International Journal of Emerging Trends & Technology in Computer Science (IJETTCS)
Web Site: www.ijettcs.org Email: editor@ijettcs.org
Volume 3, Issue 6, November-December 2014 **ISSN 2278-6856**

Dimension Modelling Techniques in Business Intelligence

Divya Sharma

Assistant Professor at PUSSGRC Hoshiarpur in Department of Information Technology

Abstract

this paper consists of the material about the different multi-dimension data modeling techniques used in Business Intelligence business intelligence and the general terms used in Business Intelligence. The paper also outlines the various advantages and disadvantages of different dimension modeling scheme such as star, snowflake or etc. schema.

Index Terms:- Dimension modeling, Star schema in dimension modeling, snowflake schema, data cube, and fact constellation.

1. INTRODUCTION

Dimension modeling is the technique associated with modelling data present in data warehouse in the form of data cubes or other data modelling schema. Dimension modelling is used to form relationship amongst data present on different computer separated by physical distance with the help of dimension table and fact tables. Dimension modelling is same as ER modelling both is used to form relationship between tables existing in normalized form. This paper contains a research study of the various multi-dimension modelling schemas available.

2. GENERAL TERMS IN BUSINESS NTELLIGENCE

2.1 BUSINESS INTELLIGENCE: Business intelligence is defined as "a set of concepts and methodologies to improve decision making in business through use of fact and fact-based systems". Transformation of raw data into business data is depicted below:

Raw data → Meaningful Information →Knowledge Discovery → Beneficial Insights →Impact Decisions → Business Benefits

2.2 DATA MODEL: A data model is a set of rules describing data and relationship amongst the data, the semantics associated with them. Data model is a very powerful data representation tool of data structure in expressing and communicating business requirements. Data model can also be defined as the precise description of data content in the system.

2.3 Types of data modeling

Conceptual data modeling: Examine the structure of the business and the rules associated with it. It is also known as information engineering. Conceptual data model deals with the storage of data. It describes what the system will contain. Conceptual data modeling describes "what the system will contain (data) and the relationship between these entities". Conceptual model does not describe primary key.

Logical data modelling Describes how the logical structure of database that will be implemented regardless of the DBMS, data types and integrity constraints that will be applied. Here two things are described "what data is stored in the database "and "what is the relationship amongst the data store in the database". This simple structure may be quite complex to implement at physical level. Logical model describes data in much detail than conceptual model. Logical model specifies the primary key and normalization is done here.

Physical data modelling: Physical data model is concerned with software and hardware environment of the data model. The physical implementation is highly dependent on the current state of technology and is subject to change as available technologies rapidly change. A technical design made five years ago is likely to be quite outdated today. It is complex model as ti describes the physical storage of data in the database which helps effective access of the data. Physical data model describe all the tables and columns. Here foreign keys are used to define relationship amongst tables. Physical data modelling a user can choose to have de-normalized data tables.

2.4 MODEL: "A model is a simplification of reality".

We model so that "We build model so that we can better understand the system we are developing".

Through a model we achieve four aims.

1). Model helps us to visualize a system as it is or as we want it to be.
2). Model permits us to specify the structure or behaviour of a system.
3). Model gives us a template that guides us in constructing a system.
4). Model documents the decision we have made.

2.5 DIMENSION TABLE: A dimension table allows keeping records of the dimensions. Each dimension may have a table associated with it this is called dimension table. Dimension table consist of the textual description of dimension of the table. For example: in a class we need to create a class data warehouse containing information such as subjects offered, students name, students roll no, student marks, etc. these dimension allows to keep track of the student performance.

Properties of dimension table:

1). Dimension table is related to fact table with the help of simple primary key.
2). these consist of the constraint used to link them to fact table.

Types of dimension tables:

1). Degenerate Dimension
2). Slowly Changing Dimension
3). Rapidly Changing Dimension
4). Role - Playing Dimension
5). Junk Dimension

Degenerate Dimension: These are the dimension present in the fact tables which are dimension without any attribute. These can be more than one. These dimensions are not joined to corresponding dimension in other dimension tables because of they already exist there. These dimensions are not as such directly useful to the user but important for queuing purposes.

Slowly Changing Dimension: A slowly changing dimension is a dimension whose attribute for a record change slowly over time, rather than change on a regular timely basis.

Rapidly Changing Dimension: A dimension that changes frequently is called Rapidly Changing Dimension.

Role - Playing Dimension: Here a single dimension is expressed differently in a fact table with the usage of views is called a role playing dimension. For example in a student table will consist of date of joining and date of birth of the student these are two different prospectuses and still related.

Junk Dimension: Junk dimension are the dimension that contain low-cardinality column / attributes such as indictors, codes, and status flags.

2.6 FACT TABLE: The fact table contains the names of the facts, or measure, as well as keys to each of the related dimension table. It can also be defined as the place where numerical measures about business data are stored. For example: in class database the fact table is the session of the admission of the student in college.

A fact table which does not contain numeric fact columns is called a “fact less fact table”.

Few properties of fact table:

1). Continuous values: they consist of continuous values.
2). Additive: may be correctly added by any dimension.
3). Semi additive: may experience additive properties on some but remain semi additive on few.
4). Fact table are generally sparse.

Types of fact:

1). Additive fact: These are the facts that can be summed up / aggregated across all dimensions in the fact table. For example in a retail store the total amount of sales that took place for a particular month or even for a day.

2). Semi Additive fact: These facts can be summed up for some dimensions in the fact table but not for all.

3). Non-additive fact: These facts cannot be summed therefore non additive in nature.

2.7 DATA WAREHOUSE MODELS

From architectural point of view there are three data warehouse models:

ENTERPRISE WAREHOUSE: enterprise warehouse collects all information about the enterprise it is useful as it provide cooperate wide data integration. Enterprise reporting helps in cross functional analysis of data. Enterprise ware house can be implemented on traditional mainframes, computer super servers, or parallel architecture platform. It may require years to design and build. It also requires massive investment of resources.

DATA MARTS: Data mart contains a specific group of data that is of value to a specific group of user. This scope is confined to a specific selected subject. Data marts are implemented on low cost departmental servers that are UNIX and WINDOW based. Data mart can be developed in week rather than in months. Data marts can be further categorized as independent and dependent. Dependent data marts are sourced directly from enterprise data warehouse. Independent data marts are sourced from external data source provider or data generated locally within a particular department or geographical area. Data marts basically focus on one department of an organisation hence are easy to design.

VIRTUAL WAREHOUSE: A virtual data base provides a set of views over operational database. Virtual database is easy to build but requires effective access over operational database.

Now we will study the various data modelling scheme such as data cube, star and snowflake scheme.

[1] 3. MULTI DIMENSION DATA MODELS SCHEMA

3.1 DATA CUBE:

Data cube allow data to be modelled and viewed in multiple dimensions. It is defined by dimension and facts.

Data cube is usually thought to be as 3-D representation but it is actually an n dimensional. A data cube is basically used for multidimensional data storage. The actual physical storage of such data may differ from its logical representation. A data cube is called a hypercube if it has more than 3 dimensions.

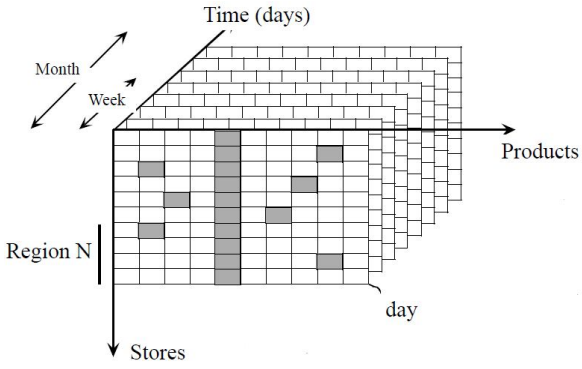

Fig 1.1: Diagram showing the data cube scheme.

The above data cube is an example of an electronic store here sale of product which can be a DVD, washing machine, dryer, etc. is monitored on daily basis, and along with is the various stores on which they are sold. All this information is stored in the form a data cube. This is done so as to monitor the trends in costumer and maintain inventory. Basically data cube is in the form of a cube where data can be easily saved at a single physical place and easily accessed by all. Due to less number of joins involved there is a decrease in the time of accessing a data as compared to rest of the schema available.

3.2 STAR SCHEMA:

It is the most commonly used schema for data modelling this schema comprises of a large central fact table consisting of no redundant data and small dimension tables which surrounds the fact table. This schema resemble star pattern and hence called star schema. In Star schema none of the tables are normalized.

For example star schema for college database is shown in figure 1.2 below. Student data base in a college will consist of student details (such as name, age, branch, etc.), subject opted by a particular student (such as subject name, subject id, etc.), marks obtained by a student (such as student id, marks obtained, subject id, etc.).

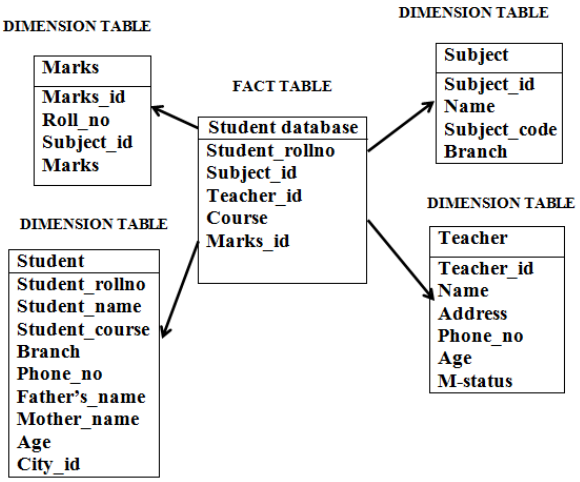

Fig 1.2: Diagram showing the star dimension scheme.

3.3 SNOWFLAKE SCHEMA

Snowflake schema is an advancement of star schema where some dimension tables are normalized that is they are further split into additional tables. These results in the graphical view of a snowflake due to such graphical representation this schema is called snowflake schema. For example this snowflake schema for college student database is represented in figure 1.3 below. Here the dimension table student detail for city are divided further into another dimension table which is a sub dimension table named city this table contains details such as city name, province, Country. This is done so as to normalize the dimension table student.

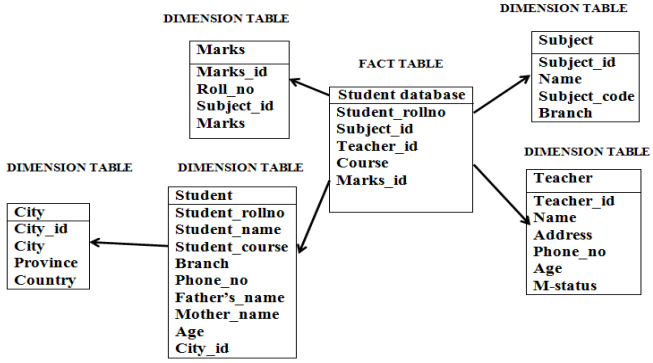

Fig 1.3: Diagram showing the snowflake dimension scheme.

Difference between star and snowflake schema is that star schema is not normalized while snowflake schema the tables are normalized therefore redundancy are removed. Such a table is easy to maintain and saves storage space. In snowflake schema more joins are needed to execute a query which reduces the effectiveness of browsing. Star schema is more popular than snowflake schema. In snowflake schema performance is adversely affected due to joins. Star schema is more efficient than snowflake schema. Star schema requires more space for storage while snowflake requires less space for storage on database. Star schema consists of less number of dimension tables than snowflake schema. Star schema is suitable for heavy end-user query workload while snowflake is not.

3.4 FACT CONSETLLATION SCHEMA:

Fact constellation allows sharing of multiple fact tables to share dimension table. This schema is also called galaxy schema.

For example in college student database will be related to library database where information such as total number of books issued and book name and date of issue and return of book and information as to whether the book has been issued to a student or teacher is stored.

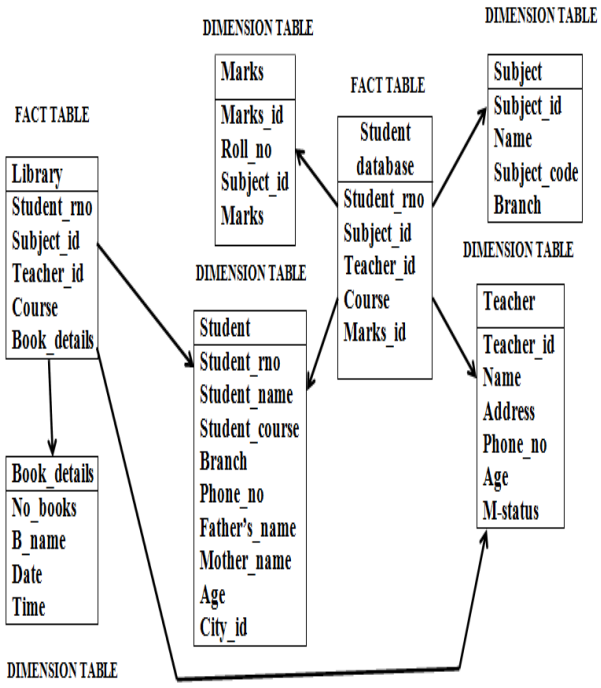

Fig 1.4: Diagram showing the Fact Constellation Scheme.

Brief comparison between star, snowflake and fact constellation schema:

Star and snowflake contains only one central fact table while fact constellation a fact table can access the dimension table of any another fact table therefore here sharing of dimension table takes place. Star is the simplest, while snowflake is a little complex while fact is the most complex to understand. Snowflake schema contains many sub dimension table while star and fact

don't. Complex join operations are written for fact constellation while less complex are involved in snowflake and star is the simplest. Performance is maximum in star while average in snowflake and minimum in fact constellation.

4. DIMENSION MODELING LIFE CYCLE

Same as in software engineering a software development life cycle is followed for modeling of data which follows a given sequence of steps. The above mentioned dimension models can be designed using this life cycle.
Various phases which are involved in dimension modelling are as follows:

1). Requirements gathering
2). Identifying the grain.
3). Identifying dimension
4). Identifying facts
5). Designing the dimension model

1). Requirements gathering: It is the process of selecting the business processes for which the dimension modelling has to done according to which requirement are gathered and documented. Here the process involved is studied and then information is analysed so that a particular dimension modelling scheme can be followed. In this phases we analysis the problem from two prospects' firstly what is to be analysed and secondly the criteria involved in performing the analysis.
2). Identifying the grain: A grain refers to the atomic level of data that can be analysed. Granularity is defined as the detailed level of information stored in a table.
3). Identify the dimensions: In this step we determine the dimension for the data model. Key features are:

- Dimension table contain attribute that describe facts of fact table.
- A dimension table must contain one atomic level of detail also called dimension grain.
- Each non key element should appear in a single dimension table.

4). Identify the facts table: Here we identify the fact table and relevant facts / measures in the table.
5). Designing the multi dimension data model: In this step a specified schema whether star, snowflake, or fact is drawn.

5. CONCULSION

The above research paper highlights the various multi-dimensional data models available and implemented now days for storage of large amount of data in a structured manner so that they can be used later for referencing, selection and extraction of any information present in it. Through these models optimal memory management can be done as shown there will be no need for repetition of information therefore no inconsistency in the database can occur. This paper shows that star schema is the simplest to design and implement at the physical level while snowflake schema and fact constellation schema is comparatively difficult to design, implement, time consuming and involve more man power. Star schema is the basic schema while fact and snowflake data modeling schemas are based on star schema.

REFERENCES

[1] Surajit Chaudhuri Umeshwar Dayal Appears in ACM Sigmod Record, March 1997 "An Overview of Data Warehousing and OLAP Technology".

[2] Chuck Ballard, Dirk Herreman, Don Schau, Rhonda Bell, Eunsaeng Kim, Ann Valencic. "Data Modeling Techniques for Data Warehousing".

[3] Jiawei Han, Micheline Kamber, Jian Pei, "Data Mining Concepts and Techniques".

[4] R N Prasad, Seema Acharya, "Fundamentals of Business Analytics".

[5] http://gama.vtu.lt/biblioteka/Information_Resources/i_part_of_information_resources.pdf

[6] http://www.databaseanswers.org/downloads/Data_Modeling_by_Example_Vol_1.pdf

[7] Gerhard Beck, "Basic Data Modeling Concepts", http://members.verizon.net/~gtbeck/data_modeling.pdf

[8] http://www.slideshare.net/RAAVIthrinath/data-modeling-28330266

AUTHOR

Ms. Divya Sharma received the M.TECH. degree in Computer Science & Engineering from Rayat & Bahra Institute Engineering and Bio-Technology (Kharar) 2012. She is currently working as Assistant Professor at PUSSGRC Hoshiarpur in Department of Information Technology, India.