Bridging the gap between the semantic web and big data: answering SPARQL queries over NoSQL databases

Hakim El Massari, Sajida Mhammedi, Noreddine Gherabi
Lasti Laboratory, National School of Applied Sciences, Sultan Moulay Slimane University, Khouribga, Morocco

Article Info

Article history:

Received Aug 24, 2021
Revised Jun 17, 2022
Accepted Jul 10, 2022

Keywords:

Big data
NoSQL
Ontology
Semantic web
SPARQL

ABSTRACT

Nowadays, the database field has gotten much more diverse, and as a result, a variety of non-relational (NoSQL) databases have been created, including JSON-document databases and key-value stores, as well as extensible markup language (XML) and graph databases. Due to the emergence of a new generation of data services, some of the problems associated with big data have been resolved. In addition, in the haste to address the challenges of big data, NoSQL abandoned several core databases features that make them extremely efficient and functional, for instance the global view, which enables users to access data regardless of how it is logically structured or physically stored in its sources. In this article, we propose a method that allows us to query non-relational databases based on the ontology-based access data (OBDA) framework by delegating SPARQL protocol and resource description framework (RDF) query language (SPARQL) queries from ontology to the NoSQL database. We applied the method on a popular database called Couchbase and we discussed the result obtained.

Corresponding Author:

Hakim El Massari
Lasti Laboratory, National School of Applied Sciences, Sultan Moulay Slimane University
Khouribga, Morocco
Email: h.elmassari@usms.ma

1. INTRODUCTION

Every business needs an efficient and reliable method to manage data accurately. Databases remain one of the most extensively used methods for keeping client information, inventory, or any other kind of corporate data. For decades, the greatly utilized paradigm for storing and managing structured data has been relational data management.

For many reasons and constraints, companies are used to storing data on various database systems (each subsidiary/organization uses its preferred system, which is often different from others). This makes accessing data across these various and heterogenous databases a complicated task. Some works have been performed to make this task easier, we will discover them in detail in the next section. These works fall into two categories; the first one consists in unifying the needed databases into a single database, this results in querying only a single database rather than many. However, this database is simply a snapshot of the used databases and once their states change, this snapshot becomes obsolete and needs to be generated again. The second approach is different in that it creates a logical layer upon the needed databases and allows using high-level queries. This approach is efficient and sustainable since it gathers data in real-time from the lower levels (databases) and allows these databases to evolve without having to create the logical layer again. Our work focuses on this second approach that allows data access, integration, quality checking, and governance through an ontology and its application in the semantic web world.

Ontology-based access data (OBDA) [1] is one of the paradigms used to implement the second approach, a powerful solution for high-level data access, [2] it allows users to create high-level queries that OBDA automatically converts into low-level queries suited for the used database (DB) engines [3]. One of the most known implementations of OBDA was Ontop, a project introduced by the Free University of Bozen-Bolzano released under the Apache license. Ontop [4] is a project that exposes the content of arbitrary relational databases as knowledge graphs [5]. These graphs are virtual, which means that the data remains in their original sources rather than being moved to a new database. By simplifying the method, Ontop helps in such circumstances by converting ontology inquiries into queries that can be effectively executed over traditional databases. Among the limitations of Ontop is that it only supports relational databases. And we all know that the emergence of increasingly large-scale applications, exposed the drawback of relational data management in managing the storing and querying of big data efficiently and horizontally. Thus, that is why in our work we have added a layer, to enable us to convert the queries to be suitable for the DB engine of not only SQL (NoSQL) databases.

In this article we proposed a method to enable integrating the semantic web with NoSQL databases, we are based on the OBDA Ontop framework to generate adequate queries to query a NoSQL database. We implement our method on Couchbase server a popular JSON documents database. The structure of this paper is as follows. In section 2, we present the literature review from previous research. The method used in section 3. The evaluation and environment in section 4. We discuss the results and conclude the paper in sections 5 and 6 respectively.

2. RELATED WORK

This section gives an overview of many methodologies linked to our work in this article. Other studies have been carried out to give approaches and tools allowing integration of the semantic web and big data. Transforming legacy data from various forms into resource description framework (RDF) is an important first step in enabling RDF-based data integration [6]–[8]. RDF is more and more used as a hinge format for combining disparate data sources. It offers a data model that enables the expansion of a wide number of current terminologies and domain ontologies while still utilizing the Semantic Web's reasoning capacity. RDF data is rapidly being published on the web, particularly in accordance with the linked data standards [9], [10]. This data is frequently supplied from heterogeneous sources that are inaccessible to data integration tools and search engines.

Since the 2000s, a lot of research has been devoted to the conversion of prevalent databases and data formats to RDF. The research has been focused on relational databases [11]. There have been several efforts to use OBDA with NoSQL. To better handle the analysis of this sort of data, Ravat *et al.* [12] relies on pre-aggregation procedures. In particular, they construct a conceptual model to represent the original RDF data in a multidimensional structure with aggregations.

On the other side, ontologies are the outcome of common knowledge that has been arranged to be machine-readable and captures a certain view of the universe that has been clearly specified. Ontologies are utilized in a variety of domains, including software engineering, information extraction, semantic search, knowledge management, recommender systems, and so on. Many studies have been published about the creating of ontology from NoSQL databases [13] and compute ontology similarity to determine the semantic similarity of initial retrieval concepts and execute extended queries [14]–[16].

Many kinds of research have focused on the response to SPARQL protocol and RDF query language (SPARQL) queries on NoSQL databases, we cite a few. Massari *et al.* [17] rely on the OBDA mechanism Ontop to answer SPARQL queries over Couchbase. Michel *et al.* [18] propose a two-step technique for converting a SPARQL query into an abstract pivot query using MongoDB to RDF mappings expressed in xR2RML, then converting the pivot query into a concrete MongoDB query. On top of key-value storage, Mugnier *et al.* [19] investigate the topic of responding to ontology-mediated queries. They define these systems' data models and fundamental queries and provide a rule language for expressing lightweight ontologies on top of data.

A few research papers were published related to our topic which quering NoSQL databases via virtual knowledge graph (VKG) using ontology and SPARQL language to delegate queries and get result from NoSQL databases. To make such databases more accessible and to enable data integration from non-relational data sources, [20] the authors applied the well-known OBDA framework to enable for querying arbitrary databases via a mediating ontology; the implementation was on the MongoDB database. Using the same DB engine, Araujo *et al.* [21] describe a unique OBDA technique based on document-oriented NoSQL databases. The technique employs an access interface with an expandable and adaptable intermediary conceptual layer capable of giving access to many types of database management systems.

3. METHOD

3.1. Background

Since past decades, OBDA has been a widely used strategy for resolving the challenge of accessing current data sources through scalable, effective, and efficient techniques [22]–[25]. An ontology, in the form of a conceptual layer, provides a common language, builds the domain, covers the data source structure, and increases the context data of unintelligible information in OBDA. Thus, Queries may be done utilizing a high-level conceptual viewpoint since users do not need to know anything about the data sources, linkages or encoding of the data. Connecting ontology and data sources is done by a declarative specification that describes mappings between the ontology and the data views. R2RML, a World Wide Web Consortium (W3C) standard for specifying mappings in an OBDA environment, was developed with this objective in mind. Ontology and mappings provide a virtual RDF graph that can be searched using SPARQL.

In the semantic web environment, query answering is critical because it enables users and applications to interact with ontologies and data. As a result, many query languages, such as SPARQL have been created. The W3C defined the SPARQL query language in 2008, and it is currently supported by the majority of RDF triple stores, which is why we chose it.

In this paper, we used the Ontop OBDA system which is used in a variety of applications. All W3C OBDA recommendations, including OWL2 QL, SWRL, R2RML, and SPARQL, are supported by Ontop, as support for all current relational databases. Ontop is accessible via a Protégé plugin, and a Java library that supports the OWL API. As ontology languages, Ontop supports RDFS and OWL2QL. OWL2QL is based on the DL-Lite family of compact description logics, ensuring that ontology inquiries may be converted into database queries in an equivalent manner.

3.2. Onto-Couchbase architecture

To illustrate the many thoughts and concepts mentioned in this article, we propose using the OBDA model, which consists of ontology and mappings as well as an intermediary conceptual layer, to access the contents of a relational database primarily, but we have added a layer that enables us to access the documents of NoSQL database. We demonstrate the onto-Couchbase system, which implements the query translation technique based on the Ontop system, allowing us to query a NoSQL database, in this instance Couchbase, and create a collection of JSON documents as a consequence. The following are the major components of the Onto-Couchbase system, as shown in Figure 1. The user part consists of OWL Ontology with SPARQL queries, then access interface consists of classes plus mapping and query adjustment, last part is a NoSQL database Choubase server in our case. More details about each part are in the sub sub-sections below.

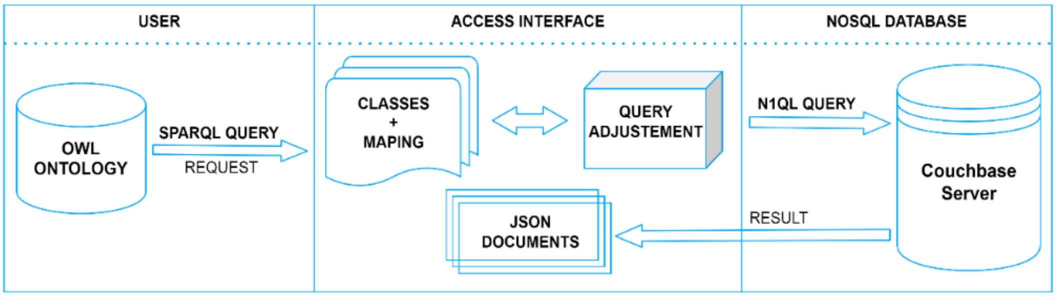


Figure 1. Architecture of onto-Couchbase

3.2.1. User

In this part, we create our ontology shown in Figure 2 using Protégé software [26]. Protégé it is an open-source platform that offers a suite of tools to a growing user community for building domain models and knowledge-based applications with ontologies. The ontology consists of all information from the database which is about a university and all staff and courses.

3.2.2. Access interface

This part is the core of our system, where we made an interface that is a component able to convert SPARQL queries and retrieving JSON documents from the Couchbase server. With the help of Java API of Ontop and Couchbase server, we were able to achieve this work and make this interface a window between the ontology and NoSQL database. The classes in our Java application, along with the mappings in Figure 3(a), expose a virtual RDF graph, which will be interrogated using SPARQL Figure 3(b) by transforming

SPARQL queries into SQL queries Figure 3(c) and get as a result N1QL query Figure 3(d) executed by Couchbase server.

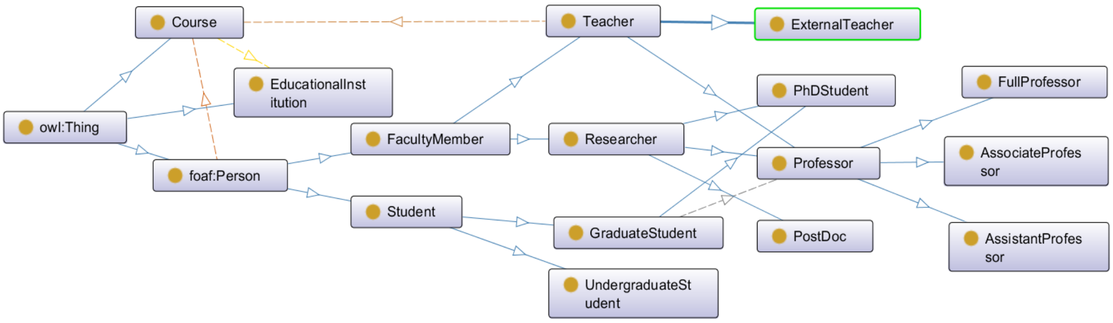
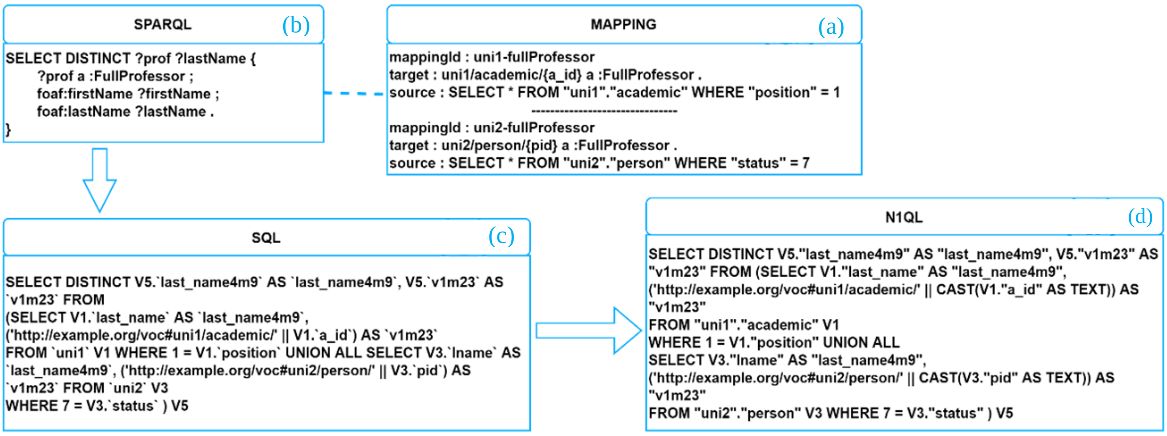


Figure 2. The ontology graphs



```
SELECT DISTINCT ?prof ?lastName {
      ?prof a :FullProfessor ;
      foaf:firstName ?firstName ;
      foaf:lastName ?lastName .
}
```

```
mappingId : uni1-fullProfessor
target : uni1/academic/{a_id} a :FullProfessor .
source : SELECT * FROM "uni1"."academic" WHERE "position" = 1
                    -------------------------------
mappingId : uni2-fullProfessor
target : uni2/person/{pid} a :FullProfessor .
source : SELECT * FROM "uni2"."person" WHERE "status" = 7
```

```
SELECT DISTINCT V5.`last_name4m9` AS `last_name4m9`, V5.`v1m23` AS
`v1m23` FROM
(SELECT V1.`last_name` AS `last_name4m9`,
('http://example.org/voc#uni1/academic/' || V1.`a_id`) AS `v1m23`
FROM `uni1` V1 WHERE 1 = V1.`position` UNION ALL SELECT V3.`lname` AS
`last_name4m9`, ('http://example.org/voc#uni2/person/' || V3.`pid`) AS
`v1m23` FROM `uni2` V3
WHERE 7 = V3.`status` ) V5
```

```
SELECT DISTINCT V5."last_name4m9" AS "last_name4m9", V5."v1m23" AS
"v1m23" FROM (SELECT V1."last_name" AS "last_name4m9",
('http://example.org/voc#uni1/academic/' || CAST(V1."a_id" AS TEXT)) AS
"v1m23"
FROM "uni1"."academic" V1
WHERE 1 = V1."position" UNION ALL
SELECT V3."lname" AS "last_name4m9",
('http://example.org/voc#uni2/person/' || CAST(V3."pid" AS TEXT)) AS
"v1m23"
FROM "uni2"."person" V3 WHERE 7 = V3."status" ) V5
```

Figure 3. Process of querying adjustments (a) mappings, (b) SPARQL query, (c) SQL query, and (d) N1QL

The produced SQL queries are often not optimal and cannot be performed directly by Couchbase server. As a result, we must modify the SQL syntax by including the adjustment query phase in order to construct an N1QL query, keeping in mind that N1QL is called SQL for JSON since it appears very similar to a SQL query. It is intended to deal with both structured and semi-structured data, and it is built on the original SQL with enhancements to allow it to function with JSON document databases by loosening its data model requirements. As a result, the query language preserves the benefits of SQL, such as its high-level (declarative) nature, while also allowing it to handle the more flexible structures inherent in the semi-structured world. On the basis of and because our DB engine does not allow slightly produced SQL dialect, we must modify the SQL syntax. For example, the operator for string concatenation in Couchbase is ||, and the CONCAT function in other relational databases; we utilized backticks instead of double quotation marks in Couchbase, and because Couchbase does not support the CAST function, we removed it. Finally, the modified SQL query is conducted over the Couchbase database and resulting in the retrieval of a JSON document.

3.2.3. NoSQL database

We chose Couchbase server a document-oriented NoSQL database for our Onto-Couchbase system. There are two universities in the database, labeled “uni1” and “uni2”. The University data was created at random using a java method based on an exciting relational structure consisting of 8 tables (Student, academic, courses, and so on). We created 2 million JSON documents, which were distributed among both “uni1” and “uni2” 1 million for each one. The reason why we chose Couchbase server is the only NoSQL database that supports SQL dialect which makes the process of translating SPARQL query to SQL query then to N1QL easy. In addition, is a distributed database that combines the characteristics of relational databases, such as SQL and ACID transactions, with the JSON flexibility and scale that characterizes NoSQL.

4. EVALUATION AND ENVIRONMENT

In this section, we will share the evaluation and environment chosen to make the experimental and draw if Onto-Couchbase is an efficient solution to cover and resolve one of the big data problems which is variety. As we describe in the previous section our system is composed of a SPARQL query and an access interface and a NoSQL database (Couchbase). We provide the database and all files with the documentation online, so that the experiment may be recreated.

To implement the Onto-Couchbase system we import the database to a cluster in Couchbase server, to do so we develop a java program to fill the database with real data. Then we used 5 distinct SPARQL queries to test the operation of the system, the queries used are (details of the queries can be found online): i) Q1: returns all permanent professor; ii) Q2: returns all faculty members; iii) Q3: returns all person; iv) Q4: returns all teachers; and v) Q5: returns information of students taking a course. The queries were tested on HPE ProLiant ML350 Gen9 Server with characteristic of Processor Intel(R) Xeon(R) CPU E5-2620 v3 @ 2.40GHz, 2397 MHz, 6 Core(s), 12 Logical Processor(s), 48 GB of memory, and 1TB SAS 12 Gbps Hard drive. Our Onto-Couchbase system is developed using Java programming language based on Couchbase server and Ontop API's.

5. RESULTS AND DISCUSSION

In this section, we present and discuss the results of the Onto-Couchbase system. Table 1 summarizes the results, and Figures 4 and 5 demonstrate the influence of the number of documents on the execution time for the five Onto-Couchbase queries. Table 1 shows the execution time for our system based on the number of documents returned.

Table 1. Query answering times vs. the number of documents returned in Couchbase

	Q1	Q2	Q3	Q4	Q5
ET[a]	21572	41498	49105	50480	51030
NDR[b]	473	1400730	3000000	644230	502675

a: Query execution time (ET), b: Number of documents returned (NDR).

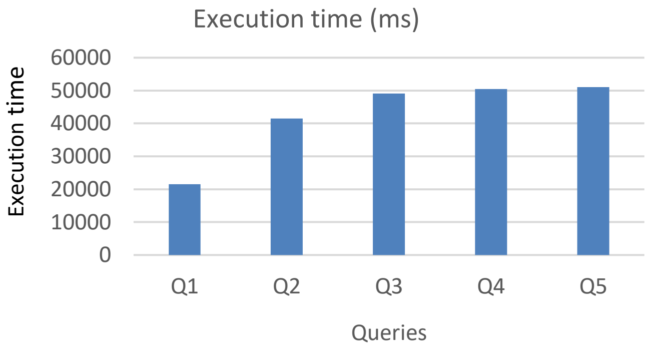


Figure 4. Execution time of all queries in milliseconds

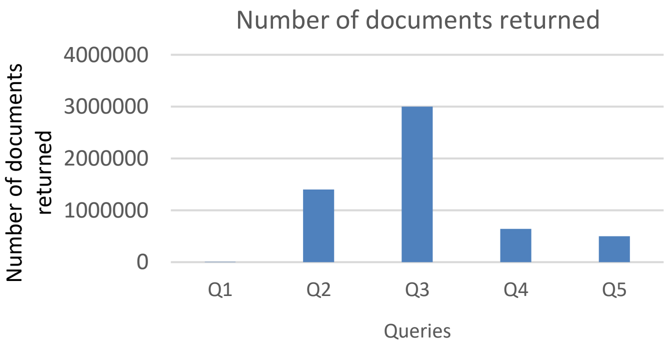


Figure 5. Number of documents retrieved

We are presently concentrating on query evaluation times. Observing the results of our experimental research, there are many factors to consider that can be made, but the most essential one is that there is no relationship between the number of documents returned and query execution time, concluded from Q3 and Q5 we got exactly the same execution time even if we retrieve a different number of documents returned. Although execution time is proportional to the complexity of the SQL query, Couchbase server supports a variety of data processing features, including filtering, deep traversal of nested documents, querying via relationships via JOINs or subqueries, grouping, combining result sets using operators, sorting, and aggregating. The major cause of this is that the Ontop system produces rewrites containing complex subqueries, consisting of unions of multiple select-project-join queries, and these kinds of queries are not efficiently evaluated. As a result, it affects the query execution time.

Our solution illustrates that OBDA is capable of integrating some of the most prevalent NoSQL database capabilities. The characteristics discussed in this study extend the well-known ontology-based data access system for effective data management by allowing high-level conceptual integration. However, we have demonstrated that using OBDA provides capabilities that goes beyond the expectations of most NoSQL developers and consumers, such as querying NoSQL databases through ontologies. The technique used in this study provides benefits for the research community, primarily because it employs Ontop to answer SPARQL queries by rewriting them into SQL queries and sending them to the database. We made this procedure easy for OBDA Systems by developing the entire project in JAVA and using APIs.

6. CONCLUSION

In this paper we tried to query heterogeneous data based on the OBDA Ontop system approach, this is achieved by adding a layer that converts a SPARQL query to an N1QL query known by Couchbase server which is used as a NoSQL database in our case. We built a system called Onto-Couchbase, based on the architecture of the OBDA Ontop system and we evaluate the implementation on a real case of data using a popular NoSQL Couchbase. The study we conducted demonstrates that Onto-Couchbase can generate queries and get JSON documents as a consequence. We conclude that the use of semantic web and ontology to query NoSQL databases will help to connect the semantic web with big data. In future work, we intend to integrate our system into Protégé software as a plugin that will help end-user to exploit it.

REFERENCES

[1] K. Bereta, G. Papadakis, and M. Koubarakis, “OBDA for the web: Creating virtual RDF graphs on top of web data sources,” *arXiv:2005.11264*, May 2020.

[2] E. Botoeva, D. Calvanese, B. Cogrel, J. Corman, and G. Xiao, “Ontology-based data access – beyond relational sources,” *Intelligenza Artificiale*, vol. 13, no. 1, pp. 21–36, Aug. 2019, doi: 10.3233/IA-190023.

[3] T. Bagosi *et al.*, “The ontop framework for ontology based data access,” in *The Semantic Web and Web Science*, Berlin, Heidelberg, 2014, pp. 67–77. doi: 10.1007/978-3-662-45495-4_6.

[4] G. Xiao *et al.*, “The virtual knowledge graph system ontop,” in *The Semantic Web – ISWC 2020*, 2020, pp. 259–277. doi: 10.1007/978-3-030-62466-8_17.

[5] D. Calvanese *et al.*, “Ontop: Answering SPARQL queries over relational databases,” *Semantic Web*, vol. 8, no. 3, pp. 471–487, Dec. 2016, doi: 10.3233/SW-160217.

[6] K. Saleh Aloufi, “Generating RDF resources from web open data portals,” *Indonesian Journal of Electrical Engineering and Computer Science (IJEECS)*, vol. 16, no. 3, pp. 1521–1529, Dec. 2019, doi: 10.11591/ijeecs.v16.i3.pp1521-1529.

[7] M. Buron, F. Goasdoué, I. Manolescu, and M.-L. Mugnier, “Obi-Wan: ontology-based RDF integration of heterogeneous data,” *Proceedings of the VLDB Endowment*, vol. 13, no. 12, pp. 2933–2936, Aug. 2020, doi: 10.14778/3415478.3415512.

[8] J. F. Sequeda, S. H. Tirmizi, O. Corcho, and D. P. Miranker, “Survey of directly mapping SQL databases to the semantic Web,” *The Knowledge Engineering Review*, vol. 26, no. 4, pp. 445–486, Dec. 2011, doi: 10.1017/S0269888911000208.

[9] L. Po, N. Bikakis, F. Desimoni, and G. Papastefanatos, “Linked data visualization: Techniques, tools, and big data,” *Synthesis Lectures on the Semantic Web: Theory and Technology*, vol. 10, no. 1, pp. 1–157, Mar. 2020, doi: 10.2200/S00967ED1V01Y201911WBE019.

[10] P. Zangeneh and B. McCabe, “Ontology-based knowledge representation for industrial megaprojects analytics using linked data and the semantic web,” *Advanced Engineering Informatics*, vol. 46, p. 101164, Oct. 2020, doi: 10.1016/j.aei.2020.101164.

[11] B. Lakzaei and M. Shamsfard, “Ontology learning from relational databases,” *Information Sciences*, vol. 577, pp. 280–297, Oct. 2021, doi: 10.1016/j.ins.2021.06.074.

[12] F. Ravat, J. Song, O. Teste, and C. Trojahn, “Efficient querying of multidimensional RDF data with aggregates: Comparing NoSQL, RDF and relational data stores,” *International Journal of Information Management*, vol. 54, Oct. 2020, doi: 10.1016/j.ijinfomgt.2020.102089.

[13] S. Mhammedi, H. El Massari, and N. Gherabi, “Composition of large modular ontologies based on structure,” in *Advances in Information, Communication and Cybersecurity*, 2022, pp. 144–154. doi: 10.1007/978-3-030-91738-8_14.

[14] V. Devarajan and R. Subramanian, “Analyzing semantic similarity amongst textual documents to suggest near duplicates,” *Indonesian Journal of Electrical Engineering and Computer Science (IJEECS)*, vol. 25, no. 3, pp. 1703–1711, Mar. 2022, doi: 10.11591/ijeecs.v25.i3.pp1703-1711.

[15] H. El Massari, N. Gherabi, S. Mhammedi, Z. Sabouri, and H. Ghandi, “Ontology-based decision tree model for prediction of cardiovascular disease,” *Indian Journal of Computer Science and Engineering*, vol. 13, no. 3, pp. 851–859, Jun. 2022, doi: 10.21817/indjcse/2022/v13i3/221303143.

[16] M. Fariss, N. El Allali, H. Asaidi, and M. Bellouki, “A semantic web services discovery approach integrating multiple similarity measures and k-means clustering,” *Indonesian Journal of Electrical Engineering and Computer Science (IJEECS)*, vol. 24, no. 2, pp. 1228–1237, Nov. 2021, doi: 10.11591/ijeecs.v24.i2.pp1228-1237.

[17] H. El Massari, S. Mhammedi, N. Gherabi, and M. Nasri, “Virtual OBDA mechanism ontop for answering SPARQL queries over couchbase,” in *Advanced Technologies for Humanity*, 2022, pp. 193–205. doi: 10.1007/978-3-030-94188-8_19.

[18] F. Michel, C. Faron-Zucker, and J. Montagnat, “A mapping-based method to query mongoDB documents with SPARQL,” in *Database and Expert Systems Applications*, 2016, pp. 52–67. doi: 10.1007/978-3-319-44406-2_6.

[19] M.-L. Mugnier, M.-C. Rousset, and F. Ulliana, “Ontology-mediated queries for NOSQL databases,” in *AAI Conference on Artificial Intelligence*, 2016, pp. 1051–1057.

[20] E. Botoeva, D. Calvanese, B. Cogrel, M. Rezk, and G. Xiao, “OBDA beyond relational DBs: A study for mongoDB,” 2016.

[21] T. H. D. Araujo, B. T. Agena, K. R. Braghetto, and R. Wassermann, “OntoMongo - ontology-based data access for NoSQL,” in *Proceedings of the X Seminar on Ontology Research*, 2017, pp. 55–66.

[22] H. El Massari, Z. Sabouri, S. Mhammedi, and N. Gherabi, “Diabetes prediction using machine learning algorithms and ontology,” *Journal of ICT Standardization*, pp. 319–338, May 2022, doi: 10.13052/jicts2245-800X.10212.

[23] M. Hajji, M. Qbadou, and K. Mansouri, “Onto2DB: towards an eclipse plugin for automated database design from an ontology,” *International Journal of Electrical and Computer Engineering (IJECE)*, vol. 9, no. 4, pp. 3298–3306, Aug. 2019, doi: 10.11591/ijece.v9i4.pp3298-3306.

[24] R. Y. Alsalhee and A. M. Abdullah, “Building Quranic stories ontology using MappingMaster domain-specific language,” *International Journal of Electrical and Computer Engineering (IJECE)*, vol. 12, no. 1, pp. 684–693, Feb. 2022, doi: 10.11591/ijece.v12i1.pp684-693.

[25] H. El Massari, N. Gherabi, S. Mhammedi, H. Ghandi, F. Qanouni, and M. Bahaj, “An ontological model based on machine learning for predicting breast cancer,” *International Journal of Advanced Computer Science and Applications*, vol. 13, no. 7, 2022, doi: 10.14569/IJACSA.2022.0130715.

[26] M. A. Musen, “The protégé project: A look back and a look forward,” *AI Matters*, vol. 1, no. 4, pp. 4–12, Jun. 2015, doi: 10.1145/2757001.2757003.

BIOGRAPHIES OF AUTHORS

Hakim El Massari received his master degree from Normal Superior School of Abdelmalek Essaadi University, Tétouan, Morocco, in 2014. Currently, he is preparing his Ph.D. in computer science at the National School of Applied Sciences, Sultan Moulay Slimane University, Khouribga, Morocco. His research areas include machine learning, deep learning, big data, semantic web, and ontology. He can be contacted at email: h.elmassari@usms.ma.

Sajida Mhammedi received her Ms Degree in Computer Engineering from Faculty of Science and Technology, Beni Mellal Morocco, she worked as a visiting researcher at the Sultane Moulay Slimane University, her research interests include machine learning, semantic web, recommendation systems, ontology, and big data. She can be contacted at email: sajida.mhammedi@usms.ac.ma.

Noreddine Gherabi is a professor of computer science with industrial and academic experience. He holds a doctorate degree in computer science. In 2013, he worked as a professor of computer science at Mohamed Ben Abdellah University and since 2015 has worked as a research professor at Sultan Moulay Slimane University, Morocco. Member of the International Association of Engineers (IAENG). Professor Gherabi having several contributions in information systems namely: big data, semantic web, pattern recognition, and intelligent systems. He has papers (book chapters, international journals, and conferences/workshops), and edited books. He has served on executive and technical program committees and as a reviewer of numerous international conferences and journals, he convened and chaired more than 30 conferences and workshops. He is member of the editorial board of several other renowned international journals: Co-editor in chief (Editorial Board) in the journal “The International Journal of sports science and engineering for children” (IJSSEC). Associate Editor in the journal International Journal of Engineering Research and Sports Science. Reviewer in several journals/Conferences. Excellence Award, the best innovation in science and technology 2009. His research areas include machine learning, deep learning, big data, semantic web, and ontology. He can be contacted at email: n.gherabi@usms.ma.