Querying Multidimensional Data

Leonardo Tininini
Istituto di Analisi dei Sistemi e Informatica "Antonio Ruberti", Rome, Italy

INTRODUCTION

A powerful, easy-to-use querying environment is without doubt one of the most important components in a multidimensional database. Its effectiveness is influenced by many aspects, both logical (data model, integration, policy of view materialization, etc.) and physical (multidimensional or relational storage, indexes, etc.). Multidimensional querying is often based on the core concepts of multidimensional data modeling, namely the metaphor of the data cube and the concepts of facts, measures and dimensions (Agrawal, Gupta, & Sarawagi, 1997; Gyssens & Lakshmanan, 1997). In contrast to conventional transactional environments, multidimensional querying is often an exploratory process, performed by navigating along dimensions and measures, increasing/decreasing the level of detail and focusing on specific subparts of the cube that appear "promising" for the required information.

BACKGROUND

Multidimensional data are obtained by applying aggregations and statistical functions to elementary data, or more precisely to data groups, each containing a subset of the data and homogeneous with respect to a given set of attributes. For example, the data "Average duration of calls in 2003 by region and call plan" is obtained from the so-called fact table, which is usually the product of complex activities of source integration (Lenzerini, 2002) on the raw data corresponding to each phone call in that year. Several groups are defined, each consisting of calls made in the same region and with the same call plan, and finally applying the average aggregation function on the duration attribute of the data in each group. The pair of values (region, call plan) is used to identify each group and is associated with the corresponding average duration value. In multidimensional databases, the attributes used to group data define the dimensions, whereas the aggregate values define the measures of data.

The term multidimensional data comes from the well known metaphor of the data cube (Gray, Bosworth, Layman, & Pirahesh, 1996). For each of the n attributes, which are used to identify a single measure, a dimension of an n-dimensional space is considered. The possible values of the identifying attributes are mapped to points on the dimension's axis, and each point of this n-dimensional space is thus mapped to a single combination of the identifying attribute values and hence to a single aggregate value. The collection of all these points, along with all possible projections in lower dimensional spaces, constitutes the so-called data cube. In most cases, dimensions are structured in hierarchies, representing several granularity levels of the corresponding measures (Jagadish, Lakshmanan, & Srivastava, 1999). Hence, a time dimension can be organized into days, months and years; a territorial dimension into towns, regions and countries; a product dimension into brands, families and types. When querying multidimensional data, the user specifies the measures of interest and the level of detail of the information required by indicating the desired hierarchy level for each dimension. In a multidimensional environment, querying is often an exploratory process, where the user "moves" along the dimension hierarchies by increasing or reducing the granularity of displayed data. The operation of drill-down corresponds to an increase in detail, for example, by requesting the number of calls by region and month, starting from data on the number of calls by region or by region and year. Conversely, roll-up allows the user to view data at a coarser level of granularity (Cabibbo & Torlone, 1997).

OLTP VS. OLAP QUERIES

Multidimensional querying systems are commonly known as On-Line Analytical Processing (OLAP) systems (Li & Wang, 1996), in contrast to conventional On-Line Transactional Processing (OLTP) systems. The two types have several contrasting features, although sharing the same requirements on fast "on-line" response times.

- *Number of records involved.* One of the key differences between OLTP and multidimensional queries is the number of records required to calculate the answer. OLTP queries typically involve a rather limited number of records, accessed through primary key or other specific indexes, which need to be processed for short, isolated transactions or to be

issued on a user interface. In contrast, multidimensional queries usually require the classification and aggregation of a huge amount of data (Gupta, Harinarayan, & Quass, 1995).

- *Indexing techniques.* Transaction processing is mainly based on the access of a few records through primary key or other indexes on highly selective attribute combinations. Efficient access is easily achieved by well-known and established indexes, particularly B+-tree indexes. In contrast, multidimensional queries require a more articulated approach, as different techniques are required, and each index performs well only for some categories of queries (Chan & Ioannidis, 1998; Jürgens & Lenz, 1999).
- *Current state vs. historical DB's.* OLTP operations require up-to-date data. Simultaneous information access/update is a critical issue, and the database usually represents only the current state of the system. In OLAP systems, the data does not need to be the most recent available and should, in fact, be time-stamped, thus enabling the user to perform historical analyses with trend forecasts. However, the presence of this temporal dimension may cause problems in query formulation and processing, as schemes may evolve over time and conventional query languages are not adequate to cope with them (Vaisman & Mendelzon, 2001).
- *Target users.* Typical OLTP system users are clerks, and the types of query are rather limited and predictable. In contrast, multidimensional databases are usually the core of decision support systems, targeted at management level. Query types are only partly predictable and often require highly expressive (and complex) query language. However, the user usually has little experience even in "easy" query languages like basic SQL: the typical interaction paradigm is a spreadsheet-like environment based on iconic interfaces and the graphical metaphor of the multidimensional cube (Cabibbo & Torlone, 1998).
- *Dimensions and measures.* Early statistical database research has already shown (Shoshani & Wong, 1985) that the standard relational model and operators (commonly used to represent and query transactional databases) are inadequate for effective representation and querying of multidimensional data. This led to the distinction between category attributes (the dimensions) and summary attributes (the measures). The distinction between dimensions and measures is also at the basis of most models for OLAP systems. However, as noted by several authors, this distinction has some drawbacks, mainly because some operations easily expressible in relational algebra become cumbersome in multidimensional models. Some authors have proposed multidimensional models with a symmetrical treatment of measures and dimensions to cope with this problem (Agrawal, Gupta, & Sarawagi, 1997; Cabibbo & Torlone, 1997; Gyssens & Lakshmanan, 1997).

EXPRESSING MULTIDIMENSIONAL QUERIES

As noted earlier, the metaphor of the data cube and the concepts of facts, measures and dimensions are fundamental to both multidimensional data modeling and querying. In particular, techniques proposed in the literature and/or implemented in commercial systems to retrieve such data are based on the idea of determining the cube of interest and then navigating along the dimensions, increasing or decreasing the level of detail through roll-up and drill-down or selecting specific subparts of the cube through the operation of slice and dice.

The query languages for multidimensional data support both these standard operations and additional ones for performance of more sophisticated calculations. A first broad distinction can be made among:

- Languages based on an algebra (usually an extension of the relational algebra), where queries are expressed by using operators representing facts, measures and dimensions. Examples of these languages are the grouping algebra proposed by Li and Wang (1996) and the algebra for "symmetrical" cubes (Agrawal, Gupta, & Sarawagi, 1997).
- Languages based on a calculus (usually an extension of the relational calculus), where queries are expressed in a more declarative way. An example is MD-CAL, a multidimensional calculus for fact tables (Cabibbo & Torlone, 1997).
- Visual languages, usually relying on an underlying algebra, and based on a more interactive and iconic querying paradigm: This is the approach of most commercial OLAP products. A visual query language for statistical aggregate data was proposed by Rafanelli, Bezenchek, and Tininini (1996) and for the MD model by Cabibbo and Torlone (1998).

Multidimensional query languages can also be classified by the type of model used to represent the data:

- Query languages based on a relational representation of multidimensional data, hence based on extensions of the relational algebra and calculus.