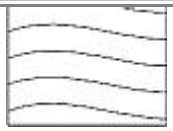
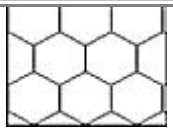
Thoughts about Data (4): Interpreting a Time-Stamped Data Warehouse

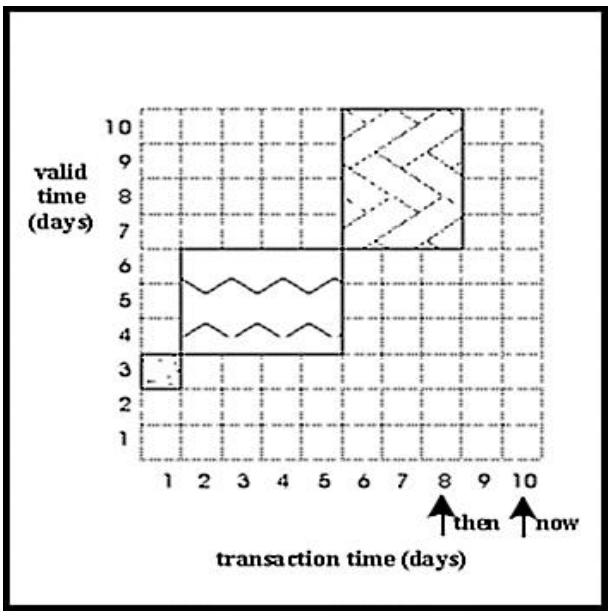
by Mike King

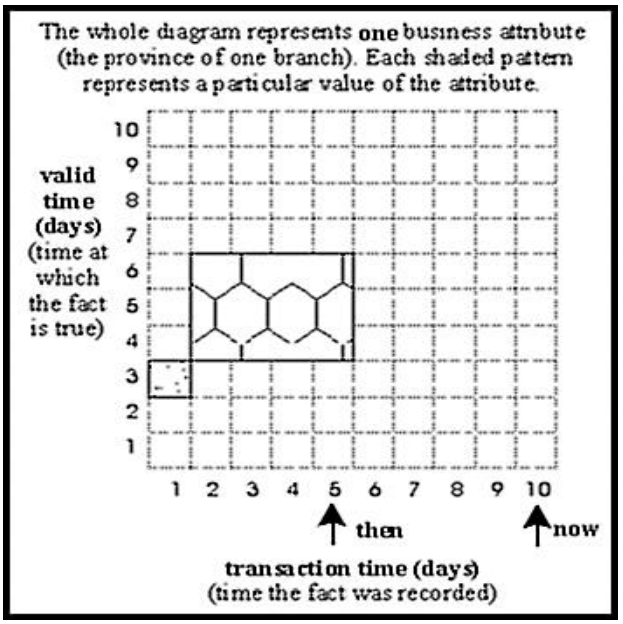
(Read the left column completely before the right column - it seems easier to read in two columns.)

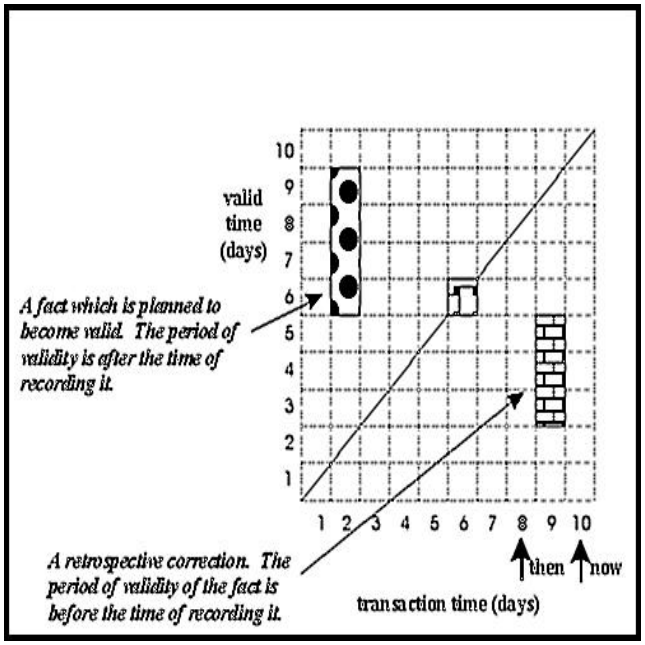
Branch Table for Bi-temporal Database

Branch-Code	Province-Code	Valid Start Time	Valid End Time	Transaction Start Time	Transaction End Time
Rndbrg	Tvl 	3	until changed	1	until changed
Rndbrg	Tvl 	4	6	2	until changed
Rndbrg	Gauteng 	7	forever	6	until changed

---







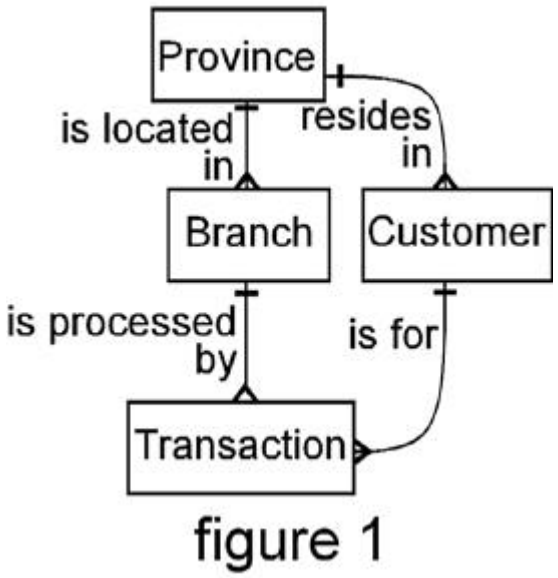


figure 1

Data Warehouses are replenished on a periodic basis (perhaps daily) from the operational databases. The latest information is extracted, cleansed and time stamped before being added to the Data Warehouse.

The interpretation of this time stamped data is not a trivial task, and if not done correctly could result in management decisions based on the wrong business understanding.

The ideas of James Clifford, Tomas Isakowitz and others can help us understand this situation. They define a bi-temporal database as one in which both the transaction times and the valid times are kept for each business fact.

The transaction time is the time at which the fact is recorded in the operational database. The valid time is the time at which the fact is true in the business. It is assumed that both time dimensions are discrete rather than continuous, and that they have the same granularity. In the following examples the granularity will be days (in practice it could be microseconds).

According to figure 1, each branch is located in one province. In a non-temporal database, the branch table would have one row for each branch (thus one province). The corresponding table for a bi-temporal database is shown. It has more than one row for the same province. On day 1, Rndbrg was scheduled to open in Tvl from day 3, indefinitely until some future change. On day 2 it was realized that the opening would be late, and that the province would change to some other name. On day 6, Rndbrg was scheduled to move into Gauteng from day 7, on a permanent basis.

Even though there are three rows for Rndbrg, they must be interpreted to conform to the entity model. That is, at any instant, Rndbrg must fall under only one province. Lets check this. Suppose we examine the database now to observe its state as it was at some previous reference time (then).

In the first example, suppose we examine it on day 10, to see what the situation was on day 5. According to the graph, on day 5 we were aware of the status on day 1 (Tvl), and also the status on days 2, 3, 4 and 5 (also Tvl).

In the second example, we examine the database on day 10 (again) to see what the situation was on day 8. According to the graph, on day 8 we were aware of everything from day 5 and before, and also the status on days 7 and 8 (Gauteng).

For each valid time, if more than one value appears in that horizontal time slice, then the value with the greatest transaction time is the valid value for that time slice.

The above discussion is in the context of having these timestamps in the operational database.

What if the operational database has no timestamps? Then the data may be time stamped as it is added to the Data Warehouse, but what is the interpretation of this?

Suppose the replenishment of the warehouse is done every day, and that there is a mechanism for extracting only new or changed facts. Then it can be seen that the timestamps are effectively transaction times, and the semantics of the database will have to indicate when the facts are intended to be true in the business.

Under these circumstances, there will be no way to construct the transaction times for the very first extraction of data from the operational database, except for transactions which have the time embedded in them as part of the semantics.