Chapter VIII

Materialized Views in Multidimensional Databases

Stefano Paraboschi
Università di Bergamo, Italy

Giuseppe Sindoni
National Institute of Statistics, Italy

Elena Baralis
Politecnico di Torino, Italy

Ernest Teniente
Universitat Politècnica de Catalunya, Spain

ABSTRACT

This chapter presents materialized views in the context of multidimensional databases (MDDBs). A materialized view is a view whose content is explicitly stored in the database. The advantage of materializing views is that it is not necessary to recompute the query every time the view is accessed. The shortcoming is that it has to be kept consistent with the updates on the base tables. However, efficient incremental maintenance techniques have been proposed. MDDBs are an ideal environment for materialized views because frequency of updates is low, MDDB data models permit easy adoption of incremental maintenance, and queries can be modeled in such a way to allow an easy definition of the view selection problem, i.e., the problem of selecting which query to materialize in an MDDB. Hence, we present the problems of choosing and maintaining materialised views with the corresponding solutions.
VIEWS

Views are database objects whose content is derived from data stored in the database. A view \( V \) can be considered as the result of the computation of a function \( f \) on the database state \( DB \): \( V = f(DB) \). In relational databases a view has the same structure of a relation, as it is labeled with a name (unique in the database schema among all tables and views), it is characterized by a set of attributes, and its instance consists of a set of tuples; the only difference is that its content is obtained executing a query on the database, and it is not explicitly created by database users.

Example 1: View \( \text{TOTALSALES} \) has two attributes \( \text{Product} \) and \( \text{TotalAmount} \) and each tuple presents the sum of the amounts of all the sales for a particular product.

```sql
create view TotalSales(Product,TotalAmount) as
select Product, sum(Amount)
from Sales
group by Product
```

Views have an important role in databases, as they are at the core of many significant database services. Using views, a database schema can be adapted to the needs of the different users of the database. Data independence, one of the main advantages of databases (Atzeni et al., 1999), is implemented at the logical level by means of views; when the database schema evolves, views are used to emulate previous schema versions, making the application code independent of the actual relational schema.

MATERIALIZED VIEWS

Views in databases typically are virtual objects, i.e., their content is computed only when the database receives a request to access them. A typical strategy consists in rewriting queries that refer to views, in order to express them in terms of the concrete relations available on the system.

Example 2: A query like

```sql
select Product, TotalSales, Brand
from TotalSales, Products
where TotalSales.Product = Products.Code
```

might be transformed by the query optimizer into query:
Related Content

Issues in Mobile Electronic Commerce
www.irma-international.org/article/issues-mobile-electronic-commerce/3275/

Regression Test Selection for Database Applications
www.irma-international.org/chapter/regression-test-selection-database-applications/4358/

Entity-Relationship Modeling and Normalization Errors
www.irma-international.org/article/entity-relationship-modeling-normalization-errors/51172/

C-MICRA: A Tool for Clustering Microarray Data
www.irma-international.org/chapter/micra-tool-clustering-microarray-data/20742/

Data Warehousing Interoperability for the Extended Enterprise
Aristides Triantafillakis, Panagiotis Kanellis and Drakoulis Martakos (2004). *Journal of Database Management* (pp. 73-84).
www.irma-international.org/article/data-warehousing-interoperability-extended-enterprise/3317/