Flexible Provenance Tracing

Liwei Wang, Wuhan University, China

Henning Koehler, The University of Queensland, Australia

Ke Deng, The University of Queensland, Australia

Xiaofang Zhou, The University of Queensland, Australia

Shazia Sadiq, The University of Queensland, Australia

ABSTRACT

The description of the origins of a piece of data and the transformations by which it arrived in a database is termed the data provenance. The importance of data provenance has already been widely recognized in database community. The two major approaches to representing provenance information use annotations and inversion. While annotation is metadata pre-computed to include the derivation history of a data product, the inversion method finds the source data based on the situation that some derivation process can be inverted. Annotations are flexible to represent diverse provenance metadata but the complete provenance data may outsize data itself. Inversion method is concise by using a single inverse query or function but the provenance needs to be computed on-the-fly. This paper proposes a new provenance representation which is a hybrid of annotation and inversion methods in order to achieve combined advantage. This representation is adaptive to the storage constraint and the response time requirement of provenance inversion on-the-fly.

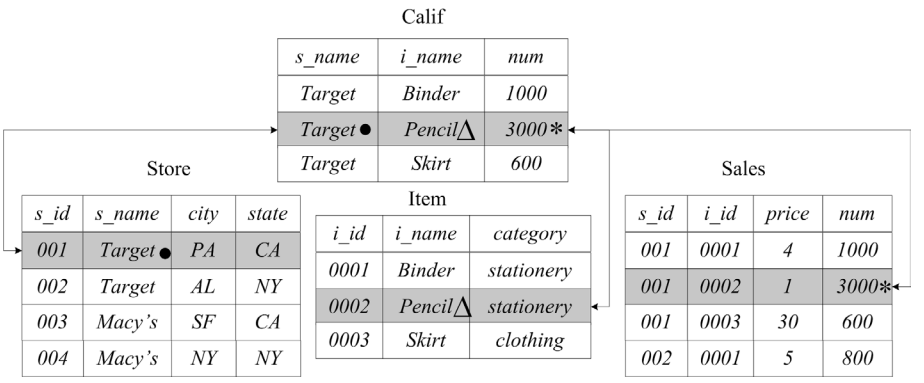
Keywords: *Annotation, Data Provenance, Inversion, Matrix Compression, Provenance Tracing*

INTRODUCTION

With the fast development of World Wide Web, more and more data from the web are used today as essential information sources. Some web-based data service applications such as selling online systems usually collect large numbers of data from multiple sources on web and do some statistical analysis and predictions based on these data. However, as the quantity of data on web explodes rapidly, the quality of data, such as correctness, currency, consistency and completeness, decreases sharply. The judgment on data quality is a very difficult and time-consuming process if we only rely on our own manual judgment. This is because some data on the web often comes from evolution or aggregation results of multiple data sources, and therefore the quality of this data relies on the quality of the sources obtained from the web. Meanwhile, the relationship between the data and its sources is very complex and hard to identify. Thus, we hope to use a kind of data provenance technology to automatically find out from where the data users unexpected were obtained when users see the anomalous and suspicious data. This issue is particularly important in the web-based applications as it

DOI: 10.4018/jssoe.2011040101

Figure 1. An example of annotation



helps users verify the reliability of the data and assess the value of the data.

The origins of a data product and the transformations by which it arrived in a database is termed the data provenance. The importance of data provenance is recognized by its applications for data quality estimation, determining resource usage and provide context to interpret data, etc (Bhagwat, Chiticariu, & Tan, 2004). The solutions of data provenance in the literature usually involve annotations that comprise of the derivation history of a data product and inversion that generates a "reverse" query to find the origins supplied to derive a data product.

Example 1. Consider an example used in the work (Cui, Widom, & Wiener, 2000), which is shown in Figure 1. A person wants to analyze the selling information of California stores, so he defined a materialized view *Calif* in the data warehouse for this purpose. In this example, if we consider the tuple < *target, pencil,* 3000 > in view *Calif*, we want to find its source in *Store*, *Item* and *Sales*.

According to the approach proposed by Bhagwat, Chiticariu, and Tan (2004), annotations usually have be recorded by adding an extra attribute to a relation, and data is moved or transformed, annotations needs to be involved. That means, when view *Calif* is defined, the annotation attached in value "Target" of the tuple < 001, *target, FA, CA* > in *Store* table need to be automatically propagated to value "Target" of the view tuple <*target, pencil,* 3000>, representing the value's derivation history, as shown in Figure 1. However, tracing from a specific result attribute to a particular attribute element in the source data is not feasible using such an approach, the amount of annotations required to be stored in *Calif* would be far too large.

Alternative approach proposed by Cui, Widom, and Wiener (2000) is to automatically derive provenance of data for the materialized view in data warehouse. Figure 2 illustrates the provenance derivation process of example shown in Figure 1. The basic idea is to re-execute the view definition with information from the tuple. Then the source data item contributing to *t* are identified. That is, three base tables *Store*, *Item* and *Sales* are joined to form an intermediate table, then query conditions "*state* = "*CA*" ∧ *s_name* = "*Target*" ∧ *i_name* = "*pencil*" ∧ *num* = 3000" are obtained from the *Calif* definition. The tuple retrieved from the intermediate table is then split to different source tables. Although inversion seems to be more optimal from a storage perspective since an inverse query identifies the provenance for an entire class of data, the information available from the tuple heavily depends on the view definition and thus it is not unusual that the