A Data Warehouse System for Monitoring University Education Process

Milan Čeliković
University of Novi Sad, Faculty of Technical Sciences, Novi Sad, Serbia
milancelik@sbb.co.yu

Slavica Aleksić
University of Novi Sad, Faculty of Technical Sciences, Novi Sad, Serbia
slavica@uns.ac.rs

Ivan Luković
University of Novi Sad, Faculty of Technical Sciences, Novi Sad, Serbia
ivan@uns.ac.rs

ABSTRACT

We present in the paper a pilot project of a data warehouse system for monitoring the accomplishment of course requirements in the lecturing process at the Faculty of Technical Sciences of the University of Novi Sad. Currently, the teaching staff keeps their own records of those data and therefore, a great amount of them is stored separately, in different systems. Accordingly, a need arises to unify the data in an integrated database of the faculty information system. Since the faculty consists of 13 departments, each comprising a number of chairs, it will become an emergent requirement. By this, the faculty management staff may obtain a basis to perform various data analysis, generate reports, and make estimations. We present a part of the project that has been already implemented. It covers the accomplishment of all course requirements during a whole semester of the lecturing process. The database schema of our data warehouse system is presented here in detail.

Key Words: Data Warehouse, Education Process

1. Introduction

In the education process, there is a law obligation for faculties to keep records about final students' grades and ECTS points, for all exams. At the Faculty of Technical Sciences of the University of Novi Sad, these data are stored in an OLTP database (db) of the faculty information system (IS). However, apart from storing values of final students' grades, there is a need to keep records about all accomplished requirements, during the whole semester in the lecturing process, i.e. in-term requirements. These values are used not only to calculate final students' grades, but also to perform various data analyses by the faculty management staff. Therefore, such data are not only a measure of success of a sole student, but may be very useful in performing various business analyses at the faculty level. For example, it is possible to generate information about trends by notifying changes of various parameters by various student generations, courses, curricula, and lecturers. In our circumstances, analyses of regularity of lesson attendance and duty accomplishment during the semester are particularly important.

In the paper, we present a db schema of a data warehouse (DW) pilot project implemented at the Faculty of Technical Sciences in Novi Sad (FTS). Currently FTS comprises 13 departments and enrolls over 9500 students, with a continuous trend of increasing this number. Accordingly, there is a continuous grow of the number of records about accomplished requirements. We estimate that a table for storing this data may reach a couple of million of tuples, only for active students, and even tens of millions, if archived data for past students are also included. Regardless the evident and continuous progress in IT development, computing the estimations and performing analyses form such an amount of data is still cumbersome. Sometimes, response times for such computations and queries may overcome hours, which is rarely acceptable for the end-users. They usually fill such software systems as completely inefficient and uncomfortable.

Information System of FTS (FTS IS) provides an OLTP database that stores only final students' grades of all exams. Data about in-term requirements (such as mid-term tests, lab projects, assignments, etc.) are still stored in a nonsystematic way. Each lecturer keeps his or her own records. There is no official software application of FTS IS that provides lecturers in that, and even more, there is no standard of structuring these data.

We propose development of an integrated IS for monitoring all accomplished in-term requirements during semesters, at the operational level. By this, such an IS should provide an OLTP database for storing not only final students' grades and ECTS points, but also all the operational data in the lecturing process.

Besides, we also propose that such an IS should comprise a software system that utilizes the OLTP database as its data source to monitor the accomplishment of course requirements during semesters. Such software should provide a considerable shortened response times in searching data and performing data analyses. It also should provide importing data from various data sources. Those data sources may be different as it concerns both the applied technology and the data structuring rules. For the purpose of various data analyses, it is necessary to provide archive data, spanning historical period of a number of years. Therefore, we utilize in this project a data warehouse approach [1] to satisfy all the aforementioned requirements and the faculty management staff needs. The system should be designed in such a way to support various organizational changes at the level of faculty or university.

2. Related Work

There are a lot of references considering design and implementation of DW systems, for example [2] and [3], that we have consulted in our project. In this section, we present selected characteristics of some DW systems designed for various universities or faculties.

The Polytechnic University of Brooklyn used a central student IS [4]. Departments also used their own databases, files and other external data sources. The goal of a new DW system was to unify all the data from various data sources and put them into a common IS. A commercial enterprise system *Peoplesoft* was used for that purposes. DW was seen as a suitable approach due to a possibility to integrate data from various systems. A similar situation is at FTS, where there are a number of smaller and independent data sources, apart from the existence of a central FTS IS.

The development of the system was performed trough a number of steps. The first one was identifying all the data sources at the university and then an analysis of their validity. The next step was an integration of collected data into a unified db, with checking the data consistency. The last step was the development of an assessment process which feed data directly into the DW, based on a web technology. After completion of a test, the system generates student's grade and store it into the db. At the very beginning, MS Access was used as a database management system (DBMS). Later, it was replaced by MySQL, due to low performance problems.

At the beginning, the management staff, i.e. the end-users of the system did not recognize its advantages. However, after a certain time, they started with its usage by performing a number of analyses, estimations and reports, and became aware of its usefulness. Great investments for implementation of the system were related not only to software costs, but also to the cost of hardware components of the system architecture. More information about the project may be found in [4].

In [5], the authors considered a usage of DW approach coupled with a decision support system in a university environment. They presented a DW system that was developed and implemented at their university. It comprises five subsystems that store:

- for each student: personnel data, assigned curricula and semester info;
- data about lecturing process performance, at the level of a sole student;
- history data about curricula modifications;
- data about students' points and grades, for all duties during semesters; and
- data about all students' prices issued by the university authorities.

The system was implemented under Oracle DBMS and reporting tools. A considerable attention was paid to the system security and safety. Access rights to the system services have been given only to the management staff, and to the people granted by the management staff. Not all the end-users have the same access rights. Management and student service staff members usually have all the system rights, while there are users not having rights to access students' personnel data.

Software Brio is used as a reporting tool. It supports the SQL syntax. Using Brio, the end-users can create their own reports. Besides, there are pre-created reports that provide an efficient response to frequently performed analyses and estimations. The database is refreshed two or three times per semester. There are special procedures applied for data validation to preserve data consistency. The end-users are granted the access to the system meta data. Accordingly, they may create their own queries and reports without any assistance of a system administrator. Therefore, each user is supposed to pass a proper training before the system usage. Besides, there are various forms of the user support in the system operation. For example, once a month there are meetings with a purpose to inform end-users about new functionalities and system changes.

3. User Requirements

A goal of our DW system is to provide complex analyses of accomplished course requirements at FTS. Therefore, a db schema of the system has to provide storing data about all accomplished requirements at the level of a sole student. Data stored in the DW database are aggregated by the requirement types. According to the FTS regulations, a course requirement may be classified as: pre-exam (interm), exam, and final. Regardless this classification, our DW system allows the end users to create a deeper classification of requirements and consequently determine a level of data granularity of their own.

According to the user requirements, the DW system is supposed to extract data from an OLTP database that contains not only personnel data about students and data about final students' grades, but also data about each accomplished requirement, for each student, for each course, and for each semester. Requirements may be, for example, tests, colloquiums, assignments, projects, as well as the other requirements proposed by the FTS regulations.

Apart from having the OLTP database as a source, there is a need to collect data from various external sources organized as files. Since we developed a pilot project only, we decided to provide the access to only one external source. It is a file that is created during the process of student recruitment. It holds data about average grades and types of secondary schools graduated, for all student candidates. In the specification of Extraction, Transformation and Loading (ETL) process, we provided data validation procedures for all extracted data.

For the sake of the pilot project, we provided generation of the following reports:

- percentage of minimal positive grades (6s), for a given combination of various dimensions;
- percentage of maximal positive grades (10s), for a given combination of various dimensions;
- average grades by courses, students, and lecturers;
- standard deviation of grades by courses, students, and lecturers; and
- percentage of lessons attendance by students, courses, and lecturers.

In a full system implementation, the end-users should be provided by a considerable larger number of pre-created reports.

By the completion of the whole DW system, the end-users will have the possibility to use algorithms that provide complex quantitative analyses based on data about the execution of the lecturing process. Potential users of such DW system are from management staff at the level of university, faculties, departments and chairs. Performing such analyses may significantly improve the management strategic and tactical decisions, as well as the overall quality of the education process.

It is important to notice that our DW system is not aimed to store all the data belonging to student service, from OLTP database of FTS IS. Instead, it holds only that part of those data that is necessary to monitor the execution of course requirements.

4. Database Schema of the DW System

Database schema of our DW system, or DW schema for short, comprises nine relation schemes implemented as tables. Two of them are fact relation schemes, whereas the rest seven are dimensions. Besides, there are two additional relation schemes implemented as materialized views for aggregate data. Despite the fact that the structure of such DW schema is not very complex, it is a constellation schema, because it has more than one fact relation scheme sharing three common dimensions. We need these two fact relation schemes for the purpose of monitoring not only the execution of student requirements, but also for monitoring a segment of student recruitment process. The DW schema diagram is presented in Figure 1. In the following text, each of the presented relation schemes will be described in short.

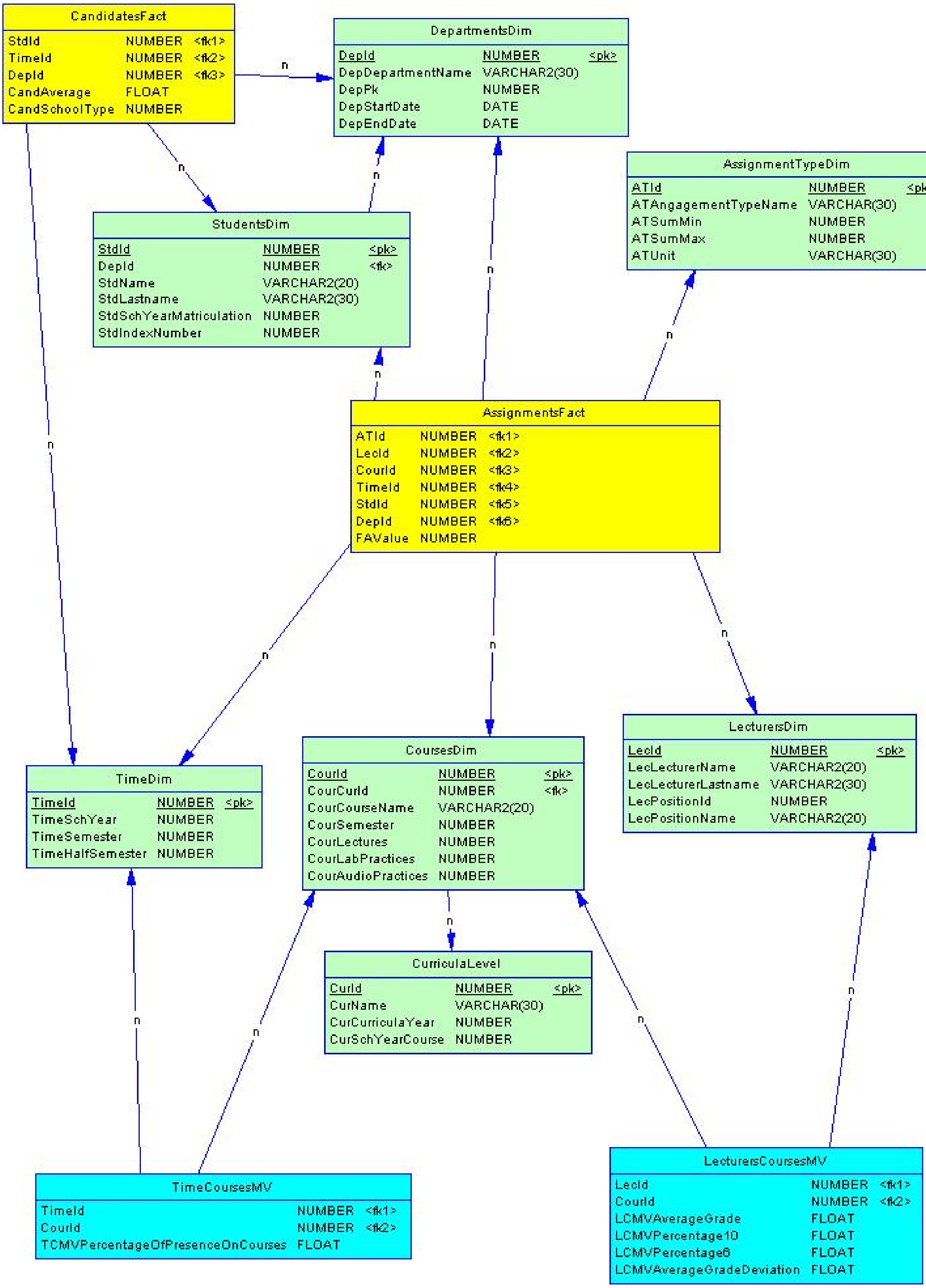

Figure 1. A diagram of the DW schema

The dimensions of the DW schema are:

- DepartmentsDIM,
- StudentsDIM,
- LecturersDIM,
- CoursesDIM with CurriculaLevel,
- AssignmentTypeDIM, and
- TimeDIM.

The fact relation schemes are:

- AssignmentsFACT and
- CandidatesFACT.

For all dimensions, except DepartmentsDIM, data updates are performed without storing history of changes.

DepartmentsDIM stores dimensional data about departments, i.e. faculty organizational units. Data for this dimension, as well as for all the other dimensions, are extracted from the source OLTP db. Apart from attributes that represent id and name of a department, the dimension comprises two additional attributes: *DepStartDate* and *DepEndDate*. They are used to record data change history, because the faculty organization structure is subjected to changes over time. Inserting a new tuple into the dimension results in setting *DepStartDate* to current date, and *DepEndDate* to null value. While "closing" the current department data, *DepEndDate* is set to the closing (i.e. current) date. By this, the tuple is declared to be out-of-date. The primary key of the dimension is a surrogate key comprising an attribute with generated auto increment values. It is transferred as a foreign key into the relation scheme AssignmentsFACT.

StudentsDIM stores dimensional data about students of the faculty. It is related to DepartmentsDIM by a foreign key, because each student is assigned formally to only one department. In this way, DepartmentsDIM is referenced from AssignmentsFACT twice. The first, directly, and the second, by means of StudentsDIM relation scheme. It leads to the violation of the third normal form (3NF) condition, and consequently raises the level of data redundancy, since the same department id data are stored twice. The existence of the transitive relationships is generally acceptable in DW systems. The relationship between AssignmentsFACT and DepartmentsDIM is introduced for the sake of improving query performances, by avoiding compulsory joining of these two tables via STudentsDIM. Each tuple from StudentsDIM comprises data about student's first and last name, id, and starting year of studying. Student id is a natural key, inherited from the source OLTP db. Data updates are performed without storing history of changes. Both of the fact relation schemes are related to StudentsDIM by foreign keys, because it is important to analyze a correlation between students' performance in the faculty education process, and successfulness and profile of the secondary education. By the rule, personnel student data in this dimension are never modified. Instead, the only operation is adding new tuples.

LecturersDIM stores dimensional data about lecturing staff of the faculty. Each tuple from LecturersDIM comprises data about names, position, and id. Lecturer id is a natural primary key, inherited from the source OLTP db. It is transferred as a foreign key into the relation scheme AssignmentsFACT. Data updates are performed without storing history of changes.

CoursesDIM stores dimensional data about faculty courses that are assigned to curricula. Therefore, we identify a two level dimension hierarchy of courses, represented by CoursesDIM, and their curricula, represented by CurriculaLevel. Each tuple from CoursesDIM comprises data about id, name, running semester, number of lessons per week, number of oral and laboratory exercises per week, and belonging curriculum. Each curriculum is described by its id, name, the first starting year, and the last running year. Course id is a surrogate primary key, comprising an attribute with generated auto increment values. The reason to introduce a surrogate key is that the natural key in the source OLTP db is too complex, since it is composed from a number of attributes. The primary key of CoursesDIM is transferred as a foreign key into the relation scheme AssignmentsFACT. Data updates are performed without storing history of changes.

AssignmentTypeDIM stores dimensional data about types of course requirements. Requirements may be: in-term (pre-exam), final exam, but also final score and grade, percentage of lessons attendance, etc. All course requirements are classified into types defined by the faculty management staff, according to their requirements. Each requirement type is described by its id, name, measurement unit, minimal allowed value, and maximal allowed value. Data updates are performed without storing history of changes. The primary key of AssignmentTypeDIM is transferred as a foreign key into the relation scheme AssignmentsFACT. For all the aforementioned dimensions except AssignmentTypeDIM, each dimension tuple corresponds to exactly one tuple from the corresponding relation from the source OLTP db. However, for AssignmentTypeDIM, a dimension tuple usually aggregates data about a number of particular in-term requirements from the source OLTP db. Therefore, there is a straightforward influence of the selected granularity of requirement types (i.e. the deepness of the requirement type classification) onto the aggregation level of fact data about the requirements from the source OLTP db.

TimeDIM is a time dimension aimed to span a number of years in monitoring the accomplishment of course requirements. The time in our DW system is not organized as a calendar with days, months, and dates. According to the management requirements, precise dates of the accomplishment of requirements are not necessary. However, the end-users are particularly interested in having a view onto the accomplished requirements by years, semesters and half-semesters. Therefore, TimeDIM comprises exactly those three attributes, by means of a satisfactory level of granularity is defined. The attribute *TimeId* is introduced as a surrogate primary key, instead of a natural key composed of the attributes *TimeSchYear*, *TimeSemester*, and *TimeHalfSemester*. *TimeId* is transferred as a foreign key into AssignmentsFACT. The data from TimeDIM are not modifiable.

AssignmentsFACT is a fact relation scheme that provides storing data about all the accomplished course requirements, at the level of particular requirement types. It is related by foreign keys to the dimensions: DepartmentsDIM, StudentsDIM, LecturersDIM, CoursesDIM, AssignmentTypeDIM, and TimeDIM. Its attribute *FAValue* provides storing measure data, each describing an accomplished requirement of the designated type. Measures are extracted as aggregate (derived) values from the source OLTP db. The aggregation is performed by all the requirements of the same requirement type, for a selected combination of all other dimensions. Such designed fact measure is semiaditive, because it cannot be summarized by any other combination of dimensions, if the requirement type is not qualified properly. By means of this fact measure, the data can be aggregated further, according to various criteria. The primary key of AssignmentsFACT is composed as a union of all foreign keys propagated from the dimensions. The data from AssignmentsFACT are not modifiable. In the process of data refreshment, the only allowed operation is the insert of new tuples.

Another fact relation scheme is CandidatesFACT. It stores data about average grades and types of secondary schools graduated, for all students. Average grade is an aggregated value for all years of secondary education. Secondary school type provides a classification of secondary schools by areas of interest, for example engineering schools, grammar schools, etc. It is under the responsibility of users to define such a classification. It is related by foreign keys to the dimensions: DepartmentsDIM, StudentsDIM, and TimeDIM. All necessary data are extracted from a file as an external source. The measure is typically nonaditive. The primary key of CandidatesFACT is a union of all foreign keys propagated from the three dimensions.

The DW schema comprises two materialized views: LecturersCoursesMV and TimeCoursesMV.

LecturersCoursesMV relates dimensions LecturersDIM and CoursesDIM. It stores aggregated data needed for analyses of the accomplished requirements grouped by courses and their lecturers. The data are derived from AssignmentsFACT. Apart from two foreign keys transferred from the dimensions, it currently comprises four aggregate attributes. Each tuple stores: percentage of maximal grades (10s), percentage of minimal positive grades (6s), average grade, and standard deviation of average grade. All these values are aggregated by applying group functions onto the AssignmentsFACT measure and qualifying the appropriate requirement types.

TimeCoursesMV relates dimensions CoursesDIM and TimeDIM. It stores aggregated data needed for analyses of the accomplished requirements grouped by courses and time units. The data are also derived from AssignmentsFACT. Apart from two foreign keys transferred from the dimensions, it currently comprises just one attribute that stores data about average percentage of lesson attendance. By the assumptions that: (i) percentage of lesson attendance must be a requirement type recorded in AssignmentTypeDIM and (ii) there are corresponding fact data about lesson attendance stored in AssignmentsFACT, it is possible to generate aggregated data for lesson attendance in TimeCoursesMV.

According to the future end-user needs, it is possible to extend both the materialized views by new attributes.

5. Conclusion

In the paper we presented some of the results embedded into the pilot project of our

DW system aimed at monitoring the faculty or university lecturing process. We developed a software system that provides reports of various types and analyses of the accomplished course requirements. The DW schema, presented in the paper, is designed to support an efficient execution of complex user queries. A considerable amount of time was spent for DW schema design. The quality of the DW schema designed has a substantial influence on the overall usability and durability of the system in usage. We believe that our database schema specification provides performing changes in the system evolution in a relatively easy way, and consequently extends its life time. The pilot system was implemented over Oracle DBMS and developed by the usage of *Oracle Warehouse Builder* [6], [7], [8].

Further development of our DW system may follow several directions. One of the goals is to extend the software so as to provide a wider number of OLAP and Data Mining analyses. For a commercial usage of the system it is compulsory prerequisite to perform a strategic feasibility study. The goal of such a study is to better motivate and actively involve management representatives into the requirements specification and software development process in order to fulfill a majority of requirements of strategic and tactical management staff in a proper way. One of the important critical success factors is also raising the motivation of potential end-users for an active usage of the system in their working processes. We believe that an implementation of the pilot system, which is partially presented here, is a possible way to reach that goal.

References

[1] Inmon W. H., *Building the Data Warehouse 3rd Edition*, John Wiley & Sons Inc., 2002.

[2] Korczak J., Oleszkiewicz, *Data Warehouse Structures for AML Applications*, 9th International Conference on Business Informatics - 1st CEE Symposium on Business Informatics, Vienna, Austria, February 25–27, 2009, Proceedings, pp. 111-118.

[3] Ahmada I., Azharb S., Lukauskisc P., *Development of a decision support system using data warehousing to assist builders/developers in site selection*, Automation in Construction, Vol. 13, Issue 4, July 2004, pp. 525-542.

[4] Polevoy A., Ingham J., *Data Warehousing: "A Tool for Facilitating Assessment"*, Frontiers in Education Conference, San Juan, Puerto Rico, November 10 - 13, 1999, Proceedings, Vol. 1, pp. 11B1/7 - 11B/11.

[5] Jacqueline F., Mindy W., Dalrymple M., *A New Focus for Institutional Researchers: Developing and Using a Student Decision Support System*, Annual Forum Paper, Minneapolis, May 17-20, 1998.

[6] Green R., Oracle9*i* Warehouse Builder: Implementation, Instructor Guide, Volumes 1 & 2, Oracle, 2003.

[7] Psomas N., Mitravinda P., Data Warehousing Fundamentals, Oracle, 2002.

[8] Shaw Lea M., *Data Warehouse* Database Design, Oracle, 2001.