Three Concepts of Time in Anchor Models

Lars Rönnbäck
August 2009

In an anchor model there are three concepts of time, each of which are stored differently in the database. To avoid confusion with common terminology we will name them *changing* time, *happening* time, and *recording* time. The names try to capture what the times represent: 'when a value is changed', 'when an event happened', and 'when information was recorded'.

Changing Time

The *changing* time for a historized attribute or historized tie is an interval defining the period of time in which its value or relation is valid in the domain of discourse being modeled. In an anchor model this interval is defined using a single date(time) column, `FromDate`. Thanks to the fact that our surrogate keys (anchor identities) never change for an instanced entity we are able to use the `FromDate` as an explicit starting time for the interval *and* as an implicit ending time. The historized attribute containing an individual's hair color could for example contain the row `{'42', 'Brown', '1953-04-25'}`. If there are no other rows with identity `'42'`, then this hair color is assumed to be the current one, i e the interval is open. Now, if the hair color changes, another row with the same identity is added, stating for example `{'42', 'Grey', '2001-02-15'}`. The `FromDate` in this row implicitly closes the interval for the previous row, hence the hair color was `'Brown'` between `'1953-04-25'` and `'2001-02-15'` and is currently `'Grey'` and has been so from `'2001-02-15'`.

Of course this poses a difficulty in the case that you should lose all hair, such that it cannot be said to have a color any longer. Since we do not allow `NULL` values in an anchor model, we cannot enter a row into the table with a `NULL` value for color and a new date. In order to cope with such situations we model a knot holding that state of existence for the hair color. It is basically a flag telling us whether an attribute (or a relation of any kind) is valid or has expired.

If you are interested in knowing the beginnings and ends of the intervals when querying a historized attribute you will therefore have to look at more than a single row. Fortunately, this is not so common when it comes to changing time. The common use case is that you are only interested in what version of a row was valid at a given time. For example the question "I would like to know what hair color the person with identity `'42'` had at the eve of the millennium". Answering this can be done using a rather simple query involving a subselect that ensures that you get the row with identity `'42'` and the latest `FromDate` on or before `'1999-12-31'`.

Happening Time

The *happening* time is used to represent the moment or interval at which an event took place in the domain of discourse being modeled. In anchor modeling this type of time is regarded as being an attribute of the event itself. It should therefore be modeled as one or two attributes depending on the event being momentaneous ("happened at") or having duration ("happened between"). Happening times are values, just like having the hair color `'Brown'` an individual could have the birth date `'1953-02-13'`, and as such it may be historized by a changing time.

As an example, say we have the anchor `PE_Performance` holding the identities for events when some actor performed a stage show. Let us assume that it contains the row `{'4711'}`. On this anchor a static attribute `PEDAT_PerformanceDate` is modeled holding the moment of time when that performance took

place. For example, it could contain the row {'4711', '1969-08-15'}. Note the difference from changing time, which in this case would have been added as a third column. The happening time is on the same level as any other attribute of the event. Compare it with another example, the attribute PENAM_PerformanceName containing the row {'4711', 'Woodstock'}. The rows from the anchor and these two attributes tell us that there is an event with identity '4711', that it is called 'Woodstock', and that it took place '1969-08-15'.

Should we make the attribute PENAM_PerformanceName historized by adding changing time through a FromDate column, we would get a row with three columns; say {'4711', 'Woodstock', '1969-08-15'}. The interpretation of this row is that there is an event with identity '4711' called 'Woodstock', and has been called so since '1969-08-15', but the name may change in the future. If the name should change and a row with a later changing time is added it doesn't mean that the event took place at another time. Therein lies the difference between happening time and changing time.

Recording Time

For maintenance and analysis reasons a third type of time is often necessary, the *recording* time. Loosely speaking it can be said to be the time when the domain of discourse became aware of a certain piece of information. Since this is information about information, we choose to represent it through metadata in anchor models. In many scenarios a single recording time per piece of information is sufficient, corresponding to the time when the data was loaded into the model. However, it can in some cases be necessary to store an array of recording times, if data has passed through a number of systems before reaching the model.

In an anchor model metadata is represented through references to a metadata structure, which also preferably should be anchor modeled. However, let us for the sake of giving an easy to follow example be lenient and allow this structure to have a flat model. As an example, one row in the metadata table is {'7', 'CRM', '2001-02-17', 'LoadPerson'}. When using metadata every row found in an anchor model table is connected to a corresponding metadata row. In other words, an extra column is needed holding that reference. If we take the row from the hair color example earlier it would become {'42', 'Grey', '2001-02-15', '7'}.

These two rows together now provide the following information. There is an individual with identity '42' whose hair turned 'Grey' on '2001-02-15' and this information came from the 'CRM' system and it was entered into our database on '2001-02-17' by the 'LoadPerson' job. If the metadata is extended with a second recording time and some other information to {'7', 'CRM', '2001-02-17', 'LoadPerson', 'Manually Entered', '2001-02-16', 'Jane Doe'}, we gain more information about this particular piece of hair color information. Now we know that there is an individual with identity '42' whose hair turned 'Grey' on '2001-02-15' and this information was 'Manually Entered' by 'Jane Doe' on '2001-02-16' into the 'CRM' system from which the 'LoadPerson' job loaded it into our database on '2001-02-17'.

Finally, adding a static tie INPE_Individual_Performance_Attended with a row {'42', '4711'}, we can further conclude that there was at least one person with brown hair attending Woodstock in 1969.

Related research

Richard Snodgrass has defined two concepts of time used in bitemporal databases [Wikipedia: http://en.wikipedia.org/wiki/Temporal_database]. They are defined as follows.

Valid time denotes the time period during which a fact is true with respect to the real world.
Transaction time is the time period during which a fact is stored in the database.

Both of these have corresponding concepts in an anchor model. The following table shows the mapping between concepts in a bitemporal database and in an anchor model:

Bitemporal Database	Anchor Model
“Valid time”	“Changing time”
“Transaction time”	“Recording time”
(neither of the above)	“Happening time”

Happening times are attributes/properties of things in the domain of discourse that take on values in the form of date(time)s. Some examples of such things are: a bank transaction, an order, and a person, having happening times such as bank transaction execution date, order placement date, order delivery date, person birth date, person deceased date, person marriage date, person divorce date, etc. Being attributes, they may in addition have both changing times and recording times. In a bitemporal database these can be handled similarly to how attributes taking on values from non-time datatypes are handled.

The reason for introducing “happening time” as a concept of its own is to avoid it being confused with the other concepts. Just because something takes on a date(time) value does not necessarily mean that it is either part of valid time or transaction time.