ABSTRACT

While incomplete information is endemic to real-world data, current multidimensional data models are not engineered to manage incomplete information in base data, derived data, and dimensions. This chapter presents several strategies for managing incomplete information in multidimensional databases. Which strategy to use is dependent on the kind of incomplete information present, and also on where it occurs in the multidimensional database. A relatively simple strategy is to replace incomplete information with appropriate, complete information. The advantage of this strategy is that all multidimensional databases can manage complete information. Other strategies require more substantial changes to the multidimensional database. One strategy is to reflect the incompleteness in computed aggregates, which is possible only if the multidimensional database allows incomplete values in its hierarchies. Another strategy is to measure the amount of incompleteness in aggregated values by tallying how much uncertain information went into their production.
INTRODUCTION

Multidimensional databases are a relatively recent and popular phenomenon. A concise description of a multidimensional database is that it is a hierarchy of aggregate values. Values higher in the hierarchy are further aggregations of those lower in the hierarchy. The utility of the hierarchical organization is that the user can easily navigate among high and low precision views of the same aggregate data using drill-down and roll-up operations. Drill-down increases the precision of aggregate data being viewed while roll-up decreases the precision. For instance, suppose that a grocery store manager uses a multidimensional database to examine monthly sales for apples and notices that sales in January were low. To analyze the poor sales, the manager might drill-down to look at monthly sales by kind of apple, or she might roll-up to view sales for all fruits combined. Several vendors already have multidimensional database products on the market, either as add-ons to existing databases or as stand-alone tools, and a “cube” operator has been proposed for inclusion in future SQL standards (Gray et al., 1996, 1997).

A database model is only a small part of reality, but complete information about even this limited part is rare. For example, suppose that in an on-line grocery store, a customer fills out a web form to make a purchase. Some customers will invariably leave a portion of the form blank. Although a database could reject an incomplete purchase order, this would result in lower sales. A well-designed database should accept and store the data about an incomplete purchase, especially if the incomplete information is not essential to the purchase order. To cope with incomplete information that is present in real data, database management systems have evolved techniques for storing and reasoning about it. The most common technique in relational and object-oriented database management systems uses a null value to represent incomplete information. An ANSI report identified 14 possible interpretations of a null value in SQL, ranging from a value that is inapplicable, to one that is known to exist, but has an unknown value (ANSI/X3/SPARC, 1975).

A multidimensional database can also have incomplete information. The base data could be incomplete. For example, a multidimensional database for an on-line grocery store could store incomplete purchase orders. Incomplete information can also appear in the derived data. The derived data is typically an aggregate or summary view of the base data. For example, a count of incomplete purchase orders might result in an incomplete count. Finally, and somewhat surprisingly, it is not uncommon to have incomplete information in the dimensions. This is often the result of an incomplete specification. For example, suppose that a customer leaves the city field in a purchase order form blank because the customer lives on a farm. Customer orders per state will be undercounted if only customers in cities are counted.

The objective of this chapter is to present techniques for managing incomplete information in a multidimensional database. Although incomplete information is endemic to real-world data, this area has received less attention in the research community than other areas (Vassiliadis, 2000), and few commercial products have
Related Content

Conflicts, Compromises and Political Decisions: Methodological Challenges of Enterprise-Wide E-Business Architecture
[www.igi-global.com/article/conflicts-compromises-political-decisions/3380?camid=4v1a](www.igi-global.com/article/conflicts-compromises-political-decisions/3380?camid=4v1a)

Semantic Enrichment in Knowledge Repositories: Annotating Semantic Relationships Between Discussion Documents
[www.igi-global.com/article/semantic-enrichment-knowledge-repositories/3347?camid=4v1a](www.igi-global.com/article/semantic-enrichment-knowledge-repositories/3347?camid=4v1a)
An Overview of Fuzzy Approaches to Flexible Database Querying
Slawomir Zadrozny, Guy de Tré, Rita de Caluwe and Janusz Kacprzyk (2008).
Handbook of Research on Fuzzy Information Processing in Databases (pp. 34-54).
www.igi-global.com/chapter/overview-fuzzy-approaches-flexible-database/20348?camid=4v1a

Evaluating XML-Extended OLAP Queries Based on Physical Algebra
www.igi-global.com/article/evaluating-xml-extended-olap-queries/3354?camid=4v1a