International Journal of Multidisciplinary Studies and Innovative Technologies

e-ISSN: 2602-4888
dergipark.org.tr/en/pub/ijmsit
Research Article

2022, 6 (2), 202-207
DOI: 10.36287/ijmsit.6.2.202
Received: Nov. 4, 2022; Accepted: Dec.12, 2022

Integrating Ontologies and Relational Databases for An Improved Data Access

Aytuğ TÜRKMEN[1*], Özgü CAN [2]

[1*]*Graduate School of Natural and Applied Sciences, Ege University, İzmir, Turkey (aytugturkmen3@gmail.com) (ORCID: 0000-0001-6801-6808)*

[2] *Department of Computer Engineering, Ege University, İzmir, Turkey (ozgu.can@ege.edu.tr) (ORCID: 0000-0002-8064-2905)*

Abstract – Today, recent rapid advancements and developments in information and communication technologies have enabled the number of connected devices to increase gradually. Thus, a significant amount of data from many sources has been produced as a result of the increase in connected devices. It is now more crucial than ever to access and draw conclusions from the data kept in relational database management systems. Ontology-Based Data Access (OBDA) is a hugely important paradigm for leveraging Semantic Web technologies to access data stored in relational databases. Users access data via a conceptual layer in the OBDA paradigm. Data is stored in relational databases and is represented by the conceptual layer as an RDF(S) or OWL ontology. In this paper, a demonstration for the integration of a relational database and an ontology is presented in order to provide an ontology-based data access. The goal of the presented demonstration is to serve as a roadmap for further research studies. For this purpose, a data set stored in MySQL relational database management system is accessed using the OBDA model. Further, the Ontop Framework that is compatible with the Protégé ontology development environment is used to provide the integration between the ontology and the relational database. Moreover, the Ontop queries are executed over SPARQL.

Keywords *– Ontology, Data Access, Relational Databases, Semantic Web, SPARQL, Ontop*

Citation: Türkmen, A., Can, Ö. (2022). Integrating Ontologies and Relational Databases for An Improved Data Access. International Journal of Multidisciplinary Studies and Innovative Technologies, 6(2): 202-207.

I. Introduction

New generation information technologies have emerged in recent years. Thus, there has been significant growth in data volumes [1]. Moreover, the exponential growth in the number and variety of data collections is expected to continue in the coming years. Furthermore, this data is heterogeneous and comes from different sources such as sensors, social media, and Internet of Things (IoT) devices. Thus, accessing and making inferences from the data that is kept in the relational database management systems has become more crucial. Moreover, heterogeneous data from different sources may cause problems for the integration and communication of new generation technologies. Semantic deficiency in data representation needs to be eliminated to resolve this problem.

The Ontology-Based Data Access (OBDA) provides users to access relational databases through a conceptual layer by using Semantic Web technologies [2]. In this approach, ontologies are used with the help of mapping to query over the database. In this way, a high-level conceptual view of the data is obtained [2]. This semantic technology makes relational database systems more accessible and interoperable by addressing issues such as conceptual modeling, query rewriting, and mapping [3]. The conceptual modeling is characterized as a representation of the RDF(S) and OWL ontologies including the data stored in traditional relational databases [2]. W3C suggested mappings like R2RML are used to translate terms from the conceptual layer to the data layer. [4]. In this way, RDF query languages are used to access the data in the relational through ontology. Ontop is a free and open-source OBDA framework [5]. It enables querying of virtual RDF graphs using RDF query languages such as SPARQL, and SWRL which is recommended by W3C by converting the queries to SQL. It also provides ease of access to many relational database systems.

In this work, an Employee Ontology is generated and used to enable the use of SPARQL queries to access data stored in a sample Employees relational database. For this purpose, the OBDA paradigm is achieved with by using the Ontop Framework. The database's tables and relationships are described by the ontology that was developed. Additionally, the mapping procedure is used to connect database entities to ontology instances. As data from different sources can be heterogeneous, data integration is a challenging task. The goal of this study is to demonstrate the integration of a relational database and an ontology to provide an ontology-based data access. Additionally, this demonstration is structured to serve as a roadmap for further research studies in this field. Therefore, semantic queries can be executed, and new relationships can be discovered with maintaining the mapping between databases and ontologies [6]. Thus, the quality of data integration will be improved.

The organization of the study is as follows: Section 2 presents a review of the literature, Section 3 describes the OBDA paradigm and an example application of semantic data integration and access, and finally the study is concluded in Section 4.

II. RELATED WORK

The OBDA approach has been crucial in current information technology advancements with the recent developments in the Semantic Web and Relational Databases fields. The developed OBDA system known as the Ontop Framework [7] has gained support from both the industry and researchers. Thus, the connection between databases and the Semantic Web has been extensively researched in the literature. In [8] the performance of the Ontop Framework is analyzed and compared with other systems. In [1], the conversion of relational databases to ontology representation is explained. The advantages and disadvantages of these methods are examined and some of the most recent mapping tools are reviewed from a theoretical perspective. Also, the drawbacks of the existing tools are discussed. A survey for convenient and user-friendly access to data sources and an ontology-based data access framework is presented in [9]. Also, the fundamental components of ontology-based data access, important theoretical findings, methods, applications, and upcoming difficulties are examined with a focus on relational data sources. In [2], a virtualized method that responds to SPARQL queries is presented. The presented method converts SPARQL queries into SQL queries over the datasets instead of materializing the triples obtained through mappings. Also, Ontop which is a sophisticated open-source OBDA framework is proposed. Ontop generates SQL queries that can be executed quickly and supports all relevant OBDA-related W3C standards. An approach is proposed by [10] named as PerfectMap. The proposed approach provides a novel method for the optimization of query rewriting in OBDA. For this purpose, the key concepts of mapping aspects and the use of perfect mappings are examined, and the complexity caused by mapping rewriting is reduced.

Besides these studies presented in the literature, the goal of this study is to serve as a guide to the key challenges for the integration of variables from different sources. For this purpose, a semantic data integration approach is demonstrated to guide future research studies.

III. MATERIALS AND METHOD

A. Materials

1. The Ontology-Based Data Access

The OBDA database access paradigm employs a conceptual layer to access data. [11]. The fundamental concept of an OBDA system is query rewriting [4]. The user of the system creates a query in the ontology language. The prime objective of the OBDA system is to rewrite the query into a new query utilizing the data source's vocabulary.

In OBDA, an ontology and a relational database are linked by using mappings. It presents a high-level conceptual view of the relational database as a result. Ontology-based user queries are presented, and answers are generated by using both the ontology and the mappings as a basis for reasoning. As shown in Fig. 1, the OBDA architecture consists of three layers [11]. The first layer of the OBDA architecture contains datasets that the system accesses. The second layer is the conceptual layer. In this layer, the user interacts with the system using the query interface and ontologies. Finally, the mapping layer maintains the mapping between the dataset attributes and ontology instances.

There are several OBDA systems that provide SPARQL endpoint functionality and enable mapping from the ontology to the data source such as Ontop, and MASTRO [12]. Ontop is the framework that this study is established on. The Ontop Framework provides a user-friendly interface to execute SPARQL queries and perform the mapping. In the next section, detailed information about Ontop will be given. MASTRO is a system for the OBDA paradigm which is developed at the University of Rome [12]. It can be used as a Protégé [13] plugin. Also, it has a commercial application called MASTRO STUDIO. Moreover, it supports R2RML mapping like the Ontop Framework [14]. Additionally, the native mapping language is also included in MASTRO.

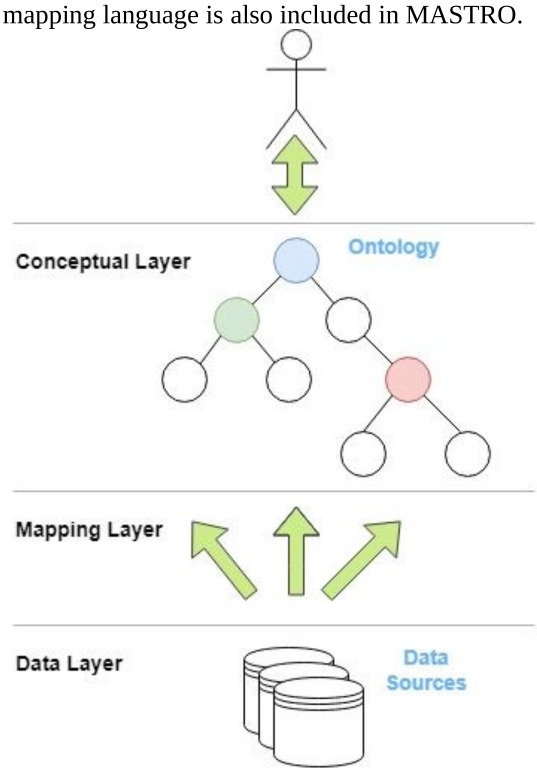

Fig. 1. The OBDA Architecture

2. Ontop Framework

The Ontop is an open-source OBDA framework developed at the Free University of Bozen-Bolzano and distributed under the Apache license [15]. The Ontop Framework enables the representation of relational databases as virtual RDF graphs via mappings. The Ontop Framework converts queries to SQL and enables querying of virtual RDF graphs by using RDF query languages such as SPARQL and SWRL, which are recommended by the W3C. The Ontop architecture consists of four layers [2]. The first is the input layer which includes concepts such as ontology, mappings, database, and queries. It is connected to the relational database after being mapped with the help of the generated ontology via the JDBC driver in this layer. The Ontop Framework makes it simple to use by supporting a wide range of primary and commercial relational databases such as MySQL, Postgres, etc. [2]. Ontop native mapping languages and R2RML are both supported by Ontop [4]. The mapping languages establish connections between the generated ontology and the relational database. The native Ontop mapping language is easy to use, whereas R2RML, a widely used standard, is offered by W3C. The Ontop can simply convert from its native mapping language to R2RML. Moreover, the Ontop provides support for the usage of RDFS and OWL as an ontology language. The second layer is the core layer. In this layer, there are operations such as optimization, query translation, and execution. In this layer, SPARQL queries are converted to SQL queries with the help of the SPARQL Query Answering Engine (Quest). Thus, queries are executed efficiently on the relational database systems. In the third layer, there are APIs named OWL API and Sesame that present Java interfaces to users. Applications layer, the fourth layer, enables end users to query databases using SPARQL. The Ontop Framework can be used in a variety of ways. In this study, the Ontop Framework is used as a plugin for Protégé. It supports the SPARQL query execution and provides a graphical interface for the mapping.

B. Method

In this study, the sample Employees database [16] is used. The dataset contains six tables and includes more than four million data. Different relational database systems, such as MySQL, PostgreSQL, Oracle, etc., may be queried using the Ontop Framework. In this study, the employee sample dataset is stored in the MySQL relational database system. Fig. 2 shows the Entity-Relationship (ER) diagram of the employee sample dataset.

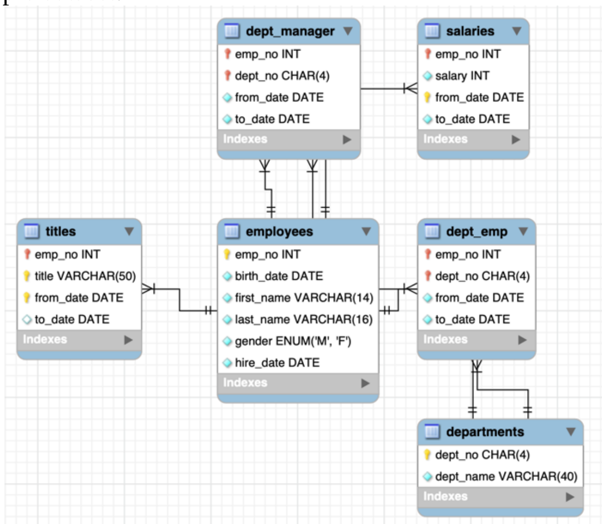

Fig. 2. The ER diagram of the employee sample dataset

Later, the Employee ontology is created. The Employee ontology was developed using the Protégé ontology editor. The classes of the Employee ontology are shown in the Fig. 3.

The Protégé ontology development environment is compatible with the Ontop Framework. Therefore, each table with primary key and foreign key mapped into the related class. Thus, four table is mapped into classes that are named as "`employees`", "`departments`", "`titles`" and "`sales`".

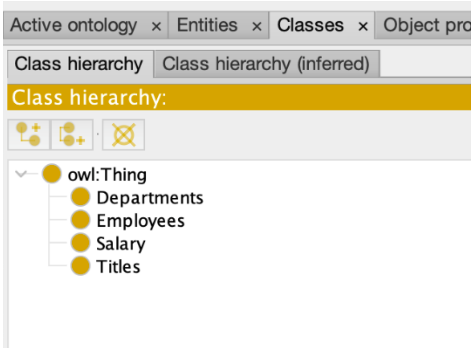

Fig. 3. Classes of the Employee Ontology

In the Employees sample dataset, there are two associative tables named as "`dept_emp`" and "`dept_manager`". These two tables establish a 1:M relationship between the "`employee`", "`salaries`", and "`titles`" tables. The related relationships are mapped into object properties as shown in Fig. 4.

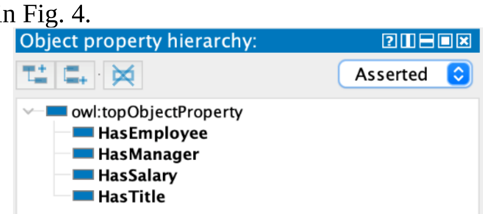

Fig. 4. The Object Properties of the Employee Ontology

Each table in the employee sample dataset has multiple columns (attributes). The columns of tables are mapped as data properties. Fig. 5 shows the data properties of the Employee Ontology. The general view of the ontology is presented in Fig. 6.

When mapping an ontology to a data source, the mapping is made by defining the target as a triples template and the source as a SQL query over the data source [17]. The mappings are presented in Fig. 7. The mapping that will enable query data from the Employees table using SPARQL queries is shown in Fig. 8.

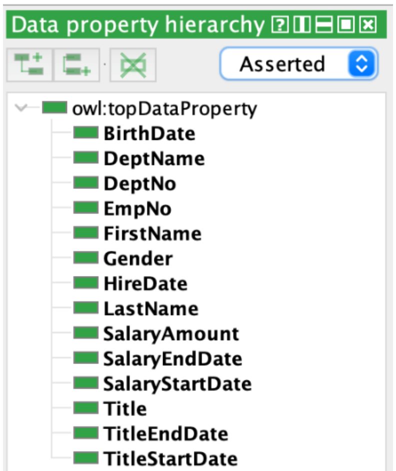

Fig. 5. The Data Properties of the Employee Ontology

The Ontop Framework supports both R2RML and Ontop native mapping languages as mapping languages. The native Ontop mapping language is simple to learn and use, while R2RML is a widely accepted standard that is published by W3C. In this study, natural mappings are used to link the Employee ontology to the data in the SQL database. Two files are produced by Ontop. The "`.obda`" file extension is used to create the first file. The mapping details between the data source and the ontology are all contained in this file. A second file, with the "`.properties`" extension maintains the link to the data source. When using the Ontop Protégé plugin to construct an ontology, a file with the "`.properties`" extension is generated automatically. The mapping between a data source and an ontology can be declared using the mapping axioms. The mapping id, source, target, and SQL query results are the four fields that each mapping consists of. The mapping id is a string that is used to identify mapping axioms. The data source was used as the source for a SQL query. In the target field, the expressions in curly brackets in the Triples template represent the columns of the source SQL tables. The predicates of triples whose objects contain data types represent data properties. At the beginning of the triples template, "`is-a`" relationship refers to classes. Also, triple templates are built to create 1:M relationships. The triple template created for the `HasTitle` object property is shown in Fig. 9. The `HasTitle` object property links `Employee` and `Titles` classes.

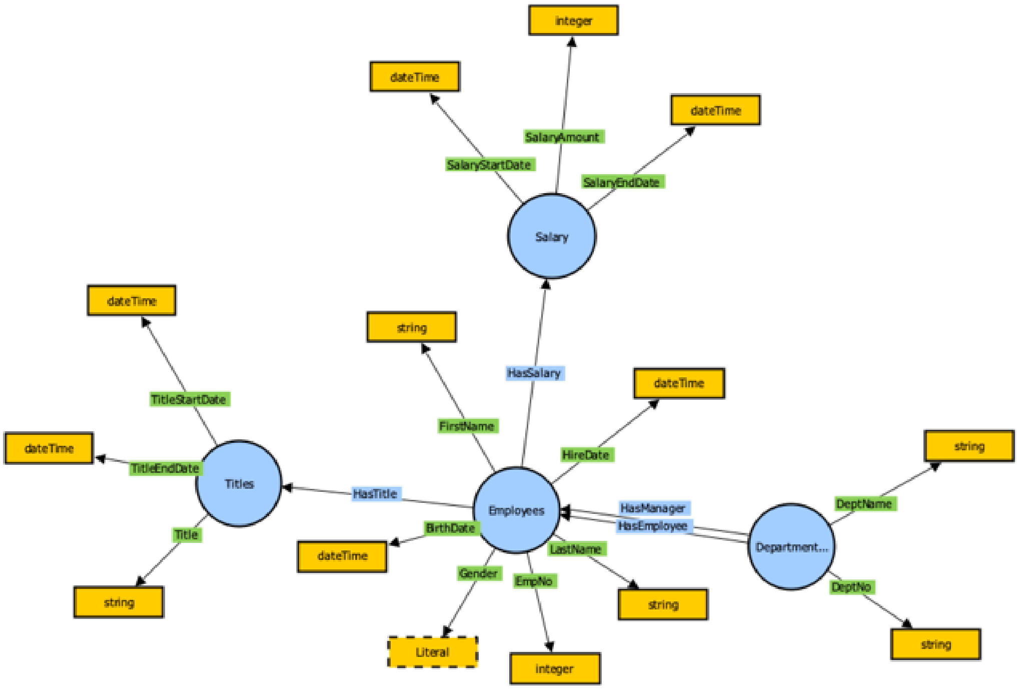

Fig. 6. The general view of the Employee Ontology

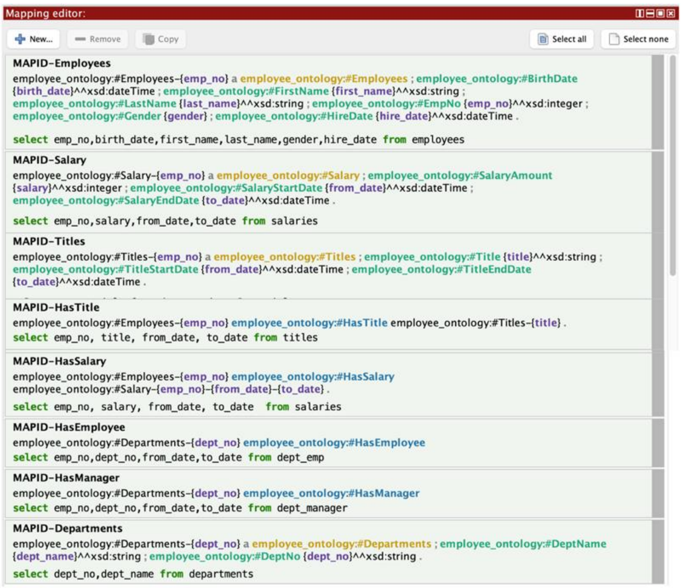

Fig. 7. The Ontop mappings

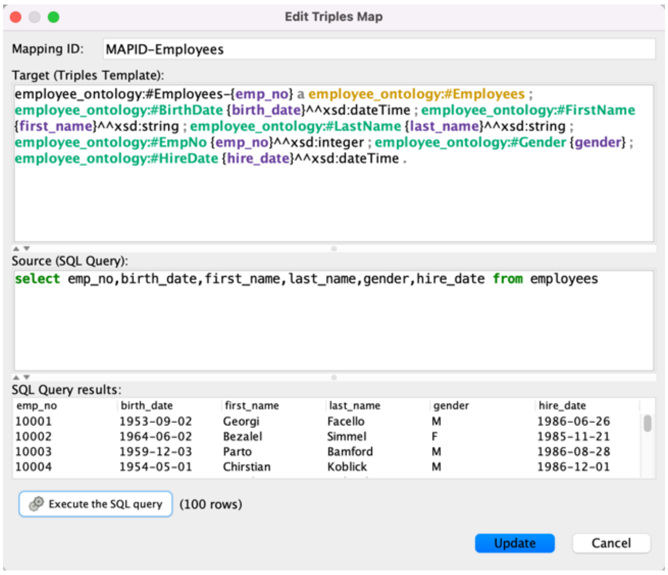

Fig. 8. The mapping of `Employees` table.

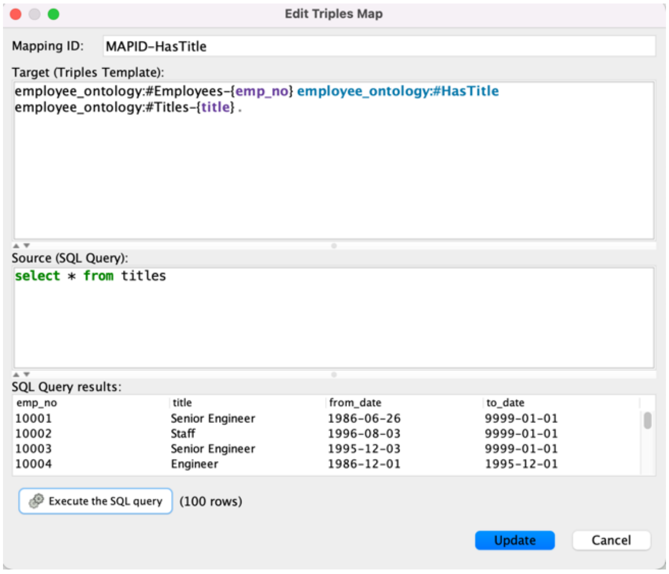

Fig. 9. The Mapping of `HasTitle` object property.

The SPARQL queries from the end user are converted into SQL queries by the Ontop Framework and then executed on the data source. Thus, Ontop uses T-mappings and database integrity constraints to convert input SPARQL queries into SQL queries [4]. The related file "`employee ontology.q`" contains a number of queries that can be used to explore the data collection. The matching SQL and SPARQL sample queries run against the data source are displayed in Tables 1 and 2.

In SPARQL queries, first, all prefix declarations are defined by abbreviating the URIs. Then, the "`SELECT`" statement's result clause is used to determine which data should be returned from the query. Finally, the "`WHERE`" expression specifies the condition while fetching the data from the Employees sample dataset.

The SPARQL and SQL queries given in Table 1 retrieve the first names of all employees from the database. Further, queries given in Table 2 retrieve the salary and employee number of employees whose salaries are bigger than "3000". The results of the related queries are given in Fig. 10 and Fig. 11, respectively.

Table 1. The Ontop Query 1

Get a list of every employee's first name.	
SQL	`SELECT e.first_name` `FROM employees e`

SPARQL	`PREFIX :`
	`<http://www.semanticweb.org/aytug/ontologies/2022/5/employee_ontology#>`
	`Select ?name`
	`where{?x a :Employees . ?x :FirstName ?name .}`

Table 2: The Ontop Query 2

Get all the salary and employee number of employees where salary > 3000	

SQL	`SELECT s.emp_no, s.salary`
	`FROM salaries s`
	`WHERE salary > 3000`

SPARQL	`PREFIX :`
	`<http://www.semanticweb.org/aytug/ontologies/2022/5/employee_ontology#>`
	`Select distinct ?x ?salary`
	`where{?x a :Salary .`
	`?x :SalaryAmount ?salary .`
	`Filter`
	`(?salary>"30000"^^xsd:int)`
	`}`

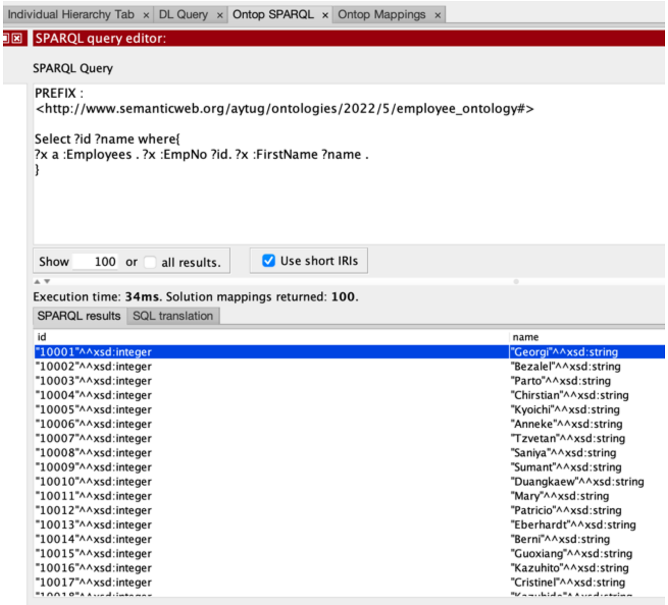

Fig. 10. The query results of Ontop Query-1

IV. CONCLUSION

A mapping between an ontology and a data source is possible using the OBDA paradigm. So, by querying multiple data sources, data access and integration are achieved.. Therefore, data access and data integration are provided by querying heterogeneous data sources. Thus, data integration is a challenging task. In recent years, the OBDA approach has gained significant importance with the increasing amount of data and developments in information technologies. Hence, the mapping between databases and ontologies should be maintained to improve the quality of data integration.

The OBDA allows non-experts to construct queries through conceptual layers using an ontology. For this purpose, the Ontop Framework is used to describe the relational databases as virtual RDF graphs. Also, the Ontop Framework allows querying virtual RDF graphs using RDF query languages like SPARQL and SWRL by translating the queries to SQL.

The Employees sample database, a MYSQL database, is utilized in this study for data integration. The Employee Ontology that is based on the Employees data model is created. For this purpose, the relations and contents in the Employees database are defined in the Employee Ontology. Additionally, the mapping is achieved to connect database entities with the ontology instances. The Ontop Framework and the generated ontology are utilized to allow the usage of SPARQL queries over the dataset. Therefore, data stored in the relational database system is accessed by utilizing the OBDA paradigm. Consequently, this study demonstrates an example semantic data integration for the conceptual representation of data and their relationships. The goal of this demonstration is to be a guide for future research studies in the field.

REFERENCES

[1] Haw, Su-Cheng, Jiawei Wilson May, and Samini Subramaniam. "Mapping relational databases to ontology representation: A review." *Proceedings of the International Conference on Digital Technology in Education*. 2017.

[2] Calvanese, Diego, et al. "How to stay ontop of your data: Databases, ontologies and more." *European Semantic Web Conference*. Springer, Cham, 2015.

[3] Pankowski, Tadeusz. "Modeling and Querying Data in an Ontology-Based Data Access System." *Procedia Computer Science* 192 (2021): 497-506.

[4] Calvanese, Diego, et al. "Ontop: Answering SPARQL queries over relational databases." *Semantic Web* 8.3 (2017): 471-487.

[5] Bagosi, Timea, et al. "The ontop framework for ontology based data access." *Chinese Semantic Web and Web Science Conference*. Springer, Berlin, Heidelberg, 2014.

[6] Can, Özgü, and Murat Osman Ünalir. "Revisiting Ontology Based Access Control: The Case for Ontology Based Data Access." In: *Proceedings of the 8th International Conference on Information Systems Security and Privacy (ICISSP 2022)*, pp. 515-518, 2022.

[7] Ontop. (2009). Retrieved July 10, 2022, from https://ontop-vkg.org.

[8] Rodriguez-Muro, Mariano, Roman Kontchakov, and Michael Zakharyaschev. "Ontology-based data access: Ontop of databases." *International Semantic Web Conference*. Springer, Berlin, Heidelberg, 2013.

[9] Xiao, Guohui, et al. "Ontology-based data access: A survey." International Joint Conferences on Artificial Intelligence, 2018.

[10] Di Pinto, Floriana, et al. "Optimizing query rewriting in ontology-based data access." *Proceedings of the 16th International Conference on Extending Database Technology*. 2013.

[11] Askar, Majid, et al. "Ontology Based Natural Language Queries Transformation into SPARQL Queries." *Baltic Journal of Modern Computing* 8.4 (2020): 719-731.

[12] Ronconi, Giacomo, et al. "The Mastro Protégé Plug-In for OBDA." JOWO. 2017.

[13] Protégé Ontology Editor. (2022). Retrieved July 10, 2022, from https://protege.stanford.edu

[14] Namici, Manuel. "R2RML mappings in OBDA systems: Enabling comparison among OBDA tools." *arXiv preprint arXiv:1804.01405* (2018).

[15] Kalayci, E., et al. "Ontology-based access to temporal data with Ontop: A framework proposal." *International Journal of Applied Mathematics and Computer Science* 29.1 (2019): 17-30.

[16] Employees Sample Database. MySQL. Retrieved July 10, 2022, from https://dev.mysql.com/doc/employee/en/

[17] Lubbad, Motaz AM. *Ontology Based Data Access with Relational Databases*. Diss. Ankara Yıldırım Beyazıt Üniversitesi Fen Bilimleri Enstitüsü, 2018.