Encyclopedia of Data Warehousing and Mining

Second Edition

John Wang
Montclair State University, USA

Volume IV
Pro–Z

Information Science REFERENCE

INFORMATION SCIENCE REFERENCE
Hershey · New York

Director of Editorial Content: Kristin Klinger
Director of Production: Jennifer Neidig
Managing Editor: Jamie Snavely
Assistant Managing Editor: Carole Coulson
Typesetter: Amanda Appicello, Jeff Ash, Mike Brehem, Carole Coulson, Elizabeth Duke, Jen Henderson, Chris Hrobak, Jennifer Neidig, Jamie Snavely, Sean Woznicki
Cover Design: Lisa Tosheff
Printed at: Yurchak Printing Inc.

Published in the United States of America by
Information Science Reference (an imprint of IGI Global)
701 E. Chocolate Avenue, Suite 200
Hershey PA 17033
Tel: 717-533-8845
Fax: 717-533-8661
E-mail: cust@igi-global.com
Web site: http://www.igi-global.com/reference

and in the United Kingdom by
Information Science Reference (an imprint of IGI Global)
3 Henrietta Street
Covent Garden
London WC2E 8LU
Tel: 44 20 7240 0856
Fax: 44 20 7379 0609
Web site: http://www.eurospanbookstore.com

Library of Congress Cataloging-in-Publication Data

Encyclopedia of data warehousing and mining / John Wang, editor. -- 2nd ed.
p. cm.
Includes bibliographical references and index.
Summary: "This set offers thorough examination of the issues of importance in the rapidly changing field of data warehousing and mining"--Provided by publisher.
ISBN 978-1-60566-010-3 (hardcover) -- ISBN 978-1-60566-011-0 (ebook)
1. Data mining. 2. Data warehousing. I. Wang, John,
QA76.9.D37E52 2008
005.74--dc22
2008030801

British Cataloguing in Publication Data
A Cataloguing in Publication record for this book is available from the British Library.

Temporal Extension for a Conceptual Multidimensional Model

Elzbieta Malinowski
Universidad de Costa Rica, Costa Rica

Esteban Zimányi
Université Libre de Bruxelles, Belgium

INTRODUCTION

Data warehouses integrate data from different source systems to support the decision process of users at different management levels. Data warehouses rely on a multidimensional view of data usually represented as relational tables with structures called *star* or *snowflake schemas*. These consist of *fact tables*, which link to other relations called *dimension tables*. A fact table represents the focus of analysis (e.g., analysis of sales) and typically includes attributes called *measures*. Measures are usually numeric values (e.g., quantity) used for performing quantitative evaluation of different aspects in an organization. Measures can be analyzed according to different analysis criteria or *dimensions* (e.g., store dimension). Dimensions may include *hierarchies* (e.g., month-year in the time dimension) for analyzing measures at different levels of detail. This analysis can be done using on-line analytical processing (OLAP) systems, which allow dynamic data manipulations and aggregations. For example, the roll-up operation transforms detailed measures into aggregated data (e.g., daily into monthly or yearly sales) while the drill-down operations does the contrary.

Multidimensional models include a time dimension indicating the timeframe for measures, e.g., 100 units of a product were sold in March 2007. However, the time dimension cannot be used to keep track of changes in other dimensions, e.g., when a product changes its ingredients. In many cases the changes of dimension data and the time when they have occurred are important for analysis purposes. Kimball and Ross (2002) proposed several implementation solutions for this problem in the context of relational databases, the so-called *slowly-changing dimensions.* Nevertheless, these solutions are not satisfactory since either they do not preserve the entire history of data or are difficult to implement. Further, they do not consider the research realized in the field of temporal databases.

Temporal databases are databases that support some aspects of time (Jensen & Snodgrass, 2000). This support is provided by means of different temporality types[1], to which we refer in the next section. However, even though temporal databases allow to represent and to manage time-varying information, they do not provide facilities for supporting decision-making process when aggregations of high volumes of historical data are required. Therefore, a new field called *temporal data warehouses* joins the research achievements of temporal databases and data warehouses in order to manage time-varying multidimensional data.

BACKGROUND

Temporal support in data warehouses is based on the different temporality types used in temporal databases. *Valid time* (VT) specifies the time when data is true in the modeled reality, e.g., the time when a specific salary was paid for an employee. Valid time is typically provided by users. *Transaction time* (TT) indicates the time when data is current in the database and may be retrieved. It is generated by the database management system (DBMS). Both temporality types, i.e., valid time and transaction time, can be combined defining *bitemporal time* (BT). Finally, changes in time defined for an object as a whole define the *lifespan* (LS) of an object, e.g., the time when an employee was working for a company.

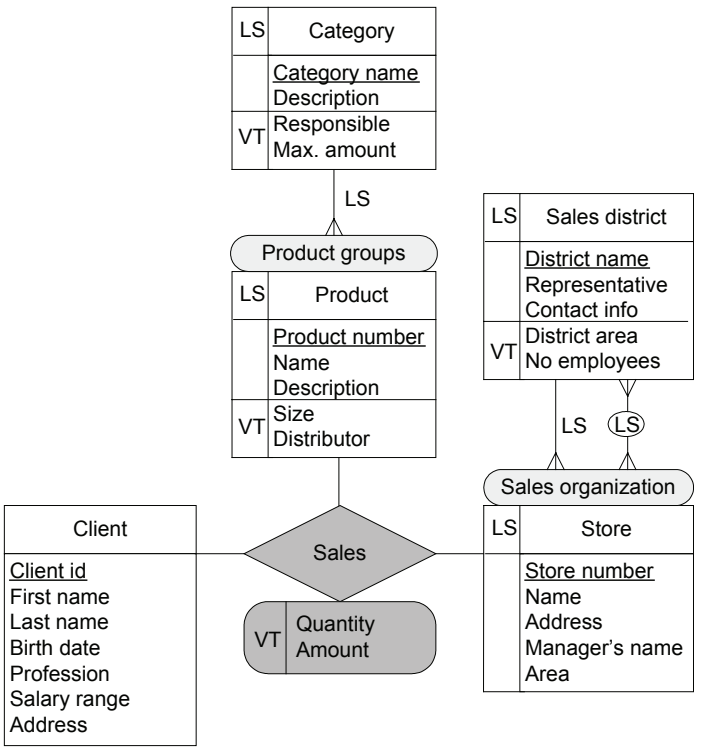
One characteristic of temporality types is their precision or *granularity*, indicating the duration of the time units that are relevant for an application. For example, the valid time associated to an employee's salary may be of granularity month. On the other hand, transaction time being system defined is typically of a millisecond granularity.

There is still lack of an analysis determining which temporal support is important for data warehouse ap-

plications. Most works consider valid time (e.g., Body, Miquel, Bédard, & Tchounikine, 2003; Wrembel & Bebel, 2007; Mendelzon & Vaisman, 2003). To our knowledge, no work includes lifespan support in temporal data warehouses. However, lifespan is important since it can help to discover, e.g., how the exclusion of some products influences sales. On the other hand, very few works relate to transaction time. For example, Martín and Abelló (2003) transform transaction time from source systems to represent valid time. This approach is semantically incorrect because data may be included in databases after their period of validity has expired. Further, transaction time coming from source system plays an important role in temporal data warehouses when traceability is required, e.g., for fraud detection. Other authors consider transaction time generated in temporal data warehouses in the same way as transaction time in temporal databases (e.g., Martín & Abelló, 2003; Mendelzon & Vaisman, 2003; Ravat & Teste, 2006). However, since data in temporal data warehouses is neither modified nor deleted, transaction time in a data warehouse represents the time when data was loaded into a data warehouse. Therefore, we propose to call it *loading time* (LT) (Malinowski & Zimányi, 2006a). LT can differ from transaction time or valid time of source systems due to the delay between the time when the changes have occurred in source systems and the time when these changes are integrated into a temporal data warehouse. Another approach (Brucker & Tjoa, 2002) considers valid time, transaction time, and loading time. However, they limit the usefulness of these temporality types for only active data warehouses, i.e., for data warehouses that include event-condition-action rules (or triggers).

The inclusion of temporal support raise many issues, such as efficient temporal aggregation of multidimensional data (Moon, Vega, & Immanuel, 2003), correct aggregation in presence of data and schema changes (Body *et al.*, 2003; Eder, Koncilia, & Morzy, 2002; Wrembel & Bebel, 2007; Mendelzon & Vaisman, 2003; Golfarelli, Lechtenbörger, Rizzi, & Vossen, 2006), or temporal view materialization from non-temporal sources (Yang & Widom, 1998). Even though the works related to schema and data changes define models for temporal data warehouses, what is still missing is a conceptual model that allows decision-making

Figure 1. An example of a multidimensional schema for a temporal data warehouse



users to represent data requirement for temporal data warehouses. This model should allow specifying multidimensional elements, i.e., dimensions, hierarchies, facts, and measures, and allow users to clearly indicate which elements they want to be time invariant and for which the changes in time should be kept.

MAIN FOCUS

In this section we present the MultiDim model, a conceptual multidimensional model (Malinowski & Zimányi, 2008a)[2] extended with temporal support (Malinowski & Zimányi, 2006a, 2006b). This model allows users and designers to represent at the conceptual level all elements required in temporal data warehouse applications.

Multidimensional Model for Temporal Data Warehouses

The MultiDim model supports different temporality types: lifespan (LS), valid time (VT), transaction time (TT), and bitemporal time (BT) coming from source systems (if available) and additionally, loading time (LT) generated in a data warehouse.

To describe our model we use the example shown in Figure 1. It includes a set of levels organized into dimensions and a fact relationship. A *level* corresponds to an entity type in the entity-relationship model and represents a set of instances called *members* that have common characteristics. For example, Product, Category, and Store are some of the levels in Figure 1. Levels contain one or several *key attributes* (underlined in Figure 1), identifying uniquely the members of a level, and may also have other *descriptive* attributes.

A *temporal level* is a level for which the application needs to keep the lifespan of its members (e.g., inserting or deleting a product). A *temporal attribute* is an attribute that keeps the changes in its value (e.g., changing a product's size) and the time when they occur. For example, the Product level in Figure 1 is a temporal level that includes temporal attributes Size and Distributor.

A *fact relationship* expresses the focus of analysis and represents an n-ary relationship between levels. For example, the Sales fact relationship between the Product, Store, and Client levels in Figure 1 is used for analyzing sales in different stores.

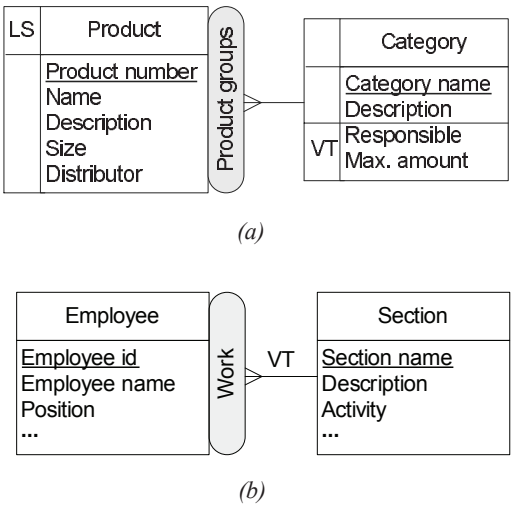
A fact relationship may contain attributes commonly called *measures*. They contain data (usually numerical) that are analyzed using the different dimensions. For example, the Sales fact relationship in Figure 1 includes the measures Quantity and Amount. In the MultiDim model measures are *temporal*, i.e., they always require a temporality type (VT, TT, BT, and/or LT).

A *dimension* allows to group data that shares a common semantic meaning within the domain being modeled, e.g., all data related to a product. It is composed of either a level or one or more hierarchies. For example, Client in Figure 1 is a one-level dimension.

Hierarchies are required for allowing users to analyze data at different levels of detail. A hierarchy contains several related levels, e.g., Product and Category in Figure 1. Given two related levels of a hierarchy, one of them is called *child* and the other *parent* depending on whether they include more detailed or more general data, respectively. In Figure 1, the Product level is a child level while the Category level is a parent level. Key attributes of a parent level define how child members are grouped for the roll-up operation. For example, in Figure 2 since Category name is the key of the Category level, products will be grouped according to the category to which they belong.

The relationships composing the hierarchies are called *child-parent relationships*. These relationships are characterized by *cardinalities* that indicate the minimum and the maximum number of members in one level that can be related to a member in another level. Child-parent relationships may include temporal support. For example, in Figure 1 the LS symbol between Product and Category indicates that the evolution on time of assignments of products to categories will be kept. Temporal support for relationships leads to two interpretations of cardinalities. The *snapshot cardinality* is valid at every time instant whereas the *lifespan cardinality* is valid over the entire member's lifespan. The former cardinality is represented using the symbol indicating temporality type next to the link between levels while the lifespan cardinality includes the LS symbol surrounded by a circle. In Figure 1, the snapshot cardinality between Product and Category levels is many-to-one while the lifespan cardinality is many-to-many. They indicate that a product belongs to only one category at every time instant but belongs to many categories over its lifespan, i.e., its assignment to categories may change. The relationship between levels may include different temporality types: LS, TT, combination of both, and/or LT.

Figure 2. Examples of temporal hierarchies: a) non-temporal relationship between a temporal and a non-temporal levels, and b) temporal relationship between non-temporal levels



Since hierarchies in a dimension may express different conceptual structures used for analysis purposes, we use a *criterion name* to differentiate them, such as Product groups or Sales organization in Figure 1.

Modeling Temporal Aspects

Our model supports temporality in an orthogonal way, i.e., hierarchies may contain temporal or non-temporal levels associated with temporal or non-temporal links. Similarly, temporal or non-temporal levels may have temporal or non-temporal attributes. This approach differs from the one used in many temporal models. We consider that it is important to allow users to choose which elements should be temporal or not.

For example, Figure 2 (a) represents a hierarchy composed by a temporal level (Product) and a non-temporal level (Category) with a temporal attribute associated with a non-temporal link. Therefore, the lifespan of products as well as the changes of responsible and maximum amount of categories are kept; on the other hand, other attributes of the levels either do not change or only the last modification is kept. The example in Figure 2 (b) shows a hierarchy with two non-temporal levels associated with a temporal link. This keeps track of the evolution of relationships between employees and sections but we do not store the changes in levels, e.g., when an employee changes its position.

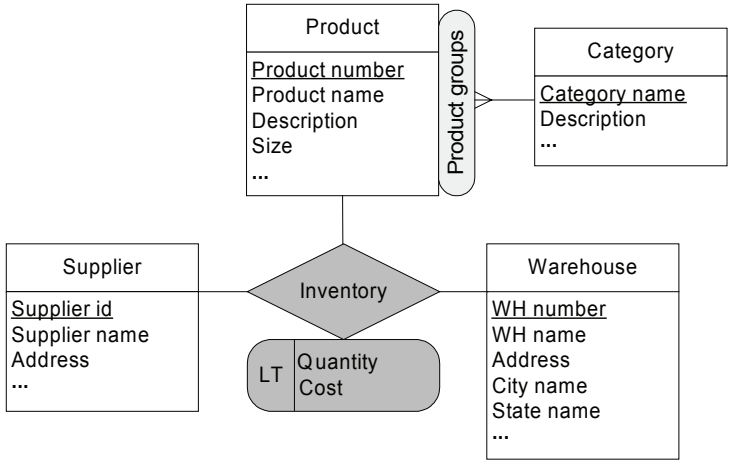
However, to ensure correct management of hierarchies in temporal data warehouses and avoid dangling references, i.e., references to non-existing elements, several constraints should be ensured (Malinowski & Zimányi, 2006a).

Another characteristic of our model is that it uses a consistent approach for providing temporal support for the different elements of a multi-dimensional model, i.e., for levels, hierarchies, and measures. Our model avoids mixing two different approaches where dimensions include explicit temporal support while measures require the presence of the traditional time dimension for keeping track of changes. Since measures are attributes of fact relationships, we provide temporal support for them in the same way as it is done for levels' attributes.

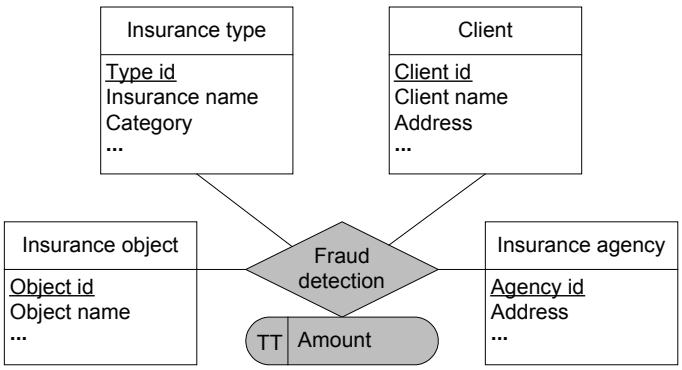
An important question is thus whether it is necessary to have a time dimension in the schema when including temporality types for measures. If all attributes of the time dimension can be obtained by applying time manipulation functions, such as the corresponding week, month, or quarter, this dimension is not required anymore. However, in some temporal data warehouse applications this calculation can be very time-consuming, or the time dimension contains data that cannot be derived, e.g., events such as promotional seasons. Thus, the time dimension is included in a schema depending on users' requirements and the capabilities provided by the underlying DBMS.

Another aspect is the inclusion of different temporality types for measures. The usual practice in temporal data warehouses is to associate valid time support with measures. However, different temporal support can be available in source systems. In Malinowski and Zimányi (2006b), we present several real-world scenarios that include different temporality types for measures enriching the analysis spectrum. The examples in Figure 3 show simplified schemas that include, respectively, loading time and transaction time for measures. The former is used when users require the history of how source data has evolved, but sources are either non-temporal or temporal support is implemented in an ad-hoc manner that can be both inefficient and difficult to automate (Yang & Widom, 1998). The schema in Figure 3 b) is used for an insurance company having as analysis focus the amount of insurance payments. Such a schema

Figure 3. Examples of schemas with a) LT and b) TT support for measures



(a)



(b)

could be used, e.g., to track internal frauds that modify the amount of insurance paid to clients, since the time when the measure values change is kept.

In Malinowski and Zimányi (2006b) we discuss several problems that may occur when data from source systems is aggregated before being loaded into temporal data warehouses. These include different time granularities between source systems and a data warehouse, measure aggregations with different time granularities, and temporal support for aggregated measures.

FUTURE TRENDS

An important issue in temporal data warehouses is the aggregation in the presence of changes in dimension data, schema, or both. Different solutions have already been proposed (e.g., Eder *et al.*, 2002; Mendelzon & Vaisman, 2003; Wrembel & Bebel, 2007; Body *et al.*, 2003). Nevertheless, these proposals require specific software and query languages for implementing and manipulating multidimensional data that vary over time. Further, they do not consider different aspects as mentioned in this paper, e.g., different temporal support in hierarchies and measures.

Another issue is the implementation of temporal data warehouses in current DBMSs, which do not yet provide temporal support. To our knowledge, there are very few proposals for implementing temporal data warehouses in current DBMSs (e.g., Martín & Abelló, 2003; Malinowski & Zimányi, 2006c; Ravat *et al.*, 1999; Mendelzon & Vaisman, 2003). However, only the latter authors provide manipulation features for the proposed structures.

CONCLUSION

Combining the two research areas of data warehouses and temporal databases, allows one to combine the achievements of each of them leading to the emerging field of temporal data warehouses. The latter raises several research issues, such as the inclusion of different temporal support, conceptual modeling, and measure aggregations, among others.

In this paper, we first proposed the inclusion of four different temporality types: three of them come from source systems (if they are available), i.e., valid time, transaction time (or combination of both), and lifespan. A new temporality type, called loading time, is generated in temporal data warehouses. It indicates when data was stored in a temporal data warehouse.

We also presented a conceptual model that is able to express users' requirements for time-varying multidimensional data. The MultiDim model allows temporal support for levels, attributes, relationships between levels forming a hierarchy, and measures. For temporal hierarchies and measures, we discussed different issues that are relevant to ensure the correct data management in temporal data warehouses.

The inclusion of temporality types in a conceptual model allows users, designers, and implementers to include temporal semantics as an integral part of temporal data warehouses. In this way, temporal extensions offer more symmetry to multidimensional models representing in a symmetric manner changes and the time when they occur for all elements of a data warehouse. Since conceptual models are platform independent, logical and physical models can be derived from such a conceptual representation.

REFERENCES

Body, M., Miquel, M., Bédard, Y., & Tchounikine, A. (2003). Handling Evolution in Multidimensional Structures. *Proceedings of the 19th International Conference on Data Engineering*, pp. 581-592. IEEE Computer Society Press.

Bruckner, R. & Tjoa, A. (2002). Capturing Delays and Valid Times in Data Warehouses: Towards Timely Consistent Analyses, *Journal of Intelligent Information Systems*, *19*(2), pp. 169-190.

Eder, J., Koncilia, Ch., & Morzy, T. (2002). The COMET Metamodel for Temporal Data Warehouses. *Proceedings of the 14th International Conference on Advanced Information Systems Engineering*, pp. 83-99. Lecture Notes in Computer Science, N° 2348. Springer.

Golfarelli, M., Lechtenbörger, J., Rizzi, S., & Vossen, V. (2006). Schema Versioning in Data Warehouses: Enabling Cross-Version Querying Via Schema Augmentation. *Data & Knowledge Engineering*, *59*(2), pp. 435-459.

Jensen, C.S. & Snodgrass, R. (2000). Temporally Enhanced Database Design. In M. Papazoglou, S. Spaccapietra, & Z. Tari (Eds.), *Advances in Object-Oriented Data Modeling,* pp. 163-193. Cambridge, MIT Press.

Kimball, R., & Ross, M. (2002). *The Data Warehouse Toolkit.* John Wiley & Sons Publishers.

Malinowski, E. & Zimányi, E. (2008a). Multidimensional Conceptual Models, *in this book*.

Malinowski, E. & Zimányi, E. (2008b). *Advanced Data Warehouse Design: From Conventional to Spatial and Temporal Applications*. Springer.

Malinowski, E. & Zimányi, E. (2006a). A Conceptual Solution for Representing Time in Data Warehouse Dimensions. *Proceedings of the 3rd Asia-Pacific Conference on Conceptual Modelling*, pp. 45-54. Australian Computer Society.

Malinowski, E. & Zimányi, E. (2006b). Inclusion of Time-Varying Measures in Temporal Data Warehouses. *Proceedings of the 8th International Conference on Enterprise Information Systems*, pp. 181-186.

Malinowski, E. & Zimányi, E. (2006c). Object-Relational Representation of a Conceptual Model for Temporal Data Warehouses. *Proceedings of the 18th International Conference on Advanced Information Systems Engineering*, pp. 96-110. Lecture Notes in Computer Science, N° 4001. Springer.

Martín, C. & Abelló, A. (2003). A Temporal Study of Data Sources to Load a Corporate Data Warehouse. *Proceedings of the 5th International Conference on Data Warehousing and Knowledge Discovery*, pp. 109-118. Lecture Notes in Computer Science, N° 2737. Springer.

Mendelzon, A. & Vaisman, A. (2003). Time in Multidimensional Databases. In M. Rafanelli (Ed.), *Multi-*

dimensional Databases: Problems and Solutions, pp. 166-199. Idea Group Publishing.

Moon, B., Vega, F., & Immanuel, V. (2003). Efficient Algorithms for Large-Scale Temporal Aggregation. *IEEE Transactions on Knowledge and Data Engineering*, *15*(3), pp. 744-759.

Ravat, F. & Teste, F. (2006). Supporting Changes in Multidimensional Data Warehouses. *International Review of Computer and Software*, *1*(3), pp. 251-259.

Wrembel, T. & Bebel, B.(2007). Metadata Management in a Multiversion Data Warehouse. In S. Spaccapietra (Ed.), *Journal on Data Semantics, VIII,* pp. 118-157. Lecture Notes in Computer Science, N° 4380. Springer

Yang, J. & Widom, J. (1998). Maintaining Temporal Views Over Non-Temporal Information Source for Data Warehousing. *Proceedings of the 6th International Conference on Extending Database Technology*, pp. 389-403. Lecture Notes in Computer Science, N° 1377. Springer.

KEY TERMS

Bitemporal: A combination of both transaction and valid time.

Conceptual Model: A model for representing schemas that are designed to be as close as possible to users' perception, not taking into account any implementation considerations.

Loading Time: A temporal specification that keeps the information when a data element is stored in a data warehouse.

Granularity: A partitioning of a domain in groups of elements where each group is perceived as an indivisible unit (a granule) at a particular abstraction level.

Lifespan: The record of the evolution of the membership of an instance into its type.

Multidimensional Model: A model for representing the information requirements of analytical applications. A multidimensional model comprises facts, measures, dimensions, and hierarchies.

Temporality Types: Different temporal support that can be provided by a system. They include transaction time, valid time, lifespan, and loading time.

Transaction Time: A temporal specification that keeps the information on when a data element is stored in and deleted from the database.

Valid Time: A temporal specification that keeps information on when a data element stored in the database is considered valid in the perceived reality from the application viewpoint.

ENDNOTES

1 They are usually called time dimensions; however, we use the term "dimension" in the multidimensional context.

2 We only include some elements of the MultiDim model in order to focus on its temporal extension.