

Instructor: John Beswetherick
Course: BUSIT 202
From: Christopher Singleton
Subject: Assignment6 – Handling Customers
Date: 02/16/2016

COVER MEMO

From what I read, chapter 8 is all about customer relationship management (CRM) in understanding when and how to apply dimensional modeling techniques throughout case scenarios based on the attributes of their customers. The dimension modeling techniques discussed are having to do with outrigger dimensions (Moving [Dim Customer] attributes that never change to their own outrigger table while referencing directly to the [Dim Customer] table using the snow-flake schema.), behavior tags (indicators) that signify customer attributes in how they are referenced by the [Dim Customer] table (Although you can use wild card searches to bring up results, I don't think this is the best method.), and creating a mini dimension bridging table on the customer for changes that explain the customer's attribute changes that can change frequently. I always understand having attributes about your customers as an opportunity with companies in how they can more refine in narrowing how they market to us. By getting to know us their customers more (Their buying habits and the decisions that they make with how they make their decisions also). The more information they collect, usually the more likely they can analyze where, when, and how customers purchase our goods. Not only that, you can also analyze habits in when your customers will make a purchase.

The key to making decisions on the changes in a layout is what information goes into what tables (how you reference tables) and why your layout corresponds to the correct conventions. Clearly based on how often the information changes and how you properly layout the database in corresponding to the changes. One of the most important questions is why changes need to be made. In this case, the number of growing attributes (flags and demographical) can become over whelming on the efficiency of data querying.

Review the dimCustomer table and any dimension tables related to customer information contained in dimCustomer.

- *Review the database diagram from the AdventureWorksDW2012 data warehouse that is found at the end of this document as well as the sample data included in the separate spreadsheet: Customer.xls*

Clearly the Customer.xls table shows the content that would be of column attributes of the customer table. In this case, many of the Customer table's attributes can be divided up between a dimensional customer outrigger table and a dimensional mini customer table as shown on the last page. For reasons, as the attribute columns continue to grow, it would be impossible to keep track or even display the information without filtering, not only that, but why clog up your dimension tables when you can move those attributes to another dimension table that references the dimension table instead. Keep in mind that in a dimensional outrigger table the attributes don't change very frequently, where as in a dimensional mini customer table things often change or become updated. In order to save a few columns, I would concatenate the customer name and increase the datatype length to nvarchar (100) or more. I would also make it so that every customer has a CustOuttriggerKey in the [Dim Customer Outtrigger] table and a MiniCustomerKey in the [Dim Mini Customer] table.

- Write a narrative that describes how the AdventureWorksDW customer dimension either adheres, partially adheres, or does not adhere to Kimball design techniques. Include in your narrative, your opinion of the current customer dimension design, what you would suggest changing, and reasoning to support your position. Your narrative should at a minimum address the following design principles:

- **Breaking name and address down into elemental parts**
Addresses are considered multivalued designed that hold only two choices, a positional design or a bridge table design. A positional design increases the number of columns which makes the table very large (named column for each attribute), where as a bridge table design holds the attributes in a separate table and is only utilized according to the query in bringing out the data, which can make for longer queries. Multivalued bridge tables are often used for demographics, the status attributes of their customers and is the most chosen when it comes to holding many attributes of their customers. When you have a very large set of demographic indicators, outriggers and/or mini dimensions does not scale gracefully, though the best choice is having a mini dimension if you have a large amount of indicators. Getting a person's name right is important when dealing with customers and can be considered an embarrassment (culturally). I would think Kimball is not too fond of creating bridge tables, but when absolutely necessary for the amount of information that you may not even use often, there may not be any choices left. Kimball stresses that you should not if at all possible create bridging tables or any other type of snowflake patterns in your design, unless it is absolutely necessary.
- **Use of dates and the date dimension**
One of the most common ways to use the date dimension is by using fixed depth hierarchies. As explained in Chapter 7, fixed depth hierarchies are many to one relationships and are specifically used for dimensional purposes in explaining time sequences or periods of time. The advantage is that you can roll up or down the information in the dimension according to what is needing to be shown with multiple data channels. Note, that dates also have a style which addresses cultural issues when doing business. Without a doubt, Kimball includes dates and his date dimensional attributes in his ideal layout. I would find it nearly impossible to analyze any information without having time sequences when querying.
- **Appropriate use of outriggers**
Dimensional outrigger table attributes are considered static, meaning that they are attributes that do not change as often. Also, the dimensional outrigger design, utilizes the snowflake schema that references the dimensional table and represents where the table's data initially came from. Kimball would not do this kind of design, because he doesn't like the idea of how the snowflake schema can slow down query responses and the lengthy amount of code it takes to query. Instead, Kimball's idea is in keeping it simple by utilizing the star schema design as his basic layout.

- Appropriate use of mini-dimensions

Mini dimensional information changes often and this means that the attributes are updated or changed often. The mini dimensional design utilizes the star schema and connects directly to the fact table. This type of dimension table's holds attributes can continually change or update and references another table. The attributes in the mini-dimensional table are usually made up of flags and other types of data attribute information that would be very important to the company doing business with the customer. This conforms to the design in what Kimball would do, because Kimball emphasizes about using the star schema in the overall design. Bridging tables are sometimes unavoidable only when appropriately necessary.

Write a narrative to explain what changes you would make to the customer dimension to accommodate AdventureWorks desire to add some information to the customer dimension to indicate “how important the customer is” and “how loyal the customer is”.

In creating the layout shown below, I created two new tables, one a [Dim Customer Outrigger] for the attributes that seldom change in the [Dim Customer] table and I also created a [Dim Mini Customer] table, for those attributes that can change often. Please note that the [Dim Customer Outrigger] table is connected directly to the [Dim Customer] table (snow flake schema) and the [Dim Mini Customer] table is connected directly to the [Fact Internet Sales] fact table (star schema). This is exactly how I would lay out the tables with one exception in that I would include the first name, middle name and last name all on one column, which would reduce the number of columns and increase efficiency in table querying. Not only that, by having attributes about your customer, you can easily query how often and how much the customer buys, which would signify an important customer. The trends on your customer attributes also would tell you what they are buying, how much they visit the store (customer loyalty) and when they are likely to buy more (time of the month when they buy).

