

Kimball Design Tip #54: Delivering Both Historical and Current Perspectives

By Margy Ross

As with most things in life, change is inevitable with dimension attributes. Most Design Tip readers are familiar with the three basic slowly changing dimension (SCD) techniques:

Type 1: Overwrite the attribute

Type 2: Add another dimension row

Type 3: Add another dimension attribute

If this is news to you, take a look at Ralph's April 22, 2003 [Intelligent Enterprise](#) column, "[Soul of the Data Warehouse Part 3: Handling Time.](#)"

When dimension attributes change, we are often asked to preserve the historically-accurate values, as well as provide the ability to roll-up historical facts based on the current characteristics. Demand for this capability is growing as the business intelligence community matures analytically. Twenty years ago, analysts were satisfied with dimension tables that were refreshed (overwritten) with current attributes at every load. Then the pendulum swung to completely and accurately capture every change using SCD Type 2. Now more people want to have their cake (or steak) and eat it, too.

We discussed a hybrid approach for supporting this requirement several years ago with [Design Tip #15: Combining SCD Techniques](#). In that Tip, we issued Type 2 rows to capture historical attribute changes, with one or more complementary Type 3 "current" attributes on each row. The Type 3 attribute is overwritten (treated as a Type 1) for the current and all previous Type 2 rows, so analysts can query on either the historically-accurate attributes or current assignments. More flexibility is delivered with this hybrid SCD approach, albeit with added complexity.

Physically, this technique could be implemented in a single dimension table with two columns (historical "as was" and current) for each attribute requiring this flexibility. Alternatively, you could handle all the current attributes in an outrigger table joined to the natural key of the dimension (such as the Employee ID), as opposed to the dimension surrogate key. The outrigger contains just one row of current data for each natural key in the dimension table; the attributes are overwritten whenever change occurs. The same natural key likely appears on multiple Type 2 dimension rows with unique surrogate keys. In the spirit of ease-of-use, the core dimension and outrigger of current values may appear as one via a view, however this approach is unacceptable if performance is negatively impacted.

If you have a large dimension table with a hundred attributes requiring historical and current tracking, the above techniques can become onerous. In this situation, you should consider including the dimension natural key as an additional fact table foreign key. You now have two similar, yet very different dimension tables associated with the fact data. First, the dimension surrogate key joins to a typical dimension with historically-accurate Type 2 data. These attributes filter or group facts by the attribute values in effect when the fact data was loaded. Secondly, the dimension natural key (or a static reference key) joins to a dimension table with just the current Type 1 values. The column labels in this table should be prefaced with "current" to reduce the risk of user confusion. These dimension attributes are used to summarize or filter facts based on the current profile, regardless of the values in effect when the fact row was loaded.

In all the situations described in this Design Tip, the query answer set may be quite different depending on which dimension table attributes are constrained or grouped by. Different results are

inevitable because different questions are being asked. However, given the vulnerability to error or misinterpretation, this capability is often reserved for the segment of the user community that understands these inherent differences.