Using Relational Algebra on the Specification of Real World ETL Processes

Vasco Santos
CIICESI - School of Management and Technology
Polytechnic of Porto
Felgueiras, Portugal
vsantos@estgf.ipp.pt

Orlando Belo
ALGORITMI R&D Centre
University of Minho
Braga, Portugal
omb@di.uminho.pt

***Abstract* — Modeling Extract-Transform-Load (ETL) processes of a Data Warehousing System has always been a challenge. The heterogeneity of the sources, the quality of the data obtained and the conciliation process are some of the issues that must be addressed in the design phase of this critical component. Commercial ETL tools often provide proprietary diagrammatic components and modeling languages that are not standard, thus not providing the ideal separation between a modeling platform and an execution platform. This separation in conjunction with the use of standard notations and languages is critical in a system that tends to evolve through time and which cannot be undermined by a normally expensive tool that becomes an unsatisfactory component. In this paper we demonstrate the application of Relational Algebra as a modeling language of an ETL system as an effort to standardize operations and provide a basis for uncommon ETL execution platforms.**

Keywords - component; Data Warehousing Systems; ETL Processes; ETL Conceptual and Logical Modeling; Relational Algebra.

I. INTRODUCTION

The addition of a *Data Warehousing System* (DWS) to the *Enterprise Information Systems* (EIS) is a natural step in the evolution of an IT infrastructure, since they are specifically oriented to data analysis, providing an integrated repository of historical data - a data warehouse - instead of the common transactional systems that are oriented for transaction processing and business activity support. A DWS usually deals with large volumes of information that come from different sources, process and store them to feed a single data repository hence enabling the development of faster and more effective analysis processes.

A DWS is intended primarily for users with decision-making capacity - the decision makers - about the future of the company, which with the help of specialized tools can analyze patterns and variations in the data, thereby extracting the information needed to respond in timely basis to the various issues and develop the basis for the daily support of its decision-making processes. Since these systems store data from various sources of information, you can also correlate the various types of data that so far, only with transactional systems, were a little difficult to correlate in the EIS. However, investing in this type of system requires large financial resources since there are components of the system that are expensive. This cost is sometimes a factor that inhibits the adoption of these systems by small and medium-sized enterprises, because they have less money available to consider such prohibitive investment. On the other hand, large enterprises are also faced with problems affecting the development of these systems, which are time and the volume of data to be processed [1]. As the volume of information to be processed periodically increases, an ETL task consumes more and more resources. Since there may be limits on the time frame reserved for this task, the solution is to increase the computing power so that you can run the task in the time required. The increased computing power has been achieved usually by acquiring computers with increased performance, which are even more expensive. However, this solution may not be feasible for all enterprises. One possible solution to overcome such problems is to take advantage of the computing power of the computers on the local networks of enterprises, which in the vast majority of the time are available and untapped. If used, this computational power will not increase significantly the cost of the system and might in turn even reduce it since these computers were already acquired by the enterprise. The use of these computers with parallelism might also contribute to reduce the time required to execute the ETL. However, commercial ETL tools are not prepared to use local network computers to execute ETL tasks. In fact even their modeling capabilities are becoming a handicap due to the use of proprietary notation. The use of proprietary notation in the modeling phase of an ETL system is discouraged because with time it will act as a reason to block changes in the ETL system. ETL commercial tools are expensive and their adoption should be highly weighted, especially in their ability to accept and import diagrams made in standard languages. It is our belief that design and execution should be different platforms in a choice for more flexibility in the development and maintenance of an ETL system.

With this in mind, and through the use of a standard database language, *Relational Algebra* (RA), we have already proposed several specifications for common ETL tasks in an effort to standardize the steps that need to be executed in a generalized environment like a grid environment. The application of such proposals in a real case ETL scenario is the main subject of this paper. Therefore, we organized this paper as follows: in section 2, we will study the most common approaches for modeling ETL processes; next, in section 3 we present a RA approach as an alternative approach to the modeling phase of an ETL system; in section 4 we apply the RA patterns to a real ETL case scenario. Finally, in the last section, we present some conclusions and final remarks.

II. Related Work

During the last few years, several proposals have been made to standardize the design of ETL processes [2-4], some of them very similar to the ones found in the current design of databases – first the conceptual model and only after the logical model and in the end the physical model. However, most of these approaches do not specify the steps needed to model the most common activities present in an ETL scenario like *Change Data Capture* (CDC), *Slowly Changing Dimensions* (SCD), *Surrogate Key Pipelining* (SKP), *Data Quality Enforcement* (DQE) and *Data Conciliation and Integration* (DCI) that maintain a DWS up-to-date. It is our belief that a formal approach for these steps is needed and the use of RA could be the basis for a formal and correct representation of ETL processes and be used in a generalized DWS structure. As such, we already presented some proposals using RA to model the SCD scenario[5], the DQE tasks [6], and the DCI tasks [7] using RA operators.

III. A Relational Algebra Approach

An ETL system is comprised of several tasks that gather data from source systems, transforms it (through cleaning, conforming and conciliating operations), and finally loading it into a data warehouse [8]. As already stated previously, the modeling phase of such a system is crucial to the success of its development, and therefore subject of intensive research by the data warehousing community.

A. RA modelling of CDC tasks

Capturing source data changes is critical for the correct maintenance of a DW. The challenge of this task varies according to the number and variety of data sources, which can be much diversified making the primary challenge the access to its data. Data needed for the DWS might be stored in Mainframes, Flat Files, XML Files, DBMS, Web Logs, etc. and the access to each of the sources might not be easy. Nevertheless, and for the purpose of modeling, let us assume that data is extracted from source systems and correctly represented in a relation's format. So, lets assume that an ETL process extracts all the data existent in a source system to load into a specific dimension, since it might not be possible to extract only new and changed data. We shall call the relation that stores the data extracted *src_dimdata*$_{S'x}$. This relation consists of a list of attributes that comprise a business key and a finite set of additional attributes (1). The business key is normally identified as a primary key and may be a single or set of attributes.

$$src\_dimdata_{S'_x} = \langle BK_{S'_x}, Att_1, \dots, Att_p \rangle \tag{1}$$

in which BK$_{S'x}$ is the business key of source S'$_x$ and 1≤x≤ n and n≥1, and Att$_1$,…,Att$_p$ are additional data attributes needed for the dimension and p≥1. Let us also assume that the previous extraction operation has been stored to facilitate the detection of new or changed tuples (2).

$$prev\_src\_dimdata_{S'_x} = \langle BK_{S'_x}, Att_1, \dots, Att_p \rangle \tag{2}$$

Then, to determine the new and changed tuples a subtraction is the only operation needed (3).

$$src\_dimdata_{S_x(BK_{S_x}, Att_1, \dots, Att_p)} \leftarrow src\_dimdata_{S'_x} - prev\_src\_dimdata_{S'_x} \tag{3}$$

In order to prepare the next extraction operation, the previous one is now discarded and the current extraction will be the next operation to be made (Eq. 4).

$$prev\_src\_dimdata_{S'_x} \leftarrow src\_dimdata_{S'_x} \tag{4}$$

From this point on a series of transformations, mainly due to quality enforcement rules, are applied to the data extracted before integrating it into the DW.

B. RA modelling of DQE, DCI and SCD tasks

Source systems of a DWS that tend to be OLTP systems supporting regular business activities, normally contain inaccurate data, unknown or null data and sometimes inconsistent data that are spread generally by operational systems' objects, constraints or rules. This imposes the inclusion of cleansing tasks in data warehouses' populating systems in order to detect and filter (and sometimes recover) all those data anomalies in the source's data before loading it into a DW. Usually, these tasks are implemented and executed through the use of proprietary tools that analyze and transform the data accordingly to predefined business requirements. So, cleansing tasks are undoubtedly important. They are mainly concerned with identifying problems in metadata and data [9-11], i.e., inconsistencies at attribute and row level, rather than defining a strategy to execute the procedures needed to deal with those problems. These problems are mainly dealt with DQE tasks which were modeled in RA in [6].

Another set of tasks normally present in an ETL process is the use of several sources, as such, all potential sources of data must be studied and analyzed for current or future integration in its DW. This multitude of sources increases the rate of failure if not dealt with proper care and attention. In today's organization is common to find different transactional systems that evolved through different periods in the organization' life, probably even in different departments or services units. Nevertheless, it is also likely to find several representations of common concepts in these systems. For instance, if one system deals in some way with clients and another system deals with customers, several misrepresentations of data might occur, like different codes, names, addresses or other information for a same entity. The correct representation of this information in a DW is often the most difficult tasks that one may find in an ETL system that aims to maintain the data warehouse up-to-date and correct.

The problem of integrating data from heterogeneous sources is mainly categorized in two aspects: schema and semantics heterogeneity. Schema integration has been widely studied by the database research community [12] and culminated in the presentation of several prototypes and algorithms [13]. Semantics integration is also a challenge that has been studied by several researchers and analyses problems not only at schema-level but also at instance-level [10, 14]. Nevertheless, these approaches do not formalize any set of logical steps needed to maintain and integrate data coming from heterogeneous sources inside a DW. Data, once extracted from a source S_x, has to be integrated, therefore transformed, in order to represent a unified and common view in the DW. The problem arises when these heterogeneous sources contain just partial information when in comparison to the data stored in the DW, and also when we have the need of storing historical information about changes that occur in the sources. This transformation phase is then comprised of several processes, mainly dedicated to the cleaning, conciliation and integration with or without dealing with changes and were modeled using RA operators in [7].

C. Modelling a SKP Process with Relational Algebra

After dealing with dimension data, the process of loading facts into the DW becomes simplified. The only task we need to perform now is the correct translation of the business keys to the correspondent surrogate keys. This process, also known as *Surrogate Key Pipeline* (SKP), lookups up, in sequence, on the auxiliary conciliation tables for the correct surrogate key.

When processing facts from different sources, each tuple lookups for the dimension's correspondent surrogate key (given their business key), and proceeds in the same way for all the remaining dimensions. The tuple is ready to be loaded into the fact table when all processes conclude with success. Error occurrences or unmatched tuples must be signaled and dealt with in the recuperation phase that is normally a human interaction phase. The SKP can also be modeled in RA, through a series of operations that prepare the fact table to be loaded into the DW and separate the unmatched tuples for further revision. Consider the following definition a common structure of a fact table from a source before the SKP (5).

$$F_{S_1} = \langle BK_{dim_1}, \dots, BK_{\mathrm{dim}_z}, m_1, \dots, m_p \rangle \tag{5}$$

where BK_{dimx} is the business key and $1 \leqslant x \leqslant z$ of the z dimensions present, and $m_1,\dots,m_p$ are the measurements present in the fact table, and the structure of the fact table after the SKP (6)

$$F'_{S_1} = \langle SK_{\mathrm{dim}_1}, \dots, SK_{\mathrm{dim}_z}, m_1, \dots, m_p \rangle \tag{6}$$

where SK_{dimx} is the correct surrogate key of each dimension and $1 \leqslant x \leqslant z$, and $m_1,\dots,m_p$ are the measurements present in the fact table.

The steps needed to map the business keys to the surrogate keys for each dimension are presented as a relational algebra tree in Fig. 1, where a conciliation table (7) is used to facilitate the mapping operation. All the following surrogate key substitutions follow the same approach with the necessary adjustments to the relational algebra expressions.

$$conc\_dimtable_1 = \langle SK_{\mathrm{dim}_1}, BK_{S_1}, \dots, BK_{S_n} \rangle \tag{7}$$

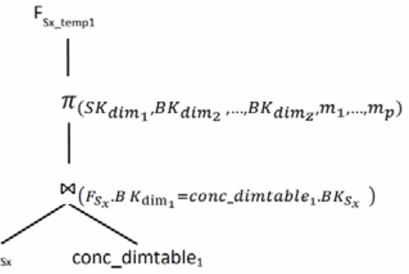
F_{Sx_temp1}
$\pi_{(SK_{dim_1}, BK_{dim_2}, \dots, BK_{dim_z}, m_1, \dots, m_p)}$
$\bowtie_{(F_{S_x}.BK_{\mathrm{dim}_1} = conc\_dimtable_1.BK_{S_x})}$
F_{Sx}
$conc\_dimtable_1$

Figure 1. Surrogate Key mapping for a dimension

The business keys from the fact table that for some reason don't find the correct correspondence in the conciliation table must be signaled for human revision therefore stored appropriately. The RA tree modeling this behavior is presented in Fig. 2.

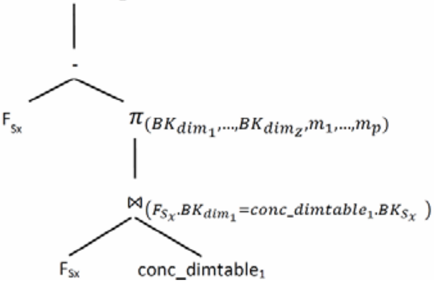
$unmatched_{Sx\_dim1}$
-
F_{Sx}
$\pi_{(BK_{dim_1}, \dots, BK_{dim_z}, m_1, \dots, m_p)}$
$\bowtie_{(F_{S_x}.BK_{dim_1} = conc\_dimtable_1.BK_{S_x})}$
F_{Sx}
$conc\_dimtable_1$

Figure 2. Unmatched Surrogate Key mappings

IV. A Real World Case Scenario

A typical ETL scenario integrates one or more data sources, the data providers, a staging area, where data, once extracted from sources, is cleaned, transformed and conformed accordingly to business necessities, and a DW (or a data mart) to receive the data prepared to support the views of decision-makers. Next, we will present and discuss a real world case scenario to apply our RA pattern proposals. The business selected is an adaption of current social network activities where users, once registered, can elaborate posts, comments and in addition tag those messages with hashtags.

A hashtag is a word or an unspaced phrase prefixed with the hash character (#). Their use in common posts acts as some kind of metadata over the contents of the post, and have become very popular in Social Network sites. These tags have also become very useful when searching for similar tagged messages since search engines return messages based in those tags. In addition, Social Networks have used these tags to check for trending subjects in any temporal time window.

A. *The Data Sources*

The case study proposed includes two different Social Networks and the need to build a Data Mart with the ability to store the use of hashtags in both sources. This Data Mart will later serve as a source for different temporal analysis to detect or search for trends in the use of hashtags. The logical design of the first source of this case study is presented in Fig. 3. The modus operandi of this social network is the following:

- Each Customer is able to own one or more Logins in the Social Network;
- Each profile (login) can elaborate posts and tag them with hashtags;
- The posts elaborated can either be primary or comments of other posts;
- The customer can also create events that classify posts (-1/+1);
- Whenever a post receives an event a notification is generated to the owner of the post;

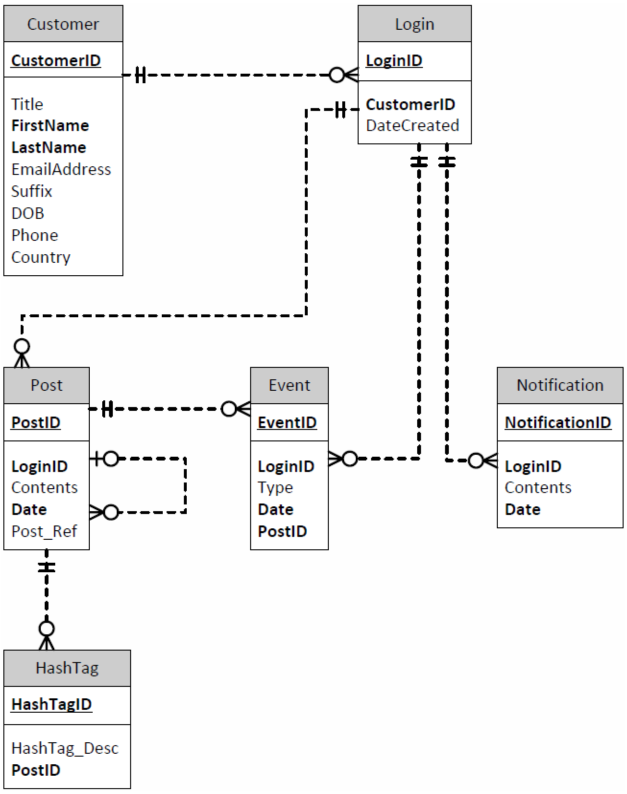

Figure 3. OLTP Logical Model of Social Network A

The logical design of the second source of this case study is presented in Fig. 4. The modus operandi of this social network is the following:

- Each Client is able to post one or more Messages;
- Each Message has an identified type and might contain attachments and hashtags;
- Clients can comment existent Messages and associate hashtags to those comments;
- A client can reshare an existent Message.

B. *The Data Mart*

The dimensional model of the Data Mart for this case study is presented in Fig. 5 (star schema). The grain of the fact table is the use of hashtags in comments or posts in the Social Networks that act as sources. The measurement used in the fact table is the number of occurrences a certain hashtag has in a day.

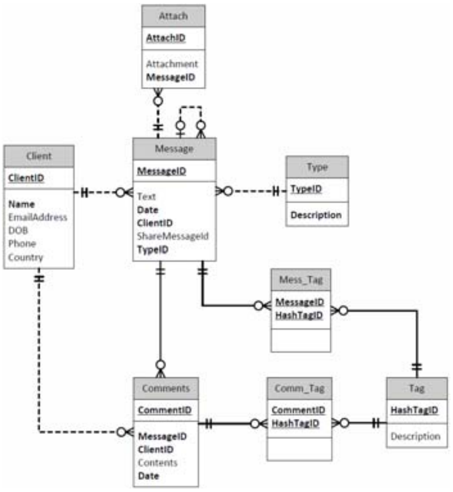

Figure 4. OLTP Logical Model of Social Network B

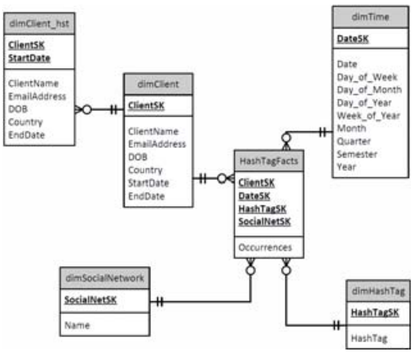

Figure 5. Data Mart Dimensional Model of the Case Study

C. ETL Modeling with Relational Algebra

In this section, an instantiation of the Relational Algebra patterns is presented to model the populating process of the star schema of Fig. 5. The dimension *dimTime* is fully populated in advance and therefore will not be part of this modeling scenario. The dimension *dimSocialNetwork* is populated with data that categorizes the sources, in this case *Social Network A* and *Social Network B*, but is fully prepared for additional Social Networks (sources). The remaining dimensions and the fact table need to be populated with data being extracted from the Social Networks OLTP Systems.

Since we are dealing with two different sources, the primary challenge will be the conciliation of data extracted. Several conciliation tables will be needed mainly to standardize common concepts and to map business keys into the surrogate keys of the Data Mart. In a high level BPMN diagram we present in Fig. 6 the workflow process to populate the Data Mart. One pool was used, with three lanes, corresponding to the population of each dimension and fact table. Each collapsed process is further expanded for better understanding of the tasks involved. Analyzing the collapsed process referent to Populating *dimHashTag* with data from *Social Network A* (Fig. 7), the basic steps are the CDC, DQE and DCI tasks, as would normally be in the populating process of a typical dimension.

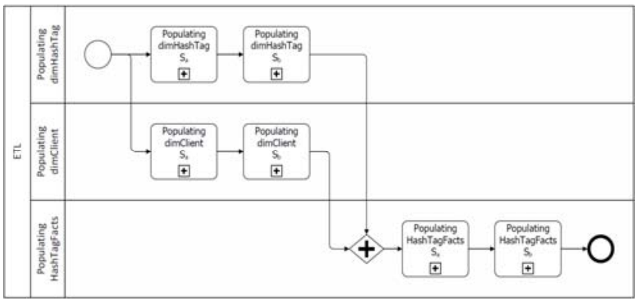
ETL
Populating dimHashTag
Populating dimClient
Populating HashTagFacts
Populating dimHashTag S$_a$
Populating dimHashTag S$_b$
Populating dimClient S$_a$
Populating dimClient S$_b$
Populating HashTagFacts S$_a$
Populating HashTagFacts S$_b$

Figure 6. High level BPMN diagram of the ETL workflow process

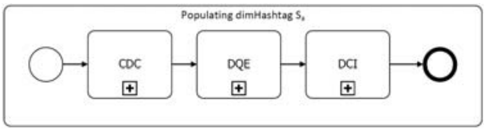

Figure 7. Expanded process for Populating *dimHashTag* with data from *Social Network A*

Digging further into each collapsed process, it is now visible the sequence of operations defined in previous sections of this paper, i.e., the relational algebra expressions and trees used to process data from OLTP systems into the Data Mart (Fig. 8, Fig. 9 and Fig. 10).

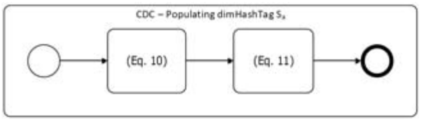

Figure 8. Expanded process for the CDC used for populating *dimHashTag* with data from *Social Network A*

Assuming that the extraction process gathers all data from table *HashTag* into table represented in (8) and through the use of the patterns proposed in (1) through (4), new and changed data is stored in *src_dimhashtag*$_{Sa}$ for future transformations.

$$src\_dimhashtag_{S'_a} = \langle HashTagID, HashTag\_Desc, PostID \rangle \tag{8}$$

$$prev\_src\_dimhashtag_{S'_a} = \langle HashTagID, HashTag\_Desc, PostID \rangle \tag{9}$$

$$src\_dimhashtag_{S_a(HashTagID, HashTag\_Desc, PostID)} \leftarrow src\_dimhashtag_{S'_a} - prev\_src\_dimhashtag_{S'_a} \tag{10}$$

$$prev\_src\_dimhashtag_{S'_a} \leftarrow src\_dimhashtag_{S'_a} \tag{11}$$

Analyzing the DQE Collapsed process (Fig. 9) it is possible to see the flow of RA operations to conform HashTag data. The table used for conforming HashTag values is represented in (12), where *HashTag_Desc* is the attribute value coming from the source and *HashTag_Desc1* is the conformed value of the attribute.

$$HashTag\_norm = \langle HashTag\_Desc, HashTag\_Desc1 \rangle \tag{12}$$

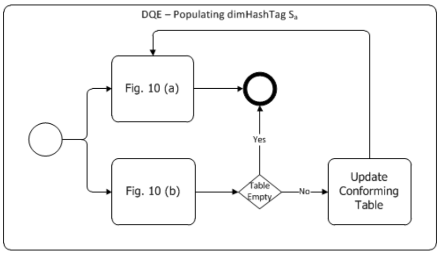

Figure 9. Expanded process for the DQE used for populating *dimHashTag* with data from *Social Network A*

The operations conform the hashtag values into table *src_dimhashtag_norm*$_{Sa}$, and unmatched tuples into a quarantine table *src_dimhashtag_exp*$_{Sa}$ for expert supervision (Fig. 10).

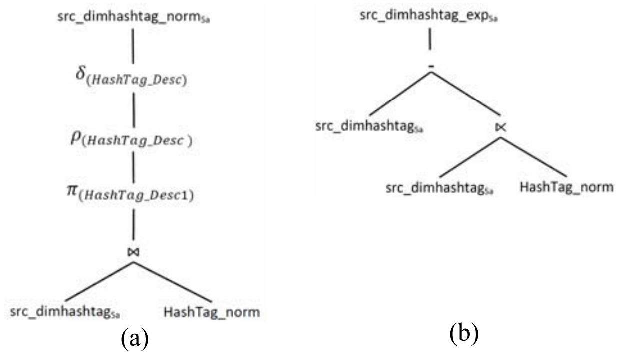

Figure 10. (a) Conforming HashTag Description (b) unmatched tuples

After conforming the hashtag values for Social Network A, the dimension *dimHashTag* from the DataMart can now be populated following the conciliation process as described in [7]. Nevertheless, due to the fact that history preservation is not relevant for this dimension only new values are significant for this process.

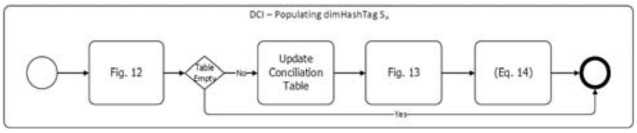

Figure 11. Expanded process for the DCI used for populating *dimHashTag* with data from *Social Network A*

The matching process will separate existing value from new values. Existing values are discarded and new values must be dealt with by expert supervision, mainly by updating the conciliation table (13) and generating the correct Surrogate Key for those new values.

$$conc\_hashtag = \langle HashTagSK, HashTag\_Desc, HashTagID \rangle \tag{13}$$

The integration of HashTag data from Social Network A into dimension *dimHashTag* is achieved through Eq. 14.

$$dimHashTag \leftarrow dimHashTag \cup conc\_dimhashtag_{S_a} \tag{14}$$

The other collapsed processes referent to the populating procedure of the dimensions are very similar to the one presented. In fact, only a few particularities exist as it can be shown in the full diagram in Appendix. Loading a fact table is nevertheless different, data has to be extracted and transformed from OLTP sources and submitted to the SKP process before loading it into the Data Mart.

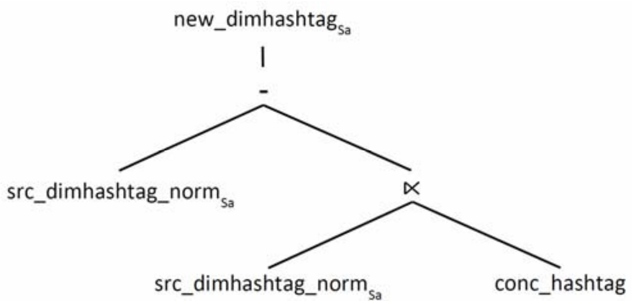

Figure 12. New hashtag data from Social Network A

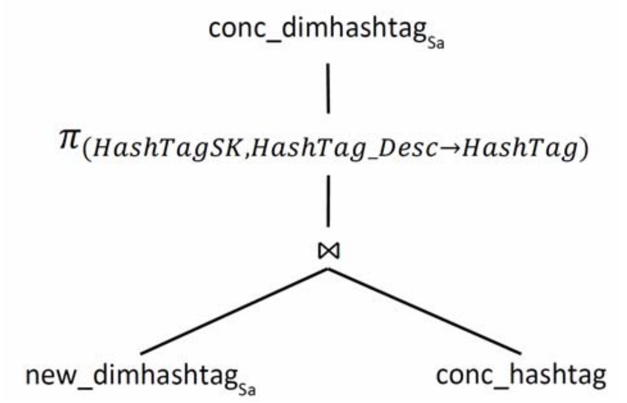

Figure 13. Surrogate Key replacement for hashtag data

V. CONCLUSIONS

In this paper we demonstrate the application of the RA language as a modeling language of an ETL system in an effort to standardize operations and provide a basis for uncommon ETL execution platforms like grid environments. We exposed RA as an effective alternative for modeling ETL systems, using patterns to model a real ETL application scenario. An ETL process is one of the most critical processes of a DWS, having considerable implementation difficulties that put in risk the success of the entire DWS. ETL modeling has gained great importance during the last years, giving excellent means to represent, discuss and analyze conceptually ETL tasks and control mechanisms, providing a “first view” of the entire system – a very useful instrument. So, it is important that we have available a standard notation to use at the ETL conceptual capable to represent the specificities of the process in a very precise way. RA satisfies all these requirements and specificities of the most common ETL systems process implementation. It is formal way to represent all the most elementary data operations that usually appear in any regular ETL system, providing an excellent procedural representation for the description and validations of data based operations. Additionally, the RA’s tree-based representation of tasks using trees facilitates the understanding of how the flow of operations (and data structures involved with) is organized, giving a clear picture for validation and optimization of the entire ETL process.

REFERENCES

[1] M. V. Mannino and Z. Walter, "A framework for data warehouse refresh policies," *Decision Support Systems,* vol. 42, pp. 121-143, 2006.

[2] P. Vassiliadis, A. Simitsis, and S. Skiadopoulos, "Conceptual modeling for ETL processes," in *Proceedings of the 5th ACM international workshop on Data Warehousing and OLAP*, McLean, Virginia, USA, 2002, pp. 14-21.

[3] P. Vassiliadis, A. Simitsis, and S. Skiadopoulos, "On the Logical Modeling of ETL Processes," in *Proceedings of the 14th International Conference on Advanced Information Systems Engineering*, 2002, pp. 782-786.

[4] J. Trujillo and S. Luján-Mora, "A UML Based Approach for Modeling ETL Processes in Data Warehouses," *Conceptual Modeling - ER 2003,* vol. 2813, pp. 307-320, 2003.

[5] V. Santos and O. Belo, "Slowly Changing Dimensions Specification a Relational Algebra Approach," *International Journal on Information Technology,* vol. 1, pp. 63-68, 2011-12-01 2011.

[6] V. Santos and O. Belo, "Modeling ETL Data Quality Enforcement Tasks Using Relational Algebra Operators," *Procedia Technology,* vol. 9, pp. 442-450, 2013.

[7] V. Santos and O. Belo, "Modelling ETL Conciliation Tasks Using Relational Algebra Operators," in *UKSim-AMSS 8th European Modelling Symposium*, Pisa, Italy, 2014, pp. 275-280.

[8] R. Kimball and J. Caserta, *The Data Warehouse ETL Toolkit - Pratical Techniques for Extracting, Cleaning, Conforming, and Delivering Data*. : Wiley Publishing, Inc., 2004.

[9] M. A. Hernández and S. J. Stolfo, "Real-world Data is Dirty: Data Cleansing and The Merge/Purge Problem," *Data Mining and Knowledge Discovery,* vol. 2, pp. 9-37, 1998.

[10] E. Rahm and H. H. Do, "Data Cleaning: Problems and Current Approaches," *IEEE Data Engineering Bulletin,* vol. 23, pp. 3-13, 2000.

[11] M. L. Lee, H. Lu, T. W. Ling, and Y. T. Ko, "Cleansing Data for Mining and Warehousing," in *10th International Conference on Database and Expert Systems Applications*, Florence, 1999, pp. 751-760.

[12] M. Lenzerini, "Data integration: a theoretical perspective," in *Proceedings of the twenty-first ACM SIGMOD-SIGACT-SIGART symposium on Principles of database systems*, Madison, Wisconsin, 2002, pp. 233-246.

[13] E. Rahm and P. A. Bernstein, "A survey of approaches to automatic schema matching," *The VLDB Journal,* vol. 10, pp. 334-350, 2001.

[14] Z. Kedad and E. Métais, "Dealing with Semantic Heterogeneity During Data Integration," in *Conceptual Modeling — ER ’99*. vol. 1728, J. Akoka, M. Bouzeghoub, I. Comyn-Wattiau, and E. Métais, Eds., ed: Springer Berlin / Heidelberg, 1999, pp. 325-339.