Chapter 3.1
Data Warehouse Maintenance, Evolution and Versioning

Johann Eder
University of Klagenfurt, Austria

Karl Wiggisser
University of Klagenfurt, Austria

ABSTRACT

Data Warehouses typically are building blocks of decision support systems in companies and public administration. The data contained in a data warehouse is analyzed by means of OnLine Analytical Processing tools, which provide sophisticated features for aggregating and comparing data. Decision support applications depend on the reliability and accuracy of the contained data. Typically, a data warehouse does not only comprise the current snapshot data but also historical data to enable, for instance, analysis over several years. And, as we live in a changing world, one criterion for the reliability and accuracy of the results of such long period queries is their comparability. Whereas data warehouse systems are well prepared for changes in the transactional data, they are, surprisingly, not able to deal with changes in the master data. Nonetheless, such changes do frequently occur. The crucial point for supporting changes is, first of all, being aware of their existence. Second, once you know that a change took place, it is important to know which change (i.e., knowing about differences between versions and relations between the elements of different versions). For data warehouses this means that changes are identified and represented, validity of data and structures are recorded and this knowledge is used for computing correct results for OLAP queries. This chapter is intended to motivate the need for powerful maintenance mechanisms for data warehouse cubes. It presents some basic terms and definitions for the common understanding and introduces the different aspects of data warehouse maintenance. Furthermore, several approaches addressing the problem are presented and classified by their capabilities.

DOI: 10.4018/978-1-60566-756-0.ch010

INTRODUCTION

The standard architecture for data warehouse systems are *multidimensional databases*, where

transactional data (*cell values*) are described in terms of master data (*dimensions* and *dimension members*). Whereas today's commercial systems are well prepared to deal with changes in the transactional data, they are, surprisingly, not able to deal with changing master data in a satisfactory way. Nonetheless, such changes frequently occur (restructuring in organizations, new laws, mergers and acquisitions, product portfolio restructuring, etc.). All these changes have to be represented in the information systems, and thus, must somehow be modeled also in the data warehouse. For data warehouses the adequate representation and treatment of such changes is even more crucial than in standard database applications, since data warehouses are intended to represent also historical data which – changes occurring – might be quite incompatible.

A simple example illustrating the problem of missing data is querying the number of inhabitants in the European Union for the last 25 years. This query seems rather straightforward and the numbers should not leave much space for interpretation. But, one has to be aware of some changes: First of all, the geopolitical entity "European Union" only exists since 1993, succeeding the "European Community", which itself was originally named "European Economic Community". Furthermore, in the considered period (1983 to 2008), the European Union grew from 12 to 27 members. Finally, with the reunification of East- and West-Germany in 1990 one of the member countries had a massive internal reorganization. So if querying the number of inhabitants from 1983 to 2008, how can the resulting numbers be compared? When querying this data from the Eurostat website, one has to choose the "geopolitical entity" (EU-27, EU-25, one or more counties, …) for which the data should be retrieved. If, for instance, EU-25 is chosen, the population for these 25 countries is returned also for the years before their membership. But of course, the overall sum of returned inhabitants for the year 1987 does not match the real number of people living in the European Union at that time. Comparing the numbers of 1990 and 1991, where the organization itself did not change, may indicate a massive increase of inhabitants. In reality, the 1991 number also contains the 16.4 million people of former East-Germany. Eurostat, for instance, takes this into account, and presents numbers of the united Germany also for the years before 1991. Another example for an unclear inclusion are the Baltic countries or Slovenia. They did not even exist before 1991, but were parts of other countries, which, of course, never were parts of the European Union. An alternative to presenting such "adjusted data" is to display the "historical truth", i.e. include the numbers of different countries only after they joined the European Union. This may make sense in some situations, in others, such results may be useless.

An example demonstrating the effect of changing semantics could be to retrieve the Gross National Product of the countries in the European Union from 1983 to 2008. Besides the problems induced by the structural changes described above, i.e. whether and how to include numbers for a specific country, this query illustrates the changing semantics problem: As of 1999 and 2002, a common European currency, the Euro, was introduced as deposit currency and cash money respectively, in many – but not all – of the member countries. Thus, before 1999 the Gross National Product of different countries was expressed in the local currency, but as of 1999 it is given in Euro. Before 1999 for comparing the GNP of different countries, it is obvious that the numbers must be brought to a common base, i.e. the same currency, to be comparable. But what about statistics for a single country? For Austria, 1 Euro exchanges 13.7603 Austrian Schillings. So, someone comparing the Austrian Gross National Product from 1990–2006 without considering the Euro would notice a giant retracement in the year 1999. But, of course, someone who knows about the Euro can divide each value given in ATS by 13.7603 and then compare the values.