Design Tip #21: Declaring The Grain

By Ralph Kimball

The most important step in a dimensional design is declaring the grain of the fact table. Declaring the grain means saying EXACTLY what a fact table record represents. Remember that a fact table record captures a measurement. Example declarations include:

* an individual line item on a customer's retail sales ticket as measured by a scanner device
* an individual transaction against an insurance policy
* a line item on a bill received from a doctor
* an individual boarding pass used by someone making an airplane flight

When you make such a grain declaration, you can have a very precise discussion of which dimensions are possible and which are not. For example, a line item of a doctor's bill (example #3) arguably would have the following dimensions:

* Date (of treatment)
* Doctor (maybe called "provider")
* Patient
* Procedure
* Primary Diagnosis
* Location (presumably the doctor's office)
* Billing Organization (an organization the doctor belongs to)
* Responsible Party (either the patient, or the patient's legal guardian)
* Primary Payer (often an insurance plan)
* Secondary Payer (maybe the responsible party's spouse's insurance plan) and quite possibly others.

If you have been following this example, I hope you have noticed some powerful effects from declaring the grain. First, we can visualize the dimensionality of the doctor-bill-line-item very precisely and we can have Yes/No discussions relative to our data sources about whether a dimension can be attached to this data. For example, we probably would exclude "treatment outcome" from this example because most medical billing data doesn't tie to any notion of outcome.

BUT, a general E/R oriented "data model" of doctor visits might well include treatment outcome. After all, in an abstract sense, doesn't every treatment have an outcome???

The discipline of insisting on the grain declaration at the beginning of a dimensional design keeps you from making this kind of mistake. A model of billable doctor visits that included treatment outcome would look like a dimensional model but it wouldn't be implementable. This is my main gripe with many of the current offerings of "standard schemas" in books and CDs. Since they have no
grain discipline, they often combine entities that don't exist together in real data sources.

A second major insight from the doctor-bill-line-item grain declaration is that this very atomic grain gives rise to a lot o

dimensions! We have listed 10 dimensions above, and those of you who are experts in health care billing probably know of a couple more. It is an interesting realization that the smaller and more atomic the measurement (fact table record), the more things you know for sure.
And hence the more dimensions! This is another way of explaining why atomic data resists the “ad hoc attack” by end users. Atomic data has the most dimensionality and so it can be constrained and rolled up in every way that is possible for that data source. Atomic data is a perfect match for the dimensional approach.

All of the grain declarations listed at the beginning of this design tip represent the lowest possible granularity of their respective data sources. These data measurements are “atomic” and cannot be divided further. But it is quite possible to declare higher level grains for each of these data sources that represent aggregations of atomic data:

* all the sales for a product in a store on a day
* insurance policy transaction totals by month by line of business
* charged amount totals by treatment by diagnosis by month
* counts of passengers and other flight customer satisfaction issues by route by month

These higher levels of aggregation will almost always have fewer, smaller dimensions. Our doctor example might end up with only the dimensions of

Month
Doctor
Procedure
Diagnosis

It would be undesirable in an aggregated fact table to include all the original dimensions of the atomic data because you would usually end up with very little aggregation!

Since useful aggregations necessarily shrink dimensions and remove dimensions, this leads to the realization that aggregated data always needs to be used in conjunction with its base atomic data, because aggregated data has less dimensional detail. Some authors get confused on this point, and after declaring that datamarts necessarily consist of aggregated data, they criticize the datamarts for
“anticipating the business question”. All of this misunderstanding goes away when aggregated data is made available TOGETHER with the atomic data from which it is derived.

The most important result of declaring the grain of the fact table is anchoring the discussion of the dimensions. But declaring the grain allows you to be equally clear about the measured numeric facts. Simply put, the facts must be true to the grain. In our doctor example, the most obvious measured fact would be “billed amount”. Other facts relating to the specific treatment received by that patient at that time are possible. But “helpful” facts like amount billed year-to-date to this patient for all treatments are not true to the grain. When fact records are combined in arbitrary ways by a reporting application, these untrue-to-the-grain facts produce nonesensical, useless results. They need to be left out of the design. Calculate such aggregate measures in your application.

In summary, try to do your dimensional designs using the following four steps, in order:

1. decide on your sources of data
2. declare the grain of the fact table (preferably at the most atomic level)
3. add dimensions for “everything you know” about this grain
4. add numeric measured facts true to the grain.

Write to me with questions or comments about declaring the grain.