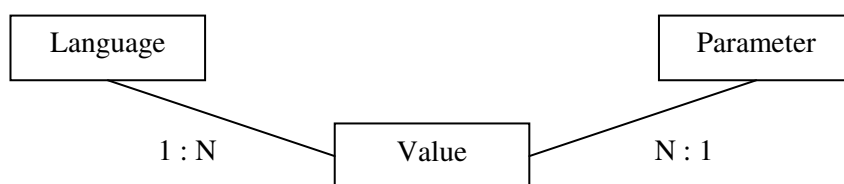


Alexis Dimitriadis, Utrecht
An extensible design for linguistic survey databases

Databases for linguistic research have a number of properties that distinguish them from the business-oriented databases that inform much of the theory and practice in the field of databases. Most crucially, the properties, categories and values that should be recorded in a research-oriented database are typically not known at the start of data collection. This frequently leads to problems, since revising a database that is already in use is a much more complicated proposition than including the same features in the original design. This talk will present a database design that, among other features tailored to linguistic databases, supports the easy modification and addition of linguistic attributes and category values. We mention here some sources of difficulties, and the approach adopted:

A research-oriented database records a variety of linguistic properties for a large number of languages (or other units of description). The number of properties can easily reach several hundred, especially if the database is added to over a long period of time, creating and managing forms and queries for so many parameters can be complicated and error-prone. But linguistic databases have another special characteristic: the exact meaning of such classificatory parameters is often important, but not obvious or universally agreed to (e.g., “weak agreement”, “tensed language”). In such cases, it is important to maintain documentation for these parameters, and to make it available to users. But general-purpose DBMSs make little provision for documentation of the attribute declarations. Our solution is to store the names, types and documentation of linguistic parameters in their own table; the value they take for each language (or construction, morpheme, etc.) is given in a separate table, which relates languages with parameters. In different terms, Parameters (“questions”) are entities in a many-to-many relationship with Languages; at the intersection of the two is the Value of a parameter for some language (the “answer”). The approach allows new parameters to be added at any time without modifying the relational design of the database; allows the new parameters to be displayed in existing forms; and provides a natural place to store the documentation of each parameter, at the Parameter table itself.



It is not uncommon in linguistic description to have a parameter that usually takes just one value per language (e.g., “noun-adjective order”), but for which a second value must occasionally be recorded. Because single-valued attributes in relational databases are much simpler than multi-valued ones, such cases present a tricky choice between simplicity and flexibility. Moreover, changing a single-valued attribute to multi-valued involves a non-trivial design change; this encourages researchers to fall back on non-optimal solutions, such as entering both values in a single field of the database or mentioning the second value in an overused “comments” field. The design described above eliminates this problem, since it supports multi-valued attributes just as straightforwardly as single-valued ones (multi-valued attributes are in fact the

default). A single-valued parameter can be changed to multi-valued by simply changing an option flag, and existing data is not affected.

The database design includes facilities for the managing lists of “enumerated” possible values (they are all stored in a single table), and a core triad of entities *Language*, *Construction* and *(Example) Text* that reflects the structure of many cross-linguistic survey databases. A database based on this design is in use for a cross-linguistic survey of reciprocals, and has been adapted for other projects.