2011-03-20 - The LinkedIn Bitemporal Data Group

Topic: entity integrity (EI) and temporal entity intgrity (TEI).

Two of the principal semantic constraints on relational tables are entity integrity and referential integrity. When a conventional (non-temporal) table is put into a valid time envelope (which, in my book and articles I refer to as effective time, since that's the terminology I find more prevalent in the IT community), then both of these constraints must be temporalized. When valid-time tables are then put into a transaction/assertion time envelope, a second level of temporalization is introduced. But the best way to introduce the topic of temporalizing these two constraints is to begin with valid time.

A terminological note, first. I don't mind if the term "valid time" pushes the term "effective time" out of circulation, since the effective time of Asserted Versioning is extensionally equivalent to the valid time of standard bitemporal theory (as per Snodgrass, and also IBM, Oracle and Teradata). But the assertion time of Asserted Versioning is not equivalent to the transaction time of standard bitemporal theory. Instead, it is a proper superset of transaction time. But that is a complex topic, and I won't introduce it until I think transaction time has been adequately treated in this discussion group. So I'll stop referring to transaction/assertion time, for now, and just talk about transaction time.

Let's say that each row in a conventional (i.e. non-temporal) table represents an object. So in a Customer table, each row represents a customer, and each customer is represented by one row. The Closed World Assumption entails that each row represents a customer. Entity integrity entails (and guarantees) that each customer is represented by at most one row.

But when a conventional table becomes a valid-time table, then any number of rows may all represent the same object. One row might represent customer C123 from 5/12/09 to 3/22/10, for example. If C123 changed her name, for example, on 3/22/10, then we add a second row for that customer, with a time period of 3/22/10 to 12/31/9999 (or whatever is the highest date/timestamp the DBMS can recognize). We use this highest date whenever we don't know if or when there will be additional changes to the recorded states of that customer.

Temporal entity integrity (TEI) is entity integrity restricted to a valid-time period. Thus, both rows for C123 satisfy TEI. Now suppose that the valid-time periods for two rows for the same object -- C123 in this example -- overlap. Maybe the two time periods are identical, maybe they share only a single "clock tick" (i.e. a single point in time at whatever level of granularity is being used for the begin and end points of the time periods).

This overlap of valid time periods, for the same object, is what a TEI violation is.

Note that in this example, I am representing a time period by means of a begin point in time and an end point in time. Following a standard convention (called closed-open), the begin point in time is included in the time period, but the end point in time is not. Instead, the end point in time is the next clock tick after the end of the time period.

Thus, if we are maintaining these tables ourselves, we cannot simply add two time periods to the PK of a table. We must also write code to enforce the no-overlap-for-same-object constraint. If we use the temporal features of DB2 10 or Oracle 11g, of course, then the vendors will take care of this no-overlap issue. In other words, the vendors will enforce TEI.

Teradata 13.10, however, has introduced a time period datatype. I don't have the details, but what it means is that a single column can be used to represent time periods.

Also observe that, without a period datatype, PK uniqueness will not enforce TEI on a valid-time table. For example, the following two PKs are unique:

{[C123] / [2/15/10] / [9/22/10]}
{[C123] / [5/22/10] / [12/13/10]}

(rows enclosed in braces, columns separated by slashes, PK columns enclosed in brackets)

However, those two rows violate TEI because they both represent C123 in the time period 5/22/10 - 9/22/10. One row might say that the customer's name is Smith, and the other row that the customer's name is Jones. This are contradictory statements, which is precisely what entity integrity -- in either its non-temporal or temporal form -- is designed to prevent.

The next step is to see what happens to TEI when a valid-time table is put into a transaction-time envelope. I'll describe that later.

But we can now begin to see that adding temporality to tables involves a lot more than adding a time period to the PKs of those tables.

