Implementation of the Multidimensional Modeling Concepts into Object-Relational Databases

Mihaela MUNTEAN, Bucharest, Romania, munteanm@inforec.ase.ro

A key to survival in the business world is being able to analyze, plan and react to changing business conditions as fast as possible. With multidimensional models the managers can explore information at different levels of granularity and the decision makers at all levels can quickly respond to changes in the business climate–the ultimate goal of business intelligence. This paper focuses on the implementation of the multidimensional concepts into object-relational databases.

Introduction

The multidimensional models view data as multidimensional cubes (hypercube) that are well suited for data analysis and most business people think about their business in multidimensional terms. Multidimensional modeling and the multidimensional models use concepts like: facts, dimensions, hierarchies, measures.

Facts are the items of interest for enterprise. Each *fact* has a *granularity* determined by the lowest level of dimensions. The *grain* of the fact is a very important characteristic and is the level of detail at which measurements are stored.

Dimensions are essential in multidimensional modeling because they characterize the *fact*- the subject that must be analyzed to understand its behavior. *Dimensions* present the context for analyzing the facts and has many attributes called *dimension attributes*. *Dimensions* have one or more *hierarchies* that are used for aggregating data. We can define multiple hierarchies for dimensions and multiple hierarchies can share one or more lowest levels. A typical *dimension* contains one or more hierarchies, together with other attributes that do not have a hierarchical relationship to any of the attributes in the dimension.

A *measure* represents the property of the *fact* that we want to analyze. *Measures* take on different values for various dimensions combinations. The *measures* may be atomic or derived, additive, no additive or semi-additive. *Additivity* is crucial to multidimensional data modeling. A *measure* is *additive* along a dimension if we can use the SUM operator to aggregate attribute values along all hierarchies defined on that dimension.

Combinations of dimension values define a cube's cells that are sparse or dense. If dimensionality of cube increases and granularity of dimensions become finer the cube become sparser. In this case we use a *multicube* (two or more cubes with one or more common dimensions). Figure 1 shows a *data cube* typically used for representing a *multidimensional model* - a cube for analyzing measures such as *quantity, price, revenue* or *quantity_sold* along the *Product, Store* and *Time* dimensions. The fact is *Sales* in a large store chain.

This multidimensional model can be easily transformed into a UML class diagram in which classes are related through associations and shared aggregation relationships. The relationship between fact and its dimensions is an *aggregation relationship*, where each part (component class) can be part of more than one whole (composed class) and the existence of a part is independent of any whole. The *Sales fact* is the composed class and its dimensions (*Product, Time, Store*) are its component classes. Figure 2. shows a part of the corresponding UML class diagram. Because the *Star schema* (Dimensional model/Kimball model) is widely accepted as the most viable data representation for multidimensional analysis in special in relational databases, the table 1 presents a comparison between UML class diagram and star schema regarding the multidimensional modeling concepts.

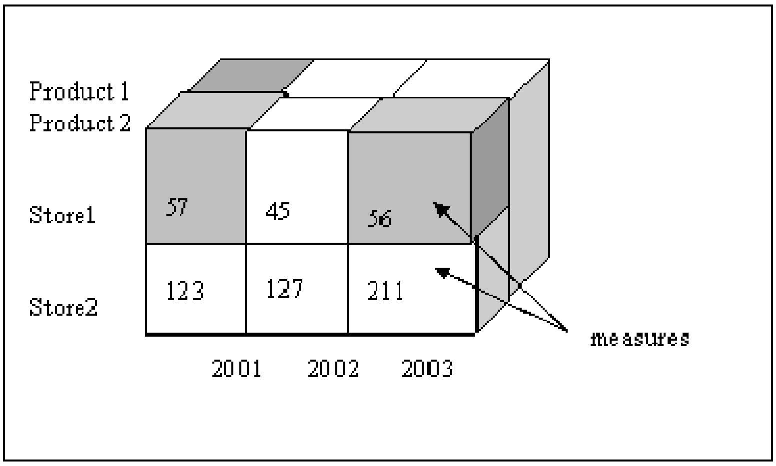

Figure 1. A data cube

Table 1. *UML class diagram versus Star schema*

Multidimensional modeling concepts	**UML class diagram**	**Star schema (Dimensional model)**
Fact	Fact class as composite class in a shared aggregation relationship of n dimensions classes. The minimum cardinality of dimension class is (1) because a fact class instance is always related to instances from all dimensions.	Fact table is the central table in a star schema characterized by a composite key, each of whose elements is a foreign key drawn from a dimension table.
Dimension	In UML class diagram a dimension is represented by a dimension class .	Dimension table contains attributes used to analyze the fact data. The dimension table is a table in a star schema with a single part primary key.
Measures (atomic/derived)	Attributes of fact class. The diagram contains derivation rules for derived attributes (using constraints).	Fact attributes. Schema don't specify derivation rules for derived attributes. The schema don't specify which attributes are atomic/derived.
Additivity	By default all measures are additive. Nonadditivity /semi-additivity by defining constraints on measures	The schema don't specify the additivity of measures.
Hierarchies	Multiple hierarchies using one-to-many association relationship. Heterogeneous dimensions using inheritance relationships.	The schema don't specify explicitly which are the hierarchies defined on dimensions.
Relationships between fact and dimensions (one-to-many or many-to-many)	Using aggregation	For handling many-to-many relationships we use some methods such as: the bridge table, denormalizing the dimension table by position flag attributes, lowering the grain of the fact table, etc.

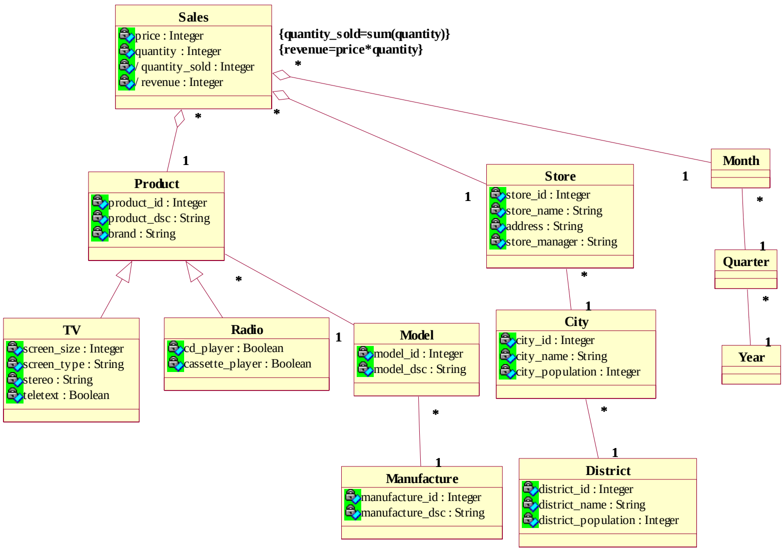

Figure 2. A part of UML class diagram

Object-relational representation

In *star schema Product dimension* is implemented as a *dimension table*. All attributes defining in the different hierarchy classes (UML class diagram) are defined as attributes within the same table –*Product table* (star schema).

We can translate a UML class diagram into an object-relational database schema (for the sake of simplicity, only *Product dimension*) using Oracle 10g. Using the object-oriented features of Oracle 10g, we can define object types (*manufacture_t, model_t, product_t, TV_t, radio_t*) and their tables. An object table is a special kind of table in which each row represents an object and it is derived from an object type. Each row is given a unique identity that behaves just like a OID. To implement subtypes, we can define object types as "*not final*" at the end of its type declaration. Oracle provides the keyword *under* to be used with the statement "*create type*" to create a subtype of a super type. To implement association relationships (one-to-many), we can use *object references:*

Create or replace type ***manufacture_t*** *as object*
(manufacture_id number, manufacture_dsc varchar2(50);
--implementation of on-to-many association using REF
Create or replace type ***model_t*** *as object*
(model_id number, model_dsc varchar2(50), model_manufacture REF manufacture_t);
Create or replace type ***product_t*** *as object*
(product_id number, product_dsc varchar2(50), brand varchar2(50),
Product_type varchar2(50), model_product REF model_t) not final;
--implementation of object tables
Create table ***manufacture of manufacture_t***
(manufacture_id primary key);
Create table ***model of model_t****(model_id primary key,*
foreign key(model_manufacture) REFERENCES manufacture on delete cascade);
create table ***product of product_t****(product_id primary key, product_type not null check(product_type in ('TV', 'Radio')), foreign key(model_product) references model on delete cascade;*
--implementation of inheritance relationship using "under"
create or replace type ***TV_t*** *under product_t*
(screen_size number, screen_type varchar2(20), stereo varchar2(20),
teletext varchar2(1));
create or replace type ***radio_t*** *under product_t(cd_player varchar2(1),*
cassette_player varchar2(1));

The relationship between fact and its dimensions is an aggregation relationship which can be represented in different ways: *cluster-*

ing technique, nesting technique. Table 2 provides some guidelines to translate a multidimensional model (its concepts) into an object-relational model (Oracle 10g object-relational model).

Table 2. Some guidelines to translate a multidimensional model into an object-relational model

Multidimensional modeling concepts	**Oracle 10g**
Fact/dimension	Object type and table of object type
Measures derived additivity	Attributes of object type Trigger/method Method
Hierarchies	REF/Nested table UNDER clause in the specification of each subtype indicating its super type
Relationships between fact and dimensions	Nesting technique, clustering technique (depending of aggregation type)

References

[1] Johanna Wenny Rahayu, David Taniar, Eric Pardede, *Object-Oriented Oracle*, Idea group Inc., 2006

[2] Juan Trujillo, Sergio Lujan-Mora, Il-Yeol Song, *Advanced Topics in Database Research*, edited by Keng Siau, Idea Group Inc., 2003