International Journal of Psychosocial Rehabilitation, Vol. 23, Issue 04, 2019
ISSN: 1475-7192

Development of a Process for Migration of Data from Relational to Non-Relational Database

Fahad Noor, Dhason Padmakumar and Dr. Sivakumar Vengusamy

Abstract--- *As we know the world is moving towards the cloud computing very fast, it has become the need of every sector. Cloud computing is one of the rapid growing and fast adapting technology so data confidentiality and other issues are also growing and it needs to be resolved and deal on prior basis. The purpose of this paper is to identify reliability issues of cloud based applications in mass media sector of Malaysia and how world have suffered due this problem and as well as how Malaysia can suffer in future if they not constantly work on this domain. This paper consists of various incidents and thorough those we can identify the importance of cloud computing security.*

Keywords--- *NoSQL Data Migration, NoSQL Security Issues, NoSQL Reliability Issues, NoSQL Reliability Techniques.*

I. INTRODUCTION

Database is considered to be one of the main components of any information system and it saves the information or data in an organized way so that user can access according to its needs, [13]. Currently there are two types of databases which are considered to be most popular Relational and Non-Relational databases. With the passage of time the amount is data is increasing and in current world each and every chunk of data is useful and to fulfill the need of current world many types of databases are available. Keeping in mind the sensitivity and importance of every piece of data sometimes it becomes necessary to migrate data from legacy database to new database. This basic steps of data migration classified as, Transport and Load (ETL) and is based on three steps (1) Extracting data from key source, (2) Transmute the extracted data, and (3) Migrating the data to new database, [19].

Relational databases mostly follow same kind of structures and it saves the data in tables, these tables are consisting of rows and columns and these columns are represented by its name. Relational databases do not require much training, it is easy to use and maintain. But due increase in volume of data relational databases are not considered to be ideal choice therefore organizations are moving towards Non-Relational databases, [5]. In inverse Non-Relational databases ha ability to deal with huge volume of data and one of the main reasons of that is it does not maintain relationships between tables. Non-Relational databases are popular due to its simple structures, accessible schema design and efficient data extraction speed.

The main difference between Relational and Non-Relational databases are capable to store both types of data structured and non-structured.

Fahad Noor, Asia Pacific University of Technology and Innovation Technology Park Malaysia, Kuala Lumpur, Malaysia.
E-mail:Fahadnoor076@gmail.com
Dhason Padmakumar, Asia Pacific University of Technology and Innovation Technology Park Malaysia, Kuala Lumpur, Malaysia.
Dr. Sivakumar Vengusamy, Asia Pacific University of Technology and Innovation Technology Park Malaysia, Kuala Lumpur, Malaysia.

Received: 17 Sept 2019 | Revised: 19 Oct 2019 | Accepted: 20 Nov 2019

Basically, non-relational data bases are categorized majorly in three different types. Key-value, Graph-Oriented and Document-Store. Non-Relational databases supports key value pair saving model, [18]. In Graph-Oriented databases the elements are indicated with the help of pointers. Whereas Document-Store databases are kind of Non-Relational databases which stores data in flat files and the data is saved in JSON or BJSON format, [7].

As it is highlighted above that on one hand organizations are moving towards Non-Relational databases due to increasing amount of data, on other hand they do not want to lose legacy system data plus want to maintain user friendly environment experience even in new version of information systems. One solution is organizations keep using their legacy or old version software or information systems only to fetch old data and after new release start maintaining the new transactions of data in that but this will increase efforts of employees and another option is that organization should prefer to migrate the old data to new released software or information system and keeping security concerns and user easiness in mind data migration is seems to be a better option. Basically, this is the main reason of conducting this research that how the data of legacy systems and Relational databases can be migrated to new systems developed on Non-Relational databases. Another reason of this research is it can benefit developers to identify security loop holes and additional efforts that needs to be given by them while adopting Non-Relational databases, that does not it a failure of Non-Relational databases actually with the passage of time companies are working to enhance securities of Non-Relational databases. Furthermore, in this study MySQL has been adopted to represent Relational databases and MongoDB is adopted to represent Non-Relational databases.

II. RELATED WORK

2.1. Why MongoDB

The main reason of focusing MongoDB is that it has many advantages likewise:

- It is an open source Non-Relational and NoSQL database.
- It is based on documents.
- It supports centralized distributed architecture; it can mirror data on more than one server with master slave approach.
- No restrictions to defined number of fields in advance while creating document.
- It is compatible with almost every platform like Java, PHP, .Net et cetera.
- Faster in terms of speed.
- Supports multiple environments like Windows, Linux et cetera.
- Optimized query performance.
- Scalable

In the figure 1 below structure of MongoDB can be found for the better understanding.

Received: 17 Sept 2019 | Revised: 19 Oct 2019 | Accepted: 20 Nov 2019

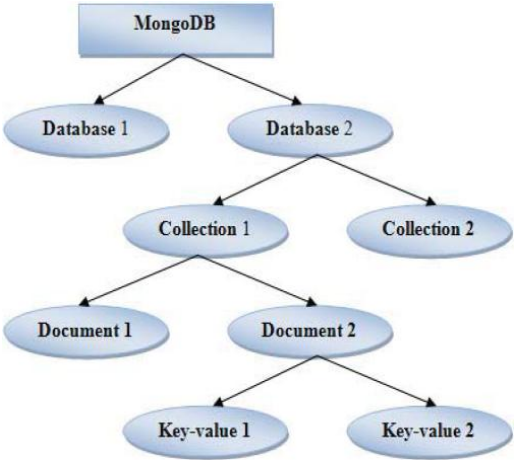

Figure 1: Structure of MongoDB [5]

2.2. Reasons to Adopt Non-Relational Databases

There are plenty of reasons to adopt non-relational databases few of them are as follows [9].

2.2.1. Current Database is not enough

When existing software database is not enough to handle the volume of data this also effects the speed of software for example retrieving the data from GBs of data.

Moreover, as it can be analyzing from existing volume of data the popular databases are not enough for example MariaDB and SQL (Structured Query Language).

2.2.2. Upgradation of Database Version

Sometimes upgrade version of database is not compatible with old versions and due to that it becomes requirement to migrate data from old version to new version.

On other hand queries in compatibility can also be faced so in this case programmers need to upgrade the existing queries as well.

2.2.3. Change in Company's Policy

Now a day's organizations are very concern about their security due to increase in cyber security incidents. So, to deal with security issues and challenges they need to upgrade the databases and it leads to migration of data from old to new version or sometimes even change in technology of databases.

2.2.4. Financial Crises or Cost Cutting in Organization

In this kind of situation most of the organizations prefer to move open source or free products rather than purchasing expensive products. So same goes with database engines as well because in financial crises organization's first priority will be always to reduce the expenses.

Received: 17 Sept 2019 | Revised: 19 Oct 2019 | Accepted: 20 Nov 2019

2.2.5. Single Database Rather than Using Multiple

Many of the information system and software uses multiple databases according to the nature of data on one hand it helps to organize the data but on the other hand it increases the load and lines of code in software along with maximizing the loop holes for attacks as well.

Non-Relational or NoSQL databases are vice versa of SQL databases. As mentioned above that as a model of non-relational databases this report will focus on MongoDB because it is one the most popular NoSQL and as well as Non-Relational database.

III. PROS OF ADOPTING NON-RELATIONAL OR NO SQL (MONGODB) DATABASES FOR PROGRAMMERS

3.3.1. JSON Formatted Files

Advantage of using NoSQL or Non-Relational databases to programmers is NoSQL supports JSON formatted files called documents and in these documents.

Because of that programmer can save his database on any level for example a programmer can save complex data in just single object or when it comes to MongoDB, programmers can save the data in form of multi-dimensional arrays because it supports BJSON (Binary Serialized Array), so developer or programmer can easily access the data with the help of single query rather than applying multiple joins or sub queries like Relational databases, [3]. Moreover, this approach not only helps programmers to retrieve data in single object but also makes the queries faster than relational databases.

3.3.2. Object ids (String Keys)

Non-Relational or NoSQL databases supports Object ids (string keys) rather than integer numbers for the uniqueness of each record, MongoDB also works on same approach[1].

These keys or objects ids are assigned automatically to each record to avoid redundancy between each record and with the help of these keys programmers can easily access or update each record, these object ids work same as primary keys in relational databases, but the processing speed is faster than relational databases.

3.3.3. Unstructured Data Support

Because of unstructured data support programmers can handle changes in data records faster than Relational databases, programmers do not need to write the ETL routines. Whereas some of the researchers emphasis on ETL in NoSQL databases as well to build proper schema of database in NoSQL as well, [20].

3.3.4. Object-Oriented Approach and Queries

Non-Relational databases supports object-oriented approach when it comes to storing data and writing queries. This approach not only enhance the performance of queries but also helps programmers to access data in a hieratically.

Received: 17 Sept 2019 | Revised: 19 Oct 2019 | Accepted: 20 Nov 2019

For an example MongoDB also supports the same approach it not only reduces line of codes for programmers but also flexible for them understand and use accordingly. MongoDB practices java script callback functions approach for its advance queries, these functions helps programmers to manipulate with retrieved data [2].

3.3.5. Object-Relational Mapping

MongoDB supports Object Relation Mapping (ORM), according to [17] it is an approach by which database relations can be mapped using entities of object-oriented languages. The vital goal of using this approach is by using this programmer get rid of writing queries. On other hand if programmers use dual way that is relation and object-oriented it can increase his efforts plus line of codes as well.

Additionally, when it comes to relational databases their performance is better with global queries which consumes huge amount of memory, but accessing the data by object-oriented works effective with small or trivial volume of data, by which systematic gap can be reduced between relational data and object, [8].

IV. CONS / LIMITATIONS OF ADOPTING NON-RELATIONAL OR NO SQL (MONGODB) DATABASES FOR PROGRAMMERS

4.1. Object-Relational Mapping

As [17] stated, one of the main problems of adopting ORM over SQL is abstraction which means it cannot abstract details of realization completely. Few of the ORM's realizations programs works like tools for code generation. The reason is abstraction helps programmers to write code in simpler way, but if programmers use ORM with knowledge of SQL as well it increases their efforts.

Another problem of ORM is inefficiency which means if programmer fetches object data from database ORM cannot identify that exactly which property of object can be use or amended, so due to that programmer needs to extract whole record which increases numbers of requests and increases processing time as well. Basically, ORM is not intelligent enough to combine the number of requests, which leads to more usage of cache.

4.4.1. Syntax Differences / No Standard Query Language

One of the biggest differences in Relational databases and MongoDB, its queries are functions based whereas relational databases for example MySQL uses SQL as a procedural language. Moreover, as mentioned above MongoDB uses callback functions approach like java scripting for its complex queries so programmers can manipulate with retrieved data, and MySQL uses store procedures, joins and subqueries for complex data retrieval.

This difference sometimes becomes big hurdle for programmers who have less experience in NoSQL databases as it can be seen that NoSQL is new and emerging technology as compare to relational databases. Another drawback of NoSQL is there is no standard query language which means each database has its own query language, [2].

4.4.2. Lack of Specialist Tools

Non-Relational database or NoSQL is an emerging database type, its popularity is growing with pace and the adoption rate by development communities is very fast. But due to its lack of specialist tools programmers are facing issues, plenty of model transformation stratagems are made on the basis of various database administrators, [6].

4.4.3. Model Transformation Challenge

Basic issue faced by programmers while adopting NoSQL databases or MongoDB is design of its physical model. It is not easy for developers to decide where to save which data and which tables or documents should be embedded together. On one contrary, if programmers will embed tables together it will enhance the performance but, in the end, will lead to data redundancy. Whereas on other hand many of the NoSQL databases does not support join operations for an example MongoDB. So, to read data from multiple tables programmers needs to write multiple queries that leads to extra line of codes in application, [6].

V. SECURITY CHALLENGES WHILE ADOPTING NON-RELATIONAL OR NO SQL (MONGODB) DATABASES

Distributed Web Applications and Could Computing has become very popular in recent years, due to that volume of data has also increased and therefore demand of database which can deal with large amount of data has also increased. As per cloud computing's requirement which is 24/7 availability of system the databases should be available and scalable. In last few years companies have adopted numerous kinds of non-relational databases. Due to increase in demand and popularity of NoSQL databases like MongoDB and huge volume of user's sensitive data in these databases increase the apprehension of security and confidentiality of data [11]. The primary concern of MongoDB designers was never security as a result the design of MongoDB have few loop holes.

5.1. Data Files of NoSQL (MongoDB)

MongoDB data-files are never encrypted automatically, and this is one the easiest loop hole for any attacker that means attacker can simply access the user's information through these files, [10]. To avoid this data should be encrypted on application level before inserting to database. Additionally, permission for file system of operating system should also be kept in consideration to prevent unauthorized access to information.

5.5.1. Client Interface

According to [11] MongoDB's client interface supports protocol on binary wire-level. MongoDB uses TCP 27017 port protocol by default and it is normally used by all numerous application drivers. Additionally, MongoDB database uses this protocol and port to achieve replication of both variants. Moreover, 1000 port is used as a HTTP server, 28017 TCP port is by default port. Basically, management level statistics are provided by this HTTP server, but this port can also be used for configuration of RESTful interface for database, this feature can be enabled by setting rest=true in configuration file of database or by command-line. Wire-level binary protocol is not encrypted and compressed, and SSL and TLS is also not supported by internal HTTP server. But HTTP server could be behind

proxy server for example like Apache HTTPD. But by default, MongoDB does not provide proper security whereas in latest updates of MongoDB there are some features which can be used to enhance the security but then also it must be applied administrator or programmer explicitly.

5.5.2. Possibility for Injection Attacks

As mentioned before, MongoDB queries are based on java script. Most of the queries written by programmer are short functions of java scripts. Not only this java script function and snippets can also be stored in database and could be reason for injection attacks, [14]. As an example in db.system.json collection. Java script is considered to be an interpreted language, so it increases the risk of injection attacks to database. From example below can be identified how java script can perform injection in database collection.

"db.myCollection.find({ a : { $gt: 3 } });

db.myCollection.find({ $where: "this.a > 3" });

db.myCollection.find("this.a > 3");

db.myCollection.find({ $where: function() {return this.a > 3;}});"

In the statements above which are extracted from [11], it can be easily seen second and third statement that where clause with values are passed as an argument from developer. Whereas in fourth statement a function is passed with along with $where object and if these statements will be saved in database and executed without validating the data then injection attacks can easily carry out.

5.5.3. No Authentication

MongoDB does not support any authentication in shared mode whereas in standalone mode programmer can enable authentication feature [15]. Difference between replica-set and standalone modes is that in replica-set client authenticates to database and earlier joining the cluster every replicated server should also authenticate. Authentication is done on the basis of pre-shared secret in both modes. So, this is another security flaw in MongoDB and MongoDB does not provide proper authentication for shared environment as mentioned above and this must be taken care by programmers explicitly only.

5.5.4. Weak Authorization

Like authentication MongoDB also does not supports authorization in shared mode, if the authentication is enabled it only supports only two kinds of users that is read-write and read-only [15]. Users with read-only permissions also have access to databases to which they are allowed. User with admin rights is called DBA user and this user have full have access to database that is to read and write and give rights to any other user.

5.5.5. Auditing

MongoDB does not have proper auditing, it does not have any proper logs for activities performed in database, [16]. For an example if new database is been created only that time the log for creation of this database appear in this file but if user performs add, update or delete operations MongoDB does not record that in its log files.

Received: 17 Sept 2019 | Revised: 19 Oct 2019 | Accepted: 20 Nov 2019

VI. MIGRATION OF DATA FROM RELATIONAL DATABASE TO NON-RELATIONAL DATABASE

At the moment, the strategies use by data migration tools is very basic, for example migrating each and every table one by one from relational to particular MongoDB collection. Assume that MongoDB's physical model is already defined. It is not possible to dictate tools to follow any pre-defined model for migration. To migrate data programmers need to use some kind of APIs and with the help of them they can write their own programs. If there is any change in model the program should also need to be written again. Currently, there is no process that can migrate the data automatically on the basis model.

These difficulties in migration of data can be deal with the solution given by [6], according to the researcher approach the characteristics of query and data of relational database need to be keep in consideration. Description tags can be extract from relational database logs or directly from database on basis of these characteristics. Moreover, description tags are should be convert to action tags, different strategies of model transformation can also be customized. The process of novel model transformation is developed based on tags and relational databases relationships. Therefore, after model transformation completion the above method can migrate data to MongoDB on the basis of model transformation result. Tool is also been developed for guidance to perform migration and transformation. This tool helps to modify transformation of model results to suit application's performance and data can be easily migrated from relational database to MongoDB. This also enhance the read performance of database.

Furthermore, according to the methodology proposed by [5] can be understand clearly by steps in figure 3. The main steps can be identified in black circles. To elaborate each step further, an example of database will be used imitating company's orders in figure 4.

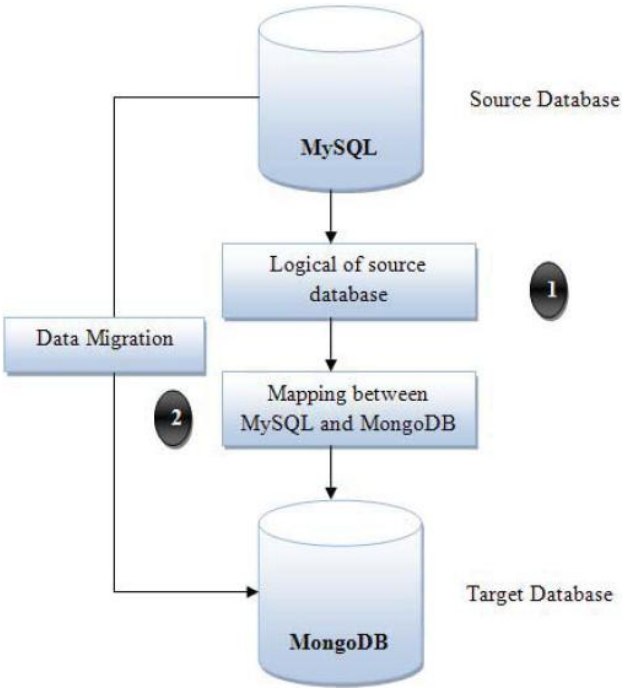

Figure 3: Process for Data Migration from MySQL to MongoDB [5]

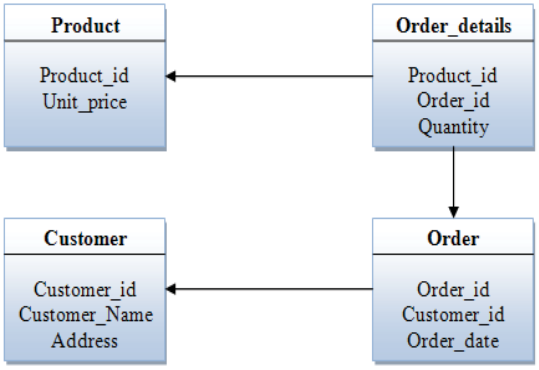

Figure 4: Relational Database Logical Structure [5]

Steps proposed by Hanine to migrate data from relational database to NoSQL (MongoDB):

"Step 1: Loading the logical structure of the source Database"

Initially it is important to specify the source of relational database. After that the source database is connected to acquire information about version and type. It is also necessary to stipulate the information about the target database (MongoDB or NoSQL). Moreover, the depiction of source database's relational model needs to be obtained. To proceed with that names of all tables are required with its relationships and attributes. Information about relationships of tables can be attain with the help of foreign keys and constraint of particular table.

"Step 2: Mapping between relational model and MongoDB model"

In this step mapping is done between Relational database model and NoSQL database (MongoDB). Basically, here a link is developed between attributes of source that is relational database and target that is NoSQL database model. The implementation of database tables and its attributes is done in different way, JDBC driver is to be use in one from them. While migrating the relational model, system will suggest some appropriate tables to migration from source to target database. The main difference between relational databases and NoSQL (MongoDB) is NoSQL databases does not support JOINs and this the main concern.

One of the solution proposed by [4] to migrate data from relational database to NoSQL is:

Basically, this proposed solution is based on two phases the first is migration and the other is mapping. Whereas this framework is based on two modules, the foremost is migration which guarantees unified data migration from relational to NoSQL (MongoDB) database, it can be done on basis of metadata hold in data dictionary and module for mapping, this module helps in execution of SQL queries without changing application code. It assists programmers to retrieve data from database and let them code queries in relational databases.

The executed queries from application are stored in NoSQL layer and these queries are converted explicit format of NoSQL database, this generated result is re-store by framework and then convert to application's definite format. On other hand, there is an open source tool use for communication between database and application, after that all these coming requests from mediator are convert with the help of module named Convert, so these queries can be supported and compatible with NoSQL database (MongoDB).

As can be seen that many researchers came up with some process and methods to migrate data from relational to NoSQL database or MongoDB, likewise according to [12] there are various way of migration:

Migration through online tools with non-stop synchronization, in this method application is always connected with database and works non-stop and it also parallelly utilizes resources from database as well to perform migration.

Migration through offline tools, in this method migration will be done by software or tool for that but the system will remain down.

Migration through scripting, in this process or method the migration of data will be done with the help of scripts, and in this process the system will remain down that mean no computation power will be used from system.

For proper understanding of process or method figure 5 can be found below.

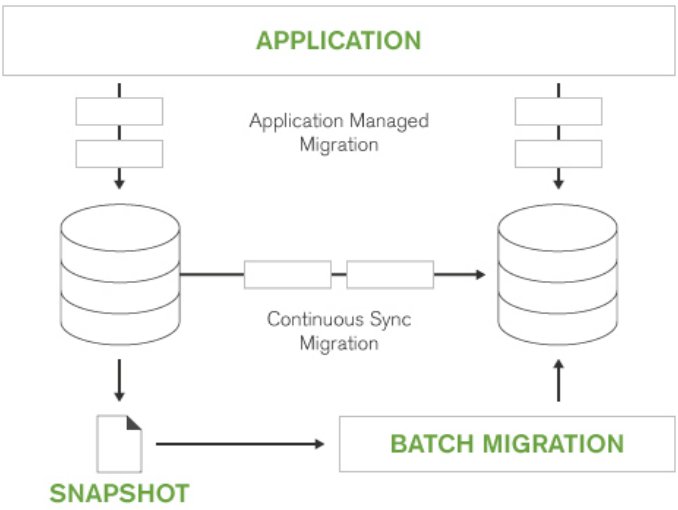

Figure 5: Migration Method from RDBMS to MongoDB [12]

VII. CONCLUSION

It is concluded that NoSQL (MongoDB) on one hand it has benefits but on the other hand it has many issues as well. As it has been discussed that by adopting NoSQL programmer must put a lot of efforts on code level and security level also. As we know that designer of NoSQL like MongoDB does not emphasis on security of database that does not mean that there was lack of capabilities, but it means that their primary focus was not security. So most of the security flaws must be taken care by their own. The application should be made secure by programmer as well if not it will face a lot of cyber-attacks and in the result, it will be useless for users. Whereas in some cases attackers breach the security of database and inject malicious java scripts. The vital issues found in NoSQL (MongoDB) is they support frail authorization and authentication mechanism between server and client, plus it lacks in database level encryption which mean the data files does not have proper encryption for example fine-grained or RBAC authorization this could also be a reason of Denial of Service attacks (DDOS) or injection attacks.

Received: 17 Sept 2019 | Revised: 19 Oct 2019 | Accepted: 20 Nov 2019

Moreover, it is found that there are multiple methods, process and techniques to migrate data from Relational databases to Non-Relational databases but there must be some measures that should be taken care as well because there is a big difference between structure and model between Relational databases and Non-Relational or NoSQL databases. On one hand, some of these methods are capable enough that they automatically generate database schema while on other hand, some scripting methods. Finally, it has been found that the present research and tools on data migration are not that much capable enough migrate the database perfectly. Therefore, it is our determination that after conducting this research we will come up with something valuable that will not only help developers to not panic about migration of data before adopting NoSQL (MongoDB) but will also help them to get aware about the security weaknesses before developing information systems using Non-Relational databases (MongoDB). This will help them to make preventions at development stage rather than giving efforts on maintenance stage of system after it goes live. Therefore, this research will also be helpful for organizations as well when it comes to saving cost on maintenance stage software.

REFERENCES

[1] Agrawal, S. et al. (2015) 'Survey on Mongodb : *An Open-',* 6(12), pp. 1–11.
[2] Boicea, A., Radulescu, F. and Agapin, L. I. (2012) 'MongoDB vs Oracle - database comparison'. doi: 10.1109/EIDWT.2012.32.
[3] Chauhan, D. (2017) 'Using the Advantages of NOSQL : A Case Study on MongoDB', (February).
[4] Database, C. and Sipos, R. (2017) 'Migration from relational to NoSQL database Migration from relational to NoSQL database'. doi: 10.1088/1757-899X/263/4/042055.
[5] Hanine Mohamed, Bendarag Abdesadik, B. O. (2015) 'Data Migration Methodology from Relational to No SQL Databases', 9(12), pp. 2566–2570.
[6] Jia, T. et al. (2016) 'Model Transformation and Data Migration from Relational Database to MongoDB'. doi: 10.1109/BigDataCongress.2016.16.
[7] Klettke, M. and Scherzinger, S. (2017) 'Schema Extraction and Structural Outlier Detection for JSON-based NoSQL Data Stores', 1, pp. 425–444.
[8] Loureno J R, C. B. and C. P. (2015) 'Choosing the right nosql database for the job', *a quality attribute evaluation Journal of Big Data* 2 18.
[9] M. Alam, S. W. K. (2006) 'Migration from relational database to object oriented database', *Journal of Computer Science*, 2(10), pp. 781–784.
[10] Noiumkar, P. and Chomsiri, T. (2014) 'A Comparison the Level of Security on Top 5 Open Source NoSQL Databases'.
[11] Okman, L. et al. (2011) 'Security Issues in NoSQL Databases'. doi: 10.1109/TrustCom.2011.70.
[12] Oliveira Fábio, Oliveira Abílio, A. B. (2018) 'Migration of Relational Databases to NoSQL - *Methods of Analysis Fábio Oliveira Abílio Oliveira Bráulio Alturas',* 9(2), pp. 227–235. doi: 10.2478/mjss-2018-0042.
[13] Robbins, R. J. (1995) 'Database Fundamentals', pp. 1–31.
[14] Ron, A. et al. (2015) 'No SQL , No Injection ? Examining NoSQL Security'.
[15] Sahafizadeh, E. and Nematbakhsh, M. A. (2015) 'A Survey on Security Issues in Big Data and NoSQL', 4(4), pp. 68–72.
[16] Sathyadevan, S., Muraleedharan, N. and Rajan, S. P. (2015) 'Enhancement of Data Level Security in MongoDB', pp. 199–200. doi: 10.1007/978-3-319-11227-5.
[17] Vasilchuk, O. I. et al. (2018) 'The Creation of Scalable Tools for Solving Big Data Analysis Problems Based on the MongoDB Database'.
[18] Verma, M. P. and Id, E. (2017) 'Overview of Indexes Used in NOSQL Databases of MongoDB Architecture', (June), pp. 1057–1059.
[19] Walek, B. and Klimes, C. (2012) 'A methodology for Data Migration between Dif- ferent Database Management Systems', 6(5), pp. 536–541.

[20] Yangui, R. and Nabli, A. (2017) 'ETL Based Framework for NoSQL Warehousing', 1, pp. 40–53. doi: 10.1007/978-3-319-65930-5.
[21] P. Mary Jeyanthi, Santosh Shrivastava Kumar "The Determinant Parameters of Knowledge Transfer among Academicians in Colleges of Chennai Region", *Theoretical Economics Letters*, 2019, 9, 752-760.
[22] P. Mary Jeyanthi, "An Empirical Study of Fraudulent and Bankruptcy in Indian Banking Sectors", *The Empirical Economics Letters,* Vol.18; No. 3, March 2019.
[23] Metaheuristic techniques", *International Journal of Business Intelligence Research,* - Volume 5, Issue 1, April-2014. URL: https://dl.acm.org/citation.cfm?id=2628938; DOI: 10.4018/ijbir.2014010105, which is in C category of ABDC List.
[24] P. Mary Jeyanthi, "INDUSTRY 4.O: The combination of the Internet of Things (IoT)and the Internet of People (IoP)", *Journal of Contemporary Research in Management,* Vol.13; No. 4 Oct-Dec, 2018, ISSN: 0973-9785.
[25] P. Mary Jeyanthi, "The transformation of Social media information systems leads to Global business: An Empirical Survey", *International Journal of Technology and Science (IJTS),* issue 3, volume 5, ISSN Online: 2350-1111 (Online). URL: http://www.i3cpublications.org/M-IJTS-061801.pdf
[26] P. Mary Jeyanthi," An Empirical Study of Fraud Control Techniques using Business Intelligence in Financial Institutions", *Vivekananda Journal of Research* Vol. 7, Special Issue 1, May 2018, Page no: 159-164.
[27] Mary Jeyanthi, S and Karnan, M.: "Business Intelligence: Artificial bear Optimization Approach", *International Journal of Scientific & Engineering Research,* Volume 4, Issue 8, August-2013.
[28] Mary Jeyanthi, S and Karnan, M.: "Business Intelligence: Optimization techniques for Decision Making", *International Journal of Engineering Research and Technology,* Volume 2, Issue 8, August-2013.
[29] Mary Jeyanthi, S and Karnan, M.: "A New Implementation of Mathematical Models with metaheuristic Algorithms for Business Intelligence", *International Journal of Advanced Research in Computer and Communication Engineering,* Volume 3, Issue 3, March-2014.
[30] Dr. Mary Jeyanthi: "Partial Image Retrieval Systems in Luminance and Color Invariants: An Empirical Study", *International Journal of Web Technology* (ISSN: 2278-2389) – Volume-4, Issue-2.
[31] Dr. Mary Jeyanthi: "CipherText Policy attribute-based Encryption for Patients Health Information in Cloud Platform", *Journal of Information Science and Engineering* (ISSN: 1016-2364)
[32] Mary Jeyanthi, P, Adarsh Sharma, Purva Verma: "Sustainability of the business and employment generation in the field of UPVC widows" (ICSMS2019).
[33] Mary Jeyanthi, P: "An Empirical Survey of Sustainability in Social Media and Information Systems across emerging countries", *International Conference on Sustainability Management and Strategy"* (ICSMS2018).
[34] Mary Jeyanthi, P: "Agile Analytics in Business Decision Making: An Empirical Study", *International Conference on Business Management and Information Systems"* (ICBMIS2015).
[35] Mary Jeyanthi, S and Karnan, M.: "Business Intelligence – soft computing Techniques", *International Conference on Mathematics in Engineering & Business Management* (ICMEB 2012).
[36] Mary Jeyanthi, S and Karnan, M.: "A Comparative Study of Genetic algorithm and Artificial Bear Optimization algorithm in Business Intelligence", *International Conference on Mathematics in Engineering & Business Management* (ICMEB 2012).
[37] Mary Jeyanthi, S and Karnan, M.: "Business Intelligence: Data Mining and Optimization for Decision Making ", 2011 *IEEE International Conference on Computational Intelligence and Computing Research* (2011 IEEE ICCIC).
[38] Mary Jeyanthi, S and Karnan, M.: "Business Intelligence: Data Mining and Decision making to overcome the Financial Risk", 2011 *IEEE International Conference on Computational Intelligence and Computing Research* (2011 IEEE ICCIC).
[39] Dr. Mary Jeyanthi, S: "Pervasive Computing in Business Intelligence", *State level seminar on Computing and Communication Technologies.* (SCCT-2015)
[40] Dr.P.Mary Jeyanthi, "Artificial Bear Optimization (ABO) – A new approach of Metaheuristic algorithm for Business Intelligence", ISBN no: 978-93-87862-65-4, *Bonfring Publication.* Issue Date: 01-Apr-2019
[41] Dr.P.Mary Jeyanthi , "Customer Value Management (CVM) – Thinking Inside the box" – ISBN : 978-93-87862-94-4, Bonfring Publication, Issue Date: 16-Oct-2019.
[42] Jeyanthi, P. M., & Shrivastava, S. K. (2019). The Determinant Parameters of Knowledge Transfer among Academicians in Colleges of Chennai Region. *Theoretical Economics Letters,* 9(4), 752-760.