Data Warehousing Development and Design Methodologies

James Yao
Montclair State University, USA

John Wang
Montclair State University, USA

INTRODUCTION

Information systems were developed in early 1960s to process orders, billings, inventory controls, payrolls, and accounts payables. Soon information systems research began. Harry Stern started the "Information Systems in Management Science" column in Management Science journal to provide a forum for discussion beyond just research papers (Banker & Kauffman, 2004). Ackoff (1967) led the earliest research on management information systems for decision-making purposes and published it in *Management Science*. Gorry and Scott Morton (1971) first used the term 'decision support systems' (DSS) in a paper and constructed a framework for improving management information systems. The topics on information systems and DSS research diversifies. One of the major topics has been on how to get systems design right.

As an active component of DSS, which is part of today's business intelligence systems, data warehousing became one of the most important developments in the information systems field during the mid-to-late 1990s. Since business environment has become more global, competitive, complex, and volatile, customer relationship management (CRM) and e-commerce initiatives are creating requirements for large, integrated data repositories and advanced analytical capabilities. By using a data warehouse, companies can make decisions about customer-specific strategies such as customer profiling, customer segmentation, and cross-selling analysis (Cunningham et al., 2006). Thus how to design and develop a data warehouse have become important issues for information systems designers and developers.

This paper presents some of the currently discussed development and design methodologies in data warehousing, such as the multidimensional model vs. relational ER model, CIF vs. multidimensional methodologies, data-driven vs. metric-driven approaches, top-down vs. bottom-up design approaches, data partitioning and parallel processing.

BACKGROUND

Data warehouse design is a lengthy, time-consuming, and costly process. Any wrongly calculated step can lead to a failure. Therefore, researchers have placed important efforts to the study of design and development related issues and methodologies.

Data modeling for a data warehouse is different from operational database data modeling. An operational system, e.g., online transaction processing (OLTP), is a system that is used to run a business in real time, based on current data. An OLTP system usually adopts Entity-relationship (ER) modeling and application-oriented database design (Han & Kamber, 2006). An information system, like a data warehouse, is designed to support decision making based on historical point-in-time and prediction data for complex queries or data mining applications (Hoffer, et al., 2007). A data warehouse schema is viewed as a dimensional model (Ahmad et al., 2004, Han & Kamber, 2006; Levene & Loizou, 2003). It typically adopts either a star or snowflake schema and a subject-oriented database design (Han & Kamber, 2006). The schema design is the most critical to the design of a data warehouse.

Many approaches and methodologies have been proposed in the design and development of data warehouses. Two major data warehouse design methodologies have been paid more attention. Inmon et al. (2000) proposed the Corporate Information Factory (CIF) architecture. This architecture, in the design of the atomic-level data marts, uses denormalized entity-relationship diagram (ERD) schema. Kimball (1996, 1997) proposed multidimensional (MD) architecture.

This architecture uses star schema at atomic-level data marts. Which architecture should an enterprise follow? Is one better than the other?

Currently, the most popular data model for data warehouse design is the dimensional model (Han & Kamber, 2006; Bellatreche, 2006). Some researchers call this model the data-driven design model. Artz (2006), nevertheless, advocates the metric-driven model, which, as another view of data warehouse design, begins by identifying key business processes that need to be measured and tracked over time in order for the organization to function more efficiently. There has always been the issue of top-down vs. bottom-up approaches in the design of information systems. The same is with a data warehouse design. These have been puzzling questions for business intelligent architects and data warehouse designers and developers. The next section will extend the discussion on issues related to data warehouse design and development methodologies.

DESIGN AND DEVELOPMENT METHODOLOGIES

Data Warehouse Data Modeling

Database design is typically divided into a four-stage process (Raisinghani, 2000). After requirements are collected, conceptual design, logical design, and physical design follow. Of the four stages, logical design is the key focal point of the database design process and most critical to the design of a database. In terms of an OLTP system design, it usually adopts an ER data model and an application-oriented database design (Han & Kamber, 2006). The majority of modern enterprise information systems are built using the ER model (Raisinghani, 2000). The ER data model is commonly used in relational database design, where a database schema consists of a set of entities and the relationship between them. The ER model is used to demonstrate detailed relationships between the data elements. It focuses on removing redundancy of data elements in the database. The schema is a database design containing the logic and showing relationships between the data organized in different relations (Ahmad et al., 2004). Conversely, a data warehouse requires a concise, subject-oriented schema that facilitates online data analysis. A data warehouse schema is viewed as a dimensional model which is composed of a central fact table and a set of surrounding dimension tables, each corresponding to one of the components or dimensions of the fact table (Levene & Loizou, 2003). Dimensional models are oriented toward a specific business process or subject. This approach keeps the data elements associated with the business process only one join away. The most popular data model for a data warehouse is multidimensional model. Such a model can exist in the form of a star schema, a snowflake schema, or a starflake schema.

The star schema (see Figure 1) is the simplest database structure containing a fact table in the center, no redundancy, which is surrounded by a set of smaller dimension tables (Ahmad et al., 2004; Han & Kamber, 2006). The fact table is connected with the dimension tables using many-to-one relationships to ensure their hierarchy. The star schema can provide fast response time allowing database optimizers to work with simple database structures in order to yield better execution plans.

The snowflake schema (see Figure 2) is a variation of the star schema model, in which all dimensional information is stored in the third normal form, thereby further splitting the data into additional tables, while keeping fact table structure the same. To take care of hierarchy, the dimension tables are connected with sub-dimension tables using many-to-one relationships. The resulting schema graph forms a shape similar to a snowflake (Ahmad et al., 2004; Han & Kamber, 2006). The snowflake schema can reduce redundancy and save storage space. However, it can also reduce the effectiveness of browsing and the system performance may be adversely impacted. Hence, the snowflake schema is not as popular as star schema in data warehouse design (Han & Kamber, 2006). In general, the star schema requires greater storage, but it is faster to process than the snowflake schema (Kroenke, 2004).

The starflake schema (Ahmad et al., 2004), also called galaxy schema or fact constellation schema (Han & Kamber, 2006), is a combination of the denormalized star schema and the normalized snowflake schema (see Figure 3). The starflake schema is used in situations where it is difficult to restructure all entities into a set of distinct dimensions. It allows a degree of crossover between dimensions to answer distinct queries (Ahmad et al., 2004). Figure 3 illustrates the starflake schema.

What needs to be differentiated is that the three schemas are normally adopted according to the differ-