# A REVIEW ON RDB TO RDF MAPPING FOR SEMANTIC WEB

V.Sitharamulu[1] and Dr. B. Raveendra Babu[2]

[1]Associate Professor, Department of Computer Science & Engg, SBIT Khammam
[2]Professor and HOD Department of Computer Science & Engg, Dean-Administration & Finance at VNR VJIET, Hyderabad

## *ABSTRACT*

*In Databases one of the active research fields is mapping relational databases (RDB) into Resource Description Framework (RDF). An enormous data is kept in the form of relational databases and accessing of data is done in the semantic web. The data stored in RDB is to be efficiently mapped to the semantic web or RDF for data availability to the users. There is a definite need for improvement in technologies for efficient mapping languages from RDB to RDF in semantic web. This paper presents an up-to-date survey of different RDB to RDF mapping languages proposed in recent times. It outlines the main features or characteristics to be considered for efficient mapping in different scenarios. The main objective of this content, pictures identification of limitations existing in the mapping languages. It also enhances the comparisons between each language and helps researchers to propose further better proposals in their future scope of work to improve better mapping techniques.*

## *KEYWORDS*

*Data bases, relational databases (RDB), Semantic Web, RDB to RDF Mapping, Mapping Characteristic Comparison.*

## 1. INTRODUCTION

In Relational Databases, one of the active research field is mapping of databases. In [1], authors have stressed on the need to fill the gap between RDF and relational model for accessing of data on semantic web. The real time applicability of semantic web is extended to areas far than web. The applications of semantic web in data exchange and integration from different data sources is elaborated in [2-6].

Different researchers provided improved platforms for allowing semantic web based applications for accessing data from relational databases. The proposed approaches have implemented a wide range of techniques form simple to highly specific and logical ways to deal with mapping language. This ongoing research had standardized protocols for efficient implementations of W3C for RDB2RDF Working Group (WG). The main highlight [7] of RDB2RDF WG for standardization is to have efficient mapping of relational database schemas and relational data into RDF and OWL.

The organization of this paper is given as: in Section 2, the recent advances of mapping languages are presented. In Section 3, study and comparison of mapping languages is presented and In Section 4comparison framework is shown with different approaches. Finally, in Section 5, we make our concluding and future remarks.

DOI:10.5121/ijfcst.2016.6203

## 2. Recent Advances in RDB-RDF Mapping Languages

The idea behind this survey is to outline the main mapping languages for RDB to RDF. The literature presented should be helpful to the research community for further exploration with different application scenario for RDB to RDF language mapping. This survey reports different comparison scenarios for RDB to RDF language mapping framework. The survey also classifies the mapping languages into different groups such as: direct, read-only general purpose, read-write general purpose and special purpose mapping.

In [8], W3C RDB2RDF Incubator Group has provided a detail survey on mapping RDB to RDF for different techniques, software tools and applications. In [9], the author has defined a mechanism that bridges between relational databases and OWL technologies using SQL.

In the section below, different techniques for mapping languages are descried with comparison on a feature by feature basis.

### 2.1 Mapping Languages

The mapping languages can be divided as given below:

- Direct mapping
- General-purpose Read-only mapping
- General purpose Read-write mapping and
- Special - purpose mapping.

**Direct Mapping:**

Direct mapping [10] is a simple technique which is used to map data directly from RDB-to-RDF. In direct approach, link table is one of the way of relational model which uses a direct means without help of constructor. The virtue of this approach is its simplicity in implementation. The applicable areas for direct mapping are simplicity which has more importance than complexity in relational scheme representation.

**Read-only General-purpose Mapping**:

The requirement of only reading data, can be provided using read-only general-purpose mapping languages such as Virtuoso, D2RQ, and R2RML. Virtuoso provides feasibility in applicability of named graphs but not feasible for static metadata implementation. D2RQ provides feasibility in applicability of static metadata but not feasible for named graphs implementation. R2RML supports the implementation of both named graphs and static metadata.

This mapping language provides a great architecture for better representation of data from RDB to RDF. As the complexity of representation is increased it makes impossible for bidirectional data transfer. A great level of practice and expertise is needed for learning and understanding of read-only general-purpose mapping language. These groups of mapping languages are preferred for a vast category of applications where the access is limited to read only and support of specific standards, compliances and implementation.

**Read-Write General-purpose Mapping**:

Read-write general purpose mapping can provide read and write data access from RDB to RDF. This approach of mapping provides bidirectional data transfer from RDB to RDF. The differences of read-only general purpose mapping language from read-write general-purpose mapping is mapping of logical tables and are not allowed for view updating problems [11]. The supporting features of read-write general-purpose mapping over read-only general purpose mapping language is the integrity constraints such as not null attributes, detecting missing data and invalid write request. In [12-14] authors have discussed an open problem of view updates in database research.

**Special-purpose Mapping:**

Special-purpose mapping languages are developed for specific features support. The mapping languages such as eD2R, R2O, and Triplify [15] are the mapping languages which are of special purpose. The difference of special-purpose mapping languages with general purpose languages are in few features only, in most of the ways they are similar to the general purpose mapping languages. The supportability of features such as named graphs, blank nodes and metadata is not supported completely. In rare cases, the implementation of blank nodes is available in eD2R and R2O. In cases where general purpose mapping languages are not applicable, these special purpose mapping languages are of great use.

## 3. STUDY AND COMPARISON OF MAPPING LANGUAGES

The comparison of different mapping languages are presented in the sub section below, which helps in choosing a specific RDBs mapping technique for general applicability:

### 3.1 Direct Mapping

In direct mapping, RDBs data is presented in a simple form for semantic web representation [10]. One of the core objectives of the direct mapping allows us to elaborate the RDB on semantic web relational table to classes and attributes to properties in RDF vocabulary. In [11], author has proposed a direct automatic mapping language. An improved heuristic is also extended for simple table-to-class and attribute-to-property mapping using RDB schema. In [10], author has proposed a Squirrel RDF [12] for direct mapping. The RDB schema names are used for generating the RDF vocabulary and RDF based tool performs the mapping of domain ontology. The mapping language discussed above uses the basic technology of direct mapping for RDB-to-RDF.

**eD2R**: eD2R[12], permits the tables in the RDBs to publish in semantic web, which allows users to access the relation data via de-reference able URIs using SPARQL queries. Another objective of eD2Rs is to present RDB data in the form of interlinked datasets shown as RDF.

**R2O**: In [13], authors have proposed R2O which is used for efficient mapping of RDB schemes with ontologism. It is also used for representation of different complex mappings in an efficient way using generic structure.

**Relational.OWL**: Relational.OWL [15] was proposed for schema components and relational data using OWL-based representation format. Relatioal.OWL uses ontology to implement schema of data in RDF and it also uses tables to map complete details regarding the primary/foreign keys and data types in the expression of ontology. The exchange of data in one to one databases is one of the best applicable fields for the Relatioal.OWL. The reuse of the existing domain vocabulary is restricted and the structures, syntax of different relational schema of the RDF representations are shown. The on demand transactions of SPARQL queries for SQL are provided by Relational.OWL using SPARQL interface.

**Virtuoso RDF Views**: The Virtuoso Universal server represents relational data on the semantic web using open link software features for RDF views [16-17]. By applying a declarative meta schema language, the mapping of SQL data to RDF vocabularies can be done.

**R2RML**: The ongoing mapping language for standardization of theW3C RDB2RDFWG for RDB-to-RDF mappings is R2RML [16]. The key objective is to allow read-only data access using a language that is vendor independent.

**R3M:** In ONTOACCESS [17], authors have implemented R3M [18] as the platform for extending mapping language. R3M is a mapping language with the feature of bidirectional data transfer from RDB to RDF. R3M can be applied for mapping tables to classes and attributes to properties with a unique data for integrity constraints.

### 3.2 Bi-Directional Relational-to-RDF Translation

The recent proposals shown in the above section are all capable of transferring data in only one direction with the features of read-only view of relational databases. Here are some of the bidirectional proposals for transfer of data from RDB to RDF and vice versa. In [17], authors have proposed a mapping language known as R3M which is an extension of D2RQ with the support of SPARQL/Update language for data manipulation on ONTOACCESS. D2RQ++ [18], also allows the bidirectional transfer of data from RDB to RDF using D2RQ's mapping language.D2RQ++ also eliminated the problem with mismatch of data and it works efficiently for open world assumptions of semantic web.

### 3.3 Other Research Projects

The need for efficient platform for RDB to RDF is projected by initiating the RDB2RDF Incubator Group (XG) [19] for efficient mapping of data from RDB to RDF on semantic web. In 2009, R2RML [20] has given a good study on the RDB2RDF WG3 which was taken for further extension by many researchers. The need for updating relational data is still in investigation. In [21], the implementation of write operation for support of R2RML is shown as in feasible. Other pushback services for web applications are used in Web 2.0 APIs.

## 4. COMPARISON FRAMEWORK

This section shows the framework of different approaches with respect to efficient implementation of different features. In total 15 different characteristics as shown below are studied on RDB to RDF mapping languages:

Data types (C1): The type of data used for mapping from SQL to XML used in RDF.

Logical Table to Class (C2): Mapping of logical table to a class in RDF.

Integrity Constraints (C3): Integrity constraints differentiate between a primary and foreign key with not null, unique and check constraints.

Transformation Functions (C4): transformation functions are used for conversion of values from RDB to RDF.

User-defined Instance URIs (C5): The User-defined instances are generated in the mapping or conversion of the data from RDB to RDF.

Literal to URI (C6): The URI generated to be stored in specific literals known as URI literal.

Vocabulary Reuse (C7): Existing RDF vocabulary used.

Named Graphs (C8): The mapping of the specific parts of RDB to RDF can be done using Named Graphs.

Blank Nodes (C9): The nodes are used to store unprocessed information.

Static Metadata (C10): The meta-data is used to describe RDF components which don't have the equivalent in RDB.

One Table to n Classes (C11): The table in RDF can be mapped numerous times with subset of attributes and records to numerous classes.

Project Attributes (C12): The attribute used to map different irrelevant and sensitive attributes such as passwords.

Select Conditions (C13): The select conditions use certain constraints to validate the outdated data to exclude from the RDF.

M:N Relationships (C14): M:N relationships are used for linking or join tables for mapping to RDF.

Write Support (C15): The facility for write support should exist for an efficient RDB to RDF mapping.

The set of 10 mapping languages are compared on the above given fifteen characteristics:
Direct Mapping: In Direct Mapping out of 15 only one characteristic i.e., Write Support (C15) is efficient and Logical Table to Class (C2) is partially supported. The remaining 13 characteristics are not supported by the direct mapping technique.

**eD2R**: In eD2R mapping out of 15 only 11 characteristics namely Data types (C1), Logical Table to Class (C2), Transformation Functions (C4), User-defined Instance URIs (C5), Literal to URI (C6), Vocabulary Reuse (C7), Blank Nodes (C9), Project Attributes(C12), Select Conditions(C13), M:N Relationships(C14), Write Support (C15) are efficient and integrity constraints(C3) is partially supported and 3 characteristics i.e. Named Graphs (C8), Static Metadata (C10) and Write Support (C15) are not supported by the eD2R mapping technique.

**R2O**: In R2O mapping out of 15 only 9 characteristic Data types (C1), Logical Table to Class (C2), Transformation Functions (C4), User-defined Instance URIs (C5), Literal to URI (C6), Vocabulary Reuse (C7), Named Graphs (C8), Blank Nodes (C9), One Table to n Classes (C11), Project Attributes(C12) and Select Conditions(C13)i.e are efficient and 2 characteristic Integrity Constraints (C3) and One Table to n Classes (C11) are partially supported and 3characteristic Static Metadata (C10),, M:N Relationships(C14) and Write Support (C15) are not supported by the R2O mapping technique.

**Relational.OWL:** In Relational.OWL mapping out of 15 only 4 characteristic i.e. Data types (C1), Blank Nodes (C9), Project Attributes(C12) and Write Support (C15) are efficient and 2 characteristic Logical Table to Class (C2) and Integrity Constraints (C3) are partially supported and 9characteristic Transformation Functions (C4), User-defined Instance URIs (C5), Literal to URI (C6), Vocabulary Reuse (C7), Named Graphs (C8),Static Metadata (C10), One Table to n Classes (C11), Select Conditions(C13) and M:N Relationships(C14) are not supported by the Relational.OWL mapping technique.

**Virtuoso RDF Views:** In Virtuoso RDF Views mapping out of 15 only 12 characteristic i.e. Data types (C1), Logical Table to Class (C2), Transformation Functions (C4), User-defined Instance URIs (C5), Literal to URI(C6), Vocabulary Reuse (C7), Named Graphs (C8), Blank Nodes (C9), One Table to n Classes (C11), Project Attributes(C12), Select Conditions(C13) and M:N Relationships(C14) are efficient and 1 characteristic Integrity Constraints (C3)is partially supported and 2characteristic Static Metadata (C10) and Write Support (C15) are not supported by the Virtuoso RDF Views mapping technique.

**D2RQ:** In D2RQ mapping out of 15 only 12 characteristic i.e., Data types (C1), Logical Table to Class (C2), Transformation Functions (C4), User-defined Instance URIs (C5), Literal to URI (C6), Vocabulary Reuse (C7), Blank Nodes (C9), Static Metadata (C10), One Table to n Classes (C11), Project Attributes(C12), Select Conditions(C13) and M:NRelationships(C14) are efficient and 1 characteristic Integrity Constraints (C3)is partially supported and 2characteristic Named Graphs (C8) and Write Support (C15) are not supported by the D2RQ mapping technique.

**Triplify:** In Triplify mapping out of 15 only 10 characteristic i.e. Data types (C1), Logical Table to Class (C2), Transformation Functions (C4), User-defined Instance URIs (C5), Literal to URI (C6), Vocabulary Reuse (C7), One Table to n Classes (C11), Project Attributes(C12), Select Conditions(C13), M:N Relationships(C14) are efficient and 1 characteristic , Integrity Constraints (C3)is partially supported and 4characteristic Named Graphs (C8), Blank Nodes (C9), Static Metadata (C10) and Write Support (C15)are not supported by the Triplify mapping technique.

**R2RML:** In R2RML mapping out of 15 only 13 characteristic i.e. Data types (C1), Logical Table to Class (C2), Transformation Functions (C4), User-defined Instance URIs (C5), Literal to URI (C6), Vocabulary Reuse (C7), Named Graphs (C8), Blank Nodes (C9), Static Metadata (C10), One Table to n Classes (C11), Project Attributes(C12), Select Conditions(C13), M:N Relationships(C14) are efficient and 1 characteristic Integrity Constraints (C3)is partially supported and 1characteristic Write Support (C15)is not supported by the R2RML mapping technique.

**R2D:** In R2D mapping out of 15 only 12 characteristic i.e. Data types (C1), Logical Table to Class (C2), Integrity Constraints (C3), Transformation Functions (C4), User-defined Instance URIs (C5), Literal to URI (C6), Vocabulary Reuse (C7), Blank Nodes (C9), One Table to n Classes (C11), Project Attributes(C12), Select Conditions(C13), M:N Relationships(C14) are efficient and 1 characteristic Write Support (C15)is partially supported and 2characteristic Named Graphs (C8) and Static Metadata (C10) are not supported by the R2D mapping technique.

**ONTOACCESS (R2M3):** In R2M3 mapping out of 15 only 9 characteristic i.e. Data types (C1), Integrity Constraints (C3), Transformation Functions (C4), User-defined Instance URIs (C5), Literal to URI (C6), Vocabulary Reuse (C7), One Table to n Classes (C11), Project Attributes(C12) and M:N Relationships(C14) are efficient and 3 characteristic Logical Table to Class (C2), Select Conditions(C13) and Write Support (C15) are partially supported and 3characteristic Named Graphs (C8), Blank Nodes (C9) and Static Metadata (C10) are not supported by the R2M3 mapping technique.

## 5. CONCLUSIONS

This paper presents a detailed review, comparisons and characteristics of various mapping languages for RDB to RDF. The categorization of different mapping languages is shown with the support of recent proposals in that domain. The advantages and limitations of the mapping languages used for RDB to RDF are briefly outlined for the use of research community in this field. The complete set of features or characteristics used for describing and proposing new mapping languages are presented. The set of limitations in existing mapping languages for RDB to RDF in semantic web can lead for efficient proposals in future work for researchers.

## REFERENCES

[1]. K. C.-C. Chang, B. He, C. Li, M. Patel, and Z. Zhang. (2004), Structured Databases on the Web: Observations and Implications. SIGMOD Record.

[2]. A. Langegger, W. Wöss, and M. Blöchl. (2008), A Semantic Web Middleware for Virtual Data Integration on the Web. In Proceedings of the 5th European Semantic Web Conference.

[3]. L. Ma, X. Sun, F. Cao, C. Wang, and X. Wang. (2009), Semantic Enhancement for Enterprise Data Management. In Proceedings of the 8th International Semantic Web Conference.

[4]. F. Manola and E. Miller. (February 2004) RDF Primer. W3C Recommendation. http://www.w3.org/TR/2004/REC-rdf-primer-20040210/.

[5]. C. Ogbuji. (May 2011) SPARQL 1.1 Graph Store HTTP Protocol. http://www.w3.org/TR/2011/ WD-sparql11-http-rdf-update-20110512/.

[6]. C. Patel, S. Khan, and K. Gomadam. (2009). TrialX: Using Semantic Technologies to Match Patients to Relevant Clinical Trials Based on Their Personal Health Records. In Proceedings of the 8th InternationalSemantic Web Conference.

[7]. H. Halpin and I. Herman. RDB2RDF Working Group Charter. http://www.w3.org/2009/08/rdb2rdf-charter. Last visited July 2011.

[8]. S. S. Sahoo, W. Halb, S. Hellmann, K. Idehen, T. T. Jr, S. Auer, J. Sequeda, and A. Ezzat. (2009) A Survey ofCurrent Approaches for Mapping of Relational Databases to RDF.http://www.w3.org/2005/Incubator/rdb2rdf/RDB2RDF_SurveyReport.pdf,.Last visited July 2011.

[9]. GuntarsBumans" (2010. Vol. 756, P:99–117). Mapping between Relational Databases and OWL Ontologies: an Example", Computer Science and Information Technologies, Scientific Papers, University of Latvia ,.

[10]. T. Berners-Lee. (2009) Relational Databases on the Semantic Web.http://www.w3.org/DesignIssues/RDB-RDF.html,. Last visited July 2011.\

[11]. F. Bancilhon and N. Spyratos. (1981) Update Semantics of Relational Views. In ACM Transactions on DatabaseSystems,.

[12]. W. Hu and Y. Qu. (2007), Discovering Simple Mappings Between Relational Database Schemas and Ontologies.In Proceedings of the 6th International and 2nd Asian Semantic Web Conference,.

[13]. J. Barrasa, O. Corcho, and A. G´omez-P´erez.( 2003), Fund Finder: A Case Study of Database-to-OntologyMapping. In Proceedings of the Semantic Integration Workshop,.

[14]. SquirrelRDF. http://jena.sourceforge.net/SquirrelRDF/.Last visited July 2011.

[15]. J. Barrasa, O. Corcho, and A. G´omez-P´erez. (2004). R2O, an Extensible and Semantically Based Database-to-Ontology Mapping Language. In Proceedings of the 2nd Workshop on Semantic Weband Databases.

[16]. C. Bizer. (2003). D2R MAP – A Database to RDF Mapping Language. In Proceedings of the 12th InternationalWorld Wide Web Conference.

[17]. D. Beckett and J. Grant. SWAD-Europe Deliverable 10.2: Mapping Semantic Web Data with RDBMSes.http://www.w3.org/2001/sw/Europe/reports/scalable_rdbms_mapping_report/, January 2003. Last visited July 2011.

[18]. S. Auer, S. Dietzold, J. Lehmann, S. Hellmann, and D. Aumueller (2009). Triplify – Light-Weight Linked Data Publication from Relational Databases. In Proceedingsof the 18th International World Wide WebConference,.

[19]. C. P. de Laborda and S. Conrad (2005). Relational.OWL – A Data and Schema Representation Format Based on OWL. In Proceedings of the 2nd Asia-Pacific Conference on Conceptual Modelling.

[20]. Erling and I. Mikhailov. RDF Support in the Virtuoso DBMS. In Proceedings of the SABREConference on Social Semantic Web, 2007.

[21]. O. Software. Mapping Relational Data to RDF with Virtuoso's RDF Views. http://virtuoso.openlinksw.com/whitepapers/relational%20rdf%20views%20mapping.html. Last visited July 2011

## AUTHORS

1. V. Sitharamulu has obtained his Master of Technology in Computer Science and Engineering from Rajasthan Vidyapeeth University, Udaipur. He is currently working as an Associate Professor in the Department of computer Science & Engineering, Swarna Bharathi Institute of Science & Technology, Khammam, Telangana. He has over 15 years of teaching experience. His area of research interest is Data Mining and Semantic Web.

2. Dr. B. Raveendra Babu has obtained his Masters in Computer Science and Engineering from Anna University, Chennai. He received his Ph.D. in Applied Mathematics at S.V University, Tirupathi. He is now working as Professor and HOD Department of Computer Science and Engineering, Dean-Administration & Finance at VNR VJIET, Hyderabad. He has over 30 years of teaching experience. He has about 60 International and National publications to his credit. His area of research interest includes Data Mining, Software Engineering, Image Processing, Pattern Analysis and Information Security.