AN EFFICIENT RELATIONAL TO COLUMN ORIENTED DATABASE SCHEMA TRANSFORMATION TECHNIQUE

NORWINI BINTI ZAIDI

FSKTM 2020 1

AN EFFICIENT RELATIONAL TO COLUMN ORIENTED DATABASE SCHEMA TRANSFORMATION TECHNIQUE

By

NORWINI BINTI ZAIDI

Thesis Submitted to the School of Graduate Studies, Universiti Putra Malaysia, in Fulfilment of the Requirement for the Degree of Master of Science

September 2019

Abstract of thesis presented to the Senate of Universiti Putra Malaysia in fulfilment of the requirement for the degree of Master of Science

AN EFFICIENT RELATIONAL TO COLUMN ORIENTED DATABASE SCHEMA TRANSFORMATION TECHNIQUE

By

NORWINI BINTI ZAIDI

September 2019

Chair: Iskandar Ishak, PhD
Faculty: Computer Science and Information Technology

NoSQL database is introduced to overcome the high demand of managing database management systems in addition to the need for managing huge amount of data in unstructured format. Thus, data migration has become an important process in database management to migrate relational database to NoSQL database due to the limitations in managing relational database. Schema transformation is an important process in data migration and there are various techniques that have been proposed to improve schema transformation and data migration from the relational database to the NoSQL database. The most common technique of schema transformation to NoSQL database is denormalization. However, schema transformation using denormalization suffers in terms of unnecessary data duplication in the NoSQL database that increases storage size. Furthermore, NoSQL database also has its limitations in terms of table joining and unable to perform queries on multiple tables. Schema transformation techniques using nested table merging describes only two related tables to merge. This inefficient schema transformation techniques lead to querying to be done on multiple tables and cause high query processing time.

This research proposed a schema transformation technique for migrating data from relational database to column oriented database. The schema transformation technique has three main steps which are denormalization with read pattern, nested and multiple nested table merging, and rowkey design to reduce data redundancy and storage size to produce efficient query performance. In this technique, the read pattern identifies the access key of the query. The nested and multiple nested table merging techniques combined the tables that have the same access key to be in a nested form. The nested and multiple nested table merging on column oriented database leads the query to be performed on a single table to retrieve the data and thus improved query performance. Meanwhile, the rowkey design helps to determine the rowkey based on access keys that are identified in the read pattern technique. The experimental results showed that the proposed schema transformation technique managed to reduce data redundancy by eight

column families thus reducing the storage size by 13.83% and improve the query performance time by 29.28% for DELL DVD dataset. While by using the *Employees* dataset, the proposed technique managed to reduce data redundancy by five column families thus reducing the storage size by 15.67% and improve the query performance time by 29.13%.

Abstrak tesis yang dikemukakan kepada Senat Universiti Putra Malaysia sebagai memenuhi keperluan untuk ijazah Master Sains

TEKNIK TRANSFORMASI SKEMA PANGKALAN DATA HUBUNGAN KEPADA PANGKALAN DATA BERORIENTASIKAN LAJUR YANG EFISEN

Oleh

NORWINI BINTI ZAIDI

September 2019

Pengerusi: Iskandar Bin Ishak, PhD
Fakulti: Sains Komputer dan Teknologi Maklumat

Pangkalan data NoSQL diperkenalkan untuk mengatasi permintaan yang tinggi bagi menguruskan sistem pengurusan pangkalan data, di samping keperluan untuk menguruskan sejumlah besar data dalam format yang tidak berstruktur. Oleh itu, pemindahan data menjadi proses yang sangat penting dalam pengurusan pangkalan data untuk pemindahan pangkalan data hubungan kepada pangkalan data NoSQL berikutan terdapat batasan dalam pengurusan pangkalan data hubungan. Transformasi skema adalah salah satu proses yang penting dalam pemindahan data dan terdapat pelbagai teknik yang telah dicadangkan untuk memperbaiki transformasi skema dan pemindahan data dari pangkalan data hubungan ke pangkalan data NoSQL. Teknik yang sering digunakan untuk transformasi skema ke pangkalan data NoSQL adalah denormalisasi. Walau bagaimanapun, transformasi skema yang menggunakan denormalisasi menyebabkan penduaan data yang tidak diperlukan di pangkalan data NoSQL yang akan meningkatkan saiz storan. Selain itu, pangkalan data NoSQL juga mempunyai batasan untuk menghubungkan jadual dan tidak dapat melaksanakan pertanyaan ke berbilang jadual. Teknik transformasi skema menggunakan penggabungan jadual bersarang melibatkan hanya dua jadual yang berkaitan akan digabungkan. Teknik transformasi skema yang tidak efisien ini membawa kepada pertanyaan yang akan dilakukan kepada berbilang jadual dan menyebabkan masa pemprosesan pertanyaan yang tinggi.

Kajian ini mencadangkan teknik transformasi skema untuk pemindahan data dari pangkalan data hubungan ke pangkalan data berorientasikan lajur. Teknik transfomasi skema mempunyai tiga langkah utama iaitu denormalisasi dengan corak baca, penggabungan jadual bersarang dan berbilang sarang dan reka bentuk kunci baris untuk mengurangkan lebihan data dan saiz storan bagi menghasilkan prestasi pertanyaan yang efisien. Dalam teknik ini, corak baca digunakan untuk mengenalpasti kunci akses pertanyaan. Teknik penggabungan jadual bersarang dan berbilang sarang menggabungkan jadual yang mempunyai kunci akses yang sama berada dalam bentuk bersarang. Penggabungan jadual bersarang dan berbilang sarang di pangkalan data

berorientasikan lajur membawa pertanyaan kepada satu jadual untuk mendapatkan data dan meningkatkan prestasi pertanyaan. Sementara itu, reka bentuk kunci baris membantu untuk menentukan kunci baris berdasarkan kunci akses yang telah dikenalpasti di dalam teknik corak baca. Keputusan eksperimen menunjukkan bahawa teknik transformasi skema yang dicadangkan berupaya mengurangkan lebihan data sebanyak lapan jadual serta mengurangkan saiz storan sebanyak 13.83% dan meningkatkan prestasi pertanyaan sebanyak 29.28% bagi dataset DELL DVD. Manakala dengan menggunakan dataset *Employees,* teknik yang dicadangkan berupaya mengurangkan lebihan data sebanyak lima jadual serta mengurangkan saiz storan sebanyak 15.67% dan meningkatkan prestasi pertanyaan sebanyak 29.13%.

ACKNOWLEDGEMENTS

Alhamdulillah, praise to Allah s.w.t who has granted me the strength, courage and patience to complete my master's research work. I would like to express my gratitude and appreciation to my supervisor Dr. Iskandar Ishak for his support in order to complete my Master study. Thank you for always having time for me and providing me his technical expertise as well as his encouragement to build my positive vibes to complete my research work and maintain my progress in track. His guidance helped me in all time of research works and writing of this thesis.

I also would like to express my appreciation to my supervisory committee, Assoc. Prof. Dr. Lilly Suriani Affendey, and Assoc. Prof. Dr Fatimah Sidi for their valuable feedbacks, comments and suggestions to my research works. I am grateful for the opportunity to have an experience in conducting research with them.

I am indebted to the management of Islamic Science University of Malaysia (USIM) that has granted me the sponsorship and the opportunity to pursue my Master program. Thanks for this amazing opportunity.

Last but not least, I would like to thank my husband and my family for supporting me spiritually throughout completing my research works. Thank you so much.

This thesis was submitted to the Senate of Universiti Putra Malaysia and has been accepted as fulfilment of the requirement for the degree of Master of Science. The members of the Supervisory Committee were as follows:

Iskandar Ishak, PhD
Senior Lecturer
Faculty of Computer Science and Information Technology
Universiti Putra Malaysia
(Chairman)

Fatimah Sidi, PhD
Associate Professor
Faculty of Computer Science and Information Technology
Universiti Putra Malaysia
(Member)

Lilly Suriani Affendey, PhD
Associate Professor
Faculty of Computer Science and Information Technology
Universiti Putra Malaysia
(Member)

ZALILAH MOHD SHARIFF, PhD
Professor and Dean
School of Graduate Studies
Universiti Putra Malaysia

Date: 12 March 2020

Declaration by graduate student

I hereby confirm that:

- this thesis is my original work;
- quotations, illustrations and citations have been duly referenced;
- this thesis has not been submitted previously or concurrently for any other degree at any other institutions;
- intellectual property from the thesis and copyright of thesis are fully-owned by Universiti Putra Malaysia, as according to the Universiti Putra Malaysia (Research) Rules 2012;
- written permission must be obtained from supervisor and the office of Deputy Vice-Chancellor (Research and Innovation) before thesis is published (in the form of written, printed or in electronic form) including books, journals, modules, proceedings, popular writings, seminar papers, manuscripts, posters, reports, lecture notes, learning modules or any other materials as stated in the Universiti Putra Malaysia (Research) Rules 2012;
- there is no plagiarism or data falsification/fabrication in the thesis, and scholarly integrity is upheld as according to the Universiti Putra Malaysia (Graduate Studies) Rules 2003 (Revision 2012-2013) and the Universiti Putra Malaysia (Research) Rules 2012. The thesis has undergone plagiarism detection software.

Signature: _____ Date: _____

Name and Matric No.: Norwini Binti Zaidi, GS47510

Declaration by Members of Supervisory Committee

This is to confirm that:

- the research conducted and the writing of this thesis was under our supervision;
- supervision responsibilities as stated in the Universiti Putra Malaysia (Graduate Studies) Rules 2003 (Revision 2012-2013) are adhered to.

Signature: _____
Name of Chairman of Supervisory Committee: _____

Signature: _____
Name of Member of Supervisory Committee: _____

Signature: _____
Name of Member of Supervisory Committee: _____

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF ABBREVIATIONS

ACID	Atomicity, Consistency, Isolation and Durability
API	Application Program Interface
BASE	Basically Available, Soft State and Eventually Consistent
CAP	Consistency, Availability or Partition Tolerance
CF	Column Family
DDI	Denormalization, Duplication, Intelligent keys
ERD	Entity Relationship Diagram
ETL	Extract, Transform and Load
GB	Gigabyte
NoSQL	Not Only SQL / Non-relational database
PK	Primary Key
Q	Query
QH	Query in HBase
QP	Query Path
RDBMS	Relational Database Management System
REST	Representational State Transfer
SQL	Structured Query Language

CHAPTER 1

INTRODUCTION

1.1 Background

For many years, relational database has become a popular database for managing data (Lourenco et al., 2015; Rocha et al., 2015). Relational database has been the database solution for most of data management systems (Monika & Ashok, 2017; Lourenco et al., 2015; Rith et al., 2014). However, with the fast increasing growth of data being created and recorded recently, relational database seems to be lacking in terms of storing large volume of data and managing different types of database schema or flexible schema (Ntarmos et al., 2014; Zhao et al., 2014b). In order to overcome the relational database limitations, one of the approaches is by using NoSQL database to support data management systems that possess big data processing capability. Compared to the relational database, NoSQL supports data which are unstructured in format.

Big data is recognized based on its main characteristics which are volume, variety, and velocity (Hashem et al., 2015; Sharma et al., 2014; Philip & Zhang, 2014; Katal et al., 2013). The volume category refers to the large size of data set, variety describes the multiple structures of data which unstructured, semi-structured, and structured data and velocity is where the data are continuously expanded and processed rapidly (Hashem et al., 2015; Sharma et al., 2014; Philip & Zhang, 2014; Katal et al., 2013). In order to meet the big data demand, organizations migrate their data from conventional relational database to NoSQL database (Ghotiya et al., 2017; Monika & Ashok, 2017; Ahmed & Gulmeher, 2015). The NoSQL database has flexible schema that fundamentally different from relational database schema. The NoSQL database does not require fix number of tables or columns and is suitable for storing unstructured data and can easily be expanded.

However, performing schema transformation and migration from existing relational database to NoSQL databases pose a lot of challenges. One of the challenges is the differences of database schema between relational database and NoSQL database (Schram & Anderson, 2012). In order to solve the different schemas issue, analysis needs to be done for the schema transformation of relational database to the NoSQL database to produce efficient NoSQL database schema thus reducing the storage size for the transformed database and improve query performance. Based on the literatures, there are no standard or fixed guidelines on how to transform the database schema from relational database and perform data migration to NoSQL database (Ouanouki et al., 2017; Goyal et al., 2016; Jia et al., 2016; Gomez et al., 2015; Lee & Zheng, 2015a; Gomez et al., 2014). Commonly, database administrator used their experiences to manage the transformation process and the migration process to the NoSQL database thus lead to inefficient database schema and suboptimal query performance (Ouanouki et al., 2017; Gomez et al., 2015).

Many researchers have proposed a number of schema transformation techniques for transforming and migrating relational database to NoSQL database. The approaches include denormalization, duplication, intelligent keys (DDI), schema conversion, conversion rules, access pattern, nested, and multiple nested. However, denormalization is the most common techniques for schema transformation to NoSQL database (Goyal et al., 2016).

Denormalization is a technique to merge related data from relational database to NoSQL database (Goyal et al., 2016; Kuderu & Kumari, 2016; Lee & Zheng, 2015b). The data are duplicated from the related tables based on the relational database relationships into a single table in NoSQL database (Goyal et al., 2016; Kuderu & Kumari, 2016; Gomez et al., 2015). The unnecessary duplication of tables using denormalization technique may not be useful if there is no data accessing for read and write operation in the merged tables and may increase the storage size. To determine which tables are to be merged, the denormalization technique requires data access pattern. The data access pattern identified the read and the write patterns that need to be analysed for NoSQL database schema design, and is used to reduce unrelated table merging (Ouanouki et al., 2017; Gomez et al., 2016; Serrano et al., 2015; HBase, 2014; Li, 2010). The data that have same access pattern are merged into a single table in the NoSQL database.

After the schema transformation is performed, the relationships between the tables in the NoSQL database are removed. The related tables with join condition from the relational database are merged into a single table in the NoSQL database. This is why join operations are not supported in the NoSQL database (Sharma et al., 2016; Zhao et al., 2014b; Arora & Aggarwal, 2013). However, the merging process into a single table using nested table merging from relational database may produce additional tables in the NoSQL database thus it may lead to multiple tables queries after the data migration is performed. In order to solve this issue, a method based on multiple nested table merging is introduced for merging more than two relations to NoSQL database. Through the multiple nested table merging technique, the access key of the table needs to be analysed to perform join key relationships in NoSQL data model to ensure high query performance and data accessing (Zhao et al., 2014a).

Each of the NoSQL database table needs to define a key for data retrieval based on its unique identification. However, incorrect selection of keys can cause poor query performance. Therefore, the selection of rowkey is based on the data access pattern on the tables in the database. The selection of rowkey in the NoSQL database should be concatenation of several primary keys or foreign keys in the existing relational database (Kuderu & Kumari, 2016; Lee & Zheng, 2015b; Serrano et al., 2015).

An approach with the combination of denormalization with read pattern, nested and multiple nested table merging, and rowkey design to denormalize the relational database to form an efficient schema transformation technique to improve schema transformation to column oriented database is proposed in this study.

1.2 Problem Statement

Transformation using denormalization technique from the relational database may lead to the merging and duplication of the data from the SQL tables into the NoSQL table. As a result, the duplication of data in the NoSQL table increases the storage size thus creating wastage of space. Previous works related to the data transformation using denormalization, have proposed schema transformation from relational into NoSQL database (Kuderu & Kumari, 2016; Karnitis & Arnicans, 2015; Lee & Zheng, 2015b; Serrano et al., 2015; Li, 2010). The previous works done by Yoo et al. (2018), Lee & Zheng (2015b), Serrano et al. (2015) managed to optimize the read performances and time yield but at the same time produced unnecessary duplication of data.

In the relational database, the normalization technique produces tables with relationship, namely: one-to-one, one-to-many or many-to-many relationships. Denormalization technique duplicates all the data from relational table to eliminate the join relationship and transform it into a single table in the NoSQL database (Yoo et al., 2018; Goyal et al., 2016; Kuderu & Kumari, 2016; Lee & Zheng, 2015b). In denormalization technique, a key is needed to be identified for unique identification for the denormalized table. However, poor selection of key from the denormalization technique can cause full table scan during data retrieval that increases query time unnecessarily thus decreasing query performances (Ouanouki et al., 2017; Zhao et al., 2014a). Typically, the primary key in relational database is unique in the table but in the NoSQL table, the selection of rowkey is based on the data access pattern of the tables in the database (Ouanouki et al., 2017).

The denormalization technique that did not consider data access pattern may duplicate and merges unnecessary tables that produces data redundancy and can increase unnecessary storage size in the data transformation process (Ouanouki et al., 2017; Gomez et al., 2016; Serrano et al., 2015; HBase, 2014; Zhao et al., 2014b; Li, 2010). The data access pattern normally refers to the read and write pattern between the joined tables from the application system (Khazaei et al., 2016; Zhu et al., 2016; "RDBMS to MongoDB Migration Guide," 2016; Klein et al., 2015; Serrano et al., 2015; Li, 2010). Based on Serrano et al. (2015), the relational tables with no access pattern do not have to be merged. The access pattern needs to be determined before implementing the schema transformation into column oriented database to improve the NoSQL database schema design.

NoSQL database also has its limitation in table join and difficulties in query on multiple tables. NoSQL database supports the column family with the column qualifier in the same table. This is to ensure high query performance in the transformed databases. Zhao et al. (2014b) and Li (2010) proposed a nested table merging technique that describes two table references relationship. However, the nested table merging technique leads to producing additional NoSQL tables that may involve querying multiple tables to produce query result. The production of data redundancy increases unnecessary storage size and query processing time. Therefore, multiple nested table merging is an important technique to be performed for joined relationships between three or more tables to improve query performance and reduce storage size by accessing a single table (Zhao et

al., 2014a). The nested and multiple nested table merging are explained in detail in Section 2.8.

To sum it up, there are two problems considered in this study for schema transformation from relational database to NoSQL database. The first one is the unnecessary data duplication using denormalization technique may not be useful that may lead to the creation of data redundancy and may increase the storage size. Another issue in schema transformation to NoSQL database is the join relationship is not supported in NoSQL database for querying purposes that produces inefficient query performance in terms of query processing time.

1.3 Research Objective

The main objective of this study is to propose a schema transformation technique from relational database to NoSQL column oriented database that utilizes the denormalization technique with read pattern, nested and multiple nested table merging, and rowkey design. The proposed transformation technique aims to reduce storage size by reducing data redundancy and to improve query performance time by merging three or more related tables from the relational database to produce a single HBase table. The selection of rowkey for unique identification based on the read pattern of the tables in the database is proposed in order to avoid full table scan in data retrieval that increases query time unnecessarily.

1.4 Research Scope

The scope of this study focuses on schema transformation technique from relational database to NoSQL column oriented database. In this study, the HBase database is used because of the column oriented database characteristic is suitable for applications which involve heavy writes and searching processing for faster transaction and makes it ideal for schema transformation (Ouanouki et al., 2017; Vohra, 2016; Serrano et al., 2015; Zhao et al., 2014a). The HBase database stores large quantities of data and uses a rowkey to access the data for querying and also suitable for audit logging systems, tracking user actions, real time analytics, monitoring systems, and message-centered systems (Vohra, 2016). The HBase database has been deployed in companies like Twitter, Facebook, Yahoo, and Adobe (Pore & Pawar, 2015; DeRoos et al., 2014; Zhao et al., 2014a). Other types of NoSQL database such as key value database, document database, and graph database are beyond the scope of this study. These databases have different unstructured format and suitable for transaction such as online shopping website for key value database; image, and video for document database; and also spatial data for graph database (Pore & Pawar, 2015; Abdullah & Zhuge, 2015; Zhao et al., 2014b).

Besides that, this study concentrates on read pattern only in denormalization technique using data access pattern. The data access pattern is used to identify how the data are written and read that are retrieved from query logs (Ouanouki et al., 2017; Ammar, 2016;

Khazaei et al., 2016; HBase, 2014). In this study, the read pattern is focuses as it is the only access pattern applied in the denormalization technique by Ouanouki et al. (2017), Serrano et al. (2015) and Li (2010). The tables that have same read pattern are merged into a single table in the NoSQL databases (Ouanouki et al., 2017; Serrano et al., 2015; Li, 2010). The write pattern is not considered in this study since the read pattern is the only access pattern used by Ouanouki et al. (2017), Serrano et al. (2015) and Li (2010) and previous works (Kuderu & Kumari, 2016; Lee & Zheng, 2015a; Lee & Zheng, 2015b; Mpinda et al., 2015; Schreiner et al., 2015; Zhao et al., 2014) did not consider data access pattern in their research works.

1.5 Thesis Structure

This thesis consists of six chapters. The first chapter begins with the background of the study, problem statement, research objective, and scope of the study. Chapter 2 reviews the related works and discusses the background of data transformation process to produce column oriented schema design. Chapter 3 describes the research methodology used in this study. Chapter 4 presents the proposed schema transformation technique from relational database to column oriented database. Chapter 5 presents the performance results of the proposed schema transformation technique. Finally, chapter 6 concludes the achievement of this study and the recommendation of future works to enhance this study.

REFERENCES

Abdullah, A., & Zhuge, Q. (2015). From Relational Databases to NoSQL Databases : Performance Evaluation. *Research Journal of Applied Science, Engineering and Technology*, *11*(4), 434–439. http://doi.org/10.19026/rjaset.11.1799

Abramova, V., Bernardino, J., & Furtado, P. (2014). Which NoSQL Database? A Performance Overview. *Open Journal of Databases (OJDB)*, *1*(2), 17–24.

Ahmed, J., & Gulmeher, R. (2015). NOSQL Databases: New Trend of Databases, Emerging Reasons, Classification and Security Issues. *International Journal of Engineering Sciences & Research Technology*, *9655*(6), 1.

Ammar, A. Ben. (2016). Query Optimization Techniques In Graph Databases, *8*(4), 1–14. http://doi.org/10.5121/ijdms.2016.8401

Arora, R., & Aggarwal, R. R. (2013). An Algorithm for Transformation of Data from MySQL to NoSQL (MongoDB). *International Journal of Advanced Studies in Computer Science and Engineering (IJASCSE)*, *2*(1), 6–12. Retrieved from http://ijascse.org/volume-2-special-issue-1/An_algorithm.pdf

Badlani, P. (2016). NoSQL in Action-A New Pathway to Database. *International Journal of Science and Research (IJSR)*, *5*(6), 872–877. http://doi.org/doi.org/10.21275/v5i6.nov164282

Chopde, N. R., & Nichat, M. K. (2013). Landmark Based Shortest Path Detection by Using A* and Haversine Formula. *GH Raisoni College of Engineering and Management, Amravati, India*, *1*(2), 298–302. http://doi.org/10.1.1.300.5943

Copyrights Oracle and/or its affiliates. (2019). Employees Sample Database, 1–8.

Craig Brown. (2011). NoSql Tips and Tricks. *Http://blog.nosqltips.com*.

DeRoos, D., Zikopoulos, P. C., Melnyk, R. B., Brown, B., & Coss, R. (2014). *Hadoop for dummies*. http://doi.org/10.1002/1521-3773(20010316)40:6<9823::AID-ANIE9823>3.3.CO;2-C

El-Sappagh, S. H. A., Hendawi, A. M. A., & El Bastawissy, A. H. (2011). A proposed model for data warehouse ETL processes. *Journal of King Saud University - Computer and Information Sciences*, *23*(2), 91–104. http://doi.org/10.1016/j.jksuci.2011.05.005

Elmasri, R., & Navathe, S. B. (2011). *Fundamentals of Database Systems* (Sixth Edit). Pearson.

George, L. (2011). HBase: The Definitive Guide. O'Reilly. Retrieved from http://books.google.com/books?hl=en&lr=&id=nUhiQxUXVpMC&pgis=1

Ghotiya, S., Mandal, J., & Kandasamy, S. (2017). Migration from relational to NoSQL database. *IOP Conference Series: Materials Science and Engineering*, *263*(4). http://doi.org/10.1088/1757-899X/263/4/042055

Gomez, A., Ouanouki, R., April, A., & Abran, A. (2014). Building an Experiment Baseline in Migration Process from SQL Databases to Column Oriented No-SQL Databases. *Journal of Information Technology & Software Engineering*, *4*(2). http://doi.org/10.4172/2165-7866.1000137

Gomez, A., Ravanello, A., Ouanouki, R., April, A., & Abran, A. (2015). Experimental Validation as Support in the Migration from SQL Databases to NoSQL Databases. In *Cloud computing 2015: The Sixth International Conference on Cloud Computing, GRIDs and Virtualization* (pp. 147–153).

Gomez, P., Casallas, R., & Roncancio, C. (2016). Data schema does matter, even in NoSQL systems! *Proceedings - International Conference on Research*

Challenges in Information Science, 1–6. http://doi.org/10.1109/RCIS.2016.7549340

Goyal, A., Swaminathan, A., Pande, R., & Attar, V. (2016). Cross platform (RDBMS to NoSQL) database validation tool using bloom filter. *2016 International Conference on Recent Trends in Information Technology, ICRTIT 2016*. http://doi.org/10.1109/ICRTIT.2016.7569537

Hajoui, O., Dehbi, R., Talea, M., & Batouta, Z. I. (2015). An advanced comparative study of the most promising NoSQL and NewSQL databases with a multi-criteria analysis method. *Journal of Theoretical and Applied Information Technology*, *81*(3), 579–588.

Hashem, I. A. T., Yaqoob, I., Anuar, N. B., Mokhtar, S., Gani, A., & Ullah Khan, S. (2015). The rise of "big data" on cloud computing: Review and open research issues. *Information Systems*, *47*, 98–115. http://doi.org/10.1016/j.is.2014.07.006

HBase, A. (2014). *Apache HBase™ Reference Guide*. Retrieved from http://hbase.apache.org/book/book.html

Ho, L., Hsieh, M.-J., Wu, J.-J., & Liu, P. (2015). Data Partition Optimization for Column-Family NoSQL databases. *International Conference on Smart City/SocialCom/SustainCom Together with DataCom*, 668–675. http://doi.org/10.1109/SmartCity.2015.146

Jaffe, D., & Muirhead, T. (2005). The Open Source DVD Store Test Application, (August), 1–2. Retrieved from www.dell.com/downloads/global/power/ps3q05-20050217-Jaffe-OE.pdf

Jia, T., Zhao, X., Wang, Z., Gong, D., & Ding, G. (2016). Model transformation and data migration from relational database to MongoDB. *Proceedings - 2016 IEEE International Congress on Big Data, BigData Congress 2016*, 60–67. http://doi.org/10.1109/BigDataCongress.2016.16

Jose, J., Tomy, T. T., Karunakaran, V., Anjali Krishna, V., Varkey, A., & Nisha, C. A. (2016). Securing passwords from dictionary attack with character-tree. *Proceedings of the 2016 IEEE International Conference on Wireless Communications, Signal Processing and Networking, WiSPNET 2016*, 2301–2307. http://doi.org/10.1109/WiSPNET.2016.7566553

Jovanovic, V., Lazovic, V., & Minic, N. (2015). SQL QUERY EXECUTION TIME BETWEEN TWO SINGIDUNUM UNIVERSITY LOCATIONS. *International Scientific Conference of IT and Business Related Research*. http://doi.org/10.15308/Synthesis-2015

Kanade, A., Gopal, A., & Kanade, S. (2014). A Study of Normalization and Embedding in MongoDB. *IACC 2014 - Proceedings: 2014 IEEE International Advance Computing Conference*, 416–421. http://doi.org/10.1109/IAdCC.2014.6779360

Karnitis, G., & Arnicans, G. (2015). Migration of Relational Database to Document-Oriented Database: Structure Denormalization and Data Transformation. *Proceedings - 7th International Conference on Computational Intelligence, Communication Systems and Networks, CICSyN 2015*, 113–118. http://doi.org/10.1109/CICSyN.2015.30

Katal, A., Wazid, M., & Goudar, R. H. (2013). Big data: Issues, challenges, tools and Good practices. *2013 6th International Conference on Contemporary Computing, IC3 2013*, 404–409. http://doi.org/10.1109/IC3.2013.6612229

Khazaei, H., Fokaefs, M., Zareian, S., Beigi-Mohammadi, N., Ramprasad, B., Shtern, M., … Litoiu, M. (2016). How do I choose the right NoSQL solution?

A comprehensive theoretical and experimental survey. *Big Data and Information Analytics*, *1*(2/3), 185–216. http://doi.org/10.3934/bdia.2016004

Klein, J., Gorton, I., Ernst, N., Donohoe, P., Pham, K., & Matser, C. (2015). Application-Specific Evaluation of No SQL Databases. *Proceedings - 2015 IEEE International Congress on Big Data, BigData Congress 2015*, 526–534. http://doi.org/10.1109/BigDataCongress.2015.83

Kuderu, & Kumari. (2016). Relational Database to NoSQL Conversion by Schema Migration and Mapping. *International Journal of Computer Engineering In Research Trends*, *3*(9), 506–513. http://doi.org/doi.org/10.22362/ijcert/2016/v3/i9/48900

Lee, C. H., & Zheng, Y. L. (2015a). Automatic SQL-to-NoSQL schema transformation over the MySQL and HBase databases. *2015 IEEE International Conference on Consumer Electronics - Taiwan, ICCE-TW 2015*, 426–427. http://doi.org/10.1109/ICCE-TW.2015.7216979

Lee, C. H., & Zheng, Y. L. (2015b). SQL-To-NoSQL Schema Denormalization and Migration: A Study on Content Management Systems. *Proceedings - 2015 IEEE International Conference on Systems, Man, and Cybernetics, SMC 2015*, 2022–2026. http://doi.org/10.1109/SMC.2015.353

Li, C. (2010). Transforming Relational Database into HBase : A Case Study. *IEEE International Conference on Software Engineering and Service Sciences, ICSESS 2010*, 683–687. http://doi.org/10.1109/ICSESS.2010.5552465

Li, Y., & Manoharan, S. (2013). A performance comparison of SQL and NoSQL databases. *IEEE Pacific RIM Conference on Communications, Computers, and Signal Processing - Proceedings*, (August 2013), 15–19. http://doi.org/10.1109/PACRIM.2013.6625441

Lotfy, A. E., Saleh, A. I., El-Ghareeb, H. A., & Ali, H. A. (2016). A middle layer solution to support ACID properties for NoSQL databases. *Journal of King Saud University - Computer and Information Sciences*, *28*(1), 133–145. http://doi.org/http://dx.doi.org/10.1016/j.jksuci.2015.05.003

Lourenco, J. R., Abramova, V., Vieira, M., & Cabral, B. (2015). NOSQL database: A Software engineering perspective. *Advances in Intelligent Systems and Computing*, *353*(January), III–IV. http://doi.org/10.1007/978-3-319-16486-1

Lourenço, J. R., Cabral, B., Carreiro, P., Vieira, M., & Bernardino, J. (2015). Choosing the right NoSQL database for the job: a quality attribute evaluation. *Journal of Big Data*, *2*(1), 18. http://doi.org/10.1186/s40537-015-0025-0

Manoj, V. (2014). Comparative Study Of Nosql Document, Column Store Databases And Evaluation Of Cassandra. *International Journal of Database Management Systems (IJDMS) Vol.6, No.4, August 2014*, *3*(5), 29–39. http://doi.org/10.5121/ijci.2013.2402

Mason, R. T. (2015). NoSQL Databases and Data Modeling Techniques for a Document-oriented NoSQL Database. *Informing Science & IT Education Conference (InSITE)*, 259–268. http://doi.org/doi.org/10.28945/2245

Monika, & Ashok. (2017). A Review to the Approach for Transformation of Data from MySQL to NoSQL. *International Journal of Engineering, Applied and Management Sciences Paradigms*, *46*(1), 9–14.

Moniruzzaman, A. B. M., & Hossain, S. A. (2013). Nosql database: New era of databases for big data analytics-classification, characteristics and comparison. *arXiv Preprint arXiv:1307.0191*, *6*(4), 1–14. http://doi.org/10.1109/ICPCA.2011.6106531

Mpinda, S. A. T., Maschietto, L. G., & Bungama, P. A. (2015). From Relational

Database to Column-Oriented NoSQL Database : Migration Process. *International Journal of Engineering Research & Technology (IJERT)*, *4*(5), 399–403.

Ntarmos, N., Patlakas, I., & Triantafillou, P. (2014). Rank join queries in NoSQL databases. *Proceedings of the VLDB Endowment*, *7*(March), 493–504. http://doi.org/10.14778/2732286.2732287

Odia, T., Misra, S., & Adewumi, A. (2014). Evaluation of hadoop/mapreduce framework migration tools. *Asia-Pacific World Congress on Computer Science and Engineering, APWC on CSE 2014*, 1–8. http://doi.org/10.1109/APWCCSE.2014.7053870

Ouanouki, R., April, A., Abran, A., Gomez, A., & Desharnais, J. M. (2017). Toward building RDB to HBase conversion rules. *Journal of Big Data*, *4*(1), 10. http://doi.org/10.1186/s40537-017-0071-x

Philip Chen, C. L., & Zhang, C. Y. (2014). Data-intensive applications, challenges, techniques and technologies: A survey on Big Data. *Information Sciences*, *275*, 314–347. http://doi.org/10.1016/j.ins.2014.01.015

Pore, S. S., & Pawar, S. B. (2015). Comparative Study of SQL & NoSQL Databases. *International Journal of Advanced Research in Computer Engineering & Technology (IJARCET)*, *4*(5), 1747–1753.

RDBMS to MongoDB Migration Guide. (2016), (June), 1–17.

Rith, J., Lehmayr, P. S., & Meyer-Wegener, K. (2014). Speaking in tongues: SQL access to NoSQL systems. *Proceedings of the 29th Annual ACM Symposium on Applied Computing - SAC '14*, 855–857. http://doi.org/10.1145/2554850.2555099

Rocha, L., Vale, F., Cirilo, E., Barbosa, D., & Mourão, F. (2015). A Framework for Migrating Relational Datasets to NoSQL1. *Procedia Computer Science*, *51*, 2593–2602. http://doi.org/10.1016/j.procs.2015.05.367

Schneider, R. D. (2012). *Hadoop for Dummies Special Edition*. *The effects of brief mindfulness intervention on acute pain experience: An examination of individual difference* (Vol. 1). http://doi.org/10.1017/CBO9781107415324.004

Schram, A., & Anderson, K. M. (2012). MySQL to NoSQL: Data Modeling Challenges in Supporting Scalability. *Proceedings of the 3rd Annual Conference on Systems, Programming, and Applications: Software for Humanity - SPLASH '12*, (January 2016), 191. http://doi.org/10.1145/2384716.2384773

Schreiner, G. A., Duarte, D., & Mello, R. dos S. (2015). SQLtoKeyNoSQL: A Layer for Relational to Key-based NoSQL Database Mapping. *Proceedings of the 17th International Conference on Information Integration and Web-Based Applications & Services*, 74:1--74:9. http://doi.org/10.1145/2837185.2837224

Serrano, Han, & Stroulia. (2015). From Relations to Multi-Dimensional Maps : Towards A SQL-to-HBase Transformation Methodology. *2015 8th IEEE International Conference on Cloud Computing*, 81–89. http://doi.org/10.1109/CLOUD.2015.21

Sharma, R., & Kashyap, Y. (2016). A study of nosql databases and working overviews. *International Journal of Recent Trens in Engineering & Research*, 43–50.

Sharma, S., Shandilya, R., Patnaik, S., & Mahapatra, A. (2016). Leading NoSQL models for handling Big Data: a brief review. *International Journal of Business Information Systems*, *22*(1), 1. http://doi.org/10.1504/IJBIS.2016.075714

Sharma, S., Tim, U. S., Wong, J., Gadia, S., & Sharma, S. (2014). A brief review on leading big data models. *Data Science Journal*, *13*(December), 138–157. http://doi.org/10.2481/dsj.14-041

Vohra, D. (2016). *Apache HBase Primer*. *Apache HBase Primer*. http://doi.org/10.1007/978-1-4842-2424-3

Winarno, E., Hadikurniawati, W., & Rosso, R. N. (2018). Location based service for presence system using haversine method. *Proceedings - 2017 International Conference on Innovative and Creative Information Technology: Computational Intelligence and IoT, ICITech 2017*, *2018–Janua*. http://doi.org/10.1109/INNOCIT.2017.8319153

Yang, C.-T., Liu, J.-C., Hsu, W.-H., Lu, H.-W., & Chu, W. C.-C. (2013). Implementation of Data Transform Method into NoSQL Database for Healthcare Data. *2013 International Conference on Parallel and Distributed Computing, Applications and Technologies*, 198–205. http://doi.org/https://doi.org/10.1109/pdcat.2013.38

Yoo, J., Lee, K.-H., & Jeon, Y.-H. (2018). Migration from RDBMS to NoSQL Using Column-level Denormalization and Atomic Aggregates *. *Journal of Information Science And Engineering 34*, *259*, 243–259. http://doi.org/10.6688/JISE.2018.34.1.15

Zaki, A. K. (2014). NoSQL Databases: New Millennium Database for Big Data, Big Users, Cloud Computing and its Security Challenges. *IJRET: International Journal of Research in Engineering and Technology*, 403–409. http://doi.org/10.15623/ijret.2014.0315080

Zhao, G., Li, L., Li, Z., & Lin, Q. (2014a). Multiple Nested Schema of HBase for Migration from SQL. *IEEE International Conference on P2P, Parallel, Grid, Cloud and Internet Computing*, 338–343. http://doi.org/10.1109/3PGCIC.2014.127

Zhao, G., Lin, Q., Li, L., & Li, Z. (2014b). Schema conversion model of SQL database to NoSQL. In *Proceedings - 2014 9th International Conference on P2P, Parallel, Grid, Cloud and Internet Computing, 3PGCIC 2014* (pp. 355–362). http://doi.org/10.1109/3PGCIC.2014.137

Zhu, F., Liu, J., Wang, S., Xu, J., Xu, L., Ren, J., & Huang, T. (2016). Hug the Elephant: Migrating a Legacy Data Analytics Application to Hadoop Ecosystem. *2016 IEEE International Conference on Software Maintenance and Evolution (ICSME)*, 177–187. http://doi.org/10.1109/ICSME.2016.14