scripts available for them to run on the Hadoop HBase environment.

Workloads

Table 3 shows the workloads on which we proposed to perform the performance testing and as a future study we can compare it with RDBMS query time.

Figure 4 depicts a virtual machine setup for the Hadoop environment to perform tests.

Design Considerations

Provided, the RDBMS sample table of [pp.production] from Table 2, the Table 3 shows the data generated on sample basis from four machines. This RDBMS table model, when transforming to HBase model and has been done by using our proposed approach.

HBase Logical Schema

We take the Row#1 and Row#5 for the same machine that generates different outputs during the production scheduling based on the shifts.

Table 4 is a mapping of table IV structure from SQL Server to HBase tables. We explain here how the single and multiple column families can be used and how the timestamp column works (Figure 5).

Conclusion and Future Work

In this research we studied the problem of transforming SQL to NoSQL providing a general purpose solution that can be further utilized in a practical scenario. Our initial results show that this approach has a great potential in order to transform large volume of data from SQL Server to Big Data based on a schema transformation which is easier while using our approach.

Table 2: Work Loads

Workload A	Loads data from RDBMS to NoSQL
Workload B	100% read operation
Workload C	Short Ranges – Intervals of data

Table 3: Dataset: Table pp. production (attributes defined)

Row#	Machine ID	Shift Production	Speed RPM	Timestamp
1	10	1,650	850	20160324
2	24	1,354	850	20160324
3	105	985	650	20160324
4	240	835	650	20160324
1	24	1,430	850	20160325

Table 4: apping of RDBMS table to Hbase table

Row Key	Value (Column Family, Column, Version and Cell)
1	info: {'machinemake': 'Macpro'} prod: {'machineid': '10', 'Macpro': '1650'@ts=' 20160324' 'Macpro': '1430'@ts=' 20160325'}
2	info: {'machinemake': 'Jet'} prod: {'machineid': '24', 'Macpro': '1354'}

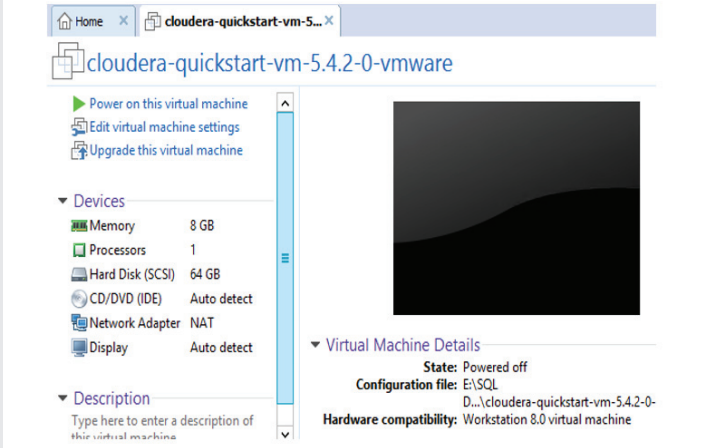

Figure 4: SQL Server column metadata settings.

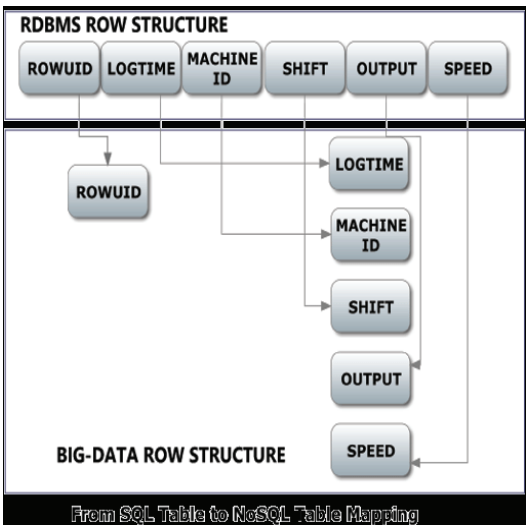

Figure 5: SQL Server column metadata settings.

Citation: Ahmad M, Siddiqui Z (2016) Leveraging Data Analytics by Transforming Relational Database Schema in to Big Data. Trends Comput Sci Inf Technol 1(1): 012-017. DOI: http://dx.doi.org/10.17352/tcsit.000002

We are planning to extend this study by analyzing the performance improvement which we gain by transforming normal data into Big Data structure and produce highly scalable analytical solution, particularly targeting a process manufacturing sector production data that is of very high volume.

As a future work, this study can be extended to produce a commercial solution that can be attached with the relational database engine to transform the data into Big Data in a very efficient manner. Industries in all over the world that manufactures products on automated machines can take benefit of this solution.

References

1. Sharma V, Dave M (2012) SQL and NoSQL Databases. International Journal of Advanced Research in Computer Science and Software Engineering 2: Link: https://goo.gl/GIu78h
2. Liu C, Fu Z, Yang Z, Xiu J (2015) General Research on Database Migration from RDBMS to Hbase. In 2015 International Symposium on Computers & Informatics. French: Atlantis Press 124-237. Link: https://goo.gl/Q071cb
3. Siddiqui Z, Abdullah AH, Khan MK, Alghamdi AS (2014) Smart environment as a service: three factor cloud based user authentication for telecare medical information system. Journal of Medical Systems 38: 1-14. Link: https://goo.gl/iJTMVT
4. Siddiqui Z, Abdullah AH, Khan MK, Alghathbar K (2011) Analysis of enterprise service buses based on information security, interoperability and high-availability using Analytical Hierarchy Process (AHP) method. International Journal of Physical Sciences 6: 35-42. Link: https://goo.gl/RsvR6g
5. Siddiqui Z, Abdullah AH, Khan MK (2011) Qualified analysis b/w ESB(s) using Analytical Hierarchy Process (AHP) method. Second international conference on intelligent systems. modelling and simulation. 100-104. Link: https://goo.gl/z9aXTw
6. Siddiqui Z, Abdullah AH, Khan MK, Alghamdi AS (2016) Cryptanalysis and improvement of 'a secure authentication scheme for telecare medical information system' with nonce verification. Peer-to-Peer Networking and Application. Springer 9: 841-853. https://goo.gl/x9oyEI
7. Alghamdi AS, Siddiqui A, Quadri SSA (2010) A common information exchange model for multiple C4i architectures. 12th international conference on computer modelling and simulation, IEEE 538-542. Link: https://goo.gl/ljWXbu
8. Siddiqui Z, Alghamdi AS, Khan MK (2011) Node level information security in common information exchange model (CIEM). International Science Journal 21: 221-223. Link: https://goo.gl/35GLdr
9. Siddiqui Z, Alghamdi AS (2014) SOA based C4I common-view interoperability model. International Science Journal 26: 175-180. Link: https://goo.gl/Lks76F
10. Siddiqui Z, Alghamdi AS (2012) A universal view SOA interoperability framework for multiple C4I applications. International Science Journal, 26: 97-100. Link: https://goo.gl/P9YHjh
11. Lee CH, Zheng YL (2015) Automatic SQL-to-NoSQL schema transformation over the MySQL and HBase databases. In Consumer Electronics-Taiwan (ICCE-TW). 2015 IEEE International Conference on 426-427. Link: https://goo.gl/2YUVhg
12. Abramova V, Bernardino J, Furtado P (2014) Experimental evaluation of NoSQL databases. International Journal of Database Management Systems 6: Link: https://goo.gl/gjTTlX
13. Jung MG, Youn SA, Bae J, Choi YL (2015) A Study on Data Input and Output Performance Comparison of MongoDB and PostgreSQL in the Big Data Environment. In 2015 8th International Conference on Database Theory and Application (DTA) 14-17. Link: https://goo.gl/A6hcri
14. Sharma S, Tim US, Gadia S, Wong J, Shandilya R, et al. (2015) Classification and comparison of NoSQL big data models. International Journal of Big Data Intelligence 2: 201-221. Link: https://goo.gl/KWFkrx
15. Borthakur D (2007) The hadoop distributed file system: Architecture and design. Hadoop Project Website 11: 21. Link: https://goo.gl/G9e3RG
16. Binani S, Gutti A, Upadhyay S (2016) SQL vs. NoSQL vs. NewSQL-A Comparative Study. database 6. Link: https://goo.gl/gSyvWz
17. Chang F, Dean J, Ghemawat S, Hsieh WC, Wallach DA, et al. (2008) Bigtable: A distributed storage system for structured data. ACM Transactions on Computer Systems (TOCS) 26: 4. Link: https://goo.gl/NiyyWN
18. Apache HBaseProject. Link: https://goo.gl/mEruf2
19. Brewer EA (2000) Towards robust distributed systems. In PODC 7. Link: https://goo.gl/2L4Itf
20. Lee CH, Zheng YL (2015) SQL-to-NoSQL Schema De-normalization and Migration: A Study on Content Management Systems. In Systems, Man, and Cybernetics (SMC). 2015 IEEE International Conference on 2022-2026. Link: https://goo.gl/j1Uf67
21. Gajendran SK (2012) A survey on nosql databases. University of Illinois. Link: https://goo.gl/7flWs4
22. Nitnaware C, Khan A (2015) A multidimensional data storage model for location based application on Hbase. In Innovations in Information, Embedded and Communication Systems (ICIIECS). 2015 International Conference on 1-5. Link: https://goo.gl/xwQUyE
23. Hsu JC, Hsu CH, Chen SC, Chung YC (2014) Correlation Aware Technique for SQL to NoSQL Transformation. In Ubi-Media Computing and Workshops (UMEDIA). 2014 7th International Conference on 43-46. Link: https://goo.gl/9RzB2C
24. George L (2011) HBase: the definitive guide. O'Reilly Media, Inc.. Link: https://goo.gl/OSAM4H
25. Vora MN (2011) Hadoop-HBase for large-scale data. In Computer science and network technology (ICCSNT). 2011 international conference on 1: 601-605. Link: https://goo.gl/VCWwgz
26. Wang Y, Xu Y, Liu Y, Chen J, Hu S (2015) QMapper for Smart Grid: Migrating SQL-based Application to Hive. In Proceedings of the 2015 ACM SIGMOD International Conference on Management of Data 647-658. Link: https://goo.gl/bhLCMg
27. Liao YT, Zhou J, Lu CH, Chen SC, Hsu CH, et al. (2016) Data adapter for querying and transformation between SQL and NoSQL database. Future Generation Computer Systems 65: 111-121. Link: https://goo.gl/4ko4sa

Citation: Ahmad M, Siddiqui Z (2016) Leveraging Data Analytics by Transforming Relational Database Schema in to Big Data. Trends Comput Sci Inf Technol 1(1): 012-017. DOI: http://dx.doi.org/10.17352/tcsit.000002