“Development of an interface that allows MDX based data warehouse queries by less experienced users”

Mariana Duprat
André Monat

Escola Superior de Desenho Industrial

Introduction

Data analysis is a fundamental resource when we are interested in collecting, organizing and comparing information. It gives us a more accurate vision of the situation in focus, allowing our decisions to be more consistent. As result, we achieve a better awareness of the context we are inserted in. Normally, it makes also possible to foresee the outcome of our immediate decisions. In brief, data analysis performs a pivotal role in modern management.

In order to be effective, proper information must be available for all ranks of a company. More specifically, strategic information must be available among high rank managers so they can take well based decisions. These high rank managers can not rely only on ordinary reports from the Information Technology (IT) sector of the companies. This type of managers frequently needs specific reports for specific decisions. Normally these reports take time to be built and are delivered only when they are no longer needed. This happens because professionals from the IT sectors of the companies are quite busy people. They are normally involved in solving daily problems. They can not stop their ordinary tasks to provide reports for strategic decision making. Unfortunately, management is a highly competitive process. There is a proper time for taking a decision and reports must be available when they are really needed. Otherwise they become useless.

For daily information, we rely on what is called OLTP (On Line Transactional Processing) Databases. These databases are not efficient in providing strategic reports. They are designed for queries concerning our routine operations. Besides, any query applied to this type of database must be expressed through a formal computing language (for instance SQL). Therefore, non IT experts can not have direct access to data stored in OLTP systems. This is the main reason to disregard OLTP systems as decision making tools.

OLAP Databases (On Line Analytical Processing) are quite different. They are designed in order to allow non IT experts to build the reports they need. OLAP Databases (and other OLAP systems) are able to store a gigantic amount of information based on historical data. Analyzing this data, managers are able to conclude about trends, market segments, client patterns among others possible conclusions.

In order to make possible analyses from non IT experts, OLAP databases, known as Data Warehouses, are built according to a quite intuitive architecture. This architecture involves a fact table and dimensions and is called star architecture or star schema. For instance, for a supermarket chain, the Food Mart example available for Microsoft OLAP systems.

The fact table stores numeric values henceforth called measures. These measures are numeric attributes of facts concerning the business involved. For instance, for a supermarket, the sell of each product could be regarded as a fact to be stored for further analysis and price, discount, weight and number of units its numeric attributes.

Dimension tables provide the relation between the fact and its features. In the example of a supermarket chain, we could regard Customer, Store, Product, Data and Time as dimensions for the fact we chose to store. These dimensions traditionally have attributes concerning the sell of a product. For instance, we can analyze the sell of food products (attribute of Product dimension) according to married and single customers (since gender is an attribute of the Customer dimension). This allows us to built reports as the one shown in Picture 2.

Analysis Manager

There are several tools for creating OLAP system according to a star schema. Microsoft makes available the Analysis Manager tool for using with her database systems created by Access or SQL Server systems. It is also possible to work with data imported from other formats and systems. This tool, combined with Excel or FrontPage, becomes an intuitive front end system for building reports from the data stored in the data warehouse.

Analysis Manager uses a formal language called MDX (multidimensional expression) for building reports. Most of time, the usage of this language is transparent to users since they may remain completely unaware about its existence. Nevertheless, when we want to use the full potential of Analysis Manager, MDX may become a necessary and visible tool. For instance, MDX is essential when we want to create new measures based in calculations over already existing measures. Some complex reports may need the usage of MDX too. This language has functions that are able to execute very useful operations, like the ordering of the members, the creation of new calculated measures based in hierarchies, the exhibit of data according to filters based on criteria defined by the user, among others.

Nevertheless, MDX is a formal computing language. The usage of such language requires an IT expert and therefore contradicts our previous wish of "*data and specific reports available to everybody*". The ideal result would be to combine the many possibilities offered by the MDX functions with the facilities offered by the interfaces of the familiar programs of Microsoft.

How complex Analysis Manager can be

Let's analyze a scenario where the usage of analysis manager may become hard for non-IT experts. Suppose we are interesting in obtaining the two top selling states from February to June of the year 1998 in a supermarket chain. In order to illustrate, we are going to use again the largely available Microsoft Foodmart 2000 OLAP database, and its cube "Sales". First, the software does not allow us to select only these two months. We can either select the whole level or one member of this level. Therefore, it is possible to select a specific Quarter or to view data concerning all months but not only the two selected months. In order to achieve such result, we should run the following MDX Query:

```
WITH MEMBER  [Measures].[Store Sales 1998 Feb-Jun] AS 'SUM([Time].[1998].[Q1].[2]:[Time].[1998].[Q2].[6],[Measures].[
Store Sales])'
SELECT
{[Store Sales 1998 Feb-Jun]} ON COLUMNS,
{[Store].[USA].Children} ON ROWS
FROM  [Sales]
```

Despite the Microsoft claims that MDX is intuitive and easy to understand we believe mangers have difficulties in dealing with such languages since even the ordinary macro languages, available in Excel spreadsheets, are known only by a minority of the users. Some Excel features, as pivot tables and reports, are known only by 10% of the users!

Using other programs, like Excel, for example, the selection of these specific members, months ranging from February to

June, would be quite simple. So, until this point we would not encounter much trouble, although we would have already left the Analysis Manager interface. However, when filtering the two top selling states, we would have no other choice. Excel does not have a tool to aid us either. Therefore, we would have to use an expression with even more complex functions, as shown bellow:

```
{TOPCOUNT ([State]. [USA].CHILDREN, 2,[Store Sales 1998 Feb-Jun]}
```

With this expression (TOPCOUNT) the two top selling states will be filtered. Now, putting the two expressions together:

```
WITH MEMBER  [Measures].[Store Sales 1998 Feb-Jun] AS 'SUM([Time].[1998].[Q1].[2]:[Time].[1998].[Q2].[6],[Measures].[
Store Sales])'
SELECT
{[Store Sales 1998 Feb-Jun]} ON COLUMNS,
{TOPCOUNT([Store].[USA].Children,2,[Store Sales 1998 Feb-Jun])} ON ROWS
FROM  [Sales]
```

We finally achieved the desired result; however, our old wish of "*data and specific reports available to everybody*" would not be fulfilled.

There are programs that work as an interface between the users and a complex data warehouse like the one we are analyzing. Nevertheless, in a scenario similar to the one shown above, the only option would be the use of the MDX language.

New software

Parameters

The existing programs that work with OLAP systems using MDX may seem complicated for less experienced users, limiting its resources. However, the simplification that is being presented is not based in the "translation" of all the existing functions nor of all possible queries in order to satisfy the needs of these users. A research of Microsoft reports was made in order to reduce the overall set of MDX functions to the most used. Common sense was also considered in the definition of this restricted set of MDX functions.

The development of a more user-friendly interface is intended to improve user understanding about queries. Using no formal language and more familiar terms, we intend to facilitate the construction of queries and reports through what Microsoft calls a "Wizard". A wizard may be understood as an interface which guides the users during the specification of a process, in our case a MDX query. Besides such "Wizards" we also specified an interface where the same queries may be easily carried. This interface holds a strong similarity with the Wizard's interface.

With all of the above in mind, simple tools that satisfy these requirements were developed. In order to facilitate the understanding of these tools, during its development, the MDX functions were separated in two categories: *visualization and new calculated measures*.

The *calculated measures* category was reduced to pre-formulated models of the top used measures (averages, percentages, sums and subtractions of values, etc.) This action was taken to allow users to make relevant calculations without the use of formulae. More complex formulas will be available, but only the ones that are necessary to fulfill the basic needs of these users.

The *visualization* category was afterwards subdivided into the following sub-categories: *visualization through dimensions* (selects the dimensions in each field), visualization by members (selects exactly which members and which parts of the hierarchy will be available), *visualization by criteria* (filtering the data by different criteria, e.g.: higher, lower, first, last, etc.) and *visualization models* (pre-formulated suggestions of visualizations for the user).

Interface

The developed interface consists on a "Wizard" with different tools available in each of its tabs. The first tab identifies the analyzed measures (which were renamed as VALUES for a better understanding). The second tab (GRID) selects the dimensions and the hierarchy levels that will be shown on the report axes (*visualization by dimensions*). The third tab (MEMBERS) will define exactly which members of each level should be selected (*visualization by members*). The fourth tab (RESTRICTIONS) has filters that hide, highlight or restrict the data based on criteria defined by the user (*visualization by criteria*). Another tab (NEW VALUES), still in development, will allow the user to create new calculated measures using the existing data, the available pre-generated formulae, and its junctions (*calculated measures*).

The RESTRICTION and NEW VALUES tabs are similar. Both are based on the "translation" of MDX formulae into simple terms. Using this method, the user will provide the data necessary for the formulae when filling out sentences that describe their needs.

The Wizard is available in a wider interface. This interface will provide the same tools, whilst also allowing the management of queries and reports.

When a grid is finalized, it is saved as a query inside the current file, allowing future comparisons and modifications. Graphics that illustrate these data will be available along with the correspondent grids. The information contained in each query can also be compared using intersection, subtraction and addition tools.

The developed program may not fulfill every user's necessity, but it will make most of its queries a lot easier.

Using the new interface

In order to illustrate how these new tools work, we are going to exemplify its usage with the same query concerning a report the two top selling states between the months of February and June in 1998.

We start selecting the first tab (VALUES) and select the analyzed measure -"Store Sales" (picture 3). On the second tab (GRID), we define the "Time" dimension (from the level "All" until the level "Month") on columns and the "Store" dimension (from the level "All Store" until the level "State") on lines. Filters are not necessary for this query. (picture 4)

Then we select the "Time" dimension on the third tab (MEMBERS) and follow its hierarchy until the "Year" level. Then we select the year of 1998, and click on the arrow to continue our trip down the hierarchy. Then we select the months of February (2) through June (6), holding down the "shift" key. (picture 5)

In order to view the sum of sales reffering to this period, it will be necessary to create a new calculated mesasures, so we will select the fifth tab (NEW VALUES). We will call this generated value "*Store Sales 1998 Feb-Jun*", by filling out the "*Name*" text box. On the "*Analysing...*" section we will select the "apparent data" option, which will work only on the already selected information. On the "*Calculate...*" section we select the "sum" option. This will give us the following sentence: the sum of (***value***) of (***members***) of the (***select dimension***) dimension. On the "*value*" dropdown box we will select the desired measure ("Store Sales"), on the "*members*" dropdown box we will select "apparent members", and, on the "*select dimension*" dropdown box we will select the "Time" dimension. This will give us the desired sum.

On the fourth tab we can restrict the apparent data to the two top selling states. On the "*show only*" section, we will select the "*x highest*" function (in the member column). This will give us the following sentence: *members with the* "__" highest (value). On the text box, we will select the amount of desired members ("2" in this case), and, on the (value) dropdown box, we will select the desired measure ("*Store Sales 1998 Feb-Jun*"). (picture 6)

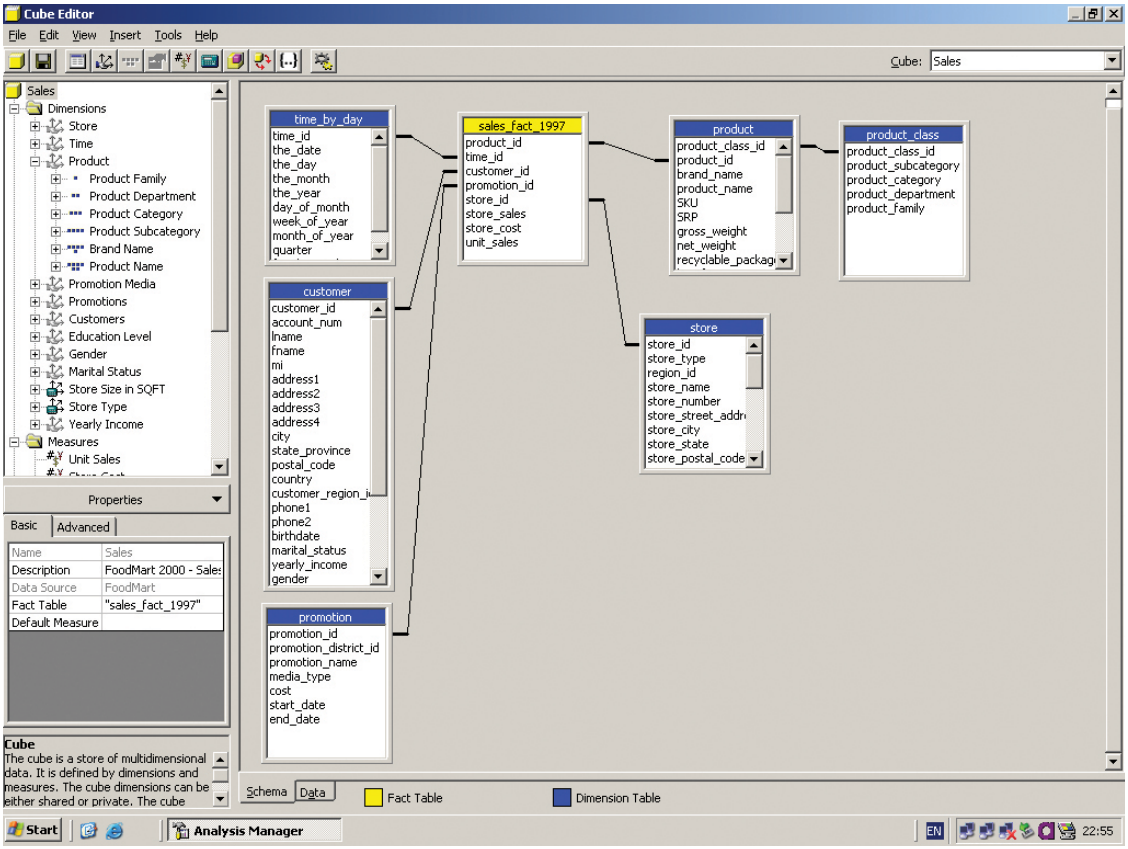
With these steps finished, we will get the desired grid, without writing any formula, and concluding that the states WA and CA are the top selling states in this period.

Conclusion

The main purpose of this work is to try to of avoid any contact between users and MDX formalism. This may become a hard task when some complex (but still common) functions are required, since those functions are not directly available on the main interface. This problem was solved by the usage of Wizards that provide these functions in an informal way. The software was broken down into different components, according to its tools. These components can be used separately, being integrated into other softwares that allow add-ins, like Excel, for example.

This software was programmed using the VB.Net language. The program can use other OLAP databases, but an OLAP system based in SQL Server 2005 was used.

Although some problems were solved by this interface, there is a lot of room for improvement and new solutions regarding interfacing OLAP systems. A new path to be traced is the development of interfaces with specific functions for each management area. The Human Resources (HR) field is already being explored on further enhancements of this interface. Basically, we are creating a specific interface for HR, where the most common calculations involving this area are "translated" into MDX by simple wizards. We believe that enterprise areas (like HR, Sales, Marketing, etc) have specific variables but also share some common calculations with the remaining areas



Name	Sales
Description	FoodMart 2000 - Sales
Data Source	FoodMart
Fact Table	"sales_fact_1997"
Default Measure	

Cube
The cube is a store of multidimensional data. It is defined by dimensions and measures. The cube dimensions can be either shared or private. The cube

time_by_day
time_id
the_date
the_day
the_month
the_year
day_of_month
week_of_year
month_of_year
quarter

sales_fact_1997
product_id
time_id
customer_id
promotion_id
store_id
store_sales
store_cost
unit_sales

product
product_class_id
product_id
brand_name
product_name
SKU
SRP
gross_weight
net_weight
recyclable_packag

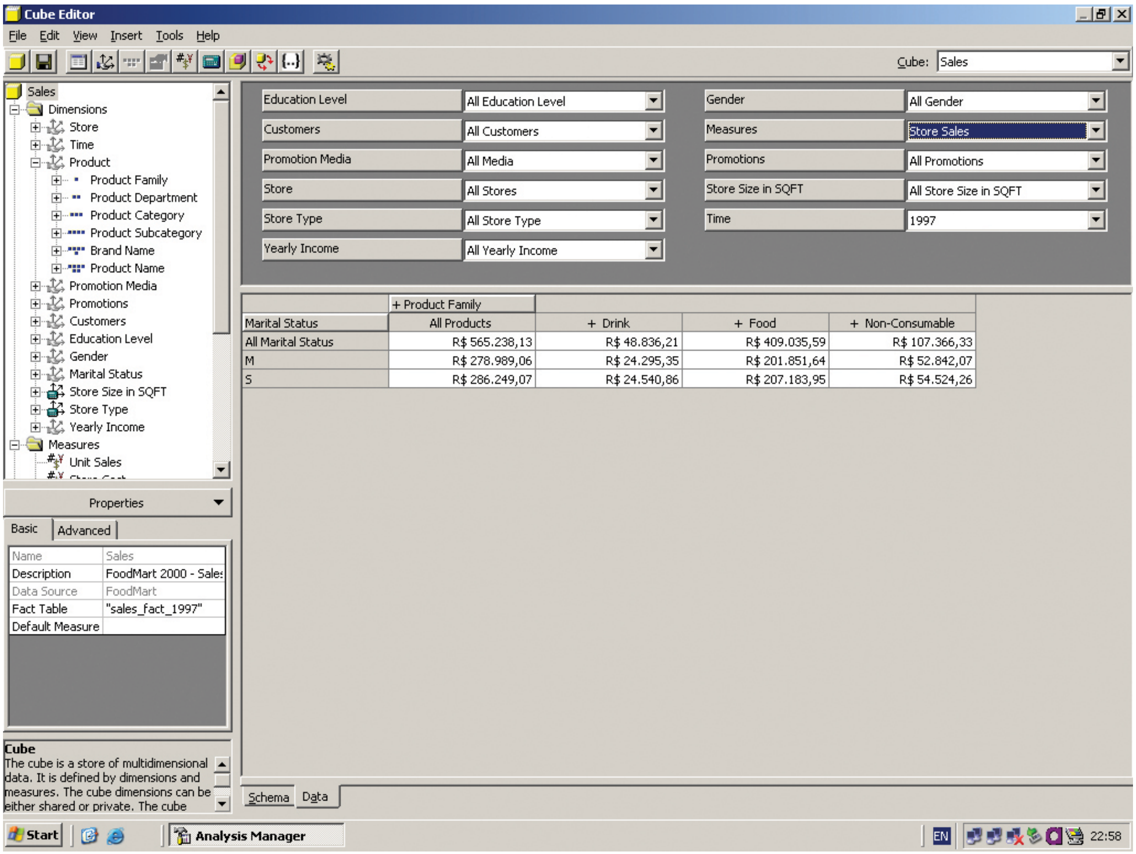
product_class
product_class_id
product_subcategory
product_category
product_department
product_family

customer
customer_id
account_num
lname
fname
mi
address1
address2
address3
address4
city
state_province
postal_code
country
customer_region_i
phone1
phone2
birthdate
marital_status
yearly_income
gender

store
store_id
store_type
region_id
store_name
store_number
store_street_addr
store_city
store_state
store_postal_code

promotion
promotion_id
promotion_district_id
promotion_name
media_type
cost
start_date
end_date

Picture 1



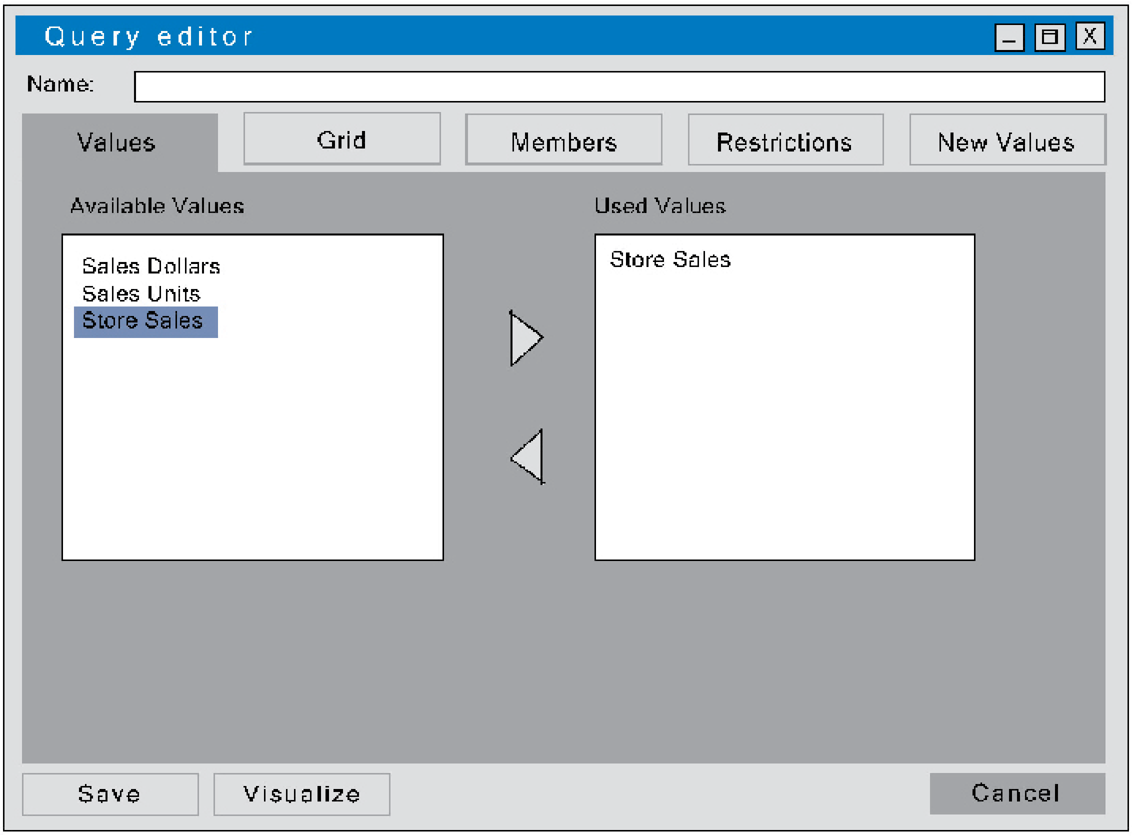
Name	Sales
Description	FoodMart 2000 - Sales
Data Source	FoodMart
Fact Table	"sales_fact_1997"
Default Measure	

Cube
The cube is a store of multidimensional data. It is defined by dimensions and measures. The cube dimensions can be either shared or private. The cube

Education Level	All Education Level
Customers	All Customers
Promotion Media	All Media
Store	All Stores
Store Type	All Store Type
Yearly Income	All Yearly Income
Gender	All Gender
Measures	Store Sales
Promotions	All Promotions
Store Size in SQFT	All Store Size in SQFT
Time	1997

	+ Product Family			
Marital Status	All Products	+ Drink	+ Food	+ Non-Consumable
All Marital Status	R$ 565.238,13	R$ 48.836,21	R$ 409.035,59	R$ 107.366,33
M	R$ 278.989,06	R$ 24.295,35	R$ 201.851,64	R$ 52.842,07
S	R$ 286.249,07	R$ 24.540,86	R$ 207.183,95	R$ 54.524,26

Picture 2



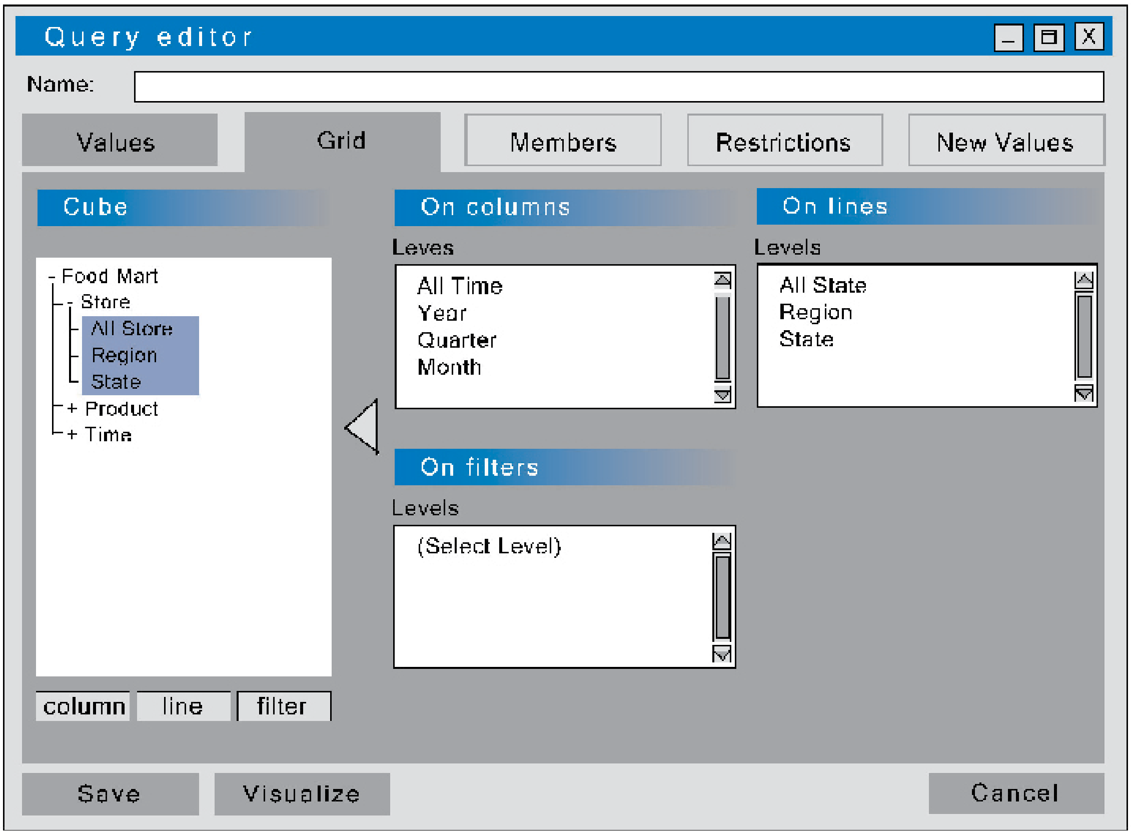
Available Values

Sales Dollars
Sales Units
Store Sales

Used Values

Store Sales

Picture 3



Cube

- Food Mart
  - Store
    - All Store
    - Region
    - State
  + Product
  + Time

On columns

Leves

All Time
Year
Quarter
Month

On lines

Levels

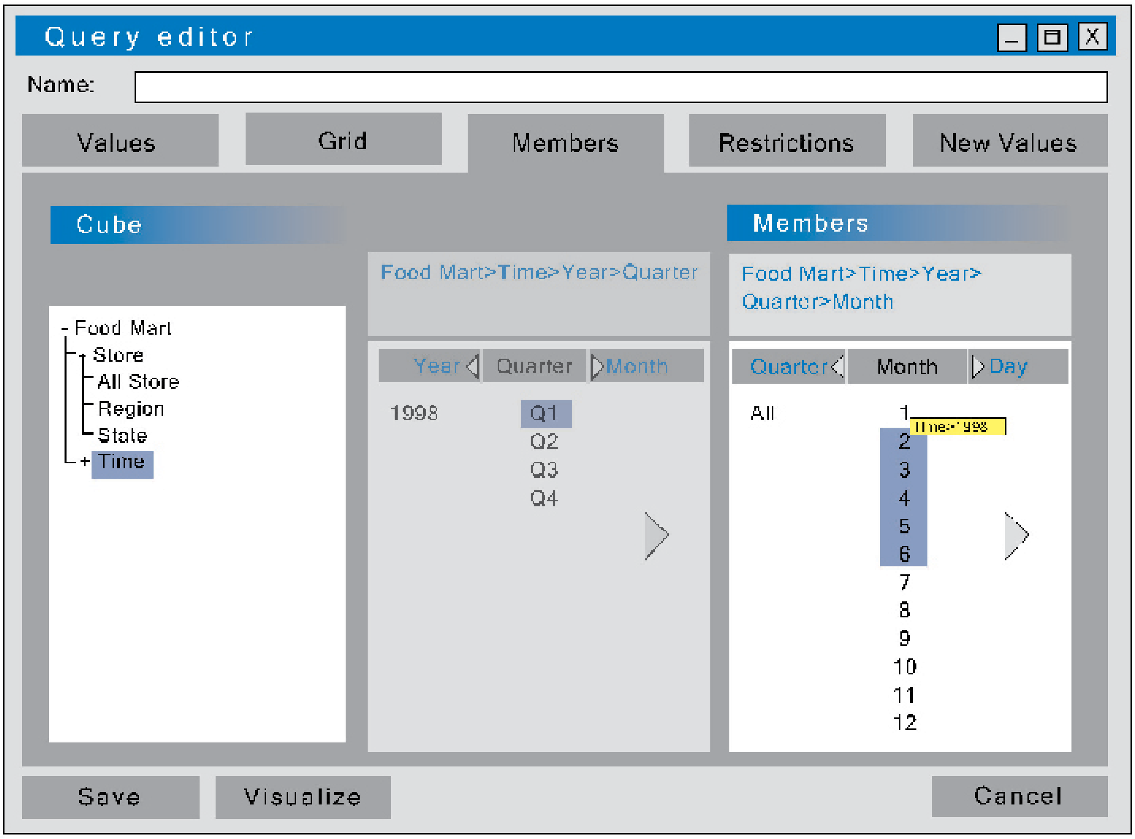
All State
Region
State

On filters

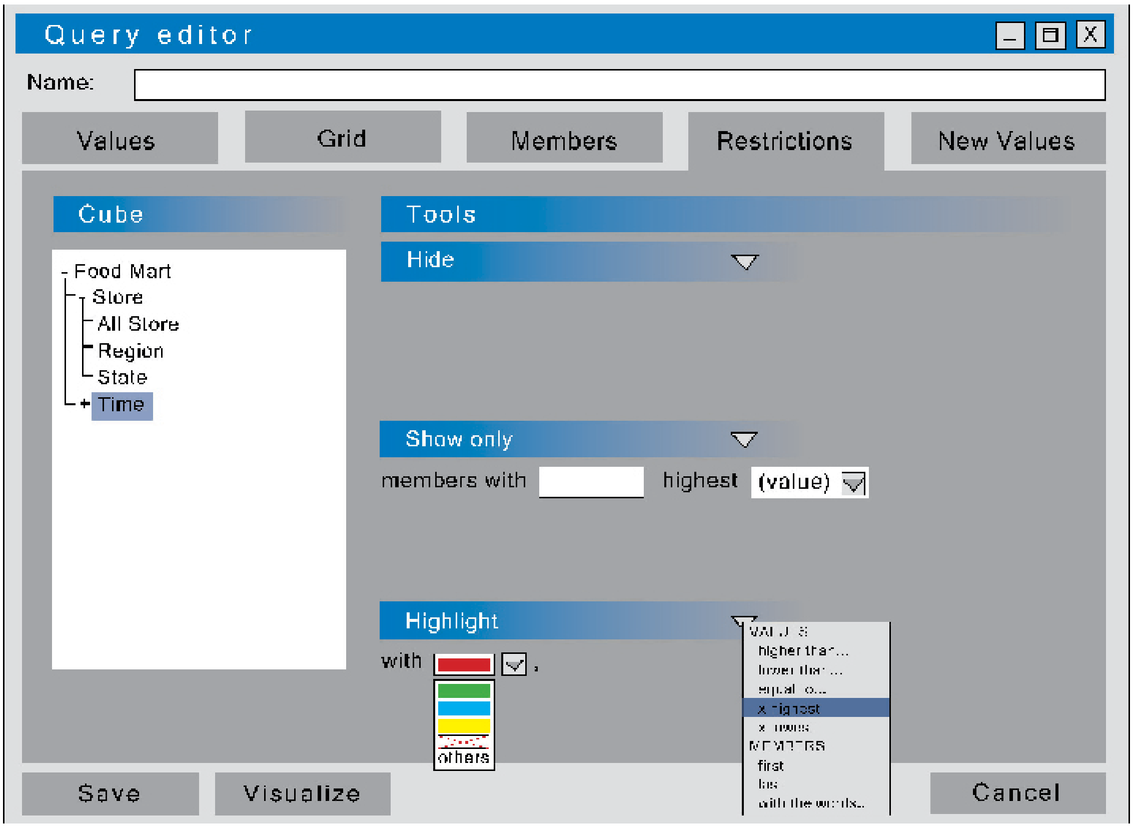
Levels

(Select Level)

Picture 4



Picture 5



Picture 6

Bibliography

WHITEHORN, Mark.; ZARE, Robert.; PASUMANSKY, Mosha., *Fast Track to MDX*. EUA: Springer, 2002.

JACOBSON, Reed., *Microsoft SQL Server 2005 Analysis Services*. EUA: Microsoft Press 2000.

SPOFFORD, George., *MDX Solutions*. EUA: John Wiley & Sons inc 2001.