Cube Algebra: A Generic User-Centric Model and Query Language for OLAP Cubes

Cristina Ciferri, Department of Computer Science, Universidade de São Paulo at São Carlos, São Carlos, Brazil
Ricardo Ciferri, Department of Computer Science, Universidade de São Paulo at São Carlos, São Carlos, Brazil
Leticia Gómez, Department of Software Engineering, Instituto Tecnológico de Buenos Aires, Buenos Aires, Argentina
Markus Schneider, Department of Computer and Information Science and Engineering, University of Florida, Gainesville, FL, USA
Alejandro Vaisman, Department of Computer & Decision Engineering, Université Libre de Bruxelles, Brussels, Belgium
Esteban Zimányi, Department of Computer & Decision Engineering (CoDE), Université Libre de Bruxelles, Brussels, Belgium

ABSTRACT

The lack of an appropriate conceptual model for data warehouses and OLAP systems has led to the tendency to deploy logical models (for example, star, snowflake, and constellation schemas) for them as conceptual models. ER model extensions, UML extensions, special graphical user interfaces, and dashboards have been proposed as conceptual approaches. However, they introduce their own problems, are somehow complex and difficult to understand, and are not always user-friendly. They also require a high learning curve, and most of them address only structural design, not considering associated operations. Therefore, they are not really an improvement and, in the end, only represent a reflection of the logical model. The essential drawback of offering this system-centric view as a user concept is that knowledge workers are confronted with the full and overwhelming complexity of these systems as well as complicated and user-unfriendly query languages such as SQL OLAP and MDX. In this article, the authors propose a user-centric conceptual model for data warehouses and OLAP systems, called the Cube Algebra. It takes the cube metaphor literally and provides the knowledge worker with high-level cube objects and related concepts. A novel query language leverages well known high-level operations such as roll-up, drill-down, slice, and drill-across. As a result, the logical and physical levels are hidden from the unskilled end user.

Keywords: Conceptual Model, Cube, Data Warehouses, OLAP, Query Language, User-Centric Model

DOI: 10.4018/jdwm.2013040103
INTRODUCTION

Nowadays, data warehouses are at the forefront of information technology applications as a way for organizations to effectively use and analyze information for business planning and decision making. Data warehouses are large repositories of analytical and subject-oriented data integrated from several heterogeneous sources over a large period of time. The technique of performing complex analysis over the information stored in the data warehouse is commonly called Online Analytical Processing (OLAP). A review of the evolution of data warehouse technology reveals that research and development has mainly focused on system aspects such as the construction of data warehouses, materialization, indexing, and the implementation of OLAP functionality. This system-centric view has led to well-established and commercialized technologies such as relational OLAP (ROLAP), multidimensional OLAP (MOLAP), and hybrid OLAP (HOLAP) at the logical and the physical levels.

However, the unskilled user such as the manager in a consulting company or the analyst in a financial institution is confronted with the problem that the handling of data warehouses and OLAP systems requires expert knowledge due to complicated data warehouse structures and the complexity of OLAP systems and query languages. Two main reasons are responsible for this problem. First, due to the lack of a generic, user-friendly, and comprehensible conceptual data model, data warehouse design is usually performed at the logical level and leads to the exposure of the logical design schemas that are difficult to understand by the unskilled user. In a ROLAP environment, for example, the user is faced with the logical design of relational tables in terms of star, snowflake, or fact constellation schemas. The proposal to alleviate the problem by providing extensions to the Entity-Relationship Model and the Unified Modeling Language, or by offering specific graphical user interfaces or dashboards for data warehouse design is not really convincing since ultimately they represent a reflection and visualization of relational technology concepts and, in addition, reveal their own problems. Second, available OLAP query and analysis languages such as MDX and SQL OLAP operate at the logical level and require the user’s deep understanding of the data warehouse structure in order to be able to formulate queries. These languages are quite complex, overwhelm the unskilled user, and are therefore inappropriate as end-user languages.

We conclude that a generic, conceptual, and user-centric data warehouse model that focuses on user requirements is missing and needed. Such a model should fulfill several design criteria. First, it should be located above the logical level. Second, it should abstract from and be independent of the models and technologies (ROLAP, MOLAP, HOLAP) at the logical level. Third, it should be able to cooperate with any of these logical models and technologies. Fourth, it should enable the user to generically and abstractly represent and query hierarchical multidimensional data. Fifth, it should have an associated query language based exclusively on the conceptual level, thus providing high-level query operations for the user. The goal of this article is to propose and formally describe a conceptual and user-centric data warehouse model and query language that satisfies these design criteria. Surprisingly, the conceptual view this model adopts is not new; on the contrary, it is well known. However, the way and resoluteness in which we offer this concept is novel. Our proposed conceptual model leverages the cube view of data warehouses but takes the cube metaphor literally. This means that the user’s conceptual world is solely the cube that the user can create, manipulate, update, and query. The cube is used as the user concept that completely abstracts from any logical and physical implementation details. Technically, this implies that cubes can be regarded as an abstract data type that provides cubes as the only kind of values (objects), offers high-level operations on cubes or between cubes such as slice, dice, drill-down, roll-up, and drill-across as the only available access methods, and hides any data representation and algorithmic details.
Soft Computing and its Applications
Siddhartha Bhattacharyya, Ujiwal Maulik and Sanghamitra Bandyopadhyay (2013).
Data Mining: Concepts, Methodologies, Tools, and Applications (pp. 366-394).
www.igi-global.com/chapter/soft-computing-its-applications/73448?camid=4v1a

Using Big Data to Improve the Educational Infrastructure and Learning Paradigm
Areej Fatemah Meghji and Naeem A. Mahoto (2016). Effective Big Data Management and Opportunities for Implementation (pp. 158-181).
www.igi-global.com/chapter/using-big-data-to-improve-the-educational-infrastructure-and-learning-paradigm/157691?camid=4v1a