



A Simpler and Semantic Multidimensional Database Query Language to Facilitate Access to Information in Decision-making

F. Palominos, F. Córdova, C. Durán, B. Nuñez

Fredi Palominos*, **Bryan Nuñez**

Mathematics and Computer Science Department

University of Santiago de Chile, Chile

Avda. Libertador B. O'Higgins 3363, Estación Central, Santiago, Chile

*Corresponding author: fredi.palominos@usach.cl

bryan.nunez@usach.cl

Felisa Córdova

School of Engineering

FinisTerra University, Santiago, Chile

Av. Pedro de Valdivia 1509, Providencia, Región Metropolitana, Chile

fcordova@uft.cl

Claudia Durán

Faculty of Engineering

Universidad Tecnológica Metropolitana, Santiago, Chile

Dieciocho 161, Santiago, Chile

c.durans@utem.cl

Abstract

OLAP and multidimensional database technology have contributed significantly to speed up and build confidence in the effectiveness of methodologies based on the use of management indicators in decision-making, industry, production, and services. Although there are a wide variety of tools related to the OLAP approach, many implementations are performed in relational database systems (R-OLAP). So, all interrogation actions are performed through queries that must be reinterpreted in the SQL language. This translation has several consequences because SQL language is based on a mixture of relational algebra and tuple relational calculus, which conceptually responds to the logic of the relational data model, very different from the needs of the multidimensional databases. This paper presents a multidimensional query language that allows expressing multidimensional queries directly over ROLAP databases. The implementation of the multidimensional query language will be done through a middleware that is responsible for mapping the queries, hiding the translation to a layer of software not noticeable to the end-user. Currently, progress has been made in the definition of a language where through a key statement, called aggregate, it is possible to execute the typical multidimensional operators which represent an important part of the most frequent operations in this type of database.

Keywords: data models, multidimensional model, OLAP, decision making, query languages.

1 Introduction

Every decision-making process in unstructured problems requires an analysis in different perspectives. The evolution of the problem over time is usually present in virtually all cases. Gradually, as information technology became present in all types of companies and organizations, data availability and the ability to learn about them increased.

In business, the information management of services as well as production processes has a strategic role that is increasingly important. It allows decision makers to have relevant information that can be transformed into explicit knowledge and that can be effectively monitored with indicators [17] [6] [13]. Consequently, it is essential to have data that is reliable, aligned with the objectives and goals of the company, not redundant, consistent and hierarchical [8].

The impossibility of analyzing the existing data in the administrative information systems while they are in operation, as well as the dispersion of the critical data in different systems, with low or no integration levels, led to the emergence of new paradigms such as data mining, business intelligence and Online Analytical Processing (OLAP) [7].

Basically, the OLAP paradigm establishes that in decision-making processes the data must be organized according to the analysis perspectives, for this reason the data is not structured according to the objects present in the problem, nor neither in lines or sequences of transactions, but according to the needs of analysis, which then translates into dimensions, which in turn are structured in hierarchies of attributes, to which indicators called measures are associated that are finally subject to analysis [9] [10] [2] [3].

The implementation of the OLAP paradigm in the mid-1990s gave rise to multidimensional databases known to many people as data warehouse [11]. As it is a technology, certain requirements have been imposed such as providing dynamic multidimensional analysis and allowing end users to perform analytical and navigational activities.

In this way, it is necessary to ensure that multidimensional databases meet certain basic operational requirements known as: having integrated data, time variants, non-volatile and oriented to issues of interest for the management of an organization or company.

Very often, multidimensional databases have been implemented on relational data base systems, in a modality known as ROLAP. The adaptation to relational technology obviously has some negative consequences, one of the most obvious being the need to transform OLAP queries into relational queries [10] [1].

On the other hand, considering the need for a functional multidimensional data model, which resembles the simplicity of the relational approach in the formulation of queries, in [14] [15] a multidimensional model has been presented that defines structures and operators with similarities to the relational model [5]. In addition, it implements the typical OLAP operations, clearly differentiating the static components of the model from the dynamic components, providing a query language in the form of a simple, more expressive algebra, with some reminiscences to characteristics of the tuple relational calculation.

Query languages in multidimensional databases, such as the case of MDX [3], are usually quite aligned with other more classic query languages. Generally, the most operational aspects have been privileged over the more formal aspects. Those who have ventured into the definition of algebras that allow multidimensional data to be operated more independently of languages, have proposed sophisticated conceptual frameworks [12] [19] [20] difficult to link with software tools and incomprehensible to end users [18] [4]. The works developed in [14] [15] present the multidimensional model in a manner similar to the relational model [5], presenting construction rules, some inherent restrictions, as well as simpler and more significant operators, which are an appropriate for the definition of a more understandable query language for end users.

This work presents the central ideas of a new data definition and manipulation language oriented to multidimensional data bases, which is based on the structure, operability and semantics of the multidimensional data model previously presented in [14] [15]. It focuses on the characterization of the language in order to subsequently address the experimental implementation of a prototype (middleware) acts over a relational database management system.

2 Components of the multidimensional data model

The following sections describe the static and dynamic components of the multidimensional data model that is being used as the basis for defining the syntax and semantics of a multidimensional query language. These components are defined more extensively in [15]. Below is a summary of its main characteristics.

2.1 Static components

Domains and Attributes: A domain is a set of atomic values of the same type of data. The characteristics of an object are called attributes and are defined as a function that associates a multidimensional object.

Hierarchy: A hierarchy [10] is a 5-tuple $(N, E, \leq T, V)$ where N is a name, E contains the attributes of the hierarchy and \leq is a partial order relationship defined on E . Thus, a hierarchy constitutes an order relationship whose effect is to produce a hierarchy of attributes, resulting in the generation of different levels of aggregation.

Dimension: In multidimensional databases it is necessary to address queries, aggregations and data analysis based on different points of view, which respond to the needs of data analysis. Each point of view is called a dimension, which we will represent as a pair (N_D, J) .

Since in reality the needs of data analysis require classifying them around attributes of different dimensions, it is necessary to articulate the dimensions in a structure called Dimensional Space (DS). It is important to emphasize that the Dimensional Space constitutes a coordinate system that allows to classify in a unique and unequivocal way each fact of the real world.

Measures: A measure is a special set of numerical type attributes (usually an aggregate data) and represents a quantitative characteristic of interest that must be monitored and analyzed. However, the measures do not constitute a hierarchy. The measurements can be as many as necessary and the Cartesian product of the domains of the measures present in a multidimensional variable is called Scalar Space (SS).

Facts: A fact represents information of interest regarding a set of real-world events related to one or more objects. The characteristics of the event are captured through the values of the attributes of the hierarchies, the dimensions and measures. The following definition makes some pressures to the definition that was previously used in [15], both in the structure and in the semantic properties of the facts contained in the multidimensional variable. A fact is a relationship between Dimensional Space and Scalar Space, as represented by the following function:

$$h : DS \sim SS$$

We will denote by H the whole $\{x/x = (d, e), d \in DS, e \in EE, e = h(d)\}$, the one that receives the name of Multidimensional Variable (VM).

Schematically H we will represent it by the pair (\vec{J}, \vec{m}) where $\vec{J}((J_1, J_2, \dots, J_n))$ represents the hierarchies of the dimensions that in turn determine the Dimensional Space of the set and, $\vec{m}((m_1, m_2, \dots, m_k))$ the k measures of H .

In this way, we can represent a set of facts, as needed, in the following ways:

$$H \Leftrightarrow (\vec{J}, \vec{m}) \Leftrightarrow ((J_1, J_2, \dots, J_n), (m_1, m_2, \dots, m_k))$$

2.2 Aggregation Functions

OLAP requires intense summarization operations to be able to respond to the different demands of providing online data analysis capability. In addition, it is necessary that these operations have consistency and allow the calculation of valid and understandable indicators for the users, for this reason the summarization operations applied over a set ξ , must be performed using a special type of functions, called aggregation [14]:

$$f(\xi) = \sum_{x \in \xi} f(x)$$

It is said that this function is additive if for two sets $\xi_1, \xi_2 : \xi_1 \cap \xi_2 = \phi$, it is fulfilled that:

$$f(\xi_1 \cup \xi_2) = f(\xi_1) + f(\xi_2)$$

Finally, we will call a **Multiple Aggregation Function (MAF)**, a function $h : DS \sim SS$, such that $\tau(\xi) = \langle F_1(\xi), F_2(\xi), \dots, F_n(\xi) \rangle$ where each $F_i, i = 1, \dots, w$ is a simple Aggregation Function.

2.3 Aggregate Facts Family (AFF)

It is a new concept and structural element added to the multidimensional model in [14], which is intended to take advantage of the additive aggregation functions, for the benefit of efficiency and versatility in the responses to queries. It is denoted as \mathcal{H} and corresponds to a set of facts such that each element of \mathcal{H} is an aggregation that is obtained directly or indirectly from an initial event H^* called a generator.

$$\mathcal{H} = \{H \mid H = F(CS(H^*))\}$$

As described in [14], for \mathcal{H} , F is an invariant multiple aggregation function, common to all elements of \mathcal{H} that differs by level of data aggregation expressed through a classification space (CS).

2.4 Dynamics components

The operators that correspond to the typical OLAP operations are defined below:

Pivoting: It is defined as a rotation (permutation) of the dimensions of a multidimensional variable H and of the elements of the hierarchies, which consists in reorganizing the arrangement of the dimensions or axes of the dimensional space, based on a different point of view, which does not alter the semantics of the set of facts or the coherence of the summaries. The operation will be denoted by:

$$H' = P(H) = P((\vec{J}, \vec{m})) = (P(\vec{J}), \vec{m}) = (\vec{J}', \vec{m})$$

Where P represents the permutation of the k dimensions and, if necessary, the redefinition of the hierarchies of each dimension of the set. Thus, H' will be a new multidimensional variable, with another order in dimensions and hierarchies but with the same number of dimensions and the same scalar space of H . So H and H' are compatible.

Roll-Up: It is a process of aggregation of data that is carried out on a set of facts, reducing their dimensions or transforming their hierarchies, through the use of a classification space (CS), constituted by defined interval projections on the domains of the attributes of the hierarchies and an aggregation function F that acts on the scalar space measures of the set of facts.

The Roll-Up operator will be defined as:

$$\Phi_{CS,F}(H) = F(CS(H))$$

Slicing-Dicing: The purpose of the Slicing-Dicing operator is to allow a subset of the facts contained in a set of facts to be selected, based on a specific selection criteria defined through a selection predicate expressed in the traditional predicate algebra.

Formally, the Slicing-Dicing operator, by analogy to the selection operator of the relational model, is defined as:

$$\sigma_{\rho(x)}(H) = \{x \in H \mid \rho(x)\}$$

Where $\rho(x)$ is a conditional expression defined on the attributes of the hierarchies of the set of fact H , expressed on the basis of operators of the algebra of predicates.

Drill-Down: The purpose of this operator is to produce greater disaggregation in the multidimensional variable, by reincorporating dimensions to the variable or attributes in the hierarchies. In addition to eventually reincorporating dimensions, you can expand the hierarchies of some dimensions and make the previously summarized measures more specific, to obtain a cube with more complex hierarchies, more facts and probably more dimensions.

This operator rather responds to the approach of the relational calculation of tuples of the relational model, because to achieve its task it requires rather to declare the type of variable that is desired. The answer requires an aggregation on the original multidimensional variable, which contains the highest level of disaggregation of the data.

Formally, let H' be a multidimensional variable such that $H' = \Phi_{F,CS_1}(H) = F(CS_1(H))$ and let CS_2 be such that CS_1 is a subset of CS_2 , the operation called Drill-Down, denoted by:

$$\Theta_{F,CS_2}(H')$$

It is equivalent to performing the following sequence of operations:

$$\Theta_{F,CS_2}(H') \Leftrightarrow \Phi_{CS_2,F}(H)$$

Since CS_1 is a subset of CS_2 , the operation produces disaggregation in the data and generates a multidimensional variable of greater degree and cardinality.

3 Elements of data definition and data manipulation

MDX [16] is one of the most used specialized languages in formulating queries to multidimensional databases. However, despite being a very good tool, it requires users to be very familiar with the use of database query languages and must use a syntax that can sometimes be complex, not very intuitive, which also requires defining very explicitly the different processes and stages related to the generation of the result of a query, which for a non-specialized user can be a great obstacle. Greater coverage of query languages requires a simpler and declarative tool that exempts the user from a too procedural declaration of data processing.

Frequently multidimensional databases, often called data warehouse, are implemented on relational systems in a modality known as ROLAP. In this context, multidimensional database queries cannot be formulated using native syntax and semantics, which implies that they must finally be translated into relational queries, adapting their semantics to this type of technology.

The following sections present some advances related to the definition of a language that will be implemented in the form of a middleware, which will allow the generation, implementation and use of multidimensional databases under a transparent ROLAP mode for users. In this way the queries will be written in a syntax and a native semantics and the middleware will be responsible for their translation into relational queries.

3.1 The data definition language

The data definition language must provide all the tools necessary to define the structural components of the data model, that is, for the model presented in [14] [15], it is necessary to allow the specification of domains and attributes, dimensions, hierarchies, measures, facts and families of facts, as well as the definition of aggregation functions required for data manipulation. In this work we present the syntax and application of the sentence that allows the creation of a multidimensional variable:

In order to exemplify its use, we will apply it in the following conceptual model that is represented in the manner described in [9].

The creation of the multidimensional variable in Figure 1 is achieved with the following statement:

```
CREATE MULTIVAR [AS] name_variable
( DIMENSION name_dimension [AS] HIERARCHY (
    attribute_name data_type [IN] LEVEL n [,] ... ) [,] ... )
MESURES ( measure_name tada_type [,] ... )
```

Figure 1: The CREATE MULTIVAR statement that allows to create the structure of a multidimensional variable, specifying the dimensions, attributes, hierarchies of attributes and the set of measures.

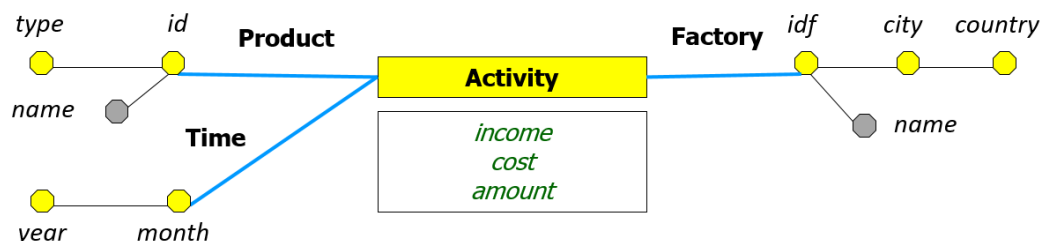


Figure 2: Multidimensional variable that records the income and costs related to the monthly production of a certain quantity of a type of products, in different factories of different countries (notation rescued from [9]).

3.2 The data manipulation language

The data manipulation language comprises different elements, including those that allow feeding the multidimensional variable. As it is a solution for data warehouse, it is assumed that data loading procedures are robust, therefore the data manipulation language should not require data update or deletion sentences (only for administrative aspects).

In a multidimensional database, queries always refer to data recovery, aggregation or disaggregation processes, related to OLAP's classic operators.

In the language whose advances are presented in this work, the operators are integrated in a single sentence, called AGREGATE, which are manifested as possible optional clauses in the consultation. The format of the sentence is as follows:

To exemplify its use, imagine that you must perform a sequence of OLAP operations on the multidimensional variable, using the following permutation:

$$h : DS \sim SS$$

And then, the following operations are performed:

1. $P(Activity)$, permutation is applied to the multidimensional variable Activity.
2. Then, the data for 2012 are selected by: $\sigma_{yy=2012}(Activity)$.
3. The data is classified according to the type of product and the city of the factory:

$$CS = Dom(city) \times Dom(factory)$$

```
CREATE MULTIVAR AS activity
( DIMENSION product HIERARCHY ( id CHAR(10) LEVEL 1,
    name CHAR(100) NON DIMENSIONAL, ptype CHAR(12) LEVEL 2 )
    DIMENSION time HIERARCHY ( mm MONTH LEVEL 1, yy YEAR LEVEL 2 ),
    DIMENSION factory HIERARCHY ( idf CHAR(12) LEVEL 1,
    name CHAR(100) NON DIMENSIONAL,
    city CHAR(50) LEVEL 2, country CHAR(50) LEVEL 3 ) )
MESURES ( income FLOAT, cost FLOAT, amount INT );
```

Figure 3: Sentence for the creation of the multidimensional variable in Figure 2 based on the CREATE MULTIVAR statement proposed in Figure 1.

```

AGGREGATE { USING [MAF] name_maf { (name_agg [,] ... ) |
                                     algebraic_expression [AS] name [,] ... }
FROM MULTIVAR name_mv
             [ name-dim (attribute_name [,] ...) [,] ... ]
[ WHERE conditional-expression ]
[ HAVING conditional-expression ]

```

Figure 4: AGGREGATE statement used to perform OLAP operations on the multidimensional variable.

4. The sum of income and costs is then calculated on the result using the following multiple aggregation function:

$$F(\text{Activity}) = (F_1(\text{Activity}), F_2(\text{Activity}))$$

Where F_1 and F_2 correspond respectively to the sum of the values of the income and cost attributes.

As can be seen, there are four different OLAP operations that are performed on the data set, including the classification of the data through a classification space CS .

The *AGGREGATE* statement allows the query to be carried out incorporating all the operations in a single instruction, acting on a multidimensional one, considerably simplifying the process of constructing the query.

<pre> AGGREGATE <i>sum(income), sum(cost)</i> FROM MULTIVAR <i>activity</i> [<i>factory</i> (<i>city</i>), <i>time</i>, <i>product</i> (<i>type</i>)] WHERE <i>yy = 2012</i> </pre>	<pre> SELECT <i>t.yy, t.mm, f.city, p.type,</i> <i>sum(income), sum(cost)</i> FROM <i>factory f, time t, product p, activity a</i> WHERE <i>a.idf = f.idf and t.date = a.date and</i> <i>p.id = a.id and yy = 2012</i> GROUP BY <i>f.city, t.yy, t.mm, p.type</i> </pre>
--	---

(a) With an appropriate multidimensional language.

(b) In SQL, emulating the multidimensional query.

Figure 5: Sequence of operations implemented as a native multidimensional query (case (a), AGGREGATE statement) versus SQL emulated query through the SELECT statement (case (b)).

Figure 5 allows comparing the implementation of the multidimensional operations described in section 3.2 in a native multidimensional language versus SQL. As it is possible to appreciate, the query formulated in a native language has a semantic load. In the first place, only the multidimensional object under study is mentioned, in this case the Activity variable, the classification criteria are specified in a simple and precise way, naming the dimensions, indicating in brackets the attributes to be considered in case the dimension considers a smaller hierarchy, to end by specifying the condition that the facts to be processed must meet ($yy = 2012$). In the case of the query in SQL, there is no multidimensional variable, the statement acts on three lookup tables: Factory, Time and Product, and the invoice Activity table. The conditions required for the correct combination of the tables run by the user and to them must be added the condition that the facts to be selected must meet. The classification space is diluted in a list of attributes in the GROUP BY clause that must explicitly mention all the attributes to consider. Permutation of dimensions (pivoting) is also subject to user responsibility.

3.3 Advantages of the language approach

The advantages associated with a language with the approach presented in this paper are many. They mostly derive from two very relevant characteristics: their greater semantic capacity compared to other multipurpose languages and their greater syntactic simplicity. In turn, these characteristics have other positive externalities:

1. Its simplicity and semantics allow generating conditions to increase the productivity of IT personnel, as well as generating greater closeness with end-users.

2. Reduce or eliminate the ambiguities that occur in the consultations, clearly separating the aspects related to the classification of the data regarding the calculation of indicators.
3. It allows a better link between the theoretical foundations of the data model, the concept of multidimensional database and the tools of definition and manipulation of data, a situation that is not possible to appreciate and incorporate it easily in contexts where multidimensional databases are emulated and the queries are translated into other conventional query languages.
4. The language is of general purpose for which it can be used in various fields, including management, production, and services.

3.4 Future work

The definition of language, however constituting a fundamental element for the construction of a fully functional tool, requires the implementation of software that respects the syntax of the language and incorporates its functional characteristics. The next stage of this work will be the development of a middleware that implements these characteristics in an interpreted environment, in which the data management will be carried out in a relational database management system. Although finally the database will correspond to a ROLAP implementation, both the definition of objects and queries to the databases will be done through the middleware, maintaining the user in the multidimensional environment, without having to worry about the translation of the queries. One of the initial applications of middleware will be in higher education, as a means to relate the design and modeling of multidimensional databases, with the use in information systems. It will allow students to experience the design, construction and consultation processes of a multidimensional database, in a native environment.

4 Conclusions

The syntax presented in this paper, shows that it is possible to considerably improve the syntactic simplicity and semantic capacity of interrogation languages to multidimensional databases. In this way, a natural treatment of dimensional objects is achieved in context, using a native syntax, more abbreviated, without the need to emulate the characteristics of multidimensional objects. On the other hand, the formulation of queries is simplified, it becomes more semantic, closer and more natural to end users, favoring access to information without the intermediation of too many IT professionals.

This language also reinforces the importance of the ROLAP approach and relational systems because it gives the opportunity to continue using the good characteristics of these systems, in a more automated and transparent way, allowing savings in licensing costs and investment in hardware, due to that in many cases it will be possible to continue using the available infrastructure.

Finally, it is a tool that has the potential to help in an important way to the development and use of support tools for decision making, which require the use of a multidimensional approach.

Funding

Acknowledgments to Mathematics and Computer Science Department, University of Santiago de Chile, USACH. To the School of Engineering, FinisTerra University, Chile, and also to de Department of Industry, Universidad Tecnológica Metropolitana, Chile.

Author contributions

The authors contributed equally to this work.

Conflict of interest

The authors declare no conflict of interest.

References

- [1] Bojičić, I.; Marjanović, Z.; Turajlić, N.; Petrović, M.; Vučković, M.; Jovanović (2016). A comparative analysis of data warehouse data models, In *2016 6th International Conference on Computers Communications and Control (ICCCC), Proceedings of, IEEE*, 151-159, 2016.
- [2] Bojicic, I.; Marjanovic, Z.; Turajlic, N.; Petrovic, M.; Vuckovic, M.; Jovanovic, V.(2017). Domain/Mapping Model: A Novel Data Warehouse Data Mode, *International Journal of Computers Communications & Control*, 12(2), 166-182, 2017.
- [3] Bouaziz, S.; Nabli, A.; Gargouri, F. (2019). Design a Data Warehouse Schema from Document Oriented database, *Procedia Computer Science*, 159, 221-230, 2019.
- [4] Boukra, D.; Boussaïd, O.; Bentayeb, F. (2010). OLAP operators for complex object data cubes, *Lecture Notes in Computer Science (Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 6295 LNCS, 103–116, 2010.
- [5] Codd, E. F. (1970). A relational model of data for large shared data banks, *Communications of the ACM*, 13(6), 377–387, 1970.
- [6] Colomo-Palacios, R.; Fernandes, E.; Soto-Acosta, P.; Larrucea, X. (2018). A case analysis of enabling continuous software deployment through knowledge management, *International Journal of Information Management*, 36(1), 142-154, 2016.
- [7] Cuzzocrea, A.; Moussa, R. (2017). Multidimensional database modeling: Literature survey and research agenda in the big data era, *2017 International Symposium on Networks, Computers and Communications (ISNCC)*, 1–6, 2017.
- [8] Eroshkin, S. Y.; Kameneva, N.; Kovkov, D.; Sukhorukov, A(2017). Conceptual system in the modern information management, *Procedia Computer Science*, 103(C), 609-612, 2017.
- [9] Golfarelli, M.; Rizzi, S. (1999). Designing the Data Warehouse: Key Steps and Crucial Issues, *Journal of Computer Science and Information Management*, 2(3), 1–14, 1999.
- [10] Hümmer, W.; Lehner, W.; Bauer, A.; Schlesinger, L. (2002). A Decathlon in Multidimensional Modeling: Open Issues and Some Solutions, *Proceedings of the 4th International Conference on Data Warehousing and Knowledge Discovery*, 275–285, 2002.
- [11] Jaroli, P.; Masson, P. (2012). Data Warehousing and OLAP Technology (Data warehousing), *International Journal of Engineering Trends and Technology*, 2, 955-960, 2012.
- [12] Moole, B. R. (2003). A Probabilistic Multidimensional Data Model and Algebra for OLAP in Decision Support Systems, *Conf. Proceedings - IEEE SOUTHEASTCON*, 18–30, 2003.
- [13] Pablo, P. V. (2016). Business intelligence applied to monitoring and meta-monitoring scenarios, In *2016 11th Iberian Conference on Information Systems and Technologies (CISTI)*, 1-6, 2016.
- [14] Palominos, F. E.; Duran, C. A.; Córdova, F. M. (2018). Multidimensional data model for the analysis of information of productive, scientific or service processes, In *7th International Conference on Computers Communications and Control*, IEEE, 17–22, 2018.
- [15] Palominos, F. E.; Duran, C. A.; Córdova, F. M. (2019). Improve efficiency in multidimensional database queries through the use of additives aggregation functions, *Procedia Computer Science*, 162, 754–761, 2019.
- [16] Piasevoli, T.; Li, S. (2016). MDX with Microsoft SQL Server 2016 Analysis Services cookbook, 2016.
- [17] Rahimi, F.; Møller, C.; Hvam, L (2016). Business process management and IT management: The missing integration, *International Journal of Information Management*, 40, 186-189, 2018.

- [18] Taleb, A.; Eavis, T.; Tabbara, H. (2011). The NOX OLAP query model: From algebra to execution, *Lecture Notes in Computer Science*, 6862 LNCS, 167–183, 2011.
- [19] Zhang, X. (2018). Design of Intelligent Management Decision Support System for Retailing Chains, In *2018 International Conference on Virtual Reality and Intelligent Systems (ICVRIS), Proceedings of*, 485-489, 2018.
- [20] Zykin, S. V.; Mosin S. V.; Poluyanov A. N. (2019). Technology of Multidimensional Data Formation Using Caching, *13th International IEEE Scientific and Technical Conference Dynamics of Systems, Mechanisms and Machines, Dynamics 2019 – Proceedings*, 1-10, 2019.



Copyright ©2020 by the authors. Licensee Agora University, Oradea, Romania.

This is an open access article distributed under the terms and conditions of the Creative Commons Attribution-NonCommercial 4.0 International License.

Journal's webpage: <http://univagora.ro/jour/index.php/ijccc/>



This journal is a member of, and subscribes to the principles of, the Committee on Publication Ethics (COPE).

<https://publicationethics.org/members/international-journal-computers-communications-and-control>

Cite this paper as:

Palominos, F.; Córdova, F.; Durán, C.; Nuñez, B. (2020). A Simpler and Semantic Multidimensional Database Query Language to Facilitate Access to Information in Decision-making, *International Journal of Computers Communications & Control*, 15(4), 3900, 2020.

<https://doi.org/10.15837/ijccc.2020.4.3900>