Comprehensive Study for Data warehouse Schema evolution Operators

Meenakshi Arora
University School of Information Technology,
Guru Gobind Singh Indraprastha University
Delhi, India
meenakshiarora87@gmail.com

Anjana Gosain
University School of Information Technology,
Guru Gobind Singh Indraprastha University
Delhi, India
anjana_gosain@hotmail.com,

***Abstract*— Data warehouse is considered as the core component of the modern decision support systems. Due to the major support of data warehouse in the daily transaction of an enterprise, the requirements for the design and the implementation of DW are dynamic and subjective. This dynamic nature of the data warehouse may reflect the evolution in the data warehouse. Data warehouse evolution may be focused on three approaches namely schema evolution, schema versioning and view maintenance. Evolution of the data warehouse may often change their data and structure (schema changes). These schema changes may be consider according to the change in structure, software and users' requirement. Schema evolution in data warehouse consists of various level namely structural level, conceptual level and behavioural level. This paper mainly focuses on schema evolution and proposes the operators to handle the creation and evolution of aggregated fact table. Our work is to do comparative study for various approaches of schema evolution**.

***Keywords*— Data Warehouse Evolution, Schema Evolution, Schema Operators, Aggreagte operator**

I. INTRODUCTION

A data warehouse is gathering of various production data, external data, archived data and internal data from different data sources. These sources are inculcated in the data warehouse and may change their schema according to the user requirements. Such changes must be supported when they populate the data warehouse. Evolution in data warehouse may be generated by change in schema, changes in software and the change in data warehouse requirement. Data warehouse evolution may be classified into three different approaches namely schema evolution, schema versioning and view maintenance [7]. Schema evolution of data warehousing consist of various levels updates that is dimension updates, structural updates, instances updates, facts updates and attributes updates. Dimension updates reflect static aspect of data warehouse evolution and structural updates reflects dynamic aspect. Schema evolution may be managed by two different approaches namely adaptational approach and versioning approach. In [8] adaptational approach existing instances have to be adapted to the new schema and the application programs that run over the database before the changes, also have to be updated. In versioning approach, new version is created over previous version and no modification is applied directly on the existing schema. Different authors have proposed different evolution operators corresponding to architecture components and quality factors they affect.

In this paper, we do a comparative study of various approaches for data warehouse schema evolution and propose operators to handle evolution of aggregated fact table.

II. LITERATURE SURVEY

In the literature, different authors [1, 2, 3, 4, 5, 6, 9, 10, 13, 14, 15] have proposed different operators to handle schema evolution at different levels. The schema evolution approach focuses on dimension updates [1, 6], instances updates [2, 10, 15], facts updates, attribute updates [3]. In [4], author has discussed about the quality factors and there affects on evolution operator. In [13, 14], authors have discussed about schema change operator and dimension instance structure change operators. These sections discuss about theses operators in detail

A. Hurtado et. al. [1999 a]

In this paper [1], author focused on Multidimensional model which consist of dimensions tables, fact tables and data cubes. Along with that author proposed seven operators for schema evolution to classify dimension updates. Those operators are Generalize operator, Specialize operator, Relate level operator,

Unrelated level operator, Delete level operator, Add instance Operator, Delete instance operator.

B. Hurtado et. al. [1999 b]

In this paper [2], author proposes four complex instance update operators in addition to instance update operators defined in [1]. The complete sets of instance update operators are add instance, delete instance, reclassify, split, merge and update instance.

C. Blaschka et.al. [1999]

Schema evolution in data warehouse plays an important role especially in decision support environment. In this paper [3], author defined a schema evolution related to algebra on formal description of multidimensional schemas and instances. It consists of 6 tuple (F, L, A, gran, class, attr). After defining the data model, authors presented a set of formal evolution operations. The following evolution operations have no effects on the model i.e. Insert level, Delete level, Insert attribute, Delete attribute, Insert classification relationship, Delete classification relationship. The evolution operations that have effects on the model are Connect attribute to dimension level, Disconnect attribute from dimension level, Connect attribute to fact, Disconnect attribute from fact, Insert / delete fact, Insert/delete dimension into fact.

In [16], author proposed FIESTA approach which focused on the evolution of multidimensional schema changes by using the ME/R model as a graphical conceptual model to represent multidimensional semantics. Thus, all modification may be done on the basis of conceptual level which is specified by the group of evolution operators.

D. C. Quix et.al. [1999]

In this paper [4], author focused on quality of the data warehouse under evolution. Here, author presented many evolution operations and there affects on quality factor. The evolution operators for base relations and views, and relate them to quality factors which are affected by these evolution operators are summarized as Add base relation/ view, Delete base relation, Add attribute to base relation/view, Delete attribute from base relation/view, Rename Relation, View or Attribute, Change of attribute domain, Add Integrity Constraint, Delete Integrity Constraint, Change to view definition.

E. Alejandro A. Vaisman et al. [2002]

In this paper [5], authors proposed an extension to the work presented in [1] [2] about dimension updates operators and view maintenance. Here, author briefly evaluate the set of operators that modify either the schema or an instance of a given dimension and proposed the visualization tool for dimension and data cubes. The complete set of update operators are described in Structural operator are generalize operator, specialize operator, relate level operator, unrelated level operator, delete level operator and in Instance operator it consist of add instance operator, delete instance operator.

F. E. Benitez. Guerrero et. al. [2004]

In this paper [6], authors propose WHES (Warehouse Evolution System) prototype that describes creation and evolution of data warehouses to support dimension and cubes update. Here, authors have proposed 16 operators to modify multidimensional schemas which are create dimension, drop dimension, rename dimension, rename level, add level, delete level, add property, delete property, create cube, drop cube, rename cube, rename measure, add axis, delete axis, add measure, delete measure.

G. C.E. Kaas et.al. [2004]

In this paper [9], authors examine the evolution properties of star and snowflake schemas. Here, authors discussed about eight evolution operations. These operations are mainly focused on dimension changes, level changes, measure attribute changes and dimension attributes changes. The complete sets of operators are Insert dimension into fact, delete dimension, Insert/ delete level, connect attribute to dimension level, disconnect attribute from dimension level, add/delete measure.

H. T.Morzy et.al. [2004]

In this paper [10], authors discussed about multiversion data warehouse which consist of elementary operations that modify a data warehouse schema. Schema change operations include: (1) creating a new level table with a given structure, (2) connecting a given level table with its sub- super level tables, (3) disconnecting a given level table from its dimension hierarchy, (4) removing a previously disconnected level from a schema, (5) adding a new attribute to a level, (6) dropping an attribute from a level, (7) changing a domain of a level attribute, (8) creating a new fact table, (9) adding a new attribute into a fact table, (10) associating a given fact table with a given dimension, (11) removing a non primary key or non foreign key attribute from a given fact table, (12) removing an association (foreign key) between a fact table and a dimension, (13) removing a fact table ,previously

disconnected from a schema, (14) renaming an attribute, (15) removing a table. These schema operations may cause problem due to absence of previous data or have to transform to a new structure. Secondly user logical queries need to be modified in order to be applicable to a data warehouse schema after change. To avoid these problem author suggests applying the operations to a new data warehouse version and, if accepted by a data warehouse administrator. When operations are successfully applies then a new version is created automatically.

I. Jarernsri L. Mitrpanont et al. [2006]

In this paper [13], authors present the technique enabling the creation of dimension schema and instance schema. Schema change operations affect to the structural change schema. The addition/ deletion of MDB schema give rise to the change of version. The schema change operations includes add/ delete dimension level, add/ delete dimension attribute, add dimension to fact, delete dimension from fact, add fact attribute to fact, delete fact attribute from fact.

The instance changes operations include add new data into an existing dimension, delete data of an existing dimension and update instance value of a dimension.

J. B. Bebel et al. [2006]

In this paper [14], authors discussed about MVDW Operators (Multiversion Data Warehouse). According to the author operators may divided in two groups: 1) Schema change operator, 2) Dimension instance structure change operators. Schema change operators consist of 15 operators. Those operators are as follows: a) Creating a new dimension, b) Creating a new level, c) Connecting level into a dimension hierarchy, d) Disconnecting a level from a dimension, e) Removing a dimension, f) Removing a level, g) Creating a new attribute for a level, h) Removing an attribute from a level, i) Changing the domain level attribute or fact attribute, j) Creating a new fact, k) Creating a new attribute for a fact, l) Removing an attribute from a fact table, m) Creating an association between a fact and a level, n) Removing an association between a fact and a level, o) Removing a fact.

Dimension instance structure change operators consists of five basic operators namely: Inserting a new level instance, Deleting a level instance, Reclassifying a level instance, Merging n instances of a level into a new instance, Splitting a level instance into n new instances. In this paper [14], authors in addition to this focused on Data warehouse version which may used for incorporating structural changes in external data sources as well as changes to a Data ware Schema resulting from changing user requirement.

K. George Papastefanatos et al. [2007]

In this paper [15], authors deal with the problem of performing what - if analysis changes that occur in schema / structure of the data warehouse sources. Here authors discussed the case study related to ETL process, extracted from an application of the Greek public sector. It consists of schema change operators namely renaming source table, renaming attributes of source table, adding / deleting attributes from source tables, modifying domain of attributes and changing the primary key of dimension table.

III. AGGREGATE FACT TABLE OPERATORS

Different authors in the schema evolution have proposed various evolution operators at different level for e.g. dimension updates, instances updates, fact updates and attribute updates. However none of these evolution operators have been defined for the aggregated fact table. Aggregates table are precompiled summaries of most granular fact table in a data warehouse that contain new metrics derived from one or more aggregate functions (AVERAGE, COUNT, MIN, MAX, etc.). These new metrics, called “aggregate facts” or “summary statistics” are stored and maintained in the data warehouse database in special fact tables at the grain of the aggregation.

We have proposed 5 evolution operators related to aggregate fact table to modify the schema of data warehouse i.e. create aggregate, delete aggregate, alter aggregate, rename aggregate, drop aggregate.

Aggregate operators for modification of schema include Create Aggregate, Delete aggregate, Alter Aggregate, Rename Aggregate and Drop Aggregate. Create operator creates a new aggregated fact table in the schema as per the users’ requirement. Delete operator removes the aggregate fact table from the schema. Alter operator change the schema by adding more attribute or modifying the aggregated fact table.

The create operator creates one way aggregate table in figure1 represented a derived aggregate fact table ‘SALES’ connected to a derived dimension table ‘CATEGORY

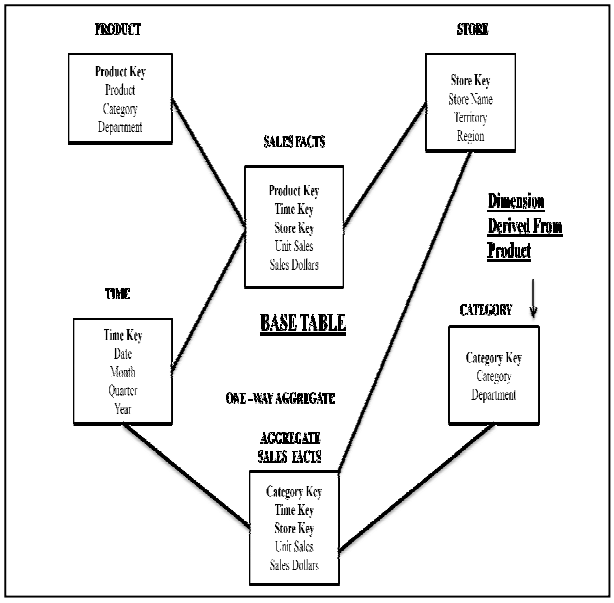

Fig: 1 Creation of Aggregate fact table and derived dimension table [11]

For defining data warehouse and data marts we examine SQL- based data mining query language called DMQL proposed in [12]. They may be defined as cube definition and dimension definition.

Creation for the aggregate fact table of Fig.1 may be defined in DMQL as follows:

```
define cube aggregate SALES FACTS
[Category, Time, Store]:

            Unit sales = Count (*),
Sales Dollars = sum (sales_in _
dollars)

define dimension TIME as (Time Key,
Date, Month, Quarter, Year)

define dimension STORE as (Store
Key, Store Name, Territory, Region)

define dimension CATEGORY as
(Category Key, Category,
Department)
```

The expression define cube define a data cube called aggregate Sales facts, which corresponds to central aggregate fact table of Figure 1. This command specifies the keys to the dimension tables, and the two measures, unit sales and sales dollars. The data cube has three dimensions namely time, store and category. A define dimension statement is used to define each of the dimensions. Similarly, other operator may be implemented.

IV. COMPARATIVE ANALYSIS

For the schema evolution approach, our comparative study is based on dimension changes, fact changes, instance changes, level changes, attribute changes, constraint changes and quality changes.

Authors works in [1, 2, 3, 5, 6, 9, 10, 13, 14] papers, supported the level evolution.

Author in [1, 2] was interested in structured evolution and instance evolution by introducing relate, unrelated level of structured evolution and add, delete instance of instance evolution.

Authors in [3, 6] were focused on dimension evolution and fact evolution. Benitez [6] presented add/ delete /rename of measure whereas Blashka [3] presented add and delete of fact. However, both authors in [3,6] presented add/ delete dimension and add/ delete attribute but only Benitez [6] proposed rename of dimension.

Author in [4] focused on quality of the data warehouse under evolution. Here, author presented many evolution operations and there affects on quality factor. Quix [4] was interested in attribute changes, constraint changes and quality changes.

Vaisman [5] proposed a visualization tool for dimensions and data cubes and also extend MDX, Microsoft's language for OLAP with a set of statements supporting dimension update operators. Author in [5] was interested in level evolution and instance evolution.

Kaas [9] examine the evolution properties of star schema and snowflakes schema. Authors [9] were interested in dimension evolution, level evolution, measure attribute evolution and dimension attributes evolution.

Authors in [10, 14] were focused on multiversion data warehouse operators which consist of elementary operations that modify a data warehouse schema and dimension instance structure. Morzy [10] presented level evolution, fact evolution, constraint evolution, attribute evolution whereas Bebel [14] presented dimension evolution, level evolution, instance evolution, fact evolution, attribute evolution.

J.L. Mitrpanont [13] presented the technique for enabling the creation of dimension schema and instance schema. Authors were interested in level evolution, attribute evolution, fact evolution.

G.Papastefanatos [15] discussed about what-if analysis changes in schema / structure of the data warehouse sources. Authors were focused on instance

evolution, attribute evolution and constraint evolution.

Comparative study for schema evolution operators proposed by different authors is given below in (Table 1):

Criteria / Author	Dimension changes	Fact changes	Instance changes	Level changes	Attribute changes	Constraint changes	Measure changes	Quality changes
Hurtado [1]				★				
Hurtado [2]			★	★				
Blaschka [3]	★	★		★	★			
Quix [4]					★	★		★
Vaisman [5]			★	★				
Benitez [6]	★	★		★	★		★	
Kaas [9]	★			★	★		★	
T.Morzy [10]		★		★	★	★		
Mitrpanont [13]		★		★	★			
Bebel [14]	★	★	★	★	★			
Papastefanatos [15]			★		★	★		

Table1. Comparative study of Schema Evolution

V. CONCLUSION AND FUTURE WORK

In this paper, we summarize the schema evolution of data warehouse and we have also proposed the operators to handle the creation and evolution of aggregated fact table. In the literature survey, different authors [1, 2, 3, 4, 5, 6, 9, 10, 13, 14, 15, 16] have proposed different operators to handle schema evolution at different levels i.e. structural level, conceptual level and behavioural level. Our comparative study is based on following criteria: dimension updates, instances updates, fact updates, level updates, attribute updates. Our future work includes implementation of these aggregate operators in different schema as per user requirements and along with that exploring other data warehouse evolution approaches, such as schema versioning and schema maintenance.

REFERENCES

[1] C.A. Hurtado, A.O. Mendelzon, A.A. Vaisman: *Maintaining Data Cubes under Dimension Updates.* Proceedings of the 15th International Conference on Data Engineering (ICDE), Sydney, Australia, March 1999.

[2] C.A. Hurtado, A.O. Mendelzon, A.A. Vaisman: Updating OLAP Dimensions. Proceedings of the 2nd International Workshop on Data Warehousing and OLAP, Kansas City, Missouri, USA, November 1999.

[3] M Blaschka, C Sapia, and G Hofling. On Schema Evolution in Multi-dimensional Databases. In 1st International Conference on Data Ware- housing and Knowledge discovery (DaWak 99), Florence, Italy, Volume1676 of LNCS, pages153-164, Springer, 1999.

[4] C. Quix. Repository Support for Data Warehouse Evolution. In Proc. of the Intl workshop DMDW, Heidelberg, Germany 1999.

[5] Vaisman A.A., Mendelzon A.O., Ruaro W., Cymerman S.G.: Supporting Dimension Updates in an OLAP Server. Proc. of the CAISE02 Conference,Canada, 2002

[6] E. Benitez- Guerrero, C.Collet, M. Adiba. THE WHES APPROACH TO DATA WAREHOUSE EVOLUTION. E-Gn osis[online], vol.2Art.2004

[7] W.Oueslati, J. Akaichi. A survey on Data warehouse evolution. International Journal of Database Management Systems (IJDMS), Vol.2, No.4, November 2010.

[8] Adriana Marotta. Data Warehouse Design and Maintenance through Schema Transformations. Master thesis , October 2000.

[9] Kaas C.E., Pedersen T.B., Rasmussen B.D.: Schema Evolution for Stars and Snowflakes. Proc. of the Intern. Conf. on Enterprise Information Systems(ICEIS 2004), Portugal, 2004

[10] Tadeusz Morzy, Robert Wrembel. On Querying Versions of Multiversion Data Warehouse. In Proc. Int. Workshop on Data Warehousing and OLAP, DOLAP'04, Washington (USA), 2004.

[11] Paulraj Ponniah. Data Warehousing Fundamental Guide. Published by John Wiley and Sons, 2001

[12] J. Han, Y. Fu, W. Wang, J. Chiang, W. Gong, K. Koperski, D. Li, Y. Lu, A. Rajan, N. Stefanovic, B. Xia, and O. R. Za _ane. DBMiner: A system for mining knowledge in large relational databases. In Proc. 1996 Int. Conf. Data Mining and Knowledge Discovery (KDD'96), pages 250{255, Portland, Oregon, August 1996.

[13] Jarernsri L.Mitrpanont, S.Fugkeaw. Direct Access Versioning for Multidimensional Database Schema Creation. Proceedings of the sixth IEEE International Conference on Computer and Information Technology (CIT'06), IEEE, 2006

[14] B.Bebel, Z.Krolinkowski, R.Wrembel. Formal approach to modeling a multiversion data warehouse. Bulletin of the Polish academy of sciences, Technical Sciences, Vol 54, No 1, 2006.

[15] G. Papastefanatos, P. Vassiliadis, A. Simitsis, Y. Vassiliou: What-if Analysis for Data Warehouse Evolution. April 2007. URL: www.dbnet.ece.ntua.gr/~gpapas/Publications/DataWarehouseEvolution-Extended.pdf.

[16] M. Blaschka, FIESTA: A Framework for Schema Evolution in Multidimensional Information Systems, Proc.6th CASE Doctoral Consortium, Heidelberg, Germany, 1999.