Modelling ETL Conciliation Tasks Using Relational Algebra Operators

Vasco Santos
CHICESI - 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
obelo@di.uminho.pt

Abstract—The design and development of a data warehousing system (DWS) tends to be an exceptional resource consuming project which in turn makes it a high risk/reward project. In order to minimize the risk, some design methodologies and tools are used along the several phases of the project. The Extract-Transform-Load (ETL) component is normally one of the most critical components of a DWS since it gathers, corrects and conforms data in order to be loaded into the Data Warehouse (DW). Data conciliation task tends to be a dull and manual intensive job that often deals with several heterogeneous sources which is critical to the correct representation of the enterprise’s information. The manual nature of this task makes it prone to errors and subject of intensive and successive monitoring. In this paper, we analyse some of the most common ETL tasks for data conciliation using a Relational Algebra approach, as an effort to standardize them for future use in a generic ETL environment. A slowly changed dimension scenario will be used to support the data conciliation modelling process designed for this work.

Keywords-component; Data Warehousing Systems, ETL Conceptual Modelling, Data Conciliation Tasks, Relational Algebra

I. INTRODUCTION

The development of a Data Warehousing System (DWS), which serves as a conformed, integrated and centralized repository[1] that stores enterprise information, is a challenging task that, if successfully, can give to an enterprise an edge over their current competitors. It can also be used as source of information for data mining tasks and therefore provide information that supports management decisions that can better adjust the enterprise’s future in these uncertain times. In order to take full advantage of a DWS, all potential sources of data must be studied and analyzed for current or future integration in its Data Warehouse (DW). This multitude of sources increases the rate of failure if not dealt with proper care and attention.

In today’s enterprises is common to find different transactional systems that evolved in different periods in the enterprises’ 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 Extract, Transform and Load (ETL) system that aims to maintain the DW up-to-date and correct. As we know, an ETL system is responsible for processing operations that gather and fetch data from disparate source systems, transforms it (clean, adapt, conciliate, etc.) in order to posteriorly load it into a DW [2].

During the last few years, several proposals have been made to design and modelling ETL processes [3-5], some of them very similar to the ones that we can find in the design of current database systems – first, designing the conceptual model and only after the logical model. However, these approaches do not specify the common steps needed to maintain conciliated data in a DWS. It is our belief that a formal approach for these steps is needed and the use of Relational Algebra could be the basis for a formal and correct representation of ETL processes and be used in a generalized DWS structure.

This paper is organized as follows: in section 2, we will study common approaches for cleaning and integrating heterogeneous source data; next, in section 3, we propose a formal specification of ETL conciliation processes through the use of Relational Algebra. Finally, in the last section, we present some conclusions and describe some lines for future work.

II. RELATED WORK

The evolution of the Information Technology field contributed to a proliferation of database applications in enterprises environments. Even now, an increasing amount of data and information surface everyday imposing on enterprises new ways to gather valuable knowledge to support more effectively their everyday business decisions and activities. Nevertheless, the correct storage of data gathered on heterogeneous sources, both inside and outside the enterprise, has been a challenge [6].

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 [7] and culminated in the presentation of several prototypes and algorithms [8]. 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 [9, 10]. Nevertheless these approaches do not formalize any
set of logical steps needed to maintain and integrate data coming from heterogeneous sources inside a DW.

In our point of view, Relational Algebra is a quite adequate formalism and language to specify such conceptual and logical steps, having the advantage of not being platform or software dependent, as happens with SQL. Based on that, in a previous work we already presented a formalized approach of one of the most common ETL tasks, Slowly Changing Dimensions (SCD) [11], using Relational Algebra has support.

III. DATA CONCILIATION MODELLING

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 (Fig. 1).

The ETL process is comprised of several tasks, mainly defined as a workflow process [12-14], that are normally executed in a dedicated environment. However, unusual DWS environments, such as grid environments, start to arise as an approach to deal with ETL performance bottlenecks [15]. It is our belief that the combination of Relational Algebra and grid environments are a valid approach to decrease the costs of a DWS implementation and performance eager ETL processes. In that sense, next, we present a Relational Algebra modelling approach prepared specifically to deal with a case of data conciliation that comes from a group of heterogeneous information sources of a DW.

Once extracted from a source \( S_x \), where \( 1 \leq x \leq n \), data 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 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.

\[
A. \text{Conciliation Phase}
\]

Let’s assume that the DW is based in a dimensional model[16]. Therefore first we deal with dimension data only after we deal with the facts. In both cases, for the conciliation phase, we need a conciliation table, for each dimension, that help us with mapping business keys from the heterogeneous sources into the surrogate keys used in the DW. The definition of a table, or relation, as presented in [17] and described in many subsequent articles and textbooks, is a set of tuples where each element of the tuple has a corresponding specific domain. Thus, we can define a relation \( \text{conc_dimtable} \) as a list of attributes having a surrogate key, and a list of business keys (derived directly from the heterogeneous sources previously referred) – (1).

\[
\text{conc_dimtable} = (SK, BK_{S_1}, ..., BK_{S_n})
\] (1)

in which \( SK \) is the surrogate key, normally a natural number, \( BK_{S_i}, ..., BK_{S_n} \) are the attributes that belong to the business key of a source \( S_i \), where \( 1 \leq x \leq n \). A business key from a specific source might not be a single attribute but rather a set of attributes, which imposes that some additional adjustments must be made to the equation. The business keys will allow us to integrate data from different sources correctly, since they will map an instance, from a specific source, to a surrogate key that will in turn guarantee data integrity in the DW.

The common structure of source data after being extracted and cleaned is presented in (2).

\[
\text{source_dimdata}_{S_x} = (BK_{S_x}, Att_1, ..., Att_p)
\] (2)

where \( BK_{S_x} \) is the business key of source \( S_x \) and \( 1 \leq x \leq n \), and \( Att_1, ..., Att_p \) are additional data attributes needed for the dimension.

A graphical approach to the conciliation process is presented in Fig. 2, where for each source processed a matching task is performed to map the business keys into the DW surrogate keys. Whenever a source tuple is unmatched, it is stored in a separate table for further revision since it will probably be new data that will need to be processed and updated with a matching correspondence in the conciliation auxiliary table (conc_dimtable).

The matching process can be expressed in Relational Algebra and we present it using Relational Algebra trees in Fig. 3. In Fig. 3(a) the join operation between source data and the auxiliary table will result in determining the matching tuples and the correct correspondence to the surrogate key. In Fig. 3(b) the subtract operation is used to remove from source data those tuples that were matched, therefore remaining those that do not have correspondence in the auxiliary table. These tuples must then be dealt with, normally through human interaction that determines the correct correspondence to a surrogate key (new or already defined) in the conciliation auxiliary table.
Figure 2. ETL Conciliation process.

Figure 3. (a) Conciliated Data; (b) Unmatched Data.

The structure of the result table after the conciliation phase is very similar to the source table, the only change is the presence of the surrogate key (SK) instead of the business key (BK) – (3).

\[
\text{conc\_dimdata}_x = \{SK, \text{Att}_1, ..., \text{Att}_p\} \tag{3}
\]

After the conciliation phase, begins the integration phase, where data coming from different sources is integrated into the dimension table. In this phase, dealing with changes [11], normally called changing dimensions, has to be adjusted to the fact that we are dealing with multiple sources.

**B. Integration Phase**

Let’s assume that in a DW the structure of one of its dimensions is comprised of two identical tables: one storing current tuples and the other storing historic tuples [18]. The common structure of such a table is presented in (4).

\[
\text{dimension\_current} = \{SK, A_1, ..., A_m, DateFrom, DateTo\} \tag{4}
\]

in which SK is the surrogate key, normally a natural number, \(A_1, ..., A_m\) are additional data attributes needed in the dimension and \(m \geq 1\). Since we are dealing with multi source data, a specific source might not contain all the information to correctly fill all dimension attributes, nevertheless each source attributes exist in the dimension’s attributes - \(\{\text{Att}_1, ..., \text{Att}_p\} \subseteq \{A_1, ..., A_m\}\).
Assuming that the extraction process only retrieves new and changed tuples from source systems, the methodology to deal with new tuples is somewhat different from the one used to deal with changed tuples, mainly due to the necessity of conforming data. Changed tuples are already integrated into the dimension so the only task needed is to preserve history, i.e., updates that occurred in source systems must be reflected in the DW dimension without losing history, therefore we transfer old data to the historical table and insert changed tuples into the dimension that holds current values. In order to simplify the interpretation of the Relational Algebra Model, we present in Fig. 4 the RA tree with the operations that identify expired data in the dimension.

Fig. 5 presents the appropriate operations to transfer expired data from the dimension, since it is only supposed to store current data, to the historical table in order to preserve history.

After removing the expired data from the dimension, the last step needed is to add the updated values to the dimension without losing correspondence with data that was obtained from other sources (Fig. 6).

Dealing with new data from conc_dimdata is based on the determination of tuples which surrogate keys do not have correspondence in the dimension. Those tuples are added to the dimension after being time stamped, identifying the validity of the information, with the particularity of leaving the dimension attributes that are not known in the current source in a null state – this is achieved by the left outer join (Fig. 7).
C. Preparing a Fact Table

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 pipelining, 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 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. Errors occurrences or not matching tuples must be signaled and dealt with in the recuperation phase that is normally a human interaction phase.

IV. CONCLUSIONS AND FUTURE WORK

An ETL process is prone to errors, critical to the success of a DWS and above all a very time and resource consuming process. The complexity increases exponentially when dealing with multiple and heterogeneous data sources, augmenting the risk of failure or, even worse, the risk of misrepresenting information in a system conceived for supporting management decision-making assessments. In this paper we presented a formal model to deal with a case of data conciliation from different sources using relational algebra has the support language. Relational algebra was used with the intention to support unusual data warehousing environments such as grid environments, where a database management system normally doesn’t exist but it has exceptional processing capabilities that are ideal to support an ETL process. In the short term, we intend to model several other standard ETL tasks and well-known problems, specifying them with relational algebra following the intent to maximize the enterprises resources in a low cost DW environment and provide a formal platform for ETL logical modelling.

REFERENCES