



Virtual OBDA Mechanism Ontop for Answering SPARQL Queries Over Couchbase

Hakim El Massari^(✉), Sajida Mhammedi, Nouredine Gherabi, and Mohammed Nasri

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

Abstract. In the last decade, the database field has become substantially diversified, as a consequence, a number of non-relational databases (known also as NoSQL) have been developed, e.g., key-value stores and JSON-document databases, XML, and graph databases. Several issues associated with big data were addressed as a result of the rise of this new generation of data services. However, in the rush to address the problems of big data and vast numbers of active users, NoSQL dropped certain of the core features of databases that make them highly performant and usable such as the global view, that permits users to access data without needing to know how they are logically organized or physically stored in their sources. We address, in this article, the challenge of how to fill the gap between NoSQL and the Semantic web, in order to enable access to such databases and integration of non-relational data sources. However, we extend the well-known framework for ontology-based data access (OBDA), intending to allow a mediating ontology to query arbitrary databases. We instantiate this framework to a popular JSON-document database called Couchbase, and implement a prototype extension of the virtual OBDA mechanism Ontop to answer SPARQL queries over Couchbase.

Keywords: NoSQL · OBDA · Ontop · Ontology · Couchbase

1 Introduction

For many decades, relational data management has been the most widely used technique for storing and manipulating structured data. However, the development of increasingly large-scale applications highlighted the limitation of relational data management to handle the storing and querying large volumes of data in an efficient and horizontal way.

This sparked a paradigm shift, requiring a new generation of databases capable of handling huge volumes of data without losing query efficiency by decreasing query expressivity and accuracy. A large variety of so-called non-relational or NoSQL (not only SQL) databases appeared. (e.g., Neo4j, Cassandra, Couchbase, MongoDB... etc.).

Moreover, this extensive choice of DBMS provides the opportunity to deal with the requirements of a diversity of modern applications and to match more closely their

differing needs with respect to data management, enabling more flexible data schemas, for instance, or more efficient (though simple) queries. Although this heterogeneity contributed to one of the core aspects of Big Data challenges: variety, as databases grow in size and heterogeneity, accessing data using native query languages is becoming more challenging and getting a more and more involved task for users.

In order to facilitate this form of access, the OBDA [1, 2] was proposed to allow users to create high-level ontological queries that will be automatically converted into low-level queries for the conventional virtual method utilized by DB engines. In practice, the distinction between the conceptual and the DB levels has been shown success, particularly where data sources have a rather complex structure and end-users have expertise in data management [3]. The OBDA method connects a database to an ontology using a declarative specification in the form of mappings that link ontology concepts to SQL views over the data. The ontology is typically described using the OWL2 QL profile of the OWL2 [4], SPARQL is used to write queries, and the database is regarded as relational [5]. Ontop takes place with the goal of making the OBDA approach practicable in such cases by automating the procedure, it converts the queries that users raise over the ontology into queries that are performed efficiently over legacy databases.

In our work, we concentrate our efforts on Couchbase, a document-based database management system that is one of the most widely used NoSQL databases today. N1QL is used to query Couchbase, and can be interpreted as a type of SQL injection into a NoSQL database, N1QL is an expressive and powerful language and a complete SQL dialect for querying, transforming, and handling JSON data. Consequently, OBDA over Couchbase can leverage this advantage to efficiently answer queries, while at the same time offering a more user-friendly query language. Accordingly, we introduce an approach to using OBDA on NoSQL databases, by instantiating the generalized OBDA framework over Couchbase as an extension of the OBDA system Ontop, the rest of the paper is structured as follows.

We begin in Sect. 2 by presenting related work from previous researches in the associated fields. To extend the well-known ontology-based data access (OBDA) framework with NoSQL Systems, we give in Sect. 3, our proposal system architecture to query large volumes of data. Then, to evaluate its feasibility, in Sect. 4, we apply this approach in an OBDA that employs a document-oriented NoSQL database in Couchbase. In Sect. 5, we perform experiments and discuss evaluation results. Finally, we draw the conclusions and present future works.

2 Related Work

The RDF [6, 7] is becoming more common as a pivot format for integrating heterogeneous data sources. It provides a single data model that allows building upon large number of existing vocabularies and domain ontologies while still taking advantage of the Semantic Web's reasoning capability. Also, it enables the use of the Web of Data, which is a rapidly expanding global knowledge base.

RDF data is increasingly being released on the Web, notably following the Linked Data principles [8, 9]. This data is often sourced from heterogeneous silos that are unavailable to data integration systems and search engines. As a result, converting legacy

data from disparate formats into RDF representations is a first step toward allowing RDF-based data integration.

In the past fifteen years, a lot of research has gone into figuring out how to convert popular databases and data formats into RDF. The main emphasis was on relational databases. The primary focus was on relational databases [7, 10], together with a set of data formats including XML [11] and CSV [12]. Besides that, with the introduction of numerous non-relational models, the database landscape has become significantly more diverse. NoSQL databases, which were initially designed as the backbone of Big Data Web applications, have gained traction and are now being used as general-purpose, commonplace databases. Nowadays, companies and organizations are using NoSQL to store large volumes of data. These data, on the other hand, are often unavailable to RDF-based data integration systems, and hence unseen to the Web of Data. Despite the fact that releasing their data could open up new integration possibilities and propel the Web of Data forward.

Over the past several years, there has been a lot of research into exposing legacy data as RDF, with two main approaches: materialization (i.e. all legacy data is converted into an RDF graph at once), or on-the-fly conversion of SPARQL queries into the query language required. When dealing with large datasets, materialization can be challenging and expensive, particularly when data freshness is on the line. Numerous methods for achieving SPARQL access to relational data have been suggested, whether in the context of RDB-backed RDF stores or in the case of RDF stores [13–15] or using arbitrary relational schemas [16–19]. R2RML [20], the W3C RDB-to RDF mapping language recommendation is a well-accepted standard many SPARQL-to-SQL rewriting techniques depend on it [17, 19, 21]. Other alternatives seek to map XML [22–24] or CSV data to RDF. RML [1] tackles the mapping of heterogeneous data formats such as CSV/TSV, XML and JSON. xR2RML [25] is an R2RML and RML extension that addresses the mapping of a wide variety of databases to RDF.

In [26], the authors suggest a method to take on the issue of querying vast quantities of statistical RDF data. To support the analysis of such data, this method relies on pre-aggregation strategies. Particularly, the authors describe a conceptual model representing original RDF data with multidimensional structure aggregates.

In another interesting work, the authors [27] have developed a SPARQL to MongoDB query mapping tool, which converts the legacy databases into an easily accessible source of data. A Virtual RDF database can be shown with all stored documents as RDF triples. The conversion takes two phases: the SPARQL query is converted into an abstract query by using mappings from MongoDB documents to RDF written in an intermediate language called xR2RML, and then the query is rewritten as a concrete MongoDB query. Consequently, they demonstrated that rewriting a query to obtain accurate answers is often feasible.

In line with the use of OBDA in NoSQL the authors in [28], study the problem of ontology-mediated query answering over key value stores. The authors create a rule-based ontology language in which keys are used as unary predicates and rules are applied at the record stage (a record is a set of key-value pairs).

Considering the fact that queries are a mixture of get and check operations that, given a path, return a set of values that can be gotten via that path. The authors examine the

challenge of answering these queries using a set of rules. Due to the lack of mappings and, as a result, no difference between user and native database query languages, this work is still outside of the OBDA framework. It's also worth noting that their ontology and query languages don't follow any Semantic Web standards. On the OBDA over NoSQL side there have been a lot of attempts, [29] proposes integrating ontology-based data access into NoSQL stores, emphasizing the importance of using ontology to search for data inconsistencies and, as a result, increase data quality in NoSQL repositories. The integration is accomplished by rewriting SPARQL queries into the native query language of the NoSQL database. The authors give eight examples of queries and how they could be optimized into queries for document or columnar stores.

Additional work in this field [3] introduced a detailed and comprehensive architecture for an OBDA solution in a Big Data scenario is proposed in the Optique project. The key factors in this work are the usability and manageability of an OBDA system. Present OBDA systems, according to the authors, have significant shortcomings such as the use of a formal query language like SPARQL and complicated mapping management. Optique aims to improve the user experience when it comes to querying and handling ontology-based access to vast amounts of data from various sources.

In a somewhat different approach, [21] extend the Ontop Ontology-Based Data Access (OBDA) system to support R2RML mappings. A Datalog program is created by converting a SPARQL query and an R2RML mapping graph. This structured representation is used for integrating and applying logic programming and SQL query optimization techniques. The optimized program is then converted into a SQL query that can be executed.

To the best of our knowledge, little work has investigated how to extend the well-known ontology-based data access (OBDA) framework, in order to allow a mediating ontology to query arbitrary databases especially heterogeneous and non-relational. The works that are based on ontology are related to the mapping and representation of data in OWL format, such as approaches related to measuring the semantic similarity of concepts [30], or approaches based on segmentation or classification [32].

More in line with our work, authors in [31], suggested that the OBDA concepts be applied to MongoDB. They explain a two-step rewriting process of SPARQL queries into the MongoDB aggregate query language, their work is an extension of Ontop [25], which is an OBDA system for relational databases. authors explain a two-step rewriting process of SPARQL queries into the MongoDB aggregate query language. Using a document-oriented MongoDB database, the latest proposed architecture was tested. In a previous work, authors have provided a systematic assessment of a subset of MongoDB data access queries. This assessment revealed that creating a fully generic framework capable of querying any NoSQL DBMS is extremely difficult. NoSQL DBMS share few query patterns, which require the use of a query translator for any NoSQL DBMS, as opposed to relation databases with a SQL (common query language).

In another approach [33] which is comparable in spirit to ours, in that it also seeks to delegate query execution to a NoSQL source engine, and relies on an object-oriented (OO) intermediate representation, which is similar to our "relational view". However, instead of mapping from the source DB to the ontology vocabulary, the mapping is from the ontology vocabulary to the OO layer.

3 OBDA with NoSQL Databases

3.1 OBDA Over Couchbase

Ontology-Based Data Access (OBDA) has been a common technique since the mid-2000s to resolve the issue of accessing current data sources through scalable methods that are both effective and efficient [10].

A conceptual layer in OBDA establishes a common vocabulary, builds the domain, covers the data source structure, and improves the context data of incomprehensive knowledge. Thus, users don't need to know about the data sources, their relationships, or how the data is encoded since queries are queried over this high-level conceptual view. The data sources and ontology are linked via a declarative specification expressed in terms of mappings that bind ontology (properties, classes ...) to data views (SQL). The R2RML [20] W3C standard was developed with the intent of offering a language for specifying mappings in an OBDA environment. The ontology and mappings from a virtual RDF graph that can be queried with SPARQL (the Semantic Web's standard query language).

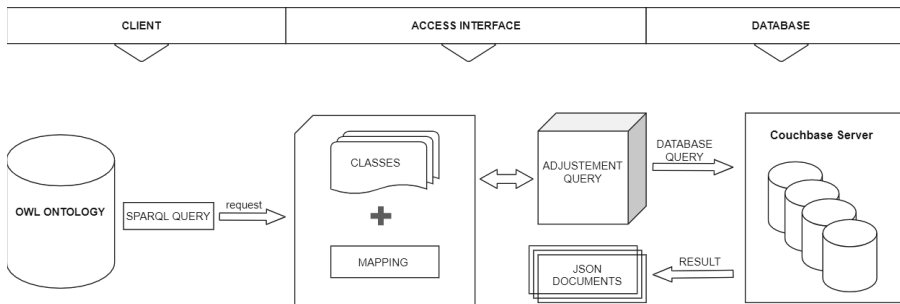


Fig. 1. The Ontop-CB project's architecture

In the context of the Semantic Web, query answering is essential because it offers a mechanism for users and applications to engage with ontologies and data. For this reason, several query languages have been developed, including SerQL, RDQL, and most recently, SPARQL. The World Wide Web Consortium (W3C) standardized the SPARQL query language in 2008, and most RDF triple stores now support it, the thing that led us to choose it.

3.2 Ontop-CB System

We adopted in this article, the Ontop OBDA system [5], which is an open-source system that is actually being used in a number of projects. Ontop supports all W3C OBDA guidelines, including OWL2 QL, SWRL, R2RML, SPARQL, as well as support for all existing relational databases. Ontop is available as a SPARQL endpoint via Sesame Workbench, a Protégé plugin, and a Java library supporting OWL API and Sesame API. Ontop allows for RDFS and OWL2QL [5] as ontology languages. OWL2QL is built on

the DL-Lite family of compact description logics [34, 35], which ensures that ontology queries can be rewritten into database queries equivalently.

To present the different notions and concepts cited in this article, we suggest the use of the OBDA model composed of ontology and mappings as well as an intermediate conceptual layer, in order to access the data of a NoSQL database. We present the Ontop-CB project which implement the query translation method based on the Ontop system, which allows to query NoSQL database, Couchbase in our case, in order to generate a set of JSON Document as a result. As illustrated in Fig. 1, the following are the key components of the onto-CB project: an OWL Ontology, an Access Interface, mappings, a NoSQL database, a SPARQL to NoSQL query adjustment, and a JSON export.

Ontology. An ontology called University Fig. 2 was created with the information systems of two universities describing students, academic staff and courses, based on University database that contains two universities named “uni1” and “uni2”.

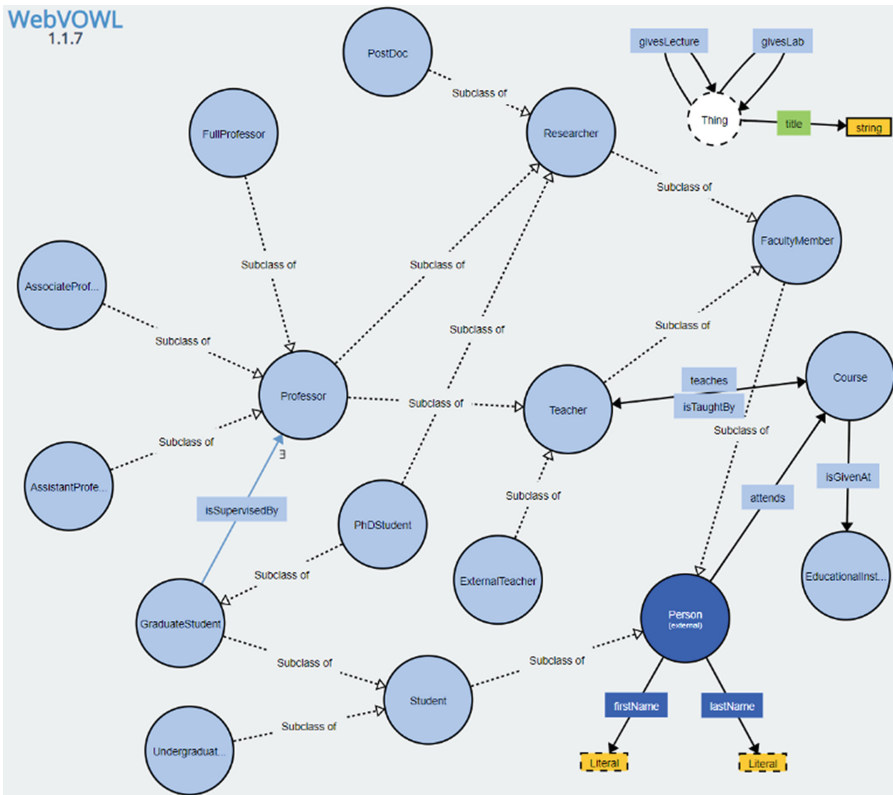


Fig. 2. The graphical representation of “University” ontology

Access Interface. This Interface is a module capable of translating SPARQL query and responding the json document from database, the model was developed using java

programming language based on Ontop API and Couchbase API. Our java program takes as input “owlFile” (Classes, Object Properties, Data Properties), obdaFile (Mappings), propertyFile (connection to database). Given an OWL file, OWLReasoner will check for consistency in ontologies, find subsumption relationships between classes, and even more [15]. The mapping assumption is comprised of two parts: A source, which is a SQL query that retrieves values from the database, and a target, that is a collection of RDF triples containing values from the source (Fig. 3). Our Java program’s classes, combined with the mappings, reveal a virtual RDF Graph, which will be queried using SPARQL Fig. 3(a) by converting SPARQL queries into SQL queries Fig. 3(c).

The generated SQL queries are not necessarily efficient and cannot be directly executed by our DB engine. Hence, we have to adjust the SQL Fig. 3(d) syntax by adding the adjustment query phase in order to generate a N1QL query, taking into account that N1QL is considered as SQL for JSON since it looks very much like a SQL query.

It is designed to work with both structured and semi-structured data, and it is based on the original SQL with extensions that it can work with JSON document database by relaxing its restrictions on the data model. Thus, the query language retains the advantages of SQL, including its high-level (declarative) nature, while enabling it to deal the more flexible structures typically found in the semi-structured world. Based on that and Since our DB engine does not support slightly generated SQL dialect, we have to adjust the SQL syntax accordingly Fig. 3(c). For instance, the operator for string concatenation is || in Couchbase and the concat function in other relational databases; another example In Couchbase, we used backtick instead of double quotation marks; and owing to the fact that Couchbase does not support CAST function lead us to eliminate it.

Finally, the adjusted SQL query is executed over Couchbase database and retrieve json document as results.

Database. The Ontop-CB project uses Couchbase a document-oriented NoSQL database for storage. The database contains two universities named “uni1” and “uni2”.

The University data was generated randomly with java method Based on the relational schema of an excited composed of 8 tables (Student, academic, courses, etc.). We generated a two million of json documents divided between both universities.

Our approach aims to exploit Ontop answers end-user’s SPARQL queries by rewriting them into SQL queries and delegating their execution to the database. To do so, we established an intermediate model layer using classes in java programming language as an intermediate layer between owl ontology and Couchbase.

The Ontop system disclose relational databases as virtual RDF graphs (VRG) by connecting the terms of the ontology to the data sources through mappings. This VRG can then be queried using SPARQL by converting the SPARQL queries into SQL queries over the relational databases.

In our system particularly in the access interface, we aim to adopt the methodology of Ontop, by retrieving the generated SQL query that is then used to query our database in Couchbase. N1QL can be used to query Couchbase Server as an expressive, effective and full SQL dialect to query, update, and manipulate json data [34]. Contrast to other NoSQL databases, Couchbase supports SQL-like query language which makes the transition to Couchbase from RDBMS much easier.

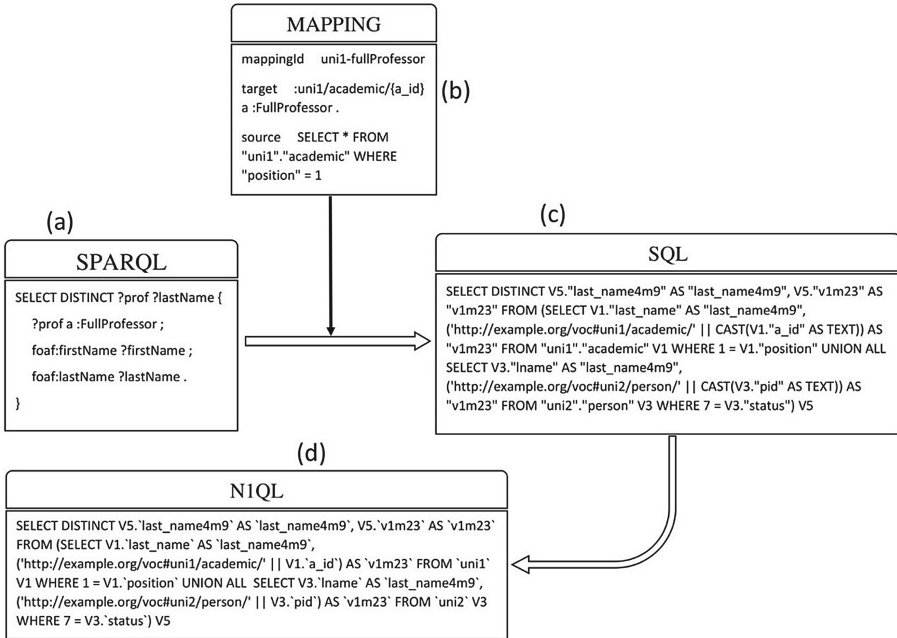


Fig. 3. Adjustment query process (a) Example of SPARQL query (b) Sample of mappings (c) The generated SQL query (d) the adjustment of generated SQL query

4 Evaluation and Environment

An evaluation has been carried out to evaluate whether OBDA over Couchbase is a practical performance solution and, in particular, whether it is capable of leveraging the document structure of Couchbase collections. We implemented an access interface for a query answering system using SPARQL as the query language, Ontop for translating and Couchbase as NOSQL DATABASE.

To realize the proposed system, “University” database is imported to cluster named “University” in Couchbase. The “University” database is available in the CSV format (the format of the CSV is unique to Couchbase).

We developed a java method to dump “University” data based on the relational schema to Couchbase. With the intention to cover all Couchbase constructions, we had to make adjustments in the database design.

Within the tests we used five different SPARQL queries. The listings for the queries can be found online. Here we describe them shortly:

- **FullProfessor:** in this query we searched for a professor with position = 1 for the university uni1 and status = 7 for uni2.
- **FacultyMembers:** this query retrieves all member of faculty.
- **PersonNames:** all person in the university database.
- **Teachers:** all teachers of the university database.
- **Courses:** names of students attended in a course

All experiments were conducted on a PowerEdge R740 server which has Intel(R) Xeon(R) Silver 4110 CPU @ 2.10 GHz with 8 Core/16 threads and 16 GB DDR4-SDRAM, with a 1.63 TB 10K RPM SAS 12 Gbps as RAID-0 hard drive cluster. The RAID controller is Dell PERC H740P Mini (integrated).

This work is done by Java programming language and ONTOP API 4.0.2 version. Couchbase-java-client:3.0.8 is used to access Couchbase. For the graphical representation, we used “WebVOWL” [36, 37] a web application for the interactive visualization of ontologies.

The database and all configuration files are available online, together with the SPARQL queries and mapping are also provided¹, in order for the experiment can be reproduced.

5 Result and Discussion

We consider a University database with 8 tables that contains information about two universities ‘uni1’ and ‘uni2’ stored into two buckets in Couchbase. The results are summarized in Table 1, and we show the impact of number of documents over execution time in Fig. 5 and Fig. 4 respectively for the 5 Ontop-CB queries.

Table 1 reports the execution times for our system Ontop-CB w.r.t number of documents returned. We did not include in this table the query rewriting time (SPARQL to SQL), due to its small time (<1000 ms) resulting from using ONTOP API under Java programming language.

We now focus on query evaluation times. Several considerations can be made when looking at the results of our experimentation, but the most important one is that no relation between the number of documents returned and query execution time, which is observed in queries 3, 4 and 5 we got a very close execution time Fig. 4, and confirmed if we compare q3 and q5, query 3 outperforms query 5 even if it returns a large amount of documents Fig. 5.

Table 1. Query answering times by the number of documents over Couchbase (in milliseconds)

Query	Q1	Q2	Q3	Q4	Q5
Execution time	21572	41498	49105	50480	51030
Number of documents returned	473	1400730	3000000	644230	502675

The execution time has a relationship with the complexity of the SQL, even though Couchbase supports a number of data processing capabilities including filtering, deep traversal of nested documents, querying through relationships using JOINS or subqueries, grouping, combining result sets using operators, sorting, aggregation, and more. The main reason for that is the system Ontop produces rewriting containing complex subqueries, composed of unions of several select-project-join queries and these types of queries are not evaluated efficiently [5]. This is explained by the fact of having a negative effect on the query execution time.

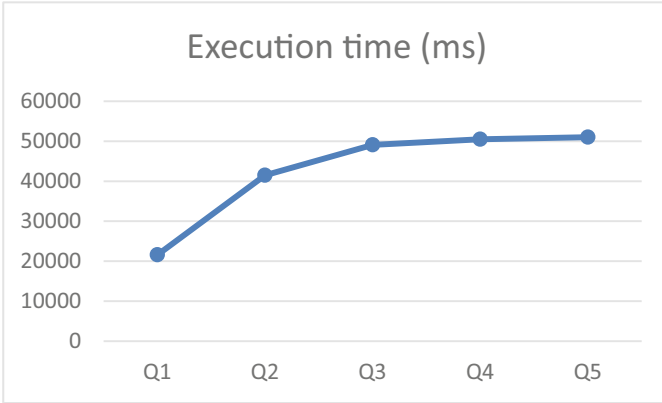


Fig. 4. Execution time in milliseconds

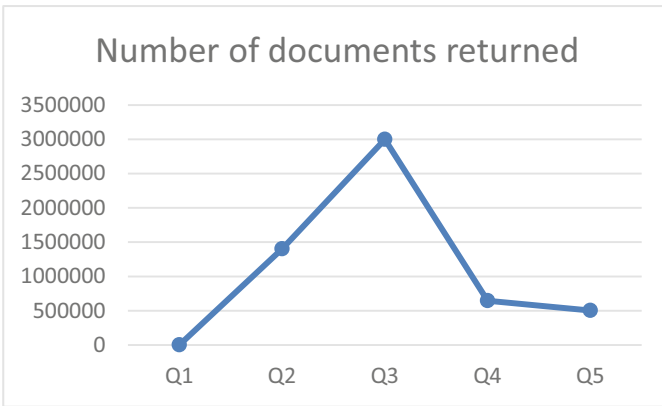


Fig. 5. The number of documents returned for each query

Our system shows that in NoSQL databases, OBDA can include some of the most popular features. The features covered in this paper are the extension of the well-known ontology-based data access system for the appropriate management of data that allows high level conceptual integration.

Nonetheless, we have shown that the use of OBDA offers functionality that go beyond the standards of most developers and users of NoSQL, such as Querying NOSQL Databases over semantic schemes.

The approach taken in this study has advantages for the research community, mainly since it uses Ontop answers SPARQL queries by rewriting into SQL queries and assigning them to the database. By programming the whole project in JAVA language using the APIs (application program interfaces), we made this process easier to OBDA Systems.

6 Conclusion

In this article we aim to encourage the creation of SPARQL interfaces to heterogeneous databases, which we agree is critical to the advancement of the Web of Data. Especially, we believe that this will aid in bridging the gap between the Semantic Web and the NoSQL database family.

A practical architecture for a virtual OBDA approach is proposed that allows SPARQL queries to be answered via arbitrary data sources. As an extension (called Ontop-CB) of the OBDA system Ontop, we've implemented this framework in the particular case of Couchbase, and we conducted an experimentation based on the real-world use case of a document reference stored in a Couchbase database. The evaluation we have carried out shows that Ontop-CB is able to generate queries and retrieve json document as results.

7 Future Work

As we prove from analysis, we found that the complexity of the query affect time of the execution, which conduct us to Improve query rewriting by reducing complex subqueries.

As a continuation of this work, we intend to minimize execution time in the following directions:

- (i) Eliminate phase of adjustment by placing an optimizer layer next to the resulting SQL.
- (ii) Integrate the phase of adjustment query within Ontop project to optimize the generated queries, more than that produce and delegate N1QL query directly to Couchbase.
- (iii) We plane to compare execution time of the two proposed solutions (i) and (ii).

Furthermore, we'll develop the GUI interfaces and add functionality, for ease of use by end users of Ontop-CB.

Footnotes

1 shorturl.at/eBT06.

References

1. Dimou, A., Vander Sande, M., Colpaert, P., Verborgh, R., Mannens, E., Van de Walle, R.: RML: a generic language for integrated RDF mappings of heterogeneous data. In: Bizer, C., Heath, T., Auer, S., and Berners-Lee, T. (eds.) Proceedings of the 7th Workshop on Linked Data on the Web, Korea (2014)
2. Bereta, K., Papadakis, G., Koubaraki, M.: OBDA for the web: creating virtual RDF graphs on top of web data sources. [arXiv:2005.11264v1](https://arxiv.org/abs/2005.11264v1) (2020)
3. Calvanese, D., et al.: Optique: OBDA solution for big data. In: Cimiano, P., Fernández, M., Lopez, V., Schlobach, S., Völker, J. (eds.) ESWC 2013. LNCS, vol. 7955, pp. 293–295. Springer, Heidelberg (2013). https://doi.org/10.1007/978-3-642-41242-4_48

4. Hogan, A.: Web ontology language. In: Hogan, A. (ed.) *The Web of Data*, pp. 185–322. Springer, Cham (2020). https://doi.org/10.1007/978-3-030-51580-5_5
5. Calvanese, D., et al.: Ontop: answering SPARQL queries over relational databases. *Semant. Web* **8**, 471–487 (2017). <https://doi.org/10.3233/SW-160217>
6. Buron, M., Goasdoué, F., Manolescu, I., Mugnier, M.-L.: Obi-Wan: ontology-based RDF integration of heterogeneous data. *Proc. VLDB Endow.* **13**, 2933–2936 (2020). <https://doi.org/10.14778/3415478.3415512>
7. Sequeda, J.F., Tirmizi, S.H., Corcho, O., Miranker, D.P.: Survey of directly mapping SQL databases to the semantic web. *Knowl. Eng. Rev.* **26**, 445–486 (2011). <https://doi.org/10.1017/S0269888911000208>
8. Po, L., Bikakis, N., Desimoni, F., Papastefanatos, G.: Linked data visualization: techniques, tools, and big data. *Synthesis Lect. Semant. Web: Theory Technol.* **10**, 1–157 (2020). <https://doi.org/10.2200/S00967ED1V01Y201911WBE019>
9. Zangeneh, P., McCabe, B.: Ontology-based knowledge representation for industrial megaprojects analytics using linked data and the semantic web. *Adv. Eng. Inform.* **46**, 101164 (2020)
10. Gherabi, N., Addakiri, K., Bahaj, M.: Mapping relational database into OWL Structure with data semantic preservation. [arXiv:1205.5922](https://arxiv.org/abs/1205.5922) [cs] (2012)
11. Calvanese, D., Liuzzo, P., Mosca, A., Remesal, J., Rezk, M., Rull, G.: Ontology-based data integration in EPNet: production and distribution of food during the Roman Empire. *Eng. Appl. Artif. Intell.* **51**, 212–229 (2016). <https://doi.org/10.1016/j.engappai.2016.01.005>
12. Spanos, D.-E., Stavrou, P., Mitrou, N.: Bringing relational databases into the semantic web: a survey. *Semant. Web* **3**, 169–209 (2012). <https://doi.org/10.3233/SW-2011-0055>
13. Bilidas, D., Koubarakis, M.: Efficient duplicate elimination in SPARQL to SQL translation. In: *Description Logics* (2018)
14. Chebotko, A., Lu, S., Fotouhi, F.: Semantics preserving SPARQL-to-SQL translation. *Data Knowl. Eng.* **68**, 973–1000 (2009). <https://doi.org/10.1016/j.datak.2009.04.001>
15. Elliott, B., Cheng, E., Thomas-Ogbuji, C., Ozsoyoglu, Z.M.: A complete translation from SPARQL into efficient SQL. In: *Proceedings of the 2009 International Database Engineering & Applications Symposium*, pp. 31–42. Association for Computing Machinery, New York (2009)
16. Matentzoglou, N., Leo, J., Hudhra, V., Sattler, U., Parsia, B.: A survey of current, stand-alone OWL reasoners. In: *ORE* (2015)
17. Vavliakis, K.N., Grollios, T.K., Mitkas, P.A.: RDOLE – publishing relational databases into the semantic web. *J. Syst. Softw.* **86**, 89–99 (2013). <https://doi.org/10.1016/j.jss.2012.07.018>
18. Priyatna, F., Corcho, O., Sequeda, J.: Formalisation and experiences of R2RML-based SPARQL to SQL query translation using morph. In: *Proceedings of the 23rd International Conference on World Wide Web*, pp. 479–490. Association for Computing Machinery, New York (2014)
19. Rodríguez-Muro, M., Kontchakov, R., Zakharyashev, M.: Ontology-based data access: ontop of databases. In: Alani, H., et al. (eds.) *ISWC 2013. LNCS*, vol. 8218, pp. 558–573. Springer, Heidelberg (2013). https://doi.org/10.1007/978-3-642-41335-3_35
20. Das, S., Sundara, S., Cyganiak, R.: R2RML: RDB to RDF mapping language. Working group recommendation, World Wide Web Consortium. W3C (2012)
21. Rodríguez-Muro, M., Rezk, M.: Efficient SPARQL-to-SQL with R2RML mappings. *J. Web Semant.* **33**, 141–169 (2015). <https://doi.org/10.1016/j.websem.2015.03.001>
22. Gherabi, N., Bahaj, M.: Outline matching of the 2D shapes using extracting XML data. In: Elmoataz, A., Mammass, D., Lezoray, O., Nouboud, F., Aboutajdine, D. (eds.) *ICISP 2012. LNCS (LNAI)*, vol. 7340, pp. 502–512. Springer, Heidelberg (2012). https://doi.org/10.1007/978-3-642-31254-0_57

23. Bikakis, N., Tsinarakis, C., Gioldasis, N., Stavrakantonakis, I., Christodoulakis, S.: The XML and semantic web worlds: technologies, interoperability and integration: a survey of the state of the art. In: Anagnostopoulos, I.E., Bieliková, M., Mylonas, P., Tsapatsoulis, N. (eds.) *Semantic Hyper/Multimedia Adaptation: Schemes and Applications*, pp. 319–360. Springer, Heidelberg (2013). https://doi.org/10.1007/978-3-642-28977-4_12
24. Yang, D., Yan, L.: Transforming XML to RDF(S) with Temporal Information. *J. Comput. Inf. Technol.* **26**, 115–129 (2018). <https://doi.org/10.20532/cit.2018.1004005>
25. Bagosi, T., et al.: The Ontop framework for ontology based data access. In: Zhao, D., Du, J., Wang, H., Wang, P., Ji, D., Pan, J.Z. (eds.) *CSWS 2014. Communications in Computer and Information Science*, vol. 480, pp. 67–77. Springer, Heidelberg (2014). https://doi.org/10.1007/978-3-662-45495-4_6
26. Ravat, F., Song, J., Teste, O., Trojahn, C.: Efficient querying of multidimensional RDF data with aggregates: comparing NoSQL, RDF and relational data stores. *Int. J. Inf. Manage.* **54**, 102089 (2020). <https://doi.org/10.1016/j.ijinfomgt.2020.102089>
27. Michel, F., Faron-Zucker, C., Montagnat, J.: A mapping-based method to query MongoDB documents with SPARQL. In: Hartmann, S., Ma, H. (eds.) *DEXA 2016. LNCS*, vol. 9828, pp. 52–67. Springer, Cham (2016). https://doi.org/10.1007/978-3-319-44406-2_6
28. Mugnier, M.-L., Rousset, M.-C., Ulliana, F.: Ontology-mediated queries for NOSQL databases. In: *AAAI Conference on Artificial Intelligence*, Phoenix, USA, pp. 1051–1057 (2016)
29. Curé, O., Kerdjoudj, F., Faye, D., Duc, C.L., Lamolle, M.: On the potential integration of an ontology-based data access approach in NoSQL stores. In: *2012 Third International Conference on Emerging Intelligent Data and Web Technologies*, pp. 166–173 (2012)
30. Daoui, A., Gherabi, N., Marzouk, A.: A new approach for measuring semantic similarity of ontology concepts using dynamic programming. *J. Theor. Appl. Inf. Technol.* **95**(17), 4132–4139 (2017)
31. Botoeva, E., Calvanese, D., Cogrel, B., Rezk, M., Xiao, G.: OBDA beyond relational DBs: a study for MongoDB. In: *Proceedings of the 29th International Workshop on Description Logics (DL 2016)*, p. 12 (2016)
32. Nejehi, R., Gherabi, N., Marzouk, A.: Towards classification of web ontologies using the horizontal and vertical segmentation. In: Noreddine, G., Kacprzyk, J. (eds.) *ITCS 2017. AISC*, vol. 640, pp. 70–81. Springer, Cham (2018). https://doi.org/10.1007/978-3-319-64719-7_7
33. Araujo, T.H.D., Agena, B.T., Braghetto, K.R., Wassermann, R.: OntoMongo - ontology-based data access for NoSQL. In: *Proceedings of the X Seminar on Ontology Research*, Brazil, pp. 55–66 (2017)
34. Calvanese, D., De Giacomo, G., Lembo, D., Lenzerini, M., Rosati, R.: Tractable reasoning and efficient query answering in description logics: the DL-Lite family. *J. Autom. Reason.* **39**, 385–429 (2007). <https://doi.org/10.1007/s10817-007-9078-x>
35. Artale, A., Calvanese, D., Kontchakov, R., Zakharyashev, M.: The DL-Lite family and relations. *J. Artif. Int. Res.* **36**, 1–69 (2009). <https://doi.org/10.1613/jair.2820>
36. Lohmann, S., Negru, S., Haag, F., Ertl, T.: Visualizing ontologies with VOWL. *Semant. Web* **7**, 399–419 (2016). <https://doi.org/10.3233/SW-150200>
37. Lohmann, S., Link, V., Marbach, E., Negru, S.: WebVOWL: web-based visualization of ontologies. In: Lambrix, P., et al. (eds.) *EKAW 2014. LNCS (LNAI)*, vol. 8982, pp. 154–158. Springer, Cham (2015). https://doi.org/10.1007/978-3-319-17966-7_21