The 10 Essential Rules of Dimensional Modeling

kimballgroup.com/2009/05/the-10-essential-rules-of-dimensional-modeling/

By Margy Ross May 29, 2009

A student attending one of Kimball Group's recent onsite dimensional modeling classes asked me for a list of "Kimball's Commandments" for dimensional modeling. We'll refrain from using religious terminology, but let's just say the following are not-to-be-broken rules together with less stringent rule-of-thumb recommendations.

Rule #1: Load detailed atomic data into dimensional structures.

Dimensional models should be populated with bedrock atomic details to support the unpredictable filtering and grouping required by business user queries. Users typically don't need to see a single record at a time, but you can't predict the somewhat arbitrary ways they'll want to screen and roll up the details. If only summarized data is available, then you've already made assumptions about data usage patterns that will cause users to run into a brick wall when they want to dig deeper into the details. Of course, atomic details can be complemented by summary dimensional models that provide performance advantages for common queries of aggregated data, but business users cannot live on summary data alone; they need the gory details to answer their ever-changing questions.

Rule #2: Structure dimensional models around business processes.

Business processes are the activities performed by your organization; they represent measurement events, like taking an order or billing a customer. Business processes typically capture or generate unique performance metrics associated with each event. These metrics translate into facts, with each business process represented by a single atomic fact table. In addition to single process fact tables, consolidated fact tables are sometimes created that combine metrics from multiple processes into one fact table at a common level of detail. Again, consolidated fact tables are a complement to the detailed single-process fact tables, not a substitute for them.

Rule #3: Ensure that every fact table has an associated date dimension table.

The measurement events described in Rule #2 always have a date stamp of some variety associated with them, whether it's a monthly balance snapshot or a monetary transfer captured to the hundredth of a second. Every fact table should have at least one foreign key to an associated date dimension table, whose grain is a single day, with calendar attributes

and nonstandard characteristics about the measurement event date, such as the fiscal month and corporate holiday indicator. Sometimes multiple date foreign keys are represented in a fact table.

Rule #4: Ensure that all facts in a single fact table are at the same grain or level of detail.

There are three fundamental grains to categorize all fact tables: transactional, periodic snapshot, or accumulating snapshot. Regardless of its grain type, every measurement within a fact table must be at the exact same level of detail. When you mix facts representing multiple levels of granularity in the same fact table, you are setting yourself up for business user confusion and making the BI applications vulnerable to overstated or otherwise erroneous results.

Rule #5: Resolve many-to-many relationships in fact tables.

Since a fact table stores the results of a business process event, there's inherently a many-to-many (M:M) relationship between its foreign keys, such as multiple products being sold in multiple stores on multiple days. These foreign key fields should never be null. Sometimes dimensions can take on multiple values for a single measurement event, such as the multiple diagnoses associated with a health care encounter or multiple customers with a bank account. In these cases, it's unreasonable to resolve the many-valued dimensions directly in the fact table, as this would violate the natural grain of the measurement event. Thus, we use a many-to-many, dual-keyed bridge table in conjunction with the fact table.

Rule #6: Resolve many-to-one relationships in dimension tables.

Hierarchical, fixed-depth many-to-one (M:1) relationships between attributes are typically denormalized or collapsed into a flattened dimension table. If you've spent most of your career designing entity-relationship models for transaction processing systems, you'll need to resist your instinctive tendency to normalize or snowflake a M:1 relationship into smaller subdimensions; dimension denormalization is the name of the game in dimensional modeling.

It is relatively common to have multiple M:1 relationships represented in a single dimension table. One-to-one relationships, like a unique product description associated with a product code, are also handled in a dimension table. Occasionally many-to-one relationships are resolved in the fact table, such as the case when the detailed dimension table has millions of rows and its roll-up attributes are frequently changing. However, using the fact table to resolve M:1 relationships should be done sparingly.

Rule #7: Store report labels and filter domain values in dimension tables.

The codes and, more importantly, associated decodes and descriptors used for labeling and query filtering should be captured in dimension tables. Avoid storing cryptic code fields or bulky descriptive fields in the fact table itself; likewise, don't just store the code in the dimension table and assume that users don't need descriptive decodes or that they'll be handled in the BI application. If it's a row/column label or pull-down menu filter, then it should be handled as a dimension attribute.

Though we stated in Rule #5 that fact table foreign keys should never be null, it's also advisable to avoid nulls in the dimension tables' attribute fields by replacing the null value with "NA" (not applicable) or another default value, determined by the data steward, to reduce user confusion if possible.

Rule #8: Make certain that dimension tables use a surrogate key.

Meaningless, sequentially assigned surrogate keys (except for the date dimension, where chronologically assigned and even more meaningful keys are acceptable) deliver a number of operational benefits, including smaller keys which mean smaller fact tables, smaller indexes, and improved performance. Surrogate keys are absolutely required if you're tracking dimension attribute changes with a new dimension record for each profile change. Even if your business users don't initially visualize the value of tracking attribute changes, using surrogates will make a downstream policy change less onerous. The surrogates also allow you to map multiple operational keys to a common profile, plus buffer you from unexpected operational activities, like the recycling of an obsolete product number or acquisition of another company with its own coding schemes.

Rule #9: Create conformed dimensions to integrate data across the enterprise.

Conformed dimensions (otherwise known as common, master, standard or reference dimensions) are essential for enterprise data warehousing. Managed once in the ETL system and then reused across multiple fact tables, conformed dimensions deliver consistent descriptive attributes across dimensional models and support the ability to drill across and integrate data from multiple business processes. The Enterprise Data Warehouse Bus Matrix is the key architecture blueprint for representing the organization's core business processes and associated dimensionality. Reusing conformed dimensions ultimately shortens the time-to-market by eliminating redundant design and development efforts; however, conformed dimensions require a commitment and investment in data stewardship and governance, even if you don't need everyone to agree on every dimension attribute to leverage conformity.

Rule #10: Continuously balance requirements and realities to deliver a DW/BI solution that's accepted by business users and that supports their decision-making.

Dimensional modelers must constantly straddle business user requirements along with the underlying realities of the associated source data to deliver a design that can be implemented and that, more importantly, stands a reasonable chance of business adoption.

The requirements-versus-realities balancing act is a fact of life for DW/BI practitioners, whether you're focused on the dimensional model, project strategy, technical/ETL/BI architectures or deployment/maintenance plan.

If you've read our Intelligent Enterprise articles, Toolkit books or Design Tips regularly, these rules shouldn't be news to you, but here we've consolidated our rules into a single rulebook that you can refer to when you are gathered to design or review your models.

Good luck!