In Defense of Foreign Keys

In his paper "Foreign Keys Decrease Adaptability of Database Designs", author Richard Wilmot [3] concludes that database designers should follow "a design practice ... that avoids foreign keys and uses instead separate records for representing all relationships among entities". This conclusion cannot go unchallenged, for it can easily be shown to be a generalization based on (a) two few instances of the common one-to-many (1:n) relationship that employs a foreign key, and (b) too few instances of the types of relationship that can occur in data bases that use files or relations that conform to the requirements of third normal form.

I agree with Wilmot on the following: What on the surface looks like a simple 1:n relationship, as in the case of the example in the paper of the relationship between department entities and employee entities, often is more complex when looked at more closely. Thus an employee may be working in two or more departments, making the relationship many-to-many (n:m), and most employees will have had a history of employment at different departments, introducing a further relationship. There are certainly many common cases of this type of thing, for example ship entities and crew-member entities, forest and tree entities, company and subsidiary entities, and so on. In such cases there is no doubt that use of just two files or entity collections and a foreign key that is not part of a primary key can lead to inflexibility if n:m aspects of the relationship ever become important. The basic problem in all of these cases of 1:n relationships is that the relationship is only apparently a 1:n relationship as far as physical reality is concerned, and is really at least a single n:m relationship.

But a fundamental point that Mr. Wilmot fails to consider is that there are plenty of pairs of entity types where the relationship is fundamentally 1:n, pure and simple, and there are no n:m considerations of any kind (objection (a) above). A graphic example is moon and crater entities. A moon can have zero or more craters, and a crater must be on one moon from the time of creation to the time of destruction. None of Mr. Wilmot's arguments apply to this simple example. The relationship is 1:n and nothing else. There are many examples of this type of fundamental 1:n relationship.

There are also examples where there is a theoretical possibility of the 1:n relationship having n:m aspects, but where the probability of these becoming relevant in practice is remote. For example, we could have state and city entities. A state can have many cities and a city is in just one state, so that the relationship appears to be 1:n. However, it could be argued that boundaries will change, so that one day Las Vegas could be in California, in which case a city would have a history of the different states it was once part of. But since such a possibility is clearly remote, no sensible designer would follow the design practice advocated by Mr. Wilmot and install a third relation or conceptual file for the eventual n:m relationship between state and city.

There is a further, though more indirect, objection to what the paper advocates (objection (b)). Mr. Wilmot argues that all relationships should be represented by separate records. He bases this argument on the common error of assuming that the most general type of relationship is the n:n relationship between entities A and B where there is a third entity R such that there is a 1:n relationship between A and R and also between B and R. In effect R represents the n:m relationship. The best-known example of this is Date's Supplier and Parts files where there is an n:m relationship between Supplier and Parts that is explicitly represented by a third file SP [2]. The important point about this type of n:m relationship is that the third file is necessary, for without it, the relationship between Supplier and Parts could not be determined at the conceptual level. Mr. Wilmot thus argues that to allow for an apparent 1:n relationship being an n:m relationship in disguise, there should be a third file like Date's SP file to contain the relationship. As a result all relationships, that is, the 1:n relationships and the n:m relationships that require a third file, would be explicity represented by records in a third file. Unfortunately, the assertion fails because of the existence of n:m relationships where a third file is not required, and would even be extremely inconvenient to include.

The type of n:m relationship to which I am referring, can be seen from a slightly modified version of the Supplier-Parts data base. Suppose that we have the relations:

SUPPLIER (SNO, STATUS, CITY)

PART (PNO, COLOR, PCITY)

where CITY gives the location of the supplier and PCITY is the city where a type of part is manufactured. Primary key fields are underlined.

If PCITY and CITY are drawn on the same domain, then SUPPLIER and PART are related. The relationship is n:m since for a supplier in Denver there can be many part types manufactured in Denver, and for a part type made in Denver there can be many Denver suppliers. Of course the relationship can be purely coincidental (that is, the relationship may have only a coincidental mode [1]), but it can have greater meaning. For example, it might be that a supplier can offer same day delivery of parts made in the city in which that supplier is located. That no third file is needed for this n:m relationship should be obvious from the SQL expression for the query: Find the supplier numbers of suppliers who offer same-day delivery of red parts:

```
SELECT SNO FROM SUPPLIER
WHERE CITY IN (SELECT PCITY FROM PART
                    WHERE COLOR = 'RED')
```

It follows that the generality of Mr. Wilmot's conclusions fail, and that the point of the paper is merely that designers should take care with prospective 1:n relationships, lest they later turn out to be common n:m relationships requiring a third relational file, something that many designers will already have learned from hard experience.

J. Bradley

Computer Science Dept.

University of Calgary, Alberta,

Canada.

REFERENCES

1. Bradley, J. "SQL/N and modes of association in relational data bases", Univ. of Calgary Research Report No. 84/145/3, 40 pages, to be published. Contains a thorough classification of the non cyclic relationships found in data bases with relational structure, together with formal definitions of modes of association for these relationships.

2. Date, C. J., Introduction to Database Systems, 3rd Ed., Addison-Wesley, Reading, Mass., 1981, Chapter 3.

3. Wilmot, R. B., "Foreign keys decrease adaptability of database designs", Comm. of ACM, Vol. 27, No. 12, Dec. 1984, page 1237.