



## Analysis of Data Mining by Drawing OLAP Cube Models

**Dr. P. C. Agrawal**

Visiting Professor, Mewar University  
Gangrar, Chittorgarh, Rajasthan, India

**Aparna Chaturvedi**

Research Scholar, Department of Computer Science  
Mewar University, Gangrar, Chittorgarh, Rajasthan, India

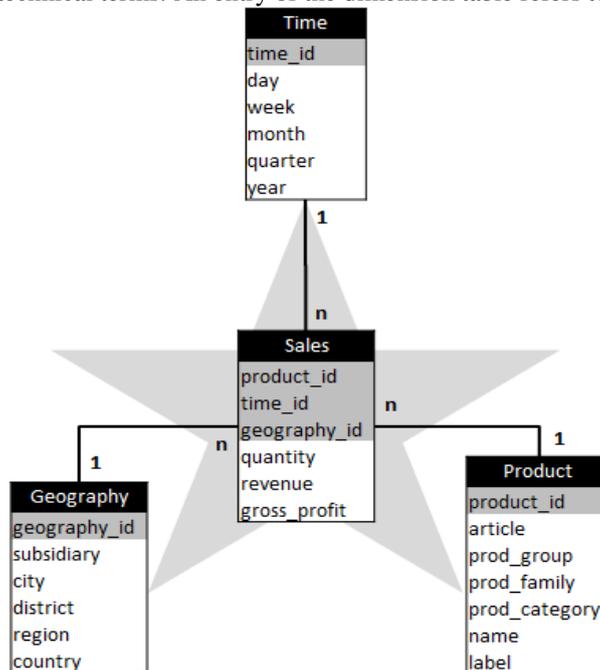
**Abstract-** In Analysis of Data Mining we study the diagrams drawn by UML Models. All type of diagrams are drawn by OLAP Cube Models. We are analyzing Production database. In Big Organization it is difficult to manage database. So by using multidimensional modeling we draw OLAP cube models. OLAP Cube models are describing the whole process in a diagrammatical way. Different steps are taken for making and finalizing Cubical Models. These Cubical Models are easy to understand and put the complicated database in meaningful manner.

**Keywords---** Data Mining, OLAP Cube Model, Query Processing, UML Model, Schema.

### I. INTRODUCTION

Data mining, the extraction of hidden predictive information from large databases, is a powerful new technology with great potential to help companies focus on the most important information in their data warehouses. Data mining tools predict future trends and behaviors, allowing businesses to make proactive, knowledge-driven decisions. The automated, prospective analyses offered by data mining move beyond the analyses of past events provided by retrospective tools typical of decision support systems. Data mining tools can answer business questions that traditionally were too time consuming to resolve. They scour databases for hidden patterns, finding predictive information that experts may miss because it lies outside their expectations.

As becomes obvious in below picture, the dimension tables (Time, Geography and Product) are grouped around the central fact table (Sales) as points of a star are grouped around its center. There is no additional table hierarchy as usual in standard databases in a star schema, causing both low complexity and high performance. Typically the dimension tables have one primary key each, which is referenced by the fact table, resulting in one foreign key per dimension table. The combination of these foreign keys is the unique id of a fact, thus comprising the primary key of the fact table. The relationship between the fact and dimension tables is n:1. A fact, in this case a sale, will always refer to just one specific product –a physical piece, to be precise- being sold in just one location at just one point in time. In technical terms: An entry of the fact table always refers to just one entry of each dimension table. On the other hand, a certain product is not sold just once, but many times, in many locations and at many points in time. Thus, a product is surely affected by multiple sales transactions – in technical terms: An entry of the dimension table refers to multiple entries of the fact table.



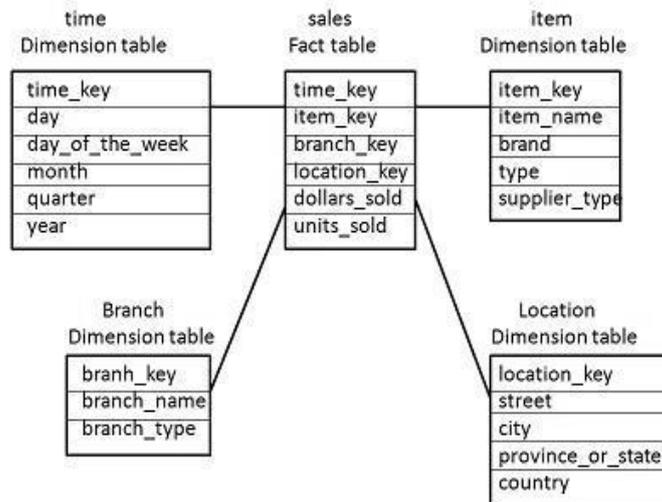
Sales and Production Table

### II. SCHEMA

Multidimensional schema is defined using Data Mining Query Language (DMQL). The two primitives, cube definition and dimension definition, can be used for defining the data warehouses and data marts.

**A) Starschema**

- Each dimension in a star schema is represented with only one-dimension table.
- This dimension table contains the set of attributes.
- The following diagram shows the sales data of a company with respect to the four dimensions, namely time, item, branch, and location.
- There is a fact table at the center. It contains the keys to each of four dimensions.
- The fact table also contains the attributes, namely dollars sold and units sold.

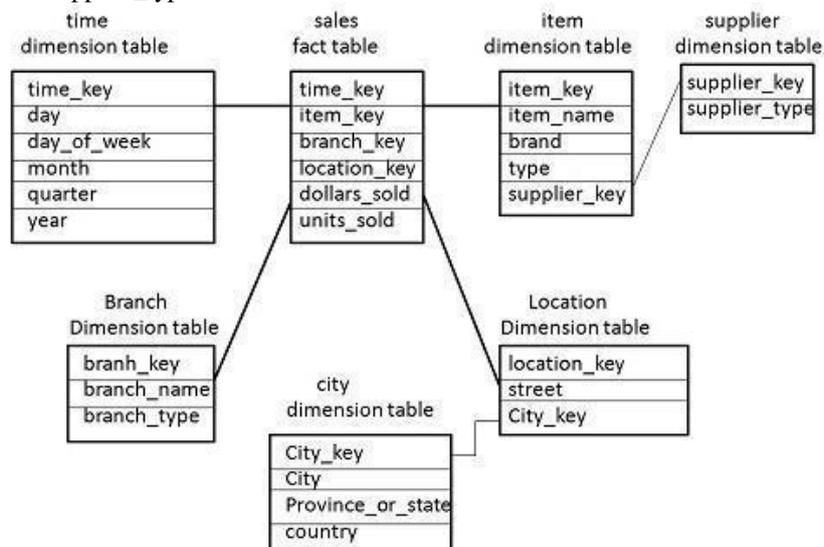


**Star Schema**

**B) Snowflake Schema**

Some dimension tables in the Snowflake schema are normalized.

- The normalization splits up the data into additional tables.
- Unlike Star schema, the dimensions table in a snowflake schema are normalized. For example, the item dimension table in star schema is normalized and split into two dimension tables, namely item and supplier table.
- Now the item dimension table contains the attributes item\_key, item\_name, type, brand, and supplier-key.
- The supplier key is linked to the supplier dimension table. The supplier dimension table contains the attributes supplier\_key and supplier\_type.



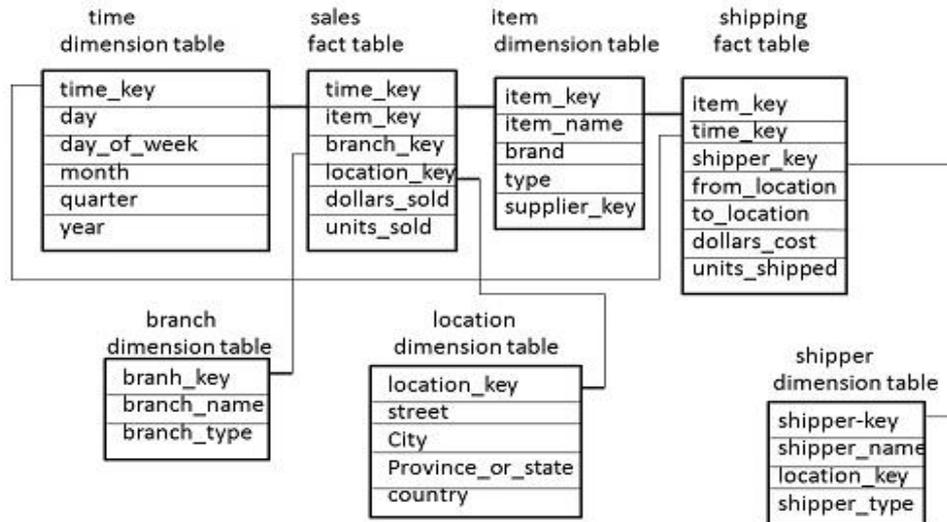
**Snowflake Schema**

**C) Fact Constellation**

A fact constellation has multiple fact tables. It is also known as galaxy schema.

- The following diagram shows two fact tables, namely sales and shipping.
- The sales fact table is same as that in the star schema.

- The shipping fact table has the five dimensions, namely item\_key, time\_key, shipper\_key, from\_location, to\_location.
- The shipping fact table also contains two measures, namely dollars sold and units sold.
- It is also possible to share dimension tables between fact tables. For example, time, item, and location dimension tables are shared between the sales and shipping fact table.



Fact Constellation

### III. QUERY PROCESSING

#### 3.1 Cube Operation

- Cube definition and computation in DMQL

**define cube** sales[item, city, year]: sum(sales\_in\_dollars) **compute cube** sales

- Transform it into a SQL-like language

SELECT item, city, year, SUM (amount) FROM SALES **CUBE BY** item, city, year

- Need compute the following **Group-Bys**

(date, product, customer),(date,product),(date, customer), (product,customer),(date), (product), (customer)

#### 3.2 Defining a Star Schema in DMQL

**define cube** sales\_star [time, item, branch, location]:

dollars\_sold = sum(sales\_in\_dollars),avg\_sales = avg(sales\_in\_dollars),units\_sold = count(\*)

**define dimension** time as (time\_key, day, day\_of\_week,month, quarter, year)

**define dimension** item as (item\_key, item\_name, brand,type, supplier\_type)

**define dimension** branch as (branch\_key, branch\_name,branch\_type)

**define dimension** location as (location\_key, street, city,province\_or\_state, country)

#### 3.3. Defining a Snowflake Schema in DMQL

**define cube** sales\_snowflake [time, item, branch, location]:

dollars\_sold = sum(sales\_in\_dollars),avg\_sales = avg(sales\_in\_dollars),units\_sold = count(\*)

**define dimension** time as (time\_key,day,day\_of\_week,month,quarter,year)

**define dimension** item as (item\_key,item\_name,brand, type,supplier(supplier\_key, supplier\_type))

**define dimension** branch as (branch\_key, branch\_name,branch\_type)

**define dimension** location as (location\_key,street,city(city\_key, province\_or\_state, country))

#### 3.4. Defining a Fact Constellation in DMQL

**define cube** sales [time, item, branch, location]:

dollars\_sold = sum(sales\_in\_dollars),

avg\_sales = avg(sales\_in\_dollars),

units\_sold = count(\*)

**define dimension** time as (time\_key, day, day\_of\_week, month, quarter, year)

**define dimension** item as (item\_key, item\_name, brand, type, supplier\_type)

**define dimension** branch as (branch\_key, branch\_name, branch\_type)

**define dimension** location as (location\_key, street, city, province\_or\_state, country)

**define cube** shipping [time, item, shipper, from\_location,to\_location]:dollar\_cost = sum(cost\_in\_dollars),unit\_shipped = count(\*)

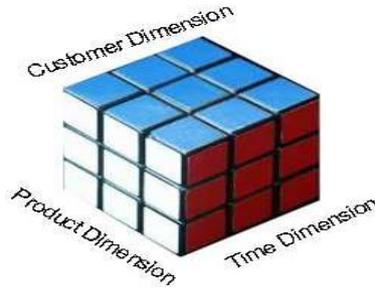
**define dimension** time as time **in cube** sales

**define dimension** item as item **in cube** sales

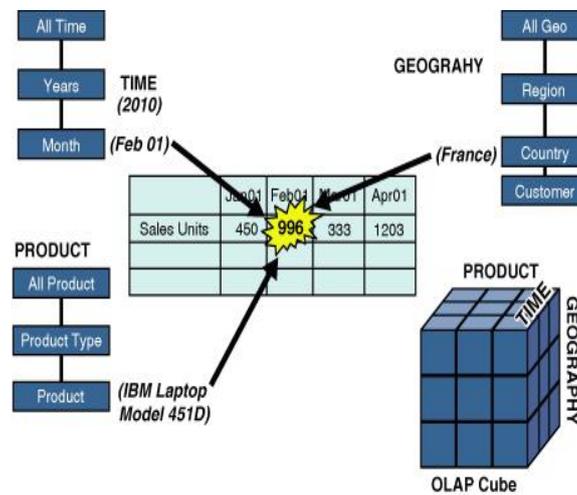
Let us then compute the two measures we have in our data cube: dollars\_sold and units\_sold

**Selects.**time\_key, sitem\_key,s.branch\_key,s.location\_key, sum(s.number\_of\_units\_sold\*s.price), sum(s.number\_of\_units\_sold)**from** time t, item i, branch b, location l, sales s **where** s.time\_key = t.time\_key and s.item\_key = i.item\_key and s.branch\_key = b.branch\_key and s.location\_key = l.location\_key **group by** s.time\_key, s.item\_key, s.branch\_key,s.location\_key

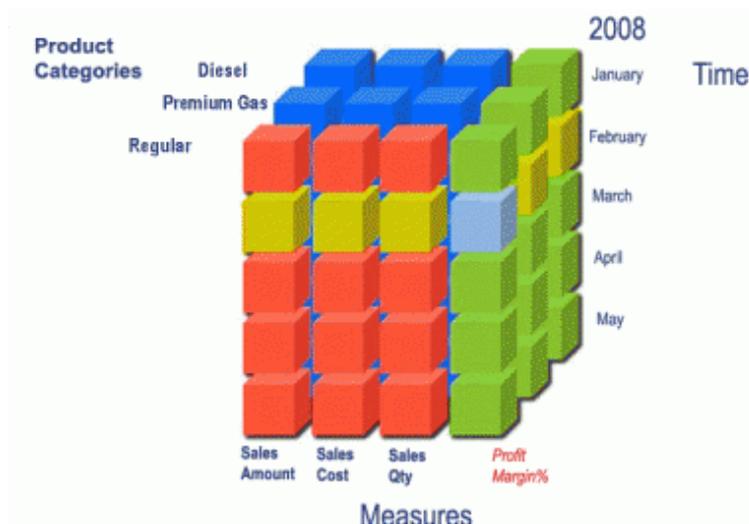
#### IV. MAKING OLAP CUBE MODELS



Three Dimensional Cube



Query generated cube



Cubical Model of Database

Our powerful data mining and analysis tools allow managers to clearly feel the beat of their business. Our dashboard enables you to follow key performance indicators on a daily basis to help you with your day to day management. Our OLAP cubes will provide you with the necessary information to help your decision making process. The OLAP database usually contains one or more OLAP cubes. Generally speaking, a cube is a geometric object with many sides (dimensions). In the case of databases, an OLAP cube contains fact tables and business dimensions. Fact tables contain facts (“measures”), such as revenue, costs, spending, quantity sold, etc. Dimensions are the business entities that provide context for the facts; loosely-speaking, they are the master files from an OLTP system, such as an account master, a product master, date master, cost center master, etc. The cube automatically ties (relates) the facts to the dimensions,

regardless of whether the relationships are direct, referenced (indirect) or even many-to-many. Internally, the OLAP cubes contain bitmap indexes that (loosely speaking) “auto-join” facts and dimensions so that end users can “slice and dice” measures by business dimensions, without needing to write any code. At this point, it’s VERY important to understand that each fact table has a statement of granularity - in other words, the dimension level at which the facts are stored. So an OLAP cube might have one fact table built at the reseller account/product/date level, and a budget fact table at the employee and quarter level. Each fact table has a statement of granularity - in other words, the dimension level at which the facts are stored. So an OLAP cube might have one fact table built at the reseller account/product/date level, and a budget fact table at the employee and quarter level. Each dimension contains attributes, which are the equivalent of relational table columns. Loosely speaking, attributes are the “by” in business user language. Attributes can form parent-child relationships, such as markets rolling up to regions, regions rolling up to zones, products rolling up to categories, months rolling up to years, etc. These relationships are known as hierarchies, and help end users to traverse these parent-child relationships through rollup and drilldown activities. It should be noted that not all attributes form parent-child hierarchies - some attributes (product color, for instance) might not be related to any other dimension attribute. Additionally, I can also build OLAP aggregations inside each OLAP fact table partition. OLAP aggregations are sometimes one of the more misunderstood features of Analysis Services. Because OLAP users are able to roll up data in OLAP cubes very quickly (e.g. roll up sales from day to year), many assume that OLAP cubes somehow store every possible combination of subtotals for each measure.

## V. CONCLUSION

We are analyzing the data mining by using UML Model. In UML Model different diagrams are drawn. In data mining we are studying collection of database. We are analyzing production and sales database. Schema shows how tables are related to each other. By writing many queries on database we create OLAP cube models. These models are easy to understand and changes big data into easy diagrams. Cubical models are performing various task. In big organization complex and bunch of data is changed into cubical models. CRISP DM-UML is an easy example of Business Intelligence.

## REFERENCE

- [1] J. Han and M. Kamber, “Data Warehouse and OLAP Technology for Data Mining” January 16, 2000.
- [2] Paulraj Ponniah, “DATA WAREHOUSING Fundamentals: A Comprehensive Guide for IT Professionals”, A Wiley-Interscience Publication JOHN WILEY & SONS, INC, 2001.
- [3] Óscar Marbán and Javier Segovia, “Extending UML for Modeling Data Mining Projects (DM-UML)”, Information Technology & Software Engineering Volume 3, Issue 2, 2013.
- [4] Antoaneta Ivanova, “Constructing Object Oriented Class for extracting and using data from data cube”, International Conference on Computer Systems and Technologies - *compSysTech'2005*.
- [5] Akintola K.G., Adetunmbi A.O. and Adeola O.S., “ Building Data Warehousing and Data Mining from Course Management Systems: A Case Study of FUTA Course Management Information Systems” ,International Journal of Database Theory and Application Vol. 4, No. 3, September, 2011.
- [6] Corinne Baragoin, Jorge Bercianos, Janez Komel, Gary Robinson, Richard Sawa and Erik Schuinder, “DB2 OLAP Server Theory and Practices”, International Technical Support Organization, April 2001.
- [7] V. Saxena, D. Arora and S. Ahmad, "Object oriented distributed architecture system through UML", *IEEE International Conference on Advanced in computer vision and information Technology*, November 28-30, 2007.
- [8] V. Saxena and G. A. Ansari “UML Modeling & Protection of Domain Based System”, International Journal of Computer Science and Network Security (IJCSNS), Vol. 8, No. 7, pp.338-344, July 2008.
- [9] V. Saxena, N.M.P. Verma and A. Pratap, “A Data Mining Technique for a Secure Electronic Payment Transaction,” International Journal of Economics and Finance, Vol. 2(4), pp. 12-17, 2010.
- [10] François Pinet, André Miralles, Sandro Bimonte ,Françoise Vernier ,Nadia Carluier4, Véronique Gouy4 and Stephan Bernard, “The use of UML to design agricultural data warehouses”, paper presented 10/4 at AGENG 2020 Conference, September 6-8 2010,
- [11] J. Rumbaugh, I. Jacobson, and G. Booch., The Unified Modeling Language Reference Manual, 2nd ed. Addison-Wesley, Boston, 2005.
- [12] Hasan H. “Using OLAP and Multidimensional data for Decision Making” IEEE International Conference on Advanced in computer vision and information Technology, Sep/oct 2001.
- [13] S. Chaudhari and U. Dayal “An overview of Data Warehousing and OLAP Technology”, ACM SIGMOD Record Volume 26 March 1997.
- [14] P.K. Chaurasia and V. Saxena, “Mobile Based Electricity Bill Deposit System Through UML,” Journal of Software Engineering and Applications, Vol. 4(3), 187-190, 2011.