W. Hümmer, W. Lehner, A. Bauer, L. Schlesinger

A Decathlon in Multidimensional Modeling. Open Issues and Some Solutions

Erstveröffentlichung in / First published in:

Data Warehousing and Knowledge Discovery. 4th International Conference, DaWaK 2002. Aix-en-Provence, 04.-06.09.2002. Springer, S. 275-285. ISBN 978-3-540-46145-6.

DOI: http://dx.doi.org/10.1007/3-540-46145-0_27

Final edited form was published in "Data Warehousing and Knowledge Discovery.4th International Conference, DaWaK 2002. Aix-en-Provence 2003", S. 275–285,
ISBN 978-3-540-44123-6
http://dx.doi.org/10.1007/3-540-46145-0_27

A Decathlon in Multidimensional Modeling: Open Issues and Some Solutions

W. Hümmer, W. Lehner, A. Bauer, L. Schlesinger
University of Erlangen-Nuremberg (Database Systems)
Martensstr. 3, Erlangen, D-91058, Germany
{huemmer,lehner,bauer,schlesinger}@immd6.informatik.uni-erlangen.de

Abstract. The concept of multidimensional modeling has proven extremely successful in the area of Online Analytical Processing (OLAP) as one of many applications running on top of a data warehouse installation. Although many different modeling techniques expressed in extended multidimensional data models were proposed in the recent past, we feel that many hot issues are not properly reflected. In this paper we address ten common problems reaching from defects within dimensional structures over multidimensional structures to new analytical requirements and more.

1 Introduction

A proper data model is the core of representing a part of the real world in the context of a database. The multidimensional data model ([43]) has proven extremely adequate for the explorative analysis of information stored in a data warehouse. Many variations of the multidimensional modeling idea were proposed in the recent past, extending the classical way of multidimensionally reflecting the world in different directions. However, no data model provides a comprehensive set of structural and operational tools necessary for a flexible and extensive analysis of information stored within a data warehouse system. After many years of research this paper provides a summary of open problems in the context of multidimensional modeling. From our point of view, these problems are of fundamental importance and need further investigation in the very near future, but we have to emphasize that the list of open problems surely is not complete.

The remainder of the paper identifies and discusses ten defects of current multidimensional data models. In identifying those defects we encourage the data warehousing community to develop adequate solutions to improve the service accomplished by a successful data warehousing infrastructure. In discussing the current state-of-the-art we are far from producing the ultimate solutions of these problems. However we want to pinpoint the single problems and produce a list following the same pattern every time.

The Surrounding Modeling Framework

The similarity to the design pattern approach by Gamma et al. ([11]) is not purely accidental. The goal is a modular data model extensible by plug-ins.

(0) To reflect a multidimensional application scenario properly, we need an extensible data model (i.e. a multidimensional meta model) to plug in certain modeling features, in the sense of modules, as needed.

Final edited form was published in "Data Warehousing and Knowledge Discovery.4th International Conference, DaWaK 2002. Aix-en-Provence 2003", S. 275–285,
ISBN 978-3-540-44123-6
http://dx.doi.org/10.1007/3-540-46145-0_27

Many application scenarios are well equipped with the idea of the simple multidimensional modeling. There are also many non-standard OLAP applications which would tremendously benefit from using OLAP technology. However, providing modeling techniques for every possible scenario does not seem like an adequate approach. Thus, the general concept is to provide a meta model from which application designers may instantiate a certain concrete multidimensional model with only the extensions they need for the actual scenario. This would help to develop and organize the modeling techniques in a very modular way. On the one hand, researchers are then able to provide new model extensions as simple plug-ins into the model. On the other hand, developers would have to demonstrate expertise only in those modules which they are really intending to use. Although many modeling proposals were made in the recent past, there is no well-known and widely accepted work providing a meta modeling framework. In proposing such a framework, the modular approach and the feature of extensibility of UML could be used as a guideline.

2 Ten Problems

The next ten sections deal with shortcomings of the multidimensional data model and sketches existing or possible solutions. We are focussing on dimensional, multidimensional aspects as well as meta information.

Problem: Unbalanced Hierarchies

(1) Complex applications modeled in an OLAP style require flexible classification structures. Current theory demands that every path from the generic top level node to any of the leaves has the same length. This is not always possible in practice.

Description: In general, two cases of unbalanced hierarchies are possible: an arbitrary subtree of the classification tree lacks an inner level in comparison to its siblings or it lacks the leaf level. In either case the result is an unbalanced hierarchy. Unbalanced hierarchies are not necessarily the result of a bad dimensional design. The real world may demand that kind of hierarchies; e.g. consider the product dimension of a bank or insurance company. Certain loan services are packaged and combined with other financial services, while the product of a simple savings account does not have any further subclassifications. The main problem of unbalanced hierarchies is that on a certain level of the classification tree the partitioning property is violated: it is not guaranteed that the sum of the sales figures for all nodes of a certain level really represents the total sum.

Discussion: The topic of unbalanced hierarchies is not treated in literature in detail ([22]). The simple approach is the introduction of dummy nodes: these are introduced where necessary to produce a regular, balanced hierarchy. This problem plays an important role also in some of the following problems. For example in the case of schema evolution (problem 10) aggregation over long periods of time has to deal with changing and thus unbalanced hierarchies. Further discussiuon of the topic also known as non-covering or heterogenous dimensions can be found in [15], [16], [33] and [42]

Final edited form was published in "Data Warehousing and Knowledge Discovery.4th International Conference, DaWaK 2002. Aix-en-Provence 2003", S. 275–285, ISBN 978-3-540-44123-6
http://dx.doi.org/10.1007/3-540-46145-0_27

Problem: Irregular Hierarchies

(2) Another defect of the common classification hierarchy concept is that only pure 1:N-relationships between classification nodes of adjacent levels are allowed to avoid irregularities in aggregations. Especially the product dimension demands to drop this restriction, i.e. a general acyclic classification graph is needed.

Description: Nowadays we are surrounded by complex products that fulfil the functionality of several products. A mobile phone SL 45 is not a phone only, it also is a personal digital assistant (PDA) and an MP3 player. In a strict classification hierarchy it can only be assigned to a single product family, e.g. to Mobile Phones. This 1:N-relationship is particularly important for aggregating along drill paths: only under this condition the sum of sales over all Articles is equal to the sum of sales over all Product Families (figure 1). In practice it might be interesting to also consider the SL 45 when calculating the sum of sales over PDAs or MP3 players. So it might be desirable to assign it to all three product families (lower half of figure 1). This means to give up the 1:N relationship in favour of a general N:M mapping which results in an acyclic classification graph for the dimension(s). However this approach causes a wrong result for the sum over all product families because the phone's sales figures go into the sum three times instead of only once thus yielding 200,100 instead of 200,000, the correct result.

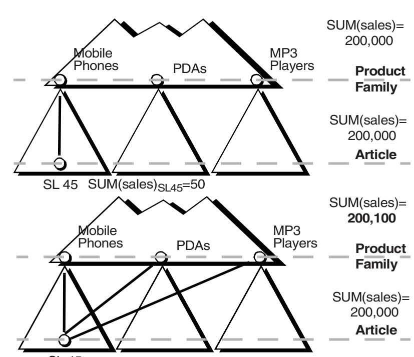

Fig. 1. 1:N Mapping vs. N:M Mapping

Discussion: Literature suggests two solutions that both are not really satisfactory ([21], [20]). The more general solution is to specify a distribution for the sales figures of the classification node in question. For example the following heuristic could be issued: 80% of all sales figures for the SL 45 are added to the Mobile Phones category and 10% to each, PDA and MP3 Players. Obviously this approach avoids the different aggregation results on different classification levels. However it only produces estimated figures, its quality depending on the quality of the distribution. The other solution is a special case of the distribution approach: the facts related to a multi-predecessor classification node are completely assigned to exactly one higher classification node (100%). Further ideas are given in [34], [40], [42].

Problem: Detailed Feature Exploration

(3) The simple notion of a classification hierarchy is not capable of reflecting complex dimensional information. Besides the hierarchical structure we demand an additional infrastructure to model complex features comprehensively describing simple classification nodes.

Description: Consider a product dimension. Simple articles are grouped to product groups, these in turn into product areas, families an so on. However, a single product does exhibit a tremendous feature list, providing extra information with regard to color,

Final edited form was published in "Data Warehousing and Knowledge Discovery.4th International Conference, DaWaK 2002. Aix-en-Provence 2003", S. 275–285,
ISBN 978-3-540-44123-6
http://dx.doi.org/10.1007/3-540-46145-0_27

packaging type, delivery information and so on. If those features may be assigned to every product in a product dimension, the classical way of multidimensional modeling may look sufficient. However, if features are local to certain classes of products, extensions to the data model are necessary. For example the set of washing machines certainly exhibits a completely different set of properties as kitchen appliances do. But both are members of the same dimension. A second issue in annotating dimensional structures consists in the problem of detailing a dimensional structure beyond the leaf nodes of a dimensional hierarchy. Again, considering a product dimension, one may wish to expand each product regarding the single parts needed to construct this specific article. These parts may either appear as products themselves in the dimension or they may reflect sub-articles which are not part of the primary dimensional structure. Obviously, strong summarizability is not required with this technique. Instead, drilling sales figures beyond the atomic level may help the analyst to gain knowledge according to relationships to other product areas.

Discussion: Addressing the first problem in annotating dimensional structures with properties local to certain classification nodes was done in [23]. Unfortunately, this work misses a seamless integration with other wishful extensions. Work considering the second issue in splitting data beyond the leaf nodes of a classification tree must be seen as a variation of many other problems described. Since a single article is made of many parts or annotated with multiple features we consider this an N:M-relationship problem (problem 2). Moreover, if we require summarizability to a certain degree, dealing with de-aggregation is necessary to come up with a reasonable solution (problem 6).

Problem: Multidimensional Constraints

(4) While the hierarchical structure of a dimension exhibits strong functional dependencies between the single category attributes, the data cells of a multidimensional data cube are not subject of any constraints beyond a given data and summarizability type. We demand an additional multidimensional constraint mechanism focussing the sparsity, i.e. the existence or absence of explicit NULL-values.

Description: A multidimensional data cube usually exhibits a high degree of sparsity, because many possible combinations of dimensional elements do not show a corresponding data entry. In the same way the relational model types and limits NULL values the multidimensional model requires (a) an additional mechanism to type the non-existent data cell values according to their meaning and (b) a constraint mechanism to explicitly allow or prohibit the existence of data values for certain combinations. An example for different types of NULL values would be the case where a difference has to be made regarding data which is known to be delivered from external data sources but has not arrived yet and data which will never appear in the data cube.

Discussion: Regarding the first issue, we may extend the domain of all possible values of a measure by two constant values, NOT_KNOWN and NOT_EXISTENT following the classical NULL value theory from the relational world ([7]). While the first value denotes a not yet available but possible NULL value, the second value gives the user (and the system!) the hint that the corresponding data entry is currently not available and will never be accessible, because this combination does not make sense in the real

Final edited form was published in "Data Warehousing and Knowledge Discovery.4th International Conference, DaWaK 2002. Aix-en-Provence 2003", S. 275–285,
ISBN 978-3-540-44123-6
http://dx.doi.org/10.1007/3-540-46145-0_27

world application. The second issue demands a way to declare multidimensional NULL constraints for sub-cubes defined over dimensional elements. In opposite to the relational model, a cell should be tagged ALWAYS_NULL to prevent any user data, NEVER_NULL to demand user data or ANY to keep the constraint unspecified. If a data cell tagged with ANY has no recorded value a NOT_KNOWN value takes place because a value may be inserted in the future. Moreover the system should have knowledge of these types so that they are excluded within an OLAP data cube and users are not permitted to explore, i.e. drill-down into those sub-cubes. Unfortunately no work in the area of multidimensional data models is aware of this problem and incorporates a solution seamlessly into the modeling framework.

Problem: Restricting Access to Multidimensional Information

(5) By integrating data from different data sources into a single data cube access restrictions must be introduced for the new (and often more valuable) information: in the static case, the data model should provide techniques to prevent certain user groups from accessing some areas of the data cube. In the dynamic case, user should not be able to gather "forbidden" data by inference applying tracker techniques.

Description: In using a data warehouse database users are suddenly able to retrieve combined and valuable information. Access mechanism compiled in the data model are required to grant access only to predefined areas of the data cube. Commercial products already restrict access to specific classification hierarchy nodes or provide only aggregated data or slices of the complete data cube. Besides these static problems, topics like inference - gathering new information from already known data - have to be considered.

Discussion: For solving the static access problem object privileges must be set up on a fine granularity, possibly on classification nodes or even on single data cells. To protect a data warehouse from inferring sensitive data we have to consider two ways to receive this sensitive data. First, there is *one-query-inference*, which generates the required information with one user query. Second, a user combines the results of multiple queries to receive the required data (audit based). This approach is called *multiple-query-inference*. Research work has been done in the area of scientific and statistical databases ([8]) but has to be adequately transferred to data warehousing. An example for a one-query-inference is to reduce the number of items of the result set by using parallel classifications, e.g. by characterizing products non-ambiguously with its feature values (problem 3). In [41] an indicator-based recognition algorithm is proposed which can be used for access control at runtime.

Problem: Missing Data

(6) While traditional analysis operators in data warehousing are defined on existing (raw) data, the application world is interested in operators to get information about areas where no data exist.

Description: Consider the following situation: Several industry companies are selling statistical data (parameterized by fact, granularity, and classification tree). Obviously, the price of data increases with the granularity and the coverage. If a customer periodically has to buy detailed data he has to spend a lot of money. The question for such a

Final edited form was published in "Data Warehousing and Knowledge Discovery.4th International Conference, DaWaK 2002. Aix-en-Provence 2003", S. 275–285,
ISBN 978-3-540-44123-6
http://dx.doi.org/10.1007/3-540-46145-0_27

customer is as follows: Is it enough to get detailed data only every other period and estimate the data of the missing periods with the risk of missing some important deviations from what is normal? If the application is satisfied with such a strategy, the data model should be able to provide tools to estimate, i.e. substitute missing data cubes.

Discussion: One strategy to estimate missing data is to use interpolation between the known detailed data of two or more periods. The disadvantage of this idea is obvious: The missing data can only be computed retrospectively. Another technique might be *deaggregation*: Usually, the drill-down operator may be applied, only if detailed data is available. If the data does not exist then the deaggregation function splits aggregated values and generates detailed data according to a predefined pattern: If an equal distribution of the data is assumed then the deaggregation function divides the aggregated value by the number of the children of this node. A more complex strategy is the usage of a *distribution pattern*: If the distribution of the data in another period is known then the same distribution can be applied to the aggregated value of the current period. For example the percentage distribution of sold video recorders in Germany in January 2001 can be used to compute the distribution in January 2002. Furthermore, if data from the preceding months are known then the data of January 2002 can be estimated by trend exploration. Related work can be found in [3], [26].

Problem: Sequence Operations

(7) The classification nodes of a dimension reflect the idea of repetitive grouping to build a classification hierarchy. The ordering of classification nodes within a dimension according to data values from inside of the data cube is not known yet.

Description: The need for sequence based analysis techniques is impressively shown by recent developments of SQL. Operators like OVER() with its ORDER BY or WINDOWING extensions enable the formulation of queries for cumulating sum or moving average values, whereby schema and fact data have to be considered in a different way. Although these operations are not yet fully standardized, the problems arising from sequence oriented queries in the relational area are well studied and many concepts are published (e.g. [35]) or already implemented in commercial database systems or in OLAP-Tools. However, especially the implementations are quite proprietary and general concepts in the multidimensional data model are missing. The dimensional attributes of one dimension form a set which is characterized by having no order. A first level of ordering dimensional data is to rely on the domain of the elements. Time for example already exhibits a natural order. Persons may be sorted according to their name or additional features. A second level of ordering may be seen in relying on data from the data cube. For example, salesmen could be ordered according to their profit, whereat profit is a regular multidimensional fact. Finally the fact data itself could be ordered for some reason.

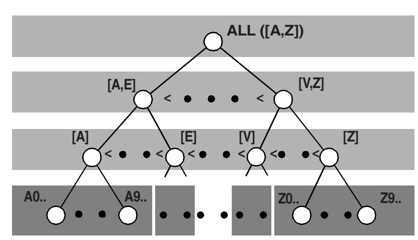

Fig. 2. Inheritance of a total order

Final edited form was published in "Data Warehousing and Knowledge Discovery.4th International Conference, DaWaK 2002. Aix-en-Provence 2003", S. 275–285,
ISBN 978-3-540-44123-6
http://dx.doi.org/10.1007/3-540-46145-0_27

Discussion: If an order is defined on a specific level of the classification hierarchy, then the order of a classification level is inherited to all elements being on a lower level. Figure 2 illustrates the inheritance. A major problem are holes in one level: If holes exist, then it is impossible to classify new elements. Figure 2 illustrates the problem, if all elements of the interval [F;U] are not members of the classification hierarchy although the parent node is indicated by [A;Z]. Beside the ordering of the classification hierarchy it is also possible to order the fact data. While the order of the first one is given by the modeler and modifications affecting the order are rare, the fact data are updated quite often. Therefore, to order the elements after each update to support only a small number of operations makes no sense. Therefore, the fact data are ordered on demand. The basis is a classification tree, which is generated ad hoc, why we call this operation *ad hoc classification*. In the same way as [35] defines relational sequence operators for selection, projection or aggregation on sequences or for concatenating and shifting have to be defined in the multidimensional data model as well.

Problem: Progressive Query Answering

(8) While almost everything is said about cleaning, scrubbing and integrating data from multiple data sources into a single consistent data warehouse database, very little work has been performed discussing the problem of approximate answers.

Description: One of the major goals and biggest problems in data warehousing is to support real-time OLAP. In the last decade, several strategies like indexing ([30]), join optimization, or preaggregation ([13], [1]) were developed. However, running a query might still take a long time. The goal of query processing is always to produce 'exact results', which is worth to take a closer look at: Is it really important to get the exact result or is it sufficient to compute a quick result, which comes close to the exact result?

Discussion: The problem of fuzzy query answers is well known in the area of statistics ([26]), where sampling is one of the basic methods. This idea could be applied to data warehousing to get a first estimation of the result on existing data, e.g. only a fraction of the fact table is read, aggregated and the result extrapolated to get an appraised value of the exact one. The main advantage of this technique is that a fast preliminary result is computed and presented to the user in a first step, while the computation of the exact result can be executed in a traditional manner in a second step. The difference betweeen the approximated and the correct value depends on the sampling technique, which has to consider the distribution of the data. Another alternative is to compute the exact value of one partial sum and extrapolate the value to get an appraised value of the result which can also be dynamically refined. After computing the first partial sum, a second partial sum is computed, a new appraised value is generated and presented to the user. This proceeds until all partial sums are calculated which corresponds to the exact value. The main advantage of the sketched idea is that in each step the user gets an estimated value which is based on more exact partial sums. The incremental characteristic is called *online aggregation* which is introduced by [18] in the relational context. Recently [32] presented the idea of Iterative Data Cubes, which is a special kind of pre-aggregation for online aggregation by handling different dimensions independently.

Final edited form was published in "Data Warehousing and Knowledge Discovery.4th International Conference, DaWaK 2002. Aix-en-Provence 2003", S. 275–285,
ISBN 978-3-540-44123-6
http://dx.doi.org/10.1007/3-540-46145-0_27

Problem: Modeling Metadata Information

(9) To achieve the aim of data warehouse systems, the analysis of integrated data, meta data are essential in order to interpret the results and benefit from the expense of setting up a data warehouse. Proposals for standards of a common meta schema have been made but have not yet got accepted.

Description: If users cannot understand and interpret the results of OLAP queries, the acceptance and benefit of the whole data warehouse is suffering. An effective data consolidation can be realized only with support of well organized and structured meta data. To enable different systems to interact with each other a standardized API to process the meta data and a common conceptual schema or a data exchange format is required. A consistent exchange format not only enables the exchange of data directly between cooperating data warehouse systems but also may serve as a base to distribute multidimensional organized data over the web by a third party provider. If data can be interpreted, downloading and integrating data cubes in a local data warehouse would no longer be a vision.

Discussion: The common warehouse metamodel (CWM, [6]) has been set up by the OMG which includes in the meantime the Open Information Model (OIM, [31]). But it has not become that widely accepted and used as it is desirable. This is due to the lack of support for some applications. A common, standardized meta data framework has to be both flexible and detailed. An imprecise meta schema has no benefit and yields to desiderative utilization. On the other hand it has to be adaptable to a broad range of applications, otherwise it misses an effective support for these and usage is limited. A possible way to reach this aim is a plug-in mechanism to get a customizable meta schema. A comprehensive meta modeling has also to consider data exchange. An XML based encoding of multidimensional data with the purpose of data exchange is shown in [29]. Tightly connected to this topic is a query definition standard. A query formulated at one system must be transferred to another one, executed and the result passed back to origin warehouse. Some work has been done in this area ([27], [26]).

Problem: Schema Evolution

(10)The multidimensional schema may change in the course of time. Examples for modifications of the schema are the insertion of a new classification node or the deletion of a classification level. The schema has to reflect changes in the real world like introduction of new products or the modification of structure of the channels of distribution.

Description: In general one may distinguish between schema evolution and schema versioning. Schema evolution means that the data is adapted to the new structure and the old schema will be lost, whereas schema versioning retains the schemata for the according validity period. This enables evaluation of the data based on an arbitrary structure, the currently valid schema, the schema of a past point in time or the correlating schema of data creation. Schema evolution causes fewer problems as data is transformed into the new structure and afterwards query processing is just the same as in a regular data warehouse. Schema versioning requires more complex solutions.

Final edited form was published in "Data Warehousing and Knowledge Discovery.4th International Conference, DaWaK 2002. Aix-en-Provence 2003", S. 275–285, ISBN 978-3-540-44123-6
http://dx.doi.org/10.1007/3-540-46145-0_27

Discussion: When storing data with its schema changes additional structures to represent the temporal aspects are necessary. One possibility is to introduce new attributes containing the time stamps for the begin and the end of the valid time. But the increased complexity of the storage schema entails an unsatisfying query performance. New concepts of query processing are necessary. Also preaggregation becomes more difficult as with schema historization another analysis dimension has to be considered. Furthermore multidimensional indices which support the evaluation of data according to different schemata are conceivable. At the user front end there is a lack of support by query languages and OLAP tools. For the relational interface e.g. TSQL, TSQL2 or TQUEL ([38]) have been developed. For further work see [16], [21].

3 Summary and Conclusion

This paper is meant to be a guide through still open problems of multidimensional modeling in data warehousing. The data warehouse is a well understood and well established technique in modern business. It reflects the core of OLAP, decision support, CRM, etc. We subject ourselves to the multidimensional data model because it enables us to efficiently follow predefined evaluation paths. Still there are several situations in real world business, that are not solved satisfyingly. The bottom line of this paper is that we have to accept that there will never be the one and only multidimensional data model. Instead we have to develop a catalogue of data warehouse patterns: whenever a problem arises the catalogue provides a detailed description of this problem and discusses the solutions at hand.

References

1 Baralis, E.; Paraboschi, S.; Teniente, E.: Materialized View Selection in a Multidimensional Database. VLDB '97, Athens, Greece, 1997
2 Brodsky, A.; Farkas, C.; Jajodia., S.: Secure databases: Constraints, inference channels, and monitoring disclosures. IEEE Trans. Knowledge and Data Engineering, 12(6):900–919, 2000.
3 Chaudhuri, S; Das, G., Narasayya, V.: A Robust, Optimization-Based Approach for Approximate Answering of Aggregate Queries. ACM SIGMOD, Santa Barbara, USA. 2001.
4 Chen, P. et al: Evaluating Aggregate Operation over Imprecise Data. IEEE Trans. on Knowledge and Data Engineering, 8(2):273-284, 1996.
5 Codd, E.F.: Codd, S.B.; Salley, C.T.: Providing OLAP (On-line Analytical Processing) to User Analysts: An IT Mandate, White Paper, Arbor Software Corporation, 1993
6 N.N.: Common Warehouse Metamodel specification. OMG (http://www.omg.org/cwm/)
7 Date, C.J.: An Introduction to Database Systems. 7th ed. Reading, Massachusettes, 2000
8 Denning, D.E.: Secure statistical databases with random sample queries. ACM Trans. on Database Systems, 5(3):291–315, 1980.
9 Dyreson, C. E.: Information Retrieval from an Incomplete Data Cube. VLDB'96, Bombay, 1996.
10 Gray, J.; Bosworth A.; Layman A.; Pirahesh, H.: Data Cube: A Relational Aggregation Operator Generalizing Group-By, Cross-Tab, and Sub-Total. ICDE´96, New Orleans, 1996

Final edited form was published in "Data Warehousing and Knowledge Discovery.4th International Conference, DaWaK 2002. Aix-en-Provence 2003", S. 275–285,
ISBN 978-3-540-44123-6
http://dx.doi.org/10.1007/3-540-46145-0_27

11 Gamma, E.; Helm, R.; Johnson, R.; Vlissides, J.: Design Patterns – Elements of Reusable Object-Oriented Software. Boston et al., 1994
12 Gupta, A.; Mumick, I.: Maintenance of Materialized Views: Problems, Techniques, and Applications. In: IEEE Data Engineering Bulletin, Special Issue on Materialized Views & Data Warehousing 18(1995)2
13 Harinarayan, V.; Rajaraman, A.; Ullman, J.D.: Implementing Data Cubes Efficiently. SIGMOD ’96, Montreal, Canada, 1996
14 Hurtado, C.; Mendelzon, A.: Reasoning about Summarizability in Heterogeneous Multidimensional Schemas. ICDT’01, 2001.
15 Hurtado, C.; Mendelzon, A.: OLAP Dimension Constraints. PODS’02, Madison, 2002.
16 Hurtado, C.; Mendelzon, A.; Vaisman, A.: Maintaining data cubes under dimension updates. ICDE’99
17 Jagadish, H. V.; Lakshmanan, L. V. S.; Srivastava, D.: What can Hierarchies do for Data Warehouses? VLDB’99, Bombay, 1999.
18 Joseph M. Hellerstein, Peter J. Haas, Helen Wang: Online Aggregation. SIGMOD Conference 1997
19 N.N.: Hyperion Essbase OLAP Server, Hyperion Software Corporation, 1999 (http://www.hyperion.com/essbaseolap.cfm)
20 Inmon, W.H.: Building the Data Warehouse, New York, John Wiley & Sons, 1996
21 Kimball, R.: The Data Warehouse Toolkit, New York, John Wiley & Sons, 1996
22 Lehner, W., Albrecht, J., Wedekind, H.: Multidimensional Normal Forms. SSDBM'98, Capri, Italy, 1998
23 Lehner, W.; Ruf, T.; Teschke, M.: CROSS-DB: A Feature-extended multi-dimensional Data Model for Statistical and Scientific Databases. CIKM'96, Rockville (MD), U.S.A., 1996
24 Lenz, H.-J.; Shoshani, A: Summarizability in OLAP and Statistical Data Bases. SSDBM’97, Olympia
25 Malvestuto, F.M.; Moscarini, M.: Computational issues connected with the protection of sensitive statistics by auditing sum-queries. IEEE SSDM’98, 1998.
26 Mason, R.: Statistical Design and Analysis of Experiments: With Applications to Engineering and Science (Wiley Series in Probability and Mathematical Statistic), John Wiley & Sons, 1989
26 N.N.: OLAP Council API, MDAPI version 2.0 specification. OLAP Council (http://www.olapcouncil.org/research/apily.htm)
27 N.N.: The Multidimensional Expressions language (MDX). Microsoft Corporation (http://msdn.microsoft.com/library/default.asp?url=/library/en-us/olapdmad/agmdxbasics_04qg.asp)
28 N.N.: Microstrategy 6, MicroStrategy Inc., 1999, (http://www.microstrategy.com/Products/index.asp)
29 N.N.: XCube, 2002, (http://www.xcube-open.org)
30 O'Neil, P.: INFORMIX and Indexing Support for Data Warehouses. In: Database Programming and Design, v. 10, no. 2, February, 1997
31 N.N.: Open Information Model, Meta Data Coalition, (http://www.mdcinfo.com/)
32 Mirek Riedewald, Divyakant Agrawal, Amr El Abbadi: Flexible Data Cubes for Online Aggregation. ICDT’01, London, United Kingdom, 2001
33 Pedersen, T. B.; Jensen, C. S.; Dyreson, C. E.: Supporting Imprecision in Multidimensional Databases Using Granularities. SSDBM’99, 1999.

Final edited form was published in "Data Warehousing and Knowledge Discovery.4th International Conference, DaWaK 2002. Aix-en-Provence 2003", S. 275–285,
ISBN 978-3-540-44123-6
http://dx.doi.org/10.1007/3-540-46145-0_27

34 Pedersen, T. B.; Jensen, C. S.; Dyreson, C. E.: A foundation for capturing and querying complex multidimensional data. Information Systems, 26(5): 383-423, 2001.

35 Praveen Seshadri, Miron Livny, Raghu Ramakrishnan: The Design and Implementation of a Sequence Database System. VLDB'96, Mumbai (Bombay), India, 1996

36 Shoshani, A.: OLAP and Statistical Databases: Similarities and Differences, PODS'97, Tuscon, Arizona

37 Smith, J.M.; Smith, D.C.P.: Database Abstractions: Aggregation and Generalization, ACM Transactions on Database Systems 2(1977)2

38 Snodgrass, R. T. (Hrsg.): The TSQL2 Temporal Query Language. Kluwer Academic Publishing, Amsterdam, 1995

39 N.N.: ISO/IEC 9075: 1999, Informatik Technology – Database Languages – SQL, 1999

40 Song, I.; Rowan, B.; Medsker, C.; Ewen, E.: An Analysis of Many-to-Many Relationships Between Fact and Dimension Tables in Dimensional Modeling. DMDW'01, 2001.

41 Steger, J.; Günzel, H.; Bauer, A.: Identifying Security Holes in OLAP Applications. IFIP WG 11.3 Conference on Database Security, Schoorl, Netherlands, 2000

42 Tsois, A.; Karayiannidis, N.; Sellis, T.: MAC: Conceptual data modeling for OLAP. DMDW'01, 2001.

43 Vassiliadis, P.; Sellis, T.: A Survey of Logical Models for OLAP Databases. In: SIGMOD Record 28(1999)4