



**Repositorio Institucional de la Universidad Autónoma de Madrid**

<https://repositorio.uam.es>

Esta es la **versión de autor** de la comunicación de congreso publicada en:  
This is an **author produced version** of a paper published in:

EICS '10: Proceedings of the 2nd ACM SIGCHI symposium on Engineering  
interactive computing systems. ACM, 2010. 187-192

**DOI:** <http://dx.doi.org/10.1145/1822018.1822047>

**Copyright:** © 2010 ACM

El acceso a la versión del editor puede requerir la suscripción del recurso  
Access to the published version may require subscription

# Feasible Database Querying Using a Visual End-User Approach

**Clemente R. Borges**

EPS – Universidad Autónoma de Madrid  
Avda. Tomás y Valiente, 11  
28049 Madrid Spain  
clemente.borges@uam.es

**José A. Macías**

EPS – Universidad Autónoma de Madrid  
Avda. Tomás y Valiente, 11  
28049 Madrid Spain  
+34 91 4972241  
j.macias@uam.es

## ABSTRACT

Querying databases is a common daily task carried out by a great deal of end-users who do not have specific skills in SQL language. Today, most of the database interaction is achieved by means of query interfaces provided by the database environment. However, most of these interfaces suffer from expressive limitations, since they are mostly based on metaphors that drastically restrict the expressiveness of the SQL language that is generated and executed in the background. In this paper, we present a visual interaction language and tool focused on easily querying databases by end-users. We make no assumption on the level of the user's experience with query languages, as our visual metaphor is intended for querying databases by unskilled end-users and also leveraging the restriction on the expressiveness of the queries created by them. We also report on some late braking results obtained by an experiment carried out with real users.

## Author Keywords

Data Warehouse, End-User Development (EUD), Usability, Visual Interfaces, Web-Based Interaction.

## ACM Classification Keywords

H.5.2 User Interfaces: Graphical User Interfaces, User-Centered Design and Prototyping. H.2.3 Data Manipulation Languages.

## General Terms

Design, Experimentation, Languages.

## INTRODUCTION

Interacting with databases has become commonplace in most end-user activities today. In the past, such interaction was carried out using sentences in textual languages that the database expert had to provide to the application using the

system prompt and/or executing programming code to obtain the desired outcome.

Nowadays, modern computing advocates the idea of people using and creating databases in order to accomplish their daily tasks. They are professionals such as engineers, scientists and freelances that may have concrete domain skills but lack programming abilities. In this regard, further support is needed in order to provide such non-programmer professionals with easy-to-use mechanisms, avoiding the need for them to learn programming languages and specifications that are usually deemed to be irrelevant for their daily work activities

Database research has recently produced notable results related to architectural and storage facilities, likely paying less attention to interaction. This has enabled databases systems to deal with large volumes of information for different purposes (such as distribution), but decreasing, however, the facility for end-users to easily access and manipulate information.

In the last decade, some database systems have included visual tools intended to manipulate models and data for supposed unskilled end-users. The existence of visual tools helps reduce the gap between users and the information they desire to query. However, the creation of good visual tools for database query is not such an easy task. Most of the existing query environments lack both visual expressiveness and/or ease of use – an essential trade-off that most end-user-intended applications should guarantee today [11]. This lack of expressiveness sometimes implies that some capabilities related to build complex queries are missing, or they are hard to be accomplished by end-users. For instance, some systems are based on a table-like metaphor that severely restrict the capability to deal with aggregation conditions, filters, and complex operations such as UNION, INTERSECTION, IN and NOT IN. On the other hand, the usability assurance in such query interfaces has been barely addressed as, in most of the cases, neither user-centered techniques have been used nor user mental model has been considered throughout the design.

In order to overcome the aforementioned drawbacks, we have created a visual query tool called VISQUE (VISual

Star-schema QUery by Example) that takes up the challenge of creating queries (specifically in *data warehouse schemas*) by example, empowering the way users can create different kinds of queries with minimal expressive limitations, and carrying through an explicit ease of use to achieve a better user experience. Our system exploits different knowledge-visualization techniques including a tree-based metaphor to easily represent the *measures* and *dimensions* of a *multidimensional* database schema, providing expressiveness enough to create complex SQL-based constructions such as set-based, nested and aggregation queries.

The main objective of our work is twofold:

- To provide end-users with the capability to create database queries, assuming no skills in SQL statements, enhancing the ease of use and so reducing the cognitive load during the interaction.
- To provide a less restrictive metaphor by which end-users can create different kinds of queries; including set-based and aggregation operations, which are not very common in other table-based database query environments.

In order to support the above objectives, we have built our authoring tool by means of using user-centered techniques such as Competitive Analysis [13] and Paper Prototyping [15]. Also, we have evaluated the current prototype by means of an experiment achieved with real users, also utilizing user-centered techniques such as the Retrospective Test [13] and usability evaluation through the USE questionnaire [10].

The rest of this paper is organized as follows. In the next section, we will present related work addressing some of the most common tools related to our approach. Next, an overview of our system is given on the basis the of the DW schema composition and query. Then, late breaking results from our last experiment with real users will be introduced in order to discuss the most relevant outcome obtained. Finally, we outline some conclusions and future work.

## RELATED WORK

In some way, the usability of a database system is as important as its capability [4], which means that the information should be organized and presented to end-users in a natural and understandable way, enabling to manipulate the data in an easy manner. Visual-query interfaces have been proposed in the past to assist users in building queries for different domains. Query by example (QBE) [20] has been the inspiration for several visual query tools (such as Microsoft Access or SQL-Server). However, QBE uses a visual table-like metaphor that is not able to deal with set-based or nested operations in a visual way. A different approach is presented in Spago BI QbE Enterprise Scope [17], which proposes the creation of queries from a fixed sequence of steps, but it still lacks the visual richness and variety of (relational) query operations. There are other

open source tools like Eclipse Birt Project [3], which also provides facilities for creating queries, but it lacks of visual expressiveness and is not suitable for unskilled users since it demands to directly manipulate SQL code. Additionally, Tableau Desktop [19] is the commercial version of a research tool called Polaris [18], which is mainly focused on data warehouse schemas. However, this tool uses a method of classification for measures and dimensions based on the type of data related to the fields. By contrast, VISQUE analyzes the structure of the model and, by using a worked algorithm, it determines the fact table with the corresponding measures, as well as the dimension tables having the dimensional attributes. Visual languages are frequently used to facilitate the creation of queries by end-users. The main goal is to facilitate the use of predefined domain concepts, as visual representations drastically reduce the abstraction level required to create programmatic representations [12]. Our approach attempts to complement other visual tools in the area, adding a novel visual representation of data warehouse schemas and including graphical representations of set-based and nested query operations that are commonly infrequent or even unavailable in other approaches.

## OUR CONTRIBUTION

Our approach is based on End-User Development (EUD) [8] paradigm. EUD is focused on a user-centered approach, and can be thought of as a set of activities and techniques that allow people (including non-professional) to create or modify software artifacts or complex data. It has been demonstrated that End-User Development techniques reduce the gentle slope of complexity and make it easy the way the user accomplishes tasks by means of computers. More specifically, our approach is based on Programming by Example [1, 7] which aims at obtaining a satisfactory trade-off between ease-of-specification and expressiveness. Programming by Example has the potential to allow users to create programmatic functionality naturally. Rather than writing a statement in SQL or dealing with abstract specifications, users simply demonstrate how to perform actions to create a database query and the system automates the creation and execution of the SQL code in the background.

Additionally, we have based on relational database systems. Specifically, we focused on the multidimensional data model, which is a fundamental modeling technique for Data Warehouses (DW). This has provided an invaluable support as a successful model to carry through data storage and exploitation. Multidimensional models simplify the manipulation of data by users, avoiding complex relationships and providing an easier way to access the required information. The collection of tables in a DW is called a schema. A common schema consists of a fact table and several dimension tables that are related to the fact table. The fact table represents the business measures (numeric values), whereas the dimension tables contain attributes that describe those measures in different

perspectives (textual descriptions). This type of structure is commonly known as a star schema, and one of its most important features is its simplicity. Users who use a star schema are benefiting since the data are easier to understand, navigate and query [6].

In order to provide the facilities previously commented and corroborate the objectives mentioned we have built VISQUE – a web-based tool that enables end-users to visually carry out queries in different database systems.

As shown in Figure 1, VISQUE is based on a client-server architecture that comprises two main parts. The front-end is a web-based authoring tool that allows users to visually interact with data elements and compose queries by example. On the other hand, the back-end includes the main functionalities of the application, and it is devoted to infer and translate the appropriate information and communicate with the database server to execute the queries and extract data.

To interact with VISQUE, first the system establishes a connection with the database server specified by the end-user, and then the tool starts to interchange data with it. VISQUE automatically sends the SQL code internally generated by the tool and then captures the result sent back by the database server in order to be displayed on the client browser. Instead of using SQL language, end-users manipulate graphical objects by means of a visual language created to improve the interaction, reduce the learning burden and adapt to the user’s mental model as much as possible. Each time the user manipulates any object, the system updates the internal information and composes a query, sends the code to be executed to the server and, once received, it finally displays the resulting data. This way, there is a minimal possibility of making an error and the user always has appropriated feedback and an updated version of both query and results.

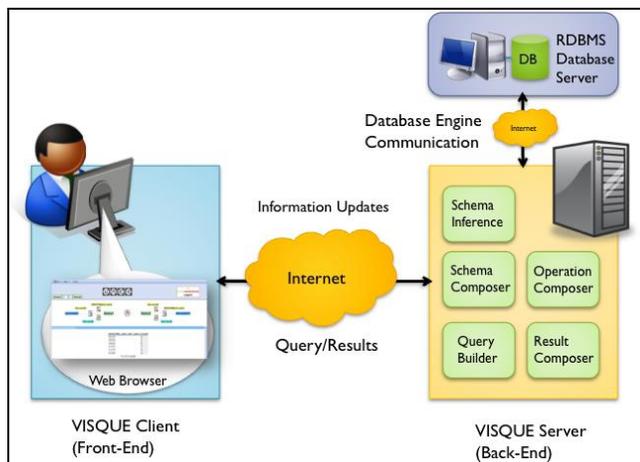


Figure 1. Main components in VISQUE.

Figure 2 depicts the user interface of our tool, which comprises five different main sections (each one represented by numbered arrows in Figure 2). (1) The first

one indicates the interaction with the star schema and the way the user can manipulate the fields, selecting them from a hierarchical structure that represents the measures (green rectangles) and dimensions (red rectangles) by using different colors and metaphors, and allowing to select (by expanding the nodes) the kind of operation that the user wants to apply (selection, filtering, ordering, etc.). (2) The second section is a box-like representation of the selected elements from the tree structure. (3) The third section is a tabbed pane representing each sub-query independently, in order to make up the global one, indicating the corresponding operations between queries, and allowing the user to see the elements of the resulting operation in a visual way. (4) The fourth section is automatically updated according to the information appearing in other sections, and it shows the data results in real-time for each query created; the data are obtained from the server whenever the query is executed. Finally, (5) the fifth section shows different icons that allow performing set-based and nested query operations. One of our tool’s more remarkable characteristics is the ability to visually represent operations such as UNION, INTERSECTION, IN and NOT IN, which is not very common in other visual query tools.

The visual representation of the schema (arrow numbered by 1 in Figure 2) is based on an XML file that includes a description of the fact table with the corresponding measures as well as the dimension tables with the attributes fields. All the information in this file is crated from a relational data source provided by the user and then is processed to obtain a JSON [5] structure that feeds the visual representation displayed in the browser. The visual presentation and interaction with the schema is based on the Spacetree model [16], which comprises an interactive tree-like component that automatically rearranges, collapses and expands elements visually in a very easy way.

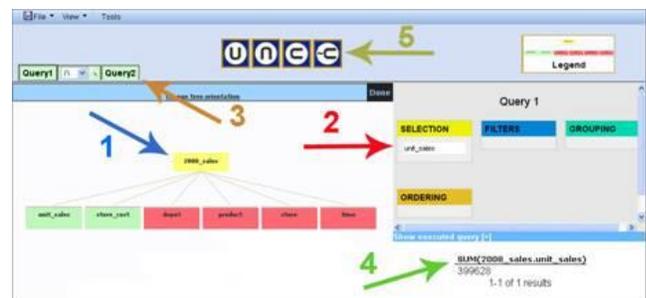


Figure 2. VISQUE user interface overview.

As mentioned, one of the most interesting issues concerning our tool is the ability to work with set-based operations (union, intersection and so on). To realize this challenge, we have changed the traditional table-like metaphor, which has several expressive limitations, to a graphical one. This visual metaphor is shown in Figure 3, which depicts a visual representation of the intersection of two queries and the final results obtained from this operation. For each sub-query depicted on the left and right of the intersection

symbol, one can see elements that indicate, using the same color metaphor, the operations that were selected and, in turn, each object showing the fields used. Also, for each operation selected, one can see the function that was applied in the selection of facts, the conditions and values for applying information filtering, and the selected field for grouping. Additionally, we can observe, at the bottom, an area showing the results, which represent the intersection between two queries. The user can manipulate all these objects in order to change their positions or values; the results will be automatically updated by the system.

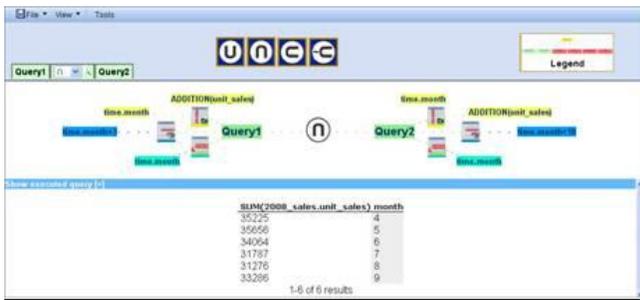


Figure 3. Intersection operator and query results.

Additionally, VISQUE is able to adapt certain characteristics of the working environment depending on the user’s experience. The first time the user utilizes the tool, it is mandatory to answer a set of questions to determine the expertise of the user concerning database usage in order to adapt the visualization for a particular query construction.

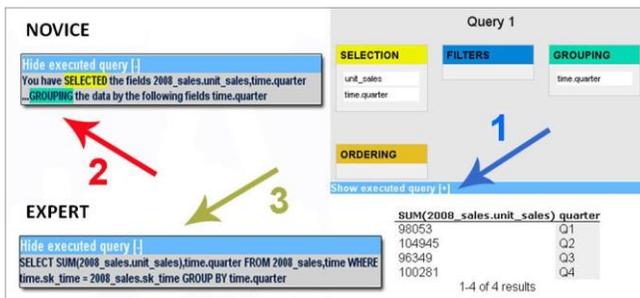


Figure 4. Different views for a query.

Figure 4 shows an example of adaptation in a query representing the units sold in each quarter. Each time the user makes a change to the query, s/he is able to observe a textual sentence representing what has been built. To do this, one must click on “Show executed query” (1), which automatically expands an area showing the mentioned sentence. In the case of an end-user having a novice profile, the tool translates the SQL sentence into a pseudo language that describes the query in a textual fashion (2), decreasing the abstraction level, increasing the usability, and allowing the user (who does not know SQL) to understand the sentence that has been executed. The pseudo language is an attempt to textually describe the boxes representing the selected elements by using the same words and colors used

on them. On the other hand, if the user has an expert profile, then the application shows exactly the same SQL sentence that has been executed on the database management system (3). This feature helps reduce the learning burden and increase the overall ease of use of the system, which can be used even as an educative tool.

### LATE BREAKING RESULTS ON EXPERIMENTAL ISSUES

To carry on with our user-centered approach, we wanted to have an initial feeling about the general usability of the system, in order to work it and improve it further.

To carry out this task, we have conducted a user test with 12 real users having domain knowledge on database systems but not on SQL language. The experiment consisted in the creation of different queries by using VISQUE, representing a notable variety of set-based and nested operation in order to extract reliable information from the experience. Users were instructed on the tool for about 10 minutes before performing the test. After the training, it was provided a new data model on which users had built the selected queries. By using the Retrospective Testing [13], all the user interaction was recorded on video and audio. This helped obtain valuable interaction information in order to improve the tool as well.

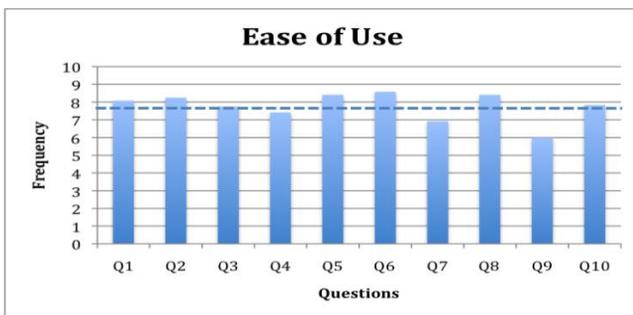
Once the interaction session with the tool was finished, end-users were requested to fill in a questionnaire based on the USE survey [10] but including some variations provided by the Purdue Usability Testing [9] and the Perceived Usefulness and Ease of Use [2] questionnaires. We based our study on obtaining the user’s satisfaction related to the ease of use of VISQUE, in order to clearly corroborate our initial hypotheses stated. This way, we considered the 10 main questions from category “Ease of Use”:

- Q1: the tool is easy to use.
- Q2: the tool is simple to use.
- Q3: the tool is user friendly.
- Q4: the tool requires the fewest steps possible to accomplish what I want to do with it.
- Q5: the tool is flexible.
- Q6: using the tool is effortless.
- Q7: I can use the tool without written instructions.
- Q8: don't notice any inconsistencies as I use the tool.
- Q9: both occasional and regular users would like the tool.
- Q10: I can use the tool successfully every time.

The results obtained from this questionnaire can be shown in Figure 5. The questionnaire was constructed using a 10-point Likert rating scale, ranging from 0 (strongly disagree)

to 10 (strongly agree) for each question. The average value is represented as a dotted line in the bar graph.

In general, we can observe good results for most claims. Questions Q1 and Q2 indicate similar results as they show the degree of agreement about the ease of use of the tool. These reflect that 100% of users have answered from 6 onwards, indicating a high perception concerning that the tool is not very complicated to use. Also, Q5: “the tool is flexible” and Q6: “using the tool is effortless” provide 100% of results with score greater than 7, which emphasizes the ease of use of our tool once again. By contrast, Q9 obtained a high level of dispersion in the answers, indicating that users probably felt that a casual user may need some initial help to use the tool self-sufficiently, in the same way as they were given an induction of VISQUE earlier in the training session.



**Figure 5. Results for “Ease of use”.**

All in all, most of the scores, including the specific questions discussed above, reflect positive values. The average for “Ease of Use” is 7.76 with a standard deviation of 0.77, indicating a high level of acceptance and an overall low degree of dispersion in the answers. In general, most of the scores show positive results, so that we can state that the users think of VISQUE as an easy-to-use tool. We also included in the questionnaire a set of open questions related to general satisfaction. Answers revealed that the tool facilitates and speeds up the task of extracting valuable information from Data Warehouses in an easy manner. All these arguments corroborate the initial hypotheses stated.

## CONCLUSION

We have presented an overview of our work, which is an attempt to deal with the problem of visualization and query in data warehouse schemas, taking into account usability features and feasibility in an explicit expressiveness/ease-of-use trade-off. Our work is aimed at supporting easy web interaction with databases for the end-users to obtain valuable information from them. The solution provided is a combination of End-User Development techniques, web-based user interface design and data models, all combined in an interactive environment called VISQUE.

The multidimensional model used helps reduce complex relations and simplifies the user’s mental model of both data and interaction. Additionally, visual representations

drastically reduce the abstraction level and support user interaction easily.

The idea of visual tools intended for database is not new at all. However, we have found deficiencies in existing applications that have been collected using heuristic evaluation of (partial) usability as well as comparative/competitive analysis commonly used in the analysis phase when designing interactive software [13]. Also, comparative analysis is useful to analyze the cognitive load and the sequence of steps typically used in data warehouse queries, checking the operations missing and noticing those that have usability problems. In this sense, our approach overcomes other systems providing the following features that outperform, on average, the tools previously discussed:

- Automatic inference of the data schema.
- Greater expressiveness and interaction in the representation and manipulation of data warehouse schemas, improving the ability to suggest possible values for filters, and supporting an expressive and interactive way to represent the data schema
- New features to build and represent, visually, different types of queries supporting the visual presentation and interaction with operations such as UNION, INTERSECTION, IN and NOT IN.
- Improved feedback, suggesting possible values in aggregation conditions, conditional clauses and controlling errors on specific data types.
- Automatic representation of the query results in real time, even from the very beginning, as the end user builds the query.
- Adaptive representation of the SQL statements according to the user profile.

For the time being, we have the definition of the visual language as well as a functional prototype of VISQUE that has been validated through several use cases. Furthermore, the test carried out with real users allowed evaluating the tool. In particular, it enabled to obtain a general perception about the ease of use. Partial correlations calculated using other dimensions (i.e., Usefulness, Satisfaction and Ease of Learning) suggested that, while both drive Satisfaction, Ease of Use and Usefulness influence one another, such that improvements in Ease of Use improve ratings of Usefulness and vice versa. In addition, Satisfaction is strongly related to the usage (actual or predicted). The items contributing to Ease of Use could be separated into two factors, Ease of Learning and Ease of Use, which are obviously highly correlated [10]. In particular, the analysis of the questionnaire reported positive results about the ease of use of the tool, which empowers to corroborate the initial hypotheses and contribute to the usability assurance in our tool.

VISQUE has been built using user-centered techniques such as Comparative Analysis, Retrospective Testing (including Thinking Aloud [14] sessions) and Paper Prototyping, a common interaction analysis technique that enabled us to capture the user's mental model matching the behavior and appearance of the web user interface very early in the development process.

Additionally, we are still studying different possibilities for the visualization of other complex query operations not covered by existing approaches. Finally, we plan to improve the tool further from the feedback obtained from the user test, as well as to achieve some additional experiments to validate the new improvements.

#### ACKNOWLEDGMENTS

The work reported in this paper is being partially supported by the founded projects TIN2008-02081/TIN and S2009/TIC-1650.

#### REFERENCES

1. Cypher A. Watch What I Do: Programming by Demonstration. The MIT Press, 1993.
2. Davis, F. D. Perceived Usefulness, Perceived Ease of Use, and User Acceptance of Information Technology, MIS Quarterly, Volume 13, Issue 3, September 1989, pp. 319-340.
3. Eclipse Birt Project. <http://www.eclipse.org/birt/>.
4. Jagadish, H.V., Chapman, A., Elkiss, A., Jayapandian, M., Li, Y., Nandi, A. and Yu, C. Making database systems usable. Proceedings of the International Conference on Management of data. Beijing, China (2007), 13 – 24.
5. JavaScript Object Notation. <http://www.json.org>.
6. Kimball, R. and Ross, M. The Datawarehouse Toolkit: Second Edition. John Wiley and Sons Inc., Hoboken, USA, 2002.
7. Lieberman, H. Your Wish is my Command. Programming By Example. Morgan Kaufmann Publishers. Academic Press, USA, 2001.
8. Lieberman, H., Paternò, F., and Wulf, V. (eds). End-User Development. Human Computer Interaction Series. Springer Verlag, 2006.
9. Lin, H., Choong, Y., Salvendy, G. A Proposed Index of Usability: A Method for Comparing the Relative Usability of Different Software Systems, Behaviour and Information Technology, Volume 16, Issue 4 & 5, Informaworld, July 1997, pp. 267 - 277.
10. Lund, A. Measuring Usability with the USE Questionnaire. Usability and User Experience Special Interest Group, Volume 8, Issue 2, October 2001.
11. Macías, J.A. and Castells, P. Providing end-user facilities to simplify ontology-driven Web application authoring. Interacting with Computers. Elsevier, Volume 9, Issue 4 (July 2007), 563-585.
12. Moya, D. and Macías, J.A. Semantic Web Query Authoring for End-Users, in M. A. Redondo, C. Bravo and M. Ortega (eds.), Engineering the User Interface: From Research to Practice. Springer Publisher, New York, USA (2009), 147-160.
13. Nielsen, J. Usability Engineering. Morgan Kaufmann Publishers, San Francisco, USA, 1993.
14. Ramey, J., Cuddihy, E., Dumas, J., Guan, Z., Van, M., Jong, M. Montreal: Does think aloud work?: how do we know. ACM, 2006, CHI '06 extended abstracts on Human factors in computing systems, págs. 45 - 48.
15. Rettig, M. Prototyping for tiny fingers. Communications of the ACM, Volume 37, Issue 4 (1994), 21-27.
16. Spacetime. JavaScript InfoVis Toolkit for information visualization. <http://thejit.org/>.
17. Spago BI QbE Enterprise Scope. <http://www.spagobi.org>.
18. Stolte, C., Tang, D. and Hanrahan, P. Polaris: a system for query, analysis, and visualization of multidimensional databases. Communications of the ACM, Volume 51, Issue 11 (2008), 75-84.
19. Tableau Desktop. <http://www.tableausoftware.com/products/desktop>
20. Zloof, M.M. Query by example. Proceedings of the National Computer Conference. New York, USA, Vol. 44 (1975), 431-438.