The Data Warehouse Design Problem through a Schema Transformation Approach

V. Mani Sharma, P. Premchand
Associate Professor, Nalla Malla Reddy Engineering College Hyderabad, Andhra Pradesh, India
Dean, Dept. Of Computer Science and Engineering, Osmania University, Hyderabad, Andhra Pradesh, India.

Abstract—The concept of this thesis is to build a tool that helps companies to create their own multidimensional model from a collection of relational sources and to make a web based environment supporting flexible views of the MD databases. The MD databases are databases with an especial organizing their own data. This way of organizing data helps to users in concrete queries of huge amount of data, which extracts the information, with the help of non-experts users in MD databases, from relational database. With this information it has to make a new MD database. The new MD database is reorganizes the data of the relational database. The new MD database has also got correspondence with the relational one, in order to extract data from the relational to the MD model. Finally A DSS will rebuild for making MD queries support. Thus the Multi Dimensional by Example in short MDBE, is required for implementing the methodology for automatically deriving MD schemas from relational sources, bearing in mind the end-users requirement also. The proposed approach begins may gathering the end-user requirements which will be mapped over the data sources as SQL queries. Based on the constraints, query preserves to make MD schemas. MDBE automatically derives MD schemas as which agree with both the input requirements and the data sources.

Index Terms—Data Warehouse, DW Design, DW Schema Evolution, DW Designs Trace, MD Database,, MD Model, MDBE, OLAP, Relational DB, Schema Transformation, SAMDSG.

I. INTRODUCTION

Database creation is a complex task and involves tuning many parameters and can be done using any database creation. If many databases need to be created, the database creation wizard will need to be used repeatedly for each new database. This can be cumbersome especially when only a minimum number of essential parameters differ for each database. Data warehouses are databases that are loaded with subsets of relevant data from a source database. These warehouses may contain informational data extracted from operational data in the source database. The tables in warehouse databases are based on the relational database from the source database. Hence, it is essential to transform structures of the source database into structures for the warehouse. Nowadays, this is done by manually exploring and creating such a mapping. This process is both tedious and time-consuming. Also, users need to be technically trained to perform this task. There are a few other shortcomings in the present system. In the warehouse schema users may add new attributes to tables; these new attributes are the aggregates of the attributes of the master database. As a result, when data is copied from the master database to the warehouse database, data for these aggregate functions need to be computed at run-time during update, causing more delay. When this update is in progress, applications accessing the warehouse will not get access to accurate data, leading to lack of synchronization. MDBE follows a classical approach, in which the end-user requirements are well-known beforehand. This approach benefits from the knowledge captured in the data sources, but guides the design task according to requirements and consequently, it is able to work and handle semantically poorer data sources. In other words, providing high-quality end-user requirements, we can guide the process from the knowledge they contain, and overcome the fact of disposing of bad quality (from a semantically point of view) data sources.

A. Motivation

These problems form the basis and the motivation for this thesis. The Semi-Automatic Multidimensional Schema Generation (SAMDSG) tool is works towards providing an interface to accept required information from users to generate a new multidimensional database and creates an empty data warehouse. For a given source database, the SAMDSG tool aims at arriving at an appropriate mapping to create a warehouse structure. After a mapping has been formalized, tables for the new warehouse are created. Then, relevant data is automatically transformed from the source database to the newly created warehouse. A framework has been built to facilitate automatic updates of data warehouses. It has been designed in a way that the there can be multiple copies of the warehouse database, where each copy is an image of the warehouse database. Copies that need to be updated are taken offline and applications that need to access the warehouse database can now access any of the other image warehouses. The SAMDSG tool also helps to switch application – Image Switcher, switches between databases in a way that is totally transparent to applications so that they do not realize existence of multiple warehouse databases. As a result, using the SAMDSG tool by end user can directly create the desired warehouse schema. A major advantage in using this SAMDSG tool is to automate the SQL script generation for schema creation and data management. The use of such a
tool gives the users less time to design DW schema more accurately and efficiently rather than developing the code itself.

B. Problem Definition

Let us consider the following scenario: Users have a large database and need to store a sub-set of data in a warehouse. The process involved in doing that is:

- Explore the source database and decide what data needs to be represented in the warehouse
- Create data warehouse by tuning parameters using database creation wizard.
- Form SQL queries to create schema for the newly created data warehouse.
- Form SQL queries to transfer appropriate data from the source database to the DW
- Periodically manage the update of the data warehouse so that changes in the source database are reflected in the data warehouse.
- Manage multiple images of the data warehouse in order to ensure availability of DW at all times
- Provide applications with transparent access to multiple images of data warehouse
- This procedure assumes that end-users are familiar with SQL and mandates them to employ other available software to create a data warehouse. Automatic update of the data warehouse needs to be implemented using advanced database concepts. This is time consuming and requires extensive technical support for non-technical users.

Let us consider an example from a sample manufacturing industry database to explain the problem better. (Note: The manufacturing industry database is referred to as the ‘source database’ in this discussion.) In this scenario, users need to store performance related information of equipments in a DW. Several tasks need to be performed to successfully create such a data warehouse. In the source database, users need to select performance-related data stored in the equipment hour’s table, in attributes - Running Hours, Uptime, MTBF Predicted and MTBF Required. Also, it is essential that the primary key of the Equipment Hours table is a part of the data warehouse. This is presented in Figure 1.

Users need to create a data warehouse by tuning parameters, using Oracle’s database creation wizard. Users need to create the Equipment Hours table in the data warehouse, with only the required attributes and transfer corresponding data. It has to be noted that the Equipment Hours table has several referential integrity constraints, due to which all the parent tables need to be a part of the data warehouse schema. Parent tables of Equipment Hours table are presented in Figure 2.

As a solution to the problems mentioned above, the SAMDSG tool that generates a new data warehouse, performs schema mapping and builds a framework for automatic update of the data warehouse. The proposed tool is to allow users to select, extract, clean, and convert data from source system structures into consistent target warehouse data structures. Also, the data from the source database is populated into the target database. The data warehouse can be populated on a frequency that meets the organization’s needs. The tool navigates users in a sequence of interactive steps and accepts the parameters to create a new data warehouse. For a given source database, the tool helps users in arriving at an appropriate mapping to create a structurally related warehouse. After a mapping has been formalized, tables for the new warehouse are created. Then, relevant data is automatically transformed from the source database to the newly created warehouse. To enable automatic update of the warehouse database, a setup has been built that manages the periodic update of the warehouse. Applications access the data warehouse through an interface that provides a simple-to-use API. Users may create multiple images of the data warehouse using the tool. The support to update all the images is provided in the framework.

II. ARCHITECTURE OF THE SYSTEM

System architecture is a vital component of an application design. Architecture translates the logical design of the
application to a structure that defines the interaction of entities in the system. The proposed system design in this thesis in order top resolve the problem of the MD database is the one who is shown in the figure 2.5. This system is composed by different components: a relational database, the thesis tool, a data warehouse, an OLAP server and a web environment.

- The Relational database is a database stored in a database manager. This database manager has to be a PostgreSQL manager.
- The thesis tool is a program built in JAVA. This tool has to connect to the relational database manager in order to read a database. It has to connect to the data warehouse manager in order to create a new database and introduce data. The thesis tool has to access to the OLAP server directory, in order to store the OLAP schema and the MDX query.
- The Data warehouse is the database that the thesis tool has created. It is store in a relational database manager. This database manager has to be a PostgreSQL manager.
- The OLAP server is an open source server belongs to the platform business intelligence, which provides a complete suite to analyses business data. This OLAP server is a module that runs into a Tomcat web server.
- The Web environment is an access to the data through http channel.

III. DATABASE CREATION

Database creation is a complex task and involves tuning many parameters. This chapter describes how SAMDSG provides a graphical interface to accept the essential parameters to generate script files, which can be executed from the command prompt to create a new database. One of the overall goals of this project it to be able to create and populate databases for data warehouses. This involves creating a blank database, in which data may be filled in. In this chapter, we discuss how our tool helps towards reaching the first step of the goal. User interface design describes how scripts are generated and executed to create a new warehouse database.

IV. SEMI AUTOMATIC GENERATION OF WAREHOUSE SCHEMA

As stated previously, the goal of this projects it to be able to create and populate databases for data warehouses. This also involves creating a data warehouse schema and loading the warehouse with subsets of relevant data from the source database. The proposed tool to allow users to select, extracts, clean, and converts data from source system structures into consistent target warehouse data structures. Also, the data from the source database is populated into the target database. The data warehouse can be populated on a frequency that meets the organization’s needs. A data warehouse depends totally on its ability draw information from across the organization. The proposed tool provides users with the ability to connect to any source database to draw the required information. Information is drawn into the warehouse by consolidating and cleansing data before populating the warehouse database. This is done automatically after users finalize the target database schema and the mapping with the source database schema. Data warehousing involves mapping subsets of relevant data from the source database to the target database. The target database schema is designed based on the data that is being transported from the source database. Hence, there is a mapping between the structure of the source database and the target database. This mapping is termed as Schema Mapping. For a given source database, the tool helps users in arriving at an appropriate mapping to create a semantically related warehouse. After a mapping has been formalized, a new warehouse is created. Then, relevant data is automatically transported from the source database to the newly created warehouse. Each mapping created for a source and target database is stored in XML files. This ensures that users can make further changes to the mapping by loading them at a later time.

The MDBE Algorithm

Declare MDBE ALGORITHM as
1. For each table in the FROM clause do
(a) Create a node and Initialize node properties;
2. For each attribute in the GROUP BY clause do
(a) Label attribute as Level;
(b) node = get node(attribute); Label node as Level;
(c) For each attr2 in follow conceptual relationships(attribute, WHERE clause) do
   i. Label attr2 as Level;
   ii. node = get node(attr2); Label node as Level;
3. For each attribute in the SELECT clause not in the GROUP BY clause do
(a) Label attribute as Measure;
(b) node = get node(attribute); Label node as Cell with Measures selected;
4. For each comparison in the WHERE clause do
(a) attribute = extract attribute(comparison);
(b) if !(attribute labeled as Level) then
   i. Label attribute as Descriptor;
   ii. node = get node(attribute); Label node as Level;
   (c) For each attr2 in follow conceptual relationships(attribute, WHERE clause) do

i. if !(attribute labeled as Level) then
A. Label attribute as Descriptor;
B. node = get node(attribute); Label node as Level;
5. For each join in the WHERE clause do
(a) /* Notice a conceptual relationship between tables may be modeled by several equality clauses in the WHERE */
(b) set of joins = look for related joins(join);
(c) multiplicity = get multiplicity(set of joins); relationships fitting = fg;
(d) For each relationship in get allowed relationships(multiplicity) do
i. if !(contradiction with graph(relationship)) then
A. relationships fitting = relationships fitting + frélationship;
(e) if !(sizeof(relationships fitting)) then return notify fail("Node relationship not allowed");
(f) Create an edge(get join attributes(set of joins)); Label edge
to relationships fitting;
(g) if (unequivocal knowledge inferred(relationships fitting)) then propagate knowledge;
6. for each g in New Knowledge Discovery(graph) do
(a) output += validation process(g); //A detailed pseudo-code of this function can be found in section 3.4.2
return output;
This tool provides users with a graphical interface to perform Schema Mapping.
• Select source database: Specify the master database for which the warehouse needs to be created
• Select tables: Users may select only relevant tables from the source database to be a part of the target database. Figure 5.2. presents a mapping diagram to show an example.
• Enforce referential integrity: If child tables are selected to be a part of the target database, the corresponding master tables need to be selected too. The master tables are internally computed by the tool and are selected automatically.

V. CONCLUSION
The SAMDSG thesis tool plays a vital role in providing support for automated data warehouses [1]. It is simple to use, highly interactive and provides an easy means to creating a new data warehouse. It also acts as a reliable tool to quickly explore schema of the source database in order to generate schema for the data warehouse. The SAMDSG tool underlying complex mechanisms from its users, except where it is absolutely appropriate and necessary to expose them. In effect, even non-technical users can create, populate and update data warehouses with minimal time and effort. Attributes from source tables can be mapped into new attributes in the warehouse database tables using aggregate functions. Then, relevant data is automatically transported from the source database to the newly created warehouse. The tool thus integrates warehouse creation, schema mapping and data population into a single general purpose tool. This tool has been designed as a component of a framework, whose users are Database Administrators. They will also be able to synchronize updates of multiple copies of the data warehouse.

Warehouse images that need to be updated are taken offline and applications that need to access the data warehouse can now access any of the other image warehouses. The Image Switcher built into this framework switches between databases in a way that is totally transparent to applications so that they do not realize existence of multiple copies of the data warehouse. It also ensures that ongoing transactions on a particular database are not interrupted when the database is scheduled to be taken offline. The thesis tool is system independent. This gives it advantages such as portability and wide application. This tool can access any database with minimal effort since there is no hard coding of information in the application.

REFERENCES


AUTHOR’S PROFILE

I Mani Sarma. V received degree in Master of Computer Applications (MCA) from madras university in 1998, Chennai, and Master of Technology in Computer Science ALCCS from IETE in 2010, New Delhi respectively pursuing Ph.D Degree in Computer Science and Engineering as a research scholar 2007-08 from Acharya Nagarjuna University(ANU), Guntur, Andhra Pradesh. Since 1998 I have been working as a senior faculty member in Holy May Group of Institutions, Hyderabad, Andhra Pradesh, India. Presently I am working as Associate Professor in Nalla Malla Reddy Engineering College, Hyderabad, and Andhra Pradesh, India. My Research interest includes Data Warehousing & Data Mining, Parallel and Distributed data Mining, and Advanced Databases Systems. I have published papers on national and International journals publications like ACST, JCA and IJICT and attended several important national and International conferences to enhance my Ph.D qualification.

Prof. P. Premchand. Professor. Department of Computer Science & Engg., University College of Engineering, Osmania University, Hyderabad-500007, Andhra Pradesh, India. Guided two candidate towards the Award of Ph.D degree from JNTU, Hyderabad.