

## Kimball Design Tip #2: Multiple Time Stamps

By Ralph Kimball

The most frequent question I get in my classes and e-mails is how to handle multiple time stamps on a fact table record. Although the correct and immediate answer is "make each time stamp a Time dimension", it is worth describing this approach carefully because it nicely illustrates a whole set of modern data warehouse design techniques, which I will emphasize in CAPS.

First, choose the **FUNDAMENTAL GRAIN** of your fact table. Every fact table I have ever designed has been a transaction grain, a periodic snapshot grain, or an accumulating snapshot grain. See the article [www.intelligententerprise.com/993003/warehouse.shtml](http://www.intelligententerprise.com/993003/warehouse.shtml) for more on fundamental grains. When you understand the fundamental grain, you will be able to judge what the meaning and relevance of multiple time stamps is to that grain. Multiple time stamps arise most frequently when the fact record is an accumulating snapshot, where the fact record represents the complete history of an order line, for example. The multiple time stamps may represent

- 1) original order date
- 2) requested delivery date
- 3) actual ship date
- 4) actual delivery date, and
1. 5) return date.

Second, for the above order line example, create five **ROLES** for a single underlying Time dimension. See the article [www.dbmsmag.com/9708d05.html](http://www.dbmsmag.com/9708d05.html) for more on roles. This means five fields in the fact table, where each field is a good foreign key linking to a dimension. The single underlying Time dimension is "exposed" to the fact table through five views, which makes each instance of the Time dimension semantically independent, as they must be.

Third, make sure that the actual foreign keys in the fact table are proper **SURROGATE KEYS**. In other words, they aren't literal SQL date/time stamps, but rather are simple anonymous integers. Resist all urges to put meaning or ordering in these keys! See the articles [www.dbmsmag.com/9805d05.html](http://www.dbmsmag.com/9805d05.html) and [www.dbmsmag.com/9806d05.html](http://www.dbmsmag.com/9806d05.html) for more on surrogate keys. If you think carefully about our order line example, you will have to agree that some order line records must contain "unknown" or "hasn't happened yet" time stamps. This is one of the classic reasons for using surrogate keys.

If your time stamps are accurate to the minute or second, then you need to split the calendar day off from the time of day and make them separate dimensions. We'll discuss variations on these time stamp designs in a future design tip.