2011-04-18 - The LinkedIn Bitemporal Data Group

On temporal referential integrity.

As I explained earlier, temporal entity integrity is the temporalized form of the relational constraint of entity integrity. As we will see now, temporal referential integrity is the temporalized form of referential integrity.

First, some useful acronyms.

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

Consider the non-temporal form of two tables. The first is the simple Customer table I've used before as an example. Here's a row in that table.

---Customer---
{ [C123] / Smith / 3 }

Braces delimit the row, slashes separate columns, and brackets delimit PK columns. We will also need a notation to indicate the column that makes the child-to-parent reference. In a non-temporal table, the reference is made by means of a FK. As we will see, in a temporal table, the reference is made by means of a TFK. In both cases, we will indicate such columns with a pound sign (#) prefix.

We use dates as our points in time, because they are shorter across the page and thus better for examples.

This row says that C123 uniquely identifier a customer whose name is “Smith” and whose customer status is “3”.

The second is a Customer Contract table, with one entry for every contract a customer has entered into with our company, and the date the contract was signed.

Normally, of course, this would be an associative table; and later on, we will treat it as such, and see how TRI works with associative tables. But for now, we’ll just treat it as a child table in an RI relationship from the Customer table.

(c) 2011, Asserted Versioning, LLC. Non-commercial copies are permitted.

Here’s a row in that table.

---Customer Contract---
{ [#C123] / [CX344] / 4-18-07 }

This row says that C123 signed contract CX344 on April 18th, 2007.

Because there is an RI relationship to the Customer table, defined on the Customer Contract table, the customer identifier PK is also a FK to the Customer table. This means, as we all know, that for as long as this row exists in the Customer Contract table, a row with PK = C123 must exist in the Customer table. In particular, it means that the insert of this Customer Contract row will fail unless a C123 row already exists in the Customer table at the time of the insert. Conversely, it means that a delete of the C123 Customer row cannot leave this Customer Contract row in its table.

What RI and TRI have in common is that both support an “existence dependency” in the real world. In this case, the dependency is that a contract is a legal agreement between our company and a customer, and therefore cannot exist without a customer as a signatory to it.

This is all straightforward. But it sets the stage for a discussion of the temporalized form of RI. Next time, we’ll convert both these tables to temporal tables, and begin to examine the implications of this conversion for the dependency relationship between them.

(c) 2011, Asserted Versioning, LLC. Non-commercial copies are permitted.