Ontology Driven for Mapping a Relational Database to a Knowledge-based System

Abdelrahman Osman Elfaki, Yousef H. Alfaifi
Faculty of Computers and Information Technology
University of Tabuk, Tabuk 71491, Saudi Arabia

***Abstract*—The mapping of a relational database system to a knowledge-based system is a key stage in developing an online analytical processing (OLAP) system. OLAP is a cornerstone in discovering hidden knowledge in any business. Hence, the existence of an OLAP system is one of the modern success factors in a business environment. Mapping has proven benefits for knowledge-based systems in terms of enabling the discovery of hidden relationships among objects and the inference of new information. However, there remains room for improvement in respect of the quality of the mapping output. Therefore, in this paper, a rule-based method for mapping a relational database to a knowledge-based system is introduced. First, the proposed mapping process, which involves converting the tables and relationships of a relational database into facts and rules for a knowledge-based system, is illustrated through the use of a detailed case study. Then the correctness of the proposed method is proved by testing the tautology results against equivalent SQL queries. In addition, the completeness of the proposed method is proved by demonstrating that the used predicates are sufficient to allow a complete modeling of the required system. Furthermore, the experimental results show that the performance of the knowledge-based system that was developed using the proposed method is much better than that of an equivalent relational database.**

Keywords—Mapping knowledge; ontology-based; relational database; online analytical processing

I. Introduction

A relational database is a well-established data model that is used to store, manipulate, and manage data. Therefore, the relational database is suitable for use in online transaction processing (OLTP) systems. On the other hand, a knowledge-based system is a suitable data model for online analytical processing (OLAP) systems [1]. As a matter of fact, it is widely acknowledged that OLAP systems should developed based on OLTP systems [2]. Hence, it follows that in order to facilitate the development of an OLAP system, it is necessary to be able to accurately map the data of an OLTP to fit the OLAP system. However, there is a significant lack of effective and efficient mapping methods for converting a relational database into a knowledge-based system. Moreover, according to [3], there is a critical need to provide for the cohabitation of relational databases and ontological technologies. An ontology is developed based on logic, therefore, finding a way to represent a relational database by using logic would bridge the gap between ontological and relational databases.

According to the literature, the ability to convert a relational database into a knowledge base not only provides the requisite support for the development of OLAP systems, but it would also offer four other main benefits. First, it would improve integration and overlapping between database and knowledge base in decision support systems [4]; second, it would provide semantic support for inferring new facts from existing data, third, it would enhance query performance [5]; and fourth, it would enhance scalability and flexibility which would then encourage companies in a wide range of sectors to replace their traditional database management systems (DBMSs) with NoSQL databases [6].

Mapping of relational databases to a knowledge-based system is an important research topic, mainly for information integration, ontology-based data access and for sharing data on the web in a form of a knowledge base that could be a subject of automatic reasoning procedures. Then a knowledge base should represent the underlying relational database as accurately as possible.

Due to the aforementioned benefits of mapping a relational database to a knowledge base, this challenge has captured the attention of the computer science community for over 20 years ago [7,8]. However, despite the fact that numerous research studies have attempted to deal with the mapping of a relational database to a knowledge-based system, the methods that have been proposed thus far suffer from a lack of proven completeness and correctness in the resulting mapping outcome. These drawbacks and the reasons for them are discussed in Section 2 “Related works”.

In an attempt to address these drawbacks, in this study, a rule-based method for mapping a relational database to a knowledge-based system is developed. One of the main issues in the related research studies is the lack of consideration that has been paid to foreign and primary keys in the mapping process. According to [9] column constraints such as primary, foreign, and unique keys are a non-trivial challenge in the mapping of a relational database to a knowledge base. Therefore, two rules are developed to support primary and foreign keys and these rules are incorporated into the proposed method. Then, the applicability, completeness, and correctness of the proposed rule-based method is tested and proved. In addition, an experiment is conducted to compare the performance of the proposed method against that of a relational database.

The contribution of this proposed ontology could be summarized as:

- Provide Free search: The proposed knowledge-based method enables the search for a database value without requiring knowledge of its table or field.
- Provide better performance.
- Provide a suitable data model for OLAP.

The remainder of this paper is structured as follows: In Section 2, an overview of related works is provided with a focus on the strengths and weaknesses of each work. Next, in Section 3, a descriptive case study is presented in order to illustrate the mapping process and the syntax and semantics for the utilized predicates are presented. Then, in Section 4, the mapping process is explained in detail. This is followed by Section 5, in which some query rules are illustrated to explain how output results could be generated from the proposed method. The correctness of the proposed method is also proved in this section. After that, in Section 6, the implementation of the proposed rules is presented to prove the applicability of the proposed method. Finally, in Section 7, the completeness and performance of the proposed method are discussed followed by a conclusion and a brief overview of intended future work.

II. Related Works

The related works that are discussed in this section are categorized into (1) long-running and well-established knowledge conversion projects, (2) works that have converted SQL databases into knowledge bases, and (3) works that have converted SQL databases into ontologies.

A. Well-Established Knowledge Conversion Projects

Three of the most famous projects that have used databases after converting them to knowledge-based are: (1) Dbpedia [10], which provides knowledge that has been extracted from different Wikimedia structured content; (2) KBpedia [11], which is structured knowledge combined from several knowledge repositories; and (3) Schema.org [12], which is a knowledge-based representation for Internet data, where Schema.org vocabulary can be used which make it a suitable platform for different knowledge-based research studies.

The aforementioned works are instances of well-established knowledge-based repositories that define knowledge in terms of concepts and the relationships between these concepts. In these knowledge-based systems the information is converted from a traditional SQL database into a knowledge graph that consists of concepts and their relationships. However, the mapping procedures that are used to convert the information from structured data (i.e., database) and unstructured data into the knowledge graph are not clarified. This ambiguity leads to a problem when attempting to integrate new databases into existing systems. Therefore, in this study, clear rules for mapping relational databases to knowledge-based systems are proposed.

In addition to aforementioned studies, there are several commercial knowledge-based systems that can produce intelligent results after analyzing the targeted database. This analysis process also requires mapping from the database to the knowledge-based system. These systems substantiate the usefulness and benefits of mapping databases to knowledge-based systems. However, a discussion of the specialized commercial knowledge-based systems that are currently available is beyond the scope of this paper.

B. Works on Converting SQL Databases into Knowledge bases

Numerous research studies have been conducted on converting SQL databases into knowledge bases, some of which date back more than 30 years. Here, only the most recent and influential of these studies are discussed.

The first of the recent studies that is worthy of mention is that by [9], who developed a set of rules for defining dependencies between a relational database and a knowledge base. However, the completeness and correctness of those rules was not presented. Another noteworthy study is that by [13], who designed a knowledge base as an architecture model for integrating different distributed DBMSs. This architecture model demonstrates the powerfulness of the knowledge-based database system in terms of scalability and availability, in addition to the advantage of facilitating the creation of integrated distributed DBMSs. However, a technical description for the transfer of data from the traditional database system (i.e., the SQL database), to the knowledge base was not provided by the authors. Hence it is difficult to prove the completeness or correctness of their work, i.e., whether it is applicable for all types of DBMS. On the other hand, the work in [14] proposed an approach for developing a knowledge-based system from open-source relational databases. However, their work is limited to a Chinese database and there is no published proof for its completeness.

More recently, the work in [4] proposed an algorithm that could be used to transfer the contents of a SQL database to a knowledge base. The proposed algorithm consists of seven general steps that describe the transformation. However, the proposed algorithm lacks technical descriptions to guide the transformation process. Finally, the work in [6] proposed a model to transform an object relational database into a NoSQL column-based database. Their work lacks of a completeness, and its correctness has not been proven.

C. Works on Converting SQL Databases into Ontologies

An ontology is considered to be a modern representation approach to the knowledge-based system [15], which means that it is considered in the domain of this study. One of the previous studies that is relevant to this study work is that in [16], who proposed a method for automatically converting a database into an ontology. Also of interest to this study is the work in [17], who proposed a rule-based system for mapping a relational database to an ontology. However, the methods that were proposed in these two previous works were not evaluated to prove their completeness.

On the other hand, the work in [18] has developed a method for mapping the entity relationship diagram (ERD) to the semantic web. This method is converting only ternary and binary relationships. Other works by the work in [3] proposed an approach for converting a relational database into an ontology, while the works in [19,20] proposed methods for automatically mapping a relational database to an ontology. More recently, the work in [21] proposed a method for converting ERDs into a knowledge-based system. Yet, again, none of the aforementioned works tested the completeness of their approaches. Lastly, the work in [22] proposed a method for mapping a relational database containing information on dengue patients to a dengue patient ontology. As this method was limited to dengue patient information, it would seem to have limited generalizability.

In summary, in the light of the above review of recent related works, it is obvious that there is a crucial need for a method that not only can map a relational database to a knowledge-based system, but which is also tested for completeness and correctness. Such a method could provide an optimal solution for the OLAP system, which is one of the current focal topics of interest among researchers and software developers due to its proven influence on business success.

III. Case Study

This study uses a tailored sales system as a case study to clarify the proposed mapping method and illustrate the technical details using a relational database. The Entity Relationship Diagram is applied to represent the objects and the relationships among these objects in the tailored sales system. ERD is a graphical representation of the entities (objects or concepts) commonly used to visualize the structure of a database and the relationships between different types of data. Fig. 1 illustrates the ERD of the sales system. This ERD includes five entities (*Seller*, *Sales*, *Items*, *Shop*, and *City*) and the relationships among these entities. In addition, primary keys (PK) and foreign keys (FK) for these entities are defined.

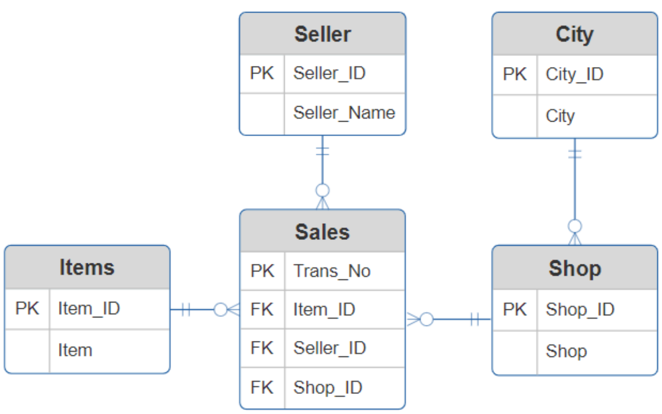

Fig. 1. The relational database of the tailored sales system by using ERD

To complete the illustration of the case study, synthetic data is used to provide a snapshot of the sales system. Tables I, II, and III represent snapshots of the *Item*, *Seller*, and *Sales* entities, respectively. These synthetic data are used to illustrate the mapping process and to later prove the completeness and correctness of the proposed method.

TABLE I. Snapshot of Items Entity

Item_ID	Item
1	Computer
2	Printer
3	Handphone

TABLE II. Snapshot of Seller Entity

Seller_ID	Seller_Name
1	Kevin
2	John
3	David

TABLE III. Snapshot of Sales Entity

Trans_No	Shop_ID	Item_ID	Seller_ID
1	1	1	1
2	1	2	2
3	1	1	2
4	2	1	1

This proposal aims to emphasize the importance of modeling OLAP systems as knowledge-based. It is crucial to acknowledge that readers from information systems or business backgrounds may not be familiar with knowledge-based notations. Therefore, in order to address this, the proposal explains a sales system using SQL (Section V).

IV. Mapping Process

In this section, the mapping process employed in the proposed method is presented by using an illustrative example based on the above case study database. A relational database consists of tables and the relationships between them. A table and its columns and values can be formalized by the following formula:

$$T\{C1(V_1,V_2,\dots V_n),C2((V_1,V_2,\dots V_n),\dots Cn(V_1,V_2,\dots V_n)\}$$

where T denotes the table, C denotes the column, and V denotes the values. For instance, Table 6 can be represented by using the above formula as:

Item {Item_ID(1,2,3), Item(computer, printer, hand phone)}.

To convert the table structure so that it can be represented as a knowledge base, five predicates should be used: member, value_of, same_rec, primary_key, and foreign_key. The syntax and semantics of each predicate are as follows:

1. *Member(C,T)*

Syntax: member(C,T)

Semantic: column C belongs to table T.

2. *Value_of(V,C)*

Syntax: value_of(V,C)

Semantic: value V belongs to column

Here, the assumption is that each column has at least one value. In the case of an empty value, zero (0) replaces the empty value.

3. *same_rec(T,V1,V2)*

Syntax: same_rec(T, $V_1, V_2, \dots, V_n$*)*

Semantic: The values V1, V2, V_n *belong to the same record in table T.* V_n *denotes the last element in the record.*

4. *primary_key(T,CK)*

Syntax: primary_key(T,C_k)

Semantic: C_k is a primary key in table T.

5. *foreign_key(T,Cf,Tr, Cr)*

Synatx: foreign_key(T,Cf,Tr, Cr)

Semantic: C_f is a foreign key in table T, where T_r is a reference table and C_r is a reference key for the foreign key C_f.

In the case of multiple primary keys, the predicate primary_key(T,Ck) is repeated to satisfy the existence of the number of primary keys. For instance, suppose there are three primary keys in table T, then the knowledge base will contain the following three predicates: primary_key(T,Ck1), primary_key(T,Ck2), and primary_key(T,Ck3). Primary key concept covers the unique constraint as well.

In the case of multiple foreign keys, the predicate foreign_key(T,Cf,Tr, Cr) is repeated to satisfy the existence of the number of foreign keys. For instance, suppose there are three foreign keys in table T, then the knowledge-based will contain the following three predicates: foreign_key(T,Cf,Tr, Cr), foreign_key(T,Cf,Tr, Cr), and foreign_key(T,Cf,Tr, Cr).

Assumption: Each column in the database should have a unique name. For instance, in the "Items" table, the primary key is "Item_ID", which is a foreign key in the "Sales" table, hence, in the "Sales" table the foreign key should have different name, i.e., "Sales_Item_ID''.

By using the above predicates, the sales system can be transferred to the knowledge -base. Table IV shows a snapshot of the knowledge representation of the sales system.

TABLE IV. SNAPSHOT OF KNOWLEDGE REPRESENTATION OF THE SALES SYSTEM

//Items Entity member(Item_ID, Items). member(Item, Items). value_of(1, Item_ID). value_of(computer, Item). same_rec(Items,1, computer). value_of(2, Item_ID). value_of(printer, Item). value_of(3, Item_ID). value_of(handphone, Item). same_rec(Items,1, computer). same_rec(Items, 2, printer). same_rec(Items, 3, handphone). primary_key(Items, Item_ID).
//Seller Entity member(Seller_ID, Seller). member(Seller_Name, Seller). value_of(1,Seller_ID). value_of(2,Seller_ID). value_of(3,Seller_ID). value_of(kevin,Seller_Name). value_of(john,Seller_Name). value_of(david,Seller_Name). same_rec(Seller,1,kevin). same_rec(Seller,2,john). same_rec(Seller,3,david). primary_key(Seller, Seller_ID).
//Sales Entity member(Trans_No, Sales). member(Shop_ID, Sales). member(Sales_Item_ID, Sales). member(Sales_Seller_ID, Sales). value_of(1,Trans_No). value_of(1,Shop_ID). value_of(1,Sales_Item_ID). value_of(1,Sales_Seller_ID). value_of(2,Trans_No). value_of(1,Shop_ID). value_of(2,Item_ID). value_of(2,Seller_ID). same_rec(Sales, 1,1,1,1). same_rec(Sales, 2,1,2,2). primary_key(Sales,Trans_No). foreign_key(Sales, Sales_Shop_ID,Shops,Shop_ID). foreign_key(Sales, Sales_Item_ID,Items,Item_ID). foreign_key(Sales, Sales_Seller_ID,Seller,Seller_ID).

D. Verification Rules

Verification rules are used to assure the integrity of a system by ensuring that the primary and foreign keys have been implemented correctly, as shown in equations (1) and (2), respectively.

1) Primary key

$$\forall \text{ T,C,V: primary\_key(T,C)} \wedge \text{member(C,T)} \wedge \text{value\_of(V1,C)} \wedge \text{value\_of(V2,C)} \wedge \text{not\_equal(V1,V2) True} \quad (1)$$

Rule 1 returns true if the primary key condition is satisfied correctly. Rule 1 denotes that column C is a member and primary key in a table T and all the values in column C are unique. The predicate not_equal(V1,V2) returns true when the two values V1 and V2 are not equal. In the case of an empty value for V1 or V2, the predicate not_equal(V1,V2) returns a false value, in which case the primary key condition is be satisfied. Therefore, Rule 1 ensures that all the values in the primary key column are unique and not null. For instance, in the "Items" table, Items = T, Item_ID = C.

2) Foreign key

$$\forall \text{ T,C,V: foreign\_key(T,Cf,Tr, Cr)} \wedge \text{member(Cf,T)} \wedge \text{member(Cr,Tr)} \wedge \text{value\_of(V1,Cf)} \wedge \text{value\_of(V2,Cr)} \wedge \text{equal(V1,V2)} \Longrightarrow \text{True} \quad (2)$$

Rule 2 returns true if the foreign key condition is satisfied correctly. Rule 2 denotes that Cf is a foreign key and a member in table T, and its reference is a column Cr which is a member of a reference table Tr. For instance, in the "Sales" table: Sales = T, Sales_Shop_ID = Cf, Shops = Tr, and Shop_ID= Cr.

In case of multiple foreign keys (two keys) rule 2 could be applied as follow:

$$\text{foreign\_key(T,Cf1,Cf2Tr,Cr1, Cr2)} \wedge \text{member(Cf1,T)} \wedge \text{member(Cf2,T)} \wedge \text{member(Cr,Tr)} \wedge \text{value\_of(V1,Cf1)} \wedge \text{value\_of(V2,Cr1)} \wedge \text{equal(V1,V2)} \wedge \text{value\_of(V3,Cf2)} \wedge \text{value\_of(V4,Cr2)} \wedge \text{equal(V3,V4)} \Longrightarrow \text{True} \quad (3)$$

Note that the aforementioned five predicates that accompany Rules 1 and 2 are demonstrated the Data Definition Language (DDL).

V. QUERY RULES

Next, some query rules were developed to prove that the knowledge-based system developed from the proposed method could produce the same results as an equivalent relational database. The query rules below represent all the types of selection command that are used in Data Manipulation Language (DML).

In any relational database system, results are achieved by answering requests that are sent via query system. To prove the correctness of the knowledge-based system that was developed using the proposed method, the extraction of the data from the transferred knowledge base is explained by a set of five rules that cover all query cases: Rule 3 demonstrates the selection from one table without any condition, which means selecting all the columns from a table. Rule 4 demonstrates the selection of a value from one column in a table. Rule 5 demonstrates the selection of a specific value from a table. Rule 6 demonstrates the selection of two different values from two different tables. Rule 7 demonstrates the selection of three different values from three different tables. These rules are presented and explained below:

1) Select all columns from one table without condition

Two steps must be followed to select all the columns from one table without condition. The first step is to define the number of columns in a target table. The second step is to use the predicate same_rec on the defined number of columns in the previous step, as follows:

1) Define number of columns in Table T ⟹ member(C,T)

Select all columns from a table ⟹

$$\forall T,V: same\_rec(T,V_1,V_2,\ldots,V_n) \quad (3)$$

2) Select all values of one column in a table:

Select one column (C) from a table (T) ⟹

$$\forall C,T,V: member(C,T) \wedge value\_of(V,C) \wedge C=X \wedge T=Y \quad (5)$$

3) Select a specific value from a table:

Select values (V) of one column(C) from a table (T), where (V) = X ⟹ $\forall C,T,V: member(C,T) \wedge value\_of(V,C) \wedge V = X \quad (6)$

4) Select two different values from two different tables:

Select value (V_1) from table (T_1) and value (V_2) from table (T_2) Where value X from T_1= value Y from T_2 ⟹

$$\forall C_1,C_2,T_1,T_2,V_1,V_2,X: member(C_1,T_1), member(C_2,T_2), value\_of(V_1,C_1), value\_of(V_2,C_2), same\_rec(T_1,V_1,V_2), same\_rec(T_2,X,V_1,Y) \quad (7)$$

5) Select three different values from three different tables:

Select value (V_1) from table (T_1) and value (V_2) from table (T_2) and value(V_3) from table (T_3) Where value X from T_1= value Y from T_2 and value Z from T3 = value X from T_1 ⟹

$$\forall C_1,C_2,C_3,C_4,T_1,T_2,T_3,V_1,V_2,V_3,V_4,X: member(C_1,T_1), member(C_2,T_1), value\_of(V_1,C_1), value\_of(V_2,C_2), member(C_3,T_2), value\_of(V_3,C_3), member(C_4, T_3), value\_of(V_4, C_3), same\_rec(T_2,V_1,V_3), same\_rec(T_3,V_2,V_4), same\_rec(T_1,X,V_1,V_2) \quad (8)$$

Note that the pattern in Rule 7 could also be used to implement the selection of multi values from multi tables.

In addition, the proposed work is not limited only to selecting processes but could also be adapted to implement other constructs. For instance, consider the following SQL code:

SELECT COUNT (Item_ID), Seller_ID FROM Sales GROUP BY Seller_ID HAVING COUNT(Item_ID) > 1;

The equivalent code in our proposed model is

∀C1,T1:member(C1,T1), T1=Sales,C1= Sales_Item_ID, value_of(X,C1), member(Sales_Seller_ID, Sales), findall(X, value_of(X, Sales_Item_ID),L), member(1,L), bigerthan(count(Sales_Item_ID),1)).

The results in Section 5, "Query Rules," demonstrate that the results from both SQL queries and knowledge-based queries are the same. This indicates that there are no missing data or loss of information. Due to the limitation of the paper size, we cannot provide more examples. However, the method of constructing a mapping from SQL to our logic notation is clear.

VI. IMPLEMENTATION OF THE PROPOSED RULES

This section discusses the result of implementing the above-discussed rules using the Prolong programming language [23] in order to prove the proposed method's applicability. The notations that are used in this section have been explained in section IV (Mapping Process).

In the implementation, first, the knowledge base in Table IV above was inserted in Prolog as facts and the query rules were added as logic rules, also by using Prolog. The following discussion and Tables V, VI, VII, VIII, and IX describe the Prolog code and results for Rules 3, 4, 5, 6, and 7, respectively.

Table V shows the Prolog implementation of Rule 3, where all columns of the database table named the "Items" table have been selected. In Table V, the predicate "member(C, Items)" is used to define the number of columns in the "Items" table, then the predicate "same_rec(Items,V_1,V_2)" has two values because there are two columns in the "Items" table.

Table VI shows the implementation and result of applying Rule 4, where all the values of the column Item in the "Items" table have been selected.

Table VII shows the code implementation and result of applying Rule 5, where a specific value, namely, "computer", has been selected from the "Items" table. Rule 5 and its implementation proves that in a knowledge-based system a search for a specific item can be done without knowing the database table, which it is not possible to do in when using a relational database system. We name this facility as a free search.

Table VIII shows the code implementation and result of applying Rule 6. The table shows the selection of the Item_ID from the "Sales" table and the selection of the "Item" from the "Items" table, where Sales_Item_ID equals Items.Item_ID.

Table IX shows the code implementation and result of applying Rule 7. The table shows the selection of Seller_ *Item_ID* and Seller_ID from the "Sales" table, the selection of the "Item" from the "Items" table and the selection of the seller name "Seller_Name" from the "Seller" table, where Sales.Item_ID equals Items.Item_ID and Sales.Seller_ID equals seller.Seller_ID.

This implementation demonstrates the applicability of the proposed knowledge-based method.

TABLE V. CODE IMPLEMENTATION AND RESULT OF APPLYING RULE 3

```
?- member(C, Items).
C = Item_ID;
C = Item.

?- same_rec(Items,V1,V2).
V1 = 1,
V2 = computer;
V1 = 2,
V2 = printer;
V1 = 3,
V2 = handphone.
```

TABLE VI. CODE IMPLEMENTATION AND RESULT OF APPLYING RULE 4

```
?- member(C, Items).
C = Item_ID;
C = Item.
?- same_rec(Items,V1,V2).
V1 = 1,
V2 = computer;
V1 = 2,
V2 = printer;
V1 = 3,
V2 = handphone
```

TABLE VII. CODE IMPLEMENTATION AND RESULT OF APPLYING RULE 5

```
?- member(C,T), value_of(V,C), V = computer.
C = Item,
T = Items,
V = computer;
False.
```

TABLE VIII. CODE IMPLEMENTATION AND RESULT OF APPLYING RULE 6

```
?-member(Sales_Item_ID,Sales),member(Item,Items),
value_of(V1,Sales_Item_ID),value_of(V2,Item),same_rec(Items,V1,V2),
same_rec(Sales,_,_,V1,_).
V1 = 1,
V2 = computer;
V1 = 2,
V2 = printer;
false
```

TABLE IX. CODE IMPLEMENTATION AND RESULT OF APPLYING RULE 7

```
?- member(Sales_Item_ID,Sales),
member(Sales_Seller_ID,Sales), value_of(V1,Sales_Item_ID),
value_of(V2,Sales_Seller_ID), member(Item,Items),
value_of(V3,Item), member(Seller_Name, seller), value_of(V4,
Seller_Name), same_rec(Items,V1,V3),
same_rec(seller,V2,V4), same_rec(Sales,_,_,V1,V2).
V1 = V2, V2 = 1,
V3 = computer,
V4 = kevin;
V1 = V2, V2 = 2,
V3 = printer,
V4 = john;
false.
```

VII. DISCUSSION

There is no doubt that the relational database system is a vigorous technique for controlling and managing daily transaction systems. On the other hand, in a system requesting analysis for historical data, other data models such as non-SQL, graphic, and knowledge-based systems, and ontologies could provide more benefits than the relational database [24]. As has been proved, the relational database is a useful structural technique for the OLTP system, where data insertion and data integrity are important issues. In the literature, the proposals for converting a relational database into a knowledge-based system have been aimed at generating useful solutions for the OLAP system. In the OLAP system, data integrity, i.e., constraints keys, is not an issue as the analysis encompasses all tables to provide a complete picture, which then assists decision makers in finding correlated facts. Thus, it is somewhat understandable that these previous works did not pay attention to ensuring the existence and correctness of constraints keys in transferring the content of a relational database to a knowledge-based system. In contrast, in this study, two clear rules were defined and added to the proposed model to ensure the existence and correctness of primary and foreign keys (Rules 1 and 2, respectively). These rules are flexible, i.e., they can be added or removed from the knowledge-based system according to the request. In knowledge engineering domain, knowledge-based systems are defined as set of facts (predicates), and set of user defined rules. The inference rules for reasoning rules are built-in mechanism in the solver tool. In our case, Prolog.

The sales system is a common and standardized system in the business world. Therefore, we have chosen a sales system as an example to explain the proposed idea. We have conducted a running example based on the sales system to illustrate the mapping process and demonstrate the effectiveness and applicability of our proposed system.

The other issue that was addressed in this study is the need to demonstrate that a knowledge-based system proposal has completeness in order to show that the proposed system is applicable. According to [25-27], a knowledge base is said to be complete if no formula can be added to the knowledge base. In other words, a knowledge-based system is considered complete if the provided facts and rules are satisfied for describing a domain. In the following, the completeness of the proposed method is presented.

1) Completeness: The famous technique for modeling a relational database is consisting of entities, attributes and association relationships. At the implementation level, entities are represented by tables, attributes are represented by fields which are known as columns, and relationships are represented by primary and foreign keys. Each table consists of fields, and these fields contain the data, and the primary and foreign keys are considered as types of special fields. In the proposed knowledge-based method, the predicate "member(C,T)" denotes the tables and its associated columns. The predicate "value_of(V,C)" denotes all the data that are stored in a table with its associated columns. The predicate "same_rec(T,V1,V2)" denotes the structure of a table by representing that table and its columns. The predicates "primary_key(C,T)", and "foreign_ key(C,T)" denote the relationships in the relational database. Rules 1 and 2 together with the primary_key() and foreign_key() predicates support data integrity in the proposed knowledge-based method. Hence, the five proposed predicates are quite enough to represent the relational database completely and there is no room to add any new predicate. Hence the proposed method is complete.

2) Correctness: The correctness of relational database system is mainly measured by the accuracy of its reports. In the proposed method, Rules 3, 4, 5, 6, and 7 correctly cover all possible outputs that could be generated by the "select" command in a relational database. The implementation of these rules was presented in Section 6 as a proof of applicability. This proof of correctness is in line with the concept of tautology [28].

3) Free search: The proposed knowledge-based method provides the ability to search for a database value without knowing its table or its field. For instance, suppose we want to look for the item "computer" and do not have any previous knowledge about the table or field to which it belongs. As shown by Table VII above, which provides an example of a free search, this type of search can be achieved with the knowledge-based system developed by using the proposed method. This represents a clear advantage over the relational database where free search is impossible.

4) Performance: An additional issue that should be considered when developing any method that deals with information systems is performance. To measure the performance of the proposed method, experiments were conducted to compare the performance of the knowledge-based system developed using the proposed method with that of a famous DBMS, namely, Microsoft SQL Server 2019. The experiment was conducted using the case study, sales system, presented in Section 3.

First, the sales system as represented in Fig. 1 and Tables I to III was implemented in Microsoft SQL Server 2019 and the reports denoted by Rules 3 to 7 were generated. Each report was generated separately and the execution time was saved, and at the end of the procedure the average execution time was calculated. Then, the whole sales system as represented by the knowledge-based system in Table IV was implemented by triggering Rules 3 to 7. Each of the rules was generated separately and the execution time was saved, and at the end of the process the average execution time was calculated. SWI Prolog software has been used for implementing the experiment in knowledge-based side. For the experiment, we have used random garneted data. We have used 50000, 100000, 200000, and 400000 records in both relational databases, and knowledge-based systems. Fig. 2 below shows the result of the experiment. The dimension of the Y-axis is in milliseconds.

From the table, it is obvious that when the proposed method was applied to a huge number of records its performance was much better than that of the traditional DBMS (MS SQL Server 2019).

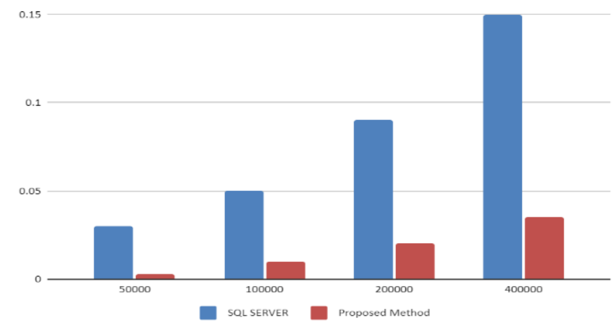

Fig. 2. Performance comparison

VIII. Conclusion

In conclusion, in this study, a proposed method of mapping a relational database to a knowledge-based system was introduced. The benefits of using a knowledge-based system instead of a relational database system in OLAP have already been proved in related works. Hence, the focus of this study was to propose and test a mapping method that would be suitable for use in OLAP systems only. The contribution of the proposed method is threefold: 1) it provides rules to support table constraints, i.e., primary and foreign keys. On the contrary of related works those neglecting table constraints due to insignificance of it in OLAP; 2) it has the ability to perform free searches; and 3) to best of our knowledge, it is the first mapping method for a relational database to a knowledge-based system that has been proved to have completeness, correctness, and good performance.

This proposal is designed to work with OLAP where query speed is not a concern. By providing a performance comparison, we demonstrate that there is no significant difference between the results obtained from our proposal and those from SQL. The main contributions are: flexibility and free search. The proposed knowledge-based method provides the ability to search for a database value without knowing its entity or its field.

In future work, we intend to develop an intelligent software tool to perform a complete mapping from a relational database to a knowledge-based system. We anticipate that the tool will work bidirectionally, i.e., it will be able to map a relational database to a knowledge-based system and vice versa. Moreover, we will consider using other solvers (or maybe SAT) for such mapping. Additionally, we will consider working with a NoSQL database. In addition, we plan to develop a framework that can be applied more broadly across different types of databases and knowledge systems.

REFERENCES

[1] S. Kozielski, and R. Wrembel, "New trends in data warehousing and data analysis," Springer Science & Business Media, pp. 71–91, October 2008. http://dx.doi.org/10.1007/978-0-387-87431-9

[2] P. Gupta, "A Book Review On "Data Warehousing, Data Mining, & OLAP,'' JIMS8M: The Journal of Indian Management & Strategy, vol. 24, pp. 64, 2019.

[3] L. Zemmouchi-Ghomari, A. Djouambi, and C. Chabane, "Proposal for a mutual conversion relational database-ontology approach," International Journal of Modern Education and Computer Science, vol. 11, pp. 13, 2018. http://dx.doi.org/10.5815/ijmecs.2018.07.02.

[4] N. Farooqi, "Tackling Approach for Transferring Database to Knowledge Base via Practical Algorithm," Life Science Journal, vol 16, 2019.

[5] F. Bry, N. Eisinger, T. Eiter, T. Furche, G. Gottlob, C. Ley, B. Linse, R. Pichler and F. Wei, "Foundations of Rule-Based Query Answering,'' Reasoning Web: Third International Summer School, Germany, pp. 1-153, September 2007.

[6] T. Fouad and B. Mohamed, "Model Transformation from Object Relational Database to NoSQL Column Based Database," Proceedings of the 3rd International Conference on Networking, Information Systems \& Security, pp. 1-5, 2020.

[7] C. Krishnamoorthy and S. Rajeev, "Artificial intelligence and expert systems for engineers," CRC press, 2018.

[8] L. Otero-Cerdeira, F. Rodríguez-Martínez and A. Gómez-Rodríguez, "Ontology matching: A literature review," vol. 42, pp. 949--971, 2015.

[9] T. Pankowski, "Using data-to-knowledge exchange for transforming relational databases to knowledge bases," International Workshop on Rules and Rule Markup Languages for the Semantic Web, pp. 256-263, 2012.

[10] M. Morsey, J. Lehmann, S. Auer, C. Stadler and S. Hellmann "Dbpedia and the live extraction of structured data from wikipedia," pp. 157-181, 2012.

[11] M. Bergman and F. Giasson, "KBpedia," Available online: https://kbpedia.org. (accessed on 2 April 2024).

[12] R. Peeters, A. Primpeli, B. Wichtlhuber and C. Bizer, "Using schema. org annotations for training and maintaining product matchers," Proceedings of the 10th International Conference on Web Intelligence, Mining and Semantics, pp. 195-204, 2020.

[13] I. Gorton, J. Klein, and A. Nurgaliev, "Architecture knowledge for evaluating scalable databases," 2015 12th Working IEEE/IFIP Conference on Software Architecture, pp. 95-104, 2015.

[14] T. Ruan, M. Wang, J. Sun, T. Wang, L. Zeng. Y. Yin, and J. Gao, "An automatic approach for constructing a knowledge base of symptoms in Chinese," IEEE International Conference on Bioinformatics and Biomedicine (BIBM), pp. 1657-1662, 2016.

[15] Y. Alfaifi, "Ontology development methodology: A systematic review and case study," 2nd International Conference on Computing and Information Technology (ICCIT) 2022, pp. 446-450, 2022.

[16] M. Dadjoo, and E. Kheirkhah, "An approach for transforming of relational databases to OWL ontology," arXiv preprint arXiv:1502.05844, 2015.

[17] K. D. Mogotlane, and J. V. Fonou-Dombeu, "Automatic conversion of relational databases into ontologies: a comparative analysis of Prot\'eg\'e plug-ins performances," arXiv preprint arXiv:1611.02816, 2016.

[18] K. Sangeeta, and P. Rao, "Onto Extractor: A Tool for Ontology Extraction from ER/EER diagrams," International Journal of Advanced Research in Computer Science, vol. 8, 2017.

[19] S. N. Mathur, "Automatic Generation of Relational to Ontology Mapping Correspondences," Thesis, 2019.

[20] C. Huve, A. Porn, and L. Peres, "Architecture for Mapping Relational Database to OWL Ontology: An Approach to Enrich Ontology Terminology Validated with Mutation Test," ICEIS (2), pp. 320-327, 2019.

[21] A. Elfaki, A. Aljaedi, and Y. Duan, "Mapping ERD to knowledge graph," 2019 IEEE World Congress on Services (SERVICES), vol 2642, pp. 110-114, 2019.

[22] R. Devi, D. Mehrotra, and H. Baazaoui-Zghal, "An r2rml-based approach to map dengue patient database to ontology," 2020 8th International Conference on Reliability, Infocom Technologies and Optimization (Trends and Future Directions)(ICRITO), pp. 790-795, 2020.

[23] W. Clocksin, and C. Mellish, "Programming in PROLOG," Springer Science \& Business Media, 2003.

[24] A. Elfaki, and Y. Alfaifi, "Systematic Approach for Measuring Semantic Relatedness between Ontologies," Electronics, vol. 12, pp. 1394, 2023.

[25] A. Elfaki, S. Phon-Amnuaisuk, and C. Ho, "Modeling variability in software product line using first order logic," 2009 Seventh ACIS International Conference on Software Engineering Research, Management and Applications, pp. 227-233, 2009.

[26] A. Elfaki, "A rule-based approach to detect and prevent inconsistency in the domain-engineering process," Expert Systems, vol. 33, pp. 3-13, 2016.

[27] J. Jaafar, K. Danyaro, and M. Liew, "Web intelligence: A fuzzy knowledge-based framework for the enhancement of querying and accessing web data," Big Data: Concepts, Methodologies, Tools, and Applications, pp. 711-733, 2016.

[28] K. A. Rasol, "Propositional Logic for Knowledge Representation and Formalization of Reasoning," 2017.