

Received March 31, 2020, accepted April 9, 2020, date of publication April 20, 2020, date of current version May 8, 2020.

Digital Object Identifier 10.1109/ACCESS.2020.2988685

Semantic Integration of Heterogeneous Databases of Same Domain Using Ontology

MUHAMMAD ASFAND-E-YAR¹ AND RAMIS ALI

Bahria University, Islamabad 44000, Pakistan

Corresponding author: Muhammad Asfand-e-yar (asfandeyar.buic@bahria.edu.pk)

This work was supported by Bahria University.

ABSTRACT Heterogeneous database integration is the study of integrating data from multiple databases. Integrating the heterogeneous database of the same domain has three main challenges that make the heterogeneity problem difficult to solve. The three problems are Semantic, Syntactic and Structural Heterogeneity. Conventional heterogeneous database integration schemes, like De-duplication Techniques, Data Warehouse, and Information Retrieval (IR) Search technique lack the capability to solve the integration of databases completely. The only reason is they cannot deal with Semantic heterogeneity problems efficiently. The semantic Web ontology model is experimented and discussed in the article, which is based on the query execution model. The ontology modeling is divided into two phases, initially to translate the database rules according to ontology rules to find an abstract ontology model. Secondly, to extend the abstract ontology model according to the databases. The method facilitates to apply similarly SPQRAL queries to search the data in the databases. Therefore, the Jena API is used to retrieve semantically similar records. The experiment is based on the two heterogeneous Universities Library Databases. The results show the effectiveness and scalability of the methodology.

INDEX TERMS Semantic Integration, Semantic Web, Ontology modelling, Library Databases.

I. INTRODUCTION

Semantic Web enables computers to process Web information and has transformed the Web into a medium where different machines can understand each other [1], [2]. Semantic Web helps the automated system to understand, share and process heterogeneous information placed on similar or different machines. Semantic Web technology is supported by the ontology model to classify and integrated the information accordingly. Ontology Models are used to extract implicit and explicit information using SPAQL queries [3]. Ontology models have the capability to integrate heterogeneous data based on the classification and relationship methods (i.e. triples) [4]. Therefore, the ontology model is used to integrate Heterogeneous databases (i.e. structured heterogeneity), as discussed in the article [3], [4]. The structured heterogeneity of databases means that the collection of various databases, storing similar concepts of real-world data but in different structures. The examples of database structures are relational databases, semi-structured databases, and unstructured databases.

The associate editor coordinating the review of this manuscript and approving it for publication was Honghao Gao¹.

Therefore, a common question arises *How efficiently access and share data residing in heterogeneous databases?* In the beginning era only Web Pages and Web Sites were used, where no databases were involved therefore No-SQL was used to extract the information. The Web was used to search for some descriptions or details of anything. Currently, Web Application is using databases. Therefore, various types of Relational Databases are designed, which can be problematic while integrating common data. Most of the Websites are backed by databases, as discussed in [5] 70% of Websites are using databases for storage. Web databases have 500 times more data than static Web. In [5], discusses the importance of databases, therefore integrating heterogeneous databases is an important task, to improve the performance of today's Web applications. Therefore, it is necessary to convert the associated databases into an equivalent ontology to integrate the information [6] resides in the databases.

A Relational Database (RDB) consists of tables that are interlinked with each other based on the same integrity constraints [7]. A table in RDB is a combination of columns (attributes) and rows (tuples). Each tuple represents a specific record. RDB is a combination of static, dynamic or behavioral structures. The static structure includes tables, columns,

data types, primary and foreign keys. Dynamic or behavioral features include functions, procedures, triggers, and views. Only the static structure of databases can be transformed into ontology because the ontology model cannot deal with the behavioral features of databases [8]. Conceptual models are similar for both ontology and database schema. RDB's entities of tables can be used as concepts in an ontology model. The RDB tables have tuples and attributes, integrity constraints, and relationships with other tables, whereas in ontology models, classes have instances and object properties, and restrictions on properties, respectively.

A single query model is required to efficiently access and share the data residing in the heterogeneous databases. Data can be represented in infinite ways in databases, known as Representational Heterogeneity (RH). To solve RH, data is represented in databases using different modeling schemes like relational, hierarchical, flat-file (i.e. unstructured database) or Object-Oriented data modeling schemes. For example, in this research work, two library databases are selected for the experiment. One of the databases is developed according to RDB rules, while the other uses flat-file data model to record similar information such as author-name, book-title, etc. Both databases are storing similar information but in different data models, therefore the representation of data is different in both databases. Hence, storing similar data in different databases have Semantic, Structural and Syntactic heterogeneity. Semantic heterogeneity means that table or attribute names in different databases are the same, but in real life, they refer to different entities or objects. Syntactic heterogeneity (or naming heterogeneity) means that table or attribute names are different in different databases, but they refer to same real-world entities or objects. Structural heterogeneity states that the structure or format of storing data in various databases is different. Conclusively, there always remains some sort of heterogeneity while integrating databases. To fully integrate databases, all such issues should be resolved.

Integrating heterogeneous databases is solved using two methods 1) Data Translation or 2) Query Translation. Through Data Translation technique, data from a database are translated [9]. The translation of data may change the representation of data. While according to Query Translation technique, the queries are translated through native software application to access information from multiple databases [10]. For example, there is a real-time database that frequently gets updated. The requirement of users is to obtain the most frequently updated data from the database. To fulfill such a requirement, either Data Translation or Query Translation can be used. Data Translation translates whole data from the database, and the process takes a lot of time, therefore it is not feasible to translate data after every update in the database. Query Translation, on the other hand, translates user queries into native database query at runtime. The Query Translation takes less time than the Data Translation process and retrieves most recently updated data to the users. Data Translation scheme is preferred for databases where data in the database

does not need many updates. An example of such databases can be a database for storing historical events of a country, as such events are part of history and require no frequent updates.

The advantage of the Data Translation Scheme is its capability of efficiently accessing data stored at a single data source. The disadvantages are 1) Any update in any database requires complex application design to update accordingly the Data Translated Scheme, which is sometimes near to impossible. 2) Data Translation Scheme increases the cost of storing information because same data is stored at various locations, therefore data replication is another disadvantage.

The database records are updated very frequently in a real-time database application. Therefore, the Query Translation technique is used in the research, because it is suitable for real-time database applications if a single platform is used to integrate the databases. Therefore, ontology models [11], [12] are used to integrate heterogeneous databases at a single platform. The ontology model also provides a query language, i.e. SPARQL, to extract the results from an ontology model.

The contribution of this article is to integrate the structural heterogeneous databases at one platform: without retranslating databases or redesigning the existing structure (i.e. RDB) or models (i.e. RDF). The methodology is discussed as, to integrate the various database schemes from RDB tuples to one Web Ontology Language (OWL) model using semantic classification. The advantage of the methodology is its efficiently accessing data without human or system errors, as the user only has to provide a query for accessing records. The user query gets automatically translated using Java platform and required data from different databases is retrieved despite having multiple discrepancies namely Semantic, Syntactic and Structural heterogeneities. Another advantage is that no data translation from multiple databases is required. The data resides only in their respective databases and is not translated into RDF triples (in this approach RDF triples are predefined) which saves a lot of time and space as shown in Figure 1.

The article is composed as in section II discusses the related work and comprehensive analytical table (i.e. Table 1.) describes the objectives of the articles, their proposed solutions and technologies used accordingly. The ontology development and integration are elaborately explained in section III. The results from the ontology model and its analysis is discussed in section IV. Section V summarizes the contributions of the ontology integration model and its benefits.

II. RELATED WORK

Integrating heterogeneous databases is a challenge for organizations and researchers, they are finding better ways to integrate without changing the original architecture of databases. The techniques used to integrate the heterogeneity of databases are the ontology modeling technique, constructing an integrated structure using data mining techniques and federated technique to extract information. Semantic classification is considered mostly by the researchers, to integrate the

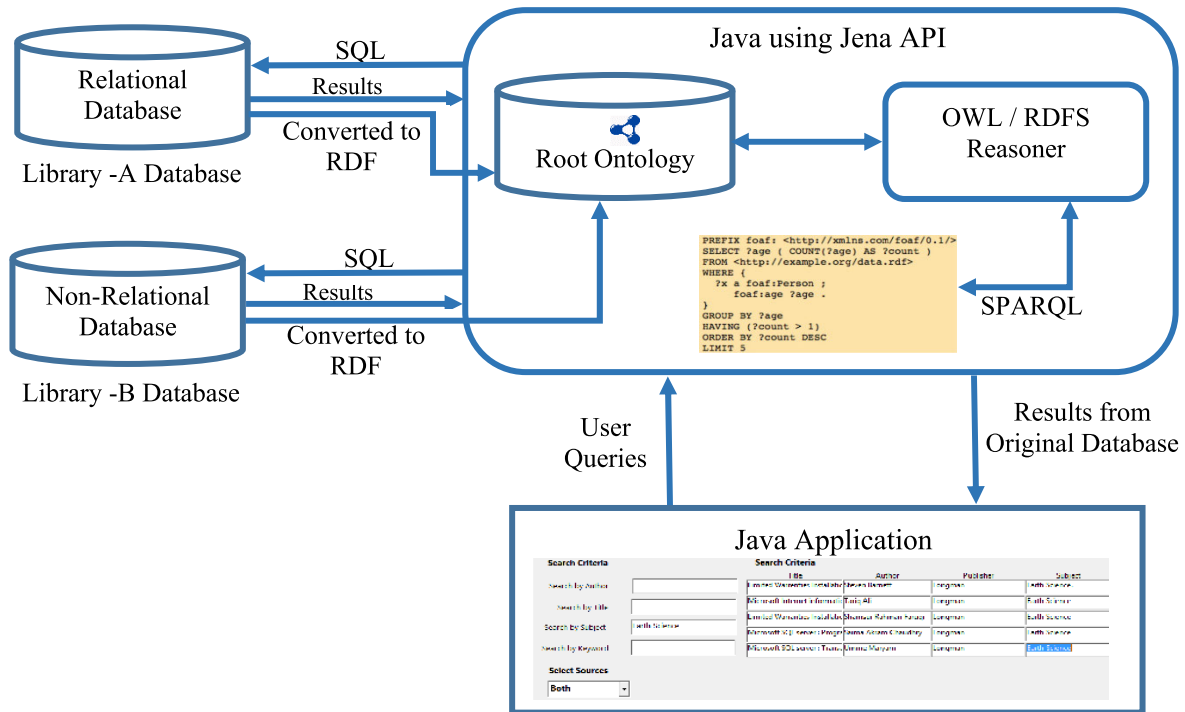


FIGURE 1. Semantic Integration Framework using Java-Jena API.

heterogeneous structured data. Therefore, it is most probable that heterogeneous structured databases also be integrated using semantic classification and ontology modeling.

A. SEMANTIC ONTOLOGY

In [13], the authors discussed an analysis of heterogeneous data sources in Big Data. The focus of the research was on three major problems namely data volume, data from heterogeneous sources, and understanding the relationship between different artifacts. Researchers use metadata to deal with data volume problem. They preferred the Semantic Web Ontology over other techniques to solve the heterogeneity problems, because according to the authors’ opinion, semantic ontology models are suitable for solving the heterogeneity problem. To solve the relationship between different artifacts, Artificial Intelligence techniques were proposed by the authors.

Verification of functional and nonfunctional software requirements is a major task in distributed business processes. Service-based systems provide a new way of integrating business processes. It is an important task to figure out the cost and reliability of service-based systems in the design approach. An approach that discusses such a problem is presented in [14]–[16]. The approach uses PRLTS (Probabilistic Reward Labeled Transition System) model to formally present both functional and nonfunctional requirements. Second, based on transformation rules, functional behaviors of services are generated and visualized using visualization tool Graphviz [49]–[52]. The approach also allows users to dynamically modify the behaviors of the model. Similarly, researchers in [17]–[21] presented a behavioral model to verify data consistency.

A graph modeling approach to integrate data from different heterogeneous sources is discussed in [22]. The authors of [22] used Neo4j as Graph Database model, but also discussed that semantic ontology model is used to solve the heterogeneity problem. The authors in [23] used multiple heterogeneous data sources like Ontologies, Networks, Unified Vocabularies, and Relational Databases to create an integrated network. In [23], a search engine is developed based on semantic ontology model and Machine Learning algorithms that allow users to query multiple data sources. Similarly, a system DBOntoLink is proposed in [24] that enhances the capability of database query languages to work with Biomedical ontology repositories. A unified query interface is proposed in the research that supports major biomedical Ontologies hosted by NCBO BioPortal.

The Semantic Web Techniques are also used to deal with RDB, for example, several RDB schema mapping rules are presented in [5] to map RDB to Semantic Web Ontology (which is Resource Description Framework (RDF)). The rules are helpful in integrating different heterogeneous databases.

In [25], four various results are evaluated to map RDB using semantic ontology models. The authors in [25], analyzed Jena with D2RQ, Jena with R2RML, KAON2, and OWL API. They evaluated the techniques based on two parameters:

- Time required to map the Relational database to ontology
- Total data retrieval time from databases

The authors, in [25], discussed that Jena with R2RML efficiently maps relational databases to ontology, and the

KAON2 retrieves information efficiently, as compared to other techniques.

Apart from mapping techniques, the other techniques were proposed in the database domain to solve the heterogeneity problem. A detailed survey was presented in [26], the survey discusses Global Schema Creation techniques are used to make a Global schema from heterogeneous Local schemas. Global Schema Creation technique is very costly, requires a lot of domain knowledge and needs the databases to be redesigned.

Another technique known as Query Translation based on declarative mappings was presented in [27]. This technique uses a Virtual Query model that gets a query from user and translates the query into many local query models. The local models execute the query according to their syntaxes, return results to the virtual model, and finally, the virtual model displays a unified result obtained from these local models to the user.

Mostly the ontology is developed based on the fields of specific databases. The authors, in [28], presented a scheme (i.e. semi-automatic) to extract ontology from RDB. Therefore, the authors proposed a Reverse Engineering approach that uses SQL Data Definition Language (DDL) as Relational Model and then transformed the model into semantic ontology. In [28], the approach was implemented in two phases. In the first phase, the authors analyzed metadata of the RDB to design a Conceptual Model, then developed an equivalent Semantic Web Ontology accordingly. In the second phase, the data is translated from RDB into RDF triples. The drawback is to repeat the phase every time for retrieving the most recently updated data from a database because database records are updated very frequently. The approach is very costly while processing.

A semi-automatic Semantic Web approach to integrating data from several database sources is proposed in [29]. The ontology model is created in two steps for each data source. In the initial step, the authors developed the ontology model from SQL-DDL. In the second step, the restrictions, object, and data properties are added to refine the initial ontology model. The limitation of [29] is that each time the databases are updated, the data-type properties in the ontology models should be updated accordingly to domains and ranges. Moreover, the mapping rules are not able to map primary key constraint in an ontology.

A set of learning rules are proposed in [30] to extract the OWL ontology model from RDB. In the approach [30], only the RDB metadata is used to create an ontology model. Therefore, no data is translated from database to ontology (i.e. no RDF triples are created from RDB tuples). The main issue of the approach [30] is complex mapping rules like primary and foreign keys are not mapped to ontology since only the metadata of databases is used to create ontology models. As a result, only a small subset of records can be retrieved from databases.

A Global schema-based mapping approach is presented in [31] where a set of mapping rules are defined to extract a

global ontology model from the database. The approach used in [31] is a semi-automatic approach, requires a lot of domain knowledge and is very time-consuming from designing perspective.

The authors, in [32], [33] presented a technique that is quite similar to the work presented by Astrova. [28]. The mapping rules are defined to create an ontology model from SQL-DDL. The proposed approach consists of rules defining different ways of creating classes, subclasses, properties, and restrictions. The authors described that the "Ontology creation process is not completely automated. There are complex databases where the automatic approach may not work, therefore a semi-automatic approach may work well". The drawback of the approaches [28], [32], [33] is that expert domain knowledge is required to apply rules on different data sources, each time, as data sources changes.

An automatic tool is discussed in [34] that creates ontology from RDB schema. The functionality of the tool is its capability to translate a schema of RDB into RDF. The disadvantages of the tool are discussed by the authors, in [34], initially, it can only translate schema and not the data, which makes the tool dependent on a software tool that retrieves data. Secondly, the tool loses some information while mapping complex RDB to RDF.

Another automatic ontology development approach was presented in [35]. The authors named the approach ontology Automatic Generation System (OGSRD) based on RDB. The approach, in [35], is used to develop an Ontology from RDB automatically, but the approach provides limited functionality, and important aspects (i.e. schema) of RDBs are not mapped to ontology.

B. DATA MINING

Authors in [40] discussed that Data Mining algorithms are not able to provide an efficient solution for both the heterogeneity and scalability of Wearable data (i.e. changing data). The authors proposed a Wearable Healthcare Ontology (WH-Ontology) to deal with the heterogeneity problem. The research, in [40], showed that the Semantic Web Ontology efficiently deals with the heterogeneity problem, and make better health-related decisions. The authors used Semantic Web Ontology to store a large amount of data generated from heterogeneous sources and to retrieve data from data sources in a unified format.

C. RETRIEVAL TECHNIQUE

DATA CIVILIZER [39] is a technique proposed in the domain of Big Data to solve the heterogeneity problem. The DATA CIVILIZER uses a graph model to link data from different sources, then uses a data discovery module to help in identifying data that is relevant to the user. The approach uses a poly-store DBMS to execute the actual query which federates query processing across disparate systems.

A hybrid search engine is presented in [37], the search engine uses the Information Retrieval technique known as "Federated Search" to integrate data from heterogeneous

TABLE 1. Analytical description of literature review.

Referenced Articles	Objectives	Metric	Proposed Solution	Techniques used
2016 [5]	Semantic Integration of multiple heterogeneous sources.	Precision and recall.	Semi-automatic approach for Data translation (RDB to RDF).	Semantic Web and ontology modeling.
2016 [13]	Digital Forensic analysis.	Total data reduction and time required to link data from heterogeneous data sources.	Multi-Disciplinary Approach (Use of Metadata, Semantic Ontology and Artificial Intelligence).	Semantic Web and Artificial Intelligence.
2016 [22]	Reduction of schema heterogeneity.	Conciseness and connectivity of graph.	Graph Model and Global Schema approach.	Big Data.
2017 [23]	Integration of medical data from multiple heterogeneous sources.	Restart probability, False positive and True positive.	SLN-SRW (Simplified Laplacian Normalization-Supervised Random Walk).	Semantic Web and Machine Learning.
2015 [25]	Comparative Analysis of state-of-the-art tools for mapping relational database to ontology.	Time required to map relational database to ontology and its efficiency.	No Solution proposed only comparative evaluation discussed.	Semantic Web, time to map ontology.
2001 [27]	Heterogeneous Bio-Medicine Database Integration.	Query efficiency and Data retrieval rate.	Semi-Automatic query translation scheme.	Databases, Biomedical informatics
2004 [28]	Reverse Engineering of RDBs to ontology.	Total mapping time and its efficiency.	Reverse Engineering, Semi-automatic approach and Data translation approach.	Semantic Web, ontology modeling.
2005 [30]	Populating Semantic Web ontology from RDBs.	Total mapping time.	Semi-automatic approach and Query Translation.	Semantic Web, time to map ontology.
2006 [31]	Mapping rules to map Relational model into OWL.	Precision and mapping time.	Semi-automatic data translation.	Semantic Web, auto ontology mapping.
2011 [33]	Extending RDBs to support Semantic Web queries.	Extending time efficiency and query efficiency.	Direct mapping.	Semantic Web Technology.
2010 [34]	Translating RDB into Semantic Web ontology.	Total translation time precision.	Automatic approach and Query Translation.	Ontology modeling and query application.
2011 [35]	Generating ontology from database automatically.	Ontology generation time efficiency.	Automatic approach.	Semantic Web, automatic ontology modelling.
2017 [36]	Reduction of semantic heterogeneity.	Integration efficiency and interoperability.	Semantic Big Data Historian (SBDH) & Global schema approach.	Cyber Physical systems, Bigdata, & semantic Web.
2016 [37]	A platform that enable users to search integrated data from multiple sources	Recall and precision.	Fuhsen (Hybrid Search Platform) and Global Schema approach.	Deep Web and linked data.
2015 [38]	Handling information from multiple sources.	Primary vertical CTR and Secondary vertical CTR.	Data driven approach.	Information Retrieval and Machine Learning.
2017 [39]	Data integration from multiples data sources.	Time required to find, prepare and analyze data.	DATA CIVILIZER and Global Schema approach.	Big Data management system.
2015 [40]	To deal with heterogeneity and scalability problem in wearable health devices.	Time required to extract data from heterogeneous data resources and its scalability.	Semantic Big Data architecture based on "Knowledge as a Service", wearable Health Care ontology and Query Translation approach.	Wearable computing, Big Data and Semantic Web Technology.
2014 [41]	Exporting RDB contents into RDF graphs.	Exporting time efficiency.	R2RML.	Web Intelligence, Mining and Semantics.
2008[42]	Transforming SQL RDBs into OWL ontology models.	Total mapping time efficiency.	OWL API.	Metadata and Semantic research.
2006 [43]	Comparing and reasoning techniques for querying DL.	Query effectiveness and precision.	KAON2.	Artificial Intelligence and reasoning.
2004 [44]	Treating RDB as virtual RDF graphs.	Precision and recall.	D2RQ.	Semantic Web Technology.
2014 [53]	Heterogeneous RDB mapping with Ontologies automatically.	Single Ontology (i.e. upper Ontology concept) is used.	Multi ontologies are combined in a single ontology	Ontology Engineering.
2009 [54]	Solving RDB query related problems.	SPARQL patterns are used to integrate the RDB queries.	To integrate queries the RDB schema is model in ontology.	SPARQL patterns and Ontology Modeling.
2003 [55]	Integration of same domain RDB schema having same field but having different relationship.	Semantic modeling to solve semantic heterogeneity.	Semantic graphs are used to solve the relationship of fields among the databases.	Ontology Engineering.
2014 [56]	Integrate the databases where data is not centralized and has schema variations.	Semantic modeling to solve the problem.	Ontology models integrates the data that has schema variations.	Ontology Engineering.
2009 [57]	Integrating the three heterogenous groups of databases for timestamp.	Ontology model is constructed.	Semantic ontology model solves the replication of data in timestamp.	Ontology modelling.
2010 [58]	Integrating RDB schema of same domain designed in SQL Server, Oracle, etc.	Semantic ontology model is used.	An ontology is modelled to solve the semantic and structural conflicts of RDB schema	Ontology modelling.

information sources. In Federated Search, a query is sent to multiple data silos (i.e. the bulk of data) such as Google Search, Oracle, etc., and data is retrieved. The retrieved data from silos are combined and shown to the user. Using the technique, a user can query multiple heterogeneous databases at once. Researchers use Precision and Recall metrics to measure the efficiency of the proposed approach [48]. The proposed scheme is based on the concept of the Global Schema approach, in which a schema or a set of rules are defined. Based on these rules all silos must be designed. In other words, all silos must comply with the global schema. One of the main disadvantages of the Global schema approach is that silos cannot be designed independently. Data cannot be retrieved from any data source that was not designed based on globally defined rules. Another main drawback of Federated Search is that it cannot deal with semantic heterogeneity problem, because Federated Search is not designed for solving the semantic heterogeneity problem. The Federated Search is based on a Query Translation approach therefore query is applied on attributes of tables in databases. If the attribute titles are different but the meanings are the same then the Federated Search cannot retrieve actual results from multiple databases simultaneously, while semantic ontology works efficiently in such cases as discussed in section III.

D. SHORTCOMINGS ACCORDING TO OUR APPROACH

In the article [28], the authors had used the semantic ontology methods to do reverse engineering for extracting ER diagrams and object models from existing relational databases. The authors had analyzed the relations of data and constructed the ontology model. Therefore, their model is based on the RDB database of Web applications. In the article [30], [31], the authors defined a set of semantic language rules that learn the RDB and automatically modeled the ontology. Their focus is on the automatic system that can visualize the RDB in the ontology model. The automatic system can only design the ontology for RDB databases. In the article [31], the authors proposed a semi-automatic system, while in [30] the authors proposed an automatic system. The article [33], presented a DLDB, which a DAML+OIL extension of ontology based on the RDB i.e. only MS-Access databases. The author used a reasoner and define the description logic to provide a solution for database complexity. In [34], [35], the authors proposed a tool that translates automatically the RDB to an ontology language (i.e. model). In [34], the tool obtains the implicit semantics of RDB for translation, while in [35], initially an ontology is constructed based on RDB and then experimented to translate the RDB into ontology model. The article [41] also explains the RDB into a graph but with the incremental method. The incremental method explains that mapping is only performed when the set of performance measurements are evaluated according to requirements. In [42], the authors developed an ontology model to map the RDB for synchronizing the SQL queries on scattered databases. The article [43] describes the construction of a logical description of RDB. The logical description is based

on the KAON2 and SHIQ, which is named as ABoxes. While in [44], the authors implemented the D2RQ that is developed from RDB and the SQL queries are model in the RDF graph.

The shortcomings of the work [28], [30], [31], [33]–[35], [41]–[44] are not integrating the structured and unstructured data from various databases. The objectives and metric of these references are explaining that they are only focusing the same domain of database but structure i.e. RDB. While our work integrates the RDB (i.e. structured) and semi-RDB (i.e. semi-structured) or non-RDB (i.e. completely-unstructured) databases.

The article [53] describes the mapping of ontologies from various heterogeneous RDB, automatically. The authors presented the solution for heterogeneous RDB databases, but not dealing with unstructured databases where the syntactical problem occurs. The syntactical problem is addressed in section III and V. The article [54] discusses the query related problems to integrate the relational databases. The authors suggested that SPARQL graph patterns give the same principle for mapping the relational schema using RDF ontology. The author presented that their work optimizes the database query using SPARQL query where the primary keys and other fields are mapped with the ontology model. The article [55] discusses the integration of the databases but with similar domain and similar field/columns name. The problem, discussed in [55], focuses on the problem of semantic heterogeneity, which explains in detail the meaningful heterogeneity, where the databases have some data and almost field names, but the relationship is different. The authors constructed an ontology model for two different databases accordingly to solve only structural heterogeneity. In the article [56] authors discuss their project to integrate the databases where the data are presented recurrently at various terminals. The efficiency of the methodology is to use simple SPARQL queries for data retrieval without using complex database queries, therefore only semantic query problem is solved. The article [57] discusses three different groups of heterogeneous and homogenous replication of data in DBMS. These groups describe the time stamping of DBMS when any type of update, change, modify and/or access to DBMS is performed. The authors in the article [57] used the semantic ontology models to solve the problem of replication of data while time stamping in the DBMS. In another article [58] supports our methodology that a top-level ontology (i.e. Root Ontology in our article) eliminates the conflicts and uncertainty in ontology mapping. It is also mentioned in the article [58] that their ontology solves semantic and structural conflicts. The article [59] discusses the structured RDB semi-automatic mapping on ontology but only for traditional Chinese medicine databases. In [60], the authors proposed a hierarchical ontology model to map the structured RDB extracted features at the RDF model. The article [61] is about a survey report that contributes the integration to RDB into RDF semantic technology. The contributes is described as around 0.33% approaches are unsuccessful while mapping ontology either automatically or semi-automatically.

Our approach can solve the syntactical conflict as well as semantic and structural conflicts, because it is focusing on the unstructured databases as well as the RDB, where majority problems occur in the field names, as discussed in section III, that similar records are maintained under the different fields, where the field names are not matching. The problem compared to our methodology is that the articles [53]–[60] are using an ontology model for RDB. While we have proposed the methodology to construct a stepwise ontology model that can be easily integrated with various other relational and/or unstructured databases. Using our method, the Root Ontology can be extended to other similar database problems, which is not possible with [53]–[60]. The novelty of our methodology is the Root Ontology that integrates entirely different databases i.e. relational and unstructured databases. Our methodology also addresses the syntactical problem, i.e. with different names of fields in the different databases but data in the fields are meaningfully similar. Our work is based on the approach presented in [35].

The techniques of integrating heterogeneous data sources are very good, and they are being used in well-known applications. For example, LinkedIn and Metasearch Engine use Federated Search, but there are some limitations to these techniques, discussed in section II.C second paragraph.

In our article, three heterogeneity challenges are discussed in section V namely Semantic, Structural, and Syntactic (i.e. Naming) heterogeneity. According to the discussion in section II, the problem is addressed by proposing the Semantic Web approach to improve the search techniques [38]. Therefore, section II, explains that Semantic Web Ontology provides the best solution for heterogeneity problem. The analytical discussion is elaborated in Table 1. The articles presented as the analytical discussion in Table 1 are not discussing the syntactic heterogeneity problem which occurs during the integration of data between heterogeneous databases.

III. ONTOLOGY DEVELOPMENT AND INTEGRATION

Semantic Web Ontology provides a solution to integrate heterogeneous databases, as discussed in section II. Hence, the methodology is discussed in section III. The methodology to develop the ontology and integrating database consists of three main phases: 1) building ontology from RDBs, 2) integrating ontology, 3) accessing data from heterogeneous databases based on integrated ontology models. Initially, RDBs are mapped to the ontology model based on ontology mapping rules discussed in section III.A. In the initial phase, two ontology models according to two different Library databases are developed. In the second phase, a Root Ontology is created, based on the two developed ontology models. The Root Ontology is an abstract ontology model of the ontologies developed in the initial phase. Then the initial models are integrated with the Root Ontology. Finally, the integrated ontology models are used to access data from heterogeneous databases simultaneously. Different tools such as TODOE (Tool for Ontology Development and Editing),

Protégé and Jena API in Java Eclipse can be used to develop and edit ontology models. TODOE is one of the earliest Dot Net based ontology development tool. TODOE provides most of the functionalities like reasoning, interfacing, editing, developing and visualization, but supports only OWL-Lite [47]. Protégé, on the other hand, provides all the above-mentioned functionalities, and besides, provides support for OWL DL and OWL Full. Based on these exceptions in TODOE, the ontology models are created using Protégé, and Jena API in Java Eclipse. The Java Eclipse is used only for GUI interfacing, if we don't use the Java Eclipse for Protégé then the same queries will be applied manually at Protégé for extracting answers also to update the ontology at any level, will be done manually. Our methodology is efficient to access data using Semantic techniques from multiple heterogeneous databases simultaneously or according to requirement.

The Library databases are the real-time databases acquired from two different Universities. Both Universities use Integrated Library Database Management Systems (ILS) to manage their records. The ILS of first University is implemented using SQL Server, and RDB schema is used (i.e. Library-A) as shown in Figure 2. The second University ILS is based on MySQL (backhand KOHA) to manage ILS, but no RDB schema is used (i.e. Library-B). Therefore, no relationships were defined between the tables in Library-B, the Library-B consists of the unstructured database. Both ILSs are using different database management schemes to store similar information about books, students, staff and other related entities, but data is heterogeneous in nature.

A. ONTOLOGY DEVELOPMENT

The conceptual model of both databases and ontology is quite similar. Databases have a set of static and dynamic attributes, but only the static features of databases are mapped to semantic ontology because of ontology restrictions. The static features of databases include tables, columns, rows, relationships, integrity constraints, primary keys and foreign keys. Converting an RDB into ontology is a direct process, and complex conversion case hardly occurs during conversion. To construct ontology models from RDB, two set of rules are used as discussed in following.

1) SIMPLE TRANSLATION RULES

The RDBs are translated to an ontology model by mapping the database tables as the ontology classes, the primary keys, non-primary keys and non-foreign keys are mapped as data property of a class. Each data property has a corresponding table name as its domain and SQL equivalent XML Schema datatype as its range. Table 2 shows the SQL equivalent XML Schema Datatypes (xsd).

Table 2 shows simple conversion rules of mapping databases with ontology's datatypes. For mapping complex functionalities like relationships, primary and foreign keys, the mapping rules proposed by authors in [41], are used accordingly. The same method is applied to both databases to generate two separate local ontology models (one for each

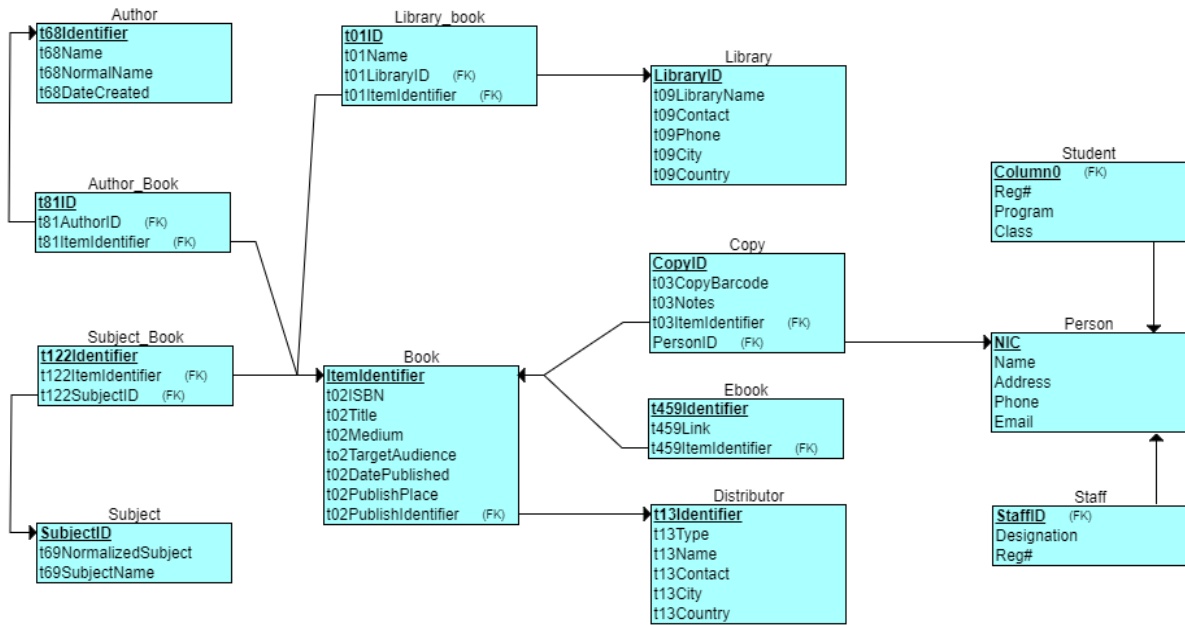


FIGURE 2. Relational Schema of Library-A database.

database). The Relational schema of the Library-A database is shown in Figure 2.

2) COMPLEX CONVERSION RULES

While translating the RDB to RDF Ontology, the many-to-many relationship in RDB is resolved by normalization (i.e. introducing another table). Therefore, due to normalization, the entities (i.e. fields name in RDB) are repeated. The repetition of entities is not possible in ontology, because in ontology a class is a URI, which should be unique. Hence, to solve such problems following techniques are used, which are explained with examples.

The database tables from both libraries are mapped to ontology classes accordingly, except the joint tables, which are used to connect two tables, in the case of a many-to-many relationship. For example, *Author* and *Book* tables in Figure 2 have a many-to-many relationship.

If two tables have a many-to-many relationship in databases, then the relation is divided into one-to-many, and many-to-one relations using a third table. It can be seen from Figure 2 that *Author_Book* table is used to relate parent tables (*Author* and *Book*). The Primary keys of both parent tables are used as foreign keys in *Author_Book* table. Therefore, *Author_Book*, *Library_book*, and *Subject_book* tables are not mapped to ontology classes, because the field names are already mapped as classes in ontology. It is also because there is no need for cardinality maintenance in the ontologies.

The database tables whose attributes at the same time are primary key and foreign key, they are made subclasses of those classes whose primary keys are used as foreign keys. Figure 2 shows two such tables. *Staff* and *Student* tables have primary keys (i.e. *StaffID* and *Column0* accord-

TABLE 2. SQL equivalent XSD datatypes.

SQL Type	XSD type
BigInt	xsd: long
Binary	xsd: base64Binary
Bit	xsd: Boolean
Char	xsd: string
Date Time	xsd: dateTime
Decimal	xsd: decimal
Float	xsd: double
GUID	xsd: string
Image	xsd: base64Binary
Int	xsd: int
Money	xsd: decimal
NChar	xsd: string
NText	xsd: string
Numeric	xsd: decimal
NVarChar	xsd: string
Real	xsd: float
SmallInt	xsd: short
SmallDateTime	xsd: dateTime
SmallMoney	xsd: decimal
Sql_Variant	xsd: anyType
Text	xsd: string
TimeStamp	xsd: base64Binary
TinyInt	xsd: unsignedByte
UDT (CLR style)	xsd: base64Binary
VarBinary	xsd: base64Binary
VarChar	xsd: string
XML	xsd: any
XML (typed)	xsd: any

ingly) that at the same time are foreign keys as well. Therefore, these tables are made subclasses of *Person* class in ontology, because primary key (i.e. *NIC*) of *Person* table is used as a foreign key (i.e. *StaffID* and *Column0*) in both

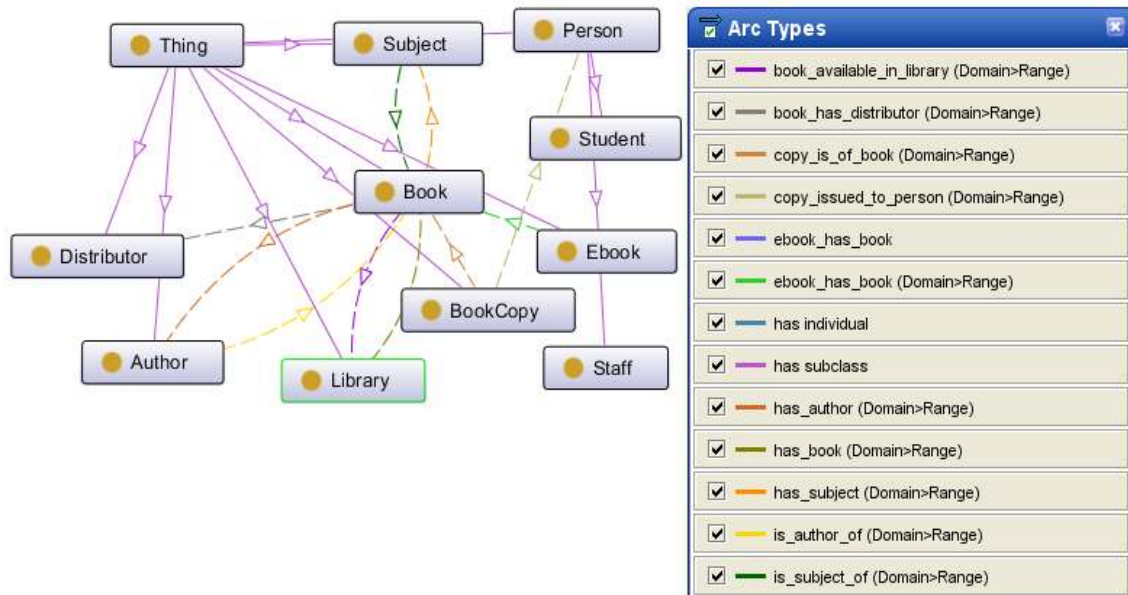


FIGURE 3. Library-A Ontology model.

tables. The connecting database tables used for many-to-many relationships are mapped to object properties in RDF ontology. For example, *Library_book* is a junction table, as shown in Figure 2, which is used to connect the *Library* and *Book* tables having a many-to-many relationship. Therefore, in ontology the two object properties “*has_book*” and “*book_available_in_library*” are defined in the ontology model to solve the many-to-many relationship problem. The first object property (i.e. “*has_book*”) has a “*Library*” class as its domain and “*Book*” class as its range. Whereas the second property (i.e. “*book_available_in_library*”) is used as a reverse object property. Therefore, the “*Book*” class as domain, and “*Library*” class as its range. Figure 3 shows the ontology model of the Library-A database created using these rules. The same set of rules is applied to Library-B ontology. The ontology model of Library-B is shown in Figure 4. The Library-B database is an unstructured database, therefore, there are no relationships among the entities. Hence, the ontology model of Library-B has only classes with data properties.

B. ONTOLOGY INTEGRATION

Ontology integration is the second phase of our methodology. A Root Ontology is created in the second phase, using Protégé, based on domain knowledge of libraries, and is shown in Figure 5. The Root Ontology consists of a collection of generic classes of Library-A ontology model and Library-B ontology model, as shown in Figure 3 and Figure 4 respectively. The main purpose of Root Ontology is to integrate the ontology models created in phase-1. To integrate Root Ontology with phase-1 ontology models, the Root Ontology is imported separately in different ontology for Library-A and Library-B, using Protégé. After importing, the following

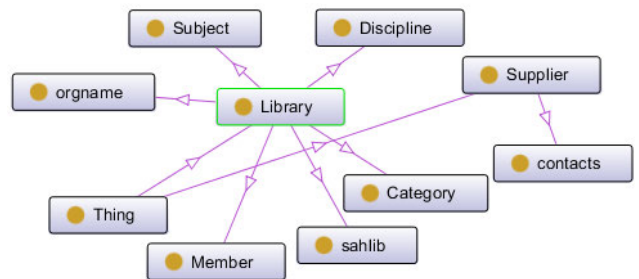


FIGURE 4. Library-B Ontology model.

steps are followed to integrate the Library-A and Library-B ontology models:

- If class names of both Root and phase-1 ontology models refer to the same real-world entity.
 - Assign the same object and data properties of phase-1 ontology while constructing the class in Root Ontology.
 - Then there is no need to include the classes and properties accordingly in the imported ontology of Library-A and Library-B of phase-2.
- If class names of Root and phase-1 ontology models are the same, but representing different real-world concepts the
 - Find the most appropriate class in Root Ontology which refers to the same concept as the class of phase-1 ontology and assign its properties to that class. Otherwise, construct a new class (the class name should be meaningfully defined accordingly).

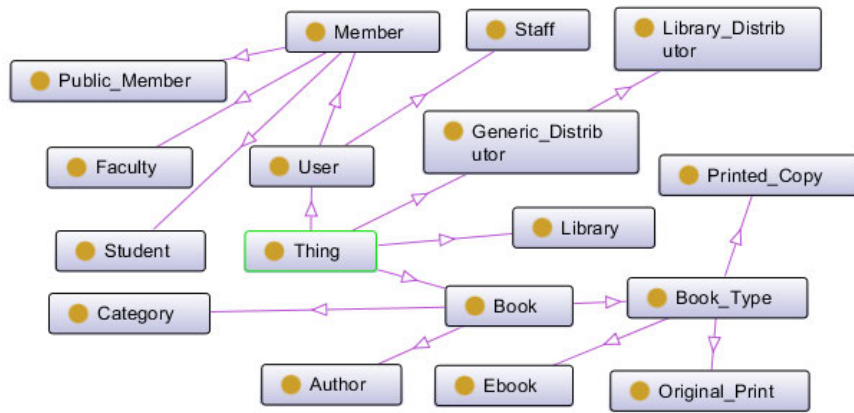


FIGURE 5. Root Ontology.

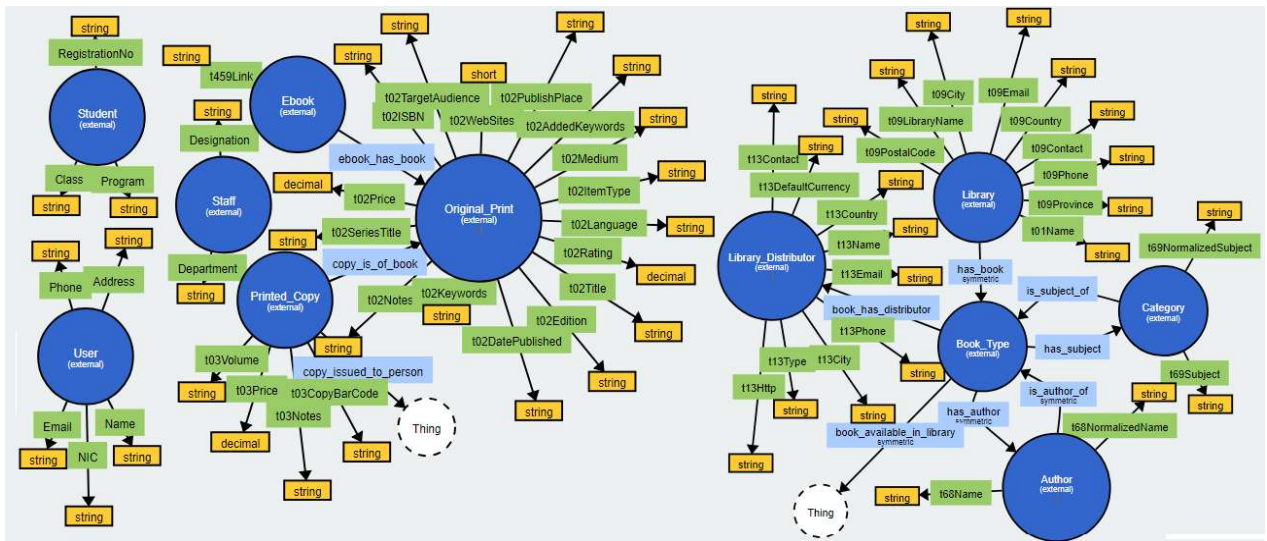


FIGURE 6. Integrated Ontology model of Library-A at phase-2.

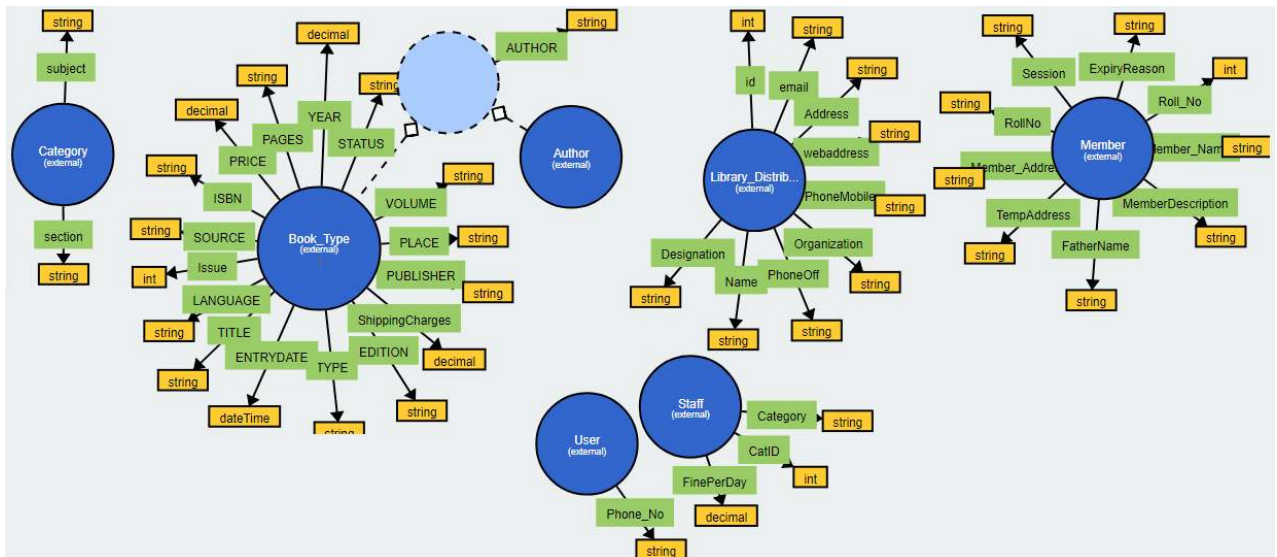


FIGURE 7. Integrated Ontology model of Library-B at phase-2.

```
SPARQL query:
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX owl: <http://www.w3.org/2002/07/owl#>
PREFIX xsd: <http://www.w3.org/2001/XMLSchema#>
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
PREFIX
ex: <http://www.semanticweb.org/ramisali/ontologies/2017/7/untitled-ontology-28#>
SELECT ?book ?author
WHERE {
    ?author ex:is_author_of ?book.
    ?author ex:t68Name "Alan Simpson"^^xsd:string
}
book author
Your_First_Computer Alan_Simpson
```

FIGURE 8. SPARQL query to display Book of author “Alan Simpson”.

```
SPARQL query:
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX owl: <http://www.w3.org/2002/07/owl#>
PREFIX xsd: <http://www.w3.org/2001/XMLSchema#>
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
PREFIX
ex: <http://www.semanticweb.org/ramisali/ontologies/2017/7/untitled-ontology-28#>
SELECT ?distributor ?book
WHERE {
    ?book ex:book_has_distributor ?distributor.
    ?distributor ex:t13Name "Pak Book
Corporation"^^xsd:string}
distributor book
Pak_Book_Corporation Your_First_Computer
```

FIGURE 9. SPARQL query to display distributor name of a Book.

- Then there is no need to include the classes and its properties in the imported ontology at phase-2.

- If a class in phase-1 ontology models has no equivalent class in Root Ontology model, the
 - Keep the same class accordingly in the ontology (it can be imported ontology for Library-A or Library-B accordingly), except to include the class in the Root Ontology. This explains that do’t remove the class form the ontology, which is defined in the phase-1 ontology model.

The integration phase is based on two ontology models. These models are referred to as phase-2 Ontology models, as shown in Figure 6 and Figure 7. To understand the relationships and roles of classes and individuals used in the ontology models (at phase-2), the Description Logic(s) of both ontology models are provided in Table 3 and Table 4. For example, the Description Logic of $Author \subseteq Book$ explains that a Book has Authors; the author can be one or many, as shown in Table 3 and Table 4. Description Logic describes the favorable trade-offs between the expressivity and scalability of the Root Ontology. Therefore, the same Root Ontology (i.e. presented in this article) can be extended to other database models of libraries, with minor changes. As shown in Table 3 and Table 4, the class names and the relationships of ontology models for Library-A and Library-B are now similar at phase-2, which were different at phase-1 ontology models. The rules and logical description of Table 3 and Table 4 are based on the Library-A and Library-B databases accordingly. The similar-

TABLE 3. Ontology description logic of library-A at phase 2.

Sub Class Relations	
Book	$Author \subseteq Book$
	$Category \subseteq Book$
Object Properties	Description Logic
has_author	\exists has_author.Thing \subseteq Book_Type $T \subseteq \forall$ has_author.Author
is_author_of	\exists is_author_of.Thing \subseteq Author $T \subseteq \forall$ is_author_of. Book_Type
Data Properties	Description Logic
t68Name	\exists t68Name.Thing \subseteq Author $T \subseteq \forall$ t68Name.string
t68NormalizedName	\exists t68NormalizedName.Thing \subseteq Author $T \subseteq \forall$ t68NormalizedName.string
Sub Class Relations	
Member	\subseteq User
Object Properties	Description Logic
copy_issued_to_person	\exists Copy_issued_to_person.Thing \subseteq Printed_Copy $T \subseteq \forall$ Copy_issued_to_person.User
Data Properties	Description Logic
Address	\exists Address.Thing \subseteq User $T \subseteq \forall$ Address.String
Email	\exists Email.Thing \subseteq User $T \subseteq \forall$ Email.String
Name	\exists Name.Thing \subseteq User $T \subseteq \forall$ Name.string
NIC	\exists NIC.Thing \subseteq User $T \subseteq \forall$ NIC.String
Phone	\exists Phone.Thing \subseteq User $T \subseteq \forall$ Phone.String
Data Properties	Description Logic
Class	\exists Class.Thing \subseteq Student $T \subseteq \forall$ Class.String
Program	\exists Program.Thing \subseteq Student $T \subseteq \forall$ Program.string
RegistrationNo	\exists RegistrationNo.Thing \subseteq Student $T \subseteq \forall$ RegistrationNo.string
Data Properties	Description Logic
Department	\exists Department.Thing \subseteq Staff $T \subseteq \forall$ Department.string
Designation	\exists Designation.Thing \subseteq Staff $T \subseteq \forall$ Designation.string

ity of class and their relationships in the two ontology models of the libraries are required, to apply semantically similar SPARQL queries for retrieving data from both databases. The only difference is the object properties of both ontology models, which are not similar because the structure and schema of both databases are different. Hence, in the procedure (experimented in this article), the data of the databases are not translated but instead, the Query Translation technique is used. Query Translation is more feasible in case of providing the most recently updated data to users, as compared to Data Translation.

Ontology integration is an important phase of our research. Due to the integration phase (i.e. phase-2), the Semantic, Syntactic and Structural heterogeneity issues are removed from ontology models based on rules discussed in bullets section III.B. The results of phase-2 integrated ontology models are semantically similar to a greater extent because similar SPQRQL queries are applied on both models to retrieve results from the heterogeneous databases of Library-A and Library-B.

C. QUERY TRANSLATION SCHEME

Query Translation is the last step. To implement the translation, Jena API is used in Java Eclipse. Therefore, in Java Eclipse, “mysql.jdbc” JDBC driver for MySQL and

Search Criteria		Search Criteria			
		Title	Author	Publisher	Subject
Search by Author	<input type="text"/>	Limited Warranties Installatic	Steven Barnett	Longman	Earth Science.
Search by Title	<input type="text"/>	Microsoft Internet informati	Tariq Ali	Longman	Earth Science
Search by Subject	Earth Science	Limited Warranties Installatic	Shamsur Rahman Faruqi	Longman	Earth Science
Search by Keyword	<input type="text"/>	Limited Warranties SQL Servi	Jay Asher	Longman	Earth Science
		Administration Companian	Arthur O'Sullivan	Longman	Earth Science
Select Sources					
Library A					

FIGURE 10. Library-A results.

Search Criteria		Search Criteria			
		Title	Author	Publisher	Subject
Search by Author	<input type="text"/>	Microsoft SQL server : Progr	Saima Akram Chaudhry	Longman	Earth Science.
Search by Title	<input type="text"/>	Microsoft Internet informati	Tariq Ali	Longman	Earth Science
Search by Subject	Earth Science	Programming SQL Distribute	James Henry translated by	Longman	Earth Science
Search by Keyword	<input type="text"/>	Limited Warranties SQL Servi	Jay Asher	Longman	Earth Science
		Microsoft SQL server : Trans	Umme Maryam	Longman	Earth Science
Select Sources					
Library B					

FIGURE 11. Library-B results.

Search Criteria		Search Criteria			
		Title	Author	Publisher	Subject
Search by Author	<input type="text"/>	Limited Warranties Installatic	Steven Barnett	Longman	Earth Science.
Search by Title	<input type="text"/>	Microsoft Internet informati	Tariq Ali	Longman	Earth Science
Search by Subject	Earth Science	Limited Warranties Installatic	Shamsur Rahman Faruqi	Longman	Earth Science
Search by Keyword	<input type="text"/>	Microsoft SQL server : Progr	Saima Akram Chaudhry	Longman	Earth Science
		Microsoft SQL server : Trans	Umme Maryam	Longman	Earth Science
Select Sources					
Both					

FIGURE 12. Results from both i.e. Library-A and Library-B.

“sqljdbc” driver for SQL Server are used. The Jena API is used in Java to create or manipulate RDF graphs. The ontology models are already created in Protégé; therefore, the ontology models are only used to apply SPARQL queries using Jena API. Various packages of Jena API such as “Ont-Model” class to translate an ontology model, “Query data structure” to create a SPARQL query, “Query Execution” interface to execute SPARQL query are used in the translation phase.

To retrieve data from MySQL and SQL Server Databases, “mysql.jdbc” and “sqljdbc” drivers are used. The connections are established with the corresponding databases. The SPARQL queries are categorized according to the native

database queries. The queries are executed, and results are collected from databases for displaying to the user. Three different functionalities are provided to users through which a user can query Library-A or Library-B databases independently, or simultaneously as shown in Figure 10, Figure 11 and Figure 12.

IV. ANALYSIS AND RESULTS

The contribution of the research is to integrate heterogeneous databases at a single platform (i.e. the proposed ontology model), without changing the structure and schema of the existing database. To perform the task, actual library databases from two Universities are selected. After analyzing

TABLE 4. Ontology description logic of library-B at phase 2.

Book	Sub Class Relations	
	Author	$\text{Author} \sqsubseteq \text{Book}$
	Category	$\text{Category} \sqsubseteq \text{Book}$
Author	Data Properties	Description Logic
	Author	$\exists \text{ Author.Thing} \sqsubseteq \text{Author}$ $\text{T} \sqsubseteq \forall \text{ Author.string}$
User	Sub Class Relations	
	Member	$\text{Member} \sqsubseteq \text{User}$
User	Data Properties	Description Logic
	Phone_No	$\exists \text{ Phone_No.Thing} \sqsubseteq \text{User}$ $\text{T} \sqsubseteq \forall \text{ Phone_No.string}$
Member	Sub Class Relations	
	Faculty	$\text{Faculty} \sqsubseteq \text{Member}$
	Public_Member	$\text{Public_Member} \sqsubseteq \text{Member}$
	Student	$\text{Student} \sqsubseteq \text{Member}$
Member	Data Properties	Description Logic
	ExpiryReason	$\exists \text{ ExpiryReason.Thing} \sqsubseteq \text{Member}$ $\text{T} \sqsubseteq \forall \text{ ExpiryReason.string}$
	FatherName	$\exists \text{ FatherName.Thing} \sqsubseteq \text{Member}$ $\text{T} \sqsubseteq \forall \text{ FatherName.string}$
Member	Member_Address	$\exists \text{ Member_Address.Thing} \sqsubseteq \text{Member}$ $\text{T} \sqsubseteq \forall \text{ Member_Address.string}$
	Member_Name	$\exists \text{ Member_Name.Thing} \sqsubseteq \text{Member}$ $\text{T} \sqsubseteq \forall \text{ Member_Name.string}$
	MembeDescription	$\exists \text{ MembeDescription.Thing} \sqsubseteq \text{Member}$ $\text{T} \sqsubseteq \forall \text{ MembeDescription.string}$
Member	Roll_No	$\exists \text{ Roll_No.Thing} \sqsubseteq \text{Member}$ $\text{T} \sqsubseteq \forall \text{ Roll_No.int}$
	Session	$\exists \text{ Session.Thing} \sqsubseteq \text{Member}$ $\text{T} \sqsubseteq \forall \text{ Session.string}$
Staff	Data Properties	Description Logic
	Category	$\exists \text{ Category.Thing} \sqsubseteq \text{Staff}$ $\text{T} \sqsubseteq \forall \text{ Category.String}$
	CatID	$\exists \text{ CatID.Thing} \sqsubseteq \text{Staff}$ $\text{T} \sqsubseteq \forall \text{ CatID.int}$

both databases, it is cleared that both databases are storing similar information, but the databases' structure, schema, technical naming of fields, semantically and syntactically are different. Library-A Management System was implemented in SQL Server and is an RDB. The Library-B database is not an RDB, and the information is redundant in tables. In Library-B there are no primary or foreign keys in the database, which is the main cause of data replication.

Some major differences between both databases are identified during research for example, 1) Technical differences: Library-A database is implemented in SQL Server, and the Library-B is implemented in MySQL. 2) Structural differences: Library-A database had foreign keys i.e. relations with other tables, and the Library-B database had no relations. 3) Semantic and Syntactic differences: tables in both databases are storing similar information about books, students, etc., but the way in which they are storing information is quite different. For example, to store information about Author Name; Library-A uses “*168AuthorName*” as a column name, while Library-B uses the “*Author*” column to store the same information. Replication of data is another major challenge faced in the Library-B database.

Domain knowledge and mapping rules discussed in section III are used to construct the ontology models. The SPARQL queries are applied to the ontology models to check the results of explicit information is according to the datasets.

The SPARQL queries are shown in Figure 8 and Figure 9 applied to the ontology models. For example, to retrieve

information about books whose author name is “Alan Simpson”, as shown in Figure 8. Similarly, to retrieve the title of Book whose distributor is “Pak_Book_Corporation” the SPARQL query, as shown in Figure 9.

The advantage of our methodology is to easily integrate any other database of the library, except to extend (i.e. import) the Root Ontology. The integrated ontology models are referred to as phase 2 ontology models. Phase 2 ontology models are validated using SPARQL queries. Table 5 shows the other six SPARQL queries used to retrieve similar information from the ontology models. There are furthermore SPARQL queries that are used to extract information. Table 5 provides complete information about our work to answer the Syntactic heterogeneity.

To design a database rarely it happens that the rules are not followed, and the database is designed according to requirements, therefore a database contains ambiguous information. Therefore, automatically mapping the database to ontology is not possible. Different approaches are proposed by authors to map a database into ontology, as shown in Table 6, but mostly they are semi-automatic. Table 6 also shows a comparison between existing approaches and our approach. The fields used to compare approaches according to Transformation type, SQL Data type transformation, RDB to RDF Mapping, Data Source Type and Ontology Language used.

To verify the results an application is developed using Java Language. The SPARQL queries shown in Table 5, are used in the Java application, the SPARQL queries are selected according to the user requirements as shown in Figures 10 to 12. Figures 10 to 12 are based on the second SPARQL query shown in Table 5, in these figures the source (i.e. libraries) are selected from a dropdown menu (i.e. Select Sources in Figure 10 to 12) to check the outputs.

The application facilitates the user in three different ways i.e. to retrieve data from a Library-A, to retrieve the data from a Library-B and if a user wants to search anything from both libraries simultaneously. As shown in Figure 10, a user searches a book in the domain of “Earth Sciences” only from Library-A or shown in Figure 11, searches in Library-B. Figure 12 shows that a user searches the same query in both libraries simultaneously. The advantage of the research work is scalability i.e. a user can search in both libraries the same query without re-constructing an integrated database or data translation techniques. The research work classifies the similar field information from heterogeneous databases of the same domain in a concept (which a domain knowledge according to ontology construction) and similar classified fields of the databases are used in SPARQL queries to search the data accordingly. A similar approach can be used for integrating similar other databases with heterogeneous structured databases. Therefore, the Root Ontology is scalable and flexibly used for any other library database.

V. CONCLUSION

In section I, it is discussed that there are two possibilities to integrate the heterogeneous databases i.e. Data Translation

TABLE 5. Sparql queries applied on ontology models.

Query Description	Library A phase 2 Ontology Model	Library B phase 2 Ontology Model
Display the name of books whose price is less than 1000 rupees.	SELECT ?book ?price WHERE { ?book ex:t02Price ?price FILTER(?price<1000) }	SELECT ?book ?price WHERE { ?book ex:PRICE ?price FILTER(?price<1000) }
Display name of books whose title is "Semantic Web"	SELECT ?book WHERE { ?book ex:t02Title "Semantic Web"^^xsd:string }	SELECT ?book WHERE { ?book ex:TITLE "Semantic Web"^^xsd:string }
Display name of book whose title is "Semantic Web" and edition is "Second"	SELECT ?book ?edition WHERE { ?book ex:t02Title "Semantic Web"^^xsd:string. ?book ex:t02Edition ?edition FILTER(?edition="second"^^xsd:string) }	SELECT ?book ?edition WHERE { ?book ex:TITLE "Semantic Web"^^xsd:string. ?book ex:EDITION ?edition FILTER(?edition="second"^^xsd:string) }
Display a list of books which were published in year 2000.	SELECT ?book WHERE { ?book ex:t02DatePublished "2000"^^xsd:decimal. }	SELECT ?book WHERE { ?book ex:YEAR "2000"^^xsd:decimal. }
Display a list of books which were published in "Pakistan"	SELECT ?book WHERE { ?book ex:t02PublishPlace "Pakistan"^^xsd:string. }	SELECT ?book WHERE { ?book ex:PLACE "Pakistan"^^xsd:string. }
Display a list of books whose language is "English"	SELECT ?book WHERE { ?book ex:t02Language "English"^^xsd:string. }	SELECT ?book WHERE { ?book ex:LANGUAGE "English"^^xsd:string. }

TABLE 6. Comparison of proposed approach with existing approaches.

Approach	Transformation Type	SQL Data type transformation	RDB to RDF Mapping	Data Source Type	Ontology Language
[45]	Data Transformation	xsd:datatypes	Complex	Heterogeneous Material Science Databases	OWL
[46]	Query Transformation	xsd:datatypes	Simple	Heterogeneous Material Science Databases	OWL
[36]	Data Translation	Not applicable	Not specified	Cyber Physical Systems	OWL
[27]	Query Translation	Not defined	Not specified	Biomedicine heterogeneous databases	Not specified
[40]	Data Translation	xsd:datatypes	Not specified	Wearable devices, Databases	RDF
Our methodology	Query Translation	xsd:datatypes	Complex	Multiple Heterogeneous Database Systems	OWL

Scheme and Query Translation Scheme, both techniques have their advantages and disadvantages, according to the nature of the databases. For real-time applications, Query Translation Scheme is preferred, whereas for an application where efficiency, Data Translation Scheme is preferred. Databases are mostly designed based on domain knowledge of the designer, and designers rarely follow the rules, therefore the database schema may contain ambiguous information. Therefore, the ontology models are semi-automatically populated using Java to integrate heterogeneous databases. The ontology model is based on the Query Translation technique

because ontology model processing is less time consuming, as compared to the total application base Query Translation techniques. Other than the efficiency of processing by using an ontology, the Root Ontology is extendable according to requirements to integrate other Library databases. The presented methodology needs minor changes in the Root Ontology if a database is updated with new fields. This semi-automatic system requires to include a new class at ontology after importing the Root Ontology; if and only if the required class is not present in the Root ontology, otherwise the same class from the Root Ontology will be used, which was not

possible according to [28], [29]. Our methodology also efficiently maps the primary and foreign keys in ontology modeling, which was not possible according to [21]. Therefore, our methodology works efficiently to integrate the ontology meaningfully (i.e. Semantically), the records are extracted using queries accordingly and correct (i.e. Structurally), and the table field names are identified uniquely for successful retrieval of information (i.e. Syntactically).

The SPARQL, based on the SPARQL Rule Language, presents the results. The SPARQL integrates the databases using Jena API to extract the results from the databases accordingly. The stepwise explanation contributes to constructing the ontology for integrating any similar databases with structural heterogeneity, as discussed in section III B. The verification of ontology models is based on the input data and output results, when the results are not according to the provided datasets then the model is not correct, as discussed in section IV. The rules of SPARQL shown in Table 3 and Table 4 retrieve the required information in both or in any of the databases, as shown in Figures 10 to 12). The rules are implemented based on the requirements of information extraction from the databases.

As discussed in section IV, the Root Ontology can be scalable by updating the ontology according to any other library database and can be integrated with existing Libraries data (i.e. Library-A and Library-B). The procedural method discussed in section III helps to integrate any heterogeneous structured databases with similar data therefore the method can be scalable to various other databases. The novelty of the approach is that the ontology is based on the RDB schema where the Syntactic heterogeneity is solved, which is not used in [35] while mapping the ontology with the database. Therefore, as discussed in section III, the mapping of RDB schema accordingly will help in minimal updating of ontology for any other RDB library databases while integrating. The WordNet can be used to extend the approach by automatic translation of databases into ontology.

The methodology presented in this article is implemented in three different phases. The first phase is mapping database schema into ontology models, as discussed in section III.A. In the second phase, the heterogeneity is removed by creating a common ontology i.e. Root Ontology model, to integrate ontology models created in phase 1, discussed in section III.B. Finally, the records are retrieved from heterogeneous databases using Jena API in Java Eclipse. The ontology integration step is not fully automatic. To fully automate the process, Machine Learning algorithms can be used in combination with Semantic Web Ontology. This will only help to find the entities which refer to the same real-world entities. The integration of Machine Learning algorithms with Semantic Web Ontology will improve the integration process.

REFERENCES

[1] T. Berners-Lee, J. Hendler, and O. Lassila, "The semantic Web," *Sci. Amer.*, vol. 284, no. 5, pp. 34–43, 2001. [Online]. Available: <https://www.jstor.org/stable/e26059181>

- [2] J. G. Breslin, D. O'Sullivan, A. Passant, and L. Vasiliu, "Semantic Web computing in industry," *Comput. Ind.*, vol. 61, no. 8, pp. 729–741, Oct. 2010, doi: [10.1016/j.compind.2010.05.002](https://doi.org/10.1016/j.compind.2010.05.002).
- [3] Z. Wang, S. Xia, and Q. Niu, "A novel ontology analysis tool," *Appl. Math. Inf. Sci.*, vol. 8, no. 1, pp. 255–261, Jan. 2014, doi: [10.12785/amis/080131](https://doi.org/10.12785/amis/080131).
- [4] Vasilecas, Olegas, Diana Bugaite, and Justas Trinkunas, "On approach for enterprise ontology transformation into conceptual model," in *Int. Conf. Comput. Syst. Technologies. CompSysTech*, vol. 6, 2006, pp. 1–8. [Online]. Available: <https://www.cct.lsu.edu/~dayong/research/paper/vasilecas06.pdf>
- [5] M. A. Hazber and R. L. X. G. G. Xu, "Integration mapping rules: Transforming relational database to semantic Web ontology," *Appl. Math.*, vol. 10, no. 3, pp. 1–21, 2016.
- [6] D. Martin, M. Paolucci, S. McIlraith, M. Burstein, D. McDermott, D. McGuinness, and B. Parsia, "Bringing semantics to Web services: The OWL-S approach," in *Proc. Int. Workshop Semantic Web Services Web Process Composition*, Jul. 2004, pp. 26–42, doi: [10.1007/978-3-540-30581-1_4](https://doi.org/10.1007/978-3-540-30581-1_4).
- [7] D. Maier, *The Theory of Relational Databases*, vol. 11. Rockville, MD, USA: Rockville. Comput. Sci. Press, 1983. [Online]. Available: <https://core.ac.uk/download/pdf/81189451.pdf>
- [8] N. F. Noy, M. Sintek, S. Decker, M. Crubezy, R. W. Ferguson, and M. A. Musen, "Creating semantic Web contents with protege-2000," *IEEE Intell. Syst.*, vol. 16, no. 2, pp. 60–71, Mar. 2001, doi: [10.1109/5254.920601](https://doi.org/10.1109/5254.920601).
- [9] Milo, Tova, and Sagit Zohar, "Using schema matching to simplify heterogeneous data translation," *Vldb*, vol. 98, pp. 24–27, Aug. 1998. [Online]. Available: <https://pdfs.semanticscholar.org/159d/f105221fff6dcf3e189c28639aad8302053a.pdf>
- [10] Krishnamurthy, Rajasekar, Raghav Kaushik, and Jeffrey F. Naughton, "XML-to-SQL Query translation literature: The state of the art and open problems," in *Proc. Int. XML Database Symp.*, Sep. 2003, pp. 1–18, doi: [10.1007/978-3-540-39429-7_1](https://doi.org/10.1007/978-3-540-39429-7_1).
- [11] Ghawi, Raji, and Nadine Cullot, "Database-to-ontology mapping generation for semantic interoperability," in *Proc. 3rd Int. Workshop Database Interoperability (InterDB)*, vol. 91, Sep. 2007, pp. 1–8. [Online]. Available: https://www.researchgate.net/publication/228699271_Database-to-ontology_mapping_generation_for_semantic_interoperability.
- [12] Ling, Haiyun, and Shufeng Zhou, "Mapping relational databases into owl ontology," *Int. J. Eng. Technol.*, vol. 5, no. 6, pp. 4735–4740, Dec. 2013. [Online]. Available: <http://pdfs.semanticscholar.org/4852/bc8abcf7f0f98e2dd97d77a5bb1d8b9ec06.pdf>.
- [13] H. Mohammed, N. Clarke, and F. Li. (2016). *An Automated Approach for Digital Forensic Analysis of Heterogeneous Big Data*. [Online]. Available: <https://www.semanticscholar.org/paper/An-Automated-Approach-for-Digital-Forensic-Analysis-Mohammed-Clarke/7fed0ff298a31fb03e9cd0e2b41ea9484e54146a>.
- [14] H. Gao, H. Miao, L. Liu, J. Kai, and K. Zhao, "Automated quantitative verification for service-based system design: A visualization transform tool perspective," *Int. J. Softw. Eng. Knowl. Eng.*, vol. 28, no. 10, pp. 1369–1397, Oct. 2018, doi: [10.1142/S0218194018500390](https://doi.org/10.1142/S0218194018500390).
- [15] H. Gao, K. Zhang, J. Yang, F. Wu, and H. Liu, "Applying improved particle swarm optimization for dynamic service composition focusing on quality of service evaluations under hybrid networks," *Int. J. Distrib. Sensor Netw.*, vol. 14, no. 2, Feb. 2018, Art. no. 155014771876158, doi: [10.1177/1550147718761583](https://doi.org/10.1177/1550147718761583).
- [16] Y. Yin, L. Chen, Y. Xu, and J. Wan, "Location-aware service recommendation with enhanced probabilistic matrix factorization," *IEEE Access*, vol. 6, pp. 62815–62825, 2018, doi: [10.1109/ACCESS.2018.2877137](https://doi.org/10.1109/ACCESS.2018.2877137).
- [17] H. Gao, W. Huang, X. Yang, Y. Duan, and Y. Yin, "Toward service selection for workflow reconfiguration: An interface-based computing solution," *Future Gener. Comput. Syst.*, vol. 87, pp. 298–311, Oct. 2018, doi: [10.1016/j.future.2018.04.064](https://doi.org/10.1016/j.future.2018.04.064).
- [18] H. Gao, Y. Duan, H. Miao, and Y. Yin, "An approach to data consistency checking for the dynamic replacement of service process," *IEEE Access*, vol. 5, pp. 11700–11711, Jun. 2017, doi: [10.1109/ACCESS.2017.2715322](https://doi.org/10.1109/ACCESS.2017.2715322).
- [19] H. Gao, S. Mao, W. Huang, and X. Yang, "Applying probabilistic model checking to financial production risk evaluation and control: A case study of Alibaba's Yu'e bao," *IEEE Trans. Comput. Social Syst.*, vol. 5, no. 3, pp. 785–795, Sep. 2018, doi: [10.1109/TCSS.2018.2865217](https://doi.org/10.1109/TCSS.2018.2865217).
- [20] Y. Yin, F. Yu, Y. Xu, L. Yu, and J. Mu, "Network location-aware service recommendation with random walk in cyber-physical systems," *Sensors*, vol. 17, no. 9, p. 2059, Sep. 2017, doi: [10.3390/s17092059](https://doi.org/10.3390/s17092059).

- [21] H. Gao, D. Chu, Y. Duan, and Y. Yin, "Probabilistic model checking-based service selection method for business process modeling," *Int. J. Softw. Eng. Knowl. Eng.*, vol. 27, no. 06, pp. 897–923, Aug. 2017, doi: [10.1142/S0218194017500334](https://doi.org/10.1142/S0218194017500334).
- [22] D. Aggarwal and K. C. Davis, "Employing graph databases as a standardization model towards addressing heterogeneity," in *Proc. IEEE 17th Int. Conf. Inf. Reuse Integr. (IRI)*, Jul. 2016, pp. 198–207, doi: [10.1109/IRI.2016.33](https://doi.org/10.1109/IRI.2016.33).
- [23] J. Peng, K. Bai, X. Shang, G. Wang, H. Xue, S. Jin, L. Cheng, Y. Wang, and J. Chen, "Predicting disease-related genes using integrated biomedical networks," *BMC Genomics*, vol. 18, no. S1, p. 1043, Jan. 2017, doi: [10.1186/s12864-016-3263-4](https://doi.org/10.1186/s12864-016-3263-4).
- [24] S. Zheng, F. Wang, J. Lu, and J. Saltz, "Enabling ontology based semantic queries in biomedical database systems," in *Proc. 21st ACM Int. Conf. Inf. Knowl. Manage. (CIKM)*, 2012, pp. 67–83, doi: [10.1145/2396761.2398715](https://doi.org/10.1145/2396761.2398715).
- [25] D. Moldovan, M. Antal, D. Valea, C. Pop, T. Cioara, I. Anghel, and I. Salomie, "Tools for mapping ontologies to relational databases: A comparative evaluation," in *Proc. IEEE Int. Conf. Intell. Comput. Commun. Process. (ICCP)*, Sep. 2015, pp. 77–83, doi: [10.1109/ICCP.2015.7312609](https://doi.org/10.1109/ICCP.2015.7312609).
- [26] Doan, AnHai, and Alon Y. Halevy, "Semantic integration research in the database community: A brief survey," *AI Mag.*, vol. 26, no. 1, p. 83, Mar. 2005. [Online]. Available: <https://doi.org/10.1609/aimag.v26i1.1801>
- [27] W. Sujansky, "Heterogeneous database integration in biomedicine," *J. Biomed. Informat.*, vol. 34, no. 4, pp. 285–298, Aug. 2001, doi: [10.1006/jbin.2001.1024](https://doi.org/10.1006/jbin.2001.1024).
- [28] Astrova, Irina, "Reverse engineering of relational databases to ontologies," in *Proc. Eur. Semantic Web Symp.*, May 2004, pp. 327–341, doi: [10.1007/978-3-540-25956-5_23](https://doi.org/10.1007/978-3-540-25956-5_23).
- [29] A. Buccella, M. R. Penabad, F. J. Rodriguez, A. Farina, and A. Cechich, "From relational databases to OWL ontologies," in *Proc. 6th Nat. Russian Res. Conf.*, Sep. 2004, pp. 1–8.
- [30] M. Li, X.-Y. Du, and S. Wang, "Learning ontology from relational database," in *Proc. Int. Conf. Mach. Learn. Cybern.*, 2005, pp. 3410–3415, doi: [10.1109/ICMLC.2005.1527531](https://doi.org/10.1109/ICMLC.2005.1527531).
- [31] G. Shen, Z. Huang, X. Zhu, and X. Zhao, "Research on the Rules of Mapping from Relational Model to OWL," in *Proc. OWLED*, Nov. 2006, pp. 1–8. [Online]. Available: <https://pdfs.semanticscholar.org/a9c1/c5dcd05f18f24d7965a3e1160211cf3b3f48.pdf>
- [32] L. Stojanovic, N. Stojanovic, and R. Volz, "Migrating data-intensive Web sites into the semantic Web," in *Proc. ACM Symp. Appl. Comput. (SAC)*, 2002, pp. 1100–1107, doi: [10.1145/508791.509008](https://doi.org/10.1145/508791.509008).
- [33] Pan, Zhengxiang, and Jeff Hefflin, "Dldb: Extending relational databases to support semantic Web queries," in *Proc. Comput. Sci. Electr. Eng.*, 2004, p. 8. [Online]. Available: <https://apps.dtic.mil/dtic/tr/fulltext/u2/a451847.pdf>
- [34] S. Zhou, G. Meng, H. Ling, and H. Zhang, "Tool for translating relational databases schema into ontology for semantic Web," in *Proc. 2nd Int. Workshop Edu. Technol. Comput. Sci.*, 2010, pp. 198–201, doi: [10.1109/ETCS.2010.208](https://doi.org/10.1109/ETCS.2010.208).
- [35] Zhang, Lei, and Jing Li, "Automatic generation of ontology based on database," *J. Comput. Inf. Syst.*, vol. 7, no. 4, pp. 1148–1154, Apr. 2011.
- [36] V. Jirkovsky, M. Obitko, and V. Marik, "Understanding data heterogeneity in the context of cyber-physical systems integration," *IEEE Trans. Ind. Informat.*, vol. 13, no. 2, pp. 660–667, Apr. 2017, doi: [10.1109/TII.2016.2596101](https://doi.org/10.1109/TII.2016.2596101).
- [37] D. Collarana, C. Lange, and S. Auer, "FuhSen: A platform for federated, RDF-based hybrid search," in *Proc. 25th Int. Conf. Companion World Wide Web*, Apr. 2016, pp. 171–174, doi: [10.1145/2872518.2890535](https://doi.org/10.1145/2872518.2890535).
- [38] V. Ha-Thuc, G. Venkataraman, M. Rodriguez, S. Sinha, S. Sundaram, and L. Guo, "Personalized expertise search at linkedin," in *Proc. IEEE. In Big Data (Big Data)*, Oct. 2015, pp. 1238–1247, doi: [10.1109/Big-Data.2015.7363878](https://doi.org/10.1109/Big-Data.2015.7363878).
- [39] D. Deng, R. C. Fernandez, Z. Abedjan, S. Wang, M. Stonebraker, A. K. Elmagarmid, I. F. Ilyas, S. Madden, M. Ouzzani, and N. Tang, "The data civilizer system," in *Proc. CIDR*, 2017, pp. 1–12. [Online]. Available: <http://cidrdb.org/cidr2017/papers/p44-deng-cidr17.pdf>
- [40] E. Mezghani, E. Exposito, K. Drira, M. Da Silveira, and C. Pruski, "A semantic big data platform for integrating heterogeneous wearable data in healthcare," *J. Med. Syst.*, vol. 39, no. 12, p. 185, Dec. 2015, doi: [10.1007/s10916-015-0344-x](https://doi.org/10.1007/s10916-015-0344-x).
- [41] N. Konstantinou, D. Kouis, and N. Mitrou, "Incremental export of relational database contents into RDF graphs," in *Proc. 4th Int. Conf. Web Intell., Mining Semantics (WIMS)*, 2014, p. 33, doi: [10.1145/2611040.2611082](https://doi.org/10.1145/2611040.2611082).
- [42] Astrova, Irina, "Rules for mapping SQL relational databases to OWL ontologies," in *Proc. Metadata Semantics*, 2009, pp. 415–424, doi: [10.1007/978-0-387-77745-0_40](https://doi.org/10.1007/978-0-387-77745-0_40).
- [43] B. Motik and U. Sattler, "A comparison of reasoning techniques for Querying large description logic ABoxes," in *Proc. Int. Conf. Logic Program. Artif. Intell. Reasoning*, 2006, pp. 227–241, doi: [10.1007/11916277_16](https://doi.org/10.1007/11916277_16).
- [44] C. Bizer and A. Seaborne, "D2RQ-treating non-RDF databases as virtual RDF graphs," in *Proc. 3rd Int. Semantic Web Conf. (ISWC)*, Nov. 2004, pp. 11–33.
- [45] S. Zhao and Q. Qian, "Ontology based heterogeneous materials database integration and semantic Query," *AIP Adv.*, vol. 7, no. 10, Oct. 2017, Art. no. 105325, doi: [10.1063/1.4999209](https://doi.org/10.1063/1.4999209).
- [46] K. Cheung, J. Hunter, and J. Drennan, "MatSeek: An ontology-based federated search interface for materials scientists," *IEEE Intell. Syst.*, vol. 24, no. 1, pp. 47–56, Jan. 2009, doi: [10.1109/MIS.2009.13](https://doi.org/10.1109/MIS.2009.13).
- [47] N. Islam, M. S. Siddiqui, and Z. A. Shaikh, "TODE: A dot net-based tool for ontology development and editing," in *Proc. 2nd Int. Conf. Comput. Eng. Technol.*, vol. 6, Apr. 2010, p. 229, doi: [10.1109/ICCET.2010.5486292](https://doi.org/10.1109/ICCET.2010.5486292).
- [48] D. Oguz, B. Ergenc, S. Yin, O. Dikenelli, and A. Hameurlain, "Federated Query processing on linked data: A qualitative survey and open challenges," *Knowl. Eng. Rev.*, vol. 30, no. 5, pp. 545–563, Nov. 2015, doi: [10.1017/S0269888915000107](https://doi.org/10.1017/S0269888915000107).
- [49] X. Zeng, G. Xu, X. Zheng, Y. Xiang, and W. Zhou, "E-AUA: An efficient anonymous user authentication protocol for mobile IoT," *IEEE Internet Things J.*, vol. 6, no. 2, pp. 1506–1519, Apr. 2019, doi: [10.1109/JIOT.2018.2847447](https://doi.org/10.1109/JIOT.2018.2847447).
- [50] G. Xu, Y. Zhang, A. K. Sangaiah, X. Li, A. Castiglione, and X. Zheng, "CSP-e2: An abuse-free contract signing protocol with low-storage TTP for energy-efficient electronic transaction ecosystems," *Inf. Sci.*, vol. 476, pp. 505–515, Feb. 2019, doi: [10.1016/j.ins.2018.05.022](https://doi.org/10.1016/j.ins.2018.05.022).
- [51] G. Xu, J. Liu, Y. Lu, X. Zeng, Y. Zhang, and X. Li, "A novel efficient MAKA protocol with desynchronization for anonymous roaming service in global mobility networks," *J. Netw. Comput. Appl.*, vol. 107, pp. 83–92, Apr. 2018.
- [52] H. Gao, W. Huang, Y. Duan, X. Yang, and Q. Zou, "Research on cost-driven services composition in an uncertain environment," *J. Internet Technol.*, vol. 20, no. 3, pp. 755–769, 2019.
- [53] H. Zhang, D. Xingchun, Z. Yuan, J. Chun, and Y. Huang, "A new approach to building ontologies from heterogeneous databases," *WIT Trans. Inf. Commun. Technol.*, vol. 46, pp. 1037–1044, Jan. 2014, doi: [10.2495/ISME20131332](https://doi.org/10.2495/ISME20131332).
- [54] J. Wang, Z. Miao, Y. Zhang, and B. Zhou, "Querying heterogeneous relational database using SPARQL," in *Proc. 8th IEEE/ACIS Int. Conf. Comput. Inf. Sci.*, 2009, pp. 475–480, doi: [10.1109/ICIS.2009.11](https://doi.org/10.1109/ICIS.2009.11).
- [55] A. Buccella and A. Cechich, "An Ontology Approach to Data Integration," *J. Comput. Sci. Technol.*, vol. 3, no. 2, pp. 62–67, Oct. 2003.
- [56] H. Yan-Hao, L. Wen-Chen, L. Bai-Qing, L. Ya-Lou, Z. Xiao-Xin, and A. Ning, "The construction of power system knowledge database based on ontology theory and semantic Web technology," in *Proc. Int. Conf. Power Syst. Technol.*, Oct. 2014, pp. 1760–1764, doi: [10.1109/POWERCON.2014.6993596](https://doi.org/10.1109/POWERCON.2014.6993596).
- [57] A. F. Nejad, S. Kharazmi, S. Bayati, S. K. Golmohammadi, and H. Abolhassani, "Semantic log based replication model for optimizing heterogeneous DBMS interaction," in *Proc. 1st Int. Conference Adv. Databases, Knowl., Data Appl.*, 2009, pp. 138–142, doi: [10.1109/DBKDA.2009.24](https://doi.org/10.1109/DBKDA.2009.24).
- [58] J.-Y. Tao, Q.-F. Juan, and W.-H. Juan, "Ontology-based research on heterogeneous database semantic integration strategies," in *Proc. 2nd Int. Workshop Edu. Technol. Comput. Sci.*, 2010, pp. 477–480, doi: [10.1109/ETCS.2010.535](https://doi.org/10.1109/ETCS.2010.535).
- [59] C.-H. Liao, G.-Y. Xiong, and C.-L. Chen, "Research on OWL ontology learning method based on relational schema," in *Proc. 2nd Int. Conf. Modeling, Simulation, Optim. Technol. Appl. (MSOT)*, Nov. 2018, pp. 159–164, doi: [10.12783/dtce/msota2018/27522](https://doi.org/10.12783/dtce/msota2018/27522).
- [60] C.-H. Liao, Y.-F. Wu, and G.-H. King, "Research on learning OWL ontology from relational database," *J. Phys., Conf. Ser.*, vol. 1176, Mar. 2019, Art. no. 022031, doi: [10.1088/1742-6596/1176/2/022031](https://doi.org/10.1088/1742-6596/1176/2/022031).
- [61] M. A. G. Hazber, R. Li, B. Li, Y. Zhao, and K. M. A. Alalayah, "A survey: Transformation for integrating relational database with semantic Web," in *Proc. 3rd Int. Conf. Manage. Eng., Softw. Eng. Service Sci. (ICMSS)*, Wuhan, China, 2019, pp. 66–73, doi: [10.1145/3312662.3312692](https://doi.org/10.1145/3312662.3312692).



MUHAMMAD ASFAND-E-YAR received the bachelor's and master's degrees from International Islamic University, Islamabad, Pakistan, in 2001 and 2004, respectively, and the Ph.D. degree from the Vienna University of Technology, Vienna, Austria, in 2011.

He worked in NUCES-FAST for two years as a Lecturer, from 2004 to 2006 at Peshawar, Pakistan. After completing Ph.D., he worked as a Researcher in a project of Secure Business Austria in 2012 for two years. He worked at Masaryk University, Brno, Czech Republic, as a Researcher for 1.5 years, in 2014. In 2015, he joined Bahria University, Islamabad, working as an Assistant Professor with the Department of Computer Science, working till now. He is currently working in different research projects related to Semantic classification and language processing. His interest areas of research are semantic classification, ontology engineering, natural language processing, deep learning, and smart building architectures.

Dr. Asfand-e-yar win a HEC, Pakistan scholarship to study Ph.D. abroad. In 2013 he wins, ERCIM scholarship for post-doc and worked at Masaryk University. At Masaryk University, he designed ontology model to integrate the heterogonous databases of building technology and structured information of physical building.



RAMIS ALI received the B.S. and M.S. degrees in computer science from Bahria University, Islamabad, Pakistan, in 2015 and 2018, respectively. He is currently working as a Junior Faculty Member with Bahria University. His research interests include linking heterogeneous data using semantic web and information retrieval techniques, image processing and machine learning techniques. His awards and honors include Gold medal in M.S. degree and Silver Medal in B.S. degree from Bahria University.

...