Slowly Changing Dimensions Are Not Always as Easy as 1, 2, 3

  By Margy Ross (https://www.kimballgroup.com/author/margy/)

  March 10, 2005

How do you deal with changing dimensions? Hybrid approaches fill gaps left by the three fundamental techniques.

Unlike most OLTP systems, a major objective of a data warehouse is to track history. So, accounting for change is one of the analyst's most important responsibilities. A sales force region reassignment is a good example of a business change that may require you to alter the dimensional data warehouse. We'll discuss how to apply the right technique to account for the change historically. Hang on to your hats — this is not an easy topic.

Data warehouse design presumes that facts, such as customer orders or product shipments, will accumulate quickly. But the supporting dimension attributes of the facts, such as customer name or product size, are comparatively static. Still, most dimensions are subject to change, however slow. When dimensional modelers think about changing a dimension attribute, the three elementary approaches immediately come to mind: slowly changing dimension (SCD) types 1, 2 and 3.

As a quick review... Type 1 is most appropriate when processing corrections; this technique won't preserve historically accurate associations. The changed attribute is simply updated (overwritten) to reflect the most current value.With a type 2 change, a new row with a new surrogate primary key is inserted into the dimension table to capture changes. Both the prior and new rows contain as attributes the natural key (or durable identifier), the most-recent-row flag and the row effective and expiration dates.With type 3, another attribute is added to the existing dimension row to support analysis based on either the new or prior attribute value. This is the least commonly needed technique.

These three fundamental techniques are adequate for most situations. However, what happens when you need variations that build on these basics to serve more analytically mature data warehouse users? Business folks sometimes want to preserve the historically accurate dimension attribute associated with a fact (such as at the time of sale or claim), but maintain the option to roll up historical facts based on current dimension characteristics. That's when you need hybrid variations of the three main types. We'll lead you through the hybrids in this column.

The Mini Dimension with "Current" Overwrite

When you need historical tracking but are faced with semi-rapid changes in a large dimension, pure type 2 tracking is inappropriate. If you use a mini dimension, you can isolate volatile dimension attributes in a separate table rather than track changes in the primary dimension table directly. The mini-dimension grain is one row per "profile," or combination of attributes, while the grain of the primary dimension might be one row per customer. The number of rows in the primary dimension may be in the millions, but the number of mini-dimension rows should be a fraction of that. You capture the evolving relationship between the mini dimension and primary dimension in a fact table. When a business event (transaction or periodic snapshot) spawns a fact row, the row has one foreign key for the primary dimension and another for the mini-dimension profile in effect at the time of the event.

Profile changes sometimes occur outside of a business event, for example when a customer's geographic profile is updated without a sales transaction. If the business requires accurate point-in-time profiling, a supplemental factless fact table with effective and expiration dates can capture every relationship change between the primary and profile dimensions. One more embellishment with this technique is to add a "current profile" key to the primary dimension. This is a type 1 attribute, overwritten with every profile change, that's useful for analysts who want current profile counts regardless of fact table metrics or want to roll up historical facts based on the current profile. You'd logically represent the primary dimension and profile outrigger as a single table to the presentation layer if doing so doesn't harm performance. To minimize user confusion and error, the current attributes should have

column names that distinguish them from the mini-dimension attributes. For example, the labels should indicate whether a customer's marketing segment designation is a current one or an obsolete one that was effective when the fact occurred — such as "historical marital status at time of event" in the profile mini dimension and "current marital status" in the primary customer dimension.

Type 2 with "Current" Overwrite

Another variation for tracking unpredictable changes while supporting rollup of historical facts to current dimension attributes is a hybrid of type 1 and type 2. In this scenario, you capture a type 2 attribute change by adding a row to the primary dimension table. In addition, you have a "current" attribute on each row that you overwrite (type 1) for the current and all previous type 2 rows. You retain the historical attribute in its own original column. When a change occurs, the most current dimension row has the same value in the uniquely labeled current and historical ("as was" or "at time of event") columns.

You can expand this technique to cover not just the historical and current attribute values but also a fixed, end-of-period value as another type 1 column. Although it seems similar, the end-of-period attribute may be unique from either the historical or current perspective. Say a customer's segment changed on Jan. 5 and the business wants to create a report on Jan. 10 to analyze last period's data based on the customer's Dec. 31 designation. You could probably derive the right information from the row effective and expiration dates, but providing the end-of-period value as an attribute simplifies the query. If this query occurs frequently, it's better to have the work done once, during the ETL process, rather than every time the query runs. You can apply the same logic to other fixed characteristics, such as the customer's original segment, which never changes. Instead of having the historical and current attributes reside in the same physical table, the current attributes could sit in an outrigger table joined to the dimension natural key. The same natural key, such as Customer ID, likely appears on multiple type 2 dimension rows with unique surrogate keys. The outrigger contains just one row of current data for each natural key in the dimension table; the attributes are overwritten whenever change occurs. To promote ease of use, have the core dimension and outrigger of current values appear to the user as one — unless this hurts application performance.

Type 2 with Natural Keys in the Fact Table

If you have a million-row dimension table with many attributes requiring historical and current tracking, the last technique we described becomes overly burdensome. In this situation, consider including the dimension natural key as a fact table foreign key, in addition to the surrogate key for type 2 tracking. This technique gives you essentially two dimension tables associated with the facts, but for good reason. The type 2 dimension has historically accurate attributes for filtering or grouping based on the effective values when the fact table was loaded. The dimension natural key joins to a table with just the current type 1 values. Again, the column labels in this table should be prefaced with "current" to reduce the risk of user confusion. You use these dimension attributes to summarize or filter facts based on the current profile, regardless of the values in effect when the fact row was loaded. Of course, if the natural key is unwieldy or ever reassigned, then you should use a durable surrogate reference key instead.

This approach delivers the same functionality as the type 2 with "current" overwrite technique we discussed earlier; that technique spawns more attribute columns in a single dimension table, while this approach relies on two foreign keys in the fact table.

While it's uncommon, we're sometimes asked to roll up historical facts based on any specific point-in-time profile, in addition to reporting them by the attribute values when the fact measurement occurred or by the current dimension attribute values. For example, perhaps the business wants to report three years of historical metrics based on the attributes or hierarchy in effect on Dec. 1 of last year. In this case, you can use the dual dimension keys in the fact table to your advantage. You first filter on the type 2 dimension table's row effective and expiration dates to locate the attributes in effect at the desired point in time. With this constraint, a single row for each natural or durable surrogate key in the type 2 dimension has been identified. You can then join to the natural/durable surrogate dimension key in the fact table to roll up any facts based on the point-in-time attribute values. It's as if you're defining the meaning of "current" on the fly. Obviously, you must filter on the type 2 row dates or you'll have multiple type 2 rows for each natural key, but that's fundamental in the business's requirement to report any history on any specified point-in-time attributes. Finally, only unveil this capability to a limited, highly analytic audience. This embellishment is not for the faint of heart.

Series of Type 3 Attributes

Say you have a dimension attribute that changes with a predictable rhythm, such as annually, and the business needs to summarize facts based on any historical value of the attribute (not just the historically accurate and current, as we've primarily been discussing). For example, imagine the product line is recategorized at the start of every fiscal year and the business wants to look at multiple years of historical product sales based on the category assignment for the current year or any prior year.

This situation is best handled with a series of type 3 dimension attributes. On every dimension row, have a "current" category attribute that can be overwritten, as well as attributes for each annual designation, such as "2004 category" and "2003 category." You can then group historical facts based on any annual categorization.

This seemingly straightforward technique isn't appropriate for the unpredictable changes we described earlier. Customer attributes evolve uniquely. You can't add a series of type 3 attributes to track the prior attribute values ("prior-1," "prior-2" and so on) for unpredictable changes, because each attribute would be associated with a unique point in time for nearly every row in the dimension table.

Balance Power against Ease of Use

Before using hybrid techniques to support sophisticated change tracking, remember to maintain the equilibrium between flexibility and complexity. Users' questions and answer sets will vary depending on which dimension attributes are used for constraining or grouping. Given the potential for error or misinterpretation, hide the complexity (and associated capabilities) from infrequent users.

About the Author: Margy Ross (https://www.kimballgroup.com/author/margy/)

(https://www.kimballgroup.com/author/margy/)

✉ (mailto:margyross1@gmail.com)

With bio

Margy is President of the Kimball Group and has focused exclusively on decision support and data warehousing for more than 30 years, specializing in program/project strategy, business requirements analysis, and dimensional modeling.

Categories

Before Diving In (https://www.kimballgroup.com/category/before-diving-in/)
Project/Program Planning (https://www.kimballgroup.com/category/project-program-planning/)
Requirements Definition (https://www.kimballgroup.com/category/requirements-definition/)
Data Architecture (https://www.kimballgroup.com/category/data-architecture/)
Dimensional Modeling Fundamentals (https://www.kimballgroup.com/category/dimensional-modeling-fundamentals/)
Dimensional Modeling Tasks (https://www.kimballgroup.com/category/dimensional-modeling-tasks/)
Fact Table Core Concepts (https://www.kimballgroup.com/category/fact-table-core-concepts/)
Dimension Table Core Concepts (https://www.kimballgroup.com/category/dimension-table-core-concepts/)
Advanced Dimension Patterns & Case Studies (https://www.kimballgroup.com/category/advanced-dimension-patterns-case-studies

ETL and Data Quality (https://www.kimballgroup.com/category/etl-and-data-quality/)

Technical Architecture (https://www.kimballgroup.com/category/technical-architecture/)

Business Intelligence Applications (https://www.kimballgroup.com/category/business-intelligence-applications/)

Archives

	2005 (https://www.kimballgroup.com/2005/)
	(https://www.kimballgroup.com/2004/) 2004 (https://www.kimballgroup.com/2004/)
	(https://www.kimballgroup.com/2003/) 2003 (https://www.kimballgroup.com/2003/)
	(https://www.kimballgroup.com/2002/) 2002 (https://www.kimballgroup.com/2002/)
	(https://www.kimballgroup.com/1999/) 1999 (https://www.kimballgroup.com/1999/)
	(https://www.kimballgroup.com/1998/) 1998 (https://www.kimballgroup.com/1998/)
	(https://www.kimballgroup.com/1997/) 1997 (https://www.kimballgroup.com/1997/)
	(https://www.kimballgroup.com/1995/) 1995 (https://www.kimballgroup.com/1995/)