Data warehousing report example

Business, Company

Star schema (simple)

“ Star” schema is actually the simplest architecture within data warehousing. This architecture is termed as star because of the diagram that looks like a star. In the center of this diagram, there is a “ fact” table that links to the “ dimension” tables. Star schema is the most common data warehousing schema that is used. Oracle recommends it to be used. Star schema is a widely supported business intelligence tool. This architecture is very useful since you can create some useful queries and in addition it is very easy to do since not many tables need to be joined (Wang 2006). Another advantage of using star schema is that it is very easy to understand the diagram and data. When using this architecture, one should remember that storage space is required since when de-normalization is taking place, the redundancy data will cause the size of table to be large. This is one of its disadvantages. Another shortcoming of using the star schema is that it takes a long time when loading data into the dimension tables.

Fact table

A fact table is the one that is found in the center and has two types of columns. The first column is the foreign keys to dimension tables and the second column is the measures of those that contain numeric facts.

Dimension Table

Dimension tables have the primary keys in each table and the attributes of the dimensions to help in describing the dimension that would mostly be in textual format. Overall the concept fits into Data Warehousing neatly since it is one of the simplest and easiest schemas to be used (Reeves 2009).

For example, a fact table can be used to store Sales data and have the dimension table attributes such as Customer details, Product details, and Store Details. This is as illustrated below.

Snowflake Schema (Complex)

This is a normalized star schema. Its dimension tables are also normalized. Snowflake schema is not a simplest schema for data warehousing because it is not easy to understand and this makes it hard for one to carry out queries. This schema is appropriate for use in small data warehouses. The use of this schema is easy to maintain or change as there are no redundancy data. When the user is performing queries, the execution time takes longer since more foreign keys need to be looked into than using other schemas (Kelly 2000). The Snowflake schema is the best to be used when the dimension table is relatively big because it reduces the space as it breaks it down. However, it increases the number of tables that the user needs to work with and this makes it harder to make quires as the number of tables needs to be joined. This is illustrated below.

The above diagram is an example of a snowflake schema, in this example the middle table is the fact table (sales), which consists of all the primary keys from the surrounding dimension tables (Product, Account, Time and Geography) These dimension tables are then branched out to other sub-dimension tables which resemble a snowflake pattern (Kimball 2008). Generally, in Data Warehousing, this concept is suitable for small data warehouse as it makes it easier to maintain the data. However, it is not appropriate for big data warehouse as it is not easy to understand big data tables.

Constellation Schema (FACT)

A constellation schema is in a position to be built for each star or snowflake schema. This schema is more complex than the star and snowflake schema since it contains multiple fact tables allowing the dimension tables to be shared amongst many fact tables. Constellation fact schema is a normal and natural consequence of the dimensional modeling. The disadvantage of this schema is that it is complicated to design as there are many alternative solutions that can be considered (Taniar 2011).

A fact constellation schema consists of different fact tables that are clearly assigned to the dimension tables, which are for relevant facts. For example there is a fact table (Sales) that is linked to the dimension tables (Product, Client, Shop and Time) and then there is another fact table (Delivery) which is also linked to the fact table (Sales) by the dimension tables that would be creating the constellation fact table. In data warehousing, constellation schema is complex to use but it is beneficial if facts tables need to be shared with other dimension tables (Ferdinandi 2000). The diagram below shows a constellation schema.

Cube

Logically, Cubes represent the results of the multi-dimensional data. A cube is visually a 3 dimensional model but can also have more than 3 dimensions, which are structured and defined by sets of dimensions and measures. For instance, a sales analysis cube can be used to measure the item sales and item costs and have dimensions of store location, product line, and financial year allowing the user to separate the item sale price and cost into various categories by the location, product line and financial year. Using cube is very beneficial as it is very fast to perform queries and easy to understand. Cube and star schema are equivalent in the data they store since they can be converted into each other (Koray 2001). The cube can make measures in three categories which are, Distributive (count, sum, Max, Min), Algebraic (Average, min, standard deviation), and Holistic (Median, Mode, Rank). Generally, the concept of cube in Data Warehousing is that it is very useful as you are able to perform fast quires that are easy to understand. The following diagram shows a cube.

Slow Changing Dimensions

The slow changing dimension is used to update and insert records in the tables. There are three types of slow changing dimensions;

•Type 0 – Ignores the value.

This ignores / reports as wrong values

•Type 1 - Overwrite the value.

This simply overwrites the changed value and easy to be implemented.

•Type 2 - Adds a new dimension row.

When using the type 2 you would be adding a new description by adding a new row. It will also add an Effective date column.

•Type 3 - Adds a new dimension column.

Using the third type of slow changing dimension will add a new column where it will save the old value. This can only record a single change. Overall in data warehousing the use of slow changing dimensions is very useful for certain dimensions because it allows you to change records in the tables.

ETL

ETL stands for Extract – Transform – Load

This concept is not difficult to understand but it is a huge task since a multi-skill staff is required.

According to Kimball, ETL consumes around 70% of the time and effort of building a data warehousing/business intelligence environment.

Kimball Data Warehouse Lifecycle Toolkit 2008 clearly explains to us what ETL is;

1. The data is extracted from the original source location (Extract)
2. You do something with it (Transform)
3. It is loaded into tables for the users to query (Load)

The use of ETL is vital in data warehousing. This is because it consumes most of the time of building a data warehousing environment.

Data Quality

The quality of data is very important because when populating data into data warehouse tables you need to have quality data into your tables for you to have a useful database. The following can be poor quality data:

- Incomplete/Missing data – Null or incomplete values
- Wrong Data – Incorrect data
- Duplicated Data – Same Data repeated more than once

The following leads to poor quality data.

- Users – Poor data entry
- Validations – Poor data validations
- System Design – Poor input form, confusing the users

Data quality can be improved by having lower chances of poor data. For example; in the database we can use validations rules, data types and referential integrity. In the software you can adapt the use of default values and pattern matches (Becker 2002). You can also train and support the staff responsible for entering the data.

The concept of data quality is generally very important in data warehousing as well as outside data warehousing. It is vital to check the quality of data so as to minimize and try to prevent it from affecting the database.

Architecture: Inman vs. Kimball

Inman

Inman is an Enterprise Data Warehouse by Bill Inman. This is a top down driven approach enterprise. This enterprise Data Warehouse contains a merged copy of the original relational data where the end user does not see. What he or she sees is data marts that are designed to his or her needs, the data marts come as the cubes M-OLAP (Berson 2002). The Inman approach is used for huge enterprise data warehouses. In each data warehouse

- The data grows by the day
- The data is cleaned and consolidated
- The data is held in relational format
- The data is slightly extended from the original

Kimball

Ralph Kimball was the first person to recognize Dimensional approach, which is a bottom up bit by bit approach based on star data. Kimball approach recommends aggregating the facts table to the lowest level that is needed.

The Risk of having the facts table to its lowest level is that later someone might request for more details and since we have the lowest level fact tables, it might not contain the data from the star fact tables.

When using Kimball approach we are able to have the facts table to its lowest level without any worries of the future queries as we are in a position to use an ODS table, which is an Operational Data Store that is not visible for the users. ODS store data which is not used can be used in future. Generally, in the architecture concept of data warehousing, it is important to use the appropriate architecture for the data warehouse needed (Hobbs 2005).

Architecture: System Support

When deciding which architecture approach to use, you would still need to look into;

•Where the data is going to be stored

• The database system that you will be using

You need to find out where the data is going to be stored once you have decided which Data Warehouse architecture you are going to use.

In Data warehousing there are 5 major types of Architectures

1. Enterprise Data Warehousing Architecture
2. Data Mart Architecture
3. Hub and Spoke Data Mart Architecture
4. Enterprise Warehouse with Operational Data Store
5. Distributed Data Warehousing Architecture

What is the database system that you are going to use will depend on which Data Warehouse structure you select e. g. Inman or Kimball.

Overall, the concept of system support is imperative as you will need to have looked at it before selecting an architecture structure for the data warehouse.

Methodology

Following a methodology gives you a formal way of tackling a subject. As in Data Warehousing, it provides you with a step by step process of developing a data warehouse.

There are over 20 mainstream methods, such as;

•Top Down = NCR/Teradata

•Bottom Up = SAS

•Middle Out = Kimball

Kimball lifecycle methodology essentially focuses on the business dimensionally structures, the data, access of the data by reports or ad-hoc quires, and lastly develops the overall Data Warehouse in steps rather than all at once (Kimball 2008).

The concept of a methodology is vital in data warehouse. So, one is able to follow the plan rather than just do all the work at once. This is not only important in data warehousing methodology since it is taken into action in many places in helping one in the entire process.

Bibliography

Wang, J. (2006). Encyclopedia of data warehousing and mining. Hershey, PA: Idea Group Reference.

Reeves, L. L. (2009). A manager's guide to data warehousing. Indianapolis, IN: Wiley Pub.

Kelly, S. (2000). Data warehousing in action. Chichester, West Sussex, England: New York.

Ralph Kimball, 2008. The Data Warehouse Lifecycle Toolkit. 2 Edition. Wiley.

Taniar, D., & Chen, L. (2011). Integrations of data warehousing, data mining and database technologies: Innovative approaches. Hershey, PA: Information Science Reference.

Ferdinandi, P. L. (2000). Data warehousing advice for managers. New York: AMACOM.

Corey, M. J., & Abbey, M. (2001). Oracle data warehousing. Berkeley, Calif: Osborne McGraw-Hill.

Becker, S. A. (2002). Data warehousing and web engineering. Hershey, PA: IRM Press.

Hobbs, L. (2005). Oracle Database 10g data warehousing. Amsterdam: Elsevier.

Berson, A., & Smith, S. J. (2002). Data warehousing, data mining, and OLAP. New York: McGraw-Hill.