Heuristic Based Approach for Automating Multidimensional Schemas Construction

Elhaj Elamin
Computer Science Department
Sudan University of Science and Technology
Khartoum, Sudan
Email: Elhajelamin [AT] hotmail.com

Abdulrahman Altalhi[*], Jamel Feki[**]
Faculty of Computing and Information Technology
[*]King Abdulaziz University, Jeddah
[**]University of Jeddah, Jeddah
Saudi Arabia
[**]*Email: J.feki [AT] uj.edu.sa*

Abstract— **In recent years, the automation process for extracting the elements of the multidimensional schema from operational data sources became the solution to facilitate this complex and time-consuming task. In this paper, we propose a semi-automatic heuristic-based approach for generating star schemas from the transactional relational database of the organization. Our approach encompasses three main phases: i) Database model extraction, ii) Reverse engineering process, and iii) Multidimensional schema generation. The distinctive feature of the reverse engineering phase is it classifies the relational tables of the source database into three types namely i) *Strong entity*, ii) *Weak entity* and iii) *Relationship*. For this proposed approach, we have defined a set of heuristic rules we applied on examples of the literature; the obtained results show that our rules are helpful for the Data warehouse designer in generating star schemas.**

Keywords-Heuristic rules; star schema; relational data source; conceptual design

I. INTRODUCTION

The conceptual design of a Data Warehouse (DW) required the usage of appropriate approach and specific software tools that are completely different from those used with transactional databases. The DW design relies on the Dimensional Model that has its own specific concepts namely Fact, Measures and Dimensions [15]. Furthermore, there is a common agreement that database design approaches are completely inappropriate for designing dimensional schemas [4].

In the DW literature, several contributions have addressed how to design a multidimensional schema for the DW. So far, several methods and techniques have been proposed in order to construct/extract the components/elements (i.e., Fact, Measures and Dimensions) of the multidimensional DW schema. However, there is a diversity between these works. Indeed, some of these works tackle on extracting one element e.g., the fact [6], others draw a complete picture by extracting all the schema elements and hence, building complete multidimensional star (or snowflake) model [7]. Additionally, we noticed that the authors of the existing approaches used different methods to extract the multidimensional elements; these methods differ according to whether this extraction starts from the transactional database schema, or from the user requirements or even from both.

Furthermore, building a multidimensional model from an Entity-Relationship model (ER) is a tedious and time-consuming task [1]. This is due to that this design task requires skilled persons in both operational system design and data warehousing. In order to alleviate this design task, recent works in this evolving research area focus on automation. More precisely, they automate the process of extracting the schema elements from transactional data source schemas and/or even from user requirements [1] [2].

In this paper, we propose a heuristic-based approach for generating star schemas; it is worth mentioning that our long-term objective based on our previous contribution [12], which gains significant advantages as it is hybrid, semi-automatic approach relying on a semantic resource. The term hybrid means that the proposed approach considers both the transactional database data model and user requirements. The use of a semantic resource aims to overcome the heterogeneity issues between concepts. The remainder of this paper is organized as follows: Section II introduces the definitions of the multidimensional concepts; Section III reveals the related work for generating star schemas; in Section IV we describe our proposed approach for constructing star schemas based on heuristic rules; Section V depicts the obtained results and discusses them. Finally, we conclude the paper and introduce future work in Section VI. Before diving into the related work, let us introduce some definitions useful for the remaining of the whole paper.

II. MULTIDIMENSIONAL CONCEPTS

The DW is a multidimensional database; the term multidimensional refers to the multidimensional data model that has specific concepts namely Fact and Dimension for building DW schemas. The Star schema is the keystone in multidimensional modeling; it is composed of one fact having measures, dimensions and hierarchies. Each of these concepts is defined as follows:

A. Facts

A fact represents the business events that have a dynamic property in organization [9]. For example, the Sales activity is a business event that we can model as a fact. A fact helps decision-makers understanding, analyzing and managing their business (increase profit, customers' loyalty…).

B. Measure

A measure belongs to a fact; it is generally a numeric attribute that describes quantitatively a fact [9]. For instance, the *Amount of sale* is a measure of the *Sale* fact. Note that the fact could be seen as an association linking several entities; each entity may play the role of a dimension in the multidimensional model.

C. Dimension

A dimension represents an axis according to which the fact's measures are recorded and then analyzed [11]. The dimension is generally built on a relational table (e.g., Customer) or a set of tables linked through foreign key and primary key attributes. In some cases, a dimension could be built on an attribute as the *SaleDate* attribute [8].

D. Parameter

The attributes of a dimension splits into two classes: Strong attributes and weak attributes. Strong attributes are called parameters and therefore are semantically organized from the lowest to the highest granularity to build hierarchies. A hierarchy of parameters enables decision-makers to aggregate the fact's measures using aggregate functions (Sum, Avg, Min…) in order to compute summarized results highly appreciated for the decisional process.

Figure 1 depicts the general shape of a multidimensional star schema made up of one central fact called Fact (e.g., *Sales*) surrounded by three dimensions. Dimension1 (e.g., *Clients*) has three parameters (e.g., Client_ID, Client_City and Client_Country). A weak attribute is a descriptive attribute associated with one parameter (e.g., Client_Name).

E. Facts vs. Dimensions

Among the complicated tasks, facing DW designers in constructing star schemas is how to identify facts and dimensions correctly in the data source. Fortunately, some characteristics may help to find out each of these concepts; but in many situations, this differentiation is ambiguous because one concept can have the characteristics of both fact and dimension. As an example, a fact table can be characterized by its attributes tend to be numeric, but the dimension table as well can be described by its attributes may appear as numeric [8].

To alleviate this ambiguity in differentiation between these two concepts, we define heuristic rules. Before that, we give more attention of how those concepts are described in a given statement. Therefore, in this context, these characteristics help to draw a line between fact and dimension concepts. As well, we can benefit from these characteristics to elaborate our rules. Table 1 depicts the characteristics of facts and dimensions [8].

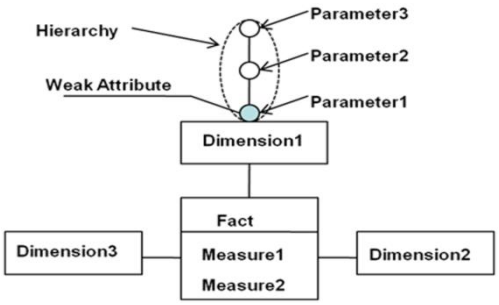

Figure 1. General shape of star schema

TABLE 1. FACT CHARACTERISTICS VS. DIMENSION CHARACTERISTICS

Fact	Dimension
Tends to be measured	Tends to be Analysis-context descriptors
Attributes tend to be numeric	Rarely, the attribute may be numeric
Can be specified at varying levels of detail	Provides context
Represents elements we can aggregate.	Controls the aggregation of measures

III. RELATED WORK

The construction of a multidimensional model; i.e., finding its elements (fact, measures, and dimensions) is a necessary step to build the DW. Several works have been proposed to identify these elements. Nevertheless, from our viewpoint, these works vary in: a) methods used to find out these elements, b) the output schema that represents these elements whether it is Star Schema or Snowflake schema.

For example, authors in [3] proposed the SAMSTAR method, its input is an Entity Relationship Diagram and its output is a set of star schemas. The heuristics used in this method rely on the observation that there is a many-to-one relationship between a fact and a dimension and therefore on a *many-to-one* relationship between their entities/tables. The classification of tables into two categories as potential facts and potential dimensions bases on the following: tables lying on the many side of the many-to-one relationship are candidate for facts, whereas tables lying on the one side of the association are candidate for dimensions. However, although authors of this method have developed an algorithm to define the components of star schemas, we underline that the initial observation for this classification is inaccurate. For example, many tables lying on the many side may be candidate for dimension role as well. Therefore, this classification may generate a wide range of candidate facts.

In order to alleviate the complexity of the DW design process, a Structured Entity Relational Model (SERM) had been proposed in [4] where the authors derive the initial DW model from a conceptual Entity relationship (ER) model. The SERM describes in three stages how to transform the ER model into structured form to obey the needs of multidimensional modeling. However, their work is manual and hence needs a high-level expertise person in the application domain of the DW.

An automated approach to derive facts had been suggested in [6] where the authors defined a set of heuristics to identify facts from a relational data source. However, some of these heuristics produce weak results. Indeed, the authors of these heuristics state that if a relational table has a high ratio of numerical attributes then it may be candidate for a fact. Furthermore, the authors did not consider the constraints related to the relational data source schema; if used, these constraints might help improving the results. Consequently, this heuristic may extract dimension as well because a table that has a large ratio of numerical attributes may be an Entity too. Consequently, this heuristic did not help to achieve the property of disjoint-classification of tables in a given relational database schema. We look to classify a table accurately as fact or dimension exclusively.

In [5] the authors build star schemas from both XML and relational database after combining these two data sources. In fact, very few works perform a reverse engineering process to classify the database tables into tables describing relationships and tables modeling entities. In fact, rules applied in this classification are variable. In our opinion, it is very suitable to improve these rules. This motivated us to suggest rules those consider specifics related to the relational database schema. Particularly, we pay attention to table classified as Weak entity. Obviously, not all schemas have tables of this type, but if the schema has some, our rules will help to produce results that are more accurate.

So far, we can conclude that the research trend in DW area goes to automate this design process. However, the automation depends on designing algorithm [1] [3] [13] or setting heuristic rules [6] [7] to identify the multidimensional elements. Some observations can be made here; first, the rules used so far are not completely defined, in the sense, we may find a weak rule that did not reflect the nature of the given relational tables to correctly generate the right elements or achieve the property of classifying every table in the schema as fact or dimension. As an example, the rules in [5] classify the Room table as fact and dimension simultaneously. Second, few works [3] [7] use ontology as a means to overcome the problems due to heterogeneity of the domains of concepts.

Table 7 (in Appendix) recapitulates the characteristics of the various approaches studied in the related work.

Relying on the conclusions of the related work section we propose, in the next section, our approach for generating star schemas from relational database source. This approach encompasses distinctive features, as it is semi automatic, hence, the DW designer can intervene to guarantee the correctness of the generated elements. Additionally, our work is stringent by the reverse engineering phase. Hereafter, we detail our proposed approach.

IV. OVERVIEW OF THE PROPOSED APPROACH

Let us remember that our long-term objective is a hybrid and semi-automatic approach for DW construction as shown in Figure 2 where the encircled portion represents the steps for generating star schemas hereafter detailed.

In this paper, our objective is to generate star schemas from relational data sources. In order to come up with this objective, we propose a novel approach (Figure 3) with some specifics for each of its three phases, namely: i) Database model extraction, ii) Reverse engineering process, and iii) Multidimensional schema generation.

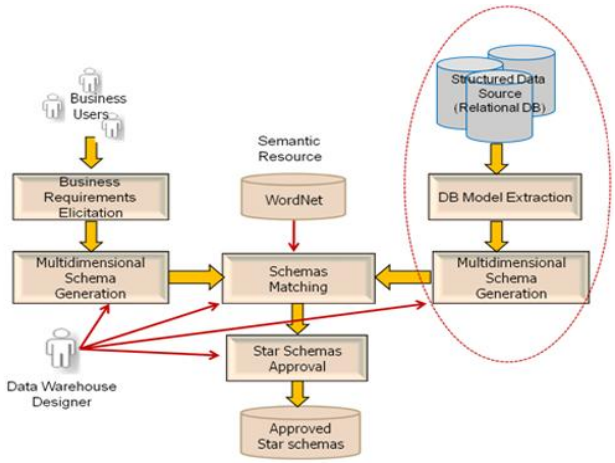

Figure 2. Hybrid, Semi-automatic Approach for the Design of Multidimensional Schemas

Hereafter, we give an overview of the three phases of our proposed approach:

A. *Database schema extraction*

We extract the tables' structures of the relational database (DB) source from the repository of the Relational Database Management System (RDBMS) by querying system views. For each table, we get its name, the name and type of each of its columns, and, in addition, the primary key and foreign key constraints, as they are vital for the next phases. In fact, these constraints have twofold objective: firstly, they help us classify the transactional database tables into three classes, namely Entity-table, Relationship-table and Weak entity-table. Secondly, they will be very useful to trace the links between tables in order to construct dimensional hierarchies.

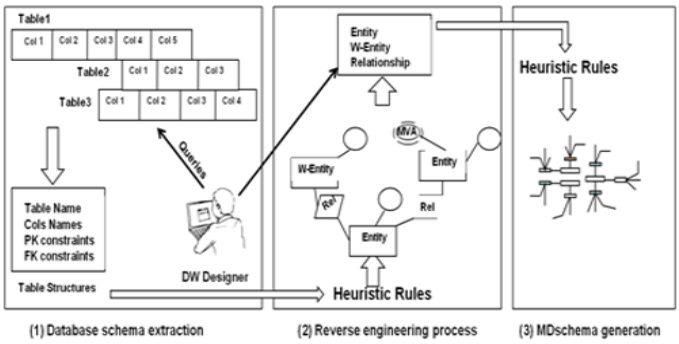

Figure 3. Heuristic Based Approach for Automating Star Schemas Construction

B. *Reverse engineering on the relational database schema*

The aim behind the reverse engineering process is to return the DB table in the relational schema to its initial state. In our

approach, we divide the initial state into three categories: strong entity, weak entity, and relationship. The reverse engineering process enables us to know which DB tables were initially entities of the real world, and which ones were relationships.

Indeed, to identify facts and dimensions, we should know which tables in the schema are suitable to be candidate for facts, and which ones could play the role of dimensions. In data warehousing literature, entities are typically used to design dimensions whereas relationships are for building facts [10] [14]. For this phase, we have defined a set of heuristic rules for identification of entities, relationships and weak entities. The involvement of the DW designer in the reverse engineering phase is important to approve the correctness of tables' classification issued from the process.

C. Multidimensional schema generation

Once the tables extracted from the source relational DB are classified, this multidimensional (MD) schema generation phase aims to building MD schemas. Defining a set of appropriate heuristics is required for the automation of this phase. These heuristics are to find out automatically the MD schemas' elements (facts, dimensions, measures, etc.).

In the next that follows, we detail our approach and we start with the heuristic rules.

Heuristics rules for the reverse engineering process:

Let us point out that in data warehousing literature, approaches starting from ER diagram build facts from relationships whereas dimensions are mainly built from entities [9]. In our framework, we pay this task a great attention, as it is a basic step to identify facts and dimensions. In order to identify entities and relationships within a relational data source, the literature works define rules for the reverse engineering process. Nevertheless, these works classify the schema tables into two categories only.

Identification of strong entities:

R1. Every table in the schema having a single-attribute primary key (PK) is a candidate Entity.

Figure 4 shows example of strong entity table.

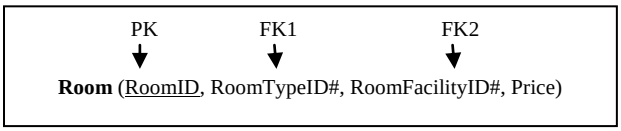

Figure 4. Example of relational table modelling a strong entity

Identification of relationships:

R2. Every table in the schema satisfying the two conditions below is candidate for a relationship:

- A primary key composed of foreign-key attributes, and
- the number of foreign keys within the primary key >1

In Figure 5, the relational table Buy that its primary key is composed of two foreign keys referring two entities (Customer and Concert) satisfies rule *R2*, consequently it classifies as a relationship.

Identification of weak entities:

A Weak entity is an entity type that does not have key attributes of its own [16]. The weak entity has an attribute that

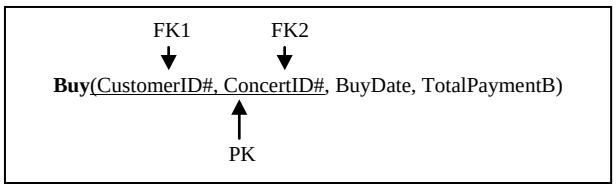

Figure 5. Example of table that models a relationship

partially identified the Entity; this attribute called the partial key. As an example, the *EntityName* attribute can represent a partial key in a table because it does not distinguish all instances of the table. The rule for identifying a weak entity is as follows:

R3. Every table in the schema satisfying the three conditions below is a candidate for a weak entity table:

- Its primary key composed of more than one attribute, one of them is a partial key, and
- the number of its primary key attributes is greater than the number of foreign key attributes in the PK, and
- only one foreign key attribute.

Figure 6 shows the *Dependent* table identified as a weak entity. Note that, in this table, we can find the same *DependentName* twice, so the key of this table partially identifies the instances of the table.

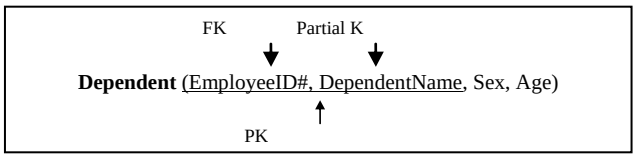

Figure 6. Example of relational table modelling a Weak entity.

We have applied these rules for the Booking schema (Figure 15) issued from [5]. The initial results show that our rules classified Room as an entity whereas Room was classified as entity and relationship at the same time in [5]. The similarity between the results of our approach and the results of the approach in [5] on the current example is that the tables classifies into two categories: Entity and relationship. The justification of this similarity is that the booking schema in [5] does not have tables describing weak entities.

In order to apply all our rules and demonstrate their accuracy, we have added the *Customer_Fellow* table to the DB schema given in Figure 15. This new table (described in figure 7) models a weak entity. After the involvement of the new table, our rules classify the schema tables into three categories: strong entity, relationship and weak entity. These results show the importance of the concept Weak-entity. Thanks to this, we

were able to identify precisely the relationship table, which is a potential fact (using rule *R2*). This precision in classification was not possible to obtain in other approaches; in fact, as an example, if we based our rules on *m:1* relationship, we can classify a weak entity table as a relationship.

Table 2 shows the data source tables classified into tables describing entity tables, relationship tables and weak entity tables.

TABLE 2. CLASSIFIED TABLES OF FIGURE 15 AND FIGURE 7

Relational tables	Class	Classification Rule
Room	Entity	R1
Payments	Entity	R1
RoomBands	Entity	R1
RoomFacilities	Entity	R1
PaymentMethods	Entity	R1
RoomTypes	Entity	R1
Customer	Entity	R1
County	Entity	R1
State	Entity	R1
City	Entity	R1
Singer	Entity	R1
Concert	Entity	R1
Buy	Relationship	R2
Bookings	Relationship	R2
Customer_Fellow	Weak Entity	R3

Additionally, we applied our rules on other schemas (e.g., Company, University) and we obtain rational results (see Appendix, figures 12, 13, 14, and 16); as these schemas include weak entity [16].

Identification of facts:

Facts are built from relationship tables [5] [10] identified in the previous step. In addition, some Entity-tables may be suitable to create facts. Mainly this occurs when it has non-key numeric attribute not included in the primary key.

We define the following two rules for fact extraction:

R4. Any relationship-Table issued from the reverse engineering process (by rule R2) is a candidate to be a fact.

R5. Any Entity-Table respecting the following conditions is a candidate to be a fact:

- has a single attribute primary key, and
- has more than one foreign key, and
- has non-key numeric attributes (so that the number of all numeric attributes in the table is >3), and
- does not have its primary key attribute as foreign key in one of the defined relationship (rule *R2*).

The justification of the number 3 in the constraint "the number of all numeric attributes in the table is >3" is as follows: The Entity-Table should have one attribute as primary key; at least two attributes as foreign keys; and one non-key attribute.

So far, we can extract the candidate facts using rules *R4 and R5*. A fact represents the start point for generating the remaining elements of the star schema (measures and dimensions). Therefore, we gave more attention to coin the rules for extracting facts; as the other star schema elements depend on the generated facts. The following rules reveal the process of identifying measures and dimensions.

Identification of Measures:

Generally, measures are numeric attributes of the tables representing facts. We extract measures using rule R6.

R6. For a given fact-table *T* (i.e., table elected as a fact), the measures are the set of numeric attributes issued from *T* Minus the set of attributes representing primary key or foreign key of *T*.

Table 3 shows measures extracted from the fact tables of our running example.

Identification of Dimensions

In data warehousing, the Dimension concept represents the axis of analysis [8]. Rationally, dimension can be any table that its primary key attribute participates as a foreign key in a fact-table. We propose the following rules to extract dimensions.

R7. Each table referred by a foreign key in a fact-table *F* extracted by rules *R4 & R5* will be a candidate dimension for *F*.

In data warehousing, decision-makers analyze the evolution of their business data through time; consequently, any data recorded in the DW must be related to the Date/Time dimension. For example, a sale is realized at a given date. As the Date/Time exists in the transactional system as an attribute (e.g., sale date), we define rule R9 for building the Date dimension.

R8. A Date or Time attribute in a table *F* transformed into a fact will be a candidate dimension for *F*.

Table 4 shows, for each fact of our running example the set of its extracted dimensions as well as the rule used to extract each dimension.

Parameters and hierarchies

In multidimensional modeling, the attributes of a dimension are organized semantically into hierarchies; these hierarchy attributes are called parameters [19].

TABLE 3. EXTRACTED MEASURES FOR EACH FACT

Fact Name	Measure
Bookings	TotalPaymentDueAmount
Buy	TotalPaymentB
Payments	PaymentAmount

TABLE 4. FACTS AND THEIR IDENTIFIED DIMENSIONS

Fact	Dimensions	Extraction rule
Bookings	Customer	R8
	Room	R8
	DateBookingMade	R9
	TimeBookingMade	R9
	BookedStsrtDate	R9
	BookedEndDate	R9
	TotalPaymentDueDate	R9
Buy	Customer	R8
	Concert	R8
	BuyDate	R9
Payments	Customer	R8
	PaymentMethods	R8

Within a dimension, some attributes could not be considered as parameters; they simply describe parameters and then called weak attributes. As well, the temporal attributes (Date and Time) in our result are organized from the lowest granularity (Day) to the highest granularity (Year).

Table 5 shows the dimensions and their identified parameters. Each hierarchy is built in levels as follows:

Step one: level one in the hierarchy represents the identifier of the dimension [5] [8].

Step two: for each dimension, level two represents the primary key attribute of a table in the schema referenced in the dimension table as foreign key.

Step three: level three extracted recursively by applying step two to other tables in the schema.

Figure 8 depicts the *Customer* dimension of our running example.

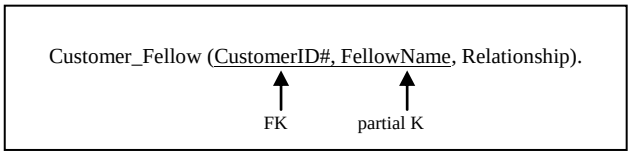

Figure 7. Our proposed additional table to the schema in [5].

TABLE 5. DIMENSIONS AND THEIR PARAMETERS

Dimension	Parameters
Customer	City
	State
	County
Concert	City
	State
	County
	Concert-Date
Room	Price
PaymentMethod	PM-Name
Date	Day
	Month
	Year

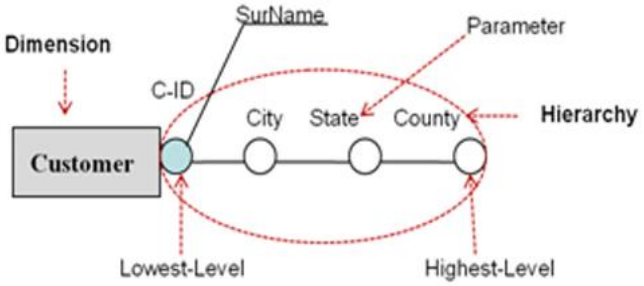

Figure 8. *Customer* dimension with its parameters organized in a hierarchy

V. RESULTS AND DISCUSSION

Our results in this paper are twofold. First, our system automatically classifies the schema tables into tables describing three categories: i) Strong Entity, ii) relationship, and iii)) Weak Entity. To the best of our knowledge, our work is the first to classify the schema tables into tables describing three categories. Hence, it is helpful issue to consider this category (tables modeling Weak Entity) while building the DW schema. Despite the importance of the Weak entity class, most of the previous works in the literature ignore this challenge. Figure 11 shows this classification for the schema in figure 7. On the other hand, the DW designer can change one table from Entity to relationship when necessary.

Second, based on our previous rules applied on schema in figure 15 enriched with the additional table in figure 7, we obtained the multidimensional model depicted in figure 9. These results are more accurate than those described in [5]. Indeed, this improvement is due to the better classification obtained with our rules that classify the Room table as a dimension; this is a rational result for two reasons, firstly, Room is not an event according to the definition of the fact concept [9]. Secondly, the Room table has its primary key as foreign key in another fact table: Bookings, so it is likely to be dimension rather than a fact. Although, some authors argue that the table may appear as both fact and dimension [8], but this will cause an ambiguous issue. We solve this ambiguity by detailed description for the characteristics of the Room table as explained above. Figure 11 shows the generated star schemas. First, our system automatically produces the facts. Second, the DW designer can select one fact and press the measure bottom, the system automatically produces the measure/s for that fact; third, the DW designer press the dimension bottom, the system in turn, produce/s all the dimensions for the generated fact; in this third step as well the identifier for each dimension is generated automatically. Additionally, the lowest screenshot shows that the DW designer can select a different fact in order to generate its measures, dimensions and the dimensions' identifiers.

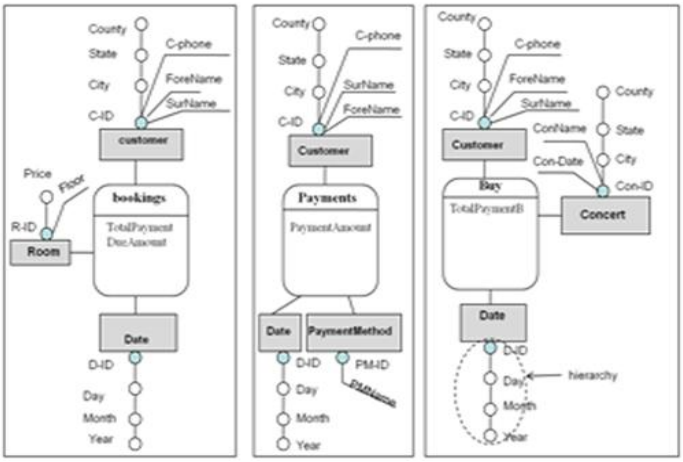

Figure 9. The resulting star schemas

Table 6 shows our results compared to other approaches

TABLE 6. OUR RESULTS COMPARED TO OTHER APPROACHES'RESULTS

The comparable property	**Our results**	**Other approaches' results**
The elements of the reverse engineering process	i. Strong entity. ii. Relationship. iii. Weak entity.	i. Entity. ii. Relationship.
Definition of the heuristic rules	Based on the nature of Entity Relationship Diagram (ERD)	Based on M:1 relationship
Facts definition	Define facts from real relationship table	May define facts from relationship table or weak entity table
Dimensions definition	Define dimension from strong entity table.	May define dimension from weak entity table
Disjointness between fact tables and dimension tables	Achieve the disjointness	Not achieve the disjointness.
The automation of generated star schemas.	Automatic	Works are manual.

The design process of a DW is a complex task. The reasons behind this complexity are varying due to the various approaches designers can follow to complete this process. One of these approaches is generating star schemas from the data source. In the following lines we want to reveal our experiment for dealing with such complexity. Our aim is to help the DW designer taking away some complexity that accompanies this research area.

We follow the method of defining heuristic rules for generating the star schemas. Indeed, this will facilitate the automation process for building DW. Of course, we are not alone in this trend, many authors' walks through this way. However, the rules defined so far have some drawbacks (incomplete, weak, inaccurate or do not reflect the data source nature). To fill this gap, we have defined complete, accurate rules that reflect the nature of the data source. Our rules are twofold: the first class rules defined for the reverse engineering process. For our knowledge, our work is the first one to classify the schema tables into tables describing four classes: strong entity, relationship, multi-valued attribute and weak entity. The second class rules defined for extracting the star schema elements (fact, measures and dimensions). Through our work, we faced this problem:

In one hand, **Room** is a fact table:
- Has more than one foreign key.
- Has non key numeric attribute *(price*).

On the other hand, Room is a dimension table:
- Its primary key attribute *(RoomID*) appear as foreign key in a fact table (Bookings).

How can we solve this ambiguity?

This question is very hard for the DW designer, but for all stakeholders. It is important to solve this ambiguity; otherwise, the resulting DW will be unclear.

Our defined rules classify *Room* table as dimension, and this agrees with the description of both fact and dimension characteristics (table 1). As well, *Room* word is not an event or process to be classified as fact [8].

VI. CONCLUSION AND FUTURE WORK

In this paper, we have defined heuristics rules for generating star schemas from operational data source (relational database). Our method encompasses three steps: firstly, we have extracted the database model; the primary key and the foreign key constraints are extracted because of their important role in the second step. Secondly, based on the primary key constraints and the foreign key constraints extracted in the previous step, we define heuristics rules to categories the input schema tables into tables describing three types namely: Strong entity, Relationship and Weak entity. This classification helps correctly identifying the origin of each table in the relational schema; this will improve the reverse engineering process. Therefore, the DW designer can guarantee that the rules for generating the star schemas produce convincing results. Our method gives attention to the nature of data source model and its concepts to correctly construct the star schemas. The third step reveals the extraction of star schema elements (facts, measures and dimensions). Our defined rules bring rationale results that differ from the work proposed by [5]. One aspect of this difference is the classification of the Room table as a dimension in our results, whereas authors in [5] classify the Room table as dimension and fact, which represent an ambiguous issue for the DW designer. In order to test our proposals, we have implemented a software prototype based on Java language and oracle 10g. As well, the NetBeans 8.0.2 is an IDE for Java that is a simple and flexible tool for designing interfaces. In the future work, we expect to extract star schemas from business requirements; this will give us the opportunity to make a matching process between schemas generated from the data source and those schemas built on the user requirements. The matching process requires the use of ontology to solve semantic issues. We expect the use of WordNet [17] as a semantic resource to solve

the semantic heterogeneity between the schemas, and then to draw the approval schema and complete our conceptual model for the data warehouse.

REFERENCES

[1] Ph. Cassandra, and K. C. Davis, "Automating data warehouse conceptual schema design and evaluation," Data mining and data warehousing. Vol. 2. 2002.

[2] J. Feki, A. Nabli, H. Ben-Abdallah and F. Gargouri, "An automatic data warehouse conceptual design approach," Encyclopedia of Data Warehousing and Mining. 2008.

[3] I. Y. Song, R. Khare, and B. Dai, "SAMSTAR: a semi-automated lexical method for generating star schemas from an entity-relationship diagram,"in Proc of the ACM 10th international workshop on Data warehousing and OLAP. ACM, 2007.

[4] M. Boehnlein, and A. Ulbrich-vom, "Deriving initial datawarehouse structures from the conceptual data models of the underlying operational information systems," in *Proc of the 2nd ACM international workshop on Data warehousing and OLAP*. ACM, 1999.York: Academic, 1963, pp. 271–350.

[5] Y. Hachaichi, J. Feki, and H. Ben-Abdallah, "Designing data marts fromXML and relational data sources," DataWarehousing Design and Advanced EngineeringApplications: Methods for Complex Construction, Part of the Advances in DataWarehousing and Mining (ADWM) Book Series (IGI Global Publication, 2009) (2009): 55-80.

[6] A. Carmè, J. N. Mazón, and S. Rizzi, "A model-driven heuristic approach for detecting multidimensional facts in relational data sources," Data Warehousing and Knowledge Discovery. Springer Berlin Heidelberg, 2010. 13-24.Y.

[7] A. Nabli, J. Feki, and F. Gargouri, "An ontology-based method for Normalization of multidimensional terminology,". Advanced Internet Based Systems & Applications. Springer Berlin Heidelberg, 2009. 235-246.

[8] C. Adamson. *Star Schema The Complete Reference*. US: McGraw-Hill Osborne Media, 2010.

[9] M . Golfarelli and S. Rizzi. *Data Warehouse design: Modern principles and methodologies*. McGraw-Hill, Inc., 2009.

[10] M . Golfarelli, D. Maio and S. Rizzi, "Conceptual design of data warehouses from E/R schemes,". System Sciences, 1998. In *Proc of the Thirty-First Hawaii International Conference on*. Vol. 7. IEEE, 1998.

[11] M. R. Jensen, T. Holmgren and Torben Bach Pedersen, " Discovering multidimensional structure in relational data,". International Conference on Data Warehousing and Knowledge Discovery. Springer Berlin Heidelberg, 2004.

[12] E. Elamin and J. Feki, " Toward an Ontology based Approach for Data Warehousing: state of the art and proposal,". In proc of the international arab conference on information technology (ACIT), pages 777-888, 2014.

[13] R. A. Abdalaziz and T. M. Ahmed, "Generating data warehouse schema,". International Journal in Foundations of Computer Science & Technology, Vol.4, No.1, 2014.

[14] L. Cabibbo and R. Torlone, "A logical approach to multidimensional databases," Advances in Database TechnologyEDBT'98. Springer Berlin Heidelberg, 1998. 183-197.

[15] R. Kimball and M. Ross. *The data warehouse toolkit: the complete guide to dimensional modeling*. John Wiley & Sons, 2011.

[16] R. Elmasri. *Fundamentals of database systems*. Pearson Education India, 2008.

[17] M . Thenmozhi and K. Vivekanandan, " An Ontology based Hybrid Approach to Derive Multidimensional Schema for Data warehouse,". International Journal of Computer Applications, 54 .2012.

[18] H. Choura and J. Feki, " MDA compliant approach for data mart schemas generation," Model and Data Engineering. Springer Berlin Heidelberg, 262-269, 2011.

[19] C. S Jensen., T. B. Pedersen, and C. Thomsen. Multidimensional databases and data warehousing. Synthesis Lectures on Data Management 2.1 .2010.

[20]

APPENDIX

Figure 10. The automatic result for classified tables into three categories

Figure 11. Generation of facts, measures, dimensions and their identifiers

Employee3 (Fname, Minit, Lname, SSN, BDate, Address, Sex, Salary, SuperSSN#, DNO#)
Department3 (Dname, Dnumber, MGRSSN#, MGRstartDate)
Dept-Location (Dnumber#, Dlocation#)
Project3 (Pname, Pnumber, Plocation, Dnum#)
Works-On (Essn#, Pno#, Hours)
Dependent3 (Essn#, DependentName, Sex, BDate, Relationship)

Figure 12. A sample extract of database schema [16].

Figure 13. Classification of company's schema tables

Figure 14. Classification of university's schema tables

TABLE 7. CHARACTERISTICS OF DW DESIGN APPROACHES

Approach	Characteristics of the Approach					Drawbacks of the Approach
	Input	Output	Used Technique	Reverse Engineering	Automation	
Song , et al., 2007 [3]	Entity Relationship Diagram (ERD)	Set of star schemas	Heuristic rules	No	Semi-automatic	1- Heuristics based on *m:1* relationships. 2- Neglects the nature of ERD and concentrate on quantitative analysis
Boehnlein, et al., 1999 [4]	Conceptual Entity Relationship Diagram	Multidimensional Data Structures	Visualization of existence dependencies	No	Manual	The extraction of multidimensional elements is manual
Hachaichi, et al., 2009 [5]	XML and Relational Data Source	Data Mart Schema	Heuristic rules	Yes	Semi-automatic	Classify tables onto two classes: Entity and relationship only
Carmè, et al., 2010 [6]	Relational Data Source	Facts	Heuristic rules	No	Semi-automatic	Some heuristics are weak

Room (RoomID, RoomTypeID#, RoomFacilityID#, RoomBandsID#, Price, Floor, AdditionalNotes)
RoomTypes (RoomTypeID, TypeDesc)
RoomFacilities (RoomFacilityID, FacilityDesc)
RoomBands (RoomBandsID, BandDesc)
Payments (PaymentID, CustmerID#, PaymentMethodID#, PaymentAmount, paymentComments)
PaymentMethods (PaymentMethodID, PaymentMethod)
County (CountyID, CountyName)
Bookings (CustomerID#, DateBookingMade, TimeBookingMade, RoomID#, BookedStartDate, BookedEndDate, ,
TotalPaymentDueDate,TotalPaymentDueAmount, BookingComments)
Customer (CustomerID, CustomerForeNames, CustomerSurNames, CustomerDOB, CustomerHomePhone, CustomerWorkPhone, CustomerMobilePhone,
CustomerEmail, CityID#)
State (StateID, StateName, CountyID#)
City (CityID, CityName, StateID#)
Singer (SingerID, SingerForeNames, SingerSurNames)
Concert (ConcertID, ConcertName, CityID#, SingerID#)
Buy (CustomerID#, ConcertID#, BuyDate, ConcertDate, TotalPaymentB)

Figure 15. The Schema of hotel's booking database [5].

STUDENT (ID_STD, First_Name, Last_Name, City, Birth_Date, Gender, Housing, Tel, Nationality, ID_FAC#)
FACULTY (ID_FAC, Extended_Name, Short_Name, City, ID_Univ#)
COURSE (ID_CRS, Name, Curr_ID#, Semester_No)
CURRICULUM (ID_Curr, Designation, Study_Years)
COURSE-RESULT (ID_STD#, ID_CRS#, Year, Grade-Oral,Grade-Crs-Sess1, Grade-Crs-Sess2, Final-Grade-Crs)
ANNUAL-RESULT (ID_STD#, Univ_Year, Term, Final-Grade-Sess1, Final-Grade-Sess2, Final- Grade, Result)
UNIVERSITY (ID_Univ, Extended_Name, Short_Name)

Figure 16. The schema of university database [18]