Chapter 9
Incremental Load in a Data Warehousing Environment

Nayem Rahman
Intel Corporation, USA

ABSTRACT

Incremental load is an important factor for successful data warehousing. Lack of standardized incremental refresh methodologies can lead to poor analytical results, which can be unacceptable to an organization’s analytical community. Successful data warehouse implementation depends on consistent metadata as well as incremental data load techniques. If consistent load timestamps are maintained and efficient transformation algorithms are used, it is possible to refresh databases with complete accuracy and with little or no manual checking. This paper proposes an Extract-Transform-Load (ETL) metadata model that archives load observation timestamps and other useful load parameters. The author also recommends algorithms and techniques for incremental refreshes that enable table loading while ensuring data consistency, integrity, and improving load performance. In addition to significantly improving quality in incremental load techniques, these methods will save a substantial amount of data warehouse systems resources.

1. INTRODUCTION

The data in the warehouse is subject-oriented, integrated, identified by a timestamp, and non-volatile (Inmon, 2002). The primary goal of data warehousing is to provide access to information that has high value for decision making (Brobst et al., 2008; Shin, 2003). It is designed to support strategic as well as tactical decisions, where each unit of data is relevant to some moment in time. Data warehouses get data from different heterogeneous sources. Because source and target tables reside in separate places a continuous flow of data
from source to target is critical to maintain data freshness in the data warehouse.

A variety of ETL tools (Kambayashi et al., 1999) and data warehouse utilities are currently used to get data from source and load it into a data warehouse. Source data is often placed in the staging area (Rahman, 2007) of a data warehouse. Upon loading operational data into the staging area the data is immediately loaded in the analytical subject areas of a data warehouse. Data can be loaded into the analytical tables using database specific transformation tools. Because data is already in staging area and needs to be moved to the analytical area of the same data warehouse, one efficient and convenient way is to load the analytical tables via database specific software, such as stored procedures. It is convenient for the database transformation engine to migrate data from source to the target because the database engine already knows the table structure, data semantics and other parameters.

An efficient, flexible, and scalable data warehousing architecture requires a number of technical advances (Widom, 1995). Incremental (a.k.a., delta) refresh technique is one such important technical advance. In this paper, we explore the possibility of using stored procedures as opposed to external transformation tools. Both full and incremental refreshes can be done by executing database specific stored procedures. Incremental refresh is very important for several reasons. Because data warehouse tables hold a large volume of data it is not practical to perform complete table re-loads at the expense of re-computation during each refresh cycle. Incremental maintenance is usually much cheaper than re-computation (Lee & Kim, 2005; Chan et al., 2000). During the last decade, organizations used to rebuild a data warehouse periodically (Santos & Bernardino, 2008) – monthly, weekly or daily usually in the overnight hours while business users were not online. As the data warehouse increases in complexity and the demand for more up to date data increases, the possibility of maintaining the data warehouse in the same fashion becomes intractable (Bokun & Taglienti, 1998). With the advent of the Internet and increased global business operations by companies and business users located around the world, frequent data freshness has become more important than ever.

In order to increase the frequency of data freshness faster loading is essential. This frequent and faster loading requires availability of additional capacity of database systems. On the other hand, in order to have data warehouse reporting tools run efficiently, enough systems resources need to be reserved for these tools. While the amount of queries requesting up-to-date information is growing and the amount of data in the data warehouses has been increasing, the time window available to refresh data warehouse has been shrinking (Lee et al., 2001; Labio et al., 1999). All these factors demand that data warehouse loads need to be performed quickly, efficiently and without resource constraints. One technique that provides these is Incremental Refresh (Bokun & Taglienti, 1998). Incremental Refresh can help maintain a healthy and stable enterprise data warehousing environment.

The design and maintenance of data warehouses is a difficult task (Naggar et al., 2002) and involves many challenging issues in database research. The ETL transformation for performing incremental refresh is quite complex (Jörg & Deßloch, 2008). In a real world situation the author of this article observed that inefficient and nonstandard methods and logic for incremental refresh cause data consistency and integrity issues. Incorrect or missing data impacts strategic decision making. Ballou & Tayi (1999) argue that data availability is an attribute of data quality. Missing records can cause the need for a full refresh of a table and associated high resource cost due to re-computing. Data warehousing typically involves handling a large number of complex query processing (Sahama & Croll, 2007) that push database management technology to its resource limits in terms of CPU time and IO operation. An