International Research Journal of Engineering and Technology (IRJET) e-ISSN: 2395-0056
Volume: 08 Issue: 03 | Mar 2021 www.irjet.net p-ISSN: 2395-0072

Preview Design and Development of Schema for Schemaless Databases

Ashwini Mandale[1] , Dr. Neeraj Sharma[2]

[1]*Research Scholar, Department of Computer Science and Engineering, Sri Satya Sai University of Technology and Medical Sciences, MP, India*
[2]*Associate Professor, Department of Computer Science and Engineering, Sri Satya Sai University of Technology and Medical Sciences, MP, India*

Abstract - *Historically, relational databases have been the general purpose go-to database of choice. But as the data grows and as the time went the structure of data has change dramatically. In order to satisfy this requirement there need to be a database that will evolve automatically as the schema of data changes and also can handle large quantity data (Big Data). As there are tools exist to assist with schema design for relational databases and absence of tools to assist with schemaless databases hence need to design and develop schema for schemaless database. Here we are designing and developing schema for schemaless databases such that such databases can be handled vey effectively and gives high performance*

***Key Words*: NoSQL, Schema, Schemaless databases**

1. INTRODUCTION

Now days the use of NoSQL databases has grown to manage unstructured data for applications to ensure performance and scalability. But still many organizations prefer to transfer data from an operational NoSQL database to a SQL-based relational database for using existing tools for machine learning, analytics, business intelligence, decision making and reporting. The existing methods of NoSQL to relational database transformation require manual schema mapping, which requires domain expertise and consumes noticeable time. Therefore, an efficient and automatic method is needed to transform an unstructured NoSQL database into a structured.

A Schema-Less Database:

1. Does not require to follow the strict rules of schema (rigid schema) (database, schema, data types, tables etc.) that one is required to live up to through the life of a system.
2. Individual values pertaining to one single column type does not enforce data type limitations.
3. It does not model a database schema, application or product instead it models the business usage.
4. Unstructured and structured data can be stored in schemaless databases.
5. There is no need to introduce additional layers for abstracting the relational model and exposing it in an object oriented format.

Due to these dynamic objects schema migrations becomes very easy. In a traditional RDBMS [1, 8], data migration scripts comes along with releases of code. Further, in case where a rollback is necessary each release should have a reverse migration script. ALTER TABLE operations can be very slow and result in scheduled downtime in RDBMS, with a schemaless database [4] this can be done very easily. As in schemaless databases there no rigid schema, 90% of the time adjustments to the database become transparent and automatic.

A brief review of the work already done in the field.

Schema database [15]: A database schema is its structure; a set of integrity constraints imposed on a database. These integrity constraints ensure compatibility between parts of the schema [6]

Limitations of sql/schema databases:

- Object/relational impedance mismatch
- Complicated to map rich domain model to relational schema
- Difficult to handle semi-structured data, e.g. varying attributes
- Schema changes
- Extremely difficult/impossible to scale
- Poor performance for some use cases

The term NoSQL was first used in 1998 for a relational database that omitted the use of SQL [9]

Schemaless: simple replication, high availability, horizontal scaling, and new query methods. These options are collectively known as schemaless or NoSQL [1]

Advantages of NoSQL:

- Higher performance
- Higher scalability
- Richer data-model
- Schema-less

Limitations / drawbacks:

- Limited transactions
- Relaxed consistency
- Unconstrained data

Types of non-relational databases [7]:

One relational database (Postgres), two key-value stores (Riak, Redis), a col-umn-oriented database (HBase), two

document-oriented databases (MongoDB [2], CouchDB), and a graph database (Neo4J)

1.1 Literature Survey

1.Inferring Versioned Schemas from NoSQL Databases and its Applications, Diego Sevilla Ruiz, Severino Feliciano Morales, and Jesús García Molina, Conference Paper · October 2015 DOI: 10.1007/978-3-319-25264-3_35

This article proposes, for inferring the schema of aggregate-oriented NoSQL databases a MDE-based reverse engineering approach is presented. Meta modelling and model transformations of Model-Driven Engineering (MDE) techniques, have been used to implement both the schema inference strategy and the applications, in order to take advantage of the abstraction and automation they provide .Two offerings of this work are first it infers conceptual QL databases discovering all the versions of the inferred entities and their relationships second it show how the inferred schemas can be used to automatically generate different software artifacts, which help to improve the productivity and code quality.

2. Method of Tracking Schema in Schemaless Databases, Ron J Mann, US20150095298A1, 2apr 2, 2015

Systems and methods are described for obtaining for insertion into schemaless database, a data object that comprises a plurality of key and value pairs. The method also includes hashing the keys associated with the plurality of key and value pairs. The hashing includes executing a hash function to generate a hashed data object [12].The method also includes data objects and determining that the hashed data objects does not match any of the first hashed data objects based on the determining that the method includes associating the hashed data object with first hashed data object and generating a schema for the database. The schema includes a hierarchy of keys that represent the second hashed data objects.

3. Automated Schema Design for NoSQL Databases, Michael J. Mior, University of Waterloo [20]
This paper explores problem of schema design in NoSQL databases for optimizing query performance by minimizing storage overhead. This approach uses the cost of executing a given workload for a given schema to guide the mapping from the application data model to a physical schema.

4.Automatic NoSQL to Relational Database Transformation with Dynamic Schema Mapping Zain Aftab,1 Waheed Iqbal , Khaled Mohamad Almustafa, Faisal Bukhari, and Muhammad Abdullah,Punjab University College of Information Technology (PUCIT), University of the Punjab, Lahore, Pakistan ,College of Computer and Information Sciences, Prince Sultan University Riyadh, Riyadh, Saudi Arabia

In this article an efficient and automatic method is proposed and evaluated to NoSQL database into a relational database automatically. Here for experiment evaluation purpose, MongoDB as a NoSQL database and MySQL and PostgreSQL a relational databases to perform transformation tasks for different data set sizes[21]. And observed excellent performance, compared to the existing state-of-the-art methods, in transforming data from a NoSQL database into a relational database.

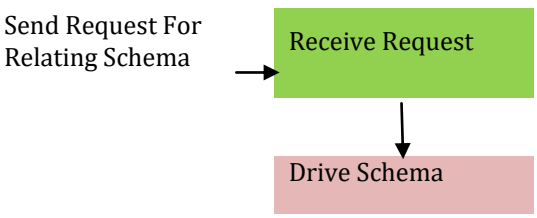
5. "A Model-Driven Approach to Generate Schemas for Object-Document Mappers" ALBERTO HERNÁNDEZ CHILLÓN, DIEGO SEVILLA RUIZ, JESÚS GARCÍA MOLINA, AND SEVERINO FELICIANO MORALES

This research proposed a model-based reverse engineering approach to infer schema models from NoSQL data. Model-driven engineering (MDE) techniques[22] can be used to take advantage of extracted models with different purposes, such as schema visualization or automatic code generation. This paper present an MDE solution to automate the usage of Object-NoSQL mappers when the database already exists.

1.2 Noteworthy contributions in the field of proposed work

A server system having one or more processors and memory stores a plurality of entities in a schemaless database. The Field of Classification Search entities are not structured in accordance with a predefined schema. The server system generates an index for the plural entities stored in the schemaless database. The index has a plurality of index entries sorted in an order. The server system receives a first request from an application and, in response to the first request, accesses an empirically-determined schema and generates a first response based on the empirically-determined schema. The empirically-determined schema is generated from the index. After generating the first response, the server system sends the first response to the application

2. Architecture of proposed methodology



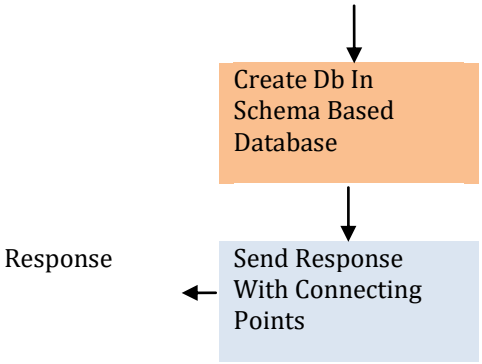


Fig -1: Architecture of proposed methodology

Proposed methodology during the tenure of the research work.

Here an efficient prototype for schemaless database will be developed. In order to design the Schemaless database following are the steps that need to be performed:

1) **Evaluating the underlying Storage to be used for the storage layer:**
Storage plays a vital role for performance of schemaless database. So, different storage format needs to evaluate in order to design storage layer for schemaless database. Some of the storage formats that will be evaluated are JSON [17, 18], Avro [19] etc

2) **Implementing Storage layer:**
In this step we need to implement a storage layer which will use the storage format evaluated in the above step. Different schema merging techniques will be considered while designing this storage layer. This layer will also be responsible for efficient reading/fetching and writing of schemaless database to the storage format

3) **Implementing a Serving Layer:**
In this step a long running server daemon will be implemented which serve the request for the database client. This layer will communicate with the storage layer in order read and write data on behalf of client.

4) **Design and implementing client:**
In this step a client for schemaless will be designed. A non ambiguous grammar will be defined and implemented in this step. A proper communication protocol between database client and server will be implemented in this step.

Tools for generating schema for schemaless databases:

- Apache Calcite:

Apache Calcite is an open source framework for building databases and data management systems. It includes a SQL parser, an API for building expressions in relational algebra, and a query planning engine

- **Apache Pig:**

It was one of the first query languages along with Hive. It has its own language different from SQL. The salient property of Pig programs is that their structure is amenable to substantial parallelization, which in turns enables them to handle very large data sets. It is not in decline in favor of newer SQL based engines.

- **Presto:**

It is an open source distributed SQL query engine for running interactive analytic queries against data sources of all sizes released by Facebook. It presto allows querying data where it lives, including Cassandra, Hive, relational databases and file systems. It performs queries on large data sets within a second. It is independent of Hadoop but integrates with most of its tools, especially Hive to run SQL queries.

- **Apache Drill:**

It provides a schema-free SQL Query Engine for Hadoop, NoSQL and cloud storage. A single query can join data from multiple data stores performing optimizations specific to each data store. It allows analysts to treat any type of data like a table even if they are reading a file. Fully standard SQL is supported by Drill.

Expected outcome of the proposed work.
The result of this research will be a prototype schemaless NOSQL database which can handle schema evolution and big data efficiently.

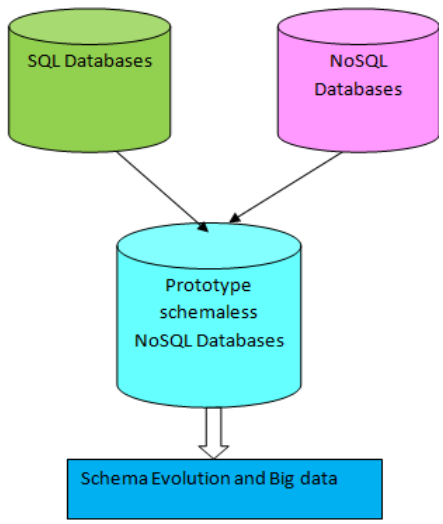


Fig -2: Prototyping schemaless NOSQL database

3. CONCLUSION

The result of this research will be a prototype schemaless NOSQL database which can handle schema evolution and big data efficiently. As there are tools exist to assist (business intelligence, machine learning, big data analytics) with schema design for relational databases and absence of tools to assist with schemaless databases hence need to design and develop schema for schemaless database.

ACKNOWLEDGEMENT

I like to acknowledge my gratitude to Dr. Neeraj Sharma for valuable suggestions in carrying my research work. I also thankful to CSE department of our university.

REFERENCES

[1] Priyanka, AmitPal, "A Review of NoSQL Databases, Types and Comparison with Relational Database" DOI 10.4010/2016.1226, ISSN 2321 3361 © 2016 IJESC

[2] Diego Sevilla , Ruiz, Severino Feliciano Morales, and Jesús García Molina " Inferring Versioned Schemas from NoSQL Databases and Its Applications",, Conference Paper · October 2015, DOI: 10.1007/978-3-319-25264-3_35

[3] avid A. Maluf, Peter B. Trad, and NASA Ames Research Center "NETMARK: A Schema-less Extension for Relational Databases for Managing Semi- structured Data Dynamically", Mail Stop 269-4

[4] Fowler, M.: Schemaless Data Structures (January 2013), http://martinfowler. Com/articles/schemaless/

[5] https://www.vividcortex.com/blog/2015/02/24/schemaless-databases-dont-exist/

[6] Stefan Edlich, "List of NoSQL Databases" Available: http://nosql- database.org, (2011)

[7] Using Spring with NoSQL databases (SpringOne China 2012)

[8] Strozzi, Carlo: NoSQL – A relational database management system. 2007–2010. – http://www.strozzi.it/ cgi- bin/CSA/tw7/I/en_US/nosql /Home % 20 Page

[9] Abdelhedi, Fatma and Ait Brahim, Amal and Atigui, Faten and Zurfluh, Gilles "logical modeling Unified for nosqql databases"(2017) In: 19th International Conference on Enterprise Information Systems,(ICEIS 2017), 26 April 2017 - 29 April 2017 (Porto, Portugal).

[10] Jacky Akoka*,a, Isabelle Comyn-Wattiaub ," Roundtrip engineering of NoSQL databases", Enterprise Modelling and Information Systems Architectures,February2018.DOI:10.18417/ emisa.si.hcm.22

[11] Ron J Mann ,"Method Of Tracking Schema In Schemaless Databases",US20150095298A1,apr 2,2015

[12] Hobberman blog

[13] https://en.wikipedia.org/wiki/ Apache_Calcite

[14] Buneman, P.: Semistructured Data. In: Sixteenth ACM SIGACT-SIGMODSIGART,Symposium on Principles of Database Systems. pp. 117–121. ACM (1997)

[15] Cánovas, J., Cabot, J.: Discovering Implicit Schemas in JSON Data. In: ICWE.pp. 68–83 (July 2013)

[16] IETF: JSON Schema Specification (Visited Apr 2015), http://json-schema.org/

[17] Klettke, M., Störl, U., Scherzinger, S.: Schema Extraction and Structural Outlier Detection for JSON-based NoSQL Data Stores. In: BTW'2105. pp. 425–444 (2015)

[18] https://avro.apache.org/

[19] https://avro.apache.org/

[20] Automated Schema Design for NoSQL Databases, Michael J. Mior, University of Waterloo

[21] Zain Aftab, Waheed Iqbal , Khaled Mohamad Almustafa, Faisal Bukhari, and Muhammad Abdullah,Punjab University College of Information Technology (PUCIT), University of the Punjab, Lahore, Pakistan ,College of Computer and Information Sciences, Prince Sultan University Riyadh, Riyadh, Saudi Arabia "Automatic NoSQL to Relational Database Transformation with Dynamic Schema Mapping"

[22] ALBERTO HERNÁNDEZ CHILLÓN , DIEGO SEVILLA RUIZ, JESÚS GARCÍA MOLINA , AND SEVERINO FELICIANO MORALES "A Model-Driven Approach to Generate Schemas for Object-Document Mappers" DOP May 7, 2019

BIOGRAPHIES

Dr. Neeraj Sharma, Associate Professor, Dept of Computer Science and Engineering, Sri Satya Sai University of Technology & Medical Sciences, Sehore, M.P., India,20 year's of teaching experience UG & PG, students, 08 research papers published in international & national journals, 11 conferences/Webinars & 3 AICTE sponsored workshops.

Ashwini Mandale, received Masters degree from Savitribai Phule University of Pune in Computer Engineering, currently she is research scholar at Sri Satya Sai University of Technology and Medical Sciences, Bhopal, India