

Design Tip #124 Alternatives for Multi-valued Dimensions

By Joy Mundy

The standard relationship between fact and dimension tables is many-to-one: each row in a fact table links to one and only one row in the dimension table. In a detailed sales event fact table, each fact table row represents a sale of one product to one customer on a specific date. Each row in a dimension table, such as a single customer, usually points back to many rows in the fact table.

A dimensional design can encompass a more complex multi-valued relationship between fact and dimension. For example, perhaps our sales order entry system lets us collect information about why the customer chose a specific product (such as price, features, or recommendation). Depending on how the transaction system is designed, it's easy to see how a sales order line could be associated with potentially many sales reasons.

The robust, fully-featured way to model such a relationship in the dimensional world is similar to the modeling technique for a transactional database. The sales reason dimension table is normal, with a surrogate key, one row for each sales reason, and potentially several attributes such as sales reason name, long description, and type. In our simple example, the sales reason dimension table would be quite small, perhaps ten rows. We can't put that sales reason key in the fact table because each sales transaction can be associated with many sales reasons. The sales reason bridge table fills the gap. It ties together all the possible (or observed) sets of sales reasons: {Price, Price and Features, Features and Recommendation, Price and Features and Recommendation}. Each of those sets of reasons is tied together with a single sales reason group key that is propagated into the fact table.

For example, the figure below displays a dimensional model for a sales fact that captures multiple sales reasons:



If we have ten possible sales reasons, the Sales Reason Bridge table will contain several hundred rows.

The biggest problem with this design is its usability by ad hoc users. The multi-valued relationship, by its nature, effectively “explodes” the fact table. Imagine a poorly trained business user who attempts to construct a report that returns a list of sales reasons and sales amounts. It is absurdly easy to double count the facts for transactions with multiple sales reasons. The weighting factor in the bridge table is designed to address that issue, but the user needs to know what the factor is for and how to use it.

In the example we're discussing, sales reason is probably a very minor embellishment to a key fact table that tracks our sales. The sales fact table is used throughout the organization by many user communities, for both ad hoc and structured reporting. There are several approaches to the usability problem presented by the full featured bridge table design. These include:

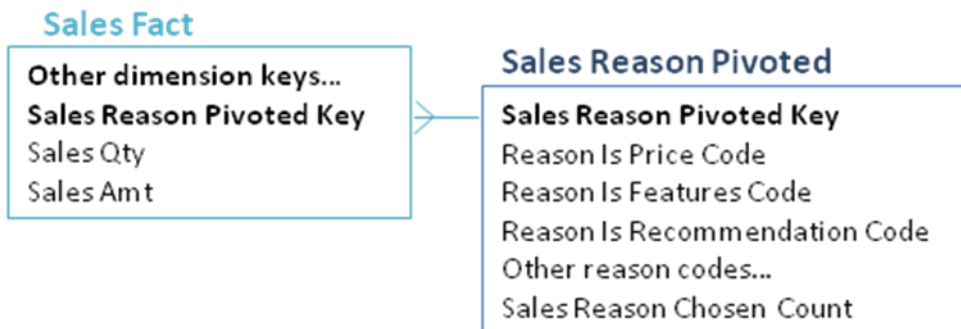
- *Hide the sales reason from most users.* You can publish two versions of the schema: the full one for use by structured reporting and a handful of power users, and a version that eliminates sales

reason for use by more casual users.

- *Eliminate the bridge table by collapsing multiple answers.* Add a row to the sales reason dimension table: "Multiple reasons chosen." The fact table can then link directly with the sales reason dimension. As with all design decisions, the IT organization cannot choose this approach without consulting with the user community. But you may be surprised to hear how many of your users would be absolutely fine with this approach. We've often heard users say "oh, we just collapse all multiple answers to a single one in Excel anyway." For something like a reason code (which has limited information value), this approach may be quite acceptable.

One way to make this approach more palatable is to have two versions of the dimension structure, and two keys in the fact table: the sales reason group key and the sales reason key directly. The view of the schema that's shared with most casual users displays only the simple relationship; the view for the reporting team and power users could also include the more complete bridge table relationship.

- *Identify a single primary sales reason.* It may be possible to identify a primary sales reason, either based on some logic in the transaction system or by way of business rules. For example, business users may tell you that if the customer chooses price as a sales reason, then from an analytic point of view, price is the primary sales reason. In our experience it's relatively unlikely that you can wring a workable algorithm from the business users, but it's worth exploring. As with the previous approach, you can combine this technique with the bridge table approach for different user communities.
- *Pivot out the sales reasons.* If the domain of the multi-choice space is small -- in other words, if you have only a few possible sales reasons -- you can eliminate the bridge table by creating a dimension table with one column for each choice. In the example we've been using, the sales reason dimension would have columns for price, features, recommendation, and each other sales reason. Each attribute can take the value yes or no. This schema is illustrated below:



This approach solves the fact table explosion problem, but does create some issues in the sales reason dimension. It's only practical with a relatively small number of domain values, perhaps 50 or 100. Every attribute in the original dimension shows up as an additional column for *each* domain value. Perhaps the biggest drawback is that any change in the domain (adding another sales reason) requires a change in the data model and ETL application.

Nonetheless, if the multi-valued dimension is important to the broad ad hoc user community, and you have a relatively small and static set of domain values, this approach may be more appealing than the bridge table technique. It's much easier for business users to construct meaningful queries.

Clearly the pivoted dimension table doesn't work for all multi-valued dimensions. The classic example of a multi-valued dimension -- multiple diagnoses for a patient's hospital visit -- has far too large a domain of possible values to fit in the pivoted structure.

The bridge table design approach for multi-valued dimensions, which Kimball Group has described many times over the past decades, is still the best. But the technique requires an educated user community,

and user education seems to be one of the first places the budget cutting axe is applied. In some circumstances, the usability problems can be lessened by presenting an alternative, simpler structure to the ad hoc user community.