



## Kimball Design Tip #13: When A Fact Table Can Be Used As A Dimension Table

By Ralph Kimball

Fact tables come in three main flavors. The grain of a fact table can be an individual transaction, where a fact table record represents an instant in time. Or, the grain can be a periodic snapshot, representing a predictable duration of time like a week or a month. Or finally, the grain can be an accumulating snapshot, representing the entire history of something up to the present. I discussed these three types in depth in an IE article, which you can find at [www.intelligententerprise.com/993003/warehouse.shtml](http://www.intelligententerprise.com/993003/warehouse.shtml).

The first fact table type, the instantaneous transaction, may give us an opportunity to capture the description of something at an exact moment. Suppose that we have a series of transactions against the customer information in your bank account. In other words, an agent in the bank periodically makes changes to your name, address, phone number, customer classification, credit rating, risk rating, and other descriptors. The transaction grained fact table that captures these transactions might look like

Cust Info Transaction Date (FK)	<<== foreign key
Account (SK)	<<== surrogate key
Responsible Agent (FK)	<<== foreign key
Cust Info Transaction Type (FK)	<<== foreign key
Account Number	<<== bank's production "key"
Name	<<== text fact(s)
Address (several fields)	<<== text fact(s)
Phone Number	<<== text fact
Customer Classification	<<== text fact
Credit Rating	<<== non additive numeric fact
Risk Rating	<<== non additive numeric fact
.. other customer descriptors	

This is a typical design for a fact table where the "measurements" recorded by the customer information transactions are changes made to textual values, such as the name, address, and other textual fields listed above. Such a fact table blurs the distinction between a fact table and a dimension table because this fact table is filled with discrete textual values and non additive numeric values that cannot be summarized, but are instead the targets of end user constraints.

Three of the four keys to this fact table are simple foreign keys (FKs) connecting to conventional dimension tables. These include the transaction date, the responsible agent, and the name of the transaction itself. The production account number is not a data warehouse join key, but rather is the bank's constant identifier for this customer account.

The remaining key is the surrogate account key. In other words, it is simply a sequentially assigned number that uniquely identifies this transaction against this account. BUT, here is the subtle point that is the secret of this whole design. This account surrogate key therefore uniquely represents this snapshot of this account at the moment of the customer info transaction, and continues to accurately describe the account until the next customer info transaction occurs at some indeterminate time in the future.

So, to make a long story short, we can use the account surrogate key as if it were a typical Type 2 SCD (slowly changing dimension) key, and we can embed this key in any OTHER fact table describing account behavior. For example, suppose that we also are collecting conventional account transactions like deposits and withdrawals. We'll call these "balance transactions" to distinguish them from the customer information transactions. This second fact table could look like

Balance Transaction Date (FK)	<<== foreign key
Balance Transaction Time of Day (FK)	<<== foreign key
Account (SK)	<<== surrogate key
Location (FK)	<<== foreign key
Balance Transaction Type (FK)	<<== foreign key
Amount	<<== additive fact
Instant Balance	<<== semi-additive fact

When we make one of these balance transaction fact records, we carefully consult our account transaction/fact table and pick out the right surrogate key to use. Normally when we process today's records, we just use the most recent surrogate key for the account. This design then perfectly links every balance transaction to the right account profile described in our first fact table. Or is it a dimension table???

Well, I hope this has got you thinking. I wrote a DBMS magazine article on human resources databases that used a similar design approach. You can find it at <http://dbmsmag.com/9802d05.html>. See also the section "Time Stamping the Changes in a Large Dimension" in the Lifecycle Toolkit book, page 233. Send me your questions and comments and I'll devote the next Design Tip to your responses.

See you in Maui. We'll discuss this design as we watch the sun set over Molokai...