

## Design Tip #83 Abstract Generic Dimensions

By Margy Ross

Childhood guessing games sometimes rely on the distinction of “person, place or thing” for early mystery-solving clues. Some modelers use these same characterizations in their data models by creating abstract person, place and/or thing (typically referred to as product) tables. While generalized tables appeal to the purist in all of us and may provide flexibility and reusability advantages for a data modeler, they often result in larger, more complicated dimension tables in the eyes of the business user.

Let’s consider a generic person or party dimension. Since our employees, customers and supplier contacts are all people, we should store them in the same dimension, right? Similarly, the same argument could be made for the geographic locations where our internal facilities, customers and suppliers reside. While a single table approach might seem clean and logical to data modelers and IT application developers, this abstraction often appears completely illogical to most business analysts for several reasons:

- 1) We collect and know vastly different information about our own internal entities than we do about external entities. Using a generic model means that some attributes are either nonsensical and/or unpopulated.
- 2) Generic attribute labels don’t supply adequate meaning and context for the business users and BI applications. For example, if a report illustrates sales by state, it’s unclear whether that refers to the state where the store is located or the state where the customer is located. It’s far preferable if the attribute is clearly labeled to denote its full meaning.
- 3) Lumping all varieties of people, places or products that our business interacts with inevitably results in larger dimension tables than if they were divided into more discreet logical entities.

Some of you might be familiar of a technique called “role-playing” where the same physical dimension table simultaneously serves multiple roles in the same fact table, such as the date dimension appearing as two uniquely labeled dimensions for the ship date and request date in a single fact table. Other role-playing examples might include the origin and destination airports, or the servicing and authorizing employees, or the dealer that sold the car versus the dealer maintaining it. In each of these examples, a single row in the dimension table could serve in multiple capacities. Role-playing is a very different concept than creating a generic dimension table that’s potentially the Cartesian product of all possible parties.

It’s worth noting that while generic dimensions are not appropriate for the dimensional model viewed by the business, we’re certainly not opposed to their usage in operational systems where they’re behind the scenes and not visible to the business. However, in most legacy situations, the operational source systems do not treat persons, places or things homogenously. It’s more likely that descriptive information about our facility locations, customer locations and supplier locations comes from a variety of different source systems. Attempting to create a generic dimension table from these various sources may create onerous integration challenges for the ETL team without any payback from a business user perspective.

Finally, let’s not forget the dimensional modelers’ mantra – easy to use and fast query performance.

In most cases, abstract dimension tables fail to deliver on either front because the abstraction process reduces clarity for the business users and inevitably creates a larger table. They should be avoided in the dimensional models that are presented to the business in your DW/BI architecture.