Proposal database entry of relationships of subvoyages

[richard.zijdeman@iisg.nl](mailto:richard.zijdeman@iisg.nl)

2020-02-02

Problem:

A voyage describes a move of property from A to B. A subvoyage defines part of a voyage somewhere between A and B. The question is how we can easily describe the temporal relationships between subvoyages in a subvoyage-as-row based dataset.

The image below provides an example of a voyage split into 4 subvoyages: s1, s2, s3, s4. In the example S1 and S2 occur at the same time, while proceeding S3. S4 succeeds all of the other subvoyages.

S4 https://coins.iisg.amsterdam/coins

S3 https://coins.iisg.amsterdam/coins

S1

S2 https://coins.iisg.amsterdam/coins

Solution 1: value-list

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Subvoyage | … | precedes | co-occurs-with | succeeds |
| S1 |  | S3,S4 | S2 |  |
| S2 |  |  |  |  |
| S3 |  |  |  | S1,S2 |
| S4 |  |  |  | S1,S2,S3 |

The first solution uses three relationship-columns with value-lists inside the columns. The great thing about it: it adds just three variables to the already existing database. The not so nice thing about it, is the use of multiple values in one ‘cell’. (Although this may become a more accepted thing in the future).

Solution 2: multiple columns

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Subvoyage | … | precedes1 | Precedes2 | co-occurs-with |
| S1 |  | S3 | S4 |  |
| S2 |  |  |  |  |
| S3 |  |  |  |  |
| S4 |  |  |  |  |

Solution 2 deals with the ‘problem’ of value list by generating multiple columns. In this example we would already generate 6 columns to describe all occurrences, which immediately underlines the downside of this solution, especially, because there usually are only a few extreme instances with multiple relationships leading to a very large, but mostly empty database.

Solution 3: Matrix-notation

The figure on the subvoyages already has a matrix-like-look. We could use that to annotate the relationships between the voyages.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Voyage | Subvoyage | Row | Column |  |
| V1 | S1 | 1 | 1 |  |
| V1 | S2 | 2 | 1 |  |
| V1 | S3 | 1 | 2 |  |
| V1 | S4 | 1 | 3 |  |

The good thing about this solution is that only two columns are added, and that the notation is fixed. The bad thing is that it requires another action (even though ever so simple) to ‘calculate’ the pre-/succeed/co-occurs relationship from the Column column.

Solution 4: the ‘Ulbe-subject-predicate-object-paradigm’

In general, if we know all of the relationships of the subvoyages, we only need to mention a predicate and object as shown below.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Subvoyage | … | relationship | subvoyage | co-occurs-with |
| S1 |  | co-occurs | S2 |  |
| S2 |  | precedes | S3 |  |
| S3 |  | precedes | S4 |  |
| S4 |  |  |  |  |

The good thing about this solution: it requires just two columns and no value-list. Downsides of this solution are (1) the need of complete information on all of the subvoyages and (2) it takes another action to reproduce the network of subvoyages and finally, there are multiple ways to write the same information, which may lead to redundancy and inconsistencies in notation. To illustrate rather than having at row S3 ‘precedes S4’, we could have written at row S4 ‘succeeds S3’.

Solution 5:

Having talked to Samantha after the meeting, I would again like to bring up the possibility of simply using the year of the subvoyage to calculate the relationship between subvoyages. Even though it is not precise, it is something that is easily recalculated or re-estimated after the database already has been completed. Moreover, it doesn’t require aggregating the relations into three categories: it could ofcourse occur, that a subvoyage starts earlier than another subvoyage (preceeds), but only ends after that particular subvoyage (which would make it also be co-occurring and perhaps succeeding).

While I am not at par with the domain/database specific arguments, from a more generic database principle, I would personally be in favour of solution 5 (or otherwhise solution 3).