Chapter 3
From Conventional to Multiversion Data Warehouse: Practical Issues

Khurram Shahzad
Royal Institute of Technology (KTH)/Stockholm University (SU), Sweden

ABSTRACT

The Data warehouse is not an autonomous data store, because it depends upon its operational source(s) for data population. Due to changes in real-world scenarios, operational sources may evolve, but the conventional data warehouse is not developed to handle the modifications in evolved operational sources. Therefore, instance and schema changes in operational sources cannot be adapted in the conventional data warehouse without loss of information. Multiversion data warehouses are proposed as an alternative to handle these problems of evolution. In this chapter we discuss and illustrate how versioning is implemented and how it can be used in practical data warehouse lifecycle. It is designed as a tutorial for users to collect and understand the concepts behind a versioning solution. Therefore, the purpose of this chapter is to collect and integrate the concepts, issues and solutions of multiversion data warehouses in a tutorial-like approach, to provide a unified source for users that need to understand version functionality and mechanisms.

INTRODUCTION

Online transaction process systems (OLTP) are used to meet day-to-day requirements of an enterprise. But OLTPs' are unable to meet decision support requirements of an enterprise, because a) their schemas are not optimized to support decision-support queries and b) they are not made to support decision making (Paulraj, 2001; Kimball, 2002).

Data is extracted from OLTP(s), transformed and loaded in data warehouse after removing inconsistencies. Therefore, the data warehouse is an integrated and materialized view of data, which is optimized to support decision making (Chaudhuri, 1997). The data warehouse works as a data source for various types of applications e.g. analytical processing, decision making OLAP, data mining tools, dashboards etc.

DOI: 10.4018/978-1-60566-816-1.ch003

Multidimensional models with central fact and surrounding dimension relations are typically used for designing a data warehouse, with two-fold benefits: on one hand they are close to the way of thinking of decision makers analyzing the data, therefore helping those users in understanding the underlying data; on the other hand, they allow designers to predict users' intentions (Rizzi, 2007).

For data population the data warehouse depends upon its operational sources (also called OLTPs). Therefore, changes in operational sources may lead to derivation of inconsistent outputs from data warehouse (Bebel, 2004). These can be divided into two types: 'i) schema changes, i.e. insert/update/delete records, ii) content changes, i.e. add/modify/ drop an attribute or a table' (Wrembel, 2004; Rundensteiner, 2000).

Inconsistent outputs, generated due to changes in operational sources, can be handled in two ways (Wrembel, 2005): 'i) evolution approach, ii) versioning approach'. According to the evolution approach, changes are made to the data warehouse and data is transformed to the changed data warehouse, after which the previous one is removed (Blaschka, 1999). But, shortcomings of the approach are identified by a number of authors [see (Bebel, 2004; Golfarelli, 2004; Golfarelli, 2006, Wrembel, 2005) for details]. Whereas, according to the versioning approach, a new version of the data warehouse is created, changes are made to the new version, data is populated in the new version and both versions are maintained (Ravat, 2006).

Most information on concepts, issues and solutions of multiversion data warehouses are spread across a number of sources in the form of white papers, conference papers, workshop papers and journal papers, and the concepts and solutions underlying versioning cannot be easily understood by a naive user from most current sources. Therefore, the purpose of this chapter is to collect and integrate concepts and solution approaches of multiversion data warehouse, in order to provide a unified source for that target audience.

The rest of the chapter is organized as follows: motivations for creating multiple versions of data warehouse are discussed in section 2; principles of versioning the data warehouse and levels of abstraction in multiversion data warehouse (MVDW) are described in section 3; a framework for version creation is presented in section 4, and a method for modeling multiversion data warehouses is presented in section 5; metadata to be stored for multiversion data warehouses is described in section 6, a method of retrieval from multiversion data warehouses is given in section 7 and in section 8 a case study is presented to discuss practical issues of implementing multiversion data warehouses. Section 9 concludes the chapter.

MOTIVATION AND REQUIREMENTS FOR DATA WAREHOUSE VERSIONS

Operational sources are structured or unstructured data stores that keep record of real-world activities by dynamically storing data about those activities (Chaudhuri, 1997; Gardner, 1998). For example, an operational store can keep record of a 'product purchase process' by storing data about: the person who purchased a product, the product that was purchased, the employee who sold the product and the order placed for purchasing the product. The data warehouse, on the other hand, is not an autonomous data store, because its information is extracted from operational sources, cleaned, transformed and loaded into it. Therefore, for population of the dimensional schema, data warehouses depend upon operational sources, and changes in operational sources may either trigger changes in the data warehouse or derivation of inconsistent results in those data warehouses (Bebel, 2004; Marian, 2001).

It is an established fact that data warehouses have four major properties: subject-oriented, time-variant, non-volatile and integrated (Paulraj, 2001; Kimball, 2002). Real-world events may bring changes to operational sources (Mitrpanont,