SQL to MongoDB (NoSQL) Migration: Evaluation and Analysis

Falgunee Parsana[1], Achyut C. Patel[2]
Department of CS & I.T-ATMIYA UNIVERSITY, Department of Statistics-Saurashtra University
[1]falguni.parsana@atmiyauni.ac.in
[2]acp2809@gmail.com

***Abstract*— Large volume of data as well as analytical data requires to migrating from structured to un-structured data (NoSQL) to characterize the data. This conversion is demanding for the reason that of the lack of routine conversion procedure and the necessity of guarantee both presentation and precise demonstration. In this paper, we evaluate normally used mapping from structured (SQL) to Unstructured database i.e., NoSQL. We have done the comparison among these two databases in terms of fetching time in order to get the best performance. In this paper, we have used MySQL database for SQL and MongoDB for NoSQL structures. This experiment provides capable and proficient results when using a multiple documents with a reference association with a different document.**

***Keywords*— NoSQL, MongoDB, Big Data**

I. Introduction

Nowadays, NoSQL Databases useful in industrial businesses in cloud environment, web-based application, Internet on Things and large volume of big data [8]. The main aim of accepting such databases is to require faster data accessibilities and analysis [7].

“NoSQL is non-relational, schema less, and can handle unstructured and different types of data efficiently” [2, 3, 5]. Unstructured data that is NoSQL does not support join query and it becomes a lesser powerful query language to fetch records from the database than any other Structured or relational Database [4]. In the same way, the structure of NoSQL database is different, as it wants convention API to cooperate with the communication with the data.

Though, migration from SQL to NoSQL database becomes challenging task as it requires validate system necessities and to maintain identical functionality and data reliability of the new NoSQL DB schema [4]. The migrating procedure consists of two different requirements: a. Modify and developing design. b. Data migrations [1]. The migrating process is comparatively easier with ensures data integrity and performance. NoSQL tasks can be achieved by either professional expert or by using GUI tools.

NoSQL Database cannot support join queries and have no concept of foreign keys. In this structure, there are three ways to represent entities relationships: by giving reference, embedding as well as mixing of embedding and by giving reference [5, 6].

In this paper, we examine and evaluate retrieval process of different NoSQL designs and relational database. The complexity of queries differs from uncomplicated queries to complex queries that involve dissimilar levels of joins and aggregation.

II. METHODOLOGY AND ANALYTICAL EVALUATION

In this segment, we present the methodology and analytical evaluation setup for compares among NoSQL different structures. First, we represent the relational model over here. Next, we represent the NoSQL model using MongoDB.

A. Relational Database

We are using a standard database named Emp.

The entity-relationship diagram (ERD) is shown in below figure.

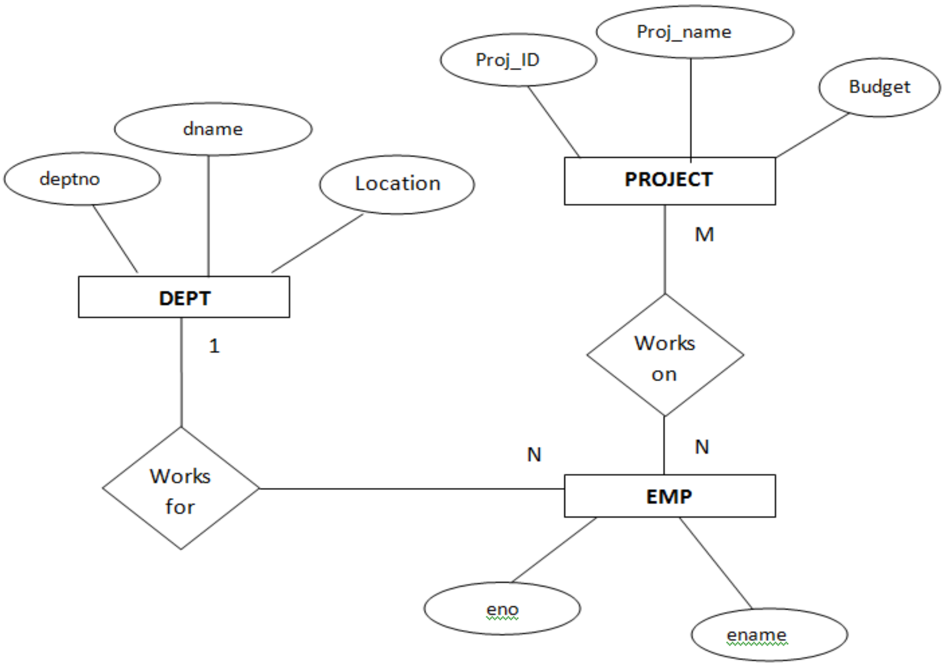

Fig.1 ERD for the standard database

The above Database of Employee contains 4 relationships. Each dept have many employee working and works on many projects as well. This many to many relationships among the two entities like employee and project represents the fact that each employee may be involve in one or many projects in the department. Here one-to-many relationship represents the parent-child relations. So logically connect together, we require primary key as well as foreign key in such tables.

B. *Structure of MongoDB Model*

By transforming a structured database into NoSQL model, MongoDB has two ways: embed and to give reference. In my paper, I have transformed the database of EMP into three MongoDB NoSQL structure. First model based on to embed entire data about employee in single collections known as EmployeeInfo collection. Then another document of dept details and child details of an employee considered as inner document.

Tables	***Relations***	***Connection***	***Records***
Emp	*1-N*	*Dept*	*2 M*
	N-M	*Works on*	
	1-N	*Child*	
Dept	*1-N*	*Project*	*100*
	1-N	*Emp*	
Project	*N-1*	*Dept*	*2 M*
	N-M	*Works on*	
Child	*N-1*	*Employee*	*40M*

Table I: Representation of Entities with their relationships

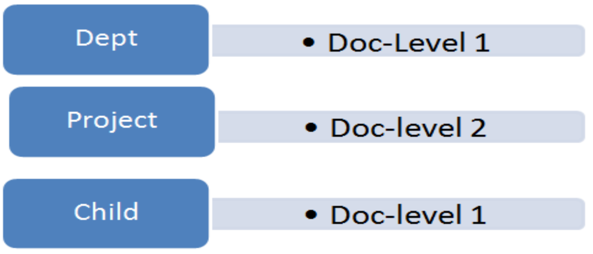

Fig. 2 Employee Collection – MongoDB model

The 2nd structure separates the data in two collections. First is Dept collection that consists of details of department and their various projects. Project details must be represented as embedded in Dept collections. And second is EMP collections consists of details of employees and their dept and projects details. In this experiment, DEPT collections consists of 100 docs as well as EMP collections consists of 2 M docs

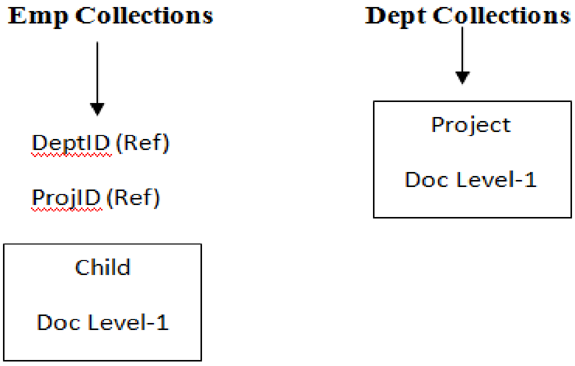

Fig. 3 MongoDB model Structure 2

The last NoSQL model presents every relational object into divided collection. All associations are implementing as using references. The Dept group has not any reference connection with any further collection. The Emp group has a reference connectivity with Dept group. The project group has reference connectivity with department collection. Child document has a reference relationship with employee collection.

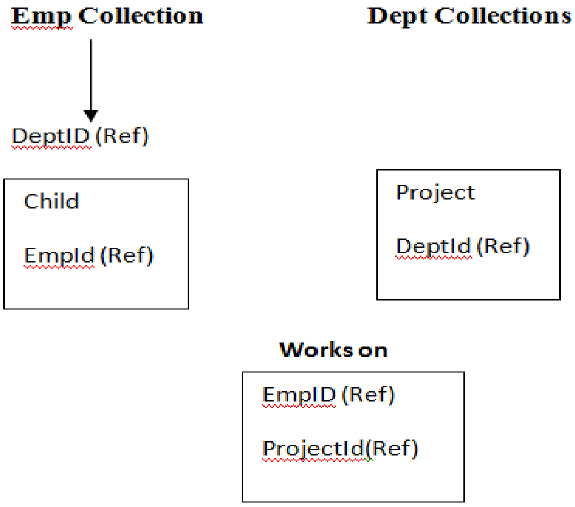

Fig. 4 MongoDB model Structure 3

III. Experiments

Here, in this part, we specify the fetch time of executing Queries. So initially, we perform setup of our experiments. Then we proposed the results of all structure. Finally we evaluate and discuss it.

Query 1:

```
Select * from EMP e, WorksOn w, Project p, Dept d, Child
c
Where d.dname='dt90' and
e.deptid=d.deptid and
c.empid=e.empid and
w.empid=e.empid and
w.projid=p.projid and
p.deptid=e.deptid;
```

Query 2:

```
Select * from EMP e, Dept d, WorksOn w
Where d.dname=’d1’ and
w.projid=’125’ and
w.empid=e.empid and
e.deptid=d.deptid;
```

A. *Results and Analysis*

Fig. 5 presents the fetching time of running Query 1. MySQL database achieved better retrieval time compared with all MongoDB structures. Recall that Query 1 retrieves all the information about one employee. Fetch times of MonogDB structure 1 (M1) and structure 2 (M2) are very close while structure 3 (M3) took 28 hours because MongoDB structure 3 uses reference to lookup data.

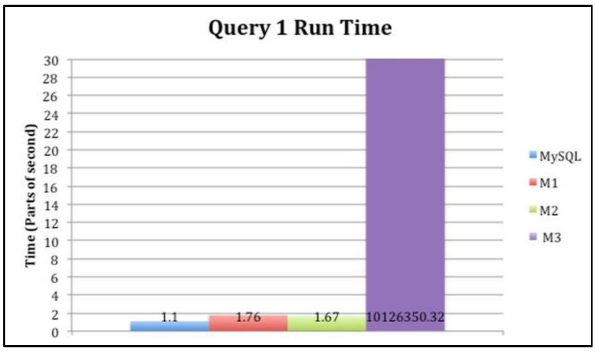

Fig. 5 MongoDB model Structure 3

IV. CONCLUSIONS

In this paper, we have performed analytical evaluation between different NoSQL structures to characterize Relational DBs. We have used Non-relational structure (MongoDB) for NoSQL and Relational DBs like MySQL as a Relational database. Entire experiments are based on three non-relational structures including embedding docs, referencing docs and both. We have set 2 complex queries to measure retrieval time accordingly. Our Experimental Database contains different tables with variety of relationships like, one-to-many and many-to-many.

References

[1] T. Jia, X. Zhao, Z. Wang, D. Gong and G. Ding, "Model Transformation and Data Migration from Relational Database to MongoDB," 2016 IEEE International Congress on Big Data (BigData Congress), San Francisco, CA, 2016, pp. 60-67.

[2] G. Zhao, L. Li, Z. Li and Q. Lin, "Multiple Nested Schema of HBase for Migration from SQL," 2014 Ninth International Conference on P2P, Parallel, Grid, Cloud and Internet Computing, Guangdong, 2014, pp. 338-343.

[3] Z. Wei-ping, L. Ming-xin and C. Huan, "Using MongoDB to implement textbook management system instead of MySQL," 2011 IEEE 3rd International Conference on Communication Software and Networks, Xi'an, 2011, pp. 303-305.

[4] J. S. van der Veen, B. van der Waaij and R. J. Meijer, "Sensor Data Storage Performance: SQL or NoSQL, Physical or Virtual," 2012 IEEE Fifth International Conference on Cloud Computing, Honolulu, HI, 2012, pp. 431- 438.

[5] M. G. Jung, S. A. Youn, J. Bae and Y. L. Choi, "A Study on Data Input and Output Performance Comparison of MongoDB and PostgreSQL in the Big Data Environment," 2015 8th International Conference on Database Theoryand Application (DTA), Jeju, 2015, pp. 14-17.

[6] S. Chickerur, A. Goudar and A. Kinnerkar, "Comparison of Relational Database with Document-Oriented Database (MongoDB) for Big Data Applications," 2015 8th International Conference on Advanced Software Engineering & Its Applications (ASEA), Jeju, 2015, pp. 41-47.

[7] S. H. Aboutorabi, M. Rezapour, M. Moradi and N. Ghadiri, "Performance evaluation of SQL and MongoDB databases for big e-commerce data," 2015 International Symposium on Computer Science and Software Engineering (CSSE), Tabriz, 2015, pp. 1-7.

[8] A. Goyal, A. Swaminathan, R. Pande and V. Attar, "Cross platform (RDBMS to NoSQL) database validation tool using bloom filter," 2016 International Conference on Recent Trends in Information Technology (ICRTIT), Chennai, 2016, pp. 1-5

[9] A. Goyal, A. Swaminathan, R. Pande and V. Attar, "Crosplatform (RDBMS to NoSQL) database validation tool usingbloom filter," 2016 International Conference on Recent Trends in Information Technology (ICRTIT), Chennai, 2016, pp. 1-5.

[10]. J. Kepner et al., "Associative array model of SQL, NoSQL, and NewSQL databases," 2016 IEEE High Performance Extreme Computing Conference (HPEC), Waltham, MA, 2016, pp. 1-9.

[11]. P. Sareen and P. Kuma, “NoSQL Database and its Comparison with SQL Database”. International Journal of Computer Science & Communication Networks, Vol: 5 No: 5, pp 293-298, 2011.

[12]. G. Zhao, Q. Lin, L. Li and Z. Li, "Schema Conversion Model of SQL Database to NoSQL," 2014 Ninth International Conference on P2P, Parallel, Grid, Cloud and Internet Computing, Guangdong, 2014, pp. 355-362.

[13]. Lv, Q., & Xie, W, “A Real-Time Log Analyzer Based on MongoDB. In Applied Mechanics and Materials, Trans Tech Publications , Vol. 571, pp. 497-501, 2014.

[14]. R. Lawrence, "Integration and Virtualization of Relational SQL and NoSQL Systems Including MySQL and MongoDB," 2014 International Conference on Computational Science and Computational Intelligence, Las Vegas, NV, 2014, pp. 285-290.

[15]. C. Győrödi, R. Győrödi, G. Pecherle and A. Olah, "A comparative study: MongoDB vs. MySQL," 2015 13th International Conference on Engineering of Modern Electric Systems (EMES), Oradea, 2015, pp. 1-6.