THE OPEN-WORLD RELATIONAL DATABASE AS A BASIS

FOR KNOWLEDGE ENGINEERING IN GEOSCIENCE

Stephen Henley,

Resources Computing International Ltd, Matlock, United Kingdom

Email steve@vmine.net

The validity of any observational science database is dependent upon the data model used in its management. Whilst the relational data model has become generally accepted as the ideal, its implementations, and more recently the development of database theory, have been dominated by the requirements of business and administrative users. Recent publications on relational database theory insist that one of the fundamental axioms of the relational database model is the closed world assumption. This requires that there must be no uncertainty in the database - and that anything missing from the database is deemed to be false. In the observational sciences it is in general impossible to structure databases in this very strict way, with all 'nulls' prohibited.

Prohibition of NULL within the relational database model, as currently defined, is a direct consequence of the use of a strict two-valued logic. However, where there are genuinely unknown items of data this can actually lead to ambiguity and inconsistency, and although there are workarounds such as the ad hoc inclusion of a NULL in commonly used implementations of SQL, it is argued that the best solution is to replace the closed world assumption by the open world assumption. This was explicitly acknowledged by the originator of the relational database model, E.F.Codd (1979, 1990).

The proponents of a strict 'closed world' approach to database management have proposed a number of methods by which 'unknown' truth values could be incorporated into their databases. Unfortunately all of these are either logically flawed or violate one or other of the axioms of the relational model. The SQL NULL is also known to be logically flawed, leading to incorrect deductions.

In real life geoscience situations - for example a geochemical survey where a large number of samples have been analysed for, say, 50 chemical elements - it is very common for analyses to be either temporarily or permanently missing for many elements in different samples (for example, some samples not analysed for all elements, or results awaited for other samples). One of the leading suggestions for allowing uncertainty in a strict closed-world database was made by Darwen and involves complicated decomposition of tables into potentially a large number of binary relations (two-column tables) as well as horizontal decomposition of tables.

Were Darwen's decomposition solution to be adopted, in the case suggested above, this could require up to 50 binary relations, each containing a list of samples for which the analysis of chemical element X is missing, and horizontal decomposition of the original relation into perhaps a very large number of relations, one for each different combination of attributes (chemical elements) in which there are 'missing data'. Apart from the complexity of the data management (which might conceivably be automated) there remains the problem of defining the results of logical operators on the 'missing data'.

In particular, it is unclear how such a solution would help in selection of all samples "WHERE Mg > Ca" if there are missing data for either (or both) of Mg and Ca in some samples, and it would be even more difficult to handle comparisons where data are constrained (such as 'below detection limit') rather than altogether absent.

Whatever the data organisation, and however much the original relation might have been decomposed to hide the fact that data are missing for a particular sample, the correct result, for that sample, from such an operator is the truth value "unknown" - prohibited in the two-valued logic of the relational model as defined under the closed world assumption.

The 'missing data' problem could be viewed in another way. Interpretation of the contents of a relation depends entirely upon the predicate under which the relation has been defined. It is generally assumed that predicates are of the form:

"Employee *emp#* exists and is named *name*" (1)

or in other words, are statements about the real world. However, without any change in the structure or content of the relation (i.e. of the contents of the database), an alternative predicate might be used:

"We know that employee *emp#* exists and is named *name"* (2)

This is a statement about our knowledge or belief about the real world, and is much closer to the real nature of a database. Given the closed world assumption, under predicate (1), the absence of an otherwise legitimate tuple implies the falsehood of the predicate - and an implied negative statement about the existence of any employee with any name which is not included in the relation. Under predicate (2), the absence of a tuple again implies the falseness of the predicate, but in this case the meaning is "we do not know ..." rather than non-existence of the employee. In either case, as we have only two-valued logic, only two logical states are allowed. In the first these states (as referring to the existence of a named employee) are true and false (while unknown cannot be represented). In the second case they are true and unknown (while false cannot be represented).

We could modify the predicate a little further:

"We know that it is [*true,false*] that
employee *emp#* exists and is named *name"* (3)

This appears at first sight to be little different from predicate (2), but it should be noted that there is an additional attribute, of truth-value type, to express our definite knowledge that a named employee does or does not exist. As with predicate version (2), the absence of a tuple indicates "we do not know ...". This allows us to express all three truth values true, false, and unknown, in a single relation within the closed world assumption. However, it does not avoid the problem identified above, that representation of missing data for particular attributes in a closed-world database can require very complicated decomposition of tables.

The only practical solution to this is to allow an open world database in which there is explicit representation of "unknown" for each data item in a table. This then allows us to move on to representation of "partially missing" data which is widespread in geoscience - expressing concepts such as "greater than ..." or "less than ..." or "between ... and ...". This is where the database starts to become useful as a means of organising data that can then be handled by knowledge engineering applications.

References

Codd, E.F., 1979 Extending the database relational model to capture more meaning. ACM Transactions on Database Systems 4(4), 397-434.

Codd, E.F., 1990 The relational model for database management: version 2, Addison-Wesley, Reading, Mass., 538pp.

Date, C.J. and Darwen, H., 1998 Foundation for object/relational databases: the third manifesto, Addison-Wesley, Reading, Mass., 496pp.

Henley, S., 2005 The man who wasn't there: the problem of partially missing data. Computers & Geosciences, 31(6), (2005) 780-785.

Henley, S., 2006 The problem of missing data in geoscience databases. Computers & Geosciences, 32(9), 1368-1377.