

Kimball Design Tip #19: Replicating Dimensions Correctly

By Ralph Kimball

The secret of building a distributed data warehouse is using conformed dimensions. In a distributed data warehouse many separate sources of measurements are maintained by different departments. These measurements are usually presented in "fact tables". One department may measure item manufacturing results, and another may measure item inventory. A third department may measure item sales, and a fourth may measure item comments and complaints. Clearly all these departments have a common interest in "item". We can build a distributed data warehouse if we can get these four departments to agree on the definition of items.

Actually, we need say this more strongly. We can build a distributed data warehouse if these four departments all use the conformed item dimension. And yes, these departments need to conform all other dimensions they have in common, such as time and customer. But we will focus only on the item dimension in this discussion.

The simplest way to conform the item dimension is for all four departments to use the same, identical item dimension table. Same keys, same attributes, same everything. Each of the four departments, of course, must then convert any private item keys in their fact tables to the public surrogate keys used in the conformed item dimension table. I described this surrogate key pipeline in an article, [Surrogate Keys](#).

A more complex way to use the conformed item dimension is to allow one of the departments to use a subset of the item dimension table. Suppose the department measuring item comments only records the comments at the brand level, not at the individual item level. It would be acceptable for this departments to use a shrunken version of the item table that only carries information down to the brand level. Of course, this would force any "drill-across" application that was combining information from this data mart with other data marts to seek the lowest common level of the various item dimension tables, which in this case would be at the brand, not individual item, level.

The real payoff of using conformed dimensions is being able to drill across separate data marts (fact tables). If you can constrain and group on the same item characteristics in each separate data mart, you can then line up the separate answer sets using the row headers that come out of the item dimension table. So, on one report line you can show item production, item inventory, and item sales at a very detailed level, and if you move up to the brand level, you can include counts of item complaints.

Drilling across is the key conceptual step in using a distributed data warehouse, and avoiding the need to have it centralized.

But administering a conformed dimension requires special discipline. The overall organization needs a "dimension authority", in this case an item czar. This dimension authority is responsible for maintaining the item dimension and replicating it successfully to all the data mart clients who make any use of item in their fact tables. We need to take seriously the task of replicating the dimension and enforcing its consistent use.

It would be a disaster if we drilled across several data marts accumulating results for a report, when

half of the data marts had yesterday's version of the dimension and half had today's. The results would be insidiously wrong. The row labels would not mean the same thing if any of the definitions of any of the reporting attributes had been adjusted. For example, if a category manager had changed the definition of one of the item categories, the reported results across these out-of-synch row headers would be wrong. And yes, category managers have the authority to change category labels and many other attributes in the item dimension.

A similar issue arises when any of the datamarts use an aggregate navigator that automatically substitutes a compressed item dimension table and an associated compressed fact table at the time the user specifies a query. For example, if the user asks for a "share" of a specific item to an entire item category, we usually perform two queries against the fact table and take the ratio in order to compute the share. The first query requests a very specific product and cannot use the compressed aggregate tables. But the second query is just getting the category total and is a prime candidate for aggregate navigation.

The moral of this story is that if the dimension authority has released a new item table, then all the aggregate tables affected by changes made in the item table must be adjusted. If some low level items were moved from one existing category to another, then not only is the item table changed, but any fact table at the category level would have to be adjusted.

We can summarize the two big responsibilities for correctly replicating dimensions:

- 1) All client data marts must deploy the replicated dimension simultaneously so that any end user drilling across these data marts will be using a consistent set of dimension attributes,

and

- 2) All client data marts must remove aggregates affected by changes in the dimension, and only make these aggregates available to the end users when they have been made completely consistent with the base fact tables and the new rollup logic.

This topic of Dimensional Replication is criterion #9 in my recommended list of dimensionally friendly criteria. Both Microsoft and Cognos have posted detailed responses to all twenty criteria that make up their "dimensionally friendly systems".

If you are an end user and you would like to rate your overall system perhaps supplied by several vendors, please contact me. I would be interested in adding such ratings to the web site.