



GUI for Data Migration and Query Conversion

Manisha B. Jadhav*, Prof. R. R. Badre

Department of Computer Engineering, MITAOE Alandi,
India

Abstract— With the appearance of Internet related advances, part of changes were seen in the information gathered, put away what's more, handled by the associations. The new approaching information was unstructured, accessible from various outside sources and amassed in high volumes inside a little time traverse. It was understood that the customary framework is wasteful to deal with this information. Answer for that is NoSQL database, NoSQL Databases are exceptionally adaptable, effective and productive. Be that as it may, the issue with these databases is that they don't utilize the customary SQL Organized Query dialect for question the information. Thus, the capacity to Migrate information from existing customary databases to these NoSQL databases and having the capacity to create proficient an enhanced capacities to question the information under these NoSQL database. A number of devices are accessible which do have the capacities to address this, yet none of them really gives a total incorporated framework for effectively doing this. This paper proposes a design which will give a GUI based Web Application that will give the clients a tick and utilize Querying include with bother free Data Migration and Automatic Query Transformation. MongoDB being the most drifting NoSQL database, this proposed framework concentrates on these viewpoints with reference to MongoDB.

Keywords— MongoDB, Migration, Query Syntax, RDBMS, Relational Model, Schema

I. INTRODUCTION

MongoDB is a Document Store NoSQL Database that gives High Availability and Automatic Scaling as the real components. It stores information in JSON organize and gives JavaScript capacities to question this information. The grammar also as the information design being totally unique, it presents the overhead of getting productivity in JavaScript questioning also, changing over questions from all Web Programs from SQL to MongoDB. The proposed framework concentrates on 2 noteworthy angles of the database world Composing enhanced inquiries should be possible just by people who have been dealing with databases from quite a while and have built up a feeling of incredibleness in the inquiry dialect. This unquestionably turns out to be more troublesome, if the reference database does not bolster the customary SQL for questioning the information. In instance of MongoDB, JavaScript capacities are utilized for questioning the information and in addition a great deal of new capacities are created once in a while by the MongoDB people group. These new capacities without a doubt help intense questioning of information yet for just the individuals who are knowledgeable with this database. Anticipating my whole heap of Application Developers to be productive in Upgraded MongoDB questioning is not a perfect circumstance. So this paper proposes a GUI based Web Application that will annihilate the need of knowing sentence structure and accessible strategy. In addition the proposed framework will likewise hold Automatic Question Correction utility which we realize an Error Free condition in the database world. Another imperative concentration zone that holds organizations from changing to intense databases like MongoDB is the overhead included in the Migration process. Migration is the procedure of exchanging either information from a solitary table or from the whole database starting with one server then onto the next or from one database engineering to another. There are numerous calculations accessible which productively change over the database mapping from SQL to MongoDB and server the reason for Migration. In any case, the genuine issue that holds up this Migration is changing over questions display in the projects that drive the applications. For an expansive scale organization which would really consider Migrating from SQL to MongoDB, this procedure includes a great deal of speculation as far as labor and time as they have a substantial number of programs with part of questions that guide their applications. With a goal to address this issue, the proposed framework goes for giving a stage where the questions will be changed over on the fly from SQL norms to MongoDB measures. This utility will be facilitated on the web server where the customer hits some time recently focusing on the database and the SQL question from the program will be changed over here and will be let go upon the basic MongoDB database.

A. MySQL Database

A database is a separate application that stores a collection of data. Each database has one or more distinct APIs for creating, accessing, managing, searching and replicating the data it holds. Other kinds of data stores can be used, such as files on the file system or large hash tables in memory but data fetching and writing would not be so fast and easy with those types of systems. So nowadays, we use relational database management systems to store and manage huge volume

of data. This is called a relational database because all the data is stored into different tables and relations are established using primary keys or other keys known as foreign keys.

1) *A Relational DataBase Management System is a software that:*

- Enables you to implement a database with tables, columns and indexes.
- Guarantees the Referential Integrity between rows of various tables.
- Updates the indexes automatically.
- Interprets an SQL query and combines information from various tables.

2) *RDBMS Terminology :*

Before we proceed to explain the MySQL database system, let's revise few definitions related to database.

- Database: A database is a collection of tables, with related data.
- Table: A table is a matrix with data. A table in a database looks like a simple spreadsheet.
- Column: One column data contains data of one and the same kind, for example the column postcode.
- Row: A row is a group of related data, for example the date of one subscription.
- Redundancy: Storing data twice, redundantly to make the system faster.
- Primary Key: A primary key is unique. A key value can not occur twice in one table. With a key, you can find at most one row.
- Foreign Key: A foreign key is the linking pin between two tables.
- Compound Key: A compound key is a key that consists of multiple columns, because one column is not sufficiently unique.
- Index: An index in a database resembles an index at the back of a book.
- Referential Integrity: Referential Integrity is used for the confirmation about the foreign key value always points to an existing row.

3) *MySQL as a Database:*

MySQL is a fast, easy-to-use RDBMS being used for many small and big businesses. MySQL is produced, showcased, and bolstered by MySQL AB, which is a Swedish organization. MySQL is ending up plainly so prevalent due to numerous great reasons: that MongoDB is especially appropriate to handle.

- MySQL utilizes a standard type of the outstanding SQL information dialect.
- MySQL is a capable program in its own privilege. It handles a vast subset of the usefulness of the most costly and capable database bundle.
- MySQL takes a shot at many working frameworks and with numerous dialects, including PHP, PERL, C, JAVA, and so forth.
- MySQL works rapidly and functions admirably even with vast information collections.
- MySQL is amicable to PHP, the most refreshing dialect for web improvement.
- MySQL underpins extensive databases, up to 50 million columns or more in a table. The default document estimate confine for a table is 4GB, however you can expand this to a hypothetical cutoff of 8 million terabytes.
- MySQL is adaptable. The open-source, GPL permit enables developers to alter the MySQL programming to fit their own particular situations.
- MySQL is customizable. The open-source ,GPL license allows programmers to modify the MySQL software to fit their own specific environments.

B. Migration Process

In order to implement the data migration, it is essential to standardize the source data and classify the data according to the data type. In my project all the functionality of mysql like create, delete, update ,process gets occurs the same can be done automatically only by making changes in mysql and then clicking button MIGRATION, this button just change the which we had changed in MySQL to MangoDB. The research demonstrates the migration of data from MongoDB to Azure Table columnar- intermediate target database and Azure Table to Neo4j graph-based target database, importantly, a direct data translation from MongoDB to Neo4j i.e. document to graph database. According to Kanade, Gopal and Kanade, MongoDB is an open source, and document based schema free database which models the data in BSON, also, it exhibits the features such as auto-sharding, map-reduce, hybrid deployments in the cloud. To appreciate the migration framework, the algorithm models two main strategies: BSON Parsing and Meta-Modeling in order to standardize and classify the data from MongoDB. In this section, the research entails the different APIs and various methods defined to implement the aforementioned strategies for efficient data migration.

C. MongoDB

NoSQL databases have risen to the occasion when it comes to unstructured modern data and the kind of real-time analytics we expect from it. Depending on your own unique business needs, MongoDB might be just the right solution for you. In this article, we'll cover the nuts and bolts of NoSQL databases and a portion of the difficulties of current enormous information examination that MongoDB is especially appropriate to handle. How MongoDB Is Changing the Business of Big Data.

II. RELATED WORK

As per "Girts Karnitis" and "Guntis Arnicans" in [1] Data Migration is a mix of 2 stages, wherein we rebuild the information from the source to meet the particulars of the objective framework and also start the exchange from the source to the goal. A few strategies, including however not limited to Schema Conversion, meta-displaying, ETL what's more, robotized information movement approach manage these means. There are 2 levels in which information is accessible in an RDBMS viz. Physical Level and Logical Level. Distinctive tree building calculations are utilized to characterize the legitimate chain of command that exists in the information. "Gansen Zhao", "Weichai Huang", "Shunlin Liang" and "Yong Tang" in [2] characterize MongoDB into a Relational Model. The Authors characterize two levels of characterizing the composition. One of which is the Micro Level in which the Schema of MongoDB Gathering can be thought to be settled and the other one is the Macro Level in the Schema of MongoDB Collection would differ for any two given occurrences of time. So on the off chance that we need to characterize the Schema of a MongoDB Collection, then considering the Micro Level we can express that the Relational Model of MongoDB .

"Thalheim" and "Wang" [3] express that keeping in mind the end goal to relocate the information, one needs an intensive comprehension of the information source, for example, information accessibility and information imperative since various information sources are composed utilizing diverse displaying semantics. MongoDB Collection maps to Table in the Relational Show. The Schema is made out of keys present in the archive the would characterize the structure of all tuples. Each esteem will compare to the estimation of a key which will be invalid if the key is absent in a record. There can likewise exist Parent Child relations between tables which can be distinguished by the Primary and Foreign Keys which will characterize the Keys in the Main Document and the Keys for the Sub Reports [4].

F. Matthes and C.Schulz in [5] state that "Tool-supported one-time process which aims at migrating formatted data from a source structure to a target data structure, whereas both structures differ on a conceptual and/or physical level".

Mongify is an tool that empowers information interpretation from SQL databases to MongoDB. It gives incorporation MySQL, PostgreSQL, SQLite, Oracle, SQL Server and DB2. It works well with all adaptations of MongoDB. MongoBooster is a Cross Platform GUI apparatus for MongoDB chipping away at a shell driven stage giving set up updates also, in corporation with Moment.js.[7] It underpins the ES6 language structure giving a genuine insight encounter. It offers help for mongoose like familiar question manufacturer API. Programming interface which empowers you fabricating inquiries utilizing anchoring language structures rather than straightforward JSON Objects. Composing more succinct and intelligible MongoDB scripts turn out to be simple with the inherent support for square variable perusing, bolt capacities and layout strings.

RoboMongo [6] does not imitate the shell of MongoDB Or maybe it basically inserts a similar domain and motor accessible with the Mongo Shell. It is as of now backings the MongoDB adaptation 3.2. RoboMongo executes the code in an inside VM in view of JavaScript as opposed to just investigate the semantics of the code, giving the client an auto finish include at runtime, including highlights that can't be acquired generally utilizing the static techniques.

NoSQL viewer stands determined be a free GUI based client for NoSQL databases like MongoDB, Cassandra, Couch base, Couch DB and Hbase. It enables users to simultaneously perform CRUD operations from one single platform, eliminating the overhead of using multiple tools for different databases. It provides easy yet powerful, gassed-up migration functionality between any supported Big Data databases.

MongoDB Connector for Hadoop enables the integration between the 2 most powerful data storage systems from the OLTP and OLAP sectors. It allows the ability to use MongoDB and Hadoop as sources and destinations for data transfers.

Database Master is easy to use database querying, administration and management tolls that provide a consistent user interface with modern styling and interface. Is simplifies the process of managing, monitoring, querying, editing, visualizing and designing relational and NoSQL database systems. It allows users to execute extended scripts in SQL, JSON and C Sharp providing all database objects like tables, views, procedures, packages, columns, indexes and triggers.

1) *Limitations of the previous systems:*

- Separate tools have to be used for all 3 applications
- Most of them are not web based.
- No GUI actually supports building queries without writing syntaxes
- No Auto Error Correction Features
- No GUI based Data Migration
- Data Migration increases the overhead of transforming queries from Web Programs and connectors
- No Universal Platform

III. PROPOSED WORK

The proposed system can be separated into a Web based database client application and a server hosted query conversion utility program.

The Web based Database Client will server as an IDE for the application developers to produce hassle free, optimized queries in MongoDB without the need to remember the syntax or without even being acquainted with the latest methods introduced or changes in different modules of the MongoDB query language. This IDE will provide a symbolic representation of available methods at different levels in the process of querying the database. For this purpose, the entire querying process is assumed to be a hierarchy starting from databases and lowering down to Collections, Documents,

Criteria and Projections. At each hierarchical level all but only the possible available methods will be displayed to the users. This display will also consider the structure and data types against all keys involved in the query. At any given point of time the user will have the option to extend, revise or execute his query. The proposed system is expected to be intelligent enough in understanding the possible errors in the user generated query and smart enough to auto correct it before execution. All changes made internally by the IDE will be logged under the history section for user references.

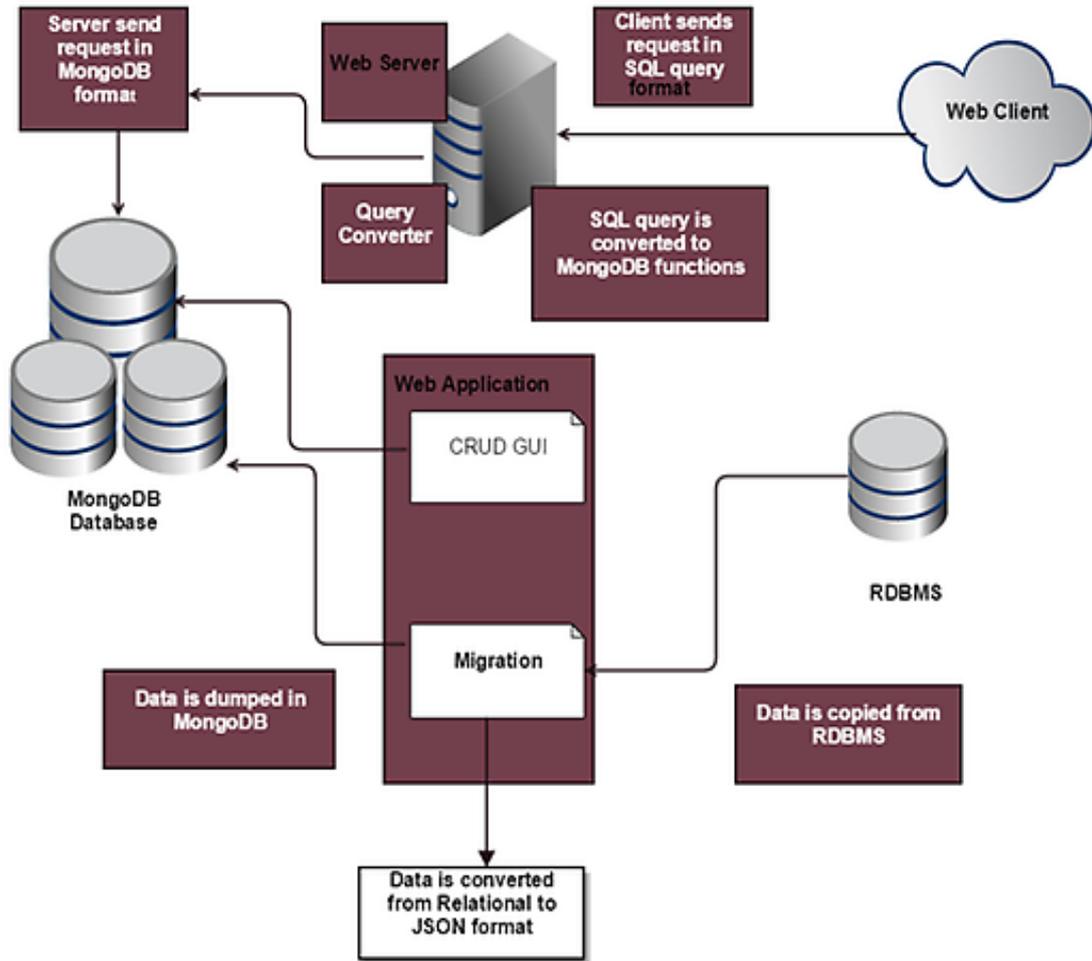


Fig. 1: System Architecture

The second component of the proposed system i.e. the query conversion utility will be hosted on the web server where the client applications hit, before connecting to the database. This utility will consume some time for redirecting the query operations for the first time when a certain query is hit. This delay will be the time required by the utility to convert the SQL query into equivalent MongoDB function. This converted query will be stored in the hosting server, for faster performance when the same query is executed the next time.

A. Process flow of query conversion

Following figures show How the query conversion is done, Firstly maintain a master table which will be the first hit contact of all queries. This table will have a mapping of the primary key tables and their dependent tables are drawn into which collections. Capture the constraints and projections in the query and your MongoDB query is ready. This will get executed on the collection identified by the MongoDB master table.

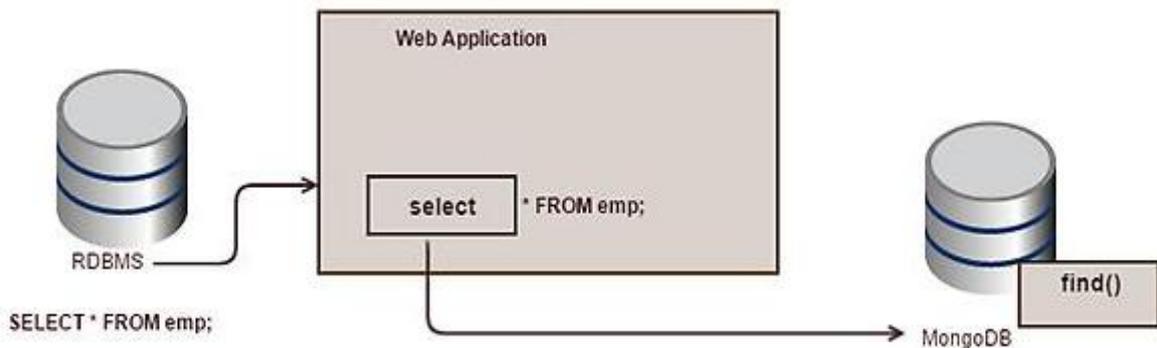


Fig. 2: step1

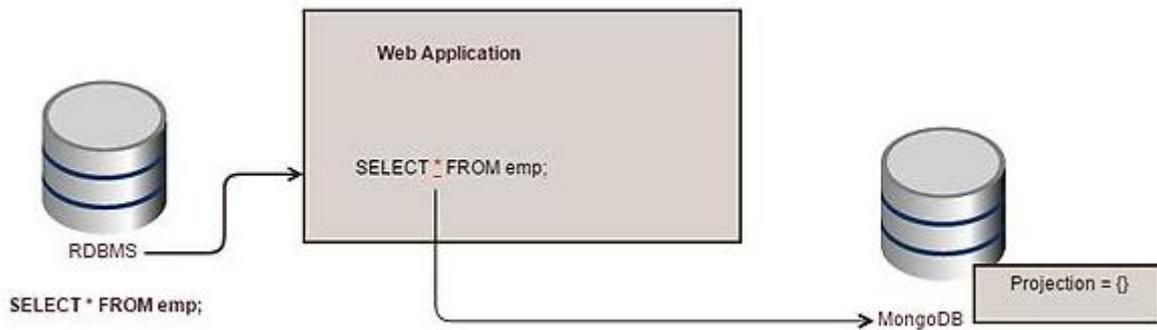


Fig. 3: step2

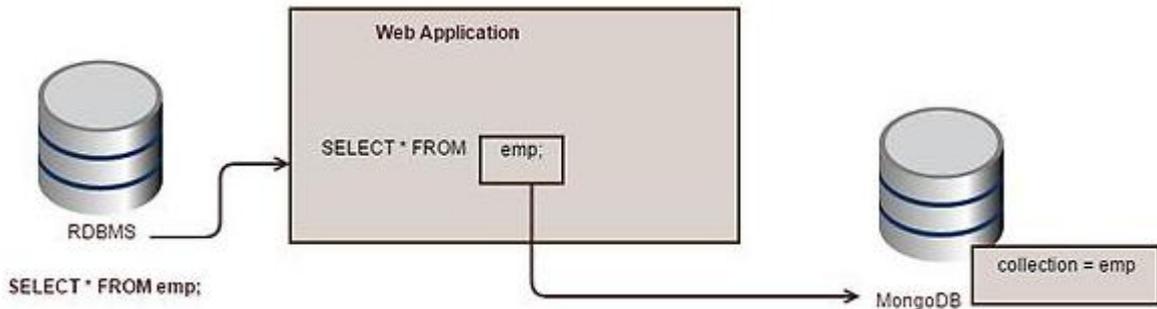


Fig. 4: step3

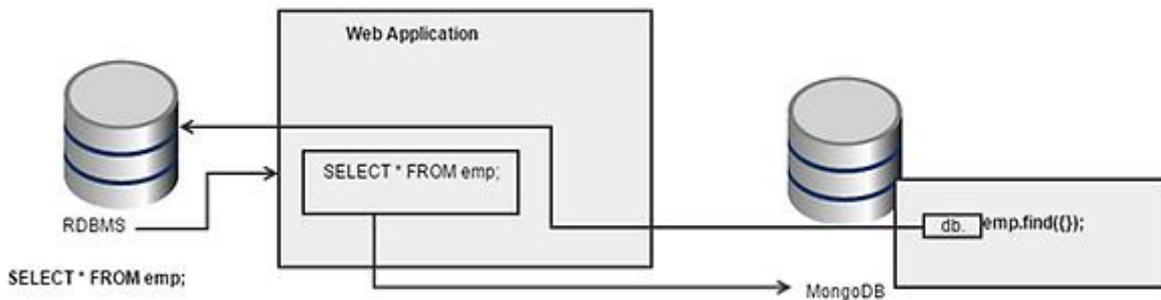


Fig. 5: step4

IV. PROPOSED ALGORITHM

A. Query Conversion Algorithm

Step 1: \$query = SELECT name from employee where salary >300 ;

Step 2: Load query into PHP SQL Parser Require_once (php"- "sql"- "parser.php); \$parser = new PHPSQLParser (\$query,true); Print \$parser

Step 3: Fetch Required Elements \$function= [SELECT]

\$projection = [SELECT] ->[0] ->[base_expr] =>name \$table = [FROM] ->[0] ->[table] =>employee \$condition[0] =[WHERE]

\$condition[1] =[WHERE] ->[0] ->[base_expr] =>salary \$condition[2] = [WHERE] ->[1] ->[base_expr] =>>

\$condition[3] =[WHERE] ->[0] ->[base_expr] =>300 Step 4: SQL to MongoDB Element Mapping

FETCH Collection collection =\$table FETCH CONDITION

IF \$condition[2] NOT EQUAL TO =

SELECT ->find(f\$condition[1]: f\$condition[2]: \$condition[3]gg)ELS SELECT >find(f\$condition[1]: \$condition[3]g)

FetchProjection

f\$projection : 1g

Step5 : FinalMongoDBQuery

db:employee:find(fsalary: f\$gt: 300gg; fname: 1g)

V. CONCLUSION

Building a tool that will reduce the pressure from the Developers to understand the syntax and will help building better queries. It will enable quick Migration activities as the tedious load of query translation. Implementing the GUI tool for CRUD operations which can be further extended to support higher hierarchies.

ACKNOWLEDGMENT

We are thankful to the Savitibai Phule University of Pune for theoretical and practical teaching in the design of this work.Finally, we thank the Prof.R.R.Badre for providing its experts for practical guidance for the present work.

REFERENCES

- [1] Girts Karnitis and Guntis Arnicans *Migration of Relational Database to Document-Oriented Database: Structure Denormalization and Data Transformation. 7th International Conference on Computational Intelligence, Communication Systems and Networks (CICSyN)*. No.6pp.113-118,2015
- [2] Gansen Zhao, Weichai Huang, Shunlin Liang, Yong Tang *Modelling MongoDB with Relational Model*. 2013 Fourth International Conference on *Emerging Intelligent Data and Web Technologies*. No.6.pp.115-121,2013
- [3] B. Thalhein and Q. Wang *Data Migration: A theoretical perspective. Data and Knowledge Engineering*, vol. 87, pp. 260-278, 2013.
- [4] Aryan Bansel Horacio GonzalezV elez Adriana E. Chis *Cloud-based NoSQL Data Migration*. 2016 *24th Euromicro International Conference on Parallel, Distributed, and Network-Based Processing*. No.7pp.224-231,2016.
- [5] F. Matthes and C. Schulz *Towards an integrated data migration process model. Software Engineering for Business Information Systems (sebis)*, 2011.
- [6] [Online]. Available: www.robomongo.org/.
- [7] [Online]. Available: www.mongobooster.com/.
- [8] [Online]. Available: <https://docs.mongodb.com/>.
- [9] [Online]. Available: <http://www.rubydoc.info/gems/mongify/>
- [10] D. Lee, M. Mani, F.Chiu, and W. Chu *translating relational schemas to XML schemas using semantic constraints. proceedings of the 11th CIKM*, 2002, pp.282-291.