Data Warehousing-A Case-based Courseware

[1]T.Krishna Kishore, [2]T. Sasi Vardhan,
[1]Dept. of CSE, St.Ann's College of Engineering & Technology, Chirala.
[2]Dept of CSIT, St.Ann's Engineering College, Chirala.

Abstract
Data warehousing is one of the important approaches for data integration and data preprocessing. The objective of this project is to develop a web-based interactive courseware to help beginner data warehouse designers to reinforce the key concepts of data warehousing using a case study approach. The case study is to build a data warehouse for a university student enrollment prediction data mining system. This data warehouse is able to generate summary reports as input data files for a data mining system to predict future student enrollment. The data sources include: (1) the enrollment data from California State University, Sacramento and (2) the related public data of California. The courseware is designed to build the data warehouse systematically using a set of 4 demonstrations covering the following data warehousing topics: fundamentals, design principle, building an enterprise data warehouse using incremental approach, and aggregation.

Keywords
data warehousing, information integration, Star schema, dimensional modeling, data mining.

I. Introduction

Every corporation/institution, small or big, has the need to make use of the large scale chronological data available, and hopefully turn it into a prediction/analytic model that supports decision making process. The data warehouse has been playing a critical role in data preprocessing and integration. It allows quick retrieval of input data for data mining or data analysis tools. The outcome of data reporting, data analysis and data mining can then be used for supporting decisions making on budget analysis, resource allocation, forecasting and prediction. To illuminate data warehousing basic concepts, design principle, and performance enhancement techniques, we developed this courseware. This web- based tool assists beginner data warehouse designers to reinforce their understanding of the basic design concepts of data warehousing via a case study. In this case study, the data sources include the student enrollment data from the California State University at Sacramento and, enrollment-related social and economical data of California. The main objective of this data warehouse is to prepare input data for an existing data mining system for student enrollment prediction [1]. Using the case study, we demonstrate the procedure to build a data warehouse and reveal some common incorrect practices which should be avoided in the design process. This project provides a self-paced learning tool not only to the students taking a course on data warehousing but also to the beginner data warehouse designers who have to build a data warehouse quickly from scratch. Fig. 1 shows the courseware tool's introduction page.

Fig. 1: Courseware tool

The interactive interface of this tool empowers user to do: report generation, knowledge assessment, tool evaluation, and illustration interaction. In addition, the tool explains strategies that enhance the performance of a data warehouse. The user can also observe experimental performance results of two different aggregation strategies presented later in this paper. This paper is organized as follows: section II briefly describes the framework and methodology for building the courseware, section III gives an overview of the case study data warehouse design, section IV discusses two strategies for data warehouse aggregation, section V includes the initial courseware evaluation from a data warehousing and data mining class in Spring 2010, and section 6 concludes the paper with the future work plan.

II. Framework of the Courseware

This courseware tool is a 3-tier web application designed using PHP, HTML, CSS and JavaScript. It is supported by MySQL queries and stored procedures. The data from the data sources resides on a MySQL server. The courseware includes a set of 4 demonstrations covering following topics: (1) fundamentals of data warehouse, (2) data warehouse design principle, (3) building an enterprise data warehouse using an incremental approach, and (4) aggregation. Each demonstration provides detailed description on building a data warehouse with text, diagram, and ready-to-go query runs. Furthermore, the theory behind each subject is outlined and a set of quiz problems are provided for self- evaluation.

III. Design

The first two important steps towards a sound design for a data warehouse are: (a) clearly identify the objective of the data warehousing project; (b) determine the user requirements through interviewing [8-10]. The objective in our case study is to prepare input data for data mining tools [4]. The primary data is the student enrollment data from California State University, Sacramento, and the secondary data is the enrollment related socio-economic facts from the California state agencies. The input data can be generated by a data mart in form of summary reports. There are two types of summary reports: (1) enrollment reports for graduate and undergraduate students for the last 30 years; and (2) annual reports of the demographic impact factors on enrollment. The following factors are identified to have significant impact on the student enrollment in California: income, employment rate, BS graduation rate, population, and tuition fees. Both type (1) and type (2) reports are required input for a data mining system to

output enrollment prediction reports for future 5 years [1].

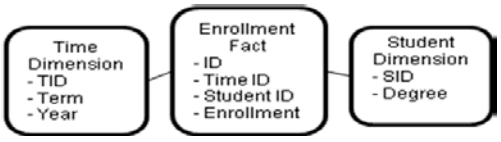

Fig. 2 : Dimensional model

Our case study data warehouse design is based on the dimensional modeling principles [9-11]. In a dimensional model, each group of dimension is placed in a dimension table; the facts are placed in a fact table. The result is a star schema. Our case study fact table contains a measurement: the enrollment count. These values can be segregated into useful subsets based on dimensions such as term, year, degree, student enrollment types, etc. An initial design of our star schema is shown in fig. 2. This data mart star schema is incrementally evolving in our case study. For example, the enrollment model is refined in the demonstration 2 of the courseware so that the data from other campuses of California State University can be integrated into it. A new dimension, University, is appended to the previous model to conform the data mart design, as shown in Fig. 3. This data mart is now ready to respond to the user queries to generate summary reports of the type (1).

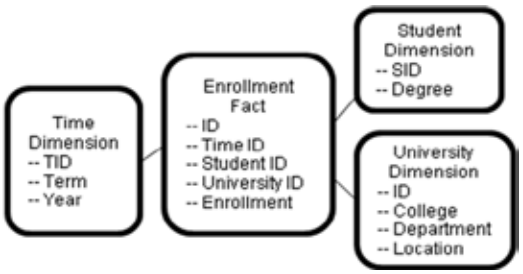

Fig. 3 : Refined dimensional model

Fig. 4 shows one such report in the form of a chart for the enrollment values of undergraduate students in the Computer Science Department for the past 30 years [1, 7]. This data can be converted into the input format required by the data mining system.

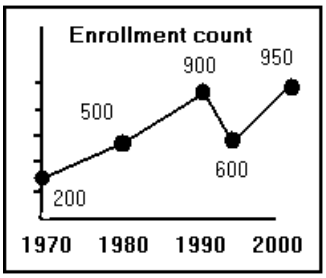

Fig. 4 : Enrollment trend

Similarly, a new star schema is built for type (2) report, the demographic impact factors on the enrollment of California State University. Since a subset of its dimensions are identical to the dimensions in the preceding star schema design, both of these models can be interlocked to form a single dimensional model as shown in fig. 5. As there are no other processes to be congregated on the enrollment data at this moment, Fig. 5 represents our final enterprise data warehouse design for the enrollment case study. Stipulating that new processes might be discovered for evaluation and measurement in the future, we can always refine the design using this incremental approach which is an effective way to avoid common data warehousing failures.

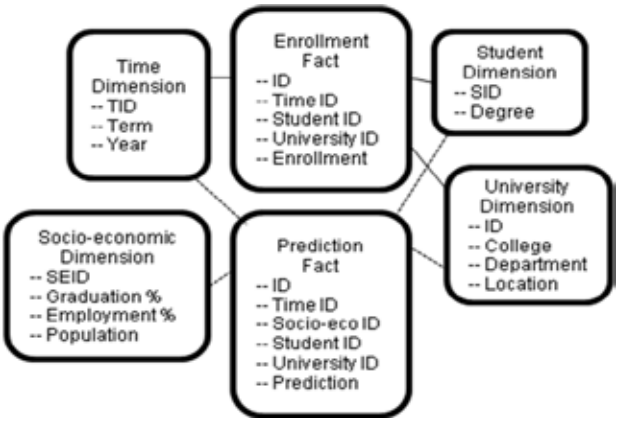

Fig. 5: Enterprise data warehouse

Answers to some questionsrequire combining information from multiple fact tables. The process of answering these questions is referred to as drilling across. To drill across, there must be equivalent dimensions related to the measurements in the fact tables. This is accomplished using the concept of conformed dimensions as discussed in [6].

IV. Aggregation

This section epitomizes two aggregation strategies that aim to improve the performance of a data warehouse. We demonstrate these strategies in the courseware to validate the performance boost of the enrollment data warehouse. Aggregate functions are acknowledged as the computational values that quantify a process of an enterprise, or as the input to the computational procedure that can be utilized to determine the measurements of a process. In the case study on enrollment, one of the processes of the university is to forecast the total number of students for the next term. The vital aggregate function here is the total of graduate and undergraduate students enrolled in all the departments of the university. Additional aggregate functions can include the average, maximum, minimum, or any other user-defined function. The first strategy is based on the opinion that the number of aggregate fact tables should be equal to the number of aggregate functions. One aggregate table is designed for each of the aggregate functions recognized. An aggregate table is designed as a fact table that has dimensions similar to the dimensions identified previously for the base fact table but with different grain (level of details). The aggregate star schema for the enrollment headcount is shown in Fig. 6. An aggregate star schema is designed for every aggregate function useful for an enterprise. Each aggregate table is then updated either by a stored procedure, a trigger or a small application program. Such programs are activated internally in the data warehouse whenever changes occur in the base fact tables. These programs calculate aggregate values in the aggregate tables by updating the old values per the new entries.

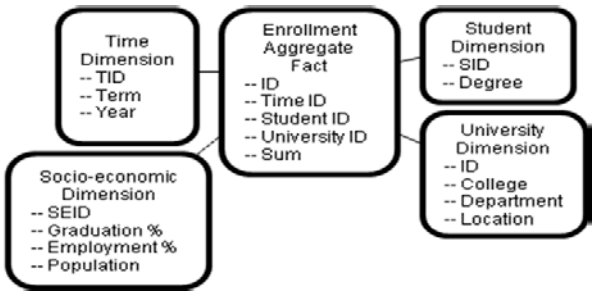

Fig. 6: Strategy 1 on aggregate schema

The second strategy is based on a new idea that an aggregate table is designed as a dimension table. One aggregate table contains keys to multiple aggregate functions referenced by the base fact table. The aggregate dimension table features the characteristics of aggregate functions. This strategy involves the cross tabulation process in which the aggregates are tabulated across a cube [3]. The aggregate values for every aggregate function are stored in a single column of the fact table. This table references the dimension tables: time, student classification, university, socio-economic dimension table and newly designed aggregate dimension table as shown in Fig 7. This approach eliminates the need of redesigning a new aggregate table for every new aggregate function. Only a row needs to be added in the aggregate dimension table with a unique key every time a new function is needed. The aggregate values for this function are calculated and stored in the base fact table. The maintenance of the aggregate table becomes easier than in the first strategy as only a few records need to be updated in the aggregate table every time there are modifications in the fact table.

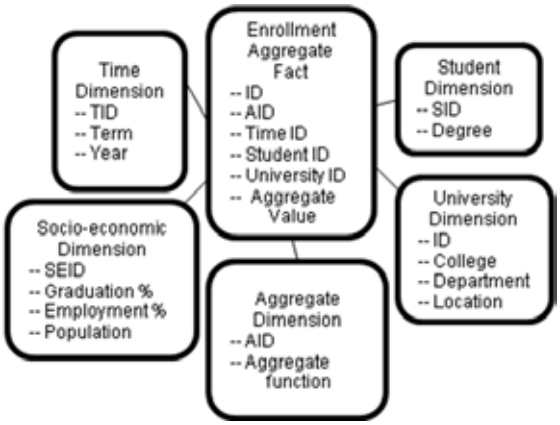

Fig. 7 : Strategy 2 on aggregate schema

By pre-aggregating the data in the fact table they reduced the amount of work the RDBMS must perform to respond to a query. As a general rule of thumb, the performance value of an aggregate fact table can be estimated if you look at how many base rows are summarized by each aggregate row. For example, a 20:1 ratio suggests there will be significant performance gains. Consequently, this method reduces the number of scans on the data warehouse records making the query executions more efficient. Hence, aggregation boosts up the performance of a data warehouse without additional cost for new hardware.

V. Courseware Assessment

One of our most important goals for developing this data warehousing courseware is to get students personally engaged in the use of the courseware to understand the fundamental concepts of data warehousing. We conducted a survey on this tool in a Data Warehousing and Data Mining class, CSC 177, in spring 2010 semester at California State University, Sacramento for the upper division undergraduate and graduate students in the Computer Science Department. The overall assessment from this student group on this courseware is extremely encouraging to us. The following is a brief summary of the positive feedbacks from the group:

1. Very accessible
2. Helpful to understand the fundamentals of DW
3. Very helpful to learn from the examples
4. Complement the course lectures very well
5. Easy to follow steps and illustrations
6. Simplicity and natural progression of the website
7. Add on quizzes can be handy in review for tests

The following list consists of the constructive comments given by the same group of students:

1. A data mining component can be added
2. A data preprocessing component is missing
3. Web presentation still need improvement
4. Explanation text is a little wordy
5. Result generated from the demo run needs further verification
6. More information should be given on the student enrollment prediction system to help understand the user requirements

For an enduring tool evaluation, we are adding an on- line feedback component for the tool users. This component collects tool evaluation data from the users and provides quantitative measurement on degree of satisfaction. It also allows the user to offer constructive suggestions to us in an on-going basis. We believe that this component is necessary for the success of a developing courseware.

VI. Conclusion

We presented a comprehensive architecture and functionalities of a web based tool for learning fundamental concepts of data warehousing. Although there are other online courseware tools such as [2] for various learning topics, we have not found an on-line courseware which is exclusively devoted for data warehousing such as ours. The main advantages of our tool include the usefulness, scope, and accessibility for beginner data warehouse designers and developers. This tool provides a whole development life cycle of a data warehouse using a case study with a set of supplementary examples. Also, this tool has the capability to provide a visual interactive user interface and personalized assistance to the beginner data warehouse developers in reviewing the key concepts of data warehouse design and development. Future research work includes strengthening of the case study structure, refinement of concept description and web presentation, and addition of new components on other related topics. The list of to-be-added case study topics includes: ETL, data mining, data preprocessing [9, 10].

References

[1]. Aksenova, Svetlana S., "Enrollment projection through data mining / Svetlana S. Aksenova", MS project report, CSUS, 2005.

[2]. Kevin C. Woods, "XML data representation and transformations for bioinformatics", MS project report, CSUS, 2007.

[3]. Jim Gray et al., "Data Cube: A Relational Aggregation operator Generalizing Group-By, Cross-Tab, and Sub-Totals", Kluwer Academic Publishers, 1997.

[4]. Jiawei Han, Micheline Kambe, Data Mining: Concepts and Techniques, 2nd Edition, Morgan Kaufmann Publishers, 2006.

[5]. Imhoff, Galemmo, Geiger, Mastering Data Warehouse Design, Wiley Publishing Inc., 2003.

[6]. Adamson, Mastering Data Warehouse Aggregates Solutions, Wiley Publishing Inc., 2006.

[7]. Computer Science Reports, Office of Institutional Research, California State University, Sacramento, [Online] Available : http://www.oi r.csus.edu/Reports/FactBook/DEPT/CSC.cfm

[8]. W. H. Inmon, "Building the Data Warehouse", John Wiley

& Sons, Inc, NY, 2005.
[9]. Ralph Kimball, Margy Ross, "The Data Warehouse Toolkit: The Complete Guide to Dimensional Modeling", Wiley Publishing Inc., 2003.
[10]. Ralph Kimball, Laura Reeves, Margy Ross, Warren Thornthwaite, "The Data warehouse Lifecycle Toolkit: Expert Methods for Designing, Developing, and Deploying Data Warehouses", Wiley Publishing Inc., 1998.
[11]. Christopher Adamson, Michael Venerable, "Data Warehouse Design Solutions", Wiley Publishing Inc., 1998.