Teaching Effective Methodologies to Design a Data Warehouse

Behrooz Seyed-Abbassi
Department of Computer and Information Sciences, University of North Florida
Jacksonville, Florida 32224, United States

Abstract

An important component for the students in the advanced database class at the University of North Florida involves the development of a data warehouse that is efficiently designed and effectively optimized for data retrieval and statistical analysis for Decision Support Systems (DSS) and Executive Information Systems (EIS). This paper describes the methods utilized to help students understand the considerations in the design process of a data warehouse. The methodologies involve the commonly used star schema and the snowflake schema as well as other alternative techniques to teach students about the essential factors to consider when designing a data warehouse.

Keywords: Teaching method, data warehouse, database design, star schema, snowflake schema, SQL

1. INTRODUCTION

Database Systems II is an advanced database course developed in 1999 for undergraduate students in the Information Systems (IS) program at the Department of Computer and Information Sciences of the University of North Florida. Since the addition of this course, it has become a popular major elective for students desiring to apply the basic knowledge learned in the first database course to more advanced database areas. In the first course, Database Systems, the topics include entity relationship, relational database system, SQL, normalization, concurrency, optimization, database design methodology, and application interfaces. After successfully completing this course, students often elect to take Database Systems II to study further utilization of database software, techniques in design and modeling, object oriented databases, web database interfaces, and data warehousing. The additional experience gained in the second database course provides students with a broader underpinning of database knowledge when they enter the job environment after graduation.

This paper describes part of the teaching methodologies used in the subject of data warehousing which is becoming an increasingly significant field in the area of database. Since the introduction of the data warehouse concept, the development of a warehouse to serve as a centralized data collection and repository for DSS/EIS is becoming more commonplace among a variety of organizations (Gatziu 1996). As data warehouses become more prevalent as a means to handle the vast amount of historical data on computers, it is important for students to be introduced to the considerations in the design of a warehouse to optimize the design process (Springsteel 2000). Of the 16-week semester, approximately 5 weeks are allocated to the foundations, design, and implementation of the data warehouse. During this time, the students learn the fundamentals of data warehousing which include a history, data requirements, historical data vs. operational data as well as information about determining the need, gathering data from different sources, and filtering the data (Rob 2000). The life cycle of a data warehouse is discussed with emphasis on the following phases: requirements analysis and specification phase; logical database design phase; business process modeling phase; physical database design phase; and database implementation and maintenance phase.

After studying the basic principles of a data warehouse, the students learn more details about the design of the warehouse using a complex relationship structure. Building on the knowledge and experience in relational database, entity relationships, SQL, and normalization from the first database course, the design considerations are presented by introducing the star schema and the snowflake schema. Then, other techniques are explored as potential methods that may also be utilized to improve the design structure. Based on the information from the lectures, the students in teams of two or three develop a data warehouse using Rational Rose, System

Architect, or ERwin for the design phase as well as the implementation phase.

2. STAR SCHEMA IN DATA WAREHOUSE DESIGN

In the design of a data warehouse, the fundamental structure utilized in a relational system is the star schema. This schema has become the design of choice because of its compact and uncomplicated structure that facilitates the query responses on the data. For students, this schema is simple to understand and provides a good introduction to the framework of a warehouse. Using a diagram of the sample star schema shown in Figure 1, the following information and requirements are described in terms of the star structure utilizing a central table as a complex relation with one level of surrounding tables.

I. Relevant information and requirements are gathered from an existing operational database in the form of records and stored in the data warehouse for future mining, analysis, and evaluation.
II. The collections of records are stored in two types of tables: one fact table and many dimensional tables.
III. The operational data, grouped into related records as historical data, are distributed into their related dimensional tables according to their types.
IV. The fact table holds the keys from each dimensional table.
V. In gathering the operational data for data warehouse, the following rules are highlighted.

i. Identical operational data from different sources for data warehouse filtering may have different formats in structures that require common representations for businesses.
ii. The function or process-oriented data in the operational database is stored as subject-oriented data in the data warehouse to facilitate decision making.
iii. Daily transactional data from the operational database are stored in dimensional tables in the data warehouse with a time dimension.
iv. The operational data with respect to update operations (Insert, Delete, and Change) are stored as read only data in the data warehouse.

VI. The relationship between each dimensional table and the fact table is a one-to-many relationship with each record in the fact table containing a unique key representing the group of records from each dimensional table.
VII. The schema design should be kept as simple as possible.
VIII. Query optimization should be considered during the design phase.

The configuration of the fact table holding a key from each dimensional table to associate all the related dimensional tables together results in the star schema. The dimensional tables surrounding the fact table can be viewed as a single level access for retrieval in a join operation. PowerPoint presentations are used for the student lectures to demonstrate the textual and graphical sample data warehouse. When necessary, SQL demonstrations are used for querying the sample data warehouse information. The sample data warehouse is also available to the students for testing different queries from data warehouse tables. By segmenting the areas of the schema into the components (through overlays), the students gain an overall perspective of a data warehouse and understand that it may contain many star schemas with each one supporting different dimensionalities through association of related records in the fact table.

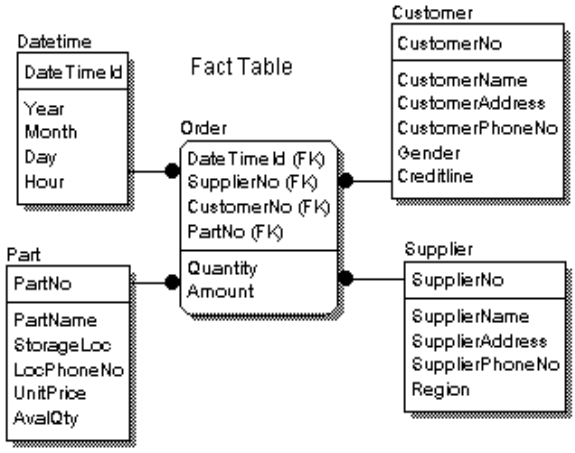


Figure 1: Star Schema with one fact table and four dimensional tables

3. CONSIDERATIONS BEYOND THE STAR SCHEMA

After studying the basic star structure, the students explore the structural issues that may be encountered when designing a data warehouse, such as the handling of multi-values and duplicate values (Inmon 1996; Kimball 1998). The general technique for storing multi-valued information in the records of a data warehouse is through higher normalization, which can be applied to the attributes in the dimensional table to reduce the redundant information. The resulting structures are called snowflakes and the utilization of snowflakes provides an excellent opportunity to review the normal form knowledge (1NF, 2NF, 3NF, and BCNF) learned from the first database course. Other design techniques, such as the introduction of null values, allowance for tuples with duplicate values, and denormalization are discussed with the advantages and disadvantages of the different techniques. The students are encouraged to be innovative in the design methodologies to apply to the development of their own data warehouse.

Snowflakes in Data Warehouse Design

In a data warehouse design, the star schema may be changed through the process of higher normalization (3NF) to a more complex structure, which adds extensions called snowflakes to handle multi-values and duplicate values in the database (Chaudhuri 1997). In the new schema, the original table is decomposed into two or more tables with the duplicate values moved to one of the new tables. This is demonstrated to the students by examples based on the star schema in Figure 1. The following are two of the examples utilized in the lecture presentation.

1. If the part location in the Part dimensional table supports multiple storage locations with multiple phone numbers, the multiple storage locations (StorageLoc) could be handled by keeping the storage data in a separate dimensional table (Location) and associating the key of the Part table to the storage table. Similarly, a new table named LocationPhone with multiple phones may be created for each storage location with an associated key from the Location table. The relation of the new tables from the fact table can be viewed as a two level access for storage location information during the join or a three level access for retrieval of phone number information during the join.

2. If the Customer dimensional table maintains a more detailed address structure (such as attribute names of Number, Street, City, State and ZipCode), this might result in dependency among the attribute names in the Customer table that will result in an unnormalized dimensional table in the star schema. The action of normalization may be applied to the tables. Assuming that there is data dependency, the ZipCode and the State may be stored in a separate table. The ZipCode from the Customer table will associate the dimensional table to the ZipCode/State dimensional table (State).

Students are shown that the issues of multi-values and duplicate values in the star schema in examples 1 and 2 are alleviated by decomposing the related dimensional tables into more normalized tables, as shown in Figure 2. The normalization of the star schema from Figure 1 converts the related dimensional tables to a higher normal form table(s). The higher normalized star schema structure results in more than one level of tables associated to the fact table. The new structure is referred to as a snowflake schema (Colliat 1996; Paplpanas 2000). This schema is more organized than the star schema in that it reduces the duplicates, but it also results in more dimensional tables from the decomposition of original table(s).

Although the snowflake structure can handle the issues of multi-values and duplicate values, the higher normalization of the design also increases the complexity of the design structure by adding more levels of tables. The expansion of the number of tables dramatically increases the number of joins required for queries thus prolonging the query retrieval time. The end result is a decrease in the efficiency of the query response time for the Decision Support System. By building the snowflake schema on top of the star schema, the students become aware of the potential overall effects of an increased number of joins on the query retrieval time and of the additional complexity in the snowflake schema through higher normalization.

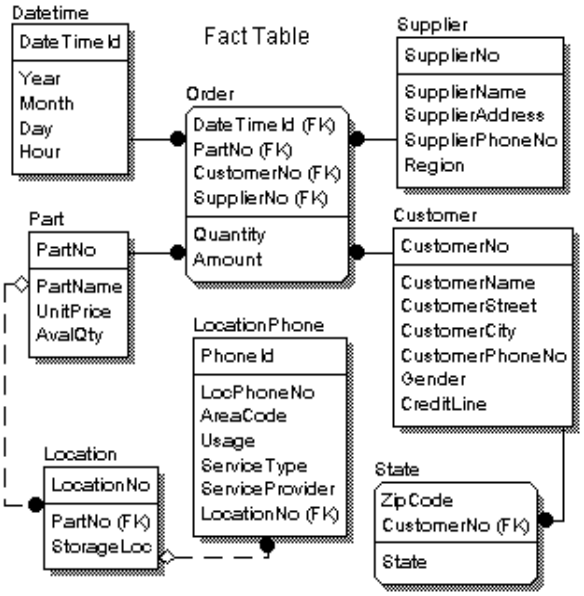


Figure 2: Snowflake Schema with a fact table and dimensional tables

Extended Star Schema with Known Number of Multi-Valued Attributes

After learning about the more prevalent star and snowflake schemas, the students are introduced to alternative methods for handling multi-values and data normalization. During the denormalization, when the number of multi-valued attributes is known, one method might be to use an extended star schema to preserve the structural efficiency of the star design by allowing multi-valued data to be present in the same dimensional tables (Seyed-Abbassi 2001). The historical data used in the data warehouse mimics the attribute names, structures, and data values used in the operational database. Because the design of a data warehouse is based on the use of one or more existing operational database systems, the data warehouse designer should have a thorough knowledge of the existing operational database record structures, attributes, and distribution of the tuples as well as the multi-valued fields or duplicate attribute names that are going to be used for modeling the data warehouse.

The extended star schema using a known number of multi-valued attributes is demonstrated to the students through examples based on the snowflake schema in Figure 2. One example considers the Part table and its structure. Rather than transforming the star schema to a snowflake by utilizing higher normalization, the design is kept in the star schema structure by employing the multiple values, such as storage locations or location phone numbers, with a special arrangement similar to normalization within the Part table. Because the warehouse designer knows in advance the number of the attribute names for StorageLocations (StorageLoc) and LocationPhoneNumbers (LocPhoneNo), the Part table can be designed to contain the necessary numbers of StorageLocation*x* (StorLoc*x*) and LocationPhoneNumber*x* (LocPhnNo*x*) attributes. At the physical level, the table can be represented by multiple attributes as shown in Figure 3.

Part Table

PartNo	PartName	Stor Loc1	Stor Loc2	Stor Loc3	…
P1	Pname1	L1	L2	L3	…
P2	Pname3	L1	L5	Blank	…
…	…	…	…	…	…

…	LocPhn No1	LocPhn No2	LocPhn No3	Unit Price	Available Qty
…	Pn1	Pn2	Pn3	UP1	Q1+Q2+Q3
…	Pn1	Pn5	Blank	UP2	Q4+Q5
…	…	…	…	…	…

Figure 3: Sample of Part Table Using a Known Number of Attribute Names

The creation of the Part table in a dimensional table with three storage locations (StorLoc) and phone numbers (LocPhnNo) gives students a vertical expansion and view of the table without higher normalization. Using PowerPoint for the lecture presentation, the students are shown different tuples for part names that can have different storage locations and location phone numbers as well as some blank attribute values in some of the tuples. The designer has allowed a maximum of three locations and location phone numbers for each part record based on knowledge from the operational database. Due to limitations on the column size within this paper, Figure 3 contains only one phone number per location. Multiple phones per location may be considered by using multiple columns for different phones.

The primary disadvantage of this method is that some of the records in the Part table may have less numbers of storage locations or location phone numbers than the maximum number of storage locations or location phone numbers in the Part table. In that case, some of the storage locations or location phone numbers will contain no value or null values in the related attribute area in the record(s). On the other hand, the advantage of this method is a compact design (denormalized) having fewer tables by allowing multiple attribute names and values to be present in one table, such as the Part table, as opposed to two or more tables using higher normalization. This method also supports the simplicity and efficiency of the design of star schema structure that results in more optimized query retrieval requiring less number of joins on the dimensional tables.

Extended Star Schema with Unknown Number of Multi-Valued Attributes

Depending on the requirements, the students are introduced to a technique for handling an unknown number of multi-valued attributes using a different form of denormalization. In the operational database system, the expansion of a company may result in parts that are stored in different locations and in new locations that are acquired to store the parts. Filtering of these types of multi-value data to add as historical data to the data warehouse will result in higher normalization and a snowflake schema. When the number of duplicate attributes or possible multiple values for the tuples of the data warehouse from different tuples of the operational databases are not known, the data warehouse can be designed using the historical data from the Part table and its locations and allowing for future expansion of historical data to be handled through distribution of multiple locations in a tuple of the dimensional tables rather than the attributes of the Part dimensional table to preserve the star schema (Seyed-Abbassi 2001). An example of this technique is demonstrated to the students as shown in Figure 4. The Part dimensional table, Storage Location (StorLoc) and LocationPhoneNumber (LocPhnNo) attribute names will be repeated only once.

Part Table

PartNo	Rec No	Part Name	Stor Loc	LocPhn No	Unit Price	Aval Qty
P1	1	Pname1	L1	Pn1	UP1	Q1
P1	2	Pname1	L2	Pn2	UP1	Q2
P1	3	Pname1	L3	Pn3	UP1	Q3
P2	1	Pname3	L1	Pn1	UP2	Q4
P2	2	Pname3	L5	Pn5	UP2	Q5
…	…	…	…	…	…	…

Figure 4: Sample of Part Table Using Unknown Number of Attribute Names

Each multiple storage location and location phone number will participate in a new tuple of the Part dimensional table. This method will allow for future expansion of a new storage location and location phone number by adding the new historical tuple to the Part dimensional table. To recognize all the records with the same PartNo, a new attribute with the name RecordNo (RecNo) is used to represent the Part records with

duplicate values that are added to the data warehouse table. As shown in Figure 4, the PartNo for first three records is P1, but each part is stored in different locations (L1, L2, L3) with different location phone numbers. The attributes, RecordNo and PartNo, are used as a composite key to provide a unique identifier in data warehouse tuple or record retrieval.

The advantage of this method is that future expansion of new parts and their locations in the operational database can be handled easily in the data warehouse tables. Storing multi-values in data warehouse star schema provides a more optimized structure than the normalized schema design of snowflakes. As shown in Figure 4, the available quantity for each stored location can be stored separately. The disadvantage of this method is that there will be some duplicate values.

Other Design Methodologies

After the basic data warehouse design lectures are completed, the students are introduced to aggregated information and aggregated tables. Multiple fact tables in the form of aggregated tables with process information are discussed to show how the data analysis process can be optimized. Instead of computing the values by accessing a lower level detailed fact table, the user can access to a summarized fact table. Another advanced technique that is considered is combining the star and snowflake (starflake) schema when the snowflake schema is not in third normal form. The resulting starflake can be designed with collections of star schema and snowflake tables. Techniques in using different OnLine Analytical Processing (OLAP) structures**,** such as OLAP client/Server architecture, server arrangement, server with multidimensional data store arrangement, server with local mini data marts, are discussed to inform the students about the tools for DSS (Teory 1999). Multidimensional OLAP (MOLP) and Relational OLAP (ROLAP) are also considered.

4. EFFECTS OF DESIGN METHODOLOGY ON DATA MINING

The design methodologies are evaluated for query retrieval and performance to help the students understand the importance of the design in the overall effectiveness of the data warehouse for the final goal of data mining. The effect of queries on the different designs during information retrieval and a simple data mining are demonstrated for the students. During the lectures, a small portion of the implemented data warehouse is selected for query processing for demonstration purposes. Utilizing the live connection in the classroom to sample the data warehouse tables, the following simple example is shown using different queries for each design method.

Example: In the Part table, retrieve all the part information, storage locations, and location phone numbers for a particular part, such as Pname1.

With the star design, the query processing for the example retrieves the information from product table without any performance problem.

```
Select PartNo, PartName, StorageLoc,
        LocPhoneNo from Part
        Where PartName = "Pname1";
```

In the snowflake design, the students are shown the following query that requires a minimum join of three tables to retrieve the necessary information.

```
Select PartNo, PartName, StorageLoc,
        LocPhoneNo
        from Part, Location, LocationPhone
        Where PartName = "Pname1"
                Part.PartNo = Location.PartNo
                Location.LocationNo =
                LocationPhone.LocationNo;
```

In this case, if any of the dimensional tables around the fact table participate in the join, it will result in a join of at least five or more tables that could result in performance issues. If the data warehouse does not use an optimizer at the physical level, it must join the tables by using the Cartesian product which slows the process immensely.

Using the denormalized case with a known number of duplicate attribute names, the following query is applied to the example and is retrieved from only the Part table based on the condition PartName = "Pname1".

```
Select PartNo, PartName, StorLocn,
        StorLocn,..., LocPhnNon,
        LocPhnNon,... from Part
        Where PartName = "Pname1";
```

Using the design with unknown number of duplicate attribute names, the following similar query is used.

```
Select PartNo, RecNo, PartName, StorLoc,
        LocPhnNo from Part
        Where PartName = "Pname1";
```

By demonstrations of various queries similar to the ones given above, the students learn the effects of normalization and denormalization in the design of a relational data warehouse. More complex examples, such as finding the total number of customers whose orders were filled from storage locations with the same zip code or area code in the last six months, are discussed during the lectures. This encourages them to consider the end results in the design process.

5. MOVING TOWARDS IMPLEMENTATION

After studying the steps in the design of a data warehouse using normalization and denormalization, the students are assigned to design and implement their own data warehouse. Due to the time constraints and benefits of group work, the students work on the project in groups of two or three to develop a small data warehouse using available SQL databases and design tools. An operational database is provided for the students to retrieve the necessary data. After completing the implementation of the warehouse, the teams are asked to run a number of queries to test the different retrieval on simple and aggregated data as part of their presentations.

6. CONCLUSION

The students in the advanced database course at the University of North Florida are introduced to the factors involved in the development of a data warehouse. One of the most critical components is the provision of a design that will produce optimal functioning of the data retrieval needs for the DSS or EIS. As described in this paper, a variety of design methodologies are presented to help students understand the importance of looking beyond a standard design. Through the experience of applying the design techniques to the development of a data warehouse, the students learn about the disadvantages and advantages of different methodologies. They are encouraged to be flexible in their approach to data warehouse design and to utilize design structures that will result in more optimized warehouses for the storage and retrieval of data through data mining by the Decision Support System or Executive Information System. This introduction to the emerging technology involved in data warehouse development provides valuable knowledge for students as they prepare to enter the business environment after graduation.

7. REFERENCES

Chaudhuri, Surajit and Umeshwar Dayal, 1997, "An Overview of Data Warehousing and OLAP Technology," *ACM SIGMOD Record*, 26(1), pp. 65-74.

Colliat, George, 1996, "OLAP, Relational, and Multidimensional Database Systems," *ACM SIGMOD Record*, 25(3), pp. 64-69.

Gatziu, Stella, Manfred Jeusfeld, Martin Staudt, and Yannis Vassiliou, 1996, "Design and Management of Data Warehouses Report on the DMDW'99 Workshop," *ACM SIGMOD Record*, 28(4), pp. 7-10.

Inmon, W. H., 1996, *Building the Data Warehouse*, John Wiley and Sons.

Kimball, Ralph, 1998, *The Data Warehouse Lifecycle Toolkit*, John Wiley and Sons.

Paplpanas, Themistoklis, 2000, "Knowledge Discovery in Data Warehouses," *ACM SIGMOD Record*, 29(3), pp. 88-100.

Rob, Peter and Carlos Coronel, 2000, *Database Systems: Design, Implementation, and Management*, Course Technology, Cambridge.

Seyed-Abbassi, Behrooz, 2001, "Designing an Optimized Data Warehouse for Data Mining," *Proceedings of 5th World Multiconference on Systemics, Cybernetics and Informatics,* July 22-25, pp. 214-219.

Springsteel, Frederick, Mary Ann Robberts, and Catherine Ricardo, 2000, "The Next Decade of the Database Course: Three Decades Speak to the Next," *SIGCSE Bulletin*, 32(1), March 8-12, pp. 41-45.

Teory, Toby, 1999, *Database Modeling & Design*, Morgan Kaufmann.

Figure 1: Star Schema with one fact table and four dimensional tables

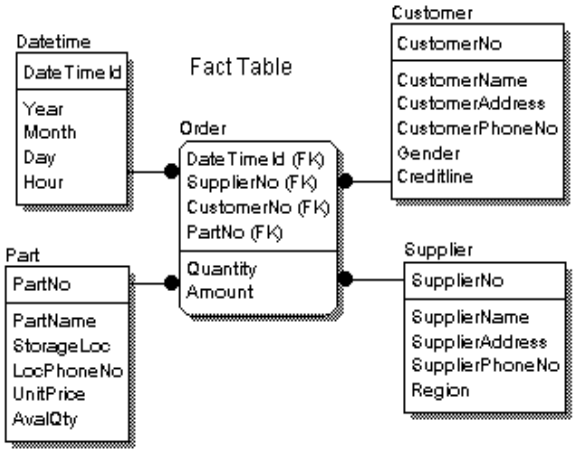
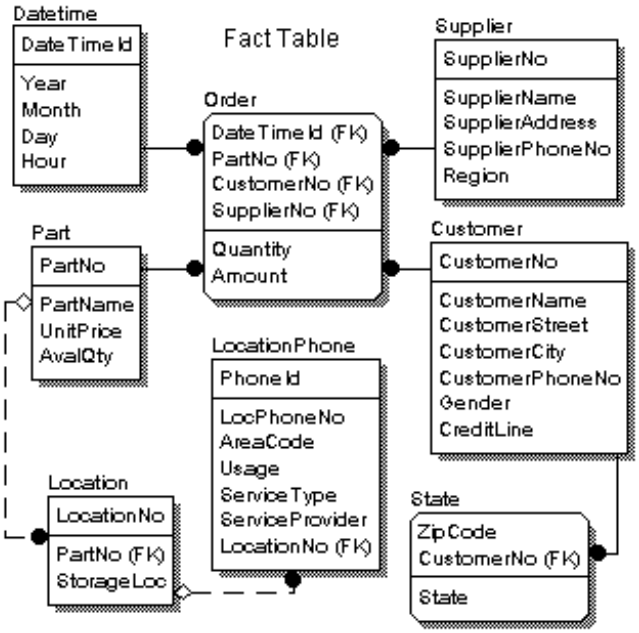


Figure 2: Snowflake Schema with a fact table and dimensional tables



Part Table

PartNo	PartName	Stor Loc1	Stor Loc2	Stor Loc3	…
P1	Pname1	L1	L2	L3	…
P2	Pname3	L1	L5	Blank	…
…	…	…	…	…	…

…	LocPhn No1	LocPhn No2	LocPhn No3	Unit Price	Available Qty
…	Pn1	Pn2	Pn3	UP1	Q1+Q2+Q3
…	Pn1	Pn5	Blank	UP2	Q4+Q5
…	…	…	…	…	…

Figure 3: Sample of Part Table Using a Known number of Attribute Names

Part Table

PartNo	Rec No	Part Name	Loc	LPhN	Unit Price	Avl Qty
P1	1	Pname1	L1	Pn1	UP1	Q1
P1	2	Pname1	L2	Pn2	UP1	Q2
P1	3	Pname1	L3	Pn3	UP1	Q3
P3	1	Pname3	L1	Pn1	UP2	Q4Q
P3	2	Pname3	L5	Pn5	UP2	5
…	…	…	…	…	…	…

Figure 4: Sample of Part Table Using Unknown Number of Attribute Name