

# Semi-Star Modeling Schema for Managing Data Warehouse Consistency

Pasha, M. A., Nasir, J. A., and Shahzad, M. K.

[marpasha@yahoo.com](mailto:marpasha@yahoo.com), [jamal4882002@yahoo.com](mailto:jamal4882002@yahoo.com), and [khurram\\_pu@yahoo.com](mailto:khurram_pu@yahoo.com)

Punjab University College of Information Technology, University of the Punjab, Lahore, Pakistan.

**Abstract:** Decision-making has always been a complicated process due to reason of ever increasing number of alternatives; the relationships between the variables involved are complex, and frequent changes. The supporting systems to decision-making include data warehouse and data mining.

Traditional DW systems have static structure of their schemas and relationships between data, and therefore they are not able to support any dynamics in their source structure and contents, resulting in inconsistent results. A novel modeling technique is proposed to support such dynamics.

**Keywords:** Data Warehouse, Multidimensional Structures, Content Changes, Schema Changes.

## 1. INTRODUCTION

For making decision-making more effective, Data warehouse (DW) technology transform data from various transactional sources into subject-oriented data. This some times introduces data inconsistencies issues due to DW schema [1]. Researchers of the domain are trying to devise reliable strategies for transferring consistent information from OLTP's and external sources to data warehouse, which can be used for decision-making. OLTPs are usually independent of each other in their architecture and do not dependent on data warehouse, whereas data warehouse with static schema is dependent on transactional sources (TS). So Changes to TS affect DW. These changes can be of two types.

1. Content changes: insertion/updation/ deletion of records occurred as a result of DML commands on database.
2. Schema Changes: addition/modification/ dropping of attribute occurred as a result of DDL commands on database.

Due to static schema of DW, both types of changes may result in providing inconsistent analytical results from DW. According to conventional approach, if new attributes, which are going to be used for analytical purpose, are added to the TS, then such attributes must be added to the dimensional schema (DS) of DW for providing consistent information. In some cases changes may require to maintain either new version of data warehouse or new schema for DW. Most of the approaches proposed for the maintenance of DW have focused on providing transactional incremental DW refreshing under content changes of TS. These approaches could not be said efficient, as many dynamic business organizations may not afford regular up-gradation and

maintenance of DS, transformation package (TP) and multidimensional structures called cubes due to high maintenance cost.

This paper presents a novel approach, called Semi-Star Modeling (SSM), for maintaining a DW under schema and content changes of transactional sources. SSM has the ability not only to meet the transactional and analytical needs of business but also accommodate the content and schema changes by decreasing the number of up-gradations. SSM offers many benefits including decrease in transformation cost, time, and maintenance of single database for analytical and transactional purpose and avoidance of regular schema up-gradations. For prototyping, SSM is implemented and tested using SQL Sever and its Analysis Services. For transferring data to fact table DTS package of SQL Server 2000 is used. Various alternative business scenarios have also been simulated for testing the flexibility of SSM.

The rest of paper is organizes as follows: Section 3 discusses an example illustrating the potential problem is discussed in section 4. Section 5 discusses the related works for solving problem; SSM is devised in section 6, section 7 contains the SSM for example, its analytical results are evaluated in section 8, section 9 has the advantages of SSM.

## 2. Illustrative Example

Let DW stores data about a company's sales. The company has sale points in multiple cities and the sales are inspected in various locations at certain time. Cities are grouped into administrative regions and products sold are grouped into categories. The DS of company's data warehouse is given in Fig.1; composed of two types of tables: Fact tables (FT) and Dimension Tables (DT), which surrounds the fact table. In the Fig.1 **Sales\_Fact** is the only fact table, surrounded by **Location**, **Time** and **Product** dimensions. The location dimension is composed

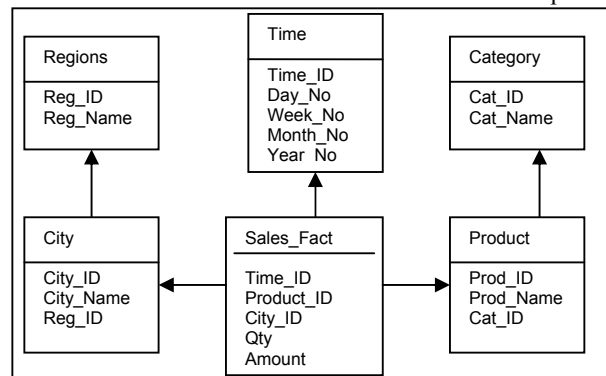


Fig. 1 An example schema of DW

of two levels: **City** and **Region**. The product dimension also has two levels: **Category** and **Product**. **Sales\_fact** table stores the information about quantity of product sold in **Qty** attribute and the total amount against the sold product in **Amt** attribute. Let us further assume that the tables store the following data.

Here are the values inserted to the tables along with the queries executed to retrieve the results.

Select \* from **product**;

Prod_ID	Prod_Name	Cat_ID
1	P1	Category 1
2	P2	Category 1
3	P3	Category 2

Select \* from **category**;

Cat_ID	Cat_Name
1	Category 1
2	Category 2
3	Category 3

Select \* from **region**;

Region_ID	Region_Name
1	Region A
2	Region B

Select \* from **city**;

City_ID	City_Name	Region_ID
1	BWP	1
2	ISL	2
3	KHI	1

Select \* from **Sales\_Fact**;

Time_ID	Prod_ID	City_ID	Qty	Amt
1	1	1	65	650
1	2	1	90	900
1	3	2	32	3200
1	1	3	20	2000

For this schema multiple analytical queries can be written, some of them on content and schema changes give consistent results while other does not.

This DW describes real world that is likely to be changed with schema and content changes. This example illustrates problems related to the retrieval of inconsistent analytical results due to changes in TS. In this example, changing the borders of regions may result in moving cities from one region to another. Such a change may have an impact on the analytical results received from DW. Assume that boundaries of region are changed in such a way that city "KHI" is moved from "Region A" to "Region B".

Let us assume a query-computing amount earned in each city before changes to the boundaries of region and its result:

City_Name	Sum (Amount)
BWP	1550
ISL	3200
KHI	2000

After moving 'KHI' from "Region A" to "Region B" results of the same query are not changed.

City_Name	Sum (Amount)
BWP	1550
ISL	3200
KHI	2000

In this query changes to contents has caused no effect on analytical results. Taking another case, the total amount earned in every region before changing the boundaries of region is:

Region_Name	Sum (Amount)
Region A	3550
Region B	3200

And the total amount earned in every region after changing the boundary of region is:

Region_Name	Sum (Amount)
Region A	1550
Region B	5200

Although no change to the database is being made except that region boundary, but inconsistent results are generated. Similar inconsistent results are generated if region name is changed, city name is changed, product category is changed, product name is changed, and name of product category is changed. Also, when the schema is upgraded this may not give desired results. For example, if an attribute to dimension is added then to reflect such changes to the end users DS, TP and multidimensional structures are required to be upgraded or multiple version of DS, TP and multidimensional structures should be maintained which cause high maintenance cost. Medium organizations with dynamic business could not afford regular up gradation and maintenance of DS, TP and multidimensional structures. So a new schema is required, which could handle the above discussed problems with minimum schema up-gradations.

## 5. EXISTING APPROACHES

The conventional approaches for the management of changes in a DW can be classified into two categories: i) Schema evolution ii) Versions extension. The former is proposed in [7] and supports only one data warehouse schema and instance. When change is applied to a schema, it is required to upgrade DS, TP and multidimensional structures. This solution is not suited for medium organizations with dynamic business where these changes are regular. As, whenever the change is applied to DW schema all data described must be converted to new schema, incurring high maintenance cost.

The later approach, propose by various authors, maintains multiple versions of DW like temporal versioning that is time stamped and implicit versioning [2]. The inabilities of temporal versioning for expressing and processing queries that compare several temporal versions of data is discussed in [5,9]. The implicit versioning is proposed in [3,6]. In this approach, versions are used for avoiding conflicts and mutual locking between OLAP queries and transactions refreshing a data warehouse. As versions are implicit created and managed by the system, these mechanisms cannot be used for analytical purpose. Permanent user defined versions of views in order to simulate changes in a data warehouse schema has also been proposed [8]. However, the approach supports only simple changes in source tables and it does not deal either with typical multi-dimensional schema or evolution of facts or dimensions. Similarly, management of multiple versions, query language capable of processing data from multi-version DW, development of mechanism of indexing on multi-version data and efficiency of processing queries addressing several DW versions needed to be addressed.

## 6. SEMI-STAR MODELING

The proposed technique, "Semi-Star Modeling", developers a new kind of schema that computes new values of facts based on original data for managing changes in TS. SSM contains two types of tables, i) Transaction tables used to handle daily transactions ii) Fact tables, with intensive data, used for analytical purpose. Both types of tables exist in single database but in different table-space will cause lesser effect on database efficiency.

Selected transactional tables also act as dimension tables for analytical purpose (as shown in Fig. 2). These dimension tables in SSM are called Shared Dimension Tables (SDT). Due to the presence of these tables SSM has ability to accommodate small schema changes with out schema up-gradation. As discussed above, if an attribute to dimension is added which is to be used for analysis it is required to upgrade DS as well as TP and multidimensional structures to reflect the changes for end users. But, SSM reduces such up-gradations.

To accommodate the content changes SSM divides the dimensions into two categories: i) Static Behavior Dimensions (SBDs) ii) Dynamic Behavior Dimensions (DBDs). In SBD the lower level dimension table is fixed with upper level dimension table, while in DBD lower level may change over time. In SSM, **Location** attribute of the above discussed example is being considered as DBD to accommodate content changes. As a result the region changes of city will not give inconsistent results.

In SSM, the SBD their primary keys identify levels, but for DBD, levels are identified by Surrogate Keys. Else the DBD's level cannot exist. In later case it is recommended to consider each level as different dimension.

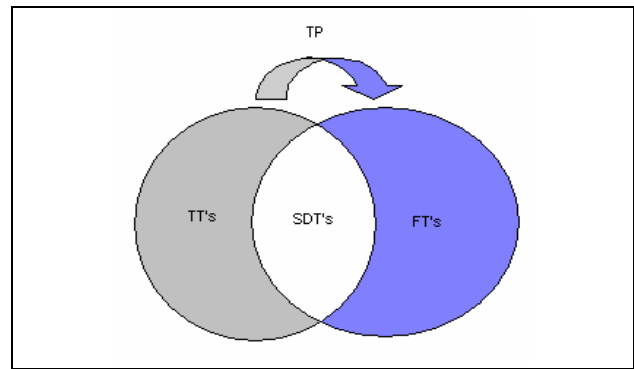


Fig. 2 SSM Transformation

With these properties SSM decreases the number of schema up-gradations in the presence of content and schema changes to the transactional system. Also the TP up-gradation is not required in the presence of small changes, but it cannot be avoided for changes that affect the facts directly.

## 7. Illustrative Example of SSM

We have already discussed the problem related to company's sale xample. This section illustrates how SSM overcome these problems. In this example, we consider the dimension of location (having tables **City** and **Region** at level 1 & 2) to be DBD and product dimension (with tables **Product** and **Category** at level 1 & 2) to be SBD. i.e. Product is fixed with category meaning category of product will never change. So following is prototype of SSM designed for the company.

Below are the analytical results of queries asked in previous examples. "Total amount" earned per region before changing the boundaries of region.

Region_Name	Sum (Amount)
Region A	3550
Region B	3200

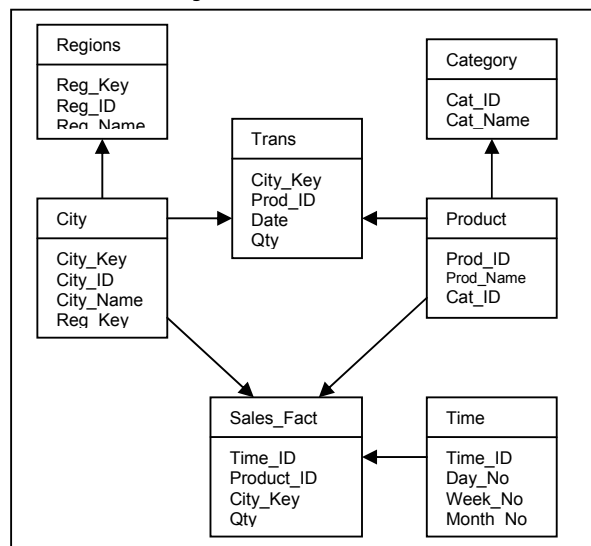


Fig. 3 Prototype of SSM for Example

“Total amount” earned per region after boundaries of region is changed. (As a result of which “KHI” is moved from Region A to Region B).

Region_Name	Sum (Amount)
Region A	3550
Region B	3200

No inconsistent results are achieved in the presence of any changes to the contents of DBDs. 1.) Name of region is changed 2.) City Name is changed. If product dimension is also made DBD following changes will also have no effect on analytical results. 1.) Product category is changed 2.) Product name is changed 3.) Name of product category is changed.

### 9. Advantage of SSM

SSM has the abilities of not only solving the TS changing problems but also its advantages includes the reduction of transformation time, no DS up-gradations required due to shared dimensions, reduction of maintenance cost, potential of accommodating schema and content changes by decreasing system up-gradations. Some of the main advantages are given below:

- Lesser Transformation Time:* In DS data from source is transformed to each DT. If multiple dimensional tables exist then total transformation time is equal to the some of all the dimensional & fact table transformations time. But with SSM, only a few dimension tables are required to be populated (as dimension tables are shared, they are already populated by the transactions) which will decrease the total transformation cost.
- Way-out to DS up-gradations:* For each problem creating update to be reflecting for the end users, it is required to upgrade DS, TP and multidimensional structures. While in SSM number the no of up-gradations to the schema is decreased, as SSM is flexible enough to accommodate such changes.
- DB utilization:* After each transformation it is required to refresh multidimensional structures called cubes. (Cubes also store data). When SSM is implemented the time saved by the TP will be used to refresh cubes, which will then be available for the end users.
- Content Changes are Accommodated:* As proved from the example SSM has the ability to accommodate the changes to the contents, which is not the case with DS.
- Schema Changes are Accommodated:* If an attribute is added to the transactional purpose which is to be used for analytical purpose. This also results in up-gradation of the DS, TP and the multidimensional structure. But if SSM is implemented it is not the case, but it just requires multi-dimensional structure up-gradation, which is not a big task as one cube can be called in another and new dimension can be added.

Resulting in reduction of no of schema up-grades.

- No Version Upgrading Cost:* In multi-version DW at one time one version can exist. If such a changes are made to the TS that it requires schema up-gradations, new DW version is transferred to the new version, increasing high maintenance cost. SSM reduces such up-gradations and maintenance cost.

### 10. SUMMARY & CONCLUSION

Commercial DW systems existing in market have static structures of their schemas and relationships between data. Such schema cannot support the dynamics of medium size business organizations. To accommodate schema and content changes Semi Star Modeling technique is devised, which support such dynamics in the presence of minimum DS and TP up-gradations. Current works focuses on the top down approach whereas bottom-up approach is needed to be explored yet. Similarly, retrieval of analytical results from multiple SSMs, conceptual Modeling of SSM, data management techniques, efficiency of processing and queries addressing several SSMs are some future research directions.

### REFERENCES

- [1] Bebel, Eder. *Creation and Management of Version in Multi-version Data Warehouse*, Proceedings of ACM Symposium on Applied Computing 2004
- [2] Eder. *Changes of Dimension Data in Temporal Data Warehouse*, Proceedings of the Dawak 2001
- [3] Kang. *Exploiting Versions for On-Line Data Warehouse Maintenance in MOLAP Servers*, Proc. of VLDB Conference, China 2002
- [4] Miquel. *A Multi-dimensional & multi-version Structure for OLAP Applications*, Proceedings of DOLAP Conference, USA 2002
- [5] Stock. *Temporal Structures in Data Warehouse*, Proceedings of Data Warehousing and Knowledge Discovery DaWalk, Italy 1999
- [6] Mohaina. *Concurrent Maintenance of Views Using Multiple Versions*, Proceedings of International DB Engineering & Application Symposium, 1999
- [7] Hurtado. *Maintaining Data Cubes under Dimension Updates*, Proceedings of ICDE Conference Australia, 1999
- [8] Bellahsene, *View adaptation in Data Warehousing Systems*, Proceedings of DEXA Conference, 1998
- [9] Mendelzon, *Temporal Queries in OLAP*, Proceedings of VLDB Conference, Egypt, 2000