Chapter 1
A Two-Tiered Segmentation Approach for Transaction Data Warehousing

Xiufeng Liu
Technical University of Denmark, Denmark

Huan Huo
University of Technology Sydney, Australia

Nadeem Iftikhar
University College of Northern Denmark, Denmark

Per Sieverts Nielsen
Technical University of Denmark, Denmark

ABSTRACT
Data warehousing populates data from different source systems into a central data warehouse (DW) through extraction, transformation, and loading (ETL). Massive transaction data are routinely recorded in a variety of applications such as retail commerce, bank systems, and website management. Transaction data record the timestamp and relevant reference data needed for a particular transaction record. It is a non-trivial task for a standard ETL to process transaction data with dependencies and high velocity. This chapter presents a two-tiered segmentation approach for transaction data warehousing. The approach uses a so-called two-staging ETL method to process detailed records from operational systems, followed by a dimensional data process to populate the data store with a star or snowflake schema. The proposed approach is an all-in-one solution capable of processing fast/slowly changing data and early/late-arriving data. This chapter evaluates the proposed method, and the results have validated the effectiveness of the proposed approach for processing transaction data.

DOI: 10.4018/978-1-5225-5516-2.ch001
A Two-Tiered Segmentation Approach for Transaction Data Warehousing

INTRODUCTION

A data warehouse is the decision-making database which holds the data extracted from transaction systems, operational data stores, or other external source systems. The process of processing the data from source systems into a central data warehouse is traditionally referred as data warehousing (Kimball & Caserta, 2004). The transformed data in a data warehouse (Inmon, 2002) are typically saved into the tables with a star schema and accessed by decision-support systems, such as Online Analytical Processing (OLAP) tools and Business Intelligence (BI) applications (March & Hevner, 2007). Data warehousing systems run ETL jobs at a regular time interval, such as daily, weekly or monthly. Operational data management systems create dynamic data through transactions. Transaction data are increasingly common across a variety of applications, such as telecommunications, bank systems, retail commerce, and website management. Transaction data consist of the records of individuals and events, and can be changed during business operations, e.g., add new orders, update or cancel existing orders. These changes are updated to the data warehouse to support decision-making purposes. The detailed transaction records in a data warehouse can trace the operations of an operational processing system, called System of Record (SOR) (Inmon, 2003). Usually, an ETL processes transaction records according to the arriving order of records, e.g., the timestamps, and the dependencies between records (if they exist). For example, when loading data to the data warehouse with a star schema, the dimension records are usually loaded first, then fact records, due to their foreign-key referencing relationship. If a fact record arrives first, the looking-up of a dimensional key will fail. Another example is to load data into snowflake schema tables where the foreign-key references also exist between normalized tables. The standard approach for loading snowflake schema tables is to load parent tables first (referenced tables), then to load child tables (referencing tables). As this approach loads data according to table dependency, there are some weaknesses: It requires extra space for storing early-arriving data; fact data loading cannot proceed until the dimension data have been loaded; and parallel loading becomes difficult, due to table dependency. Besides, another challenge in data warehousing is how to deal with loading fast-/slowly-changing data. For example, for loading of slowly changing dimension data (SCDs) (Kimball & Caserta, 2004), the traditional approach is first to check history records in the DW, then update the date attributes of the records, finally add a new record. All of the above are the challenges for a transaction data warehousing system.

This chapter proposes a two-tiered segmentation solution for a transaction data warehousing system. The proposed solution first uses a two-staging ETL to process detailed transaction records towards an SOR data warehouse (Tier-1 segmentation), then uses a second ETL process to populate the dimensional data store (DDS), which is called DDS process (Tier-2 segmentation). The two-staging ETL is responsible for populating the data from operational source systems into an SOR data warehouse, while the DDS process is responsible for populating the data from SOR into a multi-dimension data store. The two segmentations have a similar structure in which an additional data store is introduced for the ETL. The purpose of this design is to ease ETL optimizations, for example, implement parallelization and lower the complexity of data transformation. Moreover, this design is a one-stop solution to deal with early/late-arriving data, and fast/slowly-changing data (It will be discussed shortly). This solution is more efficient and less intrusive compared with the standard approach, which is, particularly, favorable for processing transaction data.

In summary, this chapter makes the following contributions: 1) The authors propose a novel 2-tier segmentation approach for a transaction data warehousing system; 2) The authors propose an all-in-