Construction Of Drug Company Data System Using Multidimensional Database Insertion

Dr. Rajaaaldeen Abad Khalid
Al-Nahrain University.
Baghdad, Iraq.
Rarajaaalden9@gmail.com

Hind Khalid. Hameed
Al-Nahrain University.
Baghdad, Iraq.
Hind.1969@yahoo.com

Abstract :

A multidimensional database is a data warehouse that supports basically the efficient execution of complex business decision queries. Every response can be significantly improved by storing an appropriate set of Appear views of the data. These views are selected from the multidimensions whose elements represent the solution space of the problem. Techniques discussed in this paper can be implemented using relational database technology. The approach is also capable of exploiting multidimensional query processing, Techniques like aggregation, this yields a practical solution with low computational overhead,In this paper. Techniqueswe propose multidimensional databases. Dimensions, dimension hierarchies and cubes are formally introduced , furthermore; we provide a mapping of the multidimensional model to the relational model and to multidimensional arrays.the results shows much progressing speed of data insertion , decision capability , and cost reduction as well comparing with traditional data insertion.

Keywords: multidimensional, data warehouse, views, multidimensional arrays.

انشاء منظومة بيانات لشركة أدوية باستخدام الأدخال متعدد الأبعاد للبيانات

د. رجاء الدين عبد خالد
جامعة النهرين
بغداد، العراق
Rarajaaalden9@gmail.com

م.م. هند خالد حميد
جامعة النهرين
بغداد، العراق
Hind.1969@yahoo.com

الخلاصة :

البيانات متعدد الأبعاد هو مخزن للبيانات يدعم بشكل أساس كفاءة اتخاذ القرارلتنفيذ الأعمال المعقدة .كل استجابة يمكن ان تبرهن بشكل ملحوظ بواسطة خزن مجموعة مناسبة من المنظورات المظهرية للبيانات . هذه المنظورات مختارة من ابعاد متعددة والتي عناصرها تمثل فضاء الحلول للمشكلة . التقنيات التي تمت مناقشتها في هذا البحث يمكن أن تنفذ بأستخدام تقنية قاعدة البيانات المترابطة . المقاربة قابلة أيضا لأستكشاف المعالجات متعددة البيانات ، التقنيات مثل تجمعات البيانات ، تعطى حلولا عملية مع حوسبة واطئة التكلفة.التقنيات التياقترحناها وتم تطبيقها هي قواعد البيانات متعددة الأبعاد. ألأبعاد ، تدرجات الأبعاد ، والمكعبات تم تصميمها وتقديمها علاوة على وضع الخارطة

لأنموذج الأدخال متعدد الأبعاد لحالة البحث و المصفوفات متعددة الأبعاد . وقد أظهرت النتائج تقدما أعلى في سرعة ادخال البيانات وقابلية صنع القرار و تقليل في التكاليف أيضا بالمقارنة مع نظم ادخال البيانات التقليدية .

1. Introduction :

The relational data model, which was introduced by Codd in 1970 and earned him the Turing Award a decade later, constitutes a significant part of the foundation of today's multi-billion-dollar database industry. During the 1990s, a new type of data model, the multidimensional data model, emerged that has since taken over from the relational model when the objective is to analyze data, rather than to perform on-line transactions.

Multidimensional data models are designed expressly to support data analyses. A number of such models have been proposed by researchers from academia and industry. In Academia, formal mathematical Models have been proposed, while the industrial proposals have typically been specified more or less implicitly by the concrete software tools that implement them. [1]

In recent database trends, data warehouses come to fill a gap in the field of querying large, distributed and frequently updated systems. Most researchers and Developers share the same general vision of what a data warehouse is [2], [4]. Data are extracted from several data sources, cleansed, customized and inserted into the data warehouse. The logical structure and semantics of the data, or else Enterprise Model, is stored in an Information Directory. Next, the data warehouse data can be filtered, aggregated and stored in smaller specialized data stores, usually called data marts. Users query the data marts and/or the data warehouse, mostly through On Line Analytical Processing (OLAP) applications. The main characteristics of such applications are (a) multidimensional view of data, and (b) Data analysis, through interactive and/or navigational querying of data [3].

A multidimensional database (MDDB) is a data repository that provides an integrated environment for decision support queries that require complex aggregation on huge amounts of historical data. An MDDB is a relational data warehouse, in which the information is organized following the so-called star-model [5] .

Each dimensions table contains all the information that is specific only to the dimension itself, while the Fact table correlates all dimensions and contains information on the attributes of interest for the intersection of all the dimensions. A new operator, the data-cube operator [3], has been proposed to perform the computation, on a single relation (the fact table), of one or more aggregate functions for all possible combinations of grouping attributes (which are the elements of the data-cube).

Many of these products suffer from the following limitations: (i) they are ad-hoc and they do not support a comprehensive “query” language similar to SQL; (ii) the user interaction is often limited to one operation at a time; (iii) viewing data in multi-dimensional perspectives involves treating certain attributes as dimensional parameters and the remaining ones as measures, and then analyzing them as a “function” of the parameters; many products treat dimensions and measures asymmetrically; and, finally,(iv) unlike for the relational model, there is no precise, commonly agreed, conceptual model for OLAP or the so-called multi-dimensional databases (MDD).Much of the success of relational databases has to do with the clear logical foundations for the data model first laid down by Codd and developed by numerous researchers subsequently. [6]

In this paper, the main impulse of this paper is to provide an application by used formal model for multidimensional databases. Since multidimensional databases are defined in terms of dimensions (which are organized in dimension (hierarchies), the model represents them formally. We also provide a mapping to relational databases and multidimensional arrays. We make a serious design choice: since querying is done in an interactive way, we give importance to the tracking of series of operations.

2. Related Work :

Research has followed the development of industrial products in the field of OLAP, data warehouses.

In [7], a model for multidimensional databases is introduced. The model is characterized from its symmetric treatment of dimensions and measures. A set of minimal (but rather complicated) operators is also introduced dealing with the construction and destruction of cubes, join and restriction of cubes and merging of cubes through direct dimensions. Furthermore, an SQL mapping is presented.

In [8], in this paper, we investigate how the multi-token list affects the performance of database proximity search. Numerous experiments have been conducted and the results show that two-token adjacent token list can achieve the best query performance in multi-token list based proximity search.

In [9], we address the issue of designing effective query languages for OLAP databases. The basis of our investigation is MD, a logical data model for multidimensional databases that, unlike other multidimensional models, is independent of any specific implementation. As such, MD provides a clear separation between practical and conceptual aspects of multidimensional analysis. In this framework, we present and compare various OLAP query languages, based on different paradigms suitable for distinct users.

In [10] , previous work, we studied the problem of inferring summarizability in a particular class of heterogeneous dimensions. In this paper, we propose a class of integrity constraints and schemas that allow us to reason about summarizability in general heterogeneous dimensions. We introduce the notion of frozen dimensions, which are minimal homogeneous dimension instances representing the different structures that are implicitly combined in a heterogeneous dimension. Frozen dimensions provide the basis for efficiently testing implication of dimension constraints, and are useful to understanding heterogeneous dimensions. We give a sound and complete algorithm for solving the implication of dimension constraints, that uses heuristics based on the structure of the dimension and the constraints to speed up its execution. We study the intrinsic complexity of the implication problem, and the running time of our algorithm.

3 . A Practical Application :

The basic components in the formation of a multi-dimensional model, made us rely on the analysis of data to describe factual data according to appropriate dimensions, in a data warehousing application for a retail company it is useful to organize data along dimensions such as products commercialized by the company, stores selling these products and days on which sales occur.

Company sales of medicines and medical supplies (The company has branches in all cities where savers with a store to sell medicines and medical supplies).The goal of the project is a statement the company's sales volume by a certain date and
Also show any product buying power of many.

Another important project goal could be the analysis of the warehouse process, where inventory levels should be measured monthly, for each product and warehouse controlled by the company. It follows that possible dimension of the Company sales of medicines and medical supplies .All data warehouse application is Product, Store, Warehouse and Time. The Product dimension may be organized into levels such as item (type_product, code_product, line_product, Manufacturer, price, datasall, overgsall, contsall, city_name, …).

Can be organized into levels, the elements of the Time dimension describe days over a period of time; this dimension may be organized into the levels day, month and year.
Represents dimensions for sales of medicines and medical supplies company, as described in project

3.1 Definitions of Dimensions.

Is a collection of relation (see figure (1).*D1,…,Dn*, F, where
Di: - is a dimension table that relation characterized by an identifier *di* that uniquely identifies each tuple *(di* is the primary key of *Di*).
F :- is a fact table that relation connecting all tables*D1,…,Dn*; the identifier of *F* is given by the foreign keys *d1,…,dn* of all the dimension tables it connects; the schema of *F* contains a set of additional attributes V(representing the values on which the aggregate functions are applied).

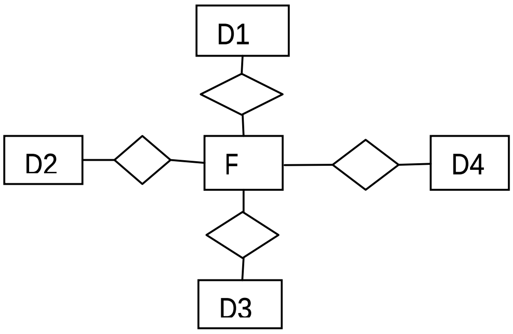


Fig. (1) planning for display the relation between dimensions.

3.2 Indexes - B-Tree Indexes

The B-tree, or balanced tree, is the most common type of index all database servers and embedded database libraries offer B-tree indexes, often as the default index type. They are usually the default because of their unique combination of flexibility, size, and overall good performance. A B-tree is a tree structure. The nodes are arranged in sorted order based on the key values. A B-tree is said to be balanced because it will never become lopsided as new nodes are added and removed. The main benefit of this balance is that the worst-case.

Performance of a B-tree is always quite good. Unlike binary trees, in which each node has at most two children, B-trees have many keys per node and don't grow "tall" or "deep" as quickly as a binary tree. See **figure (3)**

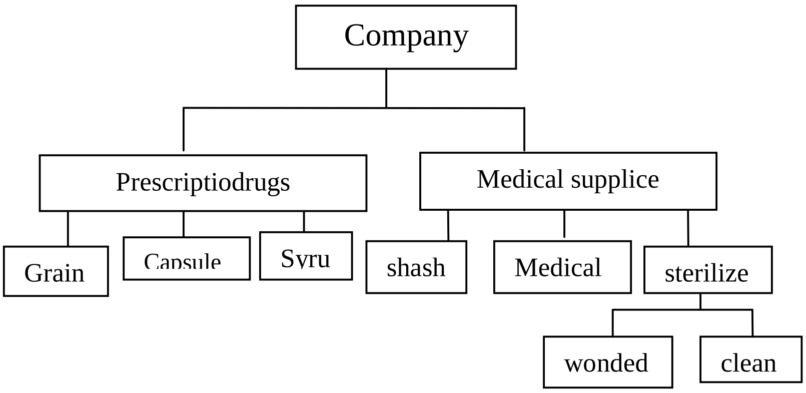
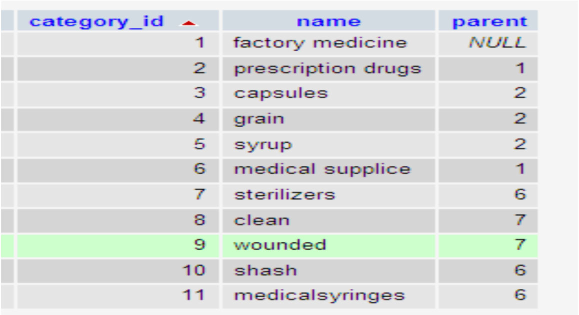


Fig. (2) represent by b-tree

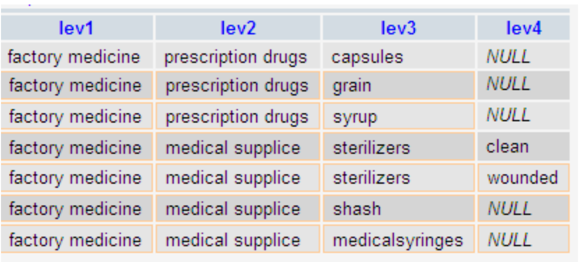
First steps represent the **figure (2)** by table using mysql, insert the content the tree, and write query to show the result (figure 3 show table)



category_id	name	parent
1	factory medicine	NULL
2	prescription drugs	1
3	capsules	2
4	grain	2
5	syrup	2
6	medical supplice	1
7	sterilizers	6
8	clean	7
9	wounded	7
10	shash	6
11	medicalsyringes	6

Fig. (3) show table

Second steps write query to result level tree (figure 4 show table)



lev1	lev2	lev3	lev4
factory medicine	prescription drugs	capsules	NULL
factory medicine	prescription drugs	grain	NULL
factory medicine	prescription drugs	syrup	NULL
factory medicine	medical supplice	sterilizers	clean
factory medicine	medical supplice	sterilizers	wounded
factory medicine	medical supplice	shash	NULL
factory medicine	medical supplice	medicalsyringes	NULL

Fig. (4) show table

3.4 View: Queries

The multidimensional database in this paper that content dimension Table (Product, Store, Time, F),where each dimension table has its corresponding attributes and with fact table F=(s,d,r,f),having s,d (time dimension is represent with the details of day)and r as foreign keys of the dimension tables and F representing the amount of sales .

The following queries can be requested on the paper:

- q1=total sales per product
- q2=total sales per product and store
- q3=total sales per product and day
- q4=total sales per product, store and day.

The queries we consider are select-join group by queries, with some restriction on the allowed select and join predicates between a dimension attribute and a constant value. Where the user provides a set of queries representative of the query that the system must be answer. Query q can be an sewed by accessing the result of a query that returns the sales group by product and store, and select from them only the tuples relative to the specified store. The restriction that all the attributes appearing in selection predicate must also appear as grouping attributes.

Join operations may be performed only between fact table and any of its dimensions. Allowed predicates are equality predicate between a dimension's identifier and the corresponding fact table foreign key, while join predicates involving non-key attributes are disallowed.

In this paper used faction found in mysql example (count, sum, avg, min, max).Grouping attributes may be drawn both from dimension and fact table attributes. In this paper used multi dimension array for view the result query because this make it very easy to access or retrieve data according to the specific data type we are concerned in, and it makes data browsing and manipulation highly intuitive to the end-user. The result table in figure (5) came through three dimensions that product, store and time, and relation with the F dimension. Show query, when the Manager General of the company to see a certain number of savers sales in a particular region and a certain date.

Medicine	Type-product	Manufacturing	No.sales
Aspirin	Child	Iraqi	3000
Aspirin	Older	German	6000
Bandol	Syrup	Indian	3000
Bandol	Copsules	Turkish	6000
Asbejk	Needl	Syrain	7000
Asbejk	Grain	Iraqi	5000

Fig. (5) multidimensional array

3.5 View: cube

In this paper we can use cube by receiving as input a table, a set of aggregating attributes and a function, computes the union of the results of the queries evaluating function, having as grouping attributes all possible combination of attributes.

When write query that result number of Medicine in all store from dimension the result that view by cube. (Show figure (6))

Medicine	store	Number
Aspirin	Baghdad	18
Aspirin	mauls	9
Aspirin	NULL	27
Bandol	Baghdad	12
Bandol	mauls	5
Bandol	NULL	17
Asbejk	Baghdad	27
Asbejk	mauls	15
Asbejk	basrah	10
Asbejk	null	52

Fig. (6) cube

4. Conclusions

In this paper, we apply the materials and the principles of multi-dimensional application through analysis and programming work of accompany that sells multiple products in multiple places, Development of the company's work by increasing the speed of data retrieval and also to view the sales company distributed in different places and see the most requested product, and you make this project using system wamp server (phpmysql) .In this system as simply and also through this system we can contact web that can work many form by web.

Since all programs in a multi-dimensional written using language (sql language) where we found how the process added and quickly retrieve the data is through this paper conclude that can basic principles applied multi-dimensionalinany appropriate language with any application we want.

6. References

1. **R. Agrawal, A. Gupta, and S. Sarawagi. “Modeling ultidimensional databases”. In ICDE, pages 232{243, 1997}.**

2. **J. Widom, "Research Problems in Data Warehousing", Proc. of 4th CIKM Conference, 1995**
3. **DWQ, "Deliverable D1.1, Data Warehouse Quality Requirements and Framework", NTUA, RWTH,INRIA, DFKI, UNIROMA, IRST, DWQ TR DWQ -NTUA - 1001, 1997, available at http://www.dbnet.ece.ntua.gr/~dwq/**

4. **S. Chaudhuri, U.Dayal, "Data warehousing and OLAP for Decision Support", Tutorials of 22ndVLDB Conference, 1996.**
5. **R.Kimbell. “The Data Warehouse Toolkit”. John Wiley&Sons, 1996.**

6. **Gray,.,Bosworth.A.,Layman,A.,and Pirahesh, H.”Datacube:a relational aggregation operator generalizing group-by,crosstable and subtotals”, Proceedings of ICDE ’96, New Orleans,LA,Feb.1996.**
7. **R. Agrawal, A. Gupta, S. Sarawagi, "Modeling Multidimensional Databases", IBM Research Report, IBM Almaden Research Center, September1995.**

8. **T. Li, H. Shen, and A. M. Rosequist. “Token List Based Data Searching in a Multi-Dimensional Massive database”. In proceeding of DMI 2008.**
9. **L. Cabbibo, R. Torlone, "A Logical Approach to Multidimensional Databases", EDBT 1998.**

10. **C. Hurtado and A. “Mendelzon.Reasoning about Summariz ability in Heterogeneous Multidimensional Schemas”. In *8th International Conference on Database Theory* (ICDT), Lecture Notes in Computer Science 1973, Springer-Verlag, pages 375–389, 2001.**