

## Kimball Design Tip #20: Sparse Facts And Facts With Short Lifetimes

By Ralph Kimball

Fact tables are built around numerical measurements. When a measurement is taken, a fact record comes into existence. The measurement can be the amount of a sale, the value of a transaction, a running balance at the end of a month, the yield of a manufacturing process run, or even a classic laboratory measurement. If we record several numbers at the same time, we can often put them together in the same fact record.

We surround the measurement(s) with all the things we know to be true at the precise moment of the measurement. Besides a time stamp, we often know things like customer, product, market condition, employee, status, supplier, and many other entities depending on the process supplying us the measurement.

We package all the things we know into descriptive text-laden dimension records and connect the facts to the dimension records through a foreign key / primary key (FK/PK) relationship.

This leads to the classic organization of a fact table (shown here with N dimensions and two facts called Dollars and Units):

```
dimkey1 (FK)
dimkey2 (FK)
dimkey3 (FK)
..
dimkeyN (FK)
Dollars
Units
```

The Dollars and Units fields are reserved placeholders for those specific measurements. This design carries the implicit assumptions that

- 1) these two measures are usually present together,
- 2) these are the only measures in this process
- 3) there are lots of measurement events, in other words, it is worthwhile to devote this fixed format table to these measures.

But what happens when all three of these assumptions break down? This happens frequently in complex financial investment tracking where every investment instrument has idiosyncratic measures. It also happens in industrial manufacturing processes where the batch runs are short and each batch type has a host of special measures. And finally, clinical and medical lab environments are dominated by hundreds of special measurements, none of which occur very frequently. All three of these examples can be described as "sparse facts".

You can't just extend the classic fact table design to handle sparse facts. You would have an unworkably long list of fact fields, most of which would be null in a given record.

The answer is to add a special "fact dimension" and shrink the list of actual numeric facts down to a

single AMOUNT field:

```
dimkey1 (FK)
dimkey2 (FK)
dimkey3 (FK)
..
dimkeyN (FK)
factkey (FK) <== additional dimension
Amount
```

The "fact dimension" describes the meaning of the measurement amount. It contains what used to be the field name of the fact, as well as the unit of measure, and any additivity restrictions. For instance, if the measurement is an inventory-like (or balance-like) fact, then it may be fully additive across all the dimensions except time. But if it is a full blown intensity measurement like temperature, then it is completely non-additive. Summarizing across non-additive dimensions requires averaging, not summing.

This approach is elegant because it is superbly flexible. You add new measurement types just by adding new records in the fact dimension, not by altering the structure of the table. You also eliminate all the NULLs in the classic design because a record only exists if the measurement exists.

But there are some significant tradeoffs. You may be generating a LOT of records. If some of your measurements give you 10 numeric results, now you have 10 records rather than the single record you had in the classic design. For extremely sparse situations, this is a great compromise. But as the density of the facts grows in the dimensional space you have created, you start papering the universe with records. At some point you have to return to the classic format.

This approach also makes applications more complicated. Combining two numbers that have been taken as part of a single measurement event is more difficult because now you have to fetch two records. SQL makes this awkward because SQL likes to do arithmetic WITHIN a record, not across records. And you have to be very careful that you don't mix incompatible Amounts in a calculation, since all the numeric measures exist within the single Amount field.

But these tradeoffs are clearly worth it if you live in the investment world, the manufacturing world, or the clinical/laboratory world.

Write to me about variations you have used on this theme and I'll talk about them in a future Design Tip.