WHAT-IF ANALYSIS IN OLAP
With a Case Study in Supermarket Sales Data

Emiel Caron[1] and Hennie Daniels[1,2]
[1] *Erasmus Research Institute of Management (ERIM), Erasmus University Rotterdam*
P.O. Box 90153, 3000 DR, Rotterdam, The Netherlands
[2] *Center for Economic Research, Tilburg University, P.O. Box 90153, 5000 LE, Tilburg, The Netherlands*

Keywords: Business intelligence, Multi-dimensional databases, OLAP, What-if analysis, Sensitivity analysis.

Abstract: Today's OnLine Analytical Processing (OLAP) or multi-dimensional databases have limited support for what-if or sensitivity analysis. What-if analysis is the analysis of how the variation in the output of a mathematical model can be assigned to different sources of variation in the model's input. This functionality would give the OLAP analyst the possibility to play with "What if ...?"-questions in an OLAP cube. For example, with questions of the form: "What happens to an aggregated value in the dimension hierarchy if I change the value of this data cell by so much?" These types of questions are, for example, important for managers that want to analyse the effect of changes in sales on a product's profitability in an OLAP supermarket sales cube. In this paper, we extend the functionality of the OLAP database with what-if analysis.

1 INTRODUCTION

An important and popular front-end application for business analysis and decision support is the *OLAP database*. OLAP databases are capable of capturing the structure of business data in the form of multi-dimensional tables which are known as *data cubes* by business information systems, as ERP systems. Manipulation and presentation of such information through interactive multi-dimensional tables and graphical displays provide invaluable support for the business decision-maker.

Currently, multi-dimensional business databases offer little support for what-if analysis. What-if analysis is defined as, the analysis of how the variation in the output of a mathematical model can be assigned to, qualitatively or quantitatively, to different sources of variation in the input of the model. Such analysis functionality would give the OLAP analyst the possibility to play with "What if...?"-questions. For example, with questions of the form: "What happens to an aggregated value in higher level cubes if I change the value of this data cell in this cube by so much?". Therefore, the central question in this paper is how OLAP database functionality can be extended with what-if analysis?

In this paper, we elaborate on a new operator that supports the analyst in answering these typical analysis questions in the OLAP database. Such an operator was first mentioned in (Caron and Daniels, 2008; Caron and Daniels, 2009), here we discuss it in more detail and apply it on a case study. For this purpose we introduce a novel notation for important concepts in OLAP databases, such as: dimensions, cells, cubes, navigational operators, lattices, upset, and additive measures. With these concepts we construct the what-if operator. An important issue for the application of this operator is that the OLAP database remains mathematically consistent during the analysis. Consistency in an OLAP database is not trivial because by changing a certain variable, the system of equations for some measure can become inconsistent. It is therefore important to discuss the conditions for consistency and solvability in OLAP databases.

This research is part of our continued work on extensions for the OLAP framework for business diagnosis. Current OLAP databases have limited capabilities for sensitivity, diagnostic, and outlier analysis. The goal of our research is to largely automate these manual diagnostic discovery processes (Caron and Daniels, 2007; Daniels and Caron, 2009). In (Sarawagi et al., 1998) and (Cariou et al., 2008) similar research approaches are taken.

The remainder of this paper is organized as follows. Section 2 introduces our notation for OLAP database concepts, followed by a definition of addi-

Caron E. and Daniels H. (2010).
WHAT-IF ANALYSIS IN OLAP - With a Case Study in Supermarket Sales Data.
In *Proceedings of the 12th International Conference on Enterprise Information Systems - Databases and Information Systems Integration*, pages 208-213
DOI: 10.5220/0002897202080213

tive measures in Section 3. In Section 4, we show that this definition together with a commutativity property ensures that the system of additive equations, represented in an OLAP database, is uniquely solvable. This is the basis for what-if analysis. In Section 5 we briefly discuss our prototype software implementation. Subsequently, we apply what-if analysis on a case study in supermarket sales data, with the software in Section 6. This case study we also use in the other sections as a running example. Finally, conclusions are discussed in Section 7. In the Appendix two figures from the case study are depicted.

2 OLAP DATABASES

2.1 Dimensions and Dimension Hierarchies

The basic unit of interest in the multi-dimensional database are numerical measures, representing countable information (Lenz and Shoshani, 1997) concerning a business process. A measure can be analysed from different categorical perspectives, which are the dimensions of the multi-dimensional data. In our notation dimensions are represented by $D_1^{i_1}, D_2^{i_2}, \ldots, D_n^{i_n}$, where each domain $D_k^{i_k}$ represents a dimension, e.g. Time, Store, Customer and so on, from the associated business process. Each domain corresponds with a dimension table in the star scheme. A domain consists of a set of *dimension levels* $i_k \in \{0, 1, \ldots, \max\}$. For example, the Time dimension might have the following levels: Day, Week, Month, Quarter, Season, and Year. The aggregation levels are organised in multiple *dimension hierarchies* or dimension paths. Thus, each domain $D_k^{i_k}$ has a number of hierarchies ordered by:

$$D_k^0 \prec D_k^1 \prec \ldots \prec D_k^{i_{\max_k}}, \qquad (1)$$

where D_k^0 is the lowest level and $D_k^{i_{\max}}$ is the highest level in D_k. Moreover, each level in the hierarchy $D_k^{i_k}$ has a unique categoric label $A_k^{i_k}$ corresponding with a column name from the dimension table.

A single instance of a dimension level $D_k^{i_k}$ is denoted by $d_k^{i_k}$, where $d_k^{i_k} \in D_k^{i_k}$. The total number of instance in $D_k^{i_k}$ is denoted by $|D_k^{i_k}|$. For example, for the Time dimension $D_k = T$ we could have the following labelled hierarchy schema: T[Month]$\prec$ T[Quarter] $\prec$ T[Year] $\prec$ T[All-Times] or in short $\mathrm{T}^0 \prec \mathrm{T}^1 \prec \mathrm{T}^2 \prec \mathrm{T}^3$, where the level instances at level 0 are $\mathrm{T}^0 = \{$1999.Q1.Jan, 1999.Q1.Feb, 1999.Q1.Mar, $\ldots\}$, at level 1 are $\mathrm{T}^1 = \{$1999.Q1, 1999.Q2, 1999.Q3, 1999.Q4, $\ldots\}$, at level 2 are $\mathrm{T}^2 = \{$1999, 2000, $\ldots\}$, and $\mathrm{T}^3 = \{$*All-Times*$\}$. An example of the instantiated dimension hierarchy is 1999.Q1.Jan $\prec$ 1999.Q1 $\prec$ 1999 $\prec$ *All-Times*, where 1999.Q1.Jan $\in T^0$, 1999.Q1 $\in T^1$, 1999 $\in T^2$, and *All-Times* $\in T^3$. In addition, the top level of a dimension always has a single level instance $D_k^{i_{\max_k}} = \{\text{All-}D_k\}$, thus $|D_k^{i_{\max_k}}| = 1$. The schema representation belonging to the hierarchy of the Time dimension is depicted in Figure 1.

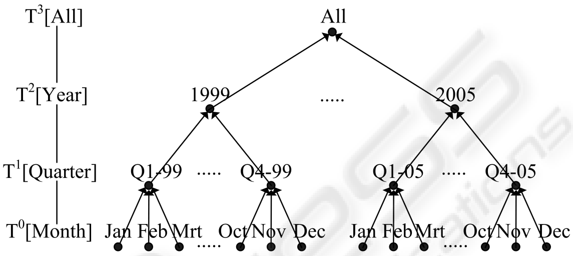


Figure 1: The left-side represents the hierarchy schema of the Time dimension; the right-hand side represents the rooted tree of its dimension hierarchy instances.

With each dimension hierarchy in domain D_k a *rooted tree* $T(D_k) = (V, E)$ is associated, called the *dimension hierarchy tree* of D_k, depicted in Figure 1 right.

The instance element $d_k^{i_k+1} \in D_k^{i_k+1}$ is called a *parent* and $d_k^{i_k} \in D_k^{i_k}$ is called its *child*. In the tree the year 1999 is the parent of the children {1999.Q1, 1999.Q2, 1999.Q3, 1999.Q4}, for example. To determine the parent of some child element in the hierarchy of a single domain $D_k^{i_k}$, we define a *1-dimensional roll-up operator* as:

$$r^{+1}(d_k^{i_k}) = d_k^{i_k+1}, \qquad (2)$$

and reversely, to determine the children of some parent element in the hierarchy, a *1-dimensional drill-down operator* is defined as:

$$r^{-1}(d_k^{i_k}) = d_k^{i_k-1}. \qquad (3)$$

These operators r^{+1} and r^{-1} can also be applied on any subset $X_k^{i_k}$ of $D_k^{i_k}$, and the operators can be applied both on the schema as on the instance level. For example, on the schema level as $r^{-1}(T^2[\text{Year}]) = T^1[\text{Quarter}]$, or on the instance level as $r^{-1}(1999)$, to determine the children of some specific year.

2.2 Cubes and Cells

The key structure in the multi-dimensional database is the data cube. A *cube* or a *sub cube C* is defined as the Cartesian product over the levels of the available domains:

$$X_1^{i_1} \times X_2^{i_2} \times \ldots \times X_n^{i_n}, \text{ where } X_k^{i_k} \subseteq D_k^{ik}. \qquad (4)$$

For example, Time^1 × Store Region^1 × Product^3, $T^1 \times S^1 \times P^3$, and so on, are cubes in the case study. Note that according to this definition also a single dimension hierarchy is composed out of cubes, e.g. the left hand side of Figure 1 shows the cubes that make up the Time dimension. An alternative representation of a full cube is given by $(D_1^{i_1}, D_2^{i_2}, \ldots, D_n^{i_n})$, or as $[i_1, i_2, \ldots, i_n]$ in shorthand notation.

A *cell* is defined as an instance element of a cube $X_1^{i_1} \times X_2^{i_2} \times \ldots \times X_n^{i_n}$:

$$(d_1, d_2, \ldots, d_n), \tag{5}$$

where $d_1 \in X_1^{i_1}$, $d_2 \in X_2^{i_2}$, …, $d_n \in X_n^{i_n}$. Accordingly, a cell contains a single instance value for each of its domains. For example, (2000.Q1, Vancouver, Food) is a cell in the example cube Time^1 × Store Region^1 × Product^3.

In addition, the instances at the lowest dimension levels of each of its domains $[0, 0, \ldots, 0]$ are cells of the *base cube* $D_1^0 \times D_2^0 \times \ldots \times D_n^0$, labelled as C_B. For example, in the financial database the base cube is represented by Time^0 × Store Region^0 × Product^0. The base cube can be aggregated to a higher hierarchical level in the domain by applying roll-up operators (see Section 2.3). When all dimension hierarchies are aggregated at the highest level, we derive the 0-dimensional *top cube* $D_1^{i_{\max_1}} \times D_2^{i_{\max_2}} \times \ldots \times D_n^{i_{\max_n}}$, labelled as C_T. The top cube consists of only one cell $(All, All, \ldots, All)$.

2.3 Navigational Operators

A number of navigational operations are available to the business analyst to manual explore OLAP cubes, allowing interactive querying and analysis of the data. In this paper, we redefine these navigational operations on cubes, i.e. on multiple dimensions, in our notation. The operations are defined on the domains of the cube C as in equation (4). Thus, these operations also hold for full cubes where $X_q^{i_q} = D_q^{i_q}$ and cells where $X_q^{i_q} = d_q^{i_q}$, since these are simply special cases of the sub cube. The most important navigational operations or queries for cubes are:

Drill-down, which de-aggregates a cube to a lower dimension level, is defined as:

$$\begin{gathered} R_q^{-1}(X_1^{i_1} \times \ldots \times X_q^{i_q} \times \ldots \times X_n^{i_n}) = \\ X_1^{i_1} \times \ldots \times r^{-1}(X_q^{i_q}) \times \ldots \times X_n^{i_n}. \end{gathered} \tag{6}$$

For example, a drill-down operation on the Time hierarchy from the level Year to the level Quarter, in the example full cube R_{Time}^{-1}(Time^2 × Store Region^3 × Product^2) results in the full cube Time^1 × Store Region^3 × Product^2.

Roll-up, the reverse of drill-down, which aggregates a cube along one or more dimension hierarchies to a higher dimension level, is defined as:

$$\begin{gathered} R_q^{+1}(X_1^{i_1} \times \ldots \times X_q^{i_q} \times \ldots \times X_n^{i_n}) = \\ X_1^{i_1} \times \ldots \times r^{+1}(X_q^{i_q}) \times \ldots \times X_n^{i_n}. \end{gathered} \tag{7}$$

Obviously, drill-down and roll-up (6) and (7) are the inverse of each other: $R_q^{+1}(R_q^{-1}(C)) = R_q^{-1}(R_q^{+1}(C)) = C$.

We refer to (Han and Kamber, 2005) for an elaborate overview on navigational operators.

2.4 Aggregation Lattice

By rolling-up the full base cube $D_1^0 \times D_2^0 \times \ldots \times D_n^0$, or one of its sub cubes $X_1^0 \times X_2^0 \times \ldots \times X_n^0$, over several associated dimensions and dimension hierarchies, in any order, a *lattice of cubes L* is formed. This lattice L has at the bottom the base cube $[0, 0, \ldots, 0]$ and at the top the cube $[i_1, i_2, \ldots, i_n]$, and is defined by the following sequence of operations, applied to the base cube:

$$\begin{gathered} R_1^{+i_1} \circ R_2^{+i_2} \circ \ldots \circ R_n^{+i_n}(D_1^0 \times D_2^0 \times \ldots \times D_n^0) = \\ D_1^{i_1} \times D_2^{i_2} \times \ldots \times D_n^{i_n}, \end{gathered} \tag{8}$$

where $R_q^{+n} = R_q^{+1} \circ \ldots \circ R_q^{+1}$. The complete lattice has $[i_{\max_1}, i_{\max_2}, \ldots, i_{\max_n}]$ at the top. Note that by *commutativity of the roll-up operators* the different orders of application yield the same result. Moreover, as a result of its definition, the lattice structure holds for full cubes, sub cubes, and cells, that might be derived from the multi-dimensional database.

With the concept of the aggregation lattice, we define the parents and children of a cube C. A *parent cube* C' in L is defined as the result of the roll-up operation $R_q^{+1}(C) = C'$. Obviously, a parent cube might have multiple child cubes, for example, the parent cube $[i_1, i_2, \ldots, i_n]$ has $[i_1 - 1, i_2, \ldots, i_n]$, $[i_1, i_2 - 1, \ldots, i_n]$, …, $[i_1, i_2, \ldots, i_n - 1]$ as its child cubes. Reversely, to determine all the child cubes of cube C in the lattice, we have to apply a drill-down operation on all its associated domains. Obviously, due to the lattice structure, a child cube usually has multiple parents, for example, the child cube $[i_1, i_2, \ldots, i_n]$, has $[i_1 + 1, i_2, \ldots, i_n]$, $[i_1, i_2 + 1, \ldots, i_n]$, …, $[i_1, i_2, \ldots, i_n + 1]$ as is parent cubes, corresponding to the different roll-ups. In addition, in the lattice the partial ordering within the dimension hierarchies is preserved.

We define, the *upset of a cube* $C = [i_1, i_2, \ldots, i_n]$ is the lattice of all ancestors of the cube C. The upset of a cube C is a sub lattice of the complete lattice, with the base cube C and top $[i_{\max_1}, i_{\max_2}, \ldots, i_{\max_n}]$. It is obtained by applying roll-up operations on the cube C repeatedly, in any order. The *downset of a cube* $C = [i_1, i_2, \dots, i_n]$ is the lattice of all descendants of the cube C. The downset of a cube $C = [i_1, i_2, \dots, i_n]$ is the sub lattice of the complete lattice with base cube $[0, 0, \dots, 0]$ and top C. It is obtained by applying drill-down operations on the the cube C repeatedly, in any order. The upset and downset of a single cell are defined similarly.

An *analysis path P* is defined as a sequence of p drill-down (roll-up) operators, as defined in equation (7), executed over the cubes of the lattice. The length of a path from the cube $[i_1, i_2, \dots, i_n]$ somewhere in the lattice to the base cube $[0, 0, \dots, 0]$ is $i_1 + i_2 + \dots + i_n$. Obviously, the sum of the indices of a cube corresponds with the number of aggregations carried out, i.e. the level of L under consideration.

2.5 Measures

Measures are derived from the column names of the fact table, and its *measure values*. The instances of the measures, are entries of the fact table. A *measure* y is defined as a function on a cube C:

$$y^{i_1 i_2 \dots i_n} : D_1^{i_1} \times D_2^{i_2} \times \dots \times D_n^{i_n} \to \mathbb{X}. \qquad (9)$$

where measure values are, for example, $\mathbb{X} = \mathbb{N}$, $\mathbb{Z}$, or $\mathbb{R}$. We also use the word *variable* instead of measure.

Data are the values of a measure y in a particular cell like, for example, sales^{232}(2000, Canada, Food)$= 70,028$. The combination of a cell and a measure is called a *data point*. The measure's upper index indicates the level of the cell on the associated dimension hierarchies. Furthermore, if a measure is not defined for a particular cell, we call the cell an *empty cell*.

If a measure is related to the base cube $[0, 0, \dots, 0]$, then the dimension hierarchies of the domains can be used to aggregate the measure values of $y^{00\dots0}(d_1, d_2, \dots, d_n)$ by typical *aggregation functions* like SUM(), COUNT(), MAX(), MIN(), or AVG().

3 ADDITIVE MEASURES

The measure $y^{i_1 \dots i_q \dots i_n}$ is defined as a *additive measure*, in the terminology of (Lenz and Shoshani, 1997), if for each cube C in the lattice $L(y)$, except the base cube, the following holds:

$$y^{i_1 \dots i_q \dots i_n}(C) = \sum_q y^{i_1 \dots (i_q - 1) \dots i_n}(R_q^{-1}(C)). \qquad (10)$$

Equation (10) also holds for all individual cells in the cube.

From our case study database, we could inspect the measure sales as a function on the sub cube C, given by 2000 × Store Country × Product Family. This cube is part of the lattice L(sales), formed by rolling-up with the SUM() aggregation function. By applying equation (10) two times we get:

$$\text{sales}^{242}(C) = \sum_k \text{sales}^{142}(R^{-1}_{\text{Year}}(C)) =$$
$$\sum_l \sum_k \text{sales}^{132}(R^{-1}_{\text{Store Region}}(R^{-1}_{\text{Year}}(C))).$$

For the cell (2000, All-Regions, Food) in C, an instantiated equation corresponding to the above drill-down operations reads:

$$\text{sales}^{242}(2000, \text{All-Regions}, \text{Food}) =$$
$$\sum_{j=1}^{4} \sum_{k=1}^{3} \text{sales}^{132}$$
$$(2000.\text{Quarter}_j, \text{Store Country}_k, \text{Food}),$$

where $S^j(2000.\text{Quarter}) = 2000.\text{Quarter}_j$. Furthermore, the additive COUNT() function is defined similarly, and treated as a special case of the SUM() function, only this operator summarizes dimension hierarchy instances instead of measure instances.

4 WHAT-IF ANALYSIS

Now we want to investigate the influence of a change in a measure value of a cell, in any cube on a higher level value of the same measure. Or in formal notation, what would be the change in $y^{j_1 j_2 \dots j_n}(d_1^{j_1}, d_2^{j_2}, \dots, d_n^{j_n})$ if the measure value $y^{i_1 i_2 \dots i_n}(d_1^{i_1}, d_2^{i_2}, \dots, d_n^{i_n})$ in a cube at a lower level is changed by the amount δ, keeping all other measure values in the cube $[i_1, i_2, \dots, i_n]$ unchanged. To solve this we consider the lattice L of cubes with base cube $[i_1, i_2, \dots, i_n]$ and top cube $[j_1, j_2, \dots, j_n]$. The values of the measure y in the cube $[i_1, i_2, \dots, i_n]$ are denoted by $x(d_1, d_2, \dots, d_n)$ and $y(d_1, d_2, \dots, d_n)$ in the higher level cubes of $[j_1, j_2, \dots, j_n]$. We distinguish between the original values of a measure without change x^r and y^r, and the values of the changed measure: x^a and y^a, where $x^a = x^r$ except for one cell c_0 in the base cube, and $\delta = x^a - x^r$.

Theorem 1. There is a unique additive measure y^a defined on all cubes in the lattice L such that:

$$y^a(c) = y^r(c) + \beta(c) \cdot (x^a - x^r), \qquad (11)$$

where:

$\beta(c) = 1$ if c_0 is a descendant of c, and
$\beta(c) = 0$ if c_0 is not a descendant of c.

Proof. To show that y^a is additive it is sufficient to show that $\beta(c)\cdot(x^a - x^r)$ is additive, because the sum of additive measures is also additive and y^r is additive. Thus, we must show that:

$$\beta(c) = \sum_q \beta(R_q^{-1}(c)), \tag{12}$$

where R_q^{-1} is a drill-down operator defined on a cube or cell in the lattice L. Now there are two cases:

case 1) c_0 is a descendant of c. In that case c_0 is also a descendant of $R_q^{-1}(c)$, which are the children of c in direction q. This property does not depend on q. So both sides of equation (12) are equal to 1.

case 2) c_0 is not a descent of c. But in that case, it also not a descendant of the children of c. Hence, both sides of equation (12) are zero.

It is important to note that, for any cell c_0 in the base cube of a lattice, that if cell c_0 is a descendant of $R_{q_1}^{-1}(c)$ then it is also a descendent of $R_{q_2}^{-1}(c)$.

Note that the measure y^a is unique. This follows from the general proposition that every additive measure with given values on the base cube is unique. To show this, now suppose that we have a (sub) lattice L with top cube $C = [j_1, j_2, \ldots, j_n]$ and base cube $C' = [i_1, i_2, \ldots, i_n]$. In the lattice the drill-down operators are commutative, different orders of application, over all analysis paths from top to base, give the same cube. Or stated formally:

$$\begin{aligned} &R_1^{-j_1} \circ R_2^{-j_2} \circ \ldots \circ R_n^{-j_n}(C) = \\ &R_n^{-j_n} \circ \ldots \circ R_2^{-j_2} \circ R_1^{-j_1}(C) = \quad C'. \end{aligned} \tag{13}$$

This property together with equation (10) ensures that the system of additive equations, represented in L has a unique solution if the measure is defined on the base cube.

Now the what-if analysis can be carried out as follows. If a base cell $x^a(d_1, d_2, \ldots, d_n)$ is changed with some δ, all elements in its upset are changed with that δ. This follows immediately from Theorem 1.

5 SOFTWARE IMPLEMENTATION

Our prototype software is implemented in MS Excel/Access in combination with Visual Basic, see (Caron and Daniels, 2009) for a detailed overview of the core functionality. For the implementation of the ideas in this paper we build on this functionality. The software connects with an OLAP database in MS Access. In MS Excel a cube can be constructed from this database and inspected via a pivot table. In such a pivot table, the analyst can do what-if analysis on a specific cell, by selecting the cell and pushing the analysis button. Now the analyst can decide to change the cell with some percentage or absolute value, see Figure 2. The result will be that the original cell value and its upset are changed with that value. In the software all changed cells are indicated with a color, see Figure 3. Next the analyst might decide to do a new analysis and build on the previous one, or he might undo his action to return to the original pivot table. After some actions the analyst can always return to the original situation because all operations are executed on a (virtual) copy of the OLAP database. Obviously, in the software only the modified cell, in some cube in the lattice, and its changed upset need to be stored for a single analysis.

6 CASE STUDY: SUPERMARKET SALES DATA

In this section we apply our prototype analysis software on an artificial but realistic supermarket sales data set, modelled as a star scheme. The data set has $164,558$ records in the sales fact table for the year 2000 for supermarkets in North America, with measures as sales, costs, revenues, and so on. Typical dimensions with hierarchies in the data set are: Time (see Section 2.1 for the hierarchy), Store Region (with hierarchy: Store Name $\prec$ Store City $\prec$ Store Region $\prec$ Store Country), Product (with hierarchy: Product Name $\prec$ Product Sub-Category $\prec$ Product Category $\prec$ Product Family), etc.

In the data set we have the lattice with base cube Month × Store Name × Product Name or $[0,0,0]$ and with top cube All-Times × All-Stores × All-Products or $[3,4,4]$. Now suppose we are inspecting the cube Year.Quarter × Country.City × Product Family or [1,1,3], with slices on Year = 2000 and Product Family = 'Food' with the additive measure store sales. In the cube we want analyse the impact of a 10% increase in the cell store sales(2000.Q1, Mexico.Acapulco, Food)= $10,820.89$ on its upset cells in [2,1,3], [1,2,3], [2,2,3], and [2,4,3], see Figure 2 in the Appendix, where a number screenshots are depicted from our software.

The result of this analysis is depicted in Figure 3, where the colored cells (with grey) indicate the changed cells in its upset, that is partly visualized, with $\delta = 1082.09$. For example, the value of the top cube changes to store sales^a(2000, All-Countries, Food)= $779,217.89$ from its original value $778,135.80$. Obviously, the part of the upset that is not visualized in the figures is also updated. If, for example, the cube (2000.Quarter, Country.Region.City, Food), also noted as [2,3,3], one would observe the impact of the induced change. This clearly shows, that the what-if analysis takes place in a full OLAP environment, with full support of OLAP's navigational operators, and not in a static reporting environment. This clearly is of benefit to the OLAP analyst.

REFERENCES

Cariou, V., Cubillé, J., Derquenne, C., Goutier, S., Guisnel, F., and Klajnmic, H. (2008). Built-in indicators to discover interesting drill paths in a cube. In *DaWaK '08: Proceedings of the 10th international conference on Data Warehousing and Knowledge Discovery*, pages 33–44, Berlin, Heidelberg. Springer-Verlag.

Caron, E. and Daniels, H. (2008). Extensions to the olap framework for business analysis. In *ICSOFT (ISDM/ABF)*, pages 240–247.

Caron, E. and Daniels, H. (2009). Business analysis in the olap context. In *ICEIS (2)*, pages 325–330.

Caron, E. A. M. and Daniels, H. A. M. (2007). Explanation of exceptional values in multi-dimensional business databases. *European Journal of Operational Research*, 188:884–897.

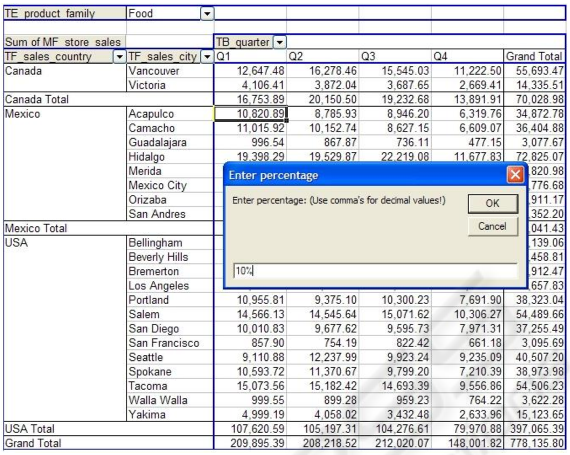
Daniels, H. A. M. and Caron, E. A. M. (2009). Automated explanation of financial data. *Intelligent Systems in Accounting, Finance & Management*, 16(1-2):5–19.

Han, J. and Kamber, M. (2005). *Data Mining: Concepts and Techniques*. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA.

Lenz, H. J. and Shoshani, A. (1997). Summarizability in OLAP and statistical data bases. In *Statistical and Scientific Database Management*, pages 132–143.

Sarawagi, S., Agrawal, R., and Megiddo, N. (1998). Discovery-driven exploration of olap data cubes. In *Conf. Proc. EDBT '98*, pages 168–182, London, UK. Springer-Verlag.

APPENDIX

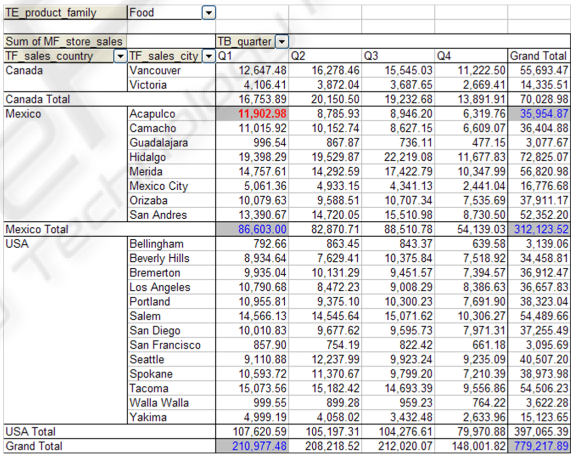


TE_product_family	Food					
Sum of MF_store_sales		TB_quarter				
TF_sales_country	TF_sales_city	Q1	Q2	Q3	Q4	Grand Total
Canada	Vancouver	12,647.48	16,278.46	15,545.03	11,222.50	55,693.47
	Victoria	4,106.41	3,872.04	3,687.65	2,669.41	14,335.51
Canada Total		16,753.89	20,150.50	19,232.68	13,891.91	70,028.98
Mexico	Acapulco	10,820.89	8,785.93	8,946.20	6,319.76	34,872.78
	Camacho	11,015.92	10,152.74	8,627.15	6,609.07	36,404.88
	Guadalajara	996.54	867.87	736.11	477.15	3,077.67
	Hidalgo	19,398.29	19,529.87	22,219.08	11,677.83	72,825.07
	Merida					820.98
	Mexico City					776.68
	Orizaba					911.17
	San Andres					352.20
Mexico Total						041.43
USA	Bellingham					139.06
	Beverly Hills					458.81
	Bremerton					912.47
	Los Angeles					657.83
	Portland	10,955.81	9,375.10	10,300.23	7,691.90	38,323.04
	Salem	14,566.13	14,545.64	15,071.62	10,306.27	54,489.66
	San Diego	10,010.83	9,677.62	9,595.73	7,971.31	37,255.49
	San Francisco	857.90	754.19	822.42	661.18	3,095.69
	Seattle	9,110.88	12,237.99	9,923.24	9,235.09	40,507.20
	Spokane	10,593.72	11,370.67	9,799.20	7,210.39	38,973.98
	Tacoma	15,073.56	15,182.42	14,693.39	9,556.86	54,506.23
	Walla Walla	999.55	899.28	959.23	764.22	3,622.28
	Yakima	4,999.19	4,058.02	3,432.48	2,633.96	15,123.65
USA Total		107,620.59	105,197.31	104,276.61	79,970.88	397,065.39
Grand Total		209,895.39	208,218.52	212,020.07	148,001.82	778,135.80

Enter percentage: (Use comma's for decimal values!)

10%

Figure 2: Here we anticipate on a 10% increase in the cell store sales(2000.Q1, Mexico.Acapulco, Food)= $10,820.89$ in the cube (2000.Quarter, Country.City, Food). This change is automatically propagated to its upset cells.



TE_product_family	Food					
Sum of MF_store_sales		TB_quarter				
TF_sales_country	TF_sales_city	Q1	Q2	Q3	Q4	Grand Total
Canada	Vancouver	12,647.48	16,278.46	15,545.03	11,222.50	55,693.47
	Victoria	4,106.41	3,872.04	3,687.65	2,669.41	14,335.51
Canada Total		16,753.89	20,150.50	19,232.68	13,891.91	70,028.98
Mexico	Acapulco	11,902.98	8,785.93	8,946.20	6,319.76	35,954.87
	Camacho	11,015.92	10,152.74	8,627.15	6,609.07	36,404.88
	Guadalajara	996.54	867.87	736.11	477.15	3,077.67
	Hidalgo	19,398.29	19,529.87	22,219.08	11,677.83	72,825.07
	Merida	14,757.61	14,292.59	17,422.79	10,347.99	56,820.98
	Mexico City	5,061.36	4,933.15	4,341.13	2,441.04	16,776.68
	Orizaba	10,079.63	9,588.51	10,707.34	7,535.69	37,911.17
	San Andres	13,390.67	14,720.05	15,510.98	8,730.50	52,352.20
Mexico Total		86,603.00	82,870.71	88,510.78	54,139.03	312,123.52
USA	Bellingham	792.66	863.45	843.37	639.58	3,139.06
	Beverly Hills	8,934.64	7,629.41	10,375.84	7,518.92	34,458.81
	Bremerton	9,935.04	10,131.29	9,451.57	7,394.57	36,912.47
	Los Angeles	10,790.68	8,472.23	9,008.29	8,386.63	36,657.83
	Portland	10,955.81	9,375.10	10,300.23	7,691.90	38,323.04
	Salem	14,566.13	14,545.64	15,071.62	10,306.27	54,489.66
	San Diego	10,010.83	9,677.62	9,595.73	7,971.31	37,255.49
	San Francisco	857.90	754.19	822.42	661.18	3,095.69
	Seattle	9,110.88	12,237.99	9,923.24	9,235.09	40,507.20
	Spokane	10,593.72	11,370.67	9,799.20	7,210.39	38,973.98
	Tacoma	15,073.56	15,182.42	14,693.39	9,556.86	54,506.23
	Walla Walla	999.55	899.28	959.23	764.22	3,622.28
	Yakima	4,999.19	4,058.02	3,432.48	2,633.96	15,123.65
USA Total		107,620.59	105,197.31	104,276.61	79,970.88	397,065.39
Grand Total		210,977.48	208,218.52	212,020.07	148,001.82	779,217.89

Figure 3: Result of the what-if analysis in the cube (2000.Quarter, Country.City, Food). Colors indicate the changed cells in the upset.