Comment on "Segment Synthesis in Logical Data Base Design"

The examples shown in Figs. 2 and 3 of [1] illustrate the existence of simple cycles in the data model between the department and the manager of each case. Such relations, which are cyclic on a one-to-one basis, are frequent, both in the conceptual construction of data models by users and in the implementation of a data base for multi-file systems. To indicate this simple equivalency, the relation (D, M) can be replaced by an equivalent single domain, say X. Because of the frequency of occurrence of binary one-to-one relations, I have termed such relations a *lexicon* [2]. On applying this simplification, the result of the analysis for minimum cover given in [1] is a single set of relations as follows:

$R1(E, X, JC)$
$R2(X, CT)$
$X(D, M)$.

Whether a lexicon is to be stored with D or M as a primary key, or possibly redundantly, is a performance-oriented consideration which has already been discussed in [3].

This approach does not eliminate the need for analysis in the case of more complex cycles, but in systems that have many functional relationships, it considerably simplifies both the analysis and the results and at the same time expresses these ubiquitous relationships in a semantically meaningful way.

Gio Wiederhold
155 Marine Road
Woodside, CA 94062

September 25, 1975

References

1. C. P. Wang and H. H. Wedekind, "Segment Synthesis in Logical Data Base Design," *IBM J. Res. Develop.* **19**, 71 (1975).
2. G. Wiederhold, *Data Base Design*, McGraw-Hill Book Co., Inc., New York (in preparation).
3. E. F. Codd, "A Relational Model of Large Shared Databanks," *Comm. ACM* **13**, 377 (1970).

The authors reply:

The introduction of binary lexicons to eliminate cycles in a graph and to simplify the analysis and formulation of data base schema is an interesting idea. However, such a simplification also brings about some fundamental problems.

Let us examine the simplified minimum cover set of relations as given above. (We underline the keys of those relations, $\underline{E}$, $\underline{X}$, and $\underline{D}$, to improve clarity.) In relation X, D is arbitrarily assigned to be the key. Strictly speaking, $R1$ and $R2$ are unnormalized relations—each contains a relational domain X. Of course, we recognize that X is simply a domain pair that is always single-valued (a binary tuple) for each instance of $R1$ and $R2$. For the moment, let us ignore this anomaly in formalism and examine the instances of the above set of relations in detail.

Before writing down explicitly all the instances in tabular form, we have to ascertain the value sets for the domain X in $R1$ and $R2$. There seem to be two plausible choices: (1) Choose the set of tuple identifiers of the relation X to be the value set for the domain X, or (2) treat X as a compound domain and take the set of binary tuples of the relation X to be the value set. For the first choice, it is necessary to rename the domain X as X_{id} to reflect the intended domain definition. The resulting schema becomes

$R1(\underline{E}, X_{id}, JC)$
$R2(\underline{X}_{id}, CT)$
$X(\underline{X}_{id}, D, M)$.

It is to be noted that the extra domain X_{id} has to be incorporated in the relation X to provide vital connections to $R1$ and $R2$ at the instance level.

For the second choice, it is equivalent to rewrite the schema as

$R1(\underline{E}, D, M, JC)$
$R2(\underline{D}, \underline{M}, CT)$
$X(\underline{D}, M)$

to reflect accurately the definition of domain X. Here $R1$ is no longer a relation in third normal form. The relation $R2$ has a key which is not minimal.

C. P. Wang
IBM System Communications Division
Yorktown Heights, NY 10598

H. H. Wedekind
Technical University
Darmstadt, West Germany

January 7, 1976