Reducing Multidimensional Data

Faten Atigui, Franck Ravat, Jiefu Song, and Gilles Zurfluh

IRIT - Université Toulouse I Capitole, 2 Rue du Doyen Gabriel Marty
F-31042 Toulouse Cedex 09
{atigui,ravat,song,zurfluh}@irit.fr

Abstract. Our aim is to elaborate a multidimensional database reduction process which will specify aggregated schema applicable over a period of time as well as retains useful data for decision support. Firstly, we describe a multidimensional database schema composed of a set of states. Each state is defined as a star schema composed of one fact and its related dimensions. Each reduced state is defined through reduction operators. Secondly, we describe our experiments and discuss their results. Evaluating our solution implies executing different requests in various contexts: unreduced single fact table, unreduced relational star schema, reduced star schema or reduced snowflake schema. We show that queries are more efficiently calculated within a reduced star schema.

Keywords: multidimensional design, data reduction, experimental assessment.

1 Introduction

Nowadays, decision support systems are based on Multidimensional Data Warehouse (MDW). A MDW schema is based on facts (analysis subjects) and dimensions (analysis axis). By definition, in a MDW, data is stored permanently and new data is periodically added. Hence a DW stores a huge volume of data in which the decision maker may well get lost during his analyses. On the other hand, the pertinence of MDW data decreases with age: while detailed information is generally considered important for recent data [11], it may be of lesser interest for older data. As data value decreases with time, we implement selective deletion at low levels of granularity according to the users' needs. This reduction is achieved mainly through progressive data aggregation: older data is synthesized.

Our objective is to provide a multidimensional analysis environment adapted to decision makers' needs, allowing them to remove the temporal granularity levels which are of little use for analysis.

This paper is composed of the following sections: Section 2 describes a state of the art of data reduction. Section 3 defines our model of multidimensional data based on reductions. Section 4 provides an evaluation of our solution in various implementation environments.

L. Bellatreche and M.K. Mohania (Eds.): DaWaK 2014, LNCS 8646, pp. 208–220, 2014.

2 Related Work

Reducing data allows us both to decrease the quantity of irrelevant data in decision making and to increase future analysis quality [12]. In the context of decision support, data reduction is a technique originally used in the field of data mining [9], [12].

In the DW context, [2] were the first to propose solutions for data deletion. More precisely, they study data expiration in materialized views so that they are not affected and can be maintained after updates.

In the multidimensional area, [11] presents a technique for progressive data aggregation of a fact. This study intends to specify data aggregation criteria of a fact due to higher levels of dimensions. The authors also propose techniques to query reduced multidimensional objects. As mentioned in [6], this work is highly theoretical but it fails to provide us a concrete example of implementation strategy. In [6], a gradual data aggregation solution based on conception, implementation and evaluation is proposed. This solution is based on a table containing different temporal granularities: second, minute, hour, month and year.

This previous work only focuses on the fact table. [5] and [6] use a temporal table for gradual data reduction. Our goal is more ambitious as it aims to study data reduction of the complete multidimensional schema. This reduction depends only on the users' needs. We intend to provide a coherent analysis environment and thus facilitate the decision maker's task by limiting the analysis to semantically coherent data.

3 Our Model

3.1 Case Study

This case study shows a multidimensional schema progression that fulfills the decision maker's needs. During the last four years, sales analysis is carried out with reference to lowest levels of granularity: product, customer and sale date. In the previous period, from 2010 to 2000, analyses are summarized according to product ranges, dates and customer cities because no analysis referring to customers and product codes is required. Before 2000, only annual sales by product ranges make sense.

The following 3 figures represent the evolution of a conceptual multidimensional schema. Each schema represents a state; it is based on a subject of analysis (fact) related to different dimensions. Each fact is composed of one or more indicators. For example, in Figure 1, the fact named "*Sale*" is composed of two indicators: Quantity and Amount. A dimension models an analysis axis; it reflects information according to which subjects of analysis are to be dealt with. For example, in figure 1, the "Sales" fact is connected to 3 dimensions: Products, Customers and Time. Dimension attributes (also called parameters or levels) are organized according to one or more hierarchies. Hierarchies represent a particular vision (perspective) of a dimension. Each schema is based on the graphic notation introduced in [3].

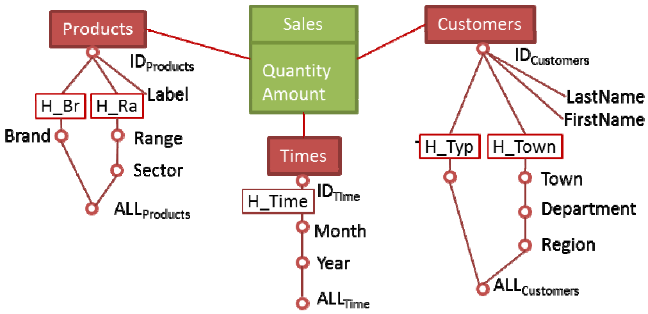


Fig. 1. Current state of the MDW

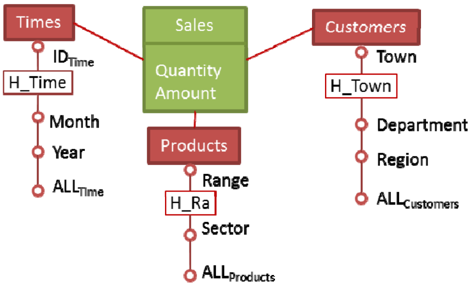


Fig. 2. First reduced state of the MDW.

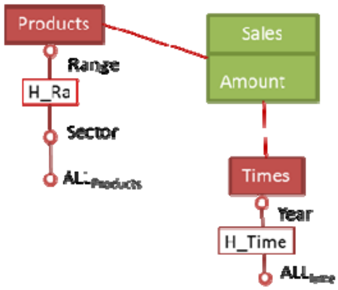


Fig. 3. Second reduced state of the MDW

3.2 Concepts

A MDW is thus modeled as a set of states. The current state corresponds to the present state of the MDW. Past states correspond to a succession of reduced states over time. Each state consists of a star schema composed of a fact and its dimensions.

Definition. A MDW is defined by $S = (n^S ; E ; Map)$ where:

– $n^S \in N$ is the name of the MDW;
– $E = \{E_1 ;\dots ; E_n\}$ is a set of states composing the MDW;
– Map: $E \rightarrow E \mid Map(E_k) = E_{k+1}$ is a derivation function defining the state named E_{k+1} obtained by the reduction of E_k.

Let us define F and D such as $F = \{F_1,\dots, F_n\}$ is a finite set of facts, $n \geq 1$ and $D = \{D_1,\dots, D_m\}$ is a finite set of dimensions, $m \geq 2$.

Definition. A state is a star schema defined for a temporal period such as $E_i = (F_i ; D_i ; T_i)$ where

– $F_i \in F$ is a fact representing a subject of analysis ;
– $D_i = \{D_{TIMES} ; D_1 ;\dots ; D_m\} \subseteq D$ is a set of dimensions associated to the fact with necessarily a temporal dimension denoted D_{TIMES} ;
– $T_i = [t_{start} ; t_{end}[$ is a temporal interval defined on the D_{TIMES} dimension and associated to the state E_i.

To define T_i, we adopt a linear and discrete time model approaching time in granular way through time observation units [13]. A temporal grain is an integer relative to a time unit; we adopt the standard time units manipulated through functions: Year, Quarter, Month, Day... For example, Year (1990) defines the instant "1990" for the year time unit. An instant is a temporal grain. We note T_{now} the current instant which is characterized by its dynamic nature, ie. T_{now} changes constantly depending on the passage of time. A time interval is defined by a couple of instants noted "t_{start}" and "t_{end}". These instants can be fixed (temporal grains) or dynamic (defined with the instant "T_{now}").

Example. The following figure represents the 3 states of our case study. It illustrates the principle of states derived by the reduction. This MDW is defined as follows: $\mathcal{E} = \{E_1 ; E_2 ; E_3\}$ with Map = { (E_1, E_2) ; (E_2, E_3) } where

- $E_1 = (F_{SALES} ;\{D_{PRODUCTS} ; D_{TIMES} ; D_{CUSTOMERS}\} ; [Year(T_{now})-4 ; Year(T_{now})[)$;
- $E_2 = (F_{SALES} ;\{D_{PRODUCTS} ; D_{TIMES} ; D_{CUSTOMERS}\} ; [Year(T_{now})-14 ; Year(T_{now})-4[)$;
- $E_3 = (F_{SALES} ; \{D_{PRODUCTS} ; D_{TIMES}\} ; [Year(1990); Year(T_{now})-14[)$.

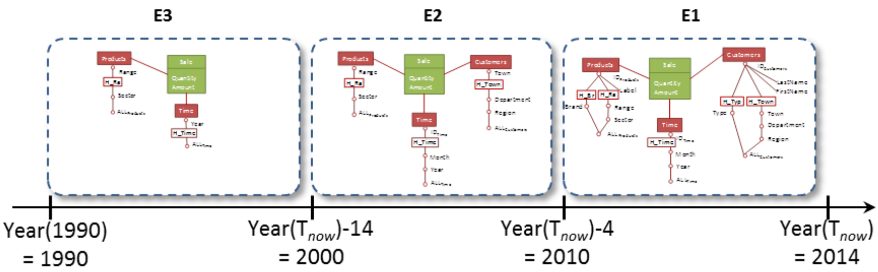


Fig. 4. Reduction principle of multidimensional schemas

The state denoted E1 and called current state, is associated to the validity interval $[Year(t_{now})-4; Year(T_{now})[$ corresponding to [2010; 2014[. The instances of this state correspond to sales between 2010 and 2014 only, according to the D_{TIMES} dimension. In the same way, the state named E2 stores data related to sales between 2000 and 2010, whereas the state denoted E3 stores data related to sales prior to 2000.

In Figure 4, 1990 is a fixed instant representing the date when the database was created. In this figure, we can also find time-variant intervals (moving over time) defined by the following instants: $Year(T_{now})-14$, $Year(T_{now})-4$ and $Year(T_{now})$. So, next year, $Year(T_{now}) = 2015$, $Year(T_{now})-4 = 2011$ and $Year(T_{now})-14 = 2001$. At each change of year, the states denoted E1, E2 and E3 will be instantly updated.

Definition. A *fact*, denoted F_i, $\forall i \in [1..n]$, is defined by (n^{F_i}, M^{F_i}) where

- $n^{F_i} \in \mathcal{N}$ is the fact name;
- $M^{F_i} = \{m_1,..., m_{pi}\}$ is a set of *measures* or indicators.

Definition. A *dimension*, denoted D_i, $\forall i \in [1..m]$, is defined by (n^{Di}, A^{Di}, H^{Di}), where

- $n^{Di} \in \mathcal{N}$ is the dimension name;
- $A^{Di} = \{ a_1^{D_i}, \ldots, a_{r_i}^{D_i} \}$ is the set of the *attributes of the dimension*;
- $H^{Di} = \{ H_1^{D_i}, \ldots, H_{h_i}^{D_i} \}$ is a set of *hierarchies*.

Hierarchies organize the attributes of a dimension, from the finest graduation (root parameter, ID_{Di}) to the most general graduation (extremity parameter, All_{Di}). Thus, a hierarchy defines the valid navigation paths on an analysis axis.

Definition. A *hierarchy*, denoted H_j (abusive notation of $\boldsymbol{H_j^{D_i}}$, $\forall i \in [1..m]$, $\forall j \in [1..h_i]$) is defined by (n^{Hj}, P^{Hj}, $\prec^{Hj}$, $Weak^{Hj}$), where:

- $n^{Hj} \in \mathcal{N}$ is the hierarchy name;
- $P^{Hj} = \{ p_1^{H_j}, \ldots, p_{q_j}^{H_j} \}$ is a set of attributes called *parameters*, $P^{Hj} \subseteq A^{Di}$;
- $\prec^{Hj} = \{(p^{Hj}_x, p^{Hj}_y) \mid p^{Hj}_x \in P^{Hj} \wedge p^{Hj}_y \in P^{Hj}\}$ is an antisymmetric and transitive binary relation between parameters. Remember that the antisymmetry means that $(p^{Hj}_{k1} \prec^{Hj} p^{Hj}_{k2}) \wedge (p^{Hj}_{k2} \prec^{Hj} p^{Hj}_{k1}) \Rightarrow p^{Hj}_{k1} = p^{Hj}_{k2}$ while the transitivity means that $(p^{Hj}_{k1} \prec^{Hj} p^{Hj}_{k2}) \wedge (p^{Hj}_{k2} \prec^{Hj} p^{Hj}_{k3}) \Rightarrow p^{Hj}_{k1} \prec^{Hj} p^{Hj}_{k3}$.
- $Weak^{Hj} : P^{Hj} \rightarrow 2^{A^{D_i} \setminus P^{H_j}}$ is an application that associates to each parameter a set of dimension attributes, called *weak attributes* (2^N represents the power set of N).

In the rest of the paper we denote each fact F_i that is an abusive notation of $F_{n^{F_i}}$. In the same way, D_i corresponds to $D_{n^{D_i}}$.

Example. The E_3 state of the previous figure is composed of one fact and two dimensions and it is valid from 1990 to 2000. The fact table named SALES contains the measure Amount. The dimension PRODUCTS contains the hierarchy H_Ra on which the parameters are organized according to their granularity level: from the lowest level Range to the highest level $ALL_{PRODUCTS}$. The other dimension is named D_{TIMES}, it is graduated by the attributes Year and ALL_{TIMES} on the hierarchy H_Time.

The abstract representation is as follows:

$E_3 = (F_{SALES} ; \{ D_{PRODUCTS} ; D_{TIMES} \} ; [t_{1990} ; t_{2000}[)$ where:

- F_{SALES} = (SALES; { Amount });
- $D_{PRODUCTS}$ = (PRODUCTS; {Range, Sector, $ALL_{PRODUCTS}$ }; {H_Ra});
- D_{TIMES} = (TIMES; { Year, ALL_{TIMES} }; {H_Time}).

The H_Ra hierarchy is defined by (n^{H_Ra}, P^{H_Ra}, $\prec^{H\_Ra}$, $Weak^{H\_Ra}$) where:

- n^{H_Ra} = H_Ra;
- P^{H_Ra} = { Range, Sector, $ALL_{PRODUCTS}$ };
- $\prec^{H\_Ra}$ = {(Range, Sector); (Sector, $ALL_{PRODUCTS}$)};

$Weak^{H\_Ra} = \emptyset$.

3.3 Reduction Operators

Deriving the reduced schema denoted E_{k+1} from a schema denoted E_k is performed through the composition of derivation operators. We define the set of these operators as $\mathcal{O}$ = {$RollUp^{reduce}$; $Drop^{reduce}$; $Slice^{reduce}$} as the minimum core of elementary operators to define the derivation.

- The $RollUp^{reduce}$ operator provides a new state in which the specified dimension is reduced by removing all the attributes under the parameter that is specified in the operator. If the specified parameter is an extremity parameter like $ALL_{D_{rollup}}$, the dimension is completely removed in the reduced state.
- The $Drop^{reduce}$ operator provides a new state in which the fact is reduced by the deletion of the specified measure.
- The $Slice^{reduce}$ operator provides a reduced state in which the instances of the specified dimension denoted D_{Slice} is reduced. The dimension instances that satisfy the predicate denoted $pred_{slice}$ are kept in the new state.

Table 1. Reduction operators on schemata

Operators	
$\mathbf{RollUp^{reduce}(E_k\ ;\ D_{rollup}\ ;\ p_{rollup}\ ;\ T_{k+1}) = E_{k+1}}$	
Inputs	$E_k = (F_k\ ;\ \mathcal{D}_k\ ;\ T_k)$: initial state; $D_{rollup} \in \mathcal{D}_k$: dimension dedicated to a reduction; $p_{rollup} \in A^{Drollup}$: reduction parameter of the D_{rollup} dimension.
Output	$E_{k+1} = (F_{k+1}\ ;\ \mathcal{D}_{k+1}\ ;\ T_{k+1})$ reduced state such as - $F_{k+1} = F_k$; - $\mathcal{D}_{k+1} = \mathcal{D}_k \setminus \{ D_{rollup} \} \cup \{ D_{new} \}^{(*)}$ with $D_{new} = (n^{Dnew}\ ;\ A^{Dnew}\ ;\ H^{Dnew})$ - $n^{Dnew} = n^{Dold}$ - $A^{Dnew} = \{ a_x \in A^{Drollup} \mid a_x = p_{rollup} \vee \forall H_j \in H^{Drollup}, p_{rollup} \prec^{Hj} a_x \}$ - $H^{Dnew} = \{ H_x \in H^{Drollup} \mid n^{Hx} = n^{Hj} \wedge P^{Hx} = \{ p_y \in P^{Hj} \mid p_y = p_{rollup} \vee p_{rollup} \prec^{Hj} p_y \} \wedge \prec^{Hx} = \{ (p^{Hj}_{x1}, p^{Hj}_{x2}) \in \prec^{Hj} \mid p^{Hj}_{x1} = p_{rollup} \vee p_{rollup} \prec^{Hj} p^{Hj}_{x1} \} \wedge Weak^{Hx} := \{ (p_{x1}, A^{Hx}_{x1}) \in Weak^{Hj} \mid p_y \in P^{Hj} \}$.
$\mathbf{Drop^{reduce}(E_k\ ;\ m_{drop}\ ;\ T_{k+1}) = E_{k+1}}$	
Inputs	$E_k = (F_k\ ;\ \mathcal{D}_k\ ;\ T_k)$: initial state ; $m_{drop} \in M_k$ is a measure of F_k.
Output	$E_{k+1} = (F_{k+1}\ ;\ \mathcal{D}_{k+1}\ ;\ T_{k+1})$ reduced state such as - $F_{k+1} = (n^{Fk}, M^{Fk} \setminus \{ m_{drop} \})$; - $\mathcal{D}_{k+1} = \mathcal{D}_k$.
$\mathbf{Slice^{reduce}(E_k\ ;\ D_{slice}\ ;\ \boldsymbol{pred}_{slice}\ ;\ T_{k+1}) = E_{k+1}}$	
Inputs	$E_k = (F_k\ ;\ \mathcal{D}_k\ ;\ T_k)$: initial state ; $D_{slice} \in \mathcal{D}_k$: dimension dedicated to a reduction; $pred_{slice}$: selection predicate on a domain denoted $dom(D_{slice})$ of D_{slice}.
Output	$E_{k+1} = (F_{k+1}\ ;\ \mathcal{D}_{k+1}\ ;\ T_{k+1})$ reduced state such as - $F_{k+1} = F_k$; - $\mathcal{D}_{k+1} = \mathcal{D}_k$ with $dom(D_{slice}) = \{ v_i \in dom(D_{slice}) \mid pred_{slice}(v_i) = \text{TRUE} \}$.

$^{(*)}$ If $A^{Dnew} = \{ALL_{D_{rollup}}\}$ then $\mathcal{D}_{k+1} = \mathcal{D}_k \setminus \{ D_{rollup} \}$

Example. In the previous example, we defined two reduced states. Each of them is defined by a derivation function. These functions are defined bellow. The first Map function, composed of two $RollUp^{reduce}$ operators, permits to define the "E2" state. The second Map function, composed of two $RollUp^{reduce}$ operators and one $Drop^{reduce}$ operator, permits to define the "E3" state.

- $RollUp^{reduce}(RollUp^{reduce}(E_1$; $D_{PRODUCTS}$; P_{RANGE} ; [Year(T_{now})-14 ; Year(T_{now})-4[) ; $D_{CUSTOMERS}$; P_{TOWN} ; [Year(T_{now})-14 ; Year(T_{now})-4[) = E_2 ;
- $RollUp^{reduce}(RollUp^{reduce}(Drop^{reduce}(E_2$; Quantity ; [Year(1990) ; Year(T_{now})-14[) ; $D_{CUSTOMERS}$; $ALL_{CUSTOMERS}$; [Year(1990) ; Year(T_{now})-14[) ; D_{TIMES} ; P_{YEAR} ; [Year(1990) ; Year(T_{now})-14[) = E_3.

4 Experimental Assessment

4.1 Data Collection

In order to make experimental assessments, we implement two types of R-OLAP databases with the Oracle DBMS and each type has two different implementations. The first type of MDW corresponds to databases without reduction. Its first implementation is called *Global Star,* consists in an unreduced R-OLAP implementation based on 4 tables (*Products*, *Customers*, *Times* and *Sales*). The second implementation is called *Global Table* in which we merge the three analysis axis (dimensions *Products*, *Customers* and *Times*) with the fact table (*Sales*); consequently this implementation is composed of a single fact table that encompasses all.

The population of the analysis axis was done as follows: (a) the dimension *Times* contains all dates from 01/01/1990 to 31/12/2013, (b) the two other dimensions contain random data defined by generation of synthetic data. Allocation of random data was made so that father attribute of a hierarchy does not have the same number of sons while respecting the integrity constraints of strict hierarchies (any son attribute of a hierarchy has a single father attribute).

We have defined various versions of non-reduced databases by ranging the tuple numbers of the dimensions *Customers* and *Products* from 10 to 40 tuples.

- |*Customers*| = 10, 20, 30, 40 tuples
- |*Products*| = 10, 20, 30, 40 tuples
- |*Times*| = 8401 tuples (from 01/01/1990 to 31/12/2013)
- |*Sales*| = |*Customers*| x |*Products*| x |*Times*| = **840 100** to **13 441 600** tuples.

Even though the dimensions *Customers*, *Products* and *Times* are integrated in the fact table of *Global Table*, the implementation details of MDW *Global table* are the same as MDW *Global Star*. The following table describes different values associated to the attributes of dimension containing variable data.

The second type of MDW corresponds to reduced databases. This type consists of three states according to the case study presented in this article (see Figure 4). We have defined two implementations of reduced databases:

- a denormalized implementation (*R-OLAP star schema* defined in fig. 5 (a)),
- a normalized implementation (*snowflake schema* defined in fig. 5 (b)).

The operations permitting to get the different states of MDB were implemented with the help of triggers in Oracle DBMS.

Table 2. Implementation details of the dimensions in *Global Star* and *Global Table*.

|*Customers*| x |*Products*|	Contents of the dimension *Customers*	Contents of the dimension *Products*
10 x 10	2 *Towns*, 2 *Departments* , 1 *Region*, 2 *Types*	2 *Ranges*, 2 *Sectors*, 2 *Brands*
20 x 20	4 *Towns*, 3 *Departments*, 2 *Regions*, 4 *Types*	4 *Ranges*, 3 *Sectors*, 4 *Brands*
30 x 30	6 *Towns*, 4 *Departments*, 2 *Regions*, 6 *Types*	6 *Ranges*, 4 *Sectors*, 6 *Brands*
40 x 40	8 *Towns*, 5 *Departments*, 3 *Regions*, 8 *Types*	8 *Ranges*, 5 *Sectors*, 8 *Brands*

E1

SALES(Quantity, Amount, IDProducts#, IDTime#, IDCustomers#)
CUSTOMERS(IDCustomers, Lastname, Firstname, Town, Department, Region, Type)
PRODUCTS(IDProducts, Range, Sector, Brand)
TIMES(IDTime, Month, Year)

E2

SALES(Quantity, Amount, Range#, IDTime#, Town#)
CUSTOMERS(Town, Department, Region)
PRODUCTS(Range, Sector)
TIMES(IDTime, Month, Year)

E3

SALES(Amount, Range#, Year#)
PRODUCTS(Range, Sector)
TIMES(Year)

(a) R-OLAP star schema

E1

SALES(Quantity, Amount, Range#, IDTime#, Town#)
CUSTOMERS(IDCustomers, Lastname, Firstname, Town#, Type#)
TOWN(Town, Department#)
DEPARTMENT(Department, Region#)
REGION(Region)
TYPE(Type)
PRODUCTS(IDProducts, Range#, Brand#)
RANGE(Range, Sector#)
SECTOR(Sector)
BRAND(Brand)
TIMES(IDTime, Month#)
MONTH(Month, Year#)
YEAR(Year)

E2

SALES(Quantity, Amount, Range#, IDTime#, Town#)
CUSTOMERS(Town, Department#)
DEPARTMENT(Department, Region#)
REGION(Region)
PRODUCTS(Range, Sector#)
SECTOR(Sector)
TIMES(IDTime, Month#)
MONTH(Month, Year#)
YEAR(Year)

E3

SALES(Amount, Range#, Year#)
PRODUCTS(Range, Sector#)
SECTOR(Sector)
YEAR(Year)

(b) R-OLAP snowflake schema

Fig. 5. R-OLAP schemata of reduced MDB

4.2 Protocol

The experimental assessment compares the execution time and the cardinalities of queries executed in two unreduced R-OLAP implementations with two types of reduced R-OLAP implementations of the same multidimensional database. This experimental assessment takes into account three criteria:

- Database volumetry: As mentioned above, we will apply queries of 4 versions for the different types of databases.
- Query types: (a) Queries containing only joins and no selection criteria on non-temporal dimensions (querying all the data of reduced database states), (b) Queries containing conditions restrictions on the data (querying certain data in certain states)
- Scope of queries: (a) queries related to one or more dimension tables, (b) queries manipulating 1, 2 or 3 states.

4.3 Results and Discussion

4.3.1 Queries without Restriction Predicates on Non-temporal Dimensions

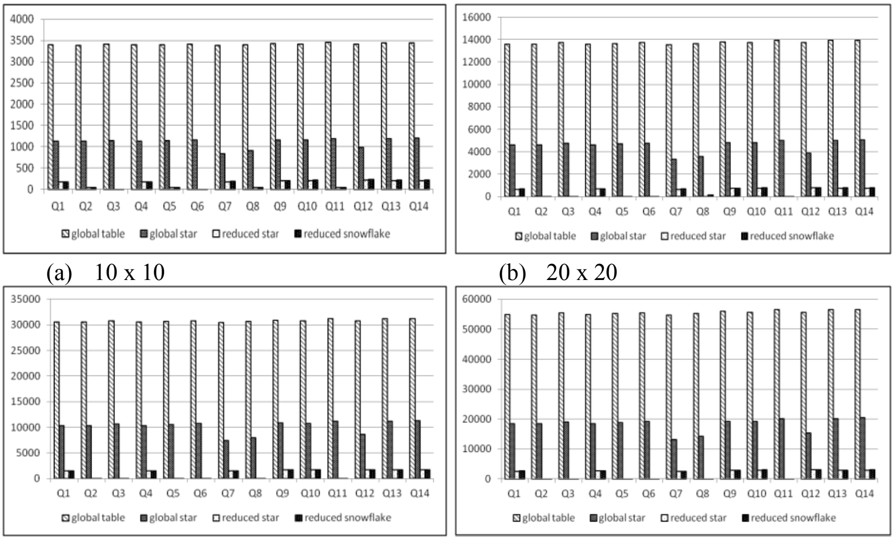
The first experimental assessment compares the theoretical execution time of queries (explain plan of the Oracle DBMS) by varying the size of the MDW in accordance with the protocol previously described. We have defined 14 queries manipulating different tables and different states. Each query is implemented in SQL.

Table 3. Queries without restriction predicates on non-temporal dimensions

Queries	States	Dimensions
Q_1: Amount of sales for the last three years	E_1	1 D: Time
Q_2: Amount and quantity of sales in 2008	E_2	1 D: Time
Q_3: Amount of annual sales before 2000	E_3	1 D: Time
Q_4: Amount of sales by cities from 2010 to 2012	E_1	2 D: Time, Customers
Q_5:Amount of monthly sales by departments from 2000 to 2005	E_2	2 D: Time, Customers
Q_6: Amount of annual sales by sector before 2000	E_3	2 D: Time, Products
Q_7: Amount of sales by cities, sectors and months in 2012	E_1	3 D: Time, Products, Customers
Q_8: Amount of annual sales by sectors and departments from 2000 to 2005	E_2	3 D: Time, Products, Customers
Q_9: Amount of monthly sales since 2000	E_1; E_2	1 D: Time
Q_{10}: Amount of annual sales per cities from 2002 to 2012	E_1; E_2	2 D: Time, Customers
Q_{11}: Amount of sales per year and range from 1990 to 2009	E_2; E_3	2 D: Time, Products
Q_{12}: Amount of sales by cities and sectors from 2002 to 2012	E_1; E_2	3 D: Time, Products, Customers
Q_{13}: Amount of annual sales	E_1; E_2; E_3	1 D: Time
Q_{14}: Amount of annual sales per ranges	E_1; E_2; E_3	2 D: Time, Products

Remark. It is impossible to define a query manipulating 3 states and 3 dimensions because the state denoted E3 is only composed of 2 dimensions.

Whatever the volumetry of database, the query execution time in a non reduced environment (the column with stripe and the gray column in the figure below) is more important than in a reduced environment (the white and black columns in the figure below). The lowest execution times are performed on the database called *Reduced* *star*: database content is reduced and the table number is limited. In each version, the average earnings are 89.43% for the size 10 X 10 to 90.26% in size 40 X 40. Whatever the database volumetry, the execution time gain is significant: about 90%. As mentioned in figure 7, the highest average execution time for unreduced MDW ranges from 3432.4 (*global table* database 10 x 10) to 55221 (*global table* database 40 x40) and this average execution time has increased by 1509%. The lowest average execution time is found within the Reduced Star databases: it ranges from 115.6 to 1720, and it has increased by 1388%, lower than 120% compared to the unreduced DW. Thus, the more the datawarehouse volumetry increases, the more the execution time gain on a reduced DW is important.



(a) 10 x 10 (b) 20 x 20

(c) 30 x 30 (d) 40 x 40

Fig. 6. Execution time in 4 versions

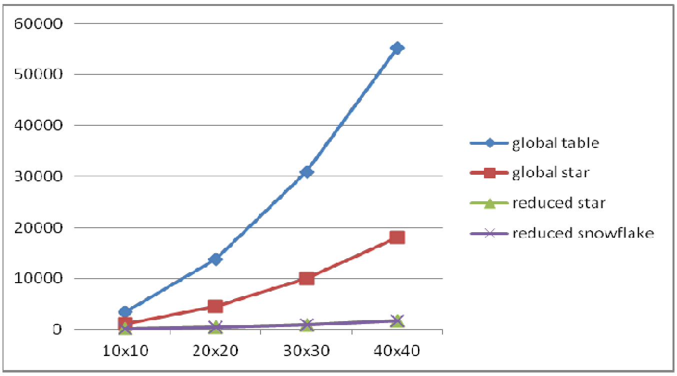


Fig. 7. Average execution time for the different versions of MDW

4.4 Queries with Restriction Predicates on Non-temporal Dimensions

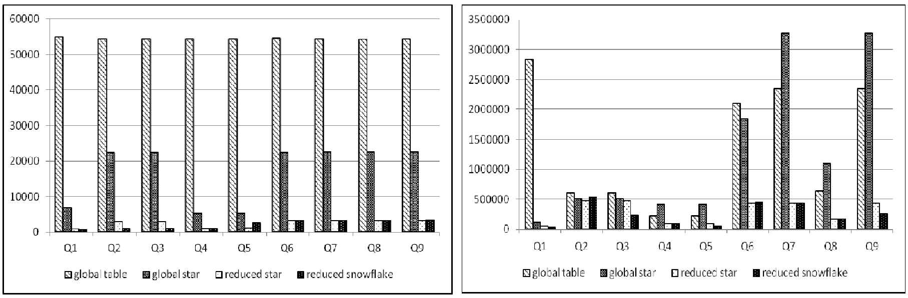
This second experimental assessment aims to analyze the impact of restriction criteria in queries on the execution time. This experimental assessment is only done on the database based on the size 40 x 40. The queries of this second experimental assessment are defined in the following table.

Table 4. Queries with restriction predicates on non-temporal dimensions

Queries	States	Dimensions
Q_1 : Amount of sales of a customer X from 2010 to 2012	E_1	1 D: Customers
Q_2: Amount of sales in a town X from 2010 to 2012	E_1	1 D: Customers
Q_3: Amount of sales in a department X from 2010 to 2012	E_1	1 D: Customers
Q_4: Amount of monthly sales of products of a range X sold in the town X since 2000	E_1, E_2	2 D: Products, Customers
Q_5: Amount of monthly sales of products of a sector X sold in the town X since 2000	E_1, E_2	2 D: Products, Customers
Q_6: Amount of monthly sales of products (All) sold in the town X since 2000	E_1, E_2	2 D: Products, Customers
Q_7: Amount of annual sales of a range X	E_1; E_2; E_3	1 D: Products
Q_8: Amount of annual sales of a range Y (the products of range X is three times more than those of range Y)	E_1; E_2; E_3	1 D: Products
Q_9: Amount of annual sales of a sector X	E_1; E_2; E_3	1 D: Products

The following figure shows execution times and cardinalities of results for the four implementations. Contrary to our expectations, the gains between unreduced and reduced DW remain in the same proportions whether we apply restriction or not. Indeed, this gain ranges from 77.54% (Q_5) to 88.27% (Q_1) with an average of 84.61%. Moreover, whatever the scope of the restriction predicates (primary key, attribute containing different values or not), the standard deviation is not very high (0.1).

In addition, even if execution times of Q_7 and Q_8 are similar, we can notice that the cardinality of the result of Q_7 is three times higher than the cardinality of the result of Q8. This is because the DBMS must review all the tuples of the tables to get the query result. So, the execution time gain is independent of the cardinality of the result.



(a) Execution time (b) Cardinalities

Fig. 8. Execution times and cardinalities for 9 queries containing restriction predicates

5 Conclusion

This paper resides within the field of MDW. Our objective is to specify aggregated schema over time in order to retain only the data useful for decision support according to the needs of users. Firstly, we define a conceptual model which allows us to specify MDW schemata composed of a set of states varying over time. Each state consists of a star schema and is defined with a mapping function, itself defined with reduction operators based on an extension of classical OLAP operators adapted to the reduction context. Secondly, we defined experimental assessments. Evaluating our solution consists in executing different queries in various environments: ROLAP schema without reduction, single fact table schema without reduction as well as star and snowflake schemata with reductions. We use multidimensional databases with different sizes; the fact table size ranges from 840,100 to 13,441,600 tuples. Whatever the database volumetry, the execution time gain between unreduced and reduced databases is significant: about 90%. Moreover, the more the datawarehouse volumetry increases, the more the execution time gain is important. These gains remain in the same proportions when we apply restriction predicates or not on the queries. Finally, the execution time gain is independent of the cardinality of the result.

In the future, we intend to extend our conceptual proposal in order to integrate other operators in the definition of the reduction function. We also intend to extend our experiments by combining our own work on reductions with that concerning indexes in a multidimensional context [6]. At last we wish to apply the principles of reduction to a reel data sample of analytic domain such as banking or insurance etc.

References

1. Boly, A., Hébrail, G., Goutier, S.: Forgetting Data Intelligently in Data Warehouses. In: Proceedings of the IEEE International Conference on Research, Innovation and Vision for the Future, pp. 220–227. IEEE Press, New York (2007)
2. Garcia-Molina, H., Labio, W., Yang, J.: Expiring data in a warehouse. In: Gupta, A., Shmueli, O., Widom, J. (eds.) 24th International Conference on Very Large Data Bases (VLDB), pp. 500–511. Morgan Kaufmann (1998)
3. Golfarelli, M., Maio, D., Rizzi, S.: Conceptual design of data warehouses from E/R schemes. In: Proceedings of the 31st Hawaii Int. Conf. on System Sciences (1998)
4. Golfarelli, M., Rizzi, S.: A survey on temporal data warehousing. International Journal of Data Warehouse and Mining 5(1), 1–17 (2009)
5. Iftikhar, N., Pedersen, T.B.: Using a Time Granularity Table for Gradual Granular Data Aggregation. In: Catania, B., Ivanović, M., Thalheim, B. (eds.) ADBIS 2010. LNCS, vol. 6295, pp. 219–233. Springer, Heidelberg (2010)
6. Iftikhar, N., Pedersen, T.B.: A rule-based tool for gradual granular data aggregation. In: Song, Cuzzocrea, Davis (eds.) Proceedings of DOLAP 2011, pp. 1–8. ACM (2011)
7. Kimball, R.: The Data Warehouse Toolkit: Practical Techniques for Building Dimensional Data Warehouses. John Wiley & Sons, USA (1996)
8. Last, M.,, Maimon, O.: Automated dimensionality reduction of data warehouses. In: Jeusfeld, M.A., Shu, H., Staudt, M., Vossen, G. (eds.) Proceedings of DMDW 2010. CEUR Workshop Proceedings, vol. 28, p. 7. CEUR-WS.org (2000)

9. Okun, O., Priisalu, H.: Unsupervised data reduction. Signal Processing 87(9), 2260–2267 (2007)
10. Ravat, F., Teste, O., Tournier, R., Zurfluh, G.: Algebraic and graphic languages for OLAP manipulations. International Journal of Data Warehouse and Mining (4), 17–46 (2008)
11. Skyt, J., Jensen, C.S., Pedersen, T.B.: Specification-based data reduction in dimensional data warehouses. Information System 33(1), 36–63 (2008)
12. Udo, I.J., Afolabi, B.: Hybrid Data Reduction Technique for Classification of Transaction Data. Journal of Computer Science and Engineering 6(2), 12–16 (2011)
13. Wang, X., Bettini, C., Brodsky, A., Jajodia, S.: Logical Design for Temporal Databases with Multiple Granularities. ACM Trans. Database Syst. 22(2), 115–170 (1997)