

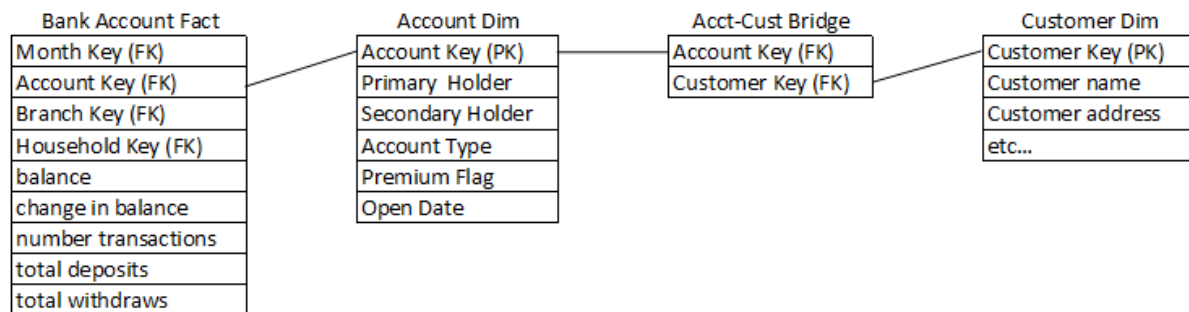
Design Tip #136 Adding a Mini-Dimension to a Bridge Table

By Ralph Kimball

Experienced dimensional modelers are familiar with the challenge of attaching a many-valued dimension to an existing fact table. This occurs when the grain of the fact table is compelling and obvious, yet one of the dimensions possesses many values at that grain. For example, in the doctor's office a line item on the doctor bill is created when a procedure is performed. The grain of individual line item is the most natural grain for a fact table representing doctor bills. Obvious dimensions include date, provider (doctor), patient, location, and procedure. But the diagnosis dimension is frequently many-valued.

Another common example is the bank account periodic snapshot, where the grain is month by account. The obvious dimensions in this case are month, account, branch, and household. But how do we attach individual customers to this grain since there may be "many" customers on a given account?

The solution in both cases is a bridge table that contains the many-to-many relationship needed. For the bank account example it looks like this:



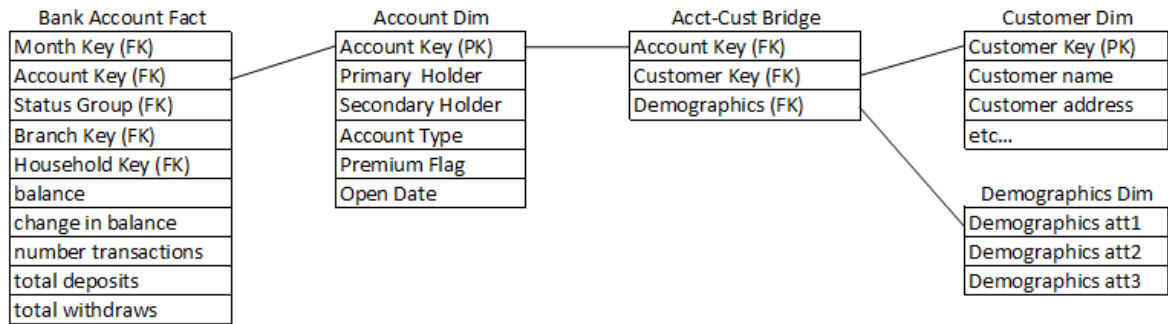
The account-to-customer bridge table is what relational theorists call an "associative" table. Its primary key is the combination of the two foreign keys to the account dimension and the customer dimension. On close examination, we discover that all these bridge table examples turn out to be associative tables linking two dimensions.

In the bank account example, this bridge table can get very large. If we have 20 million accounts and 25 million customers, the bridge table can grow to hundreds of millions of rows after a few years, if both the account dimension and the customer dimension are slowly changing type 2 dimensions (where we track history in these dimensions by issuing new records with new keys whenever there is a change).

Now the experienced dimensional modeler asks "what happens when my customer dimension turns out to be a so-called rapidly changing monster dimension?" This could happen when rapidly changing demographics and status attributes are added to the customer dimension, forcing numerous Type 2 additions to the customer dimension. Now the 25 million row customer dimension threatens to become several hundred million rows.

The standard response to a rapidly changing monster dimension is to split off the rapidly changing demographics and status attributes into a mini-dimension, which we will call the demographics dimension. This works great when this dimension attaches directly to the fact table along with a dimension like customer, because it stabilizes the large customer dimension and keeps it from growing every time there is a demographics or status change. But can we get this same advantage when the customer dimension is attached to a bridge table, as in the bank account example?

The solution is to add a foreign key reference in the bridge table to the demographics dimension, like this:



The way to visualize the bridge table is that for every account, the bridge table links to each customer and each customer's demographics for that account. The key for the bridge table itself now consists of the account key, customer key, and demographics key.

Depending on how frequently new demographics are assigned to each customer, the bridge table will grow, perhaps significantly. In the above design, since the grain of the root bank account fact table is month by account, the bridge table should be limited to changes recorded only at month ends. This will take some of the pressure off the bridge table. In my classes, as designs become more complex, I usually say at some point that "it's not Ralph's fault." The more meticulously we track changing customer behavior, the bigger our tables get. It always helps to add more RAM...