ISSN 2348-1196 (print)
International Journal of Computer Science and Information Technology Research ISSN 2348-120X (online)
Vol. 3, Issue 1, pp: (317-321), Month: January - March 2015, Available at: www.researchpublish.com

SQL2MongoDB

[1]Diksha Koul, [2]Devayani Pawar, [3]Radhika Ranade, [4]Vishakha Patil

[1,2,3,4] Savitribai Phule Pune University, K.K.W.I.E.E.R., Nasik, India

Abstract: **For the last two decades, relational databases proved to be more powerful and superior because of features like clear semantics and easy to use .Structural data management support is provided by relational databases. However, in few past years development in IT sector put forward the concept of big data, which deals with extreme large volume and variety in data type and structures. To handle such big data, relational databases are not suitable because of strict constraints on data structure and data relations, and so on. On the other side, MongoDB is getting popular because of its capability in dealing with big data and complex structures. Currently there is no tool for the migration of relational databases to Mongo databases. This migration requires conversion of relational database query like SQL to NoSQL database query like MongoDB query. The modelling allows handling NoSQL data with structured query processing language.**

Keywords: **Algorithms, SQL, MongoDB, NoSQL, big data, Standardization, XML, JSON.**

I. INTRODUCTION

Relational data model proposed by E.F.Codd act as basis for modern relational databases. The relational model is based on relational algebra and relational calculus, with the primary operators like union, intersection, selection, projection and join. Relational databases present a view of simple and unified data model. Relations acts as abstraction of entities, and the association and connection between entities are also abstracted as relations. All operations on the entities and their connections are performed on the corresponding relations. Tables and views are representation of relations in relational databases, with tables as physical representation of relations and views as virtual representation of relations. Entity is a row in a table and set of entities represent a table. Properties of the entities corresponding to each row are represented by columns. Structural Query Language (SQL) is used for the data retrieval and management of structural data. SQL encourage the development and deployment of relational databases. SQL[1] acts as an essential part of the databases relational databases. Because of the higher demand in various sectors, the relational database system could not fully meet the needs of the society. It is seen that the efficiency of the relational database system is quite low with a huge amount of data. Relational database system does not provide a good amount of scalability, making the system inconvenient for rapidly increasing data. However, some features like high maintenance cost[2] of the relational database system could not satisfy user needs. In few past years, a lot of new database systems integrated with new technologies have come out. These database systems had different models, like Key-value stores model, Table-oriented model, Document-oriented model, Graph-oriented model, etc. Most of these systems rely on models other than relational algebra. Generally, these database systems are called NoSQL database systems.

II. RELATED WORK

Robert J. Moore in association with RJMetrics[3] worked on the translation of MySQL to mongodb. The "query object" system is the first piece of work. This system represents relational database queries as objects. These objects allow to represent most SQL-style queries independent of the platform on which they will ultimately be run. In other words, by using code the same exact query object could be built and then extraction of data from MySQL server can be done. Obviously, these query objects require a "query renderer" to translate them into the SQL language of your choice. The rendering system is pretty robust for the SQL platforms, but was extremely weak for MongoDB (only the simplest SELECT queries would work.

ISSN 2348-1196 (print)
International Journal of Computer Science and Information Technology Research ISSN 2348-120X (online)
Vol. 3, Issue 1, pp: (317-321), Month: January - March 2015, Available at: www.researchpublish.com

A. The missing pieces:

The translator only supports SELECT queries, do not have support for JOINs across multiple tables and would have limited support for complex WHERE clauses including embedded parentheses. The query does not renderer's support of MongoDB queries to include support for query properties such as:

- Aggregate functions like MAX(), MIN(),AVG(), COUNT() and SUM().
- WHERE clauses, including IN statements and the OR operator
- GROUP BY clauses
- ORDER BY clauses
- LIMIT clauses

III. ARCHITECTURE DIAGRAM

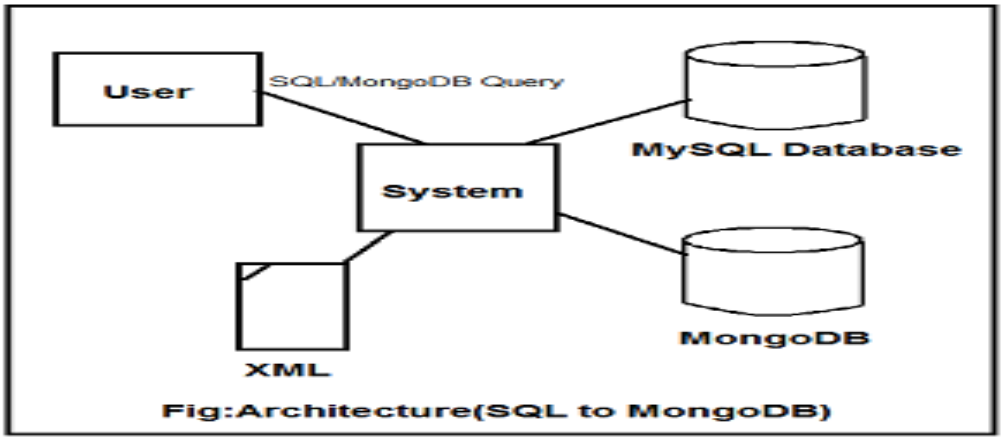

Fig. 1. Architecture of SQL to Mongo converter

IV. CONVERSION OF SQL TO MONGO

MongoDB is an open source document-oriented NoSQL database which stores data in the form of JSON-like objects. Each database in MongoDB consists of collections which are equivalent to an RDBMS database consisting of SQL tables. Tables stores data in rows which is equivalent to collection which stores data in the form of documents. A document has a JSON-like structure (known as BSON in MongoDB) while relational databases stores data in its set of columns. So for mapping relational to non relational database, XML is converted into JSON format.

A. Converting XML to JSON:

XML is a document-oriented format, while JSON is a structured data format. Xml uses elements ,attributes,and content text,while JSON uses unordered collections of name/value pairs amd arrays of values.

B. A Pragmatic Approach:

A single structured XML element might come in seven[4] flavors:

1. an empty element
2. an element with pure text content
3. an empty element with attributes
4. an element with pure text content and attributes
5. an element containing elements with different names
6. an element containing elements with identical names
7. an element containing elements and contiguous text

The following table shows conversion patterns between XML and JSON.

ISSN 2348-1196 (print)
International Journal of Computer Science and Information Technology Research ISSN 2348-120X (online)
Vol. 3, Issue 1, pp: (317-321), Month: January - March 2015, Available at: www.researchpublish.com

Pattern	XML	JSON
1	<e/>	"e": null
2	<e>text</e>	"e": "text"
3	<e name="value" />	"e":{"@name": "value"}
4	<e name="value">text</e>	"e": { "@name": "value", "#text": "text" }
5	<e> <a>text text </e>	"e": { "a": "text", "b": "text" }
6	<e> <a>text <a>text </e>	"e": { "a": ["text", "text"] }
7	<e> text <a>text </e>	"e": { "#text": "text", "a": "text" }

When the XML [5] tag structure has no duplicate tags, then collection of name/value pairs can be used. Both JSON and XML structure can be abstracted as a multi-branch tree structure. For example: Let we have the following XML document:

```
<Employees>
 <Employee>
  <name> Abd El-Aziz </name>
   <age> 30 </age>
    <phone> 9791126517 </phone>
     <salary> 3000$ </salary>
 </Employee>
</Employees>
```

Fig. 2. XML Document without Duplicate Tags

Then the corresponding JSON format is :

```
{
”Employees”:
{
”Employee”:
{
”name”: ”Abd El-Aziz”,
”age”: 30,
”phone”: ”9791126517”
”salary”: ”3000 $”
                    }
}
}
```

Fig. 3. The Corresponding name/value JSON format columns.

When XML tag structure contain duplicate tags, then an ordered collection of values (an array) is used. In JSON, data objects can also be abstracted into multi-branch tree structure. Tags that are repeated can be combined into an array with an integer subscript starting from 0 and is hidden. For example:

Let we have the following XML document:

```
<Employees>
<Employee>
<name> Abd El-Aziz </name>
<age> 30 </age>
<phone> 9791126517 </phone>
</Employee>
<Employee>
<name> Omar </name>
<age> 20 </age>
<phone> 9792226517 </phone>
</Employee>
</Employees>
```

Fig: 4. XML Document has Duplicate Tags

ISSN 2348-1196 (print)
International Journal of Computer Science and Information Technology Research ISSN 2348-120X (online)
Vol. 3, Issue 1, pp: (317-321), Month: January - March 2015, Available at: www.researchpublish.com

Then the corresponding JSON format is :

```
{
”Employees”:
{
”Employee”:
[
{
”name”: ”Abd El-Aziz”,
”age”: 30,
”phone”: ”9791126517”
},
{
”name”: ”Omar”,
”age”: 20,
”phone”: ”9792226517”
}
]
}
}
```

Fig. 5. The Corresponding ordered list JSON format

C. Mapping Relational Databases (SQL) to Mongo DB:

Each database in MongoDB consists of collections which are equivalent to an RDBMS database consisting of SQL tables. Tables store data in rows which is equivalent to collection which stores data in the form of documents. A document has a JSON-like structure (known as BSON in MongoDB) while relational databases stores data in its set of columns.

To understand the mappings better, consider an example of an SQL table 'users' and its corresponding structure in MongoDB. As shown in figure, each row in the SQL table transforms to a document and each column to a field in MongoDB.

V. CONCLUSION

MongoBD is a schema less database having deep query ability. It uses internal memory for storing the (windowed) working set, enabling faster access of data. Conversion / mapping of application objects to database objects is not needed in MongoDB. It uses JSON format for data storage. Relative to the XML, JSON is easy for machines to parse and generate. A text format is used by JSON that is completely independent of the language. Compared to XML, with the above characteristics, JSON has obviously higher parsing efficiency and easy preparation advantages. Therefore, we presented how to translate query from SQL to MongoBD using XML document to JSON format.

REFERENCES

[1] Ramon Lawrence, "Integration and virtualization of relational SQL and NoSQL systems including MySQL and MongoDB", CSCI International Conference, Vol. 1, pp. 285 – 290, March 2014.

[2] Jing Liu, Xin Dong, and Alon Halevy, "*Answering Structured Queries on Unstructured Data*" , June 2006.

[3] Robert J Moore, "Query converter tool", in querymongo.com

[4] Stefan Goessner, "Converting Between XML and JSON "xml.org, May 2006.

[5] A. A. Abd El-Aziz and A. Kannan, "JSON signature," in ResearchGate.

[6] "SQL to MongoDB mapping chart",mongodb.org.