

Schema Evolution for Data Warehouse: A Survey

Meenakshi Arora

University School of Information Technology,
Guru Gobind Singh Indraprastha University
Delhi, India

Anjana Gosain

University School of Information Technology,
Guru Gobind Singh Indraprastha University
Delhi, India

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, Aggregate operator

1. INTRODUCTION

In today's information technology world, there is an explosive growth of information in various industrial sectors which led to operate huge data warehouse for collecting many different type of relevant information. This evolve us in handling a huge amount of data and achieving success in enhancements of knowledge and understanding of business environment at all level. 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 [16]. Schema evolution of data warehousing consist of various levels updates that is dimension updates, structural updates, instances updates, facts updates, attributes updates, cube updates, hierarchy updates, measure updates, quality updates and constraints 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 [17] 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.

2. LITERATURE SURVEY ON SCHEMA EVOLUTION

In the literature, different authors [1,2,3,4,5,6,7,8,9, 10,11,12,13,14,15] have proposed different operators to handle schema evolution at different levels. The schema evolution approach focuses on dimension updates [1,3,6, 8,9,13,15], instances updates [2,7,13,14], facts updates and attribute updates [3,5,6,8,10,12,13]. In [4], author has discussed about the quality factors and there affects on evolution operators. In [5,11,15], authors have discussed about the hierarchy updates. These sections discuss about theses operators in detail.

2.1. 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 explained in table 1: -

Table 1: Evolution operators

Operator	Description
Generalize operator	In this operator creation of new level (ln) to which a pre-existent one, l, roll up.
Specialize operator	It adds a new level ln to a dimension and becoming lowest level of the dimension
Relate level operator	This operator defines a roll up function between two independent levels belonging to the same dimension
Unrelated level operator	It allows deleting the relation between two levels
Delete level operator	It allows to delete a level then to define new relations between levels

Add instance Operator	It allows adding an instance to a level in the dimension
Delete instance operator	It allows deleting an instance to a level in the dimension

2.2.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] and. The complete set of instance update operators is as follows in table 2: -

Table 2: Evolution operators

Operators	Description
Add Instance	Inserting new level instance to given level
Delete Instance	Deleting an instance of either a top, or in-between, or a bottom level
Reclassify	Changing a association between a sublevel instance and its super level instance
Split	Splitting a given instance of a given level into several instance of the same level
Merge	Merging several instances of a given level into a one instance of the same level
Update	Updating several instances of a given level

2.3.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) explained in Table 3: -

Table 3: Description of tables

F	It defines finite set of fact names
L	It defines finite set of dimension level name
A	It defines finite set of attributes name
Gran	It is a function that associates a fact with a set of dimension level names
Class	It is relation defined on the level name
Attr	It is a function mapping an attribute to a given dimension level

In this paper [3], author defines various evolution operations listed in table 4 as follows: -

Table 4: Evolution operators

Operator	Description
These operators have no effects on the model or instances	
Insert level	It consists on extending the MD model by a new dimension level. This operation extends the set of levels without changing the classification relationships and creating isolated element. Classification relationship may be defined independently.

Delete level	It consists of deleting dimension level from an MD model but this dimension is not being connected to the fact. Instances are deleted automatically together with the dimension level
Insert attribute	It consists of creating new attribute without attaching it to a dimension level or fact. Assigning an existing attribute to a dimension level or fact is a separate operation (connect attribute).
Delete attribute	It consists on deleting an attribute which is a disconnected attribute means they are not attached to a dimension level or fact level
Insert classification relationship	It defines a classification relationship between two existing dimension levels.
Delete classification relationship	It consists on deleting a classification relationship without deleting the equivalent dimension levels.

These operators have effects on the model or instances

Connect attribute to dimension level	It consists on connecting an existing attribute to a dimension level.
Disconnect attribute from dimension level	It consists on disconnecting an attribute from a dimension level. This operation is all about removing the deleted attribute function
Connect attribute to fact	It consists on connecting an existing attribute to a fact. This operation defines the function that maps coordinates of cube to measures
Disconnect attribute from fact	It consists on disconnecting an existing attribute from a fact. This operation delete the function that maps coordinates of cube to measures
Insert fact	It consists of extending the MD model by a new fact. The operation extends the set of facts without attaching dimension levels to this fact. Since dimensions for this fact have to be defined separately.
Delete fact	It consists of removing an existing fact from the MD model but this fact must not be connected to any dimension or any attributes. This model has no effect on instances but has an effect on the MD model.
Insert dimension into fact	It consists on inserting a dimension at a given dimension level into an existing fact. Therefore, increasing the number of dimension by one This operation has an effect the computing of the new fact

Delete dimension	It consists on deleting a dimension which is connected to a fact from it. Additionally, an aggregation function has to be provided which defines the fact values have to be aggregated over the deleted dimension (e.g. summation).
------------------	---

2.4.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 below in table 5: -

Table 5: Evolution operators

Evolution Operator	Description
Add base relation/ view	Creating new relation or view
Delete base relation/ view	Removing the view
Add attribute to base relation/view	Addition of attribute in the view
Delete attribute from base relation/view	Removing of attribute in the view
Rename Relation, View , or Attribute	Rename of view and attribute
Change of attribute domain	Modification of domain attribute
Add Integrity Constraint	It allows addition of constraint
Delete Integrity Constraint	It allows deletion from the constraint
Change to view definition	It allows modification in the view

2.5.Bouzeghoub et.al. [2000]

In this paper [5], authors present a formal model for evolution of a data warehouse schema. Their formal model only supports two schema operators described in table 6: -

Table 6: Evolution operators

Operator	Description
Add a view	It creates a new view for a data warehouse
Delete a view	It deletes a view from the data warehouse.

2.6.Chen et al. [2002]

In this paper [6], authors present the schema evolution operators that are listed in table 7: -

Table 7: Evolution operators

Evolution Operator	Description
Drop an Attribute	It allows the deletion of level from the dimension.
Add an Attribute	It allows addition of an attribute operator adds a level to a dimension
Rename an Attribute	It allows the operator to change the name of a level in a dimension
Drop a Table	It allows the operator to deletes a fact or a dimension to a schema.
Add a Table	It allows the addition of a fact or dimension to a schema.
Rename a table	It allows changes the name of a fact or a dimension in a schema.

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

In this paper [7], 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 table 8: -

Table 8: Evolution operators

STRUCTURAL OPERATOR	INSTANCE OPERATOR
Generalize operator Relate levels operator Unrelated levels operator Delete level operator Specialize operator	Add Instance operator Delete Instance operator

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

In this paper [8], 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 described in table 9 as follows: -

Table 9: Evolution operators

Dimension operators		Cube operators	
Operators	Description	Operators	Description
Create Dimension	It allows creation of a new dimension table	Create cube	It allows the creation of a new cube.
Drop Dimension	It allows the deletion of existing dimension table	Drop Cube	It allows the deletion of a given cube.

Rename Dimension	It allows changing name of given dimension table	Rename cube	It allows the change of the name of a given cube.
Rename Level	It allows changing name of given level.	Rename Measure	It allows the change of the name of a given measure.
Add Level	It allows adding of a new level to a given dimension.	Add axis	It allows addition of an axis of analyse to a given cube
Delete Level	It allows the deletion of a level from a given dimension.	Delete axis	It allows the deletion of a given axis of analyse from a given cube
Add property	It allows the addition of attribute to given dimension or a given level.	Add measure	It allows addition of a measure to a given cube
Delete property	It allows the deletion of a property from a dimension or from a given level.	Delete measure	It allows the deletion of a measure from a given cube

2.9.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 discussed in table 10: -

Table 10: Evolution operators

Operators	Description
Insert dimension into fact	It consists on inserting a dimension at a given dimension level into an existing fact
Delete dimension	It consists on deleting a dimension which is connected to a fact from it.
Insert level	It consists on extending the multidimensional model by a new dimension level

Delete level	It consists of deleting dimension level from a multidimensional model
Connect attribute to dimension level	It consists on connecting an existing attribute to a dimension level. It may effects on the model or instances
Disconnect attribute from dimension level	It consists on disconnecting an attribute from a dimension level. This operation is all about removing the deleted attribute function
Add measure	It allows addition of a measure to a given cube.
Delete measure	It allows the deletion of a measure from a given cube.

2.10. 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 includes: -

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.

2.11. Golfarelli et.al. [2004]

In this paper [11], authors present a data ware schema graph supporting evolution. The schema evolution operators that supported are described in table 11: -

Table 11: Evolution operators

Operators	Description
Add _A	The add operator insert a new level in a dimension
Delete _A	The delete operator deletes a level from a dimension.
Add _F	This operator inserts a new hierarchy between two levels in a dimension that have a one- to-many relationships.
Delete _F	In this operator deletion of hierarchy from a dimension.

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

In this paper [12], 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 listed in table 12: -

Table 12: Evolution operators

Operators	Description
Add _L	It consists of adding new level in multidimensional model
Delete _L	It consists of deleting a new level dimension from multidimensional model
Add _A	It consists of adding dimension attribute in given dimension table.
Delete _A	It consists of deleting an dimension attribute from the given dimension table
Add _F	It consists of adding a dimension into given existing fact
Delete _F	It consists of removing the dimension which is connected to a existing fact
Add _{FA} to fact	It consists of adding fact attribute into the fact table
Delete _{FA} to fact	It consists of deleting the fact attribute from the fact table.

Along with that, 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. Authors in [12] discussed about level modification, attribute modification, fact modification and fact attribute modification.

2.13. B. Bebel et al. [2006]

In this paper [13], authors discussed about MVDW Operators (Multiversion Data Warehouse). According to the author operators may divided in two groups i.e. schema

change operator and dimension instance structure change operator. The complete sets of operators are given below in table 13: -

Table 13: Evolution operators

Schema Change Operator	Dimension Instance Structure Change Operators
<p>Schema change operators consist of 15 operators. Those operators are as follows: -</p> <ol style="list-style-type: none"> 1. Creating a new dimension 2. Creating a new level 3. Connecting level into a dimension hierarchy 4. Disconnecting a level from a dimension 5. Removing a dimension 6. Removing a level 7. Creating a new attribute for a level 8. Removing an attribute from a level 9. Changing the domain level attribute or fact attribute 10. Creating a new fact 11. Creating a new attribute for a fact 12. Removing an attribute from a fact table 13. Creating an association between a fact and a level 14. Removing an association between a fact and a level. 15. Removing a fact. 	<p>Dimension instance structure change operators consists of five basic operators namely:</p> <ol style="list-style-type: none"> 1. Inserting a new level instance 2. Deleting a level instance. 3. Reclassifying a level instance. 4. Merging n instances of a level into a new instance. 5. Splitting a level instance into n new instances.

2.14. George Papastefanatos et al. [2007]

In this paper [14], 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 listed in table 14 as follows: -

Table 14: Evolution operators

Operator	Description
Add table	It consists of adding the table into source table.
Delete table	It consists of deleting the attribute from source table.
Rename table	It consists of changing the name of source table.
Rename attribute	It consists of changing the name

	of attribute of the particular table
Modify	It consists of modifying the domain of attributes
Constraint	It consists of changing the primary key of the dimension table

2.15. S. Banerjee et al. [2009]

In [15], author worked on various approach of schema evolution. The schema evolution operators are the core features of a data warehouse that are defined in the generalized model. Generalized model we have already discussed in [1]. Authors have summarized various schema evolution operators based on change in dimension, change in fact and change to a cube. Change in dimension consist of add/delete dimension to schema, rename dimensionIn schema. Add/Delete level to dimension. Rename level In dimension. Add/Delete hierarchy to dimension. Change in fact consists of Add/Delete fact to schema, Rename factIn schema. Change to a cube consists of add dimension to fact in cube and delete dimensionfromfactIncube. In this paper [15], authors have proposed hierarchies operators for schema evolution.

The operators may be defined in table 15: -

Table 15: Evolution operators

Operator	Description
Multiple hierarchies	It consists of a dimension can have multiple paths to roll-up or drill-down information
Non-covering hierarchies	Its consists of a parent level in a non-covering hierarchy is an ancestor of the child level
Non-onto hierarchies	An instance of a parent level in a dimension can exist without a corresponding data instance in the child level to drill-down
Non-strict Hierarchies	It consists of Two levels in a dimension can have a many-to-many relationship

3. 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 represented in table 16: -

Table 16: Aggregate fact table operators

Operator	Description
Create Aggregate	It allows creation of new aggregate table.
Delete Aggregate	It allows deletion of given aggregate table.
Alter Aggregate	It allows altering of given aggregate table (adding, modifying)
Rename Aggregate	It allows the change of the name of a given aggregate table
Drop Aggregate	It allows the deletion of existing aggregate table

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 figure.1 represented a derived aggregate fact table ‘SALES’ connected to a derived dimension table ‘CATEGORY’.

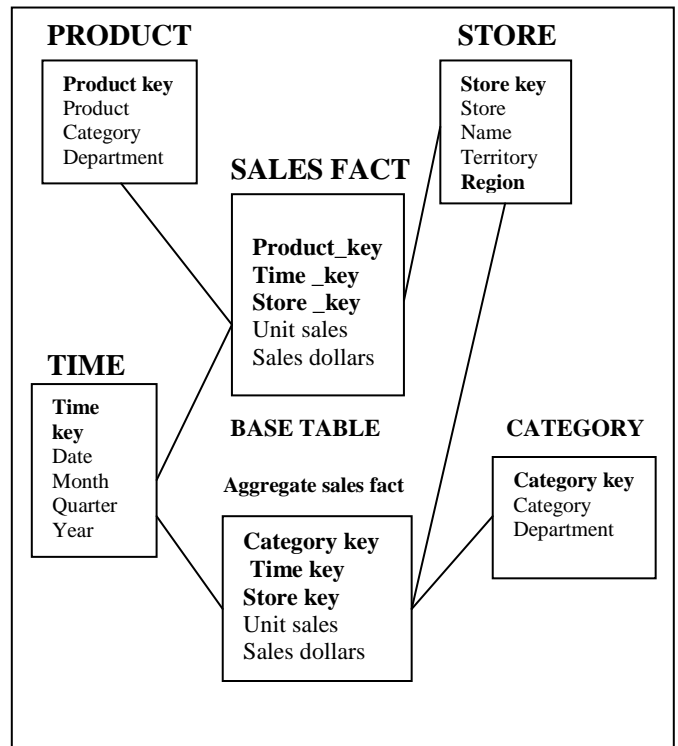


Fig: 1 Creation of Aggregate fact table and derived dimension table [18]

The rename operator change the name of the derived aggregate fact table ‘SALES’ to the ‘MARKET’ fact table in figure 2.

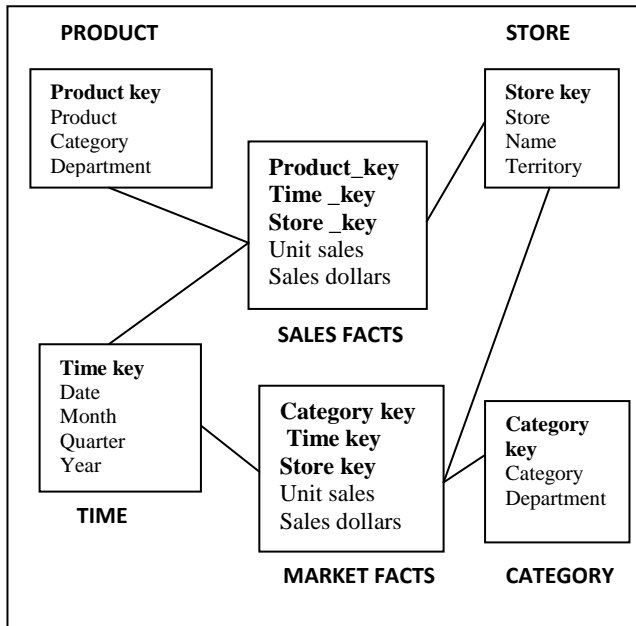


Fig: 2 Renaming of Aggregate fact table and derived dimension table [18]

For defining data warehouse and data marts we examine SQL- based data mining query language called DMQL proposed in [19]. 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.

4. 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, 7, 8, 9, 10, 11, 12, 13, 15] 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, 8, 13, 15] were focused on dimension evolution and fact evolution. Benitez [8] presented add/delete /rename of measure whereas Blashka [3] presented add and delete of fact. However, both authors in [3, 8] presented add/ delete dimension and add/ delete attribute but only Benitez [8] proposed rename of dimension. Later in paper [6], author defined add/ delete/ rename operators for attribute, fact and dimension table.

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

Authors in [5], proposed the formal model for the data warehouse. There formal model represents the view of the data warehouse after the schema evolution. The formal model of data warehouse consists of dimensions, levels, facts, hierarchy, measures and attributes.

Vaisman [7] 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 [7] 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, 13] 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 [13] presented dimension evolution, level evolution, instance evolution, fact evolution, attribute evolution.

Golfarelli [11] present a data warehouse schema graph supporting evolution that allow adding/ deleting nodes and edges.

J.L. Mitranont [12] 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 [14] 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.

S.Banerjee [15] focused on various approaches of data warehouse and summarized different schema evolution operators. Authors in [15] proposed hierarchies operators for evolution.

Comparative study for schema evolution operators proposed by different authors is given below in (Table 17):

Table 17: Comparative study for schema evolution operators

Criteria Author	Attribute changes	Constraint changes	Cube changes	Dimension changes	Fact changes	Hierarchy changes	Instance changes	Level changes	Measure changes	Quality changes	View changes
Hurtado [1]								★			
Hurtado [2]							★	★			
Blaschka [3]	★			★	★			★			
Quix [4]	★	★								★	★
Bouzeghoub [5]	★			★	★	★		★	★		★
Chen [6]	★			★	★						
Vaisman [7]							★	★			
Benitez [8]	★		★	★	★			★	★		
Kaas [9]	★			★				★	★		
T.Morzy [10]	★	★			★			★			
Golfarelli [11]						★		★			
Mitranont [12]	★				★			★			
Bebel [13]	★			★	★		★	★			
George [14]	★	★					★				
Banerjee [15]			★	★	★	★		★			

5. 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, 7, 8, 9,10,11,12,13,14,15] 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, hierarchy updates, instances updates, fact 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.

6. 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 Warehousing 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] Bouzeghoub, M., and Z. Kedad, "A Logical Model for Data Warehouse Design and Evolution," Proceedings of the 2nd International Conference on Data Warehousing and Knowledge Discovery (DaWaK), London, UK, September 4-6, 2000, pp. 178-188.
- [6] Chen, J., Chen, S., Rundensteiner, E.: A transactional model for data warehouse maintenance. In: Spaccapietra, S., March, S.T., Kambayashi, Y. (eds.) ER 2002. LNCS, vol. 2503, pp. 247–262. Springer, Heidelberg (2002)
- [7] 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
- [8] E. Benitez- Guerrero, C.Collet, M. Adiba. THE WHES APPROACH TO DATA WAREHOUSE EVOLUTION. E-Gn osis[online], vol.2Art.2004
- [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] Golfarelli, M., Lechtenböcker, J., Rizzi, S., Vossen, G.: Schema Versioning in Data Warehouses. In: Wang, S., Tanaka, K., Zhou, S., Ling, T.-W., Guan, J., Yang, D., Grandi, F., Mangina, E.E., Song, I.-Y., Mayr, H.C. (eds.) ER Workshops 2004. LNCS, vol. 3289, pp. 415–428. Springer, Heidelberg (2004)
- [12] Jarernsri L.Mitranont, 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
- [13] 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.
- [14] 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.
- [15] S.Banerjee, K.C.Davis, Modeling Data Warehouse Schema Evolution over Extended Hierarchy Semantics, S.Spaccapietra et.al (EDs): Journal on Data Semantics XIII, LNCS 5530, pp.72-96,2009.@Springer- Verlag Berlin Heidelberg 2009.
- [16] W.Oueslati, J. Akaichi. A survey on Data warehouse evolution. International Journal of Database Management Systems (IJDMS), Vol.2, No.4, November 2010.
- [17] Adriana Marotta. Data Warehouse Design and Maintenance through Schema Transformations. Master thesis, October 2000.
- [18] Paulraj Ponniah. Data Warehousing Fundamental Guide. Published by John Wiley and Sons, 2001
- [19] 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. Zane. 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.
- [20] M. Blaschka, FIESTA: A Framework for Schema Evolution in Multidimensional Information Systems, Proc.6th CASE Doctoral Consortium, Heidelberg, Germany, 1999.