

Kimball Design Tip #43: Dealing With Nulls In The Dimensional Model

By Warren Thornthwaite

Most relational databases support the use of a null value to represent an absence of data. Nulls can confuse both data warehouse developers and users because the database treats nulls differently from blanks or zeros, even though they look like blanks or zeros. This design tip explores the three major areas where we find nulls in our source data and makes recommendations on how to handle each situation.

Nulls as Fact Table Foreign Keys

We encounter this potential situation in the source data for several reasons: either the foreign key value is not known at the time of extract, is (correctly) not applicable to the source measurement, or is incorrectly missing from the source extract. Obviously, referential integrity is violated if we put a null in a fact table column declared as a foreign key to a dimension table, because in a relational database, null is not equal to itself.

In the first case, especially with an accumulating snapshot fact table, we sometimes find columns tracking events which have not yet occurred. For example, in an orders tracking accumulating snapshot, a business might receive an order on the 31st, but not ship until the next month. The fact table's Ship_Date will not be known when the fact row is first inserted. In this case, Ship_Date is a foreign key to the date dimension table, but will not join as users expect if we leave the value as null. That is, any fact reporting from the date table joined on Ship_Date will exclude all orders with a null Ship_Date. Most of our users get nervous when data disappears, so we recommend using a surrogate key, which joins to a special record in the date dimension table with a description like "Data not yet available."

Similarly, there are cases when the foreign key is simply not applicable to the fact measurement, such as when promotion is a fact table foreign key, but not every fact row has a promotion associated with it. Again, we'd include a special record in the dimension table with a value such as "No promotion in effect."

In the case where the foreign key is missing from the source extract when it shouldn't be, you have a few options. You can assign it to another special record in the appropriate dimension with a meaningful description like "Missing key," or assign a specific record such as "Missing key for source code #1234," or write the row out to a suspense file. In all cases, you will need to troubleshoot the offending row.

Nulls as Facts

In this case, the null value has two potential meanings. Either the value did not exist, or our measurement system failed to capture the value. Either way, we generally leave the value as null because most database products will handle nulls properly in aggregate functions including SUM, MAX, MIN, COUNT, and AVG. Substituting a zero instead would improperly skew these aggregated calculations.

Nulls as Dimension Attributes

We generally encounter dimension attribute nulls due to timing or dimension sub-setting. For example, perhaps not all the attributes have been captured yet, so we have some unknown attributes

for a period of time. Likewise, there may be certain attributes that only apply to a subset of the dimension members. In either case, the same recommendation applies. Putting a null in these fields can be confusing to the user, as it will appear as a blank on reports and pull-down menus, and require special query syntax to find. Instead, we recommend substituting an appropriately descriptive string, like "Unknown" or "Not provided."

Note that many data mining tools have different techniques for tracking nulls. You may need to do some additional work beyond the above recommendations if you are creating an observation set for data mining.