A Brave New World

Sónia Patrícia F. C. B. Quaresma Gonçalves

Information Systems and Computing Department
National Statistical Institute
Portugal

sonia.quaresma@ine.pt

Abstract: Every time an NSI launches a new survey, several services have to be prepared to deal with it. In particular its databases must be able to accommodate the incoming data. Currently, we use very large databases called Data Warehouses to store this information. The value of the data warehouses is their ability to support business intelligence. These specialized analytical databases typically provide support for complex multidimensional calculations and data aggregations, and are able to perform them in an acceptable amount of time. The data warehouses' capabilities for data dissemination is regarded as an important contribution to quality. However, as we will show, data warehouses can and should be making greater contributions for the overall quality in statistical institutes. It's through the process we will explain in this paper that simple data shall be transformed into meaningful information, comparable not only across the dimensions of its own project, but also with transverse variables whose source could be distinct surveys. When integrating information, sometimes we have to recover old information, which is difficult, and we may have to work with different classifications, trying to build bridges between projects and helping people to compromise. Each case is unique!

It is a hard process for all the people involved but it is worth the effort, because this is the approach that will lead us to a new perception of reality, and will ultimately help us build our brave new world.

Keywords: Data Warehouses, Improving Process Quality, Recommended Practices

1. Introduction

For analysis purposes data is stored in Data Warehouses (DWS), which contain the historical, integrated data of all the company, used by the analysts to make decisions. To bring that data near analysts, OLAP (On-Line Analytical Processing) tools appeared. By means of multidimensionality, this kind of tools allows non-expert users to formulate their own queries and obtain the results interactively (without assistance from the IT department).

In this paper we firstly address the differences between OLTP and OLAP databases in section 2, making a quick review of the role databases play in statistical offices and how they evolved to satisfy the institutions needs.

A data warehouse has three distinct components: static, dynamic and evolutionary. They will be presented in section 3, with particular emphasis in the dynamic component or ETL Process. We'll describe the problems we have been facing in the last years, explaining how we solved them and giving examples from the Portuguese NSI.

In section 4 we present our conclusions.

2. Why Data Warehouses?

To understand the present necessity for Data Warehouses and OLAP tools, we will briefly discuss their evolution and the role they currently play in NSIs.

Statistical Offices always produced a lot of data and as technological solutions provided ways to store this data they quickly embraced them.

In the early days, long before relational databases, when we first transferred statistical data to a computer, the original paper survey was captured as a single enormous record with many fields. Such a record could easily have been 1,000 bytes in size distributed across 50 fields. This era can be called the Storing Ages – all our problems consisted in storing the data, and the computer was extremely useful in solving them.

However, several problems arose when manipulating this data. It was difficult to keep consistency because each record stood on its own. For example, an enterprise's name and address appeared many times, in the same and in different surveys, and small differences or misspellings were hard to detect and solve. These inconsistencies in the data were rampant, because all the instances of the name and address were independent, and updating any of this data was a messy transaction (database operation).

As the attention shifted to transaction processing, the community's efforts concentrated on eliminating redundancy from the databases. Entity/Relationship models and normalization rules were developed and the On-line Transaction Processing (OLTP) was born.

The Transactional Ages lasted a long time and will never really end, as transaction processing is always needed. However the main concern shifted again from easy manipulation and storing towards querying and efficiently retrieving the data. The Data Warehouses and On-line Analytical Processing (OLAP) approaches appeared to solve those problems.

In the industry, querying databases was always the main goal, particularly in institutions like statistical offices whose purpose is to disseminate statistic information. Retrieving the information previously stored in the database was of capital importance. However, the relational databases used during statistic production or data collection have dozens of tables linked together by a spider web of joins, and do not reflect the users' understanding of the facts.

Another kind of database had to be created; data warehouses. Why?

1. Because the data has to be stored in the way best suited for the users to analyze.
2. Because the response time has to be satisfactory, which in most cases means that complex processing and preparation of data is needed before populating the table.
3. Because multiple data sources have to be taken into account without increasing the complexity of the query for the user.

So for analytical purposes data is stored in DWs, which contain the historical, integrated data of all the company, used by the analysts to make decisions.

To bring the data to the analysts, OLAP (On-Line Analytical Processing) tools appeared. By means of multidimensionality, this kind of tool allows non-expert users to formulate their own queries and obtain the results interactively (without the assistance of the IT department).

The multidimensionality is based on the duality of facts-dimensions. A fact represents a subject of analysis, while its dimensions show the different points of view we can use to study it. To support this multidimensionality the database has to be designed in a special way.

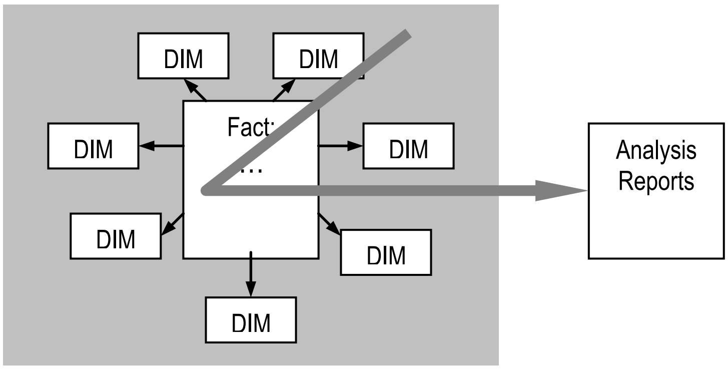

Figure 1 - The Star Schema

In the logical design phase, the data should be represented by means of a star schema, having one central Fact Table (containing measures) surrounded by multiple Dimension Tables (containing discrete descriptive attributes) [1].

The data is organized in cubes, which are defined over a multidimensional space, consisting of several dimensions. Each dimension comprises a set of aggregation levels. Typical OLAP operations include the aggregation or de-aggregation of information along a dimension (roll-up or drill-down), the selection of specific parts of a cube (slicing) and the re-orientation of the multidimensional view of the data on the screen (pivoting) [2].

A good definition of OLAP would be:

"...On-Line Analytical Processing (OLAP) is a category of software technology that enables analysts, managers, and executives to gain insight into data through fast, consistent, interactive access to

a wide variety of possible views of information that has been transformed from raw data to reflect the real dimensionality of the enterprise as understood by the user.

OLAP functionality is characterized by dynamic multidimensional analysis of consolidated enterprise data supporting end user analytical and navigational activities including calculations and modelling applied across dimensions, through hierarchies and/or across members, trend analysis over sequential time periods, slicing subsets for on-screen viewing, drill-down to deeper levels of consolidation, rotation to new dimensional comparisons..." [3].

3. Building a Data Warehouse

A Data Warehouse is a collection of technologies aimed at enabling the knowledge worker to make better and faster analysis and to support subsequent decisions. It's widely accepted that Data Warehouse architecture can be formally understood as composed by layers of materialized views on top of each other [4]. The DW comprises three different aspects:

- Static – the structure that supports the data.
- Dynamic – the processes used to populate the physical structure.
- Evolution – the data refreshment and administration processes.

Several layers compose the DW's architecture. Data from one layer is derived from data in the lower layer. Data sources form the lowest layer. They can be structured data stored in operational databases, or unstructured/semi-structured data stored in files. We refer this data as micro data.

An optional layer is the Operational Data Store (ODS), which serves as buffer for cleaning and transforming data.

The Data Warehouse itself is built on top of the data sources or the ODS when available, aggregating the detailed data from the lower levels. The static component of DWs resides mainly in this central layer and stores macro data.

The top layer is formed by specialized datamarts and OLAP databases targeted to very specific users and containing highly aggregated data. OLAP is a trend in database technology, based on the multidimensional view of data, which is employed at the client level [4].

3.1 Data Warehouse Databases

The emergence of data warehousing was initially a consequence of the observation that OLTP and OLAP reflected different needs and could not coexist efficiently in the same database environment. Different techniques and database features have been introduced to deal with specific data warehouse problems.

The first major problem was that the tables were too big. For instance if we're storing data from a monthly survey and each month we receive 250.000 answers, by the end of the year we will have stored 3 million records. In five years we will have 15 million records and 5 years is a standard period for a time series analysis.

Huge amounts of data have to be readily available to provide quick answers to any ah-doc query of the users. Table partitioning has been introduced to deal with these large tables, providing more manageable structures with the consequent performance benefits.

This larger table usually stores only the facts, or the answers to our surveys, but dimension tables containing the classifications have to be created also. Each fact may have several measures (usually numeric [5].). For example, the sales fact may have the sales value in euros and the weight of the merchandise in kilograms. Besides these numerical attributes, each fact is characterized by its dimensional attributes. As explained in section 2, facts are analyzed with regard to data in the dimensions. These tables "surround" the fact table and allow us to decode the facts, and also to perform the aggregations or de-aggregations along the classification explicitly defined in the dimension.

Dimensions usually have associated with them hierarchies that specify aggregation levels and hence the granularity in which the data is viewed. There is no formal way of deciding which attributes should become dimensions and which attributes should become measures. It is left as a database design decision.

If the attributes intrinsically define a hierarchy, they probably form a dimension. If they do not and are numeric values than we're probably dealing with measures. The cardinality of the tables can be the basis of another good heuristic; dimension tables are significantly smaller than fact tables.

Once we've distinguished dimension attributes from fact measures, other database design features are of concern. More often than not the dimension tables will be small, absurdly small when compared with the unique fact table. This can lead us to think that performing joins between the tables should not be a heavy operation upon the database. Let's not be fooled by this apparent smallness. All the dimensions have to be related with the facts table and not among themselves. Specifically for data warehousing environments, where data updates are less frequent and ad-hoc queries are more common, bitmap indexes [6], and bitmap join indexes have been developed providing both performance benefits and storage savings.

Other technological solutions to data warehouses are materialized views, clustering and complex queries optimization.

These database features haven't always been available, so even the database design has to change in time to encompass the technical advances.

In the last year an effort was made at the Portuguese NSI to update all the data warehouse structure in order to take advantage of the most recent techniques available. We were successful on our task and the change proved fruitful. As an example, typical exploration of "External Trade" data, which took as long as 3 minutes, is now executed in 20 seconds.

3.2 ETL Process

Inmon defined data warehouses as "*subject-oriented, integrated, time-varying, non-volatile collections of data that is used primarily in organizational decision making."*[7].

Data warehousing became an important strategy to integrate heterogeneous information sources in organizations, and to enable their analysis. This happens because data from data sources or ODS is cleaned, transformed, aggregated and integrated to provide more accurate and useful data to the user.

The data processing chain is known as Extraction, Transformation and Loading Process (ETL). The first phase is where we collect all data from the various data sources and ODS. The methodology followed is usually data-driven, this means that all available data, from survey's or administrative sources, is retrieved and will integrate the data warehouse. The basic approach for the multidimensional model design is bottom-up which is common and suitable for data exploration [8].

User requirements are considered and aggregations, calculations and other transformations are accounted for in the design model and will be implemented in the second phase of the ETL Process.

The third phase is when we populate our fact and dimension tables with the appropriate records; it's the moment to take advantage of all the technical features described in the previous section.

As to the second phase, where data is transformed in order to be integrated into the DW, it is the most delicate part of the process and responsible for the overall quality of the data. Typical issues are: inconsistent data, incompatible data structures, different data granularity and incomplete classifications.

Most of the problems we faced occurred when integrating information. Every new project we include in the Portuguese NSI data warehouse is integrated with all other projects already there. Why do we make this integration effort? Because only if we use shared dimensions can we compare data from distinct surveys.

For example on one hand every year an economic complex survey is applied to several companies. From this survey, data such as the number of employees, the total business volume or the capital of the company becomes available. On the other hand, the tourism survey has information, for each tourist resort, of the number of available beds, the number of people that slept in the hotel, that ate in the restaurant or that used other facilities. But a tourist resort is also a company so if we're able to integrate information from both surveys we can compare it not only within the same survey but also across surveys.

This is added value; it enables the production of derived products and may reduce the burden upon the respondents, in some cases. It also provides a new and more intertwined vision and understanding of Portuguese society better reflecting reality.

However integrating information, especially old information, is difficult and raises some problems. We will explore these issues separately due to their importance for the overall data quality in the following sections.

3.2.1 Out of Range Values

The most common task performed during the transformation phase is data cleansing. What is this? As stated before, measures are analyzed with regard to data in the dimensions that surround the fact table. So we have to make sure that every dimensional attribute in the fact table is also available in the dimension table, i.e., the range of possible values in the fact is restricted by the dimension values. The detection of this problem may indicate a problem with the application developed to collect the survey data, and this information can and should be used to improve the collection processes quality.

The corrective steps may be the filtering of the incorrect records or the translation to other codes. These options must be presented to the statisticians responsible for data production and who should choose which solution to adopt. Data ownership is always respected and ensured, and the data warehouse analyst role is only to help and assist the statisticians along the process.

3.2.2 Values Absence

Another problem may be the absence of values in the fact table for a dimensional field. Sometimes, a question is not answered or, due to the questionnaires specificity it does not apply to every respondent. In these cases it is common to leave it unanswered. However as all the facts have to be classified for every dimension we have to alter the dimension to include the appropriate option and reclassify the null fields in the fact table.

Once again the option to alter the dimension table should be left to the statisticians consideration.

3.2.3 Incomplete and Ragged Dimensions

Data aggregations or de-aggregations performed by the user, during the analysis, are made along the classification explicitly defined in the dimension.

During the transformation/integration of the data we have to make sure that a hierarchy exists and is complete. A dimension determines the granularity adopted for representing facts [9]. A hierarchy determines how fact instances may be significantly aggregated and selected during the analysis process.

A dimension is structured as a tree with different levels. The top level, also known as “All Level”, includes every element of the dimension. At the lower level, the leaves of the tree level, each member, or each leaf is directly attached to the fact.

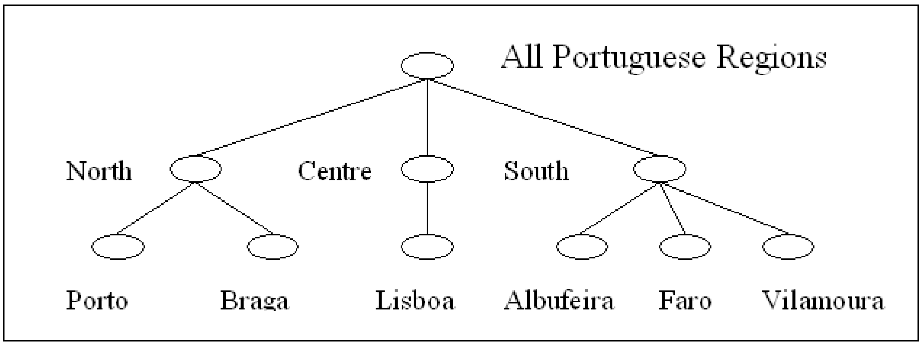

Figure 2 - Example of Geographical Dimension with 3 Levels

The intermediate levels or branches must connect every leaf with the All Level. The resulting acyclic directed graph defines the hierarchy.

If a leaf doesn't have a father, i.e. isn't connected with a node at a superior level, which defines a coarser granularity to the facts, an intermediate member has to be created, or the leaf may be linked directly to the top of the tree – All Level.

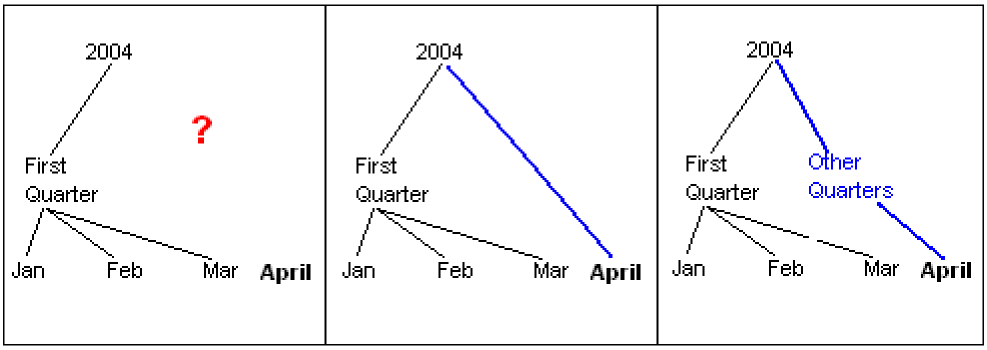

Figure 3 - Solutions for Unconnected Leaves

Connecting the leaf directly to the top of the tree creates ragged or unbalanced dimensions, which is discouraged. The data producers should be helped and encouraged by data warehouse analysts to find a residual member for the intermediate level, which can encompass all the leaves unconnected and relate them with the top level.

3.2.4 Unbalanced Classification

When integrating different data sources, like administrative sources or older information sometimes we find that not every fact is classified at the lower level, i.e. the code of the fact is present in the dimension table, but not in one of its leaves.

What this means is that the fact classification is unbalanced, some facts have lower granularity than others.

Company	Region	Sales_Value
One	Lisboa	2.548 €
Two	Faro	965 €
Two	Porto	726 €
Three	Albufeira	1.369 €
Three	**North**	3.142 €

Figure 4 - Example of Unbalanced Facts Classification

Most measures are additive, which means that when we aggregate the facts through the dimension, the value we obtain to the dimension member in the upper level is equal to the sum of all children. For example, relating to the geographical dimension illustrated in figure 2, where the North Region of Portugal only has 2 children, Porto and Braga, the sales value for a company in the North Region would be the sum of all sales in Porto and in Braga. However if we only have the sales value for the North Region we don't know how to split it between both cities.

Some OLAP tools allow us to state that the aggregation should not be computed. In this case all the values for all dimension members in intermediate levels have to be prepared in the transformation phase.

We already made some tests with this kind of solution and we cannot recommend it. Building the structure in this way takes much longer (2/3 months longer) but the major problem with this approach is that the user does not understand why he can perform some de-aggregations and not others. This kind of situation causes the user some distress and frustration, and should be avoided.

Unfortunately the other solutions usually available in the literature are to split the value evenly among all member children (with the remark that it's an estimated value), create a parallel classification or hide all the levels where de-aggregation is not possible.

In classifications where this is a persistent problem we've created a residual member for each sub-tree. If the value of the members of the sub-tree is available then the residual will be 0, or if none of the members has values the residual will be equal to the parent member. Otherwise it will be the difference between the parent member and the sum of its children.

This is also the solution we implemented to a similar problem, the "Cross-Level Additive" problem: the sum of all children of a particular dimension member is not equal to their parent value. In this case the residual member is created with the difference value. This way all de-aggregations and other navigational data explorations remain available.

3.2.5 Multiple Hierarchies

Sometimes a dimension has more than one hierarchy. The most common example in the literature is the civil versus the fiscal year, which ends in March. This situation per se is not a problem once the facts are classified at monthly level (or lower).

The problem we have at the Portuguese NSI is a little different. In the geographic dimension there are three different levels of the territorial units nomenclature (NUTS), which de-aggregates to municipalities. The dimension could be built as the Time dimension from the academic examples [10], but the codes of the municipalities and the codes of the territorial units nomenclature are not as stable as the months of the year. In the last years not only have the codes changed several times, but also the path of aggregation has been altered.

For instance in 2002 Municipality 1 could be aggregated to NUTS A and Municipality 2 to NUTS B, and now both municipalities have been altered to aggregate to NUTS A.

Changes in the dimension reflect the new situation, at the cost of loosing the old aggregation path. If we build a new dimension to reflect the changes we'll have as many different geographies as changes, making the maintenance more difficult.

So to solve this problem we have the facts classified at the municipality level and also at the lower level of the territorial units nomenclature. Two sub-dimensions thus form the dimension and the aggregation hierarchy is explicitly in the facts.

Having the facts classified at both levels does not increase the effort made upon the database because the classification is made during the transformation/integration and has great benefits.

Particularly in demographical data, which intensively uses geographical classifications, the Portuguese NSI has integrated in the data warehouse data from the past 24 years. Last time the geographical dimension changed, all the DW projects encompassed in the demography theme like births, deaths, marriages and divorces were changed accordingly to accommodate the new classification in just seven days.

3.3 Data Warehouse Evolution

As we've seen building a data warehouse involves an everlasting *design phase,* where the designer has to produce various modeling constructs, accompanied by a detailed physical design for efficiency reasons (involving indexing, clustering, a continuous analysis of the most common queries, etc.) To top all these, the designer must deal with the data warehouse processes too, which are complex in structure, large in number and hard to code at the same time. Dealing with the data warehouse as set of layered, materialized views is, thus, a very simplistic view. As it has been indicated, the data warehouse refreshment process can already consist of many different sub-processes like *data cleaning, archiving, transformations, and aggregations.*

To make the picture complete, we must add the *evolution* phase, which is a combination of design and administration: as time passes, new data is requested by the end users, classifications change, new sources of information become available, and the data warehouse architecture must evolve to meet these challenges

In a DW you can, respectively, make the following graceful changes to the design after the data warehouse is up and running by:

1. Adding new unanticipated facts (that is, new additive numeric fields in the fact table), as long as they are consistent with the fundamental granularity of the existing fact table.
2. Adding completely new dimensions, as long as there is a single value of that dimension defined for each existing fact record.

3. Adding new, unanticipated dimensional attributes.
4. Breaking existing dimension records down to a lower level of granularity from a certain point in time forward.

These points have already been illustrated in the previous section, their point being that the Data Warehouse is not a static structure, it's rather like an alive and expanding creature whose performance is dependent upon a good multidimensional design to begin with and a good integration process design, not only when populating the table for the first time but during all its life time.

4. Conclusions

We briefly presented the database evolution towards querying and analysis, in the retrieving perspective as opposed to the storing perspective.

Despite all the attention that data warehouses nowadays receive its development was forced by the needs of the industry rather than by academic studies, products to implement multidimensionality and other solutions to improve the analysis performance where developed by the industry without much formal backup from the academia.

Therefore we feel that it is important to share our work in the definition of a multidimensional database design and accompanying ETL process. Our major problems in the last two years are depicted along with the solutions we implemented, real examples are provided to clarify the methodology. Database design features as partitioning tables, bitmap and bitmap join indexes have been described and heuristics to determine which attributes should be treated as measures or dimensions have been presented. Various techniques, adopted at the Portuguese National Statistic Institute, to solve problems during Data Warehouse design have been discussed and its importance in the future data exploration has been illustrated.

DW technology is still in its childhood if not in its infancy, and it's natural that solutions implemented today can be improved tomorrow. Our purpose is to be aware of the technological possibilities and to develop computational solutions to help and assist the statisticians along the process of improving the overall quality of the institute products'.

5. References

[1] Kimball R.: *The Data Warehouse Toolkit*. John Wiley & Sons. 1996.

[2] OLAP Council. OLAP AND OLAP Server Definitions. 1997.

[3] *A Survey of Logical Models for OLAP Databases*. SIGMOD Record. 1999.

[4] Vassiliadis P.: *Data Warehouse Modeling and Quality Issues*. Ph.D. Thesis. 2000.

[5] Agrawal R., Gupta A. and Sarawagi S.: *Modelling Multidimensional Databases*. In Proc. 13th Int. Conf. Data Engineering (ICDE). 1997.

[6] Lomet D. and Salzberg B.: *The Hb-Tree: a multidimensional indexing method with good guaranteed performance*. ACM Trans. On Database Systems, vol. 15, n. 44, pp.625-658, 1990.

[7] Inmon W. H.: *Building the Data Warehouse*. Wiley and Sons. 1996.

[8] List B., Bruckner R. M., Machaczek K. and Schiefer J.: A Comparison of Data Warehouse Development Methodologies - Case Study of the Process Warehouse. 2002.

[9] Golfarelli M., Maio D. and Rizzi S.: *Conceptual Design of Data Warehouses from E/R Schemes*. In Proc. Hawaii Int. Conf. on System Sciences, vol. VII, Kona, Hawaii. 1998.

[10] Lechtenborger J. and Vossen G.: *Multidimensional Normal Forms for Data Warehouse Design*. Elsevier Science. 2002.