Proceedings of the *International Conference on Theory and Application of Mathematics and Informatics ICTAMI 2005* - Alba Iulia, Romania

DATABASES - DATA WAREHOUSE

Ileana ştefan

Abstract. Data warehouse is a subject-oriented, integrated, variable in time and non-volatile data collection, with the purpose of supporting decision processes in administrative areas. A data store is a technology designed for data administration and analysis.

1. Introduction

IBM specialists, with the purpose of accessing data stored in non-relational systems, first used the concept of data warehouse. The father of "data warehousing" is considered to be Bill Inmon.

2. Motivation Scenario

Data warehousing is a subject-oriented data collection, integrated and variable in time, non-volatile, used by decision-making processes within administrative areas and, it actually represents a technology for data administration and analysis. The potential advantages of data warehousing are the high probability for covering investments, the increased competition advantage and the high productivity of the general decision authorities.
The architecture of data warehouse includes various elements, namely:

- Operational data stored in data bases;
- A data warehouse manager who performs operations on data analysis with the purpose of assuring the coherence, the change and the mixing of source data, as well as for producing de-normalization and groups, safe data's filing and saving;

- The loading administrator performs operations for loading and extracting data from data warehouse;
- Inquiry administrator performs operations for managing users' inquiries;
- Access instruments for final users: reporting and inquiring instruments, instruments for developing applications, for the executive information system, for on-line analytical processing and for data extracting.

Data warehousing manages five main information flows:

- The internal flow that includes data extracting, deleting and loading from the source to data warehouse;
- The ascending flow includes the process of adding value to data from the warehouse by means of summarizing, packing and distributing the data;
- The descending flow includes filing and safe saving of data from data warehouse;
- The external flow includes the process of presenting data to the disposal of final users;
- The meta-flow includes the processes of administrating meta-data.

3. Designing data warehouses

When designing a database for a data warehouse, this should be designed so as to answer ad-hoc inquiries within the constraining limits of acceptable performances. In a data warehouse there will be performed a large number of inquiries about circumstances, analyzed in different ways. For example, when studying the evolution of a real estate agency, the following inquiries can be made:

- What is the average number of new clients recorded in the last month by each subsidiary compared to the same month of the last two years;
- What the prospects are for the number of clients from each main city of Romania who will look for estates to rent in the following year. This number will be calculated on the basis of the increasing rate of the last five years;

- What is the average amount of rented estates in each subsidiary, with a monthly rent of more than 300 Euros, for the last six months;
- What is the total number of estates visited by the tenants, according to the estate type, for each month of 2004.

The real estate agency has the following structure:

Subsidiary: $(No\_Subsidiary, City, Area, Address, Tel\_no, Fax\_no)$
Estate: $(No\_Estate, Address, Area, City, Type, Rooms, Rent, No\_Owner, No\_Persons, No\_Subsidiary)$
Tenant: $(No\_Tenant, First\_Name, Surname, Address, Tel\_no., Type, Maximum\_Rent, No\_Subsidiary)$
Owner: $(No\_Owner, First\_Name, Surname, Address, Tel\_no)$
Visit: $(No\_Tenants, No\_Estate, Date, Commentaries)$

All these inquiries will use data resulting from business transactions. Such examples of circumstantial data associated to the evolution study of the real estate agency are the followings: estates visiting, their inspections and the renting agreements.

4. Designing star-shaped schemes

A star-shaped scheme is a logical structure which includes a central circumstantial table (containing circumstantial data), surrounded by dimensional tables (containing reference data). The circumstantial table contains a foreign key for each dimensional table. This structure uses the characteristics of circumstantial data so as the circumstances are generated by past events and it is improbable to be changed, no matter the way they are analyzed. As the massive number of data from the warehouse is represented within circumstances, the circumstantial tables can get very large, compared to the dimensional tables. For this reason it is very important for the circumstantial data to be treated as reference data only for reading and which will not be changed in time. For making the difference between circumstances and dimensional data, it is necessary to identify the main transactions within each application in the business area. For each circumstantial table it is necessary to identify the key dimensions applicable to each circumstance. The star-shaped scheme will be analyzed for the data referring to the visit to the estates within the research on the real estate evolution as it is presented in the bellow figure.

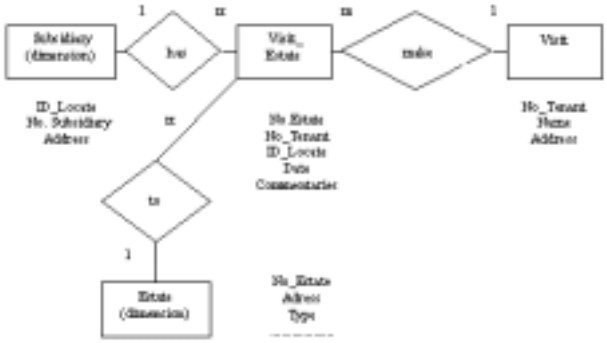

Figure 1: The star-shaped scheme

Each visit to an estate is represented in the "$Visit\_Estate$" circumstantial table, which is found within the star-shaped scheme. Around "$Visit\_Estate$" table there are the candidate dimensional data included in the "$Subsidiary$", "$Estate$" and "$Tenant$" dimensional tables. The date for the visit is to be found in the circumstantial table owing to the special characteristics of this dimension type.

5. Designing circumstantial tables

For the design of circumstantial tables we have to consider the relation between the value of stored data and the cost of their storing. The dimension of these tables can be very large and for the design to be good we have to consider the following factors:

- We have to identify the necessary time for each application of supporting decisions. For example to get an answer for the possible estates that are rented from the following month, we may not need the analysis of the rents form the last months, but the examination of the same month in the last two years;
- Identifying the requirements concerning the statistics samples applied to certain sub-collection data, compared to the requirements concerning more detailed. For example to get the answer to the above question we could compare the average of the agreements referring to the estates for rent in each city, instead of the detailed agreements within each subsidiary;

- Identifying the columns that will be deleted. For instance the columns that can be deleted are those referring to fields containing information about the condition of the estate, descriptions with long texts, group values, intermediary values;

- Reducing dimensions form the circumstantial tables which leads to big reductions of the corresponding table's dimensions;

- Determining an optimal use of foreign intelligent and non-intelligent keys. A foreign intelligent key contains the unique identifier for a certain entity, for example the area of "No_Estate", "No_Owner". A foreign non-intelligent key contains information referring to the unique identifier, for instance "ID_Locate". By the use of foreign intelligent keys it is avoided to unify the parent tables. For instance on interrogations about the visits to a certain estate, one can use the information stored in "$Visit\_Estate$" table, while the interrogations about the visits within a certain number of subsidiaries will require the unification of "$Visit\_Estate$" and "$Subsidiary$" tables. However the use of intelligent keys has the disadvantage that if the values, contained within columns of the foreign key, change, such changes will result into the necessity of costing upgrading of the circumstantial table;

- Determining the optimum way of handling the issue when introducing the time in the circumstantial tables. The real format of the stored date/time dimension depends on the users' requests for interrogation;

- Structuring circumstantial tables with the purpose of contributing to the maintenance capacity. As the dimensions of circumstantial tables can become very large, problems can occur regarding the administration, saving and updating operations for data warehouse. The interrogations usually refer only to parts of the circumstantial table at a certain moment and therefore it is goods to structure tables into little fragments, based on time periods. The fragmentation strategy should correspond to the requirements imposed to the applications for decision support.

6. Designing dimensional tables

The restructuring of dimension tables is not costing with the restriction that the basic keys of the circumstantial tables are not to be changed The use

of star-shaped schemes can lead to the increase of interrogations' performances by using referential information's de-normalizing within one dimensional table. For example, the information referring to the visits of the estates can include questions about:

- The types of the visited estates;
- The subsidiary where the estates are registered;
- The estates held by a certain owner.

6. Conclusion

Such interrogations can be performed efficiently if all information which imposes constraints is in the same table. This can be obtained by de-normalizing all supplementary data related to an estate, within one dimensional table. Star-shaped estates

$$\begin{array}{rl} \textbf{Estate} & (No\_Estate, Address\_Estate, Type, Rooms, Rent, \\ \textbf{(de-normalizing):} & No\_Subsidiary, Address\_Subsidiary, \\ & No\_Owner, Name\_Owner, Type\_Owner) \end{array}$$

References

[1]Anahory S., Murray, *Data Warehousing in the Real World: A Practical Guide for Building Decision Support System*, Harlow: Addison Wesley , 1997.

[2]Ceri S., Pellegatti G., *Distributed Database - Principles and System*, McGraw-Hill, 1984.

[3]Connoly T. , Begg C., Strachan A., *Database System - A Practical Approach to Design*, Implementation and Management, Addison Wesley, 1998.

[4]Devlin B., *Data Warehousing: Architecture to Implementation.* Harlow: Addison Wesley, 1998.

[5]Lindsay B. G., *Notes on Distributed Databases*, IBM Research Report RJ2571.

[6]Oszu T., Valduriez P., *Principles of Distributed System*, second Edition, Prentice Hall, 1999.

Ileana Ştefan
Department of Computer Science
"Petru Maior" University of Tg-Muresş, MUREŞ county , 540088, ROMÂNIA
email:*ileana.stefan@ea.upm.ro*