Conceptual Modeling for Data Warehouse and OLAP Applications

Elzbieta Malinowski
Universidad de Costa Rica, Costa Rica

Esteban Zimányi
Université Libre de Bruxelles, Belgium

INTRODUCTION

The advantages of using conceptual models for database design are well known. In particular, they facilitate the communication between users and designers since they do not require the knowledge of specific features of the underlying implementation platform. Further, schemas developed using conceptual models can be mapped to different logical models, such as the relational, object-relational, or object-oriented models, thus simplifying technological changes. Finally, the logical model is translated into a physical one according to the underlying implementation platform.

Nevertheless, the domain of conceptual modeling for data warehouse applications is still at a research stage. The current state of affairs is that logical models are used for designing data warehouses, i.e., using *star* and *snowflake* schemas in the relational model. These schemas provide a multidimensional view of data where *measures* (e.g., quantity of products sold) are analyzed from different perspectives or *dimensions* (e.g., by product) and at different levels of detail with the help of *hierarchies*. On-line analytical processing (OLAP) systems allow users to perform automatic aggregations of measures while traversing hierarchies: the roll-up operation transforms detailed measures into aggregated values (e.g., daily into monthly sales) while the drill-down operation does the contrary.

Star and snowflake schemas have several disadvantages, such as the inclusion of implementation details and the inadequacy of representing different kinds of hierarchies existing in real-world applications. In order to facilitate users to express their analysis needs, it is necessary to represent data requirements for data warehouses at the conceptual level. A conceptual multidimensional model should provide a graphical support (Rizzi, 2007) and allow representing facts, measures, dimensions, and different kinds of hierarchies.

BACKGROUND

Star and snowflake schemas comprise relational tables termed *fact* and *dimension tables*. An example of star schema is given in Figure 1.

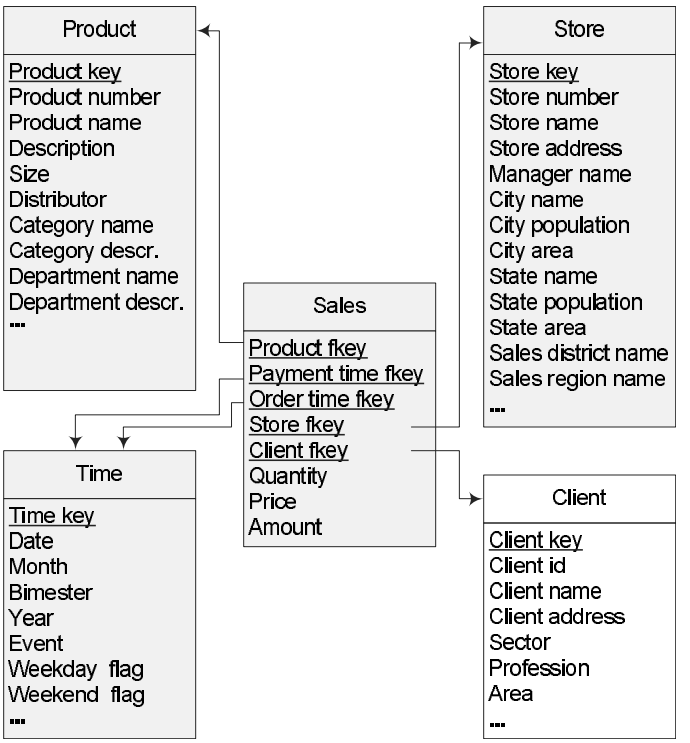
Fact tables, e.g., Sales in Figure 1, represent the focus of analysis, e.g., analysis of sales. They usually contain numeric data called *measures* representing the indicators being analyzed, e.g., Quantity, Price, and Amount in the figure. *Dimensions*, e.g., Time, Product, Store, and Client in Figure 1, are used for exploring the measures from different analysis perspectives. They often include attributes that form *hierarchies*, e.g., Product, Category, and Department in the Product dimension, and may also have descriptive attributes.

Star schemas have several limitations. First, since they use de-normalized tables they cannot clearly represent hierarchies: The hierarchy structure must be deduced based on knowledge from the application domain. For example, in Figure 1 is not clear whether some dimensions comprise hierarchies and if they do, what are their structures.

Second, star schemas do not distinguish different kinds of measures, i.e., additive, semi-additive, non-additive, or derived (Kimball & Ross, 2002). For example, Quantity is an additive measure since it can be summarized while traversing the hierarchies in all dimensions; Price is a non-additive measure since it cannot be meaningfully summarized across any dimension; Amount is a derived measure, i.e., calculated based on other measures. Although these measures require different handling during aggregation, they are represented in the same way.

Third, since star schemas are based on the relational model, implementation details (e.g., foreign keys) must be considered during the design process. This requires technical knowledge from users and also makes difficult the process of transforming the logical model to other models, if necessary.

Figure 1. Example of a star schema for analyzing sales



Fourth, dimensions may play different roles in a fact table. For example, the Sales table in Figure 1 is related to the Time dimension through two dates, the order date and the payment date. However, this situation is only expressed as foreign keys in the fact table that can be difficult to understand for non-expert users.

Snowflake schemas have the same problems as star schemas, with the exception that they are able to represent hierarchies. The latter are implemented as separate tables for every hierarchy level as shown in Figure 2 for the Product and Store dimensions. Nevertheless, snowflake schemas only allow representing simple hierarchies. For example, in the hierarchy in Figure 2 a) it is not clear that the same product can belong to several categories but for implementation purposes only the primary category is kept for each product. Furthermore, the hierarchy formed by the Store, Sales district, and Sales region tables does not accurately represent users' requirements: since small sales regions are not divided into sales districts, some stores must be analyzed using the hierarchy composed only of the Store and the Sales region tables.

Several conceptual multidimensional models have been proposed in the literature[1]. These models include the concepts of facts, measures, dimensions, and hierarchies. Some of the proposals provide graphical representations based on the ER model (Sapia, Blaschka, Höfling, & Dinter, 1998; Tryfona, Busborg, & Borch, 1999), on UML (Abelló, Samos, & Saltor, 2006; Luján-Mora, Trujillo, & Song, 2006), or propose new notations (Golfarelli & Rizzi, 1998; Hüsemann, Lechtenbörger, & Vossen, 2000), while other proposals do not refer to graphical representations (Hurtado & Gutierrez, 2007; Pourabbas, & Rafanelli, 2003; Pedersen, Jensen, & Dyreson, 2001; Tsois, Karayannidis, & Sellis, 2001).

Very few models distinguish the different types of measures and refer to role-playing dimensions (Kimball & Ross, 2002, Luján-Mora *et al.*, 2006). Some models do not consider the different kinds of hierarchies existing in real-world applications and only support simple hierarchies (Golfarelli & Rizzi, 1998; Sapia *et al.*, 1998). Other models define some of the hierarchies described in the next section (Abelló *et al.*, 2006; Bauer, Hümmer,