

Design Tip #82 Turning the Fact Table on its Head

By Joy Mundy

The grain of the fact table most often comes directly from the grain of the transaction table from which the data is sourced. Occasionally it makes sense to pivot the facts so that we actually create *more* fact table rows than there are rows in the source.

This counterintuitive occurrence is most likely when the source system isn't a transaction system, such as one that captures sales events, but an analytic system like a forecasting, promotion, or financial analysis system. For example, let's say we're building a fact table to hold budget and actual data. Our source table was designed to support the budget process; it contains financial information by Month, Account and Department with facts ActualAmt and BudgetAmt. The beginning data modeler starts to create a similar structure in the data warehouse database. But interviews with the business community help us understand that there are several versions of budget. We need for the data warehouse to track several drafts of the budget during the budget development process.

The solution here is to create a fact table with four dimensions: Month, Account, Department and Scenario. The new Scenario dimension would have a handful of rows, including "Actual," "Budget Draft 1 FY07," and "Final Budget FY07." Our fact table contains only one measure, Amount. The fact table is normalized to be "long and skinny" rather than "short and fat." The new structure is more flexible and easily accommodates an arbitrary number of draft and final budgets.

It's easy for an outsider to see this solution. It's surprising how difficult it can be, while in the trenches, to pull your mind away from the structure you've been handed and think about creative alternatives. Here are some hints that you should think about this approach to pivot the fact table and add a new dimension.

1. Excessive number of facts. What does excessive mean? A hundred facts in a fact table are excessive; ten is not. Somewhere in the middle, perhaps around thirty measures, you cross into the grey area towards excessive.
2. Naming conventions to group measures. If you have a ton of facts, your fact column names probably use prefixes and suffixes to help users find the facts they're looking for.
3. Many measures in a row are null. Of the, say, 100 facts that could apply to a row, only a subset of them tend to be populated at any one time.

If all these conditions are true, consider normalizing the fact table by creating a fact dimension. The fact dimension could contain several columns that would help users navigate the list of facts. Admittedly, our long-skinny fact table will have a lot more rows than the short-fat one. It will use somewhat more disk space as well, though depending on the sparsity of facts, it may not be all that much bigger. The biggest downside is that many users want to see the facts on a row. It would require pretty good SQL skills to formulate the query to unpivot the data back to get several measures on the same row in a report. Luckily we don't need to write this SQL; a query or reporting tool can do this work for us. And if we use an OLAP database as the user presentation layer, this notion of a fact dimension is entirely natural.