5.1 Data Warehouse Modeling

Data warehouse modeling is the process of designing the schemas of the detailed and summarised data of the data warehouse. The aim of data warehouse modeling is to design a schema representing the reality, or at least a part of the reality, which the data warehouse is required to support.

Data warehouse modeling is an important stage of building a data warehouse for two main reasons. Firstly, through the schema, data warehouse users have the ability to visualise the relationships among the warehouse data, so as to use them with greater ease. Secondly, a well designed schema allows an effective data warehouse architecture to emerge, to help reduce the cost of implementing the warehouse and improve the efficiency of using it.

Data modeling in data warehouses is rather different from data modeling in operational database systems. The main functionality of data warehouses is to support DSS processes. Thus, the aim of data warehouse modeling is to make the data warehouse efficiently support complex queries on long term information.

In contrast, data modeling in operational database systems focuses on efficiently supporting simple transactions in the database such as retrieving, inserting, deleting and changing data. Moreover, data warehouses are designed for users with general information knowledge about the enterprise, whereas operational data base systems are more oriented toward use by software specialists for creating specific applications.
Modeling warehouse data requires information about both the source data and the target warehouse data. The source data can be treated as inputs which are transformed into the target warehouse data. How this transformation happens is required to be reflected in data warehouse modeling.

Multidimensional data modeling is a commonly used technique to conceptual and visualize schemas by using the major components of the business, such as customers, products, services, prices and sales. This data modeling technique is especially used for summarising and rearranging data and presenting views of the data to support DSS. Particularly, multidimensional data modeling focuses on numeric data such as sales, counts, balances and costs.

In multidimensional data modeling, the data warehouse is designed to collect facts on one or more measures, each measure depending on a set of dimensions. For example, a sales measure may depend on three dimensions: products, times and locations.

Facts are collections of related data items, which are stored within fact tables in the data warehouse. Dimensions are collections of the items of one component of the business, such as the products dimension, the times dimension and the locations dimension for sales. The items of a dimension are stored within a dimension table in the data warehouse.

The primary key of a fact table is a concatenation of the primary keys of one or more dimension tables. Thus, every row in the fact table is associated with one and only one row from each dimension table.

Measures are the non key attributes of fact tables, and they represent information relating to the dimensions key attributes of the fact table.
The non key attributes of a dimension table may be organised as a dimension hierarchy. For example, the times dimension may consist of the dates, months and weeks attributes; the products dimension may consist of the category, model and producer attributes; and the locations dimension may consist of the city, region and country attributes.

There are two kinds of schemas used in multidimensional data modeling: star schemas and snowflake schemas. A star schema typically has one fact table, and a set of smaller tables. The links between the primary keys of the fact table and the foreign keys in the dimension tables can be visualized as a radial pattern with the fact table in the middle.

The dimension tables may contain data redundancies. For example, in the dimension table Locations(LocationID, Address, City, Region, Country), the City and Region information may be repeatedly stored for the locations in the same cities. This kind of data redundancy incurs storage overheads and may lead to update anomalies and poor update performance.

If necessary, snowflake schemas can be used to avoid such data redundancies. A snowflake schema is the result of normalizing the dimensions of a star schema, in which there are links between primary keys and foreign keys of tables in the dimension hierarchy. Fully normalizing the dimension tables may not be necessary in a data warehouse environment. Since there are generally no updates occurring to individual rows in the dimension tables, although new rows may be added when the data warehouse is refreshed with new data, and existing rows may be deleted.

when the data warehouse is purged of out of date data, the issue of update anomalies and poor update performance will generally not arise in the data warehouse.
In addition, the storage consumption of the data warehouse is dominated by the fact tables and the space saved by normalizing the dimension tables would generally be comparatively small. Moreover, unnormalized dimension tables can reduce the time required to combine information in the fact table with dimension information, which is a main performance criterion of a data warehouse.

5.2 Types of models

Figure 5.1. Relational model

The hybrid model is based on the relational model, with two changes that derive from dimensional modeling practices: (1) Create a relationship from the PREMIUM table to each table in the upper portion of the hierarchy,
and (2) Add the time dimension. The Existing star snow galaxy schemas of data warehouse will be discussed and hybrid model will be suggested.
Figure 5.3. Hybrid model
The revised model, can be represented with a star or a snowflake schema. At the center of
the star is the event table, capturing the critical information about the event (the fact table). Surrounding the event are the dimensions of resources, agents, location, and time period as they relate to the event, resulting in a star schema.

FIG 5.6: Sample Snowflake Schema

5.3 A Data Warehouse Hybrid Model

HYBRID model the revised model, can be represented with a star or a snowflake schema. At the center of the star is the event table, capturing the critical information about the event (the fact table). Surrounding the event are the dimensions of resources, agents, location, and time period as they relate to the event, resulting in a star schema. Additional tables are included to provide information about time period and manufacturer. As in REA and REAL models, the particular process being modeled influences which resources, events, agents, and locations are included and the number of tables used to represent each. Additional information changes the star model into a
snowflake model. Control information can also be added with links from agents to economic unit, or agent information can be linked to resource information.

**Data Warehouse Synchronism Model**

Relational and dimensional modeling are often used separately, but they can be successfully incorporated into a single design when needed. Doing so starts with a normalized relational model and then adds dimensional constructs, primarily at the physical level. The result is a single model that can provide the strengths of its parent models fairly well: it represents entities and relationships with the precision of the traditional relational model, and it processes dimensionally filtered, fact aggregated queries with speed approaching that of the traditional dimensional model. Real world experience was the motivation for this analysis: on three separate data warehousing projects where I worked as programmer, architect, and manager, respectively, I found a consistent pattern of data/database behavior that lent itself far more to a hybrid combination of dimensional and relational modeling than to either one alone. This article discusses the hybrid design and provides a fully functional reference implementation.

**FIG 5.7 . Data Warehouse Synchronism Model**
Dynamic Synchronism Model

- Dynamic Synchronism – updates on transactional environment reflected immediately on analytical environment
- DW loading divided among various maintenance transactions
- Data portions based on business niches
  - Solution for parallelization of static and dynamic loadings
- Hybrid Synchronism Model
- Portions are synchronized in different time intervals
- Based on Characteristics of:
  - Transaction/Operational Environment
  - Analytical Applications Environment

Hybrid tables

- After further experience and discussion it was realized the functional tables could be hybridized to satisfy specific reporting needs and to provide a transitional bridge to the future.
- Definition – a hybrid functional table is one that has data derived from disparate systems. (normally legacy and PeopleSoft)
- Hybrid tables can become transitional tables.
- With time, hybrid tables can become normal functional tables. (When the legacy data is no longer required, the columns cease to be filled or are removed.)

<table>
<thead>
<tr>
<th>Stdnt No.</th>
<th>Addr_type</th>
<th>Addr_line1</th>
</tr>
</thead>
<tbody>
<tr>
<td>12345678</td>
<td>L</td>
<td>BVRIT, NSP</td>
</tr>
<tr>
<td>23456789</td>
<td>M</td>
<td>LBRCE, MYM</td>
</tr>
<tr>
<td>34567890</td>
<td>B</td>
<td>VSCE, KMM</td>
</tr>
</tbody>
</table>

Table:5. 1
### Converted PeopleSoft

<table>
<thead>
<tr>
<th>Stdnt No.</th>
<th>Addr_type</th>
<th>PS.Addr_type</th>
<th>Addr_line1</th>
</tr>
</thead>
<tbody>
<tr>
<td>12345678</td>
<td>L</td>
<td>LOC</td>
<td>BVRIT, NSP</td>
</tr>
<tr>
<td>23456789</td>
<td>M</td>
<td>Mail</td>
<td>LBRCE, MYM</td>
</tr>
<tr>
<td>34567890</td>
<td>B</td>
<td>BUS</td>
<td>VSCE, KMM</td>
</tr>
</tbody>
</table>

**Table : 5.2**

Table name (address) is unchanged

Code changes kept to a minimum

**Fig 5.8.** : Hybrid table process
Table 5.3 : Hybrid Table

**Original — Data Warehouse Table Name/Structure and Column Names**

<table>
<thead>
<tr>
<th>Stuno</th>
<th>Gender</th>
<th>Vet_code</th>
<th>Citizenship</th>
<th>Fin_Aid_Ind</th>
<th>Mar_Sat</th>
<th>Eth_Org</th>
<th>DOB</th>
</tr>
</thead>
<tbody>
<tr>
<td>Student #</td>
<td>M/F</td>
<td>1</td>
<td>IND</td>
<td>Y/N</td>
<td>S</td>
<td>6</td>
<td>WW/DD/YYYY</td>
</tr>
</tbody>
</table>

**First Degree of Hybridization**

<table>
<thead>
<tr>
<th>Student #</th>
<th>Old Stuno</th>
<th>Gender</th>
<th>Vet_code</th>
<th>PS_MIL_St</th>
<th>Citizenship</th>
<th>Fin_Aid_Ind</th>
<th>Mar_Sat</th>
<th>PS_Mar_St</th>
<th>Eth_Org</th>
<th>DOB</th>
<th>Sp Interests</th>
</tr>
</thead>
<tbody>
<tr>
<td>M</td>
<td>M</td>
<td>M</td>
<td>1</td>
<td>Vet</td>
<td>Ind</td>
<td>S</td>
<td>Single</td>
<td>6</td>
<td>Band</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Second Degree of Hybridization**

<table>
<thead>
<tr>
<th>Student #</th>
<th>Old Stuno</th>
<th>Gender</th>
<th>Vet_code</th>
<th>PS_MIL_St</th>
<th>Citizenship</th>
<th>PS_Mar_Sat</th>
<th>Eth_Org</th>
<th>DOB</th>
<th>Sp Interests</th>
</tr>
</thead>
<tbody>
<tr>
<td>M</td>
<td>M</td>
<td>Male</td>
<td>1</td>
<td>Vet</td>
<td>Indian</td>
<td>S</td>
<td>Single</td>
<td>6</td>
<td>Caucasian</td>
</tr>
</tbody>
</table>

**Third Degree of Hybridization**

<table>
<thead>
<tr>
<th>Student #</th>
<th>Gender</th>
<th>PS_MIL_St</th>
<th>PS_Citzn</th>
<th>PS_Mar_Sat</th>
<th>PS_Eth_Org</th>
<th>DOB</th>
<th>Sp Interests</th>
<th>Ambassador</th>
<th>Collection ID</th>
</tr>
</thead>
<tbody>
<tr>
<td>M</td>
<td>Male</td>
<td>Vet</td>
<td>Indian</td>
<td>Single</td>
<td>Caucasian</td>
<td>YYYY/MM/DD</td>
<td>Band</td>
<td>YES</td>
<td>12345</td>
</tr>
</tbody>
</table>