



Generation of Rules from the Relational Database Structure

Olegs Verhodubs

EasyChair preprints are intended for rapid dissemination of research results and are integrated with the rest of EasyChair.

March 6, 2023

Generation of Rules from the Relational Database Structure

Olegs Verhodubs

oleg.verhodub@inbox.lv

Abstract. The structure of the relational database can provide rule-based and expert systems with rules. For this purpose, it is necessary to use the relational database table structures and relationships between database tables. This paper continues previous researches aimed at retrieving of knowledge from different information sources as OWL (Web Ontology Language) [1] ontology and raw text from some webpage or book.

Keywords: Rules, Databases, Expert Systems, Semantic Web

I. Introduction

The lack of information is as harmful to any system, as an overabundance of information. Here harm to the system means negative impact on system efficiency, reliability and adaptability. The inability to provide the necessary amount of information has been a problem for a long time, but now this problem is leveled. On the contrary, the problem of lack of information turned into its opposite when there was too much information, but the problem of system management did not disappear from this. To some extent, the situation with a lack of information is less painful than it is now, when systems and organizations suffer from an overabundance of information. This was so, because earlier it was possible to use historical models to control systems. Now it is necessary to respond to a huge flow of information that is pouring uncontrollably from all sides.

In this situation, it is obvious that the information should be processed in some way. There are several ways, how information can be processed. The first way is to rank information according to some criteria, or prioritize it to highlight more important and less important information. By responding to information that is more important and ignoring less important information, the system can be kept in a stable state. The main task of this way is selection of adequate criteria for the importance of information. In any case, this way discards some of the information without considering it further, which can lead to problems if this information is erroneously assessed as unimportant. The second way is to process all of information thus making a kind of squeeze out of the information. This way seems preferable, since in this case all information is taken into account. A kind of squeeze out of the information can be knowledge, represented in the form of rules. In meaning, information and knowledge is akin to function and derivative in mathematics. In turn, rules are used in expert systems and rule-based systems in order to inference utilizing some facts. Rules can be described differently, but it is also possible to use OWL language to code them. The use of rules in the notation of OWL is convenient, because there are many tools namely semantic reasoners to work with them.

This research continues previous researches in the field of knowledge retrieving from various sources such as OWL ontologies [2], [3] and raw text [4] of webpages or books. Thus, this research supplements previous ones, because expands the information base, from which knowledge that is rules can be generated. Now information base for generation of rules are OWL ontologies, raw texts and databases. The possibility of rule generation from various sources is necessary for expert systems and rule-based systems. One of such a rule-based system is a Keyword Search Engine Enriched by Expert System Features [5].

This paper has several sections. The next section describes, how rules can be generated from relational databases. After that comes the conclusion.

II. Database structure to rules

Database is an instance of data portion, stored in DBMS (database management system). DBMS is a storage, where databases are located. That is, one DBMS stores many databases. A relational DB has an established structure that is either not being modified at all, or is modified extremely rarely. An established or static data structure can be used to generate rules. A relational database is such a database, which is built on a relational data model. A relational data model is not the only data model, but it is the most common. That is why generation of rules from relational database (relational data model) is covered here, however any static data model is suitable to generate rules.

A relational database is based on relational data model, which means that all data is divided into interconnected tables. Each table in the relational database defines one entity, whether it is an object, process or phenomenon. The use of the word “defines” means that for each entity a set of properties is specified. A set of properties in a relational database is traditionally displayed as columns in the table. For example, the structure of the table that describes the entity “Car” can be represented like this (Fig.1):

Car

Registration_No	Make	Model	Color	Year
...

Fig.1. A “Car” table structure

A “Car” table (Fig. 1) has the properties registration_no, make, model, color and year that is why it is possible to generate the following rule:

IF *registration_no* **and** *make* **and** *model* **and** *color* **and** *year* **THEN** *Car* (I)

Usually, any database has many tables, and each table provides the potential rule-based system with at least one rule.

A table in a relational database can have primary key and foreign keys. A primary key is the column or columns that contain values that uniquely identify each row in a table [6]. A foreign key is a column or columns of data in one table that refers to the unique data values -- often the primary key data - in another table [7]. Thus, foreign keys link together two or more tables in a relational database. Tables linked by foreign keys give an opportunity to generate one more type of rule. For example, the structures of two tables “Student” and “Faculty” that are linked by means of foreign keys can be represented like this (Fig. 2.):

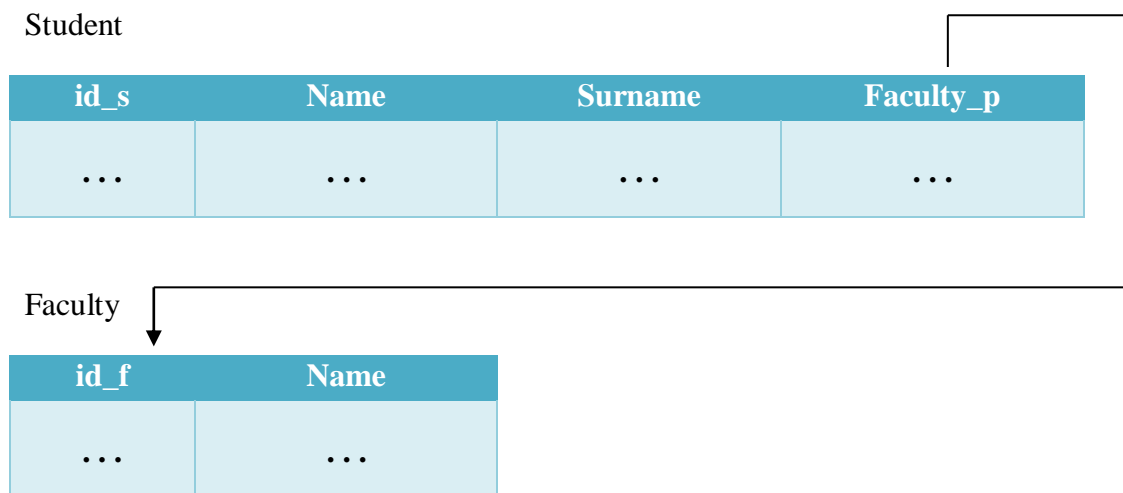


Fig.2. “Student” and “Faculty” tables.

The “Student” and “Faculty” tables are linked, and this is provided with foreign key that is included in the table “Student”. Such a structure of two tables is possible to use for generating the following rule:

IF *Student* **and** *property(Student, Faculty_p, foreign key)* **and** *Faculty*
THEN Student ∈ Faculty (II)

Here the entry *property(Student, Faculty_p, foreign key)* means that the table “Student” has the property (column) “Faculty_p”, which is a foreign key.

A similar case, where in addition there is one more linked table, also makes it possible to generate a rule. There are tables “Student”, “Faculty” and “University”; table “Student” is linked to table “Faculty” and table “Faculty” is linked to table “University” (Fig. 3):

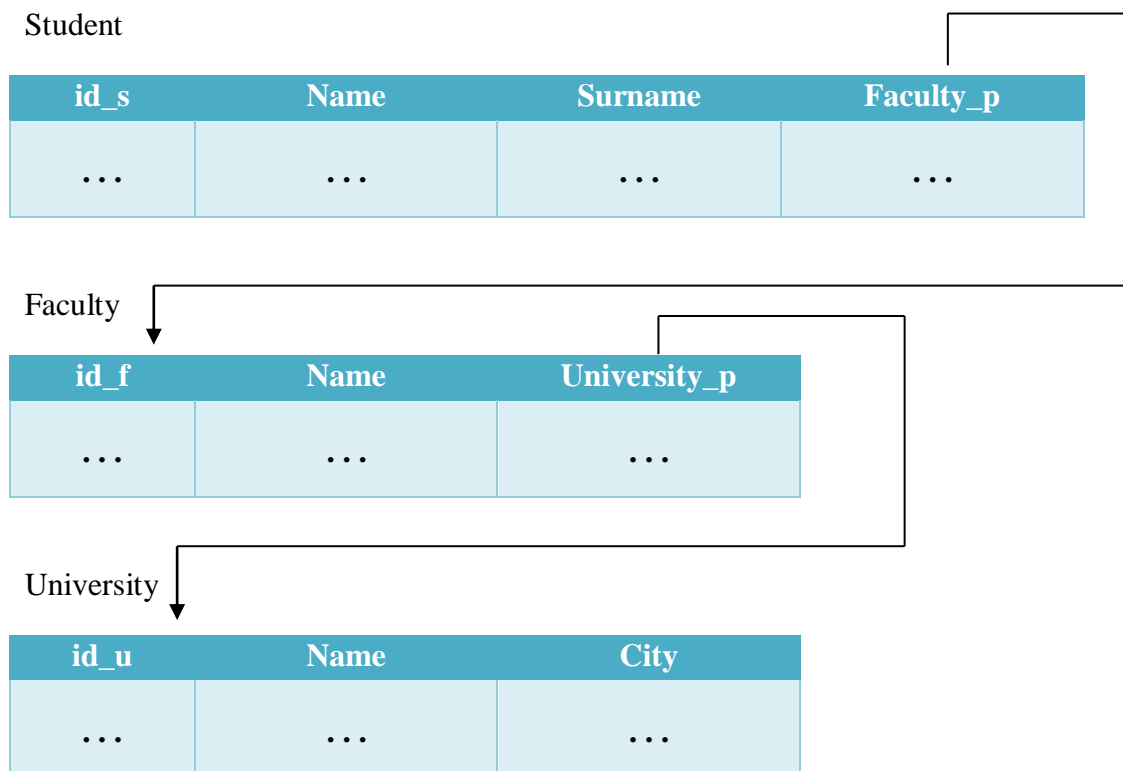


Fig.3. “Student”, “Faculty”, “University” tables.

The rule that is generated from the structure can be called as “transitive” and it looks like this:

IF

Student and property(Student, Faculty_p, foreign key) and

Faculty and property(Faculty, University_p, foreign key) and University

THEN

Student \in University

(III)

Here the entry *property(Student, Faculty_p, foreign key)* means that the table “Student” has the property (column) “Faculty_p”, which is a foreign key and the entry *property(Faculty, University_p, foreign key)* means that the table “Faculty” has the property (column) “University_p”, which is a foreign key.

The structure of three tables in relational database, where one of the tables has two foreign keys, can use to generate a rule, too. For example, there are three tables “Car”, “Driver”, “Organization” and the table “Car” has two foreign keys (Fig. 4):

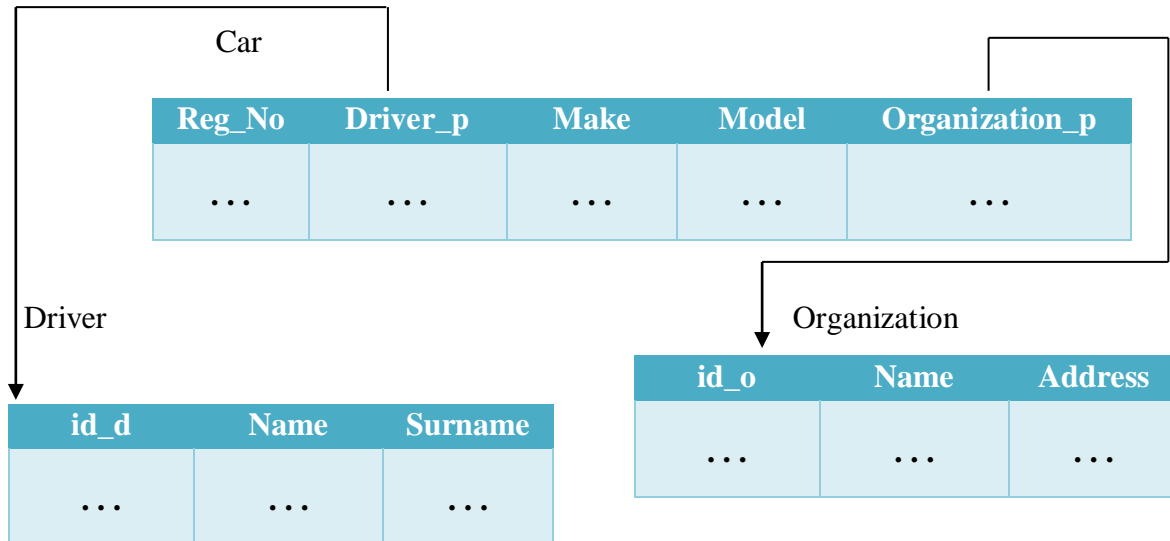


Fig.4. “Car”, “Driver”, “Organization” tables.

Such a structure can be used to generate the following rule:

IF *property*(*X*, *Driver_p*, *foreign key*) **and** *property*(*X*, *Organization_p*, *foreign key*)
THEN Car (IV)

Here the entry *property*(*X*, *Driver_p*, *foreign key*) means that some table has the property (column) “Driver_p”, which is a foreign key and the entry *property*(*X*, *Organization_p*, *foreign key*) means that some table has the property (column) “Organization_p”, which is a foreign key, too. But it is better to rephrase this rule in such a way: if something belongs to driver and organization, then this is a car.

There is confidence that it is possible to generate other rules from the structure of a relational database, but these cases will not be considered here. It is worth paying a little attention to the question of how to use the obtained rules. Of course, there is a way to program an algorithm that will use the received rules, i.e. this method involves creating your own inference engine. Another way is to use an existing inference engine. Jena inference engine can be utilized for reasoning [8], and this inference engine (or reasoner) is rather comfortable, because gives an opportunity to work in different modes. One of such modes in Jena is “general purpose rule engine” [9], which is exploited to inference based on “IF..THEN” rules. The syntax of rules in the notation of Jena is the following:

[rule1: (?f :mother ?a) (?u :sister ?f) -> (?u :aunt ?a)] (V)

This rule in Jena notation is similar to this entry:

IF (f is mother_of a) and (u is sister_of f) **THEN** (u is aunt_of a) (VI)

It should be said that Jena was developed to work with the semantic data within the Semantic Web paradigm, and it was much more powerful than it is needed here.

III. Conclusion

The process of knowledge generating from information can help people to cope with an ever-increasing amount of information. Knowledge in the form of rules are used in different expert and rule-based systems, that is why the task of knowledge retrieving is a key task for functioning of many real life systems. This paper is dedicated to the task of rule generation from the relational database structure. In total, four cases, when a rule can be generated from the relational database, are described in this paper. Supplementing the knowledge base of the expert system or rule-based system with rules, retrieved from the database structure, complements those rule generation capabilities that have been described previously, such as rule generation from ontologies and raw text. Despite the usefulness of this research for many rule-based systems, this research conceived as another means to populate the Keyword Search Engine Enriched by Expert System Features with rules. Historically, Keyword Search Engine Enriched by Expert System Features grew out of the Semantic Web Expert System [10] and expanded its capabilities.

There are some complexities that could hinder to generate rules from the relational database structure. The first obstacle is limited access to the database and database structure. Unlike webpages, only specified users can access the databases and database structure. The second hurdle concerns database table names. For example, some table could have the name “Cars”, but not “Car” that was why generated rule would be incorrect. Thus it turns out that it is necessary to preprocess database table names.

Acknowledgments

This research was carried out at the expense of the author.

References

- [1] <https://www.w3.org/OWL/> [Accessed: 28.02.2023]
- [2] Verhodubs O.: Grundspenkis J. Evolution of ontology potential for generation of rules (2012)
- [3] Verhodubs O.: Ontology as a Source for Rule Generation (2014)
- [4] Verhodubs O.: Experiments of Rule Extraction from Raw Text (2021)
- [5] Verhodubs O.: Keyword Search Engine Enriched by Expert System Features (2020)
- [6] <https://www.ibm.com/docs/en/iodg/11.3?topic=reference-primary-keys> [Accessed: 02.03.2023]
- [7] <https://www.techtarget.com/searchoracle/definition/foreign-key#:~:text=A%20foreign%20key%20is%20a,tables%20in%20a%20relational%20database> [Accessed: 02.03.2023]
- [8] <https://jena.apache.org/> [Accessed: 04.03.2023]
- [9] <https://jena.apache.org/documentation/inference/index.html#rules> [Accessed: 04.03.2023]
- [10] Verhodubs O., Grundspenkis J.: Towards the Semantic Web Expert System (2011).