International Journal of Database Theory and Application
Vol.8, No.3 (2015), pp.15-26
http://dx.doi.org/10.14257/ijdta.2015.8.3.02

Extracting Entity Relationship Diagram (ERD) From Relational Database Schema

Hala Khaled Al-Masree

Mutah University, Jordan, Alkarak
Hala83990@gmail.com

Abstract

Database Reverse Engineering (DBRE) is an operation used to extract requirements from any system. The operation is implemented to facilitate the understanding of the system that has a little documentation about design and architecture. DBRE is a very important process used when database designers would like to expand the system or transition to the latest technology in DBRE fields. In the relational database model DBRE try to extract Entity Relationship Diagram (ERD) from relational database schema. Database designers find content of the data for a lot of attributes are not related with their names. In this paper, proposed methodology used to extract ERD from relational database schema with the attributes related with their names, both types of entities regular and weak entity, relationships and keys, which are found in the table that has extracted the relational database schema. The basic inputs of this approach are relational database schema that generated from database. The relational database schema used to extract the information about ERD. After that, obtain information that contain keywords help database designers to extract entities and their attributes semantics related with their names from relational database schemas. Then, Determine primary keys, foreign keys and constraints of the database system. In the final step, the ERD is successfully extracted

Keywords: *DBRE, ERD, Relational database schema***.**

1. Introduction

Since 1976 chen [8] founded the ERD during his researches, the ERD was considered a very good way to do a conceptual model for databases. ERD is graphical way to give information about database in the system. Using ERD diagram database designers can convert this information to build a database tables. Information that has been obtained about database helps to reach to relational database schema. When database designers need to turn from relational database schema to ERD they need a methodology leads them to do that. In many cases, adjustments are made to the database such as writing database code without using ERD, because the people who have modified these adjustments do not saw the importance to use ERD. Therefore it is necessary propose the approach resolve this case.

[5] DBRE also includes these objectives: to provide missing or alternative documentation, to assist with maintenance, to migrate from one hardware or software platform to another, etc. Many companies or specialists in database design need retrieval ERD from relational database schema to achieve the DBRE objectives [16,18].

In this paper, propose methodology in order to provide way to those who need DBRE to extract ERD from relational database schema [10]. The proposed methodology facilitated utilization ERD when the database designers need to develop the databases depending on ERD. Database design is built by the analysis of the problems then extract relational database schema. In addition the database in the most companies maybe needs to modify by different database design specialists. This is should lead to think about approach to reverse engineering [9] from relational database schema to ERD.

ISSN: 2005-4270 IJDTA

This paper presents a new methodology based on DBRE approach. The methodology must be able to extract ERD from relational database schema. The ERD is expressed as entities, its attributes and relationship between entities. The proposed methodology based on relational database schema as a basic input. Moreover the relational database schema used as input to extract the helpful information to build ERD. The information that has been obtained from relational database schema leads the database designers to convert the relational database schema to ERD.

The rest of the paper is organized as follows. Section 2 talk about some related paper with this paper. Section 3 gives a brief description of the problem we intend to deal with. Section4 present an overview of an approach using block diagram. Section 5-6 describes the proposed approach. An example illustrates the proposed approach in detail presented at section7. Section9 presents some limitation on proposed approach and concludes this paper.

2. Related Work

Entity Relationship Diagram (ERD) is familiar diagram which aims to present the database structure in Conceptualized form. A lot of papers talk about ERD. One of the most basic things that must have attention English syntax to build good ERD depends on type of word (noun, verb, adjective, and gerund) [8]. It also contributes to build excellent schema.

AER one diagram that developed by P.S. Dhabe et.al [1]. AER diagram depend on the ERD. Methodology used to determine which the best form of notation between ERD and UML depend on uses and purposes of this methodology by HelenC et.al [4]. The ERD is the most diagram used to represent anything can generated from database cause it simply and understandable diagram. A set of form model schemas of an operational relational database used to extract ERD schema by N.Mfourga[5]. This is evidence importance of ERD in researches.

Reverse engineering is the procedure used to extracting knowledge or design information from anything made by specialists in the field of database. Depending on relational database was extracted UML conceptual schema with OCL constraints and derivation introduce by AtlanMod team et.al [6]. An approach based on the system display forms, table schema and data instance as input to extract ERD proposed by Dowming Yeh et.al [7].

Most of papers talk about ideas to develop the methodologies that used in Database. Database designers must use special tools in order to reach to the best implementation of these methodologies. ERDraw[3] tool used for automatically represent ERD and translate it to relational database schema by Shuyun Xu et.al. Marking diagrams made researchers turning to think how marking diagram automatically one of these researches proposed a diagram tool to do semi-automatic assessment that help to marking diagram in 2006 by F.Batmaz[2]. There are many enterprises demand modification on systems from time to time. Enterprises may need modifications in the system, but the database designers who build the UML no longer work in this enterprise. This is why making the specialists in the database field propose approach and tool to deal with this case. One of these specialist Manar H. Alalfi *et al.* [9] proposed approach and tool to extract UML from schema.

Companies need to make adjustments from time to time on their systems. The needs of companies to conduct the modifications give an indication to their need for helpful methodologies. The methodologies help database designers to refer to any part of the system at any time to conduct the required modifications. The need for modifications to the systems has made a lot of people develop methodologies. The methodologies that talk about DBRE help database designers to extraction ERD and do the suitable modifications on the systems. [5] One of the methodologies used to extract ERD from relational database schema, but this methodology based on forms. The forms that used in the previous methodology may be not suitable for any system. This make an ERD

extraction process does not understood to any system. This reason creating an opportunity to make extracting ERD from relational database schema is the subject of this paper, through rules to suit any system.

3. The Problem

The first view to the relational database schema some information becomes clear. But on closer examination some important information will be defined. Therefore, it must follow certain rules that help the extraction of ERD from relational database schema without relying on the forms.

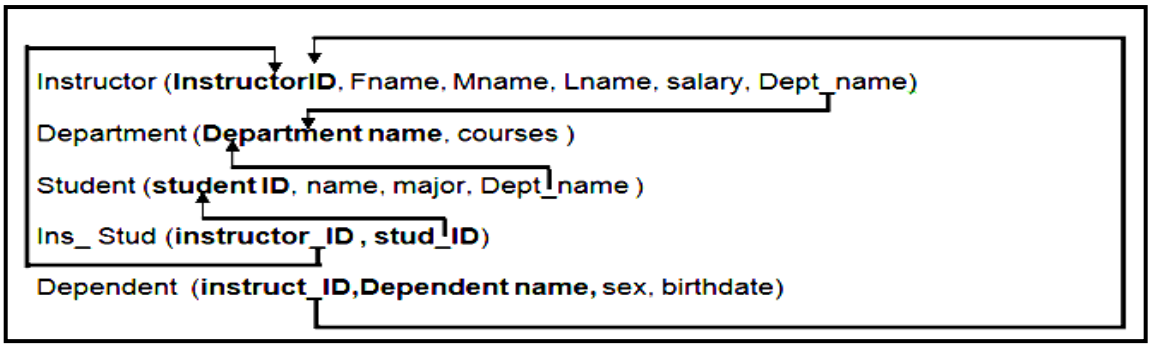

Figure 1. An Example of Relational Database Schema Generated from ERD

An example of relational database schema generated from ERD shown in Figure 1. It is clear that this database contains information about entities and their attributes and relationship between these entities. However, on a deeper study, other structures may be determined. For example, *each instructor works on one department. This relationship is represented by the attribute* ***Dept_name*** *in the relation* ***Instructor****, pointing to the row in the relation* ***Instructor*** *that describes the instructors.* Another example, *each student is taught from at least one instructor and one instructor taught at least one student. This relationship is represented by the relation* ***Ins_ Stud****. In this relation the attribute* ***instructor_ID*** *points to a row in the relation* ***Instructor*** *that describes the instructors. The attribute* ***stud_ID*** *points to a row in the relation* ***Student*** *that describes the students.* When any system requires modifications or developments on ERD then obtain new relational database schema, database designers need a way to extract ERD from schema based on reverse engineering. In addition the database in some enterprises maybe needs to modify by different database designers. This cases leads to talk about main objective of this paper is to present methodology for extract ERD from relational database schema.

4. The Proposed Methodology

The main components of the proposed methodology is shown in (Figure2). The proposed methodology is composed by two steps: Information Extraction and ERD Extraction. Information extraction step focuses on the attribute, key, and constraint determination. Information extraction done by extracting information about ERD from relational database schema that obtained from database. The information about ERD is extracted. ERD extraction focuses on information about ERD which will lead database designers to ERD. The following subsections describe each process in details.

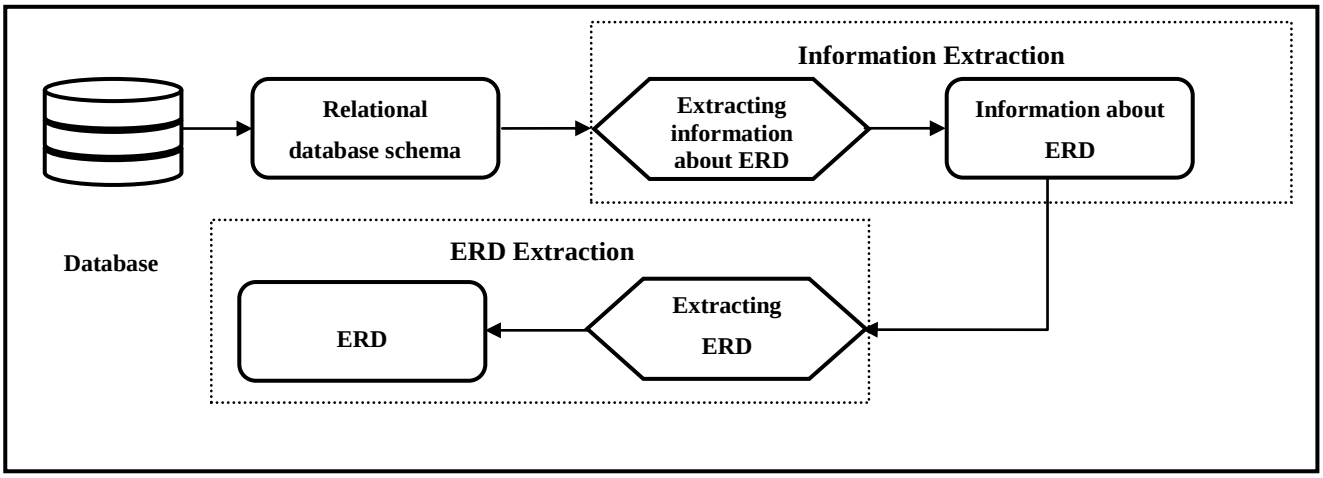

Figure 2. Block Diagram that Summarizes the Major Steps in our Approach

5. Information Extraction

The process of extracting information about ERD uses the relational database schema of the source database as main input. This is done through these steps: entity extraction then derive keys and attributes, relationship extraction, cardinality determination, and schema integration. Each step will be described in detail.

5.1 Primary and Foreign Keys Derivation

The aims of keys and attribute derivation from entity obtain the true meaning of each field in relational database schema. Database designers have to examine all attribute for each relation in the relational database schema for uniqueness, as suggested by (Al hajj [11]).

The good starting point to extract primary key excluded the attribute that brook the presence of null value to reduce the effort of checking uniqueness. Finally it is necessary to follow the same criterion to pick out a primary key. Primary keys used to derive foreign keys in order to define association between relations. During this process, every primary key is examined whether it is referred in fields of other relations (Al hajj [11]). The referring field is a foreign key. The rule that used to determine foreign key presented in Table1.

Table 1. Primary and Foreign Keys Extraction Rules

Primary and foreign key extraction rules
❖ The referring fields must match with referred fields. ❖ The values of the referring field in relation must be a subset of those of the referred field.

5.2 Entity Extraction

By grouping homogeneous pieces of information specialists in database design can derive the entities and determine the entities type. The entities are obtained through gather fields whose linked attributes belong to the same relation. Each relation names maybe an entity or relationship between two entities. Database designers are able to know how distinguish that this relation is an entity or relationship. Database designers can extract entities and the semantic of attributes through the adoption of the following rules shown in Table2.

Table 2. Entities Extraction Rules

Regular Entity Rules
❖ All relations contain linked attributes is mapped regular entity. ❖ Consider all linked attributes in this relation are attributes to this regular entity. ❖ Assume the regular entity come from relation1. ❖ Assume the related relation with relation1 is relation2. ❖ Consider the attribute in relation1 that related with relation2 foreign key in relation2, primary key in relation1. If primary key in relation1 related with relation2 as foreign key that indicate to existence a relationship between these two relations.
Weak Entity Rules
❖ All relation with two attributes at least and one of these attributes is key depends on primary key in another relation is mapped as weak entity. ❖ If existence of relation1 depend on relation2 that mean the key in relation1 is partial key. ❖ Partial key in relation1 with primary key in relation2 will be primary key in relation1. ❖ If partial key from relation1 does not exist in relation2 as part of primary key, therefore the existence of this relation will be cancelled.

Now the database designers can translate this relation into entity and determine the entity type. The linked attributes [12] of the relation become the attributes of entity. Then identify the primary and foreign keys.

5.3 Relationship Extraction

Each relationship is translated into a relationship between entities. The rule that lead to extract relationship between entities is formulated in Table3. Functional dependencies and cardinality identified between attributes will be easy and useful in this case.

Table 3. Relationship Extraction Rule

Relationship Extraction Rule
❖ All relation contains more than one attribute and each attribute related with another relation as part of primary key is mapped as relationship between two entities. ❖ All relation have primary key related with another relation as foreign key, database designers must mapped relationship between those two entities. ❖ All relation is mapped as weak entity has weak relationship between this relation and the relation related with it.

5.4 Cardinality Determination

The cardinality between entities has an important role to facilitate the ERD extracting then, mapped it. When database designers need to determine cardinality from relational database schema they follow the rules shown in Table4.

Table 4. Cardinality Determination Rule

Cardinality Determination Rules
❖ Suppose database designers need to know cardinality between two relations (relation1, relation2). ❖ If relation1 contains foreign key (primary key in relation2) this indicates the cardinality between relation1 and relation2 maybe (1:1) or (1: M). ❖ If cardinality (1:1) this means the relation1 have total participating line and the relation2 have partial participating line. So, the primary key in relation2 is placed in relation1 as foreign key. ❖ If cardinality (1: M) this means the relatin1 has (M) and relation2 have (1). For that, the primary key in relation2 is placed in relation1 as foreign key.

6. Entity Relationship Diagram (ERD) Extraction

The ERD extraction uses the information that extracted in extracting information process in the previous steps to extract ERD and mapped it. This is done through represent the entities, attributes and relationships using the notations that belong to ERD. The information extraction process will be clarified in detail through example in the following section.

7. Example

In this section the block diagram process will illustrate in detail by the follow example. The rules that used to extract information to help extract the ERD in the previous steps will be used to generate the ERD.

7.1 Keys Determination and Entity Extraction

Primary and foreign key determination is the first step that database designers apply it in the information extraction stage. After that, extract an entity. Database designers will follow the rules that shown in Table1, Table2 help to extract primary, foreign key and entities.

At the first, the relations shown in Figure1 will be examined if contains linked attribute. The (**InstructorID**, Fname, Mname, Lname, Salary) are linked attributes. So the **Instructor** is **regular entity**. The content of **Instructor** and **Student** relations it is attributes for it except (Dept_name) it attribute to another relation. The **Dept_name** is foreign key related with primary key in **Department** relation. For the (Fname, Mname, Lname) they participate in one attribute *name*. So, this attribute it is **composite attribute**. The primary key in the **Instructor** entity is **InstructorID**. That is clear through the rules that have been followed in Table 1. Figure3 shown **Instructor** entity that extract from relation database schema.

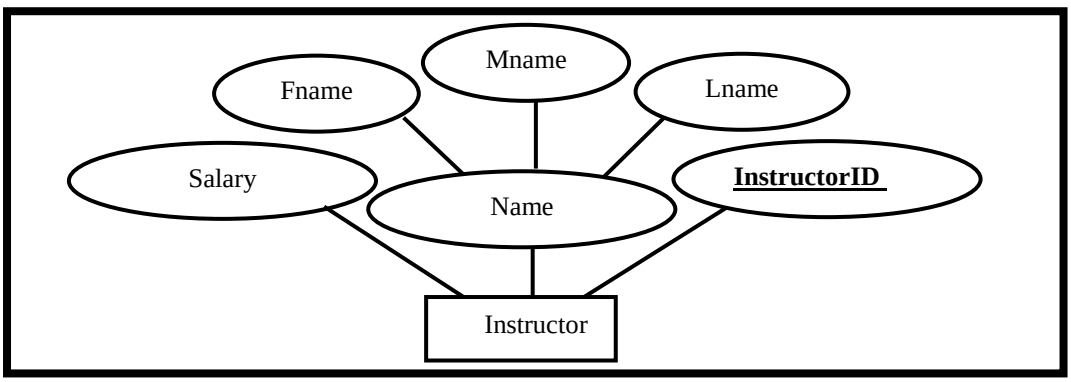

Figure 3. Representation of the Instructor Relation

The **department and Student** relations have (**Departmentname**, courses), (**StudentID**, name, major). The rules that presented in Table1 leads database designers to determine primary key. The attributes in the **Department** and **Student** relations are linked attributes. Follow the rules in Table 2 indicates that **Department** and **Student** relations mapped as regular entity shown in Figure4, Figure5.

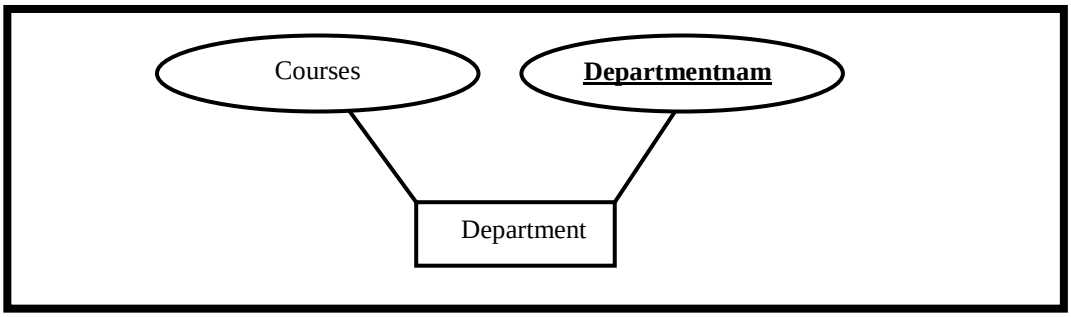

Figure 4. Representation of the Department Relation

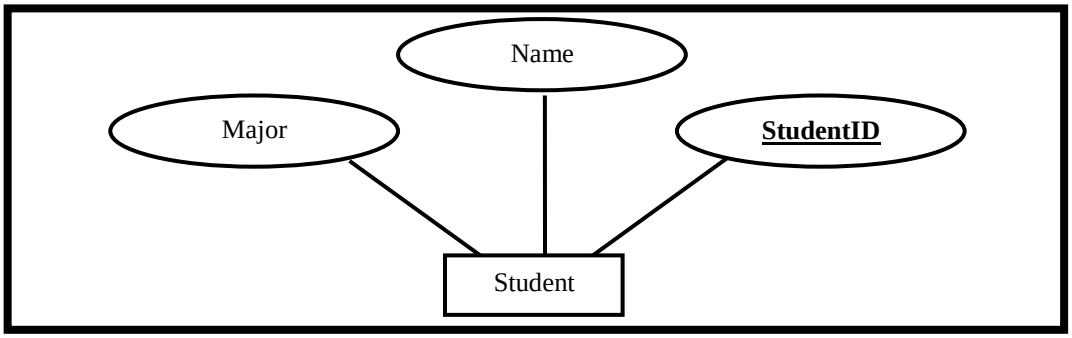

Figure 5. Representation of the Student Relation

When database designers view the relational database schema for the first time some information becomes clear. In addition on deeper study some important information will be determined. Figure1 contain **Dependent** relation, from the first view maybe the database designers believe that **Dependent** mapped as regular entity. Follow the rules that shown in Table2 by database designers indicate that the **Dependent** relation mapped as weak entity. Figure6 has shown the representation of this weak entity.

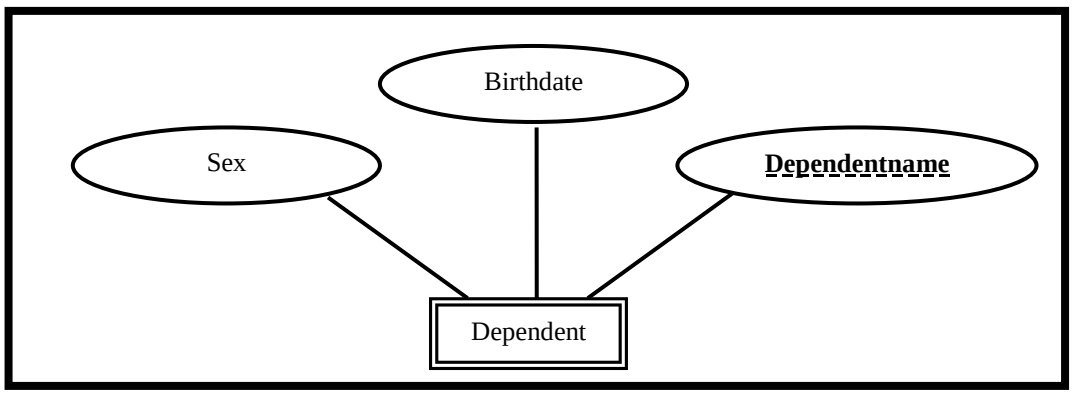

Figure 6. Representation of the Dependent Relation

7.2 Relationship Extraction

All relation with attributes related with another relation is mapped as relationship between two entity. The primary key in this relationship will be attributes that related with other relation. The mapping of **Ins_Stud** relation in Figure1 has shown in Figure7

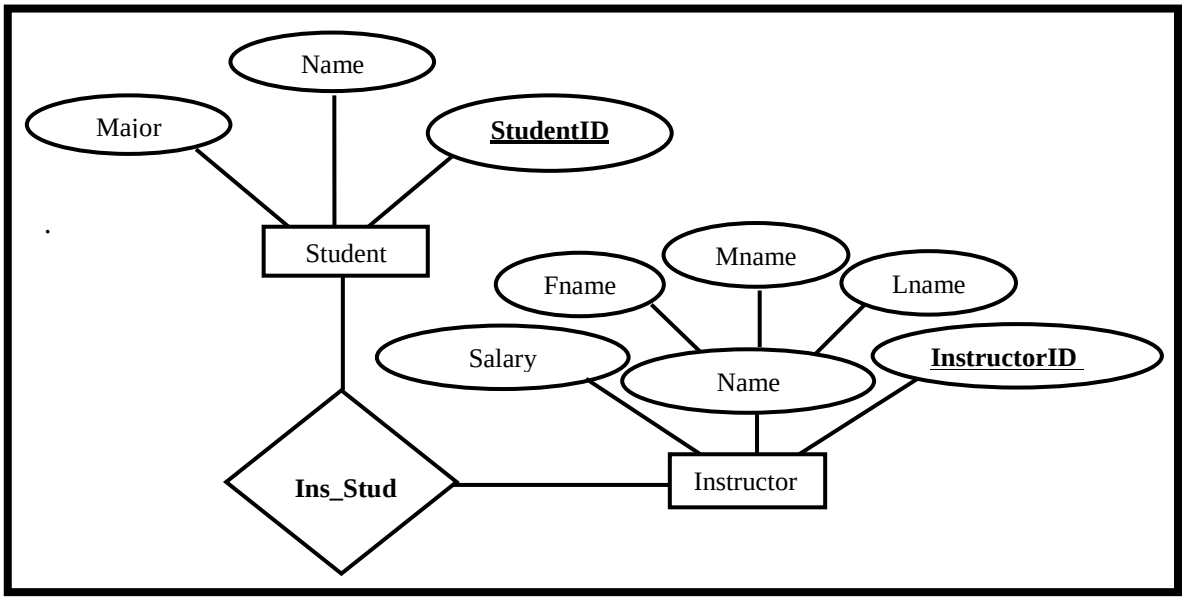

Figure 7. Representation of the Ins_Stud Relation

Extract relations from relational database schema will become easy and understandable when the database designers follow the rules that formulated in Table3. Figure 8-10 shows the representation of relationships between entities that shown in Figure1.

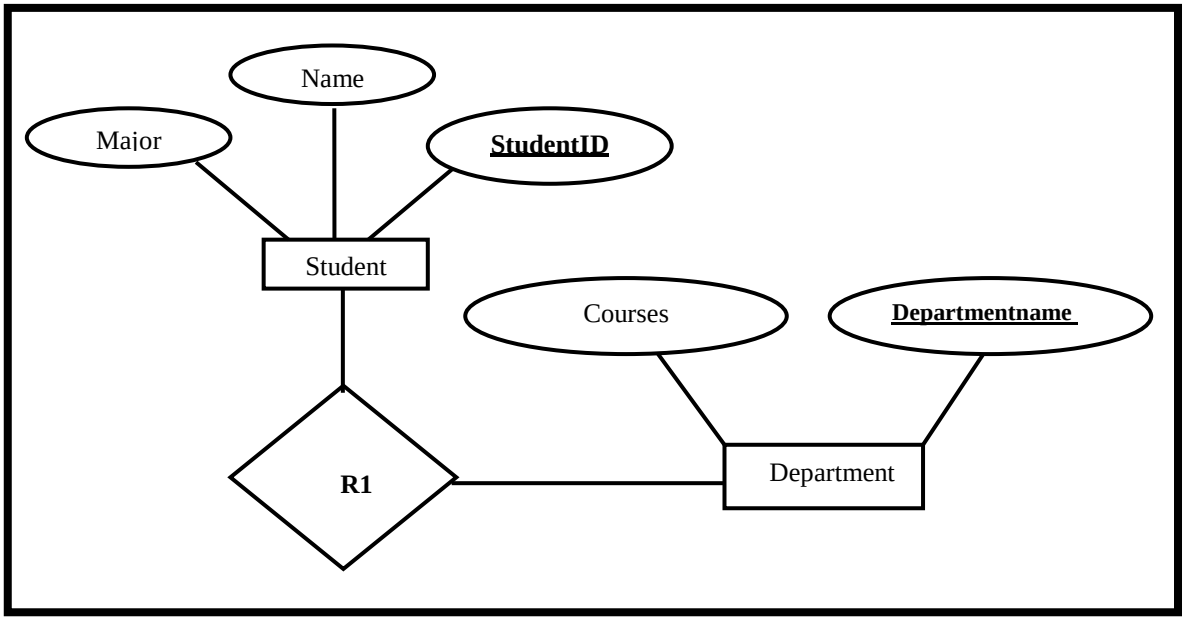

Figure 8. Representation of R1 between Student and Department Relations

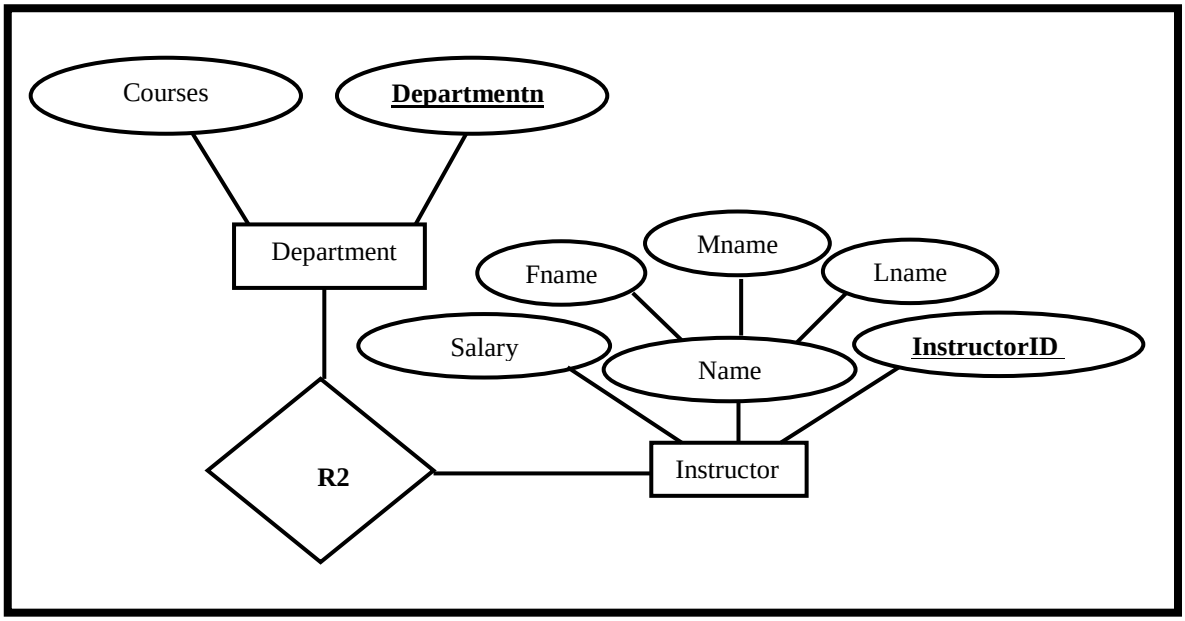

Figure 9: Representation of R2 between Instructor and Department Relations

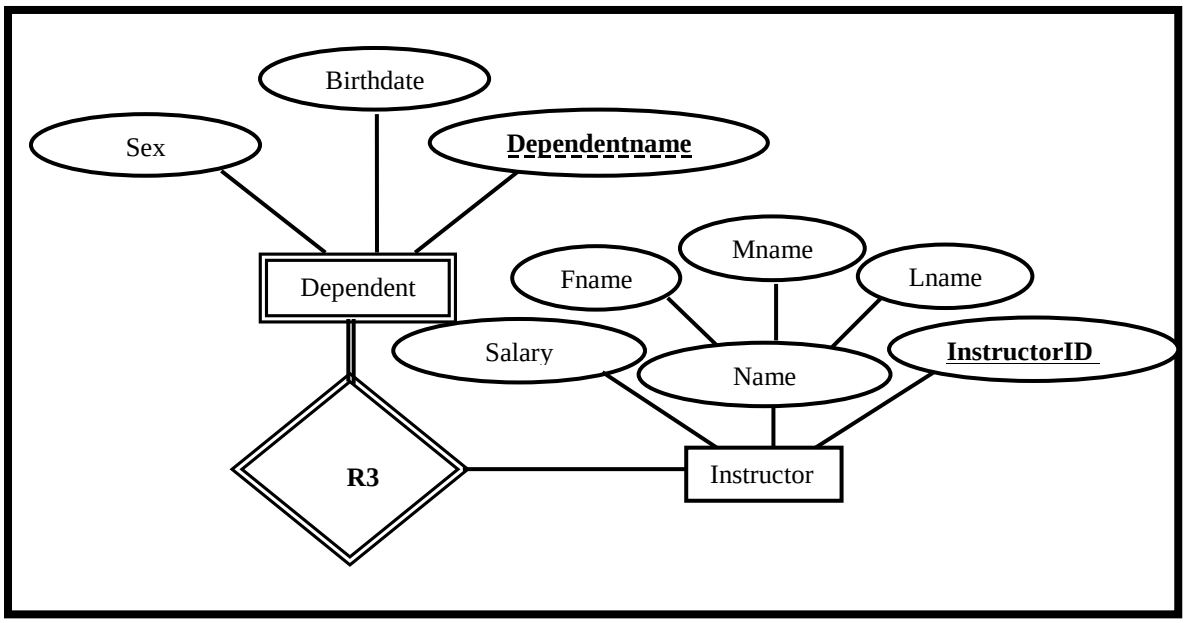

Figure 10. Representation of R3 between Instructor and Dependent Relations

7.3 Cardinality Determination

The rules that formulated in Table 4 lead database designer to determine the cardinality between entities. Table5 has shown the cardinality between each two entities.

Table 5. Cardinality between Entities

Relationship	*Cardinality*
- Ins_Stud - R1 - R2 - R3 - (R1, R2, R3)	(M: M) between student and instructor. (1: M) between department and student. (1: M) between department and instructor. (1: M) between instructor and dependent. - The cardinality between entities that related with R1, R2 or R3 maybe (1: 1). - The line from side of entity contains foreign key is total participating line. - The line from side of entity does not contain foreign key is partial participating line.

7.4 ERD Extraction

Database designers used the information that extracted to lead them to extract ERD. Figure11 has shown the ERD for relational database schema that shown in Figure1.

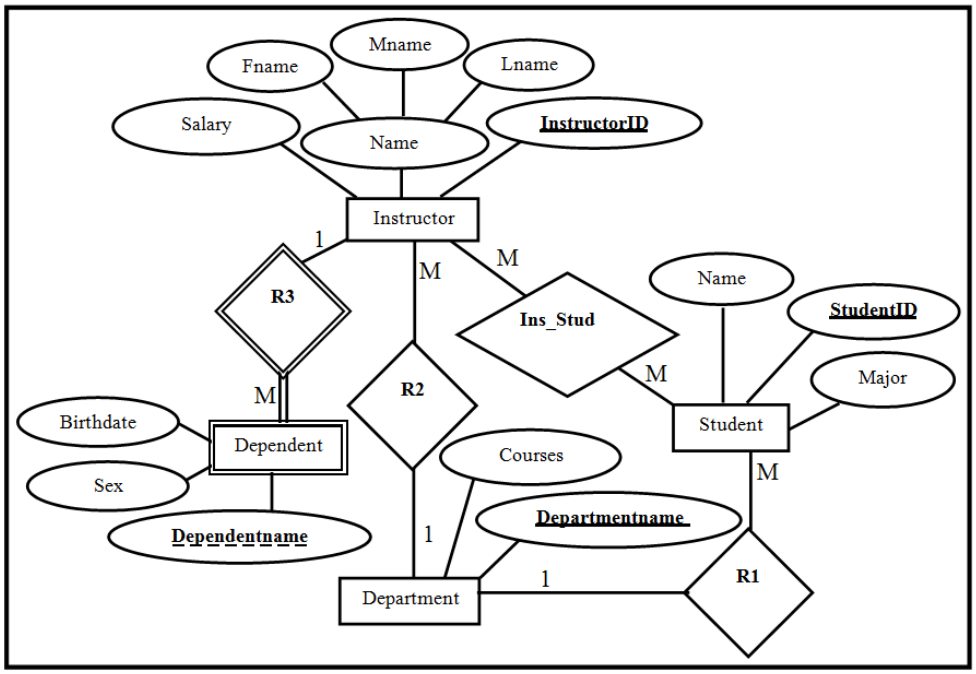

Figure 11. Representation of ERD Extraction from Relational Database

8. Conclusion and Future Work

In this paper, we have presented an approach used for extracting ERD from relational database schema. The relational database schema does not have primary keys proclaimed and we cannot say the database is a relational database. Furthermore, the descriptions for some fields of the database are having some shortage. In addition to that, most methods based in DBRE are inapplicable under these restrictions. For this objective it is necessary to be able to analyze the information about ERD that extracted from relational database schema and understand it. The proposed in this paper use the relational database schema as input. After analyzing the database, the relational database schema is inputted to organize and understand the structure of a database. The information that helps database designers to construct ERD is extracted. There are some procedures, fields and words

comparison, data and code analysis, used to determine primary and foreign keys, followed by determining the entities and their attributes. In the final step, relationships are determined with cardinality and constraints, and the ERD is extracted. In the future work, we will try to overcome some of proposed methodology limitation such as; this methodology has been applied only in the binary relationship. Therefore in the proposed methodology it is not easy to recognize between relationship and weak entity in all cases, especially when cardinality in relationship is (M: M) and the relationship have attribute. We will try to tackle these limitation by replicate this study to validate the information extraction results using more advanced rules. In particular, our results are based on keywords. However, the usefulness of extract ERD from relational database schema may depend on the table that contains data or script code of the database. Finally, in order to strengthening the approach and increasing its efficiency we are continuing to study cases. Addition to that, we will automate the proposed methodology.

References

[1] P. S. Dhabe, "Articulated Entity Relationship (AER) diagram for complete automation of relational database normalization", **(2010)**, pp. 84-100.
[2] F. Batmaz and C. J. Hinde, "A diagram drawing tool for semi–automatic assessment of conceptual database diagrams", **(2006)**, pp. 71-84.
[3] S. Xu, L. Yu and S. Lu, "ERDraw: An XML-based ER-diagram Drawing and Translation Tool", Computers and Their Applications, **(2003)**, pp. 143-146.
[4] H. Purchase, R. Welland, M. McGill and L. Colpoys, "Comprehension of diagram syntax: an empirical study of entity relationship notations", International Journal of Human-Computer Studies, vol. 61, no. 2, **(2004)**, pp. 187-203.
[5] N. Mfourga, "Extracting entity-relationship schemas from relational databases: a form-driven approach", Reverse Engineering, Proceedings of the Fourth Working Conference on. IEEE, **(1997)**.
[6] V. Cosentino and S. Martinez, "Extracting UML/OCL Integrity Constraints and Derived Types from Relational Databases", 13th International Workshop on OCL, Model Constraint and Query Languages, **(2013)**, pp. 43-52.
[7] D. Yeh, Y. Li, and W. Chu, "Extracting entity-relationship diagram from a table-based legacy database", Journal of Systems and Software, vol. 81, no. 5, **(2008)**, pp. 764-771.
[8] P. P.-S. Chen, "English sentence structure and entity-relationship diagrams", Information Sciences, vol. 29, no. 2, **(1983)**, pp. 127-149.
[9] M. H. Alalfi, J. R. Cordy and T. R. Dean, "SQL2XMI: Reverse engineering of UML-ER diagrams from relational database schemas", Reverse Engineering, 2008, WCRE'08. 15th Working Conference on. IEEE, **(2008)**.
[10] C. Ordonez, "Extending ER models to capture database transformations to build data sets for data mining", Data & Knowledge Engineerin, vol. 89, **(2014)**, pp. 38-54.
[11] Reda Alhajj, "Extracting the extended entity-relationship model from a legacy relational database", Information Systems, vol. 28, no. 6, **(2003)**, pp. 597-618.
[12] T. Lee, "Attribute extraction and scoring: A probabilistic approach", Data Engineering (ICDE), 2013 IEEE 29th International Conference on. IEEE, **(2013)**.
[13] A. Maatuk, A. Ali and N. Rossiter, "Relational database migration: A perspective", Database and Expert Systems Applications, Springer Berlin Heidelberg, **(2008)**, pp. 676-683.
[14] P. Thi Thu Thuy, "RDB2RDF: completed transformation from relational database into RDF ontology", Proceedings of the 8th International Conference on Ubiquitous Information Management and Communication. ACM, **(2014)**, pp. 88-95.
[15] P. de-Jesus, L. and P Sousa, "Selection of reverse engineering methods for relational databases", Software Maintenance and Reengineering, 1999, Proceedings of the Third European Conference on. IEEE, **(1999)**.
[16] I. D. Baxter and M. Mehlich, "Reverse engineering is reverse forward engineering", Reverse Engineering, 1997. Proceedings of the Fourth Working Conference on. IEEE, (1997).
[17] N. Lammari, I. Comyn-Wattiau, and J. Akoka. "Extracting generalization hierarchies from relational databases: A reverse engineering approach", Data & Knowledge Engineering, vol. 63, no. 2, **(2007)**, pp. 568-589.
[18] D. W. Rosen, N. Jeong and Y. Wang, "A Method for Reverse Engineering of Material Microstructure for Heterogeneous CAD", Computer-Aided Design, vol. 45, no. 7, **(2013)**, pp. 1068-1078.

