Data Warehouse Schemas using Multidimensional Data Model for Retail

Kheri Arionadi Shobirin [1], Adi Panca Saputra Iskandar [2], and Ida Bagus Alit Swamardika [3]

[1][2] Department of Electrical and Computer Engineering, Post Graduate Program, Udayana University
[3] Department of Electrical and Computer Engineering, Udayana University
Email: KheriAS@gmail.com

***Abstract*—A data warehouse are central repositories of integrated data from one or more disparate sources from operational data in On-Line Transaction Processing (OLTP) system to use in decision making strategy and business intelligent using On-Line Analytical Processing (OLAP) techniques. Data warehouses support OLAP applications by storing and maintaining data in multidimensional format. Multidimensional data models as an integral part of OLAP designed to solve complex query analysis in real time.**

***Index Terms*: Data Warehouse, OLTP, OLAP, Multidimensional**

I. INTRODUCTION

There are three or more leading approaches to storing data in a data warehouse — the most important approaches are the dimensional approach and the normalized approach.

The dimensional approach refers to Ralph Kimball's approach in which it is stated that the data warehouse should be modeled using a Dimensional Model/star schema. The normalized approach, also called the 3NF model (Third Normal Form) refers to Bill Inmon's approach in which it is stated that the data warehouse should be modeled using an E-R model/normalized model.

In a dimensional approach, transaction data are partitioned into "facts", which are generally numeric trans-action data, and "dimensions", which are the reference information that gives context to the facts.

For example, a sales transaction can be broken up in-to facts such as the number of products ordered and the total price paid for the products, and into dimensions such as order date, customer name, product number, order ship-to and bill-to locations, and salesperson responsible for receiving the order.

II. DATA WAREHOUSE

A. ER Modelling

An entity–relationship model (ER model) describes inter-related things of interest in a specific domain of knowledge. An ER model is composed of entity types (which classify the things of interest) and specifies relationships that can exist between instances of those entity types.

In software engineering an ER model is commonly formed to represent things that a business needs to remember in order to perform business processes. Consequently, the ER model becomes an abstract data model that defines a data or information structure that can be implemented in a database, typically a relational database.

ER diagram use three basic graphic symbol to conceptualize the data: entity, relationship and attributes.

- **Entity**. An entity is an object that exists. It doesn't have to do anything; it just has to exist. In database administration, an entity can be a single thing, person, place, or object. Data can be stored about such entities. A design tool that allows database administrators to view the relationships between several entities is called the entity relationship diagram (ERD).
- **Relationship**. A relationship, in the context of databases, is a situation that exists between two relational database tables when one table has a foreign key that references the primary key of the other table. Relationships allow relational databases to split and store data in different tables, while linking disparate data items.
- **Attributes**. A relationship, in the context of databases, is a situation that exists between two relational database tables when one table has a foreign key that references the primary key of the other table. Relationships allow relational databases to split and store data in different tables, while linking disparate data items.

B. ETL Process

ETL (Extract, Transform and Load) is a process in data warehousing responsible for pulling data out of the source systems and placing it into a data warehouse [1]. ETL involves the following tasks:

- **Extracting**. Extracting the data from source systems (SAP, ERP, other oprational systems), data from different source systems is converted into one consolidated data warehouse format which is ready for transformation processing.
- **Transformed**. Transforming the data may involve the following tasks:
  - Applying business rules (so-called derivations, e.g., calculating new measures and dimensions),
  - Cleaning (e.g., mapping NULL to 0 or "Male" to "M" and "Female" to "F" etc.),
  - Filtering (e.g., selecting only certain columns to load),
  - Splitting a column into multiple columns and vice versa,
  - Joining together data from multiple sources (e.g., lookup, merge),
  - Transposing rows and columns,
  - Applying any kind of simple or complex data validation

(e.g., if the first 3 columns in a row are empty then reject the row from processing)

- **Loading**. Loading the data into a data warehouse or data repository other reporting applications.

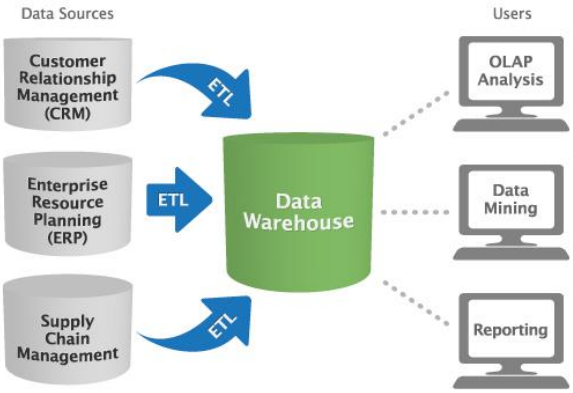

Fig. 1. Data Warehouse Design.

C. Data Warehouse Objectives

The main objectives of bulding a datawarehouse in business [2] are:

- **Availability**. Data is structured to be ready and available for analyticalprocessing activity such as OLAP, data mining, querying, reporting and any other decision supporting applications.
- **Easy Accessible**. Data warehouse content must be understable and labelled meaningfully. It must also return query results with minimal wait times.
- **Consistently**. Data warehouse must be reliable and its data must be collected from variety of sources in a relation, cleansed and quality checked.
- **Adaptive to Change**. Data warehouse must be able to handle changes. The existing data and applications should not be affected with changes, asking new questions and adding new data to the warehouse.
- **Security and Protection**. The data warehouse must control access to the confidential information.
- **Improve Decision Making**. The data warehouse must have trusted data to support decision making process.
- **Subject-Oriented**. A data warehouse is organised around major subjects such as customer, supplier, product and sales. Hence, data warehouse typically provide asimple and brief view around particullar subject issues by excluding data that re not useful in the decision support process.
- **Time Variant**. Historical data is kept in data warehouse. For example one can retrieve data from 3 months, 6 months, 12 months or even older data from a data warehouse.

D. Star Schema

The star schema separates business process data into facts, which hold the measurable, quantitative data about a business, and dimensions which are descriptive attributes related to fact data. Examples of fact data include sales price, sale quantity, and time, distance, speed, and weight measurements. Related dimension attribute examples include product models, product colors, product sizes, geographic locations, and salesperson names.

A star schema that has many dimensions is sometimes called a centipede schema. Having dimensions of only a few attributes, while simpler to maintain, results in queries with many table joins and makes the star schema less easy to use.

1) Fact Tables

Fact tables record measurements or metrics for a specific event. Fact tables generally consist of numeric values, and foreign keys to dimensional data where descriptive information is kept.[4] Fact tables are designed to a low level of uniform detail (referred to as "granularity" or "grain"), meaning facts can record events at a very atomic level. This can result in the accumulation of a large number of records in a fact table over time. Fact tables are defined as one of three types:

- Transaction fact tables record facts about a specific event (e.g., sales events)
- Snapshot fact tables record facts at a given point in time (e.g., account details at month end)
- Accumulating snapshot tables record aggregate facts at a given point in time (e.g., total month-to-date sales for a product)

Fact tables are generally assigned a surrogate key to ensure each row can be uniquely identified. This key is a simple primary key.

2) Dimensional Tables

Dimension tables usually have a relatively small number of records compared to fact tables, but each record may have a very large number of attributes to describe the fact data [2]. Dimensions can define a wide variety of characteristics, but some of the most common attributes defined by dimension tables include:

- Time dimension tables describe time at the lowest level of time granularity for which events are recorded in the star schema
- Geography dimension tables describe location data, such as country, state, or city
- Product dimension tables describe products
- Employee dimension tables describe employees, such as sales people
- Range dimension tables describe ranges of time, dollar values, or other measurable quantities to simplify reporting

Dimension tables are generally assigned a surrogate primary key, usually a single-column integer data type, mapped to the combination of dimension attributes that form the natural key.

3) Advantage

Star schemas are denormalized, meaning the normal rules of normalization applied to transactional relational databases are relaxed during star schema design and implementation. The benefits of star schema denormalization are:

- Simpler queries - star schema join logic is generally simpler than the join logic required to retrieve data from a highly normalized transactional schema.
- Simplified business reporting logic - when compared to highly normalized schemas, the star schema simplifies common business reporting logic, such as period-over-period and as-of reporting.
- Query performance gains - star schemas can provide performance enhancements for read-only reporting applications when compared to highly normalized schemas.
- Fast aggregations - the simpler queries against a star schema can result in improved performance for aggregation operations.
- Feeding cubes - star schemas are used by all OLAP systems

to build proprietary OLAP cubes efficiently; in fact, most major OLAP systems provide a ROLAP mode of operation which can use a star schema directly as a source without building a proprietary cube structure.

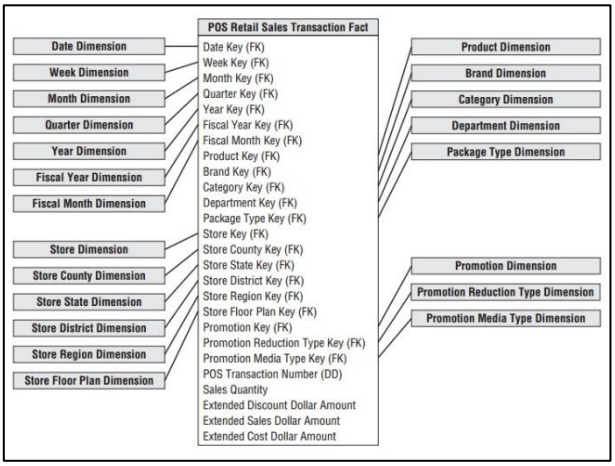

Fig. 2. Fact Tables and Star Schema for Retail.

4) Disadvantage

The main disadvantage of the star schema is that data integrity is not enforced as well as it is in a highly normalized database. One-off inserts and updates can result in data anomalies which normalized schemas are designed to avoid. Generally speaking, star schemas are loaded in a highly controlled fashion via batch processing or near-real time "trickle feeds", to compensate for the lack of protection afforded by normalization.

Star schema is also not as flexible in terms of analytical needs as a normalized data model. Normalized models allow any kind of analytical queries to be executed as long as they follow the business logic defined in the model.

Star schemas tend to be more purpose-built for a particular view of the data, thus not really allowing more complex analytics.

Star schemas don't support many-to-many relationships between business entities - at least not very naturally. Typically these relationships are simplified in star schema to conform to the simple dimensional model.

E. Multidimensional for Retail Schema

The facts collected by the POS system include the dimension of Time (can be clasified as several dimension), dimension of Product, dimension of Sales, dimension of Store (can be clasified as several dimension) and dimension of Promotion (can be clasified as several dimension).

When we have a product dimension, the table have duplicates data of product as much as transaction created in POS. To reduce data duplicates in product dimension table, the table can be normalize into starflake model.

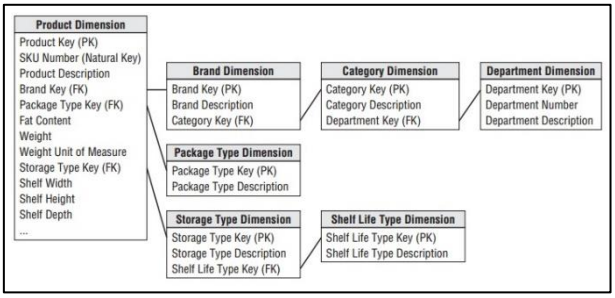

Fig. 3. Starflake Schema for Product Dimension

III. CONCLUSION

Basically we can implement multidimensional data model for retail. But when its faced with a more complicated data want to be drill down and a huge data to be calculate, Multidimensional data model with star schema is not enough.

We need another schema to be used. Starflake schema can help us reducing a data duplication based on normalization procedure.

REFERENCES

[1] R. Kimball, “Kimball Dimensional Modeling Techniques.”
[2] B. Inmon, “Building The Data Warehouse (2005) Fourth Edition.”.