

2011-04-24 - The LinkedIn Bitemporal Data Group

On temporal referential integrity.

Continuing our discussion of temporal referential integrity, I'll begin by repeating the list of acronyms I will be using. They are:

EI: entity integrity

TEI: temporal entity integrity

RI: referential integrity

TRI: temporal referential integrity

FK: foreign key

TFK: temporal foreign key

PK: primary key

TPK: temporal primary key

Last time, I set up a simple example of an RI relationship between two non-temporal tables. Now it's time to see what happens to the relationship when the tables are both temporal tables.

Although the theme of this discussion group is bitemporal data, we can actually examine TRI by focusing on only the single dimension of valid time. The reason for this is that existence dependencies, as explained earlier, are dependencies in valid time, i.e. in the real world time in which things exist and change. The same was true for TEI as well, which we also discussed in an earlier comment.

The other temporal dimension – transaction time or Asserted Versioning's superset of transaction time, i.e. assertion time – comes into play as a “container” for data in valid time. When a row of data is created, its transaction time usually begins Now(), i.e. at the moment it is created; and it extends to the end of time, represented by the highest temporal value the DBMS can recognize (for which we use 12/31/9999, and shorten to 9999). The transaction time of a row of data is brought to an end only if we no longer wish to claim that the row is the best and most accurate statement we can make about the object it represents, in the valid time period associated with the row. Most often, that happens when we realize that data was entered incorrectly, and so the statement made by the row is in fact not true.

The semantic relationships of TEI and TRI, among rows of valid-time data, exist only within shared transaction time. For semantic purposes, when we end the transaction time of a row on a specific time X, it has the same effect as if we logically deleted the row.

So we will ignore transaction time in the following examples, and assume that all the rows exist in shared transaction time. This is helpful in understanding TEI and TRI because it makes the sample tables look like tables many of us are already familiar with,

those being valid-time tables which IT professionals usually call version or history tables. These tables add two dates (or timestamps) to the primary keys of tables which would otherwise be non-temporal tables.

(to be continued)