Data warehouse architecture on the base of dimensional modelling

Zlatinka Covacheva

Abstract: *The present talk deals with the data warehouse architecture. Dimensional modelling, "star" and "snowflake" schemes are defined. The advantages of the dimensional model are described. A comparison between dimensional and entity relation modelling is given. An application of dimensional modelling in the BTC PLC data warehouse architecture is presented.*

Key words: *data warehouse architecture, dimensional modelling, entity relation modelling, star scheme, snowflake scheme*

INTRODUCTION

In today's environment, the most valuable and vulnerable asset to any company is its data. Organizations are gathering huge amounts of data. A data warehouse provides an enterprise-wide view of this information to manage the organization as a whole. Business intelligence is the crucial piece that transforms information into knowledge and wisdom [1].

Early builders of data warehouses already consider their systems to be key components of their information technologies strategy and architecture. Numerous examples can be cited of highly successful data warehouses developed and deployed for businesses of all sizes and all types.

Throughout the history of systems development, the primary emphasis had been given to the operational systems and the data they process. The fundamental requirements of the operational and analysis systems are different: the operational systems need performance, whereas the analysis systems need flexibility and broad scope [3].

Despite all the changes in the platforms, architectures, tools, and technologies, a remarkably large number of business applications continue to run in the mainframe environment of the 1970's. By some estimates, more than 70 percent of business data for large corporations still resides in the mainframe environment. These systems, generically called legacy systems, continue to be the largest source of data for analysis systems [3].

The applications using data warehouse require high query performance. This requirement is in conflict with the need to maintain in the data warehouse updated information. The DW configuration problem is the problem of selecting a set of views to materialize in the DW that answers all the queries of interest while minimizing the total query evaluation and view maintenance cost.

Today's data warehousing systems provide the analytical tools afforded by their precursors. But their design is no longer derived from the specific requirements of analysts or executives and, data warehousing systems are most successful when their design aligns with the overall business structure rather than specific requirements [4].

Data warehousing systems are most successful when data can be combined from more than one operational system. When the data needs to be brought together from more than one source application, it is natural that this integration be done at a place and in the form independent of the source applications.

The data warehouse may very effectively combine data from multiple source applications such as sales, marketing, finance, and production. Many large data warehouse architectures allow for the source applications to be integrated into the data warehouse incrementally.The primary reason for combining data from multiple source applications is the ability to cross-reference data from these applications. Nearly all data in a typical data warehouse is built around the time dimension. Time is the primary filtering criterion for a very large percentage of all activity against the data warehouse. An analyst

may generate queries for a given week, month, quarter, or a year. Another popular query in many data warehousing applications is the review of year-on-year activity.

The design and architecture of a data warehouse must be flexible enough to grow and change with the business needs. A data warehouse contains a copy of transaction data specifically structured for querying and analysis.

A data warehouse can be normalized or denormalized. It can be a relational database, multidimensional database, flat file, hierarchical database, object database, etc. Data warehouse data often gets changed. And data warehouses often focus on a specific activity or entity [5].

The data warehouse model outlines the logical and physical structure of the data warehouse. This model is likely to be less normalized than an operational system model [2].

DIMENSIONAL MODELLING

Dimensional modelling (DM) is a logical design technique that seeks to present the data in a standard, intuitive framework that allows for high-performance access [6]. It is inherently dimensional, and it adheres to an application that uses the relational model with some important restrictions. Every dimensional model is composed of one table with a multipart key, called the fact table, and a set of smaller tables called dimension (or lookup) tables. Each dimension table has a single-part primary key that corresponds exactly to one of the components of the multipart key in the fact table. This characteristic "star-like" structure is often called a star join. The term star join dates back to the earliest days of relational databases. A star schema can be simple or complex. A simple star consists of one fact table; a complex star can have more than one fact table.

When one dimension is presented using a hierarchy of dimension tables, a "snowflake like" structure is obtained. The snowflake schema is an extension of the star schema where each point of the star explodes into more points. The main advantage of the snowflake schema is the improvement in query performance due to minimized disk storage requirements and joining smaller lookup tables. The main disadvantage of the snowflake schema is the additional maintenance efforts needed due to the increase number of lookup tables.

A fact table, because it has a multipart primary key made up of two or more foreign keys, always expresses a many-to-many relationship. The most useful fact tables also contain one or more numerical measures, or "facts," that occur for the combination of keys that define each record. The most useful facts in a fact table are numeric and additive. Additivity is crucial because data warehouse applications almost never retrieve a single fact table record; rather, they fetch back hundreds, thousands, or even millions of these records at a time, and the only useful thing to do with so many records is to add them up.

Dimension tables, by contrast, most often contain descriptive textual information and usually represent real physical objects, processes and phenomena. They have statistical character and provide a way to organize the data.

DIMENSIONAL MODELLING vs ER MODELLING

DM is different from, and contrasts with, entity-relation modelling (ER modelling). The key to understanding the relationship between DM and ERM is that a single ERM diagram breaks down into multiple DM diagrams [6].

ER modelling is a logical design technique that seeks to remove the redundancy in data. At least, it is necessary to separate out the redundant data into distinct tables.

The ER modelling technique is used to illuminate the microscopic relationships among data elements. The highest art form of ER modelling is to remove all redundancy in the data. This is immensely beneficial to transaction processing because transactions are

made very simple and deterministic. The success of transaction processing in relational databases is mostly due to the ER modelling.

.The ER model for the enterprise has hundreds of logical entities! High-end systems such as SAP have thousands of entities. Each of these entities usually turns into a physical table when the database is implemented.

But the greatest shortages of the ER model for the data warehouse building purposes are the following [6]:

- End users cannot understand or remember an ER model. End users cannot navigate an ER model. There is no graphical user interface (GUI) that takes a general ER model and makes it usable by end users.
- Software cannot usefully query a general ER model. Cost-based optimizers that attempt to do this are notorious for making the wrong choices, with disastrous consequences for performance.
- ER modelling does not really model a business; rather, it models the micro relationships among data elements. ER modelling does not have "business rules," it has "data rules." Few if any global design requirements in the ER modelling methodology speak to the completeness of the overall design.
- ER models are wildly variable in structure.

The wild variability of the structure of ER models means that each data warehouse needs custom, handwritten and tuned SQL. It also means that each schema, once it is tuned, is very vulnerable to changes in the user's querying habits, because such schemes are asymmetrical. By contrast, in a dimensional model all dimensions serve as equal entry points to the fact table. Changes in users' querying habits don't change the structure of the SQL or the standard ways of measuring and controlling performance.

ER models do have their place in the process of data warehouse building. First, the ER model should be used in all legacy OLTP applications based on relational technology. This is the best way to achieve the highest transaction performance and the highest ongoing data integrity. Second, the ER model can be used very successfully in the back-room data cleaning and combining steps of the data warehouse. This is the ODS (operational data store).

However, before data is packaged into its final queryable format, it must be loaded into a dimensional model. The dimensional model is the only viable technique for achieving both user understandability and high query performance in the face of ever-changing user questions [6].

The dimensional model has a number of important data warehouse advantages that the ER model lacks:

- The dimensional model is a predictable, standard framework. Report writers, query tools, and user interfaces can all make strong assumptions about the dimensional model to make the user interfaces more understandable and to make processing more efficient. For instance, because nearly all of the constraints set up by the end user come from the dimension tables, an end-user tool can provide high-performance "browsing" across the attributes within a dimension via the use of bit vector indexes. Metadata can use the known values in a dimension to guide the user-interface behavior. The predictable framework offers immense advantages in processing. Rather than using a cost-based optimizer, a database engine can make very strong assumptions about first constraining the dimension tables and then "attacking" the fact table all at once with the Cartesian product of those dimension table keys satisfying the user's constraints. Using this approach it is possible to evaluate arbitrary n-way joins to a fact table in a single pass through the fact table's index.

- The predictable framework of the star join schema withstands unexpected changes in user behavior. Every dimension is equivalent. All dimensions can be thought of as symmetrically equal entry points into the fact table. The logical design can be done independent of expected query patterns. The user interfaces are symmetrical, the query strategies are symmetrical, and the SQL generated against the dimensional model is symmetrical.
- It is extensible to accommodate unexpected new data elements and new design decisions. All existing tables (both fact and dimension) can be changed in place by simply adding new data rows in the table, or the table can be changed in place with a SQL alter table command. Data should not have to be reloaded. Extensibility also means that no query tool or reporting tool needs to be reprogrammed to accommodate the change. And finally, extensibility means that all old applications continue to run without yielding different results.
- There is a body of standard approaches for handling common modelling situations in the business world. Each of these situations has a well-understood set of alternatives that can be specifically programmed in report writers, query tools, and other user interfaces. These modelling situations include:

  -Slowly changing dimensions, where a "constant" dimension such as Product or Customer actually evolves slowly and asynchronously. Dimensional modelling provides specific techniques for handling slowly changing dimensions, depending on the business environment.

  -Heterogeneous products, where a business needs to track a number of different lines of business together within a single common set of attributes and facts, but at the same time it needs to describe and measure the individual lines of business in highly idiosyncratic ways using incompatible measures.

  -Pay-in-advance databases, where the transactions of a business are not little pieces of revenue, but the business needs to look at the individual transactions as well as report on revenue on a regular basis.

  -Event-handling databases, where the fact table usually turns out to be "factless."
- There are a lot of administrative utilities and software processes that manage and use aggregates. Recall that aggregates are summary records that are logically redundant with base data already in the data warehouse, but they are used to enhance query performance. A comprehensive aggregate strategy is required in every medium- and large-sized data warehouse implementation. All of the aggregate management software packages and aggregate navigation utilities depend on a very specific single structure of fact and dimension tables that is absolutely dependent on the dimensional model.

APPLICATION OF DIMENSIONAL MODELLING IN BTC PLC DATA WHAREHOUSE ARCHITECTURE

The Bulgarian Telecommunications Company's (BTC) data warehouse integrates the information from a lot of operational systems. These systems are of On-Line Transaction Processing type and contain current data about the customers, services, telecommunications network, personnel etc. The main part of this information is aggregated and entered every month into the data warehouse structure.

The BTC PLC data warehouse architecture is based on the dimensional modelling. This model is chosen on the basis of its advantages described above.

The program environment for design and development of data warehouse of BTC PLC includes ORACLE products – ORACLE Designer /2000, ORACLE Developer /2000, ORACLE Discoverer /2000 and ORACLE Express Server 6.2.

The information in the data warehouse is very extensive in volume and varied in content. The model chosen allows for a fast and easy query design for the end users from the Headquarters of the Company.

There exists a set of numerous fact tables, containing the values of the main variables presenting the development of the Company. These data have quantity character and are used for statistical analysis and forecasting purposes. They contain the number of the customers, the number of the exchanges and telephone lines, the number of calls, the revenue, the investments, the number of employees, the salaries, etc.

All fact tables have several dimension tables, presenting the main attributes by which the variables are described. The main dimensions are the regions of the Company (according to the geographical structure of the country) and the time periods (months, quarters, years). Other dimensions present the types of the services (one party telephone lines, two party telephone lines, ISDN, leased lines, etc.), the customer types (residential or business), the exchange types (analogue or digital), the network types (up to 10000 lines or over 10000 lines), etc. Most of the dimensions have hierarchical structures.

For instance, one of the most valuable variables is the traffic of the telephone services. It is presented by a star scheme, which consists of a fact table and a lot of dimension (lookup) tables, describing the structure of the information. This scheme is presented on Fig.1.

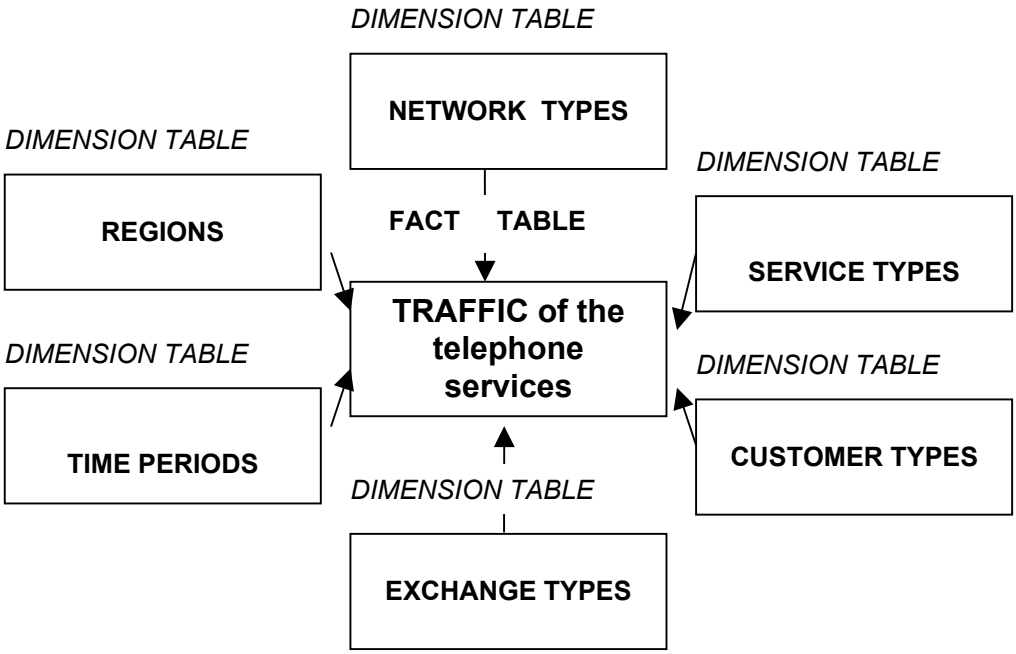

Fig.1. Star scheme of the telephone services traffic

Analogously, a lot of other schemes present the revenue and quality of the services, the projected, installed and operated capacity of the exchanges, the investments, the personnel, etc.

CONCLUSIONS

A data warehouse is a structured extensible environment designed for the analysis of non-volatile data, logically and physically transformed from multiple source applications to align with business structure, updated and maintained for a long time period, expressed in simple business terms, and summarized for quick analysis.

Data warehousing systems have become a key component of information technology architecture. A flexible enterprise data warehouse strategy can yield significant benefits for a long period.

Dimensional modelling provides efficient data processing and understandable user interface. It presents a very flexible and easily changeable data structure. Dimensional modelling is very useful for data analysis applications in the dynamically variable environment of the modern business world.

REFERENCES

[1] Gill, H., P. Rao, The Official Computing Guide to Data Warehousing, Que Corporation, USA,1996.

[2] Greenfield L., Different Aspects of Data Warehouse Architecture, http://www.dwinfocenter.org/architect.html

[3] Gupta V., An Introduction to Data Warehousing, System Services Corporation, Chicago, Illinois, http://www.system-services.com

[4] Inmon B., Data Mart Does Not Equal Data Warehouse, DM Review., May 1998

[5] Kimball R., A Definition of Data Warehousing, http://www.dwinfocenter.org/defined.html

[6] Kimball R., The Data Warehouse Toolkit: How to Design Dimensional Data Warehouses, John Wiley, 1996

ABOUT THE AUTHOR

Assoc.Prof. Zlatinka Covacheva, PhD, Higher College of Telecommunications and Posts, Phone: (+359 2) 62 30 21/123 or (+359 2) 955 85 43, E-mail: zkovacheva@hctp.acad.bg or zkovacheva@hotmail.com.