World Academy of Science, Engineering and Technology
International Journal of Computer and Information Engineering
Vol:2, No:11, 2008

Coalescing Data Marts

N. Parimala, and P. Pahwa

***Abstract*—OLAP uses multidimensional structures, to provide access to data for analysis. Traditionally, OLAP operations are more focused on retrieving data from a single data mart. An exception is the drill across operator. This, however, is restricted to retrieving facts on common dimensions of the multiple data marts. Our concern is to define further operations while retrieving data from multiple data marts. Towards this, we have defined six operations which coalesce data marts. While doing so we consider the common as well as the non-common dimensions of the data marts.

***Keywords*—Data warehouse, Dimension, OLAP, Star Schema.

I. Introduction

DATA Warehouse [1], [2] is a subject oriented, non-volatile collection of data used to support strategic decision-making. The warehouse is the central point of data integration for business intelligence. It is the source of data for data marts within an enterprise and delivers a common view of enterprise data. The data in a data warehouse is structured in a uniform way, along dimensions and facts.

OLAP [3, 4] is the most important approach for analyzing data in a data warehouse. Relevant work includes OLAP data modeling and querying [5], [6], [7]. Using OLAP one can query and analyze data stored as a star schema from many different perspectives. However, most operations are concerned with analyzing data only from one star schema with the exception of drill across. In drill across [8], [9], [10] facts from multiple star schema can be retrieved if they have common dimensions. The common dimensions are used to essentially perform a join between the two star schemata. This may not be adequate in all situations.

Assume for instance there are three data marts, depicted as cubes in Fig. 1, Fig. 2 and Fig. 3. These cubes are representing information of the sales of items locally by clerks, purchase of items locally by customers and sales of items globally by clerks. Let us say that the company wants to know the overall performance of the sales of items. This will help the company get a total view of its operations. This information that has to obtained is the totality of information from the first and the third cubes. The operation that is most popular in OLAP when retrieving data from more than one cube is drill across.

Drill across is performed using common dimensions. The non-common dimensions do not appear in the result. In our example if one were to drill across `SalesLocal` and `SalesGlobal` then the resultant cube will have the items and the `Clerk` dimensions. It will not have the `Local` or the `Global` address information. Thus, the company can no way get the overall picture.

So we need an operation which can combine more than one cube without losing non common dimensions. Accordingly, we study the semantic relationships between the non-common dimensions. The relationship that is considered is 'is-a' relationship. Intuitively, if two dimensions exhibit a is-a relationship with a common dimension, then they can be viewed as members of the parent dimension and can be combined in a meaningful way.

If two dimensions D1 and D2 are in a is-a relationship with a dimension D, then two cases arise regarding the instances of D1 and D2 viz.

a) Instances of D1 and D2 are overlapping
b) Instances of D1 and D2 are disjoint.

For example, if `Customer` and `Clerk` are in a is-a relationship with Person, there can be clerks who are customers as well. On the other hand, if Address is of two kinds – `Local` and `Global` – then there is going to be no common instance between them. We define multiple forms of the *coalesce* operation to combine two cubes so that the different ways of combining the two cubes can be handled.

There is a second aspect about the desired result. We do not wish to lose any information while combining the cubes. For example, if an item, say, Pens is sold locally but not globally, then this fact is present in the `SalesLocal` cube of Fig. 1 but not in that of Fig. 3. It is necessary that this data be present when the cubes are combined if we want to know the overall performance. However, we lose such information when we perform drill across. This is because when we drill across only the common instances of the common dimensions are considered. In fact, at the lower level a equi join which is an inner join on common dimension is performed. As a result, facts about the instances which belong to one of the cubes and not to the other are ignored. The operation *OuterCoalesce* is introduced to combine cubes without losing the facts which are present in only one of the cubes.

Drill across has been studied extensively in the literature [8], [9], [10]. Apart from the common form where drill across is performed using common instances of common dimensions, drill across has been defined when there is a relationship among dimensions of different schemata. Reference [10], [11] has introduced the notion of conforming dimensions where two dimensions in different schema are conforming if there instances exactly coincide. Reference [9] defines four kinds of relationships between dimensions (derivation, generalizations, association and flow). Using these relationships it is possible to drill across by using the operation change base which helps to substitute one dimension with the other and view the information in a new space. Reference [10] introduces compatible dimensions as those dimensions which share some information and this information is consistent. In all the above definitions, dimensions which can be used for drilling across

Manuscript received September 8, 2006

Prof. Parimala N. is with Jawahar Lal Nehru University, Delhi, India (phone: 91-11-26704770; email: parimala@mail.jnu.ac.in).

P. Pahwa is with Guru Premsukh Memorial College of Engg, Delhi, India (phone: 09810392902; email: payal.chhabra@yahoo.com).

are those dimensions that share some common instances. Our work differs from all drill across navigation since we take into consideration non-common dimensions. Some authors have defined algebraic operations of union and intersection has been defined to combine two star schemata. In [13] the union operation combines two cubes if they are compatible. Similar is the treatment for intersection of two cubes. These operations are inapplicable in our case as the two cubes may not necessarily be compatible. The join operation as defined in [9] combines all the common and the non common dimensions of the participating cubes. There is no semantic definition for such a combination. The proposals here vary from algebraic operations on cubes since here the operations which combine cubes are permitted only when certain relationships hold.

The rest of the paper is organized as follows. Section 2 introduces the data model used in this paper. Section 3 defines the 'is-a' relationship between dimensions. The operations themselves are defined in section 4. Section 5 is the concluding section.

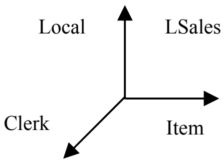

Fig. 1 SalesLocal

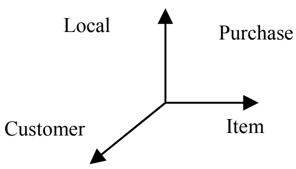

Fig. 2 PurchaseFact

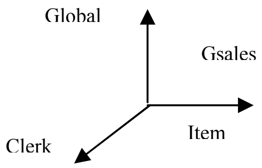

Fig. 3 SalesGlobal

We use the above examples throughout the paper.

II. The Data Model DM

In this section we describe the data model DM that will be used throughout the paper. The definition includes the concepts required to define a data mart. The data model defined here is defined in terms of a Cube which is composed of dimensions and a fact scheme.

A dimension consists of dimensional attributes. Each dimension consists of a graph which represents the hierarchy of levels of the dimensional attributes. The attributes can be rolled up along the edges and in the direction of the edge. For example, product dimension consists of two dimensional attributes – item and category as shown in Fig. 4. Item can be grouped into category and this is performed using the roll-up function.

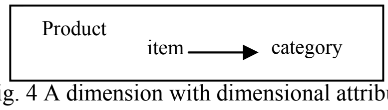

Fig. 4 A dimension with dimensional attributes

A fact scheme associates measure to the attributes of dimensions and are used to represent the actual data given by the facts. For example, the daily sales can be represented by a fact scheme that associates item with a `Clerk` and a `Local` address. Below we define these terms more rigorously.

A. Dimension

A Dimension is composed of

- a set of dimensional attributes V. Each attribute has a set of instances associated with it.
- a connected, directed graph D(V,E). Every vertex in the graph corresponds to an aggregation *Level*, and an edge (ai, aj) reflects that ai can be rolled up to aj. An instance of aj decomposes into a collection of instances of ai. Each *Level* corresponds to a granularity in the *Dimension*.

B. Fact Scheme

A *Fact* scheme is an expression of the form f [D_1 : A_{11}, D_2 : A_{21} … D_n : A_{n1}] → [M_1, M_2 .. M_n] where A_{i1} is an attribute of dimension D_i. M_1, M_2, ….M_n are distinct measures.

C. Cube

A *Cube* has the following components

- N dimensions
- The fact scheme
- Set of n tuples of the form (a_{11}, a_{21} …a_{n1}, m_1, m_2 … m_n) where a_{ij} is a value of attribute A_{ij} of dimension D_i and m_l is a value of measure M_i.

D. Common Dimensions

Two dimensions D1 and D2 are common if they do not differ in the set of dimensional attributes and the graph is identical.

E. Non Common Dimensions

Two dimensions D1 and D2 are non-common if they differ in at least one attribute.

III. 'is-a' Relationship

In this section we define 'is-a' relationship between dimensions in order to be able to combine two cubes. Let C1 and C2 be two cubes with D1 and D2 as dimensions respectively. Let D1 and D2 be non–common dimensions. As defined above, a dimension consists of attributes at different levels. If an attribute A_i of dimension D1 and an attribute A_j of dimension D2 are specializations of an entity C then we create a new dimension D with C as one of the attributes. The dimension D contains the intersection of the subgraphs of D1 and D2. For example, consider Fig. 5. `Local` is an attribute of the dimension `Local_address` and `Global` is an attribute of `Global_address` dimension. Let these be specializations of an entity `Addresss`. A new dimension, say, `Location` is created with Address as an attribute. The graph that is common between `Local_address` and `Global_address` is part of the `Location` dimension as shown in the Fig. 5. Notice that in the `Location` dimension, it is possible to roll-up from Address to Country.

We first define the operation *reduction* in order to find the intersection of dimensions. Subsequently, we explain the manner in which a new dimension can be created.

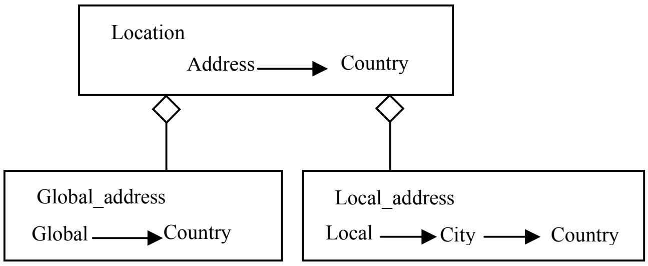

Fig. 5 'is-a' relationship between dimensions

A. Intersection of Dimensions

We define a function *reduction* on a tree as follows. Consider a tree t1(V1,E1) where V1 is the set of vertices and E1 is the set of directed edges. The tree t2(V2, E2) is the *reduction* of t1(V1,E1) if V2 $\subseteq$ V1 and an edge (vi,vj) belongs to E2 when

a) vi and vj belong to V2
b) there exists a path from vi to vj in t1
c) all nodes except vi and vj are not in V2

A dimension which contains a graph can be viewed as multiple trees with the lowest level as the root and the aggregation levels along a path as the nodes and the edges maintained as in the dimension definition. Given two dimensions, if the trees in the respective dimension can be reduced to a tree containing the common attributes, then this tree will be part of the intersection of the two dimensions. This can be applied repeatedly to each tree of the dimension. More formally, consider two dimensions D1(V1,E1) with A_i ε V1 for 1≤ i ≤ n and edges Ai → Ai+1and D2(V2, E2) with B_i ε V2 for 1≤ i ≤ m and edges Bi → Bi+1. Let V be the set of common attributes of D1 and D2. In other words V = V1 ∩ V2. Then, reduction of D1(V1,E1) to D(V,E) which is the same as reduction of D2(V2,E2) to D(V,E) is the intersection of D1 and D2.

B. Creation of a Dimension

Consider two dimensions D1(V1,E1) with A_i ε V1 for 1≤ i ≤ n and edges Ai → Ai+1and D2(V2, E2) with B_i ε V2 for 1≤ i ≤ m and edges Bi → Bi+1. Without loss of generality, we can assume that A_1 and B_1 are in a 'is-a' relationship with C_1. If the 'is-a' relationship is not with respect to the lowest level attributes, then appropriate roll-up operations can be performed so that the lowest level attributes in both the cubes exhibit 'is-a' relationship. Replace A_1 with C_1. Similarly replace B_1 with C_1. The intersection of D1 and D2 is the new dimension. The new dimension D is the parent dimension of both D1 and D2.

IV. Operations

The operations here have been defined in terms of the kind of decision support information that is sought. We define operations for coalescing two cubes along common dimensions as well as along the common and the non-common dimensions.

In the case of common dimension, two cases arise

a) include only the common instances
b) include all the instances.

Case (a): In the first case, information of only the common instances may be required. For example, for the cubes of Fig.1 and Fig. 2, let us say that it is required to know the comparative figures of sales and purchase of clerks. Then only the instances who are clerks as well as customers participate.

Case (b): In this case, all the instances participate in decision making. For example, it may be required to know the total sales and purchase figures immaterial of who sold or purchased the items.

Case (a) is the typical drill across query. For this we define the operation *InnerCoalesce*. *OuterCoalesce* handles the case when all the instances of the common dimension are included in the resultant cube.

In the case of non-common dimension, it is possible to coalesce only if there is a 'is-a' relationship. Recall that if two dimensions D1 and D2 are in a 'is-a' relationship with a dimension D, then two cases arise regarding the instances of the attributes of D1 and D2 viz.

a) the instances are overlapping
b) the instances are disjoint.

Case (a): In this case the instances of the attributes of D1 and D2 are overlapping. In our example, for the cubes of Fig.1 and Fig.2, if clerk and customer are in 'is-a' relationship with people, then their instances can overlap as there can be clerks who are customers as well.

Case (b): In this case the instances of the attributes of D1 and D2 are disjoint. In our example, for the cubes of Fig.1 and Fig.3, if `Local-address` and `Global_address` are in a 'is-a' relationship with Location, then there are no common instances between the attributes `Local` and `Global`. However, it may be required to know the overall sales performance.

Summarizing, then, there are six different ways of combining the instances of the two cubes which is shown in the Table I and Table II.

TABLE I
COMBINING INSTANCES OF TWO CUBES

Common instances of C	*InnerCoalesce*
All instances of C	*OuterCoalesce*

TABLE II
COMBINING INSTANCES OF TWO CUBES

Common Dimension C	*New Dimension D*	*Operation*
Common instances	Common instances	*CommonInnerCoalesce*
Common instances	All instances	*AllInnerCoalesce*
All instances	Common instances	*CommonOuterCoalesce*
All instances	All instances	*AllOuterCoalesce*

We now define each operation individually

(i) InnerCoalesce

The above operation joins the two cubes on the common dimensions. For the common dimensions the join is an equi-join.

Let the dimensions of the two cubes C1 and C2 be D_1--D_b, D_{b+1}----D_n and D_1--D_b, D'_{b+1}----D'_s. where D_1--D_b are the common dimension. Let E and E′, the fact schemes of the two cubes, be of the form

E:[D_1--D_b:A_{11}---A_{b1},D_{b+1}: A_{b1+1}---D_n: A_{n1}] $\rightarrow$[M,M_1 --- M_m]

and

E':[D_1--D_b:A_{11}--A_{b1},D'_{b+1} :B'_{b1+1}-- D'_s:B'_{s1}]$\rightarrow$[M′,M'_1 -- M'_p]

where A_{11}--A_{b1} are the dimensional attributes which are common. Note that, if the levels of the attributes are not the same in the two fact schemes, then one of them can be rolled up to bring the two fact schemes at the same level on the common dimensions.

C1 InnerCoalesce C2 is a cube C which consists of the dimensions D_1-- D_b. The measures are M, M′, M_{1}---- M_m ,M'_1---- M'_p. The fact scheme is

E: [D_1--D_b : A_{11}--A_{b1}] $\rightarrow$ [M, M′, M_{1}---- M_m ,M'_1---- M'_p].

The instances of D_1, D_2 …D_b are those which are common in both the cubes. That is, an instance of each attribute of the common dimension exists in C if it exists in both C1 and also in C2. It also consists of a set of n tuples of the form (a_{11},a_{21}…a_{b1},m,m′,m_1..m_m, m'_1.. m'_p) where a_{11} is the value of the attribute A_{11} of dimension D_1 and m is the value of measure M_1.

Example 1 Suppose information about items which have been purchased and sold for cubes C1 and C2 is to be found. Then C1 *InnerCoalesce* C2 gives the desired result as follows:

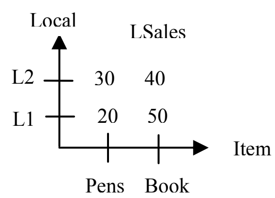

Fig. 6 Cube C1

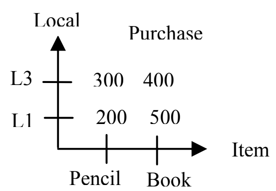

Fig. 7 Cube C2

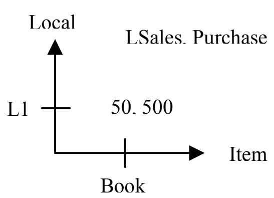

Fig. 8 C1 InnerCoalesce C2

(ii) OuterCoalesce

This operation also joins the two cubes on the common dimensions. But the common dimensions are joined with the semantics of outer join.

Let the dimensions of the two cubes C1 and C2 be D_1--D_b, D_{b+1}----D_n and D_1--D_b, D'_{b+1}----D'_s. where D_1--D_b are the common dimension. Let E and E′, the fact schemes of the two cubes, be of the form

E:[D_1--D_b:A_{11}---A_{b1},D_{b+1}: A_{b1+1}--- D_n: A_{n1}]$\rightarrow$[M,M_1 --- M_m]

and

E′:[D_1--D_b:A_{11}--A_{b1},D'_{b+1} :B'_{b1+1}-- D'_s :B'_s]$\rightarrow$[M′,M'_1 -- M'_p]

where A_{11}--A_{b1} are the dimensional attributes which are common. Note that, if the levels of the attributes are not the same in the two fact schemes, then one of them can be rolled up to bring the two fact schemes at the same level on the common dimensions.

C1 OuterCoalesce C2 is a cube C which consists of the dimensions D_1-- D_b. The measures are M, M′, M_{1}---- M_m ,M'_1---- M'_p. The fact scheme is E: [D_1--D_b:A_{11}--A_{b1}] $\rightarrow$ [M, M′, M_{1}---- M_m ,M'_1---- M'_p]. It also consists of a set of n tuples of the form (a_{11},a_{21}…a_{b1},m,m′,m_1..m_m, m'_1.. m'_p) where a_{11} is the value of the attribute A_{11} of dimension D_1 and m is the value of measure M_1.

The instances of D_1, D_2 …D_b are those which are in either of the cubes. That is, an instance of each attribute of the common dimension exists in C if it exists either in C1 or in C2.

Example 2 Suppose it is desired to find all the sale and purchase information for the items, shown in cubes C1 and C2,. Then C2 *OuterCoalesce* C1 gives the desired result as follows:

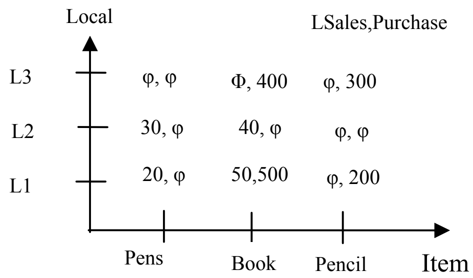

Fig. 9 C1 OuterCoalesce C2

(iii) CommonInnerCoalesce

The operations defined above took into account only the common dimensions. This operation combines the cubes on the common as well as a non common dimension provided 'is-a' relationship exists for some attribute of the non common dimension of each cube as explained above.

Let the dimensions of the two cubes C1 and C2 be D, D_1--D_b, D_{b+1}----D_n and D′, D_1--D_b, D'_{b+1}----D'_s. where D_1--D_b are the common dimension and D and D′ are the non common dimensions. Let E and E′, the fact schemes of the two cubes, be of the form

E:[D:A,D_1--D_b:A_{11}---A_{b1},D_{b+1}:A_{b1+1}--D_n:A_{n1}]$\rightarrow$[M,M_1--M_m]

and

E′:[D′:B,D_1--D_b:A_{11}--A_{b1},D'_{b+1}:B'_{b1+1}--D'_s:B'_{s1}]$\rightarrow$[M′,M'_1--M'_p]

where A_{11}--A_{b1} are the dimensional attributes which are common. Let the attribute A of D and the attribute B of D′ exhibit a 'is-a' relationship with an attribute C. Construct the dimension D″ as explained above.

C1 CommonInnerCoalesce C2 is a cube C which consists of the dimensions D″D_1-- D_b. The measures are M,M′,M_{1}--

M_m,$M'_{1----}M'_p$. The fact scheme is E: [D", D_1… D_b:A_{11}…A_{b1}]→ [M, M′, $M_{1----}M_m$,$M'_{1----}M'_p$]. Again it also contains a set of n tuples of the form as described in Inner coalesce. The tuple is a set of instances as described below:

The instances of D_1, D_2 …D_b are those which are common in both the cubes. That is, an instance of each attribute of the common dimension exists in C if it exists in both C1and also in C2. The instances of D″ are those which are in both D and D′. In particular, the instances of C are those which are in both A and B. In other words, the overlapping instances of A and B form the instances of D".

Example3: Suppose it is desired to find the common items which are purchased by customer who are clerks as well then C3 CommonInnerCoalesce C4 gives the answer as shown below:

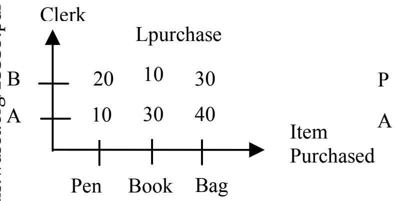

Fig. 10 Cube C3

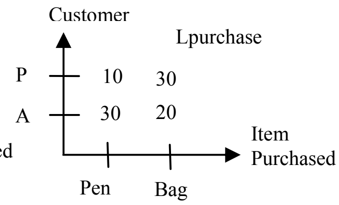

Fig. 11 Cube C4

C3 CommonInnerCoalesce C4 as shown below gives the result of the above query.

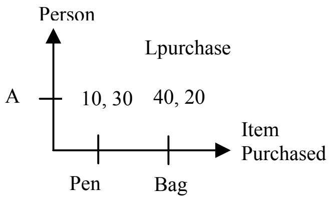

Fig. 12 C3 CommonInnerCoalesce C4

(iv) CommonOuterCoalesce

This operation also combines the cubes on the common as well as a non common dimension provided 'is-a' relationship exists for some attribute of the non common dimension of each cube as explained above. The operations defined above took into account only the common instances of the common dimensions. Here, the union of instances of the common dimension is taken.

Let the dimensions of the two cubes C1 and C2 be D, D_1--D_b, D_{b+1}----D_n and D′, D_1--D_b, D'_{b+1}----D'_s. where D_1--D_b are the common dimension and D and D′ are the non common dimensions. Let E and E′, the fact schemes of the two cubes, be of the form

E:[D:A,D_1--D_b:A_{11}---A_{b1},D_{b+1}:A_{b1+1}--D_n:A_{n1}]→[M,M_1--M_m]

and

E′:[D′:B,D_1--D_b:A_{11}--A_{b1},D'_{b+1}:B'_{b1+1}---D'_s:B'_{s1}]→[M′,M'_1---M'_p]

Where $A_{11...}A_{b1}$ are the dimensional attributes which are common. Let the attribute A of D and the attribute B of D′ exhibit a 'is-a' relationship with an attribute C. Construct the dimension D″ as explained above.

C1 CommonOuterCoalesce C2 is a cube C which consists of the dimensions D″D_1-- D_b. The measures are M, M′, M_{1----} M_m ,M'_{1----} M'_p. The fact scheme is E: [D", D_1--D_b:A_{11}…A_{b1}] → [M, M′, M_{1----} M_m ,M'_{1----} M'_p]. Again it also contains a set of n tuples of the form as described in Inner coalesce. The tuple is a set of instances as described below:

The instances of D_1, D_2 …D_b are those which in either of the cubes. The instances of C are those which are in both A and B. In other words, the overlapping instances of A and B form the instances of C.

Example 4: Suppose it is required to find all the items which are purchased by customers who are clerks as well. This can be answered using CommonOuterCoalesce for the cubes of Fig. 10 and Fig. 11 which gives the following result:

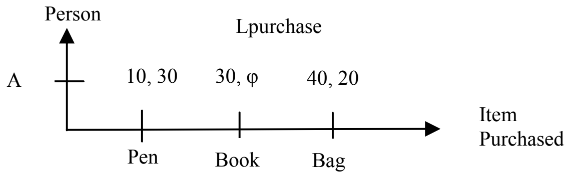

Fig. 13 C3 CommonOuterCoalesce C4

(v) AllInnerCoalesce

This operation also combines the cubes on the common as well as a non common dimension provided 'is-a' relationship exists for some attribute of the non common dimension of each cube as explained above. The operations defined above took into account only the common instances of the non common dimensions. This operation handles those cases where the attributes which are specializations are disjoint.

Let the definitions of the two cubes C1 and C2 be as defined in *CommonInnerCoalsece*.

C1 AllInnerCoalesce C2 is a cube C which consists of the dimensions D″D_1-- D_b. The fact scheme is

E: [D", D_1--D_b:A_{11}…A_{b1}] → [M, M′, M_{1----} M_m ,M'_{1----} M'_p]. Again it also contains a set of n tuples of the form as described in Inner coalesce. The tuple is a set of instances as described below:

The instances of D_1, D_2 …D_b are those which are in both C1 and C2. The instances of C are the union of instances which are in A and B.

Example5: Consider the common items purchased by all the customers and clerks.

C3 AllInnerCoalesce C4 gives the answer to the query.

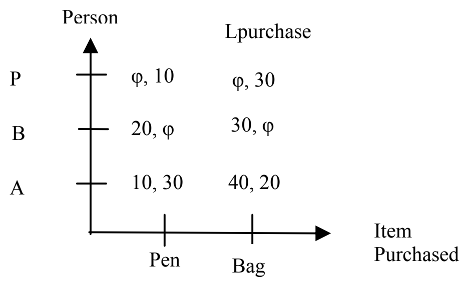

Fig. 14 C3 AllInnerCoalesce C4

(vi) AllOuterCoalesce

The operation defined above took into account only the common instances of the common dimensions. This operation includes all the instances for both the common as well as the non common dimension.

Let the definitions of the two cubes C1 and C2 be as defined in *CommonInnerCoalsece*.

C1 AllOuterCoalesce C2 is a cube C which consists of the dimensions $D''D_1$-- D_b. The fact scheme is

$$E: [D'', D_1\text{--}D_b:A_{11}\ldots A_{b1}] \rightarrow [M, M', M_{1----} M_m, M'_{1----} M'_p].$$

Again it also contains a set of n tuples of the form as described in Inner coalesce. The tuple is a set of instances as described below:

The instances of D_1, D_2 …D_b are those which are in either of C1 and C2. The instances of C are the union of instances which are in A and B.

Example 6: Suppose an organization wants the complete purchase information about all the persons immaterial of whether they are clerks or customers. Then the desired result can be obtained from C3 AllOuterCoalesce C4.

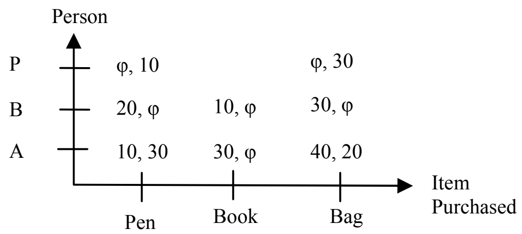

Fig. 15 C3 AllOuterCoalesce C4

V. Conclusion

In this paper we have combined two cubes along common as well as along non common dimensions. The main concern while combining two cubes using common dimensions has been to consider the non-common instances at par with the common instances. We have also shown that the non-common dimensions can also be used to combine cubes if there is a 'is-a' relationship. We show the manner in which a new dimension can be created.

We define six operations to combine the instances of two cubes. The six operations are *InnerCoalesce, Outer Coalesce,CommonInnerCoalesce, CommonOuterCoalesce, AllInnerCoalesce* and *AllOuterCoalesce*. The first two consider only the common dimensions whereas the last four take into account the non-common dimensions as well.

Just as common dimensions alone can be used to coalesce two cubes, it can be argued that non-common dimensions alone can also be used to coalesce two cubes. However, we find that the result may not help in decision making. For example, if we coalesce cubes of Fig. 2 and Fig. 3 along non-common dimensions alone, then the resultant cube will have Address and Person as dimensions. The data in the resultant cube is essentially about the items which are sold and purchased. In the absence of item dimension, any meaningful decision cannot be taken.

It can be argued that coalescing two dimensions into one cube introduces a lot of null values but we believe that unnecessary null values must be avoided when storing information but may not be avoidable when it is essential to get a global picture.

Reference [9] has also considered 'is-a' relationship. However, here it is assumed that a dimension already exists exhibiting the relationship. We believe that while combining data marts, extra dimension such as those in [9] do not exist. Therefore, we propose to create a new one.

References

[1] W. H. Inmon: Building the Data Warehouse. John Wiley and Sons, Second Edition 1996.

[2] R. Kimball.: The Data Warehouse Toolkit, John Wiley and Sons 1996.

[3] A. Abello, J. Samos, F. Saltor: "Implementing Operations To Navigate Semantic Star Schemata" in 6th International Workshop on Data Warehousing and OLAP (DOLAP 2003).New Orleans(USA),November 2003.Pages 56-62.ACM Press 2003 ISBN 1-58113-727-3.

[4] S. Chaudhari, U. Dayal: "An Overview of Data Warehousing and OLAP Technology" SIGMOD Record,26(1):65-74 1997.

[5] M. Golfarelli, D. Maio, S. Rizzi: "The Dimensional Fact Model: A Conceptual model for Data Warehouses" Int.Tour of Cooperative Inf. Systems 7, 2&3 1998.

[6] P. Pahwa and Parimala N.: "Conceptual Design of Data Warehouses From XML Schemata using CCM" 2nd Int. Conference on Intellectual Capital, Knowledge Management and Organizational Learning, Dubai 2005.

[7] C. Li and X.S. Wang: "A data model for supporting on-line analytical processing" in proceedings of the 5th International Conference on Information and Knowledge Management (CIKM'96), pages 81-88 1996.

[8] L. Cabibbo, R. Torlone: "On the Integration of Autonomous Data Marts" in 16th Int. Conference on Scientific and Statistical Database 2004.

[9] A. Abello, J. Samos, F. Saltor: "On Relationships Offering New Drill Across Possibilities" in Int. Workshop on Data Warehousing and OLAP (DOLAP2002).ACM 2002.

[10] L. Cabibbo, R. Torlone"Dimension Compatibility for Data Mart Integration"Proc. 12th Italian Symposium on Advanced Databases systems, Caglian Italy,2004,PP 6-17.

[11] W. Giovinazzo. O-O Data Warehouse Design. Prentice Hall, 2000.

[12] R. Agarwal, A. Gupta, S. Sarawagi: "Modelling Multidimensional Databases" in proceedings of 13th International Conference on Data Engineering (ICDE'97), Pages 232-243. IEEE Computer Society, 1997.