

Slowly Changing Dimensions

Up to this point we have pretended that each dimension is logically independent from all the other dimensions. In particular, dimensions have been assumed to be independent of time. Unfortunately, this is not the case in the real world. While dimension table attributes are relatively static, they are not fixed forever. Dimension attributes change, albeit rather slowly, over time. Dimensional designers must engage business representatives proactively to help determine the appropriate change-handling strategy. We can't simply jump to the conclusion that the business doesn't care about dimension changes just because its representatives didn't mention it during the requirements process. While we're assuming that accurate change tracking is unnecessary, business users may be assuming that the data warehouse will allow them to see the impact of each and every dimension change. Even though we may not want to hear that change tracking is a must-have because we are not looking for any additional development work, it is obviously better to receive the message sooner rather than later.

When we need to track change, it is unacceptable to put everything into the fact table or make every dimension time-dependent to deal with these changes. We would quickly talk ourselves back into a full-blown normalized structure with the consequential loss of understandability and query performance. Instead, we take advantage of the fact that most dimensions are nearly constant over time. We can preserve the independent dimensional structure with only relatively minor adjustments to contend with the changes. We refer to these nearly constant dimensions as *slowly changing dimensions*. Since Ralph Kimball first introduced the notion of slowly changing dimensions in 1994, some IT professionals—in a never-ending quest to speak in acronymese—have termed them *SCDs*.

For each attribute in our dimension tables, we must specify a strategy to handle change. In other words, when an attribute value changes in the operational world, how will we respond to the change in our dimensional models? In the following section we'll describe three basic techniques for dealing with attribute changes, along with a couple hybrid approaches. You may decide that you need to employ a combination of these techniques within a single dimension table.

Type 1: Overwrite the Value

With the type 1 response, we merely overwrite the old attribute value in the dimension row, replacing it with the current value. In so doing, the attribute always reflects the most recent assignment.