Data Warehouse Snowflake Design and Performance Considerations in Business Analytics

Jiangping Wang and Janet L. Kourik
Walker School of Business and Technology, Webster University, St. Louis, Missouri, USA
Email: {wang, kourikjl}@webster.edu

***Abstract*—Snowflake is a data warehouse schema design where dimension tables are normalized on top of a star schema design. Snowflake schema is generally not recommended due to its performance overhead in joining the normalized dimension tables. However, the Snowflake schema can be extended in a way to improve performance for business analysis activities. In business analytics paradigm, two distinct environments are complementary and work together to provide effective business analytics. Firstly, the data warehouse environment transforms operational data into information. Secondly, the analytical environment delivers information to end users for further data analysis and decision making. The snowflake schema bridges the gap between the two environments. Snowflake schema facilitates the mapping of wide dimension structures with many dimension attributes to analytical processing hierarchies. The snowflake schema makes navigation along hierarchies easier and supports flexible analysis such as drilldown and rollup. This paper examines the two complementary business intelligence environments, roles played by the snowflake design in mapping from data warehouse to analytics, and performance considerations in snowflake design with case studies.**

***Index Terms*—data warehouse, snowflake design, business intelligence, business analytics**

I. Introduction

Business intelligence (BI) is a paradigm where enterprises integrate their operational data to make decisions based on data analytics. The goal of business intelligence is to present information to the end user in a way that supports decision making [1]. Decisions based on data can greatly enhance enterprise knowledge management and customer relationship management. Analytics and business intelligence capabilities have become competitive differentiators for many enterprises [2] and [3].

Before data can be used effectively for BI query processing and presentation, operational data must be transformed into decision support data. To prepare for the data warehouse environment transaction data must be extracted and integrated from multiple sources. Further, the operational data must be transformed into information so that necessary business understanding can be achieved and knowledge can be extracted from the data [4].

Data warehouses are fundamental to the business intelligence environment, encompass data from the entire enterprise, and focus on enterprise-wide business processes. The need for more intensive and complex analytics is the motivation for another environment to support online analytical processing (OLAP). OLAP makes the contents of the data warehousing environment available to users in an optimized structure, including strategic information, for decision making. OLAP supports multidimensional data analysis that focuses on analytical process and the relationship among business subject areas. OLAP environment maps data elements from the data warehouse to its own data structure in order to deliver information in an advanced, yet easy to use, interface.

The design of BI systems, that are important in facilitating the decision-making process, differs significantly from the customary operational or transaction database. Transaction databases are designed for running day-to-day business activities and supporting the operational needs of staff; therefore data manipulation is crucial to the design of transaction systems. In transaction systems, data are stored in detail and require users to execute queries to summarize or aggregate data for reporting. Accuracy is essential in order to manage data at the granularity of a logical business transaction and is achieved in part by normalizing the data storage mechanism often referred to as tables.

Normalization is a design process for transaction databases designed to minimize data redundancy and reduce data anomalies during data manipulation. Data anomalies occur when inserting, updating, or deleting data and may introduce unintended errors in the database. Such anomalies are a barrier to the data integrity (accuracy) required to serve employees on the front line of interaction with customers. Essentially, normalization is an abstract data design approach that increases data integrity in the database. Operational database design reflects data integrity concerns and tends to be highly normalized to maintain efficient daily transaction support.

Having summarized traits of operational databases and the context for BI and data warehousing, the next section will examine the two database environments used in BI.

Manuscript received on March 2, 20 15; revised on August 25, 2015.

doi: 10.12720/jait.6.4.212-216

The third section looks at the star schema design used to build many data warehouses. The fourth section depicts the snowflake schema as an alternative and its role in BI environments. The fifth section examines the benefits in terms of performance as well as the drawbacks of the snowflake design. The last section summarizes the findings.

II. Distinctive Environments in Business Intelligence

Business intelligence is a comprehensive and integrated environment to capture, integrate, and analyze data with the purpose of generating and presenting information to create intelligence about a business. To support business decision making, business intelligence in general encompasses two distinctive environments, namely data warehousing and analytical. Both environments are needed to provide a comprehensive solution in tools and techniques. Data warehousing environment transforms data to information and analytical environment transforms information to knowledge. Data warehouse provides an integrated, subject-oriented, non-volatile, and time variant environment to collect and store data. To put data into data warehouse, data from operational databases are integrated, cleansed and transformed into usable information that is ready to support complex data representation.

To provide information for decision making, data warehouse has to be designed in subject oriented schema structure, where data is collected and organized by business areas to better answer business questions. Therefore, data warehouse is generally implement under star schema that consists of a fact table representing business measures and surrounding dimension tables representing business subject areas.

With tremendous complexities and performance demands in business intelligence, it is almost impossible for top management to retrieve data warehousing data and answer business questions without a sophisticated analytical environment. The analytical environment provides users advanced tools to access information content and transform information into knowledge. It concentrates on transforming information from data warehouse into knowledge for decision makers to reach timely and accurate strategic decision in their business. As part of analytical environment, OLAP plays key roles in managing data, transforming data, analyzing data, and presenting data to end user [5]. The two environments work together complementarily in business intelligence to empower users to make sound business decisions based on the accumulated knowledge of the business as reflected on historic operational data, as depicted in Fig. 1.

Each of the two environments has distinct characteristics and plays roles in the process of business intelligence. The two environments have to work together coherently. Data warehouse design has to take into consideration how the data in data warehouse will be processed and analyzed by end users. In turn, OLAP hierarchies for drilling down and rolling up have to map to underlying data warehouse dimension hierarchies at various granularity levels. Therefore, data warehouse design directly affects how the data will be used in the analytical environment.

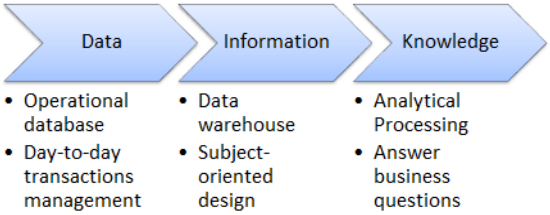

Figure 1. Data, information, and knowledge.

III. Star Schema in Data Warehouse Design

The star schema is used in data warehouse design because the existing normalized operational database does not yield a structure that serves advance data analysis requirements well. The star schema is a dimensional modeling technique for data warehouse where critical business measures, such as sales and revenue, are captured and examined from perspectives of multiple dimensions. Star schema focuses on dimensions and facts, mapping business subject areas with business measures. Star schema benefits include simple implementation and an intuitive process for user.

Dimensions provide views by subject areas, such as customer, region, and product. A dimensional table is generally flat and wide, consisting of many attributes that describe the dimension. Since table joins are computationally expensive, it is generally recommended to denormalize dimension tables to reduce possible joins between tables so performance of queries can be satisfied. Denormalized dimension tables connect to the fact table by foreign keys, which is intuitive to business users simplifying business measures analysis. For example, as depicted in the Fig. 2, the business measures are units, sales, and cost, which can be analyzed by aggregating in subject areas such as sales channel, customer, and product.

One of the features of star schema is its wide dimension table. A dimension is wide and flat with many textual descriptive attributes so measures can be fully described during analysis. For instance, the TIME dimension, in addition to the key for months, MONTH_ID, may contain many other attributes. If the table is laid out with columns and rows, the table is extended horizontally.

Another feature of a wide dimension in star schema design is its multiple hierarchies. Dimension tables often consist of multiple hierarchies so that analysis along any hierarchy can be performed. In the example of above TIME dimension, there exist hierarchies such as calendar year and fiscal year. Analysis can be performed by drilling down along the calendar year hierarchy from all years, to calendar year, calendar quarter, and month. Similarly, it can also be performed along the fiscal year hierarchy from all year, to fiscal year, fiscal quarter, and month.

With their flat and wide structure, dimension tables in star schema are not normalized. For optimized query performance, this denormalized design allows attributes in dimensions to participate in queries by relating directly to fact table without incurring any extra overhead of joining normalized tables. However, for many database professionals, this denormalized nature may cause problems in not only wasted storage space but also table updating and maintenance. For example, dimensional tables will be subject to changes under situations of slowly changing dimensions (SCD). If the attribute for the update in the dimension table is duplicated across multiple records, update anomalies may occur.

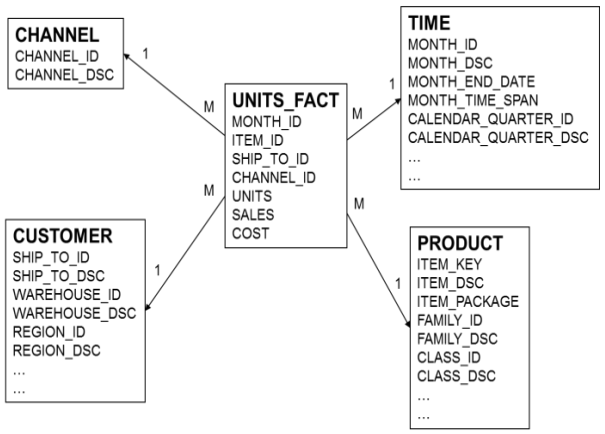

Figure 2. Sample star schema.

IV. Snowflake Schema Design

Snowflake schema is a variation of star schema in data warehouse design. Typical snowflake schema can be achieved by normalizing a dimensional table to reach semantic simplicity. In snowflake schema, each hierarchical level is stored in a separate dimension table. Since levels of dimension hierarchy relate closely to path of analysis, snowflake design facilitates data filtering operations along the dimension hierarchies and simplifies user navigation through dimension tables. With original star schema fact table in the center, if all hierarchies in all dimension tables are normalized, the design resembles the intricate arrangements of a snowflake.

There typically exist many potential hierarchies in a wide dimension for a given business subject area. However, not all hierarchies need be normalized in order to maintain a simple and easy to understand design. To better support decision making and map objects in the OLAP environment from data warehousing environment, the snowflake schema needs be designed keeping in mind the hierarchies that will participate in analytical processes. If a potential hierarchy will not be required when mapping to OLAP layer, it should be ignored in the normalization process to minimize design complexity and performance overhead. This approach balances between a "pure" star schema and a "pure" snowflake schema and produces an optimal design. A sample snowflake design is shown in Fig. 3, where TIME dimension is normalized along two hierarchies, calendar year hierarchy and fiscal year hierarchy, since analysis will be performed heavily along these two paths.

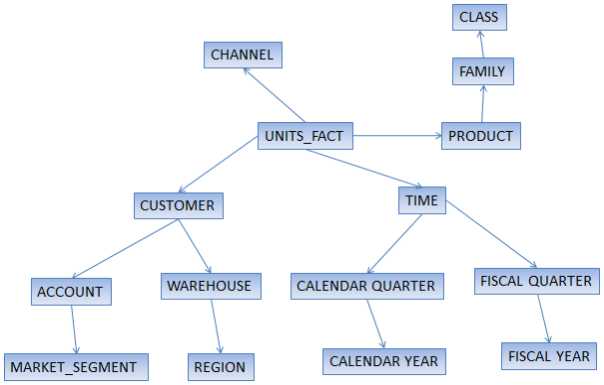

Figure 3. Snowflake schema.

OLAP hierarchies are mapped to data warehouse design. Fig. 4 lists typical mappings between OLAP objects to data warehouse objects. The OLAP cube maps to data warehouse fact table or a view that is based on a fact table. Measures from fact table directly map to cube measures. OLAP dimensions and hierarchies map to sub-dimensions in data warehouse. For instance, on the data warehouse side, there is only one flattened out dimension table TIME that has embedded hierarchies, such as calendar year and fiscal year. On the OLAP side, the mapping based on the snowflake schema will generate multiple hierarchical views. This is beneficial to the mapping process and easy in maintenance. Obviously, with the snowflake design where all the sub-dimensions are in their own normalized tables, the mapping process will be more simplified compared to the mapping performed in the original star schema.

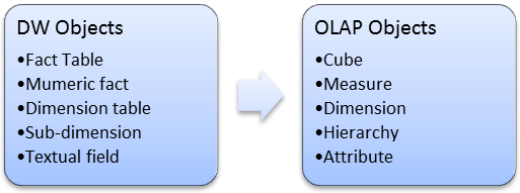

Figure 4. OLAP to data warehouse mappings.

Snowflake schema bridges data warehousing environment to analytical environment by mapping hierarchies and multi-dimensions easily. Snowflake schema reflects how users view the data in their organizations. The normalized multiple dimension tables represent levels in the dimensional hierarchy. It is intuitive to understand since it matches business subject areas and the relationship among them. With snowflake design, existing alignment between data processes such as drilldown and rollup, and dimension hierarchies makes transformation from data to information and from information to knowledge easier, and consequently enhances business decision making.

V. Performance Considerations

If data warehouses were used directly for data analysis under either snowflake or star schema design, aggregate

calculations would be done on the fly at any level above the base level in each dimension taking a significant amount of time in query processing. OLAP system enables quick and easy information retrieval by mapping data to underlying data warehouse. OLAP environment focuses on cubes that aggregate business measures for each unique combination of dimension hierarchies. In viewing data, analysts use dimension hierarchies to recognize trends at one level, drill down to lower levels to identify reasons for these trends, and roll up to higher level to see what effect these trends have on a larger sector of the business.

The snowflake schema is really useful when a data warehouse maintains multiple fact tables representing different aggregation levels. The aggregate fact tables are pre-calculated and work with sub-dimension in hierarchies to speed up query operations. For example, the TIME dimension can be snowflaked to Year-Quarter-Month-Day. To speed up query operation for aggregated values for roll-up operations, multiple fact tables related to each level (year, quarter, and month) of aggregation in the location dimension can be created. Each fact table matches a level along the hierarchical structure within the dimension, as demonstrated in Fig. 5. The aggregate tables are pre-computed at the data-loading phase rather than at run time. The purpose of this technique is to save processor cycles at run time, thereby speeding up data analysis.

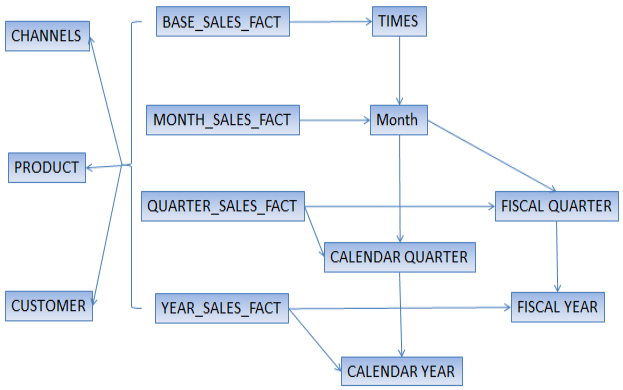

Figure 5. Multiple fact tables.

With this design, aggregate tables contain all of the aggregate data so a query against them just selects the aggregated data, instead of performing calculations to generate the value. Queries select the aggregates directly from the fact table by applying the appropriate filter to each dimension, which will significantly improve the performance, as the sample queries and results compared in Fig. 6 and Fig. 7.

Fig. 6 Sample Query 1 shows a query with aggregate function SUM() and GROUP BY clause that is executed against data of close to 300,000 records in a sample fact table where calculation is performed on the fly and result is retrieved in 0.10 seconds. In contrast, Fig. 7 Sample Query 2 shows another query that fetches the exact same results from aggregate table using filters, instead of using any aggregate function, which retrieves the same result in 0.01 second – 10 times faster.

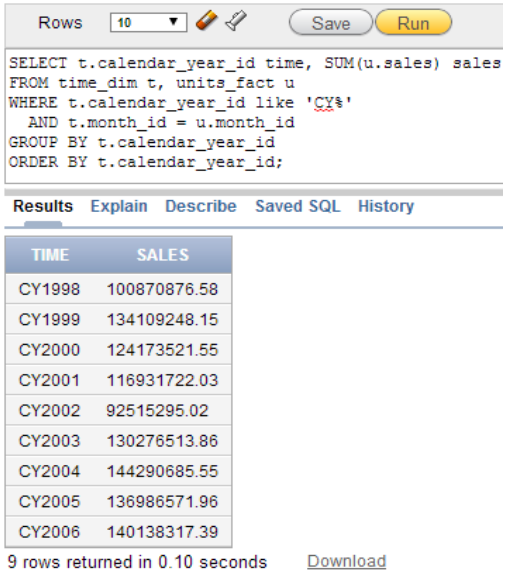

Figure 6. Query 1 with aggregate function returned in 0.10 seconds.

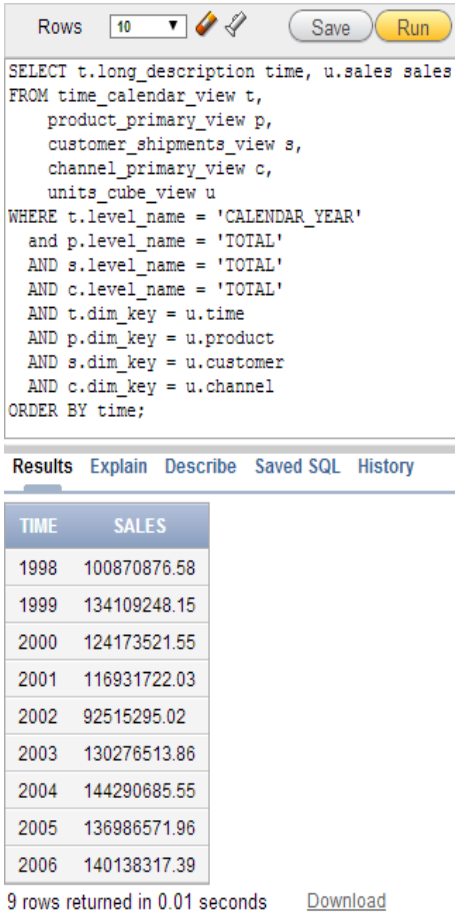

Figure 7. Query 2 without aggregate function returned in 0.01 seconds.

The normalization process may bring some other benefits into the picture. Dimension tables in the data warehouse are usually very large containing multiple sets of attributes at different granularities. For example demographic or geographic information in the customer dimension table may be separated as sub-dimensions. By implementing snowflake design, normalization is performed on the large and very wide dimension tables,

which makes navigation along hierarchies easier. It may also help optimize complex queries by implementing a heuristic-based query rewriting technique. With the snowflake design, structures in a data warehouse are easier to update and maintain. Normalization reduces data redundancies and, in turn, reduces data anomalies. With a large number of attributes in a dimension table, it is possible that a set of related attributes are updated less frequently than others. Having multiple dimension tables for sub-dimensions allows for queries to work with fewer records. The chances of data anomalies are greatly reduced.

Normalization allows long text fields in dimension tables to be eliminated. Normalization reduces storage space requirements on holding many attributes of dimension tables, especially those involving long text fields that are repeated. If the data is sparse, where a large number of attributes are empty for each dimension record, those attributes that are rarely populated could be in their own sub-dimension table. The savings in space could be generous in many cases. However, since query performance in data analysis is commonly more important than storage efficiency, a fully normalized snowflake structure may not be the best approach. In many cases it may be appropriate to normalize certain dimensions that are directly involved in analysis processes and create partial snowflake structures in order to achieve significant storage savings at the price of an insignificant decrease in query efficiency.

VI. Conclusions

Data analytics in business intelligence requires robust data warehouse design to support flexible querying across multiple complex dimension relationships. Snowflake schema is a method of normalizing the dimension tables in a star schema and creating sub-dimensions for hierarchical levels. The snowflake schema is suitable for mapping flattened out dimension structure to OLAP hierarchies. It bridges the gap between the data warehouse environment and data analytics environment in business intelligence and facilitates the mapping between the two. Snowflake schema makes navigation along hierarchies easier and analysis such as drilldown and rollup possible. It works well with multiple aggregate fact tables where performance of aggregation analysis will be greatly enhanced. Lastly the snowflake schema saves processor cycles at runtime with appropriate filters on dimensions applied on aggregates directly in the fact table. With the snowflake design, structures in a data warehouse are easier to update and maintain. Effective business decision making requires better information delivery. The snowflake schema in data warehouse design plays important roles in supporting business analytics.

References

[1] Cody, W. F. Kreulen, and J. T. Krishna, V. Spangler, and W. S., "The integration of business intelligence and knowledge management," *IBM Systems Journal*, vol. 41, no. 4, pp. 697-713, 2002

[2] M. J. Liberatore and W. Luo, "The analytics movement: Implications for operations research," *Interfaces*, vol. 40, no. 4, pp. 313-324, 2010

[3] A. McAfee and E. Brynjolfsson, "Big data: The management revolution," *Harvard Business Review*, vol. 90, no. 12, pp. 60-68, October, 2012

[4] A. Sen and A. P. Sinha, "A comparison of data warehouse development methodologies," *Communications of the Association of Computing Machinery (ACM)*, vol. 48, no. 3, pp. 79-84, 2005.

[5] J. Wang, J. L. Kourik, and P. E. Maher, "Identifying characteristics and roles of OLAP in business decision support systems," *Journal of Business and Educational Leadership*, vol. 3, no. 1, Fall, pp. 90-99, 2011

Jiangping Wang is an associate professor of computer science at Webster University. He has a B.A. from Chongqing University, China, an M.S. from the University of Leeds, United Kingdom and a Ph.D. from the Missouri University of Science and Technology, Rolla, Missouri, USA. Dr. Wang's areas of teaching include database design, database applications, data warehousing, web databases, database in web services, and distributed application development. His areas of research include database management systems, decision support systems, business intelligence, e-commerce data processing, and software project management.

Janet L. Kourik is a Professor in the Mathematics and Computer Science Department of Webster University in St. Louis, Missouri, US. She has a B.S.C.S. from Webster University, an M.A. from Webster University and a Ph.D. from Nova Southeastern University. Dr. Kourik's areas of teaching include database concepts and applications, information systems, operating systems, and distributed systems. Her areas of research include databases and analytics, agile methods, informatics, and computer science education.