Association for Information Systems
AIS Electronic Library (AISeL)

AMCIS 1999 Proceedings

Americas Conference on Information Systems (AMCIS)

December 1999

Dimension Hierarchy Diagrams

Donald Berndt
University of South Florida

Dimension Hierarchy Diagrams

Donald J. Berndt
Information Systems and Decision Sciences
College of Business Administration
University of South Florida
dberndt@coba.usf.edu

Introduction

There has been explosive growth in the use of data warehousing technology in the construction of decision support infrastructures. Industrial applications have highlighted the dichotomy between on-line transaction processing and data warehousing, with a focus on read-only ad-hoc analysis of business data [Inmon 1992]. Data warehouse development is based on business processes and associated decision-making tasks, and therefore may result in critical information system components. The design and enhancement of important decision-making activities may be a source of competitive advantage. While some traditional transaction processing systems are purchased "off the shelf," the promise of competitive advantage may favor custom development of some data warehouse components. Therefore, design tools that support technical development and communication with business users are essential.

Most data warehouses are designed using *dimensional modeling* techniques (for thorough coverage of data warehouse design, see [Kimball 1996] or [Kimball 1998]). The dimensional model or *star schema* consists of a center *fact table* with a set of *dimension tables* radiating from the fact table. The center fact table usually contains numeric attributes that can be aggregated across the dimensions in various combinations, supporting roll-up/drill-down operations. For example, a water resource management data warehouse might have a center table with monthly rainfall levels and dimensions such as time, data collection site, measurement type, and responsible agency (see Figure 1). Dimensions provide the attributes for query specification and are usually designed as hierarchies for roll-up/drill-down operations. While facts provide the substance, the dimensions provide the ability to specify a wide range of interesting queries—the spice. Therefore, careful design effort toward providing a rich set of dimension attributes, usually organized as a hierarchy, will create a robust ad-hoc query environment for analytic processing.

The Anatomy of a Dimension

In most examples of data warehouse design, hierarchical dimensions are depicted as "pure" hierarchies composed of a simple cascade of one-to-many relationships (though reality may force us to use more complex arrangements). For example, consider again the

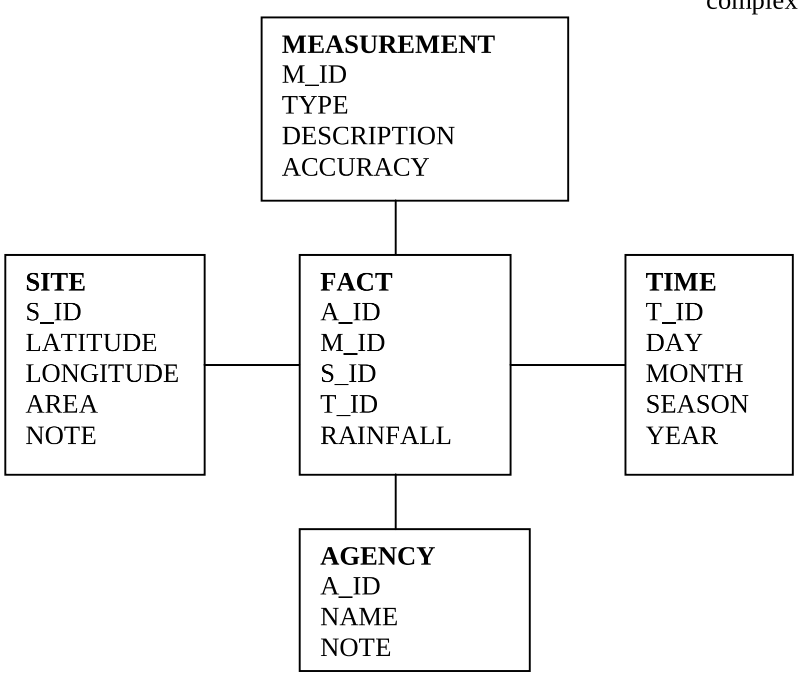

Figure 1: A Simple Star Schema

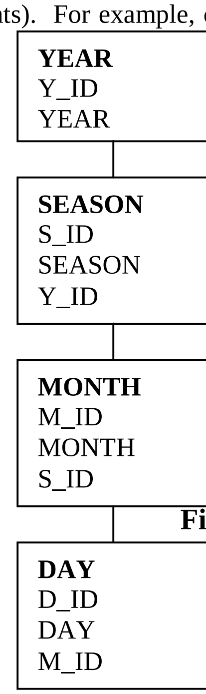

Figure 2: Time Dimension

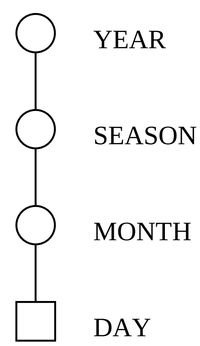

Figure 3: A Simple DHD

design fragment in Figure 1, and the expanded temporal hierarchy in Figure 2. The day a rainfall observation was made falls within a month, a month within a season, and a season within a year (see Figure 2). This clean hierarchical arrangement allows us to easily aggregate rainfall data as we move "up" the hierarchy, exploring the data at an appropriate level of detail.

A hierarchical arrangement also gives us the freedom as designers to *denormalize* the hierarchies, producing a single relation with repeated attribute values. Collapsing the hierarchy in Figure 2 would produce a relation where each row would contain a day, as well as repeated values for a month, a season, and a year. The reason for denormalization in any database is usually performance. This is particularly true in data warehouse environments where browsing performance is one of the design grails. Many designers advocate denormalized dimensions for performance reasons. However, good database design calls for clean design and denormalization with documented justification.

It is important to note that the dimension hierarchies may fall from a state of purity in many applications. The strict cascade of one-to-many relationships is applicable where the higher levels in a hierarchy cleanly partition the lower levels, such as counties within a state. However, a geographic dimension with zip codes, census tracts, and informal user-defined areas does not lead to a single hierarchy, but rather a branched structure consisting of multiple hierarchies. A hierarchy that does not fully partition a certain level can be made complete by simply creating an "other" category for those items unaccounted for by the original partitions, assuming that the categories are mutually exclusive. If the categories are not mutually exclusive, the hierarchy could be modeled using many-to-many relationships, with the potential for "double counting" in roll-up operations.

Dimension Hierarchy Diagrams

The design of dimensions is a critical part of data warehouse design. As discussed above, the design of dimensions determines the ultimate flexibility of the query environment. While pursuing the design of several data warehouses, the concept of a *dimension hierarchy diagram* (DHD) evolved as a useful modeling tool. Dimension hierarchy diagrams are simple tree structures, much like the biological representations used to relate organisms. For example, a DHD corresponding to Figure 2 is shown in Figure 3.

Figure 3 is a dimension hierarchy diagram that represents the hierarchical arrangement of the time dimension from the rainfall data example. The DHD notation uses circles and squares to represent tables connected by a solid line to represent the hierarchical structure. Tables corresponding to DAY, MONTH, SEASON, and YEAR are shown in Figure 3. The DAY table is represented as a square denoting a table that will be physically instantiated, whereas circles denote tables that will be denormalized. A set of dimension tables to be denormalized will collapse down the hierarchy until a physically instantiated table is reached. Branched structures represent multiple hierarchies where lower level attributes are grouped in different and incompatible ways. Figure 4 extends the previous example with new dimension tables for odd/even months and months that include the letter "r" (for those shellfish lovers). The location of the branch points is meant to convey a relative sense of the level of aggregation (the higher the coarser). Roll-up/drill-down operations make use of the hierarchical branches, with no direct mapping between separate branches (e.g., OE_MONTH and R_MONTH) being maintained, except through a shared lower level. The presence of a *subdimension* (i.e., support table) is represented as a horizontal line that divides a circle or square. In Figure 4, the DAY table has a subdimension to handle HOLIDAY details. A more complex design that includes subdimensions is often referred to as a *snowflake schema* rather than a star schema. The DHD is usually annotated with table names and example values where appropriate.

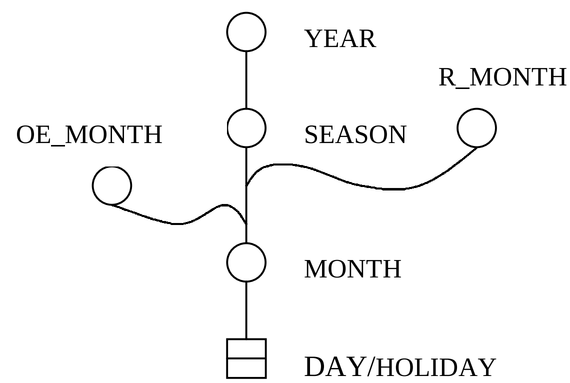

Figure 4: A Branched DHD

The use of tree structures in relational database design includes the representation of *is-a* and *part-of* hierarchies [Teorey 1994], as well as more informal diagrams used as shorthand. Dimension hierarchy diagrams are a formalization that reflect the particular needs of dimension design in the data warehouse environment, including conciseness for ongoing communication with end-users, an emphasis on denormalization (i.e., performance), and the depiction of relative levels of aggregation for roll-up/drill-down operations.

Conclusions

As we noted in the introductory section, data warehousing is a rapidly evolving technology that is finding widespread application in the business community. As a potential source of competitive advantage, custom design work can lead to decision support environments that accentuate the unique business processes. Central to these design activities is the need to represent the dimensions in a star schema or dimensional model. Clear representations are important both at the design stage and as ongoing documentation for end users.

Dimension hierarchy diagrams provide a concise notation for representing data warehouse dimensions, including hierarchical relationships, levels of granularity, denormalization, as well as subdimensions and snowflake schemas. This paper outlines the DHD notation and provides some examples. We are currently using dimension hierarchy diagrams in a large data warehousing development project in the healthcare environment [Berndt et al. 1998; Studnicki et al. 1998]. The diagrams are being used in design team meetings, as well as in larger team meetings and discussions with end users. This ongoing project will provide valuable experience with these design artifacts and provide the basis for future work in refining dimension hierarchy diagrams.

References

[Berndt et al. 1998] D. Berndt, A. Hevner, J. Studnicki, "CATCH/IT: A Data Warehouse to Support Comprehensive Assessment for Tracking Community Health," *Proceedings of the American Medical Informatics Association (AMIA) Annunal Symposium*, November 1998.

[Studnicki et al. 1997] J. Studnicki, B. Steverson, B. Myers, A. Hevner, D. Berndt, "Comprehensive Assessment for Tracking Community Health (CATCH)," *Best Practices and Benchmarking in Healthcare*, September/October: 196-207, 1997.

[Kimball et al. 1998] R. Kimball, *The Data Warehouse Lifecycle Toolkit*, John Wiley & Sons, New York, 1998.

[Kimball 1996] R. Kimball, *The Data Warehouse Toolkit*, John Wiley & Sons, New York, 1996.

[Teorey 1994] T. Teorey, *Database Modeling & Design*, Morgan Kaufmann, San Francisco, Second Edition, 1994

[Inmon 1992] W. Inmon, *Building the Data Warehouse*, Wiley, New York, 1992.

Acknowlegements

I would like to thank the anonymous reviewers for their insightful comments and suggestions. In particular, the comments on partial or incomplete hierarchies were both interesting and helpful.