On the Design of Relational Database Schemata

CARLO ZANIOLO
Sperry Research Center
and
MICHEL A. MELKANOFF
University of California at Los Angeles

The purpose of this paper is to present a new approach to the conceptual design of relational databases based on the complete relatability conditions (CRCs).

It is shown that current database design methodology based upon the elimination of anomalies is not adequate. In contradistinction, the CRCs are shown to provide a powerful criticism for decomposition. A decomposition algorithm is presented which (1) permits decomposition of complex relations into simple, well-defined primitives, (2) preserves all the original information, and (3) minimizes redundancy.

The paper gives a complete derivation of the CRCs, beginning with a unified treatment of functional and multivalued dependencies, and introduces the concept of elementary functional dependencies and multiple elementary multivalued dependencies. Admissibility of covers and validation of results are also discussed, and it is shown how these concepts may be used to improve the design of 3NF schemata. Finally, a convenient graphical representation is proposed, and several examples are described in detail to illustrate the method.

Key Words and Phrases: relational databases, schema design, functional dependencies, multivalued dependencies, minimal covers, decomposition
CR Categories: 4.33, 5.9

1. INTRODUCTION

The purpose of this paper is to present a new approach to the conceptual design of relational databases which ensures the complete relatability of data while eliminating redundancy.

Modern integrated database systems enable a community of users to share data through a common database model called the schema. The need for data independence and user convenience suggests that the schema, rather than describe the physical organization of the data in the storage media, should display the logical association or relationships of interest to the users of the database.

Part of this research was completed while C. Zaniolo was with the Sperry Research Center.

Authors' present addresses: C. Zaniolo, Bell Laboratories, Holmdel, NJ 07733; M. A. Melkanoff, Computer Science Department, University of California, Los Angeles, CA 90024.

The capability of the schema to define exactly and unambiguously all the relationships of interest to the users has been named complete relatability following GUIDE and SHARE [18]. The relationships found in databases may be classified as (1) one-to-one, (2) many-to-one, and (3) many-to-many.

Various data models, such as the hierarchical, the network, and the relational model, have been proposed for a database schema. The degree of data relatability achievable with the proposed models, their potential for data independence, and their suitability for efficient implementation have received wide attention and are not discussed here. This paper concentrates on the general problem of designing database schemata which provide complete relatability to the users. We analyze the problem using the framework and the mathematical formalism of the relational model. A companion paper will present further applications of this technique [28].

1.1 Design of a Database Schema

There are, basically, two approaches to the design of a relational database schema: an *analytical* and a *synthetic* approach [16].

The analytical approach has been proposed by Codd [10]. Here the designer produces a first tentative model of the database in the form of a set of relations of assorted degree. Then he describes the meaning or intension of his relations and constructs, if one is not available, a sample of his database content. As described in [9] and [10], those relations which are characterized by a hierarchical organization or by certain dependency structures have undesirable characteristics and should therefore be removed. The designer must detect such relational forms and decompose them into lower order relations by a procedure called normalization. This approach has been applied and developed by several authors [13, 16, 22].

A synthetic approach which yields a formal design algorithm is presented in [26] and [6]. In this approach, the designer is expected to define initially the database attributes and the functional dependencies among these attributes. Algorithms are then given to design the proper relational schema by determining the minimum cover of these functional dependencies. The synthetic approach and the various problems connected with it are discussed in more detail in Section 2.4.

2. LOGICAL DEPENDENCIES IN RELATIONS

2.1 Definitions and Notation

Considering the database as a set of time-varying relations, we denote by

$$R(A_1, A_2, \ldots, A_n) \tag{2.1}$$

a *relation* of degree $n \geq 1$ where $A_1, A_2, \ldots, A_n$ are called the attributes of R. Each attribute A_i uniquely names a *domain* of R denoted $\mathrm{DOM}(A_i)$ which is the set of all possible values for that attribute. The relation R is defined as the subset of the Cartesian product of its domains,

$$R(A_1, A_2, \ldots, A_n) \subseteq \mathrm{DOM}(A_1) \times \mathrm{DOM}(A_2) \times \cdots \times \mathrm{DOM}(A_n).$$

While the domains are not necessarily distinct, they form uniquely named attributes in R, and the order of the attributes in a relation becomes immaterial. $\Omega = \{A_1, A_2, \ldots, A_n\}$ denotes the attribute set of that relation and $|\Omega| = n$ denotes its cardinality. Relation (2.1) is also written $R(\Omega)$. Any subset $\Delta \subseteq \Omega$, is called a *subset* or a *combination* of attributes of R. If $A \in \Omega$, then $r[A]$ denotes the A-component of $r \in R(\Omega)$. Similarly, if $\Delta \subseteq \Omega$, then $r[\Delta]$ denotes the subtuple of r of size $|\Delta|$ containing the components of r corresponding to the elements of Δ. We also refer to $r[A]$ and $r[\Delta]$ as the A-value and the Δ-value of r. Unless otherwise stated, we assume that all relations are in first normal form [9]; that is, all attribute values are assumed to be simple rather than composite such as relations. The set of Δ values for some tuples of R is called the *projection* of R, denoted by

$$\Pi_\Delta \cdot R = \{r_1 \mid r \in R \text{ and } r_1 = r[\Delta]\}.$$

Since $\Pi_\Delta \cdot R(\Omega)$ is itself a relation with attribute set Δ, we use the abbreviated notation, $\Pi R(\Delta) = \Pi_\Delta \cdot R(\Omega)$.

The symbols $\cup$, $\cap$, and $-$ denote the usual set union, intersection, and difference operators, while $\varnothing$ denotes the empty set. Two sets having empty intersection are called disjoint. The *natural join* of two relations, $R(\Omega)$ and $S(\Delta)$, denoted $R(\Omega) \cdot S(\Delta)$, is a relation with attribute set, $\Omega \cup \Delta$, defined as

$$R(\Omega) \cdot S(\Delta) = \{r \mid r[\Omega] \in R(\Omega) \text{ and } r[\Delta] \in S(\Delta)\}.$$

When $\Omega \cap \Delta = \varnothing$, the natural join reduces to the Cartesian product. Natural joins are commutative and associative.

2.2 Functional Dependencies

We shall briefly review the concept of functional dependency which has proved very useful in analyzing the logical structure of relations.

Functional Dependency (FD). The attribute combination, Δ, is said to be functionally dependent (FD) in R on the attribute combination Γ, when for any pair of tuples $r_1, r_2 \in R$,

$$(r_1[\Gamma] = r_2[\Gamma]) \Rightarrow (r_1[\Delta] = r_2[\Delta]);$$

that is, the equality of the Γ values implies the equality of the Δ values.

Whenever Δ is FD on Γ we also say the Γ uniquely determines, or functionally determines Δ, which we denote as "$\Gamma \rightarrow \Delta$." When Δ is not FD on Γ, we may write "$\Gamma \not\rightarrow \Delta$." We are only interested in those FDs which are always valid for a given database; that is, they are expressions of the intension of the relation, and are not due to coincidence. The definition of FD is always verified when $\Gamma \supseteq \Delta$. Any FD, $\Gamma \rightarrow \Delta$, is then called a *trivial* FD.

The first important application of FDs arises in defining the concept of candidate keys of a relation. A *candidate key* is a minimal combination of the attributes of a relation which uniquely determines all the remaining attributes.

Candidate Key. A combination, $\Delta \subseteq \Omega$, is a candidate key for $R(\Omega)$ iff

(1) $\Delta \rightarrow \Omega$, and
(2) for every combination Δ' properly contained in Δ, $\Delta' \not\rightarrow \Omega$.

Since there are no duplicate rows in a relation, no two rows may have the same values of a candidate key. There always exists at least one candidate key in any relation. If the relation, $R(\Omega)$, contains only trivial FDs, then Ω is the only candidate key. Otherwise, one or more proper subsets of Ω are the candidate keys. When depicting a relation, one usually selects a particular candidate key (when there are more than one) which is designated by underlining the corresponding attributes; it is then called the primary key.

A second aspect of FDs is represented by their formal properties [13]. These are important in the design of relational schemes and we use them to such effect later.

Formal Properties of FDs. Let Γ, Δ, Λ, and Ψ be nonempty subsets of Ω. Then the FDs of a relation $R(\Omega)$ have the following properties:

F1. *Reflexivity*: If $\Gamma \supseteq \Delta$, then $\Gamma \to \Delta$.
F2. *Augmentation*: If $\Gamma \to \Delta$ and $\Lambda \supseteq \Psi$, then $(\Gamma \cup \Lambda) \to (\Delta \cup \Psi)$.
F3. *Transitivity*: If $\Gamma \to \Delta$ and $\Delta \to \Psi$, then $\Gamma \to \Psi$.
F4. *Pseudotransitivity*: If $\Gamma \to \Delta$ and $(\Delta \cup \Lambda) \to \Psi$, then $(\Gamma \cup \Lambda) \to \Psi$.
F5. *Additivity*: If $\Gamma \to \Delta$ and $\Gamma \to \Psi$, then $\Gamma \to (\Delta \cup \Psi)$.
F6. *Distributivity* (with respect to union): If $\Gamma \to (\Delta \cup \Lambda)$, then $\Gamma \to \Delta$ and $\Gamma \to \Lambda$.

These properties have been treated extensively in the previous literature. The following two properties, not explicitly stated in previous works, are equally important:

F7. *Projectability* (implication from a relation to its projections): If $\Gamma \to \Delta$ in $R(\Omega)$ and $\Gamma \subset \Psi \subseteq \Omega$, then $\Gamma \to (\Delta \cap \Psi)$ in $\Pi R(\Psi)$.
F8. *Reverse Projectability* (implication from a projection to the relation): If $\Gamma \to \Delta$ in a projection of R, then $\Gamma \to \Delta$ in R.

The validity of F7 and F8 follows immediately from the definition of FD. $\Gamma \to \Delta$ is called a *full FD* when, for every proper subset $\Gamma' \subset \Gamma$, we have $\Gamma' \not\to \Delta$; otherwise, it is called a *partial FD*.

2.3 Anomalies and Normal Forms

Relations where some attribute is partially or transitively dependent on a candidate key present various anomalies [10]. These anomalies occur when tuples are added, deleted, or updated.

2.3.1 *Anomalies Resulting from Partial FDs.* Let us consider a relation, STOCK, which describes the current content of a hypothetical stockroom. A snapshot of its content at a certain instant of time might be

STOCK	(<u>SUPPLIER,</u>	<u>ITEM,</u>	COLOR)	
	WOODMAN	CHAIR	BROWN	
	WOODMAN	TABLE	BLACK	(2.2)
	WOODMAN	SOFA	BROWN	
	HOUSEMAN	DESK	GREEN	
	HOUSEMAN	SOFA	BROWN	

The domains, SUPPLIER, ITEM, and COLOR denote, respectively, the code names of parts and a standard set of colors. The instantaneous content of our relation is completely defined by the set of rows of a table. The interpretation, however, is only partially specified by its headings. A qualifying statement, such as the following, is required: "Our relation describes current suppliers and the colors of each part item currently in stock. While one item may have many suppliers it has only one color." The first part of that statement specifies that we are interested in the logical relationships between, on the one hand, the items and their suppliers, and, on the other hand, between the items and their color. The second part of the statement specifies that the first relationship is many-to-many, while the second is many-to-one. In relational database jargon the current content of relations is called *extension*, while their meaning is called *intension* [11]. While the extension of a database is subject to continuous changes, its intension undergoes only a slow evolution (if any). The database intension is therefore assumed to be time independent unless otherwise specified.

Relation (2.2) suffers from the following anomalies. Under the important assumption[1] that a key attribute cannot include an undefined value [10], no information on the color of a new part can be introduced unless at least one supplier of that part is active. Furthermore, deleting the last supplier of a part will obliterate the information regarding that part's color. Finally, a change in the color of a part will require a complete search through the file and a possible update on multiple records, if one wants to avoid database inconsistencies. Now it may be seen that relation (2.2) has only one candidate key, the combination {SUPPLIER, ITEM}. Furthermore, the attribute COLOR is partially dependent on that combination since ITEM alone uniquely determines the value of COLOR. Codd suggests that, under these conditions, the anomalies may be removed by decomposing the relation in such a way that in the new relations no attribute will be partially dependent upon a candidate key. Relations which satisfy this condition are said to be in second normal form (2NF) [10]. Thus relation (2.2) may be decomposed into the pair:

ΠSTOCK	(SUPPLIER,	ITEM)
	WOODMAN	CHAIR
	WOODMAN	TABLE
	WOODMAN	SOFA
	HOUSEMAN	DESK
	HOUSEMAN	SOFA

(2.3)

ΠSTOCK	(ITEM,	COLOR)
	CHAIR	BROWN
	TABLE	BLACK
	SOFA	BROWN
	DESK	GREEN

(2.4)

By using the pair of relations (2.3) and (2.4), the anomalies of schema (2.2) are avoided. The suppliers and colors of a part can now be changed independently of each other.

[1] This assumption and its consequences are discussed further in Section 2.4.

2.3.2 *Anomalies Resulting from Transitive FDs.* The other type of dependency structure identified by Codd as a source of anomalies is the transitive dependency on candidate keys, which can still be present in second normal form relations. Assume, for instance, that a financial company needs to keep records of its employees (EM) managing the accounts (AC#) of various customers, along with the telephone extensions of the employees (TX). That information can be modeled by the following relation:

ACMG (AC#,	EM,	TX)
4218	A. MILLER	428
7531	A. MILLER	428
1537	B. BROWN	753
8532	T. LEIGH	915

(2.5)

Assuming that an account has only one manager and an employee has only one telephone extension, the dependency structure of (2.5) may be depicted by the following:

$$\begin{tikzcd} AC\# \arrow[r] \arrow[dr] & EM \arrow[d] \\ & TX \end{tikzcd}$$

Note that "AC# → TX" is a transitive FD. Under the important assumption that the key, AC#, cannot have an undefined value, a telephone extension can be recorded only for those employees who currently manage at least one account. Once the telephone extension of an employee is changed, a number of records equal to the number of accounts managed by that employee must be changed.

It was suggested that the anomalies which characterize relation (2.5) are due to the presence of a transitive FD; here again a decomposition of relation (2.5) was proposed to remove anomalies. Thus decomposing (2.5) according to EM → TX, we obtain the relations (2.6) and (2.7):

ΠACMG (AC#,	EM)
4218	A. MILLER
7531	A. MILLER
1531	B. BROWN
8532	T. LEIGH

(2.6)

ΠACMG (EM,	TX)
A. MILLER	428
B. BROWN	753
T. LEIGH	915

(2.7)

It may be seen that the anomalies have indeed been eliminated by the decomposition.

Relations in 2NF which are free from transitive dependencies are said by Codd to be in third normal form (3NF). The original definition of 3NF [10] distinguishes between those attributes which participate in at least one candidate key and the remaining attributes. That definition has been amended in [11] in order to handle certain anomalies not removed through its application. The new definition, lacking the somewhat arbitrary distinction between prime and nonprime attributes, is as follows.

Boyce-Codd Normal Form (BCNF). A relation R is in BCNF if it is in first normal form, and for every subset Γ of attributes of R, if some attribute not in Γ is FD on Γ, then all the attributes of R are FD on Γ.

An optimal set of 3NF relations is also proposed in [10] and [12] as the principal model of the database, an optimal set of relations being defined as that containing the fewest relations. The various normal forms 2NF, 3NF, and BCNF, proposed by Codd and his co-workers may be achieved through decompositions based on the following proposition.

PROPOSITION 2.1. *If* $\{\Lambda, \Gamma, \Delta\}$ *is a three-element partition of the attribute set of* $R(\Omega)$ *and* $\Gamma \rightarrow \Delta$, *then*

$$R(\Omega) = \Pi R(\Lambda \cup \Gamma) \cdot \Pi R(\Gamma \cup \Delta).$$

Thus the presence of a nontrivial FD in R is a sufficient, though not necessary, condition for its decomposition.

2.4 Design of Relational Database Schemata

The basic concept underlying the search for suitable normal forms may be described as an attempt to develop a design methodology for relational database schemata. One might recognize two main approaches to this effect: (1) the "case study" approach, and (2) the minimum cover techniques. The latter can in turn be divided into (a) the analytic and (b) the synthetic approach. We shall examine these in turn and suggest an alternate criterion for relational database design: complete data relatability.

2.4.1 *The Case Study Approach.* This approach which was considered in Section 2.3 is beset by two difficulties: (1) the possibility of multiple decompositions of certain relations, and (2) the general problem of defining as well as eliminating anomalies. We shall consider these in turn.

Turning back to relation (2.5), for instance, it may be seen that, in addition to (2.6) and (2.7), it allows a second decomposition based on the FD "AC# → EM." This decomposition yields the relations (2.8) and (2.9):

ΠACMG (AC#,	EM)
4218	A. MILLER
7531	A. MILLER
1537	B. BROWN
8532	T. LEIGH

(2.8)

ΠACMG (AC#,	TX)
4218	428
7531	428
1537	753
8532	915

(2.9)

Although this second schema is also formally in BCNF, it only aggravates the anomalies of (2.5). If the key of (2.9) cannot have an undefined value, the telephone extension of an employee can only be listed when he currently manages at least one account. Changing the telephone extension of an employee managing several accounts still requires the updating of multiple records of (2.9). Moreover, when an account is assigned to or taken from an employee, both relations (2.8) and (2.9) must be updated. Thus an analysis of the anomalies leads to the conclusion that decomposition (2.6) and (2.7) should be chosen.

When a relation possesses many decompositions, selecting the proper one through a complete analysis of the anomalies presented by the alternative schemata may be quite laborious. Such a procedure is also hampered by the lack of precise definition of an anomaly, for the semantic constraints characterizing many real-life situations are quite complex, and no clear-cut description of anomalies can be given without knowing the operations envisioned by the user. Thus consider, for example, relation (2.7). We might still find it anomalous due to the fact that a new telephone extension (TX) cannot be listed until it is assigned to an employee, or that it may be lost when an employee leaves the firm. One might argue that such happenings are quite unimportant and do not constitute a "real" anomaly. But this implies that the presence of an anomaly can only be recognized on an individual case basis and depends both on the semantics of the relation and on a user's usage of a relation rather than on the dependency structure of the relation.

2.4.2 *The Minimum Cover Technique.* A more systematic approach to the problem is supplied by the concept of *minimal cover* of the set of FDs of a relation. A *cover* is a set of FDs from which all others can be derived through the formal properties of FDs (Table I). The problem of obtaining an efficient minimal cover algorithm has received wide attention. A useful analogy between FDs and Boolean functions is utilized in [13] while the algorithm proposed in [5] uses an analogy betweed FDs and the production grammars of formal languages. These two papers also supply two interesting instances, the first of an analytic and the second of a synthetic approach to the design of a database schema. The approach taken in [13, 14, 16, 22] is clearly analytic. A relation is assumed to exist. Then, by analyzing the structure of FDs between the attributes of the relation, the designer can decompose it into a set of lower order relations satisfying certain requirements (such as being in 3NF for instance).

On the other hand, the approach taken in [26] and [6] is clearly synthetic. The database attributes and the set of FDs characterizing them are regarded as semantic primitives that the designer is expected to synthesize directly from his understanding of the case at hand. An algorithm is then used to obtain a minimal cover for the FDs of the database and an optimal set of 3NFs is finally constructed from it. The validity of the synthetic approach is limited by a problem acknowledged in [6], and which may be illustrated by the following example. A company is divided into departments (DP), subdivided into sections (SEC), each section and department having its own manager (MGR). Following the synthetic approach, we start by describing the various functional dependencies between these attributes. Thus we find the following FDs:

$$f_1\colon \text{SEC} \to \text{DP},$$
$$f_2\colon \text{DP} \to \text{MGR},$$
$$f_3\colon \text{SEC} \to \text{MGR}.$$

The application of the cover algorithm, which naturally operates on the basis of purely syntactic rules, would eliminate f_3 as the transitive composition of f_1 and f_2. However, this elimination would be semantically incorrect since f_3 defines the manager of a section and not the manager of a department. The analytic approach instead would reveal this problem immediately since the designer would find it impossible to enter the two distinct names of a department manager and of a section manager under the same column heading "MGR." This would force the designer either to use two different attributes for managers of sections, or else to change the statement of intension associated with the relation. Either solution will solve the problem. One could certainly argue that the same kind of semantic judgment could be used by the designer in assessing the validity of the elimination of f_3 from the cover. But the two situations are quite different. In the case of the analytic approach, once the designer accepts the fact that his relations represent plausible instances of the database content, he can proceed algorithmically with assurance of correctness. Under the synthetic approach every conclusion is tentative and subordinate to a semantic validity check.

2.4.3 *Complete Data Relatability.* In view of the problem discussed in Section 2.4.1 it appears that elimination of anomalies is not an adequate criterion for designing a database. We propose a much more stringent and general criterion for relational database design, that is, *the complete relatability of data.* We describe this concept by referring to our previous examples.

Consider, for example, relation (2.5). From the fact that AC# is underlined as a primary key we can only infer the FDs

$$\text{AC\#} \rightarrow \text{EM} \qquad \text{and} \qquad \text{AC\#} \rightarrow \text{TX}.$$

An examination of the relations (2.8) and (2.9) leads to the same conclusion. On the other hand the underlined keys of relations (2.6) and (2.7) yield

$$\text{AC\#} \rightarrow \text{EM} \qquad \text{and} \qquad \text{EM} \rightarrow \text{TX},$$

from which AC# → TX is inferred by transitivity. Thus relations (2.6) and (2.7) reveal an important dependency which was concealed in the other two schemata. We shall say that schemata (2.6) and (2.7) "ensure complete data relatability" whereas schemata (2.5), (2.8), and (2.9) do not.[2]

Similar considerations apply to relation (2.2). The schema represented by this relation reveals only the partial FD:

$$\{\text{SUPPLIER, ITEM}\} \rightarrow \text{COLOR}$$

whereas the much more significant

$$\text{ITEM} \rightarrow \text{COLOR}$$

cannot be inferred from (2.2). On the other hand decomposition of (2.2) into the pair

$$\Pi\text{STOCK}(\underline{\text{SUPPLIER}}, \underline{\text{ITEM}}) \qquad \text{and} \qquad \Pi\text{STOCK}(\underline{\text{ITEM}}, \text{COLOR})$$

[2] We define complete relatability in a precise mathematical fashion in Section 5.4.

depicts both the FD

$$\text{ITEM} \rightarrow \text{COLOR}$$

and the many-to-many relationship between parts and supplies revealed by the fact that SUPPLIER and ITEM are both needed to specify the key in the relation.

The criterion of complete data relatibility provides a unified basis for the general development of relational schemata. Thus, for example, normalization of hierarchies into 1NF cannot be justified on the basis of update anomalies, while it is a general consequence of the requirement for complete relatability. For indeed, any hierarchical organization is a specialized representation where certain associations between attributes are emphasized at the cost of hiding others. It is precisely this difficulty which led Codd to propose the use of 1NF which gives a uniform view of the data whereby the various associations can be derived with equal ease. We have also traced the data relatability justification for the 2NF and 3NF above. Furthermore, the various minimal cover algorithms find a natural justification in terms of data relatability. They can be viewed as processes for eliminating redundancy and pleonasms from the definition of the database schema while preserving the complete relatability data. The dependencies appearing in a cover, in fact, define implicitly all the others because of the formal properties of FDs.

Finally it may be reassuring to note that the decomposition of a relation according to the criterion of complete relatability will eliminate certain anomalies which are not removed by BCNF.

The concept of complete relatability therefore supplies a unified justification for the major concepts developed to design relational database schemata. It plays a major role in the following sections.

3. MULTIVALUED DEPENDENCIES

3.1 Many-to-Many Relationships

Functional dependencies can only model one-to-one and many-to-one relationships. Thus the natural many-to-many associations which often arise between database attributes are not properly modeled by functional dependencies. This problem has long been recognized (see, e.g., [13]), but early attempts to deal with it failed to produce any viable solution. The correct solution was finally achieved through the introduction of multivalued dependencies[3] which include the usual functional dependencies as a special case.

We shall preface the formal presentation by means of an example which will demonstrate the problem.

Let us consider the previous relation (2.2) with a modified statement of intension: We shall assume that whenever a part may have more than one color, every supplier of that part supplies all its colors. A possible snapshot of such a

[3] Multivalued dependencies were discovered independently by Fagin [15] and Zaniolo [27]. Previously Delobel and Léonard [14] had defined the related concept of "first-order hierarchical decomposition."

relation is as follows:

MSTOCK(SUPPLIER,	ITEM,	COLOR)
WOODMAN	CHAIR	BROWN
WOODMAN	TABLE	BLACK
WOODMAN	SOFA	BROWN
HOUSEMAN	DESK	GREEN
HOUSEMAN	SOFA	BROWN
HOUSEMAN	CARPET	RED
HOUSEMAN	CARPET	YELLOW
HOUSEMAN	CARPET	BLUE
BLAND	CARPET	RED
BLAND	CARPET	YELLOW
BLAND	CARPET	BLUE

(3.1)

Since no FD exists between the attributes of that relation, the only candidate key is the combination of all three attributes.

Schema (3.1) is misleading since it appears to imply that each supplier only produces a few colors of a given part. Instead, we actually have a many-to-many relationship between supplier and items, and a similar relationship between items and colors. Furthermore, all the anomalies of relation (2.2) are present here. Since no attribute composing the primary key may have an undefined value, the colors of an item only appear when there is at least one supplier for that part. Adding, deleting, or changing one of the colors of any part requires searching through the file and updating a record for every supplier of that part (since every supplier will have to supply the new set of colors). Remarkably, relation (3.1) is decomposable into the pair

ΠMSTOCK(SUPPLIER,	ITEM)
WOODMAN	CHAIR
WOODMAN	TABLE
WOODMAN	SOFA
HOUSEMAN	DESK
HOUSEMAN	SOFA
HOUSEMAN	CARPET
BLAND	CARPET

(3.2)

ΠMSTOCK(ITEM,	COLOR)
CHAIR	BROWN
TABLE	BLACK
SOFA	BROWN
DESK	GREEN
CARPET	RED
CARPET	YELLOW
CARPET	BLUE

(3.3)

The relations (3.2) and (3.3) describe the natural relationships of the database far better than (3.1): These relations show immediately that there exists a many-to-many relationship between SUPPLIER and ITEM and between ITEM and COLOR. Therefore, it seems obvious that a designer should use the pair (3.2) and

(3.3) rather than the original (3.1). However, he will find no guidance from FD-oriented concepts concerning relational database schemata since relation (3.1) is formally in BCNF. In the case of relation (2.5), for instance, the designer could detect, from the statement of intension of the relation, the pattern of partial FD which was previously described as a source of anomalies. Similarly, when confronted with relations such as (3.1), the designer needs the definition of a pattern of logical dependencies which allow an immediate recognition of the irregular nature of such relations. Since the logical dependencies characterizing (3.1) are not definable in terms of FDs, we need a new kind of dependency between attributes in order to describe the structure of (3.1). From the statement of intension of relation (3.1) we derive directly the fact that the set of colors of an item is the same for every supplier of that item. Therefore, the set of colors of an item is effectively a function of ITEM only and not of supplier. We therefore speak of a *multivalued dependency* (MD) of the attribute COLOR upon the attribute ITEM. The peculiarity of relation (3.1) is that COLOR is multidependent (MD) on ITEM alone. A relation where COLOR is MD on the full combination {SUPPLIER, ITEM} is not decomposable and is also free from the above-mentioned anomalies.

3.2 Definition of Multivalued Dependencies

We shall now define the concept of MD between two (possibly nondisjoint) subsets of attributes of a relation $R(\Omega)$.

If Θ and Δ are two attribute combinations in a relation $R(\Omega)$, and $r \in R$, then the set of Δ-values associated with the value $r[\Theta]$ is denoted by $M_\Delta(r[\Theta])$. Formally, we may write

$$M_\Delta(r[\Theta]) = \{r'[\Delta] \mid r' \in R \text{ and } r'[\Theta] = r[\Theta]\}.$$

Therefore, M_Δ denotes the R-induced mapping from $\Pi R(\Theta)$ *into* the *family of subsets* of $\Pi R(\Delta)$. When $\{\Theta, \Delta\}$ is a dichotomy of Ω, $M_\Delta(r[\Theta])$ is called the *image set* of $r[\Theta]$ under R. For instance, in relation (3.1), if Θ = {SUPPLIER, ITEM} and Δ = {COLOR}, then

$$M_\Delta(\text{HOUSEMAN, CARPET}) = \{\text{RED, YELLOW, BLUE}\}.$$

Let Γ and Δ be two combinations of attributes of $R(\Omega)$, and let Λ be their complement: $\Lambda = \Omega - (\Gamma \cup \Delta)$. Let M_Δ denote the R-induced mapping from $\Pi R(\Gamma \cup \Lambda)$ into families of subsets of $\Pi R(\Delta)$. Then Δ will be said to be *multidependent* on Γ in R iff, for each pair of tuples $r_1, r_2 \in \Pi R(\Lambda \cup \Gamma)$, the following is true:

$$(r_1[\Gamma] = r_2[\Gamma]) \Rightarrow (M_\Delta(r_1) = M_\Delta(r_2)).$$

When $|M_\Delta| = 1$ for any value of the combination $(\Lambda \cup \Gamma)$, this definition reduces to the definition of FD. Thus FDs are a special case of MDs. When $\Lambda = \varnothing$ (i.e., $\Gamma \cup \Delta = \Omega$), or when $\Delta \subseteq \Gamma$, the previous definition is satisfied in any relation. These MDs are named trivial. We use the notation

$$\Gamma \twoheadrightarrow \Delta \qquad \text{and} \qquad \Gamma \not\twoheadrightarrow \Delta$$

to indicate that Δ is or is not MD on Γ. It follows from the previous definition that $\Gamma \twoheadrightarrow \Delta$ in $R(\Omega)$ if and only if $M_\Delta(r) = M_\Delta(r[\Gamma])$ for each $r \in \Pi R(\Gamma \cup \Lambda)$. This alternative definition of MD is used in [2].

Relation (3.1), for instance, is characterized by a nontrivial MD. Indeed, ITEM $\twoheadrightarrow$ COLOR since the set of colors of an item is the same for any supplier of that item.

Directly from the definition of MD we can derive that $\Gamma \twoheadrightarrow \Delta$ if and only if $\Gamma \twoheadrightarrow (\Delta - \Gamma)$. This property holds for the case in which $\Gamma \supseteq \Delta$, as we assume that "$\Gamma \to \varnothing$" (where $\varnothing$ is the empty set) is a valid, although trivial, MD. Because of this property we could build a theory of multidependencies assuming that the left- and right-hand-side attribute combinations are disjoint. This restriction, however, is not necessary. Thus we opt for the more general definition which provides a more uniform treatment of MDs and FDs. A further generalization of the MD concept can, moreover, be achieved by letting the left side be the empty set. Therefore, $\varnothing \twoheadrightarrow \Delta$ is valid in $R(\Omega)$ if the Δ-values in the relation are independent of the Λ-values, where $\Lambda = \Omega - \Delta$. This occurs whenever $M_\Delta(r_1) = M_\Delta(r_2)$ for any $r_1, r_2 \in \Pi R(\Lambda)$. For a further discussion of this useful generalization, the reader should refer to [15].

We say that $R(\Omega)$ is *decomposable* into the pair of projections $\Pi R(\Omega_1)$ and $\Pi R(\Omega_2)$ when

(1) Ω_1 and Ω_2 are proper subsets of Ω, and
(2) $R(\Omega) = \Pi R(\Omega_1) \cdot \Pi R(\Omega_2)$.

We now present a fundamental theorem.

PROPOSITION 3.1. *Let Γ and Δ be subsets of Ω and let $\Lambda = \Omega - (\Gamma \cup \Delta)$. Then $\Gamma \twoheadrightarrow \Delta$ in $R(\Omega)$ iff*

$$R(\Omega) = \Pi R(\Lambda \cup \Gamma) \cdot \Pi R(\Gamma \cup \Delta).$$

The proof of this proposition is omitted since it has already appeared in [27] and [15]. If Γ is empty, then the join reduces to a Cartesian product and $R(\Omega)$ is equal to the Cartesian product of $\Pi R(\Lambda)$ and $\Pi R(\Delta)$. As a direct corollary of Proposition 3.1, it follows that the presence of a nontrivial MD in a relation is both a sufficient and a necessary condition for its decomposability into a pair of subprojections. A relation which contains only trivial MDs is called *atomic*. Such a relation is in fourth normal form (4NF) as defined by Fagin [15]. However, note that a relation in 4NF is not necessarily atomic.

The symmetry of Λ and Δ in Proposition 3.1 yields the following corollary.

PROPOSITION 3.2. *In a relation $R(\Omega)$, $\Gamma \twoheadrightarrow \Delta$ iff $\Gamma \twoheadrightarrow \Lambda$, where*

$$\Lambda = \Omega - (\Delta \cup \Gamma).$$

This is called the complementation property of MDs.

3.3 Formal Properties of MDs

The multivalued dependencies of a relation $R(\Omega)$ have the following properties.

M0. *Reflexivity*: If $\Gamma \supseteq \Delta$, then $\Gamma \twoheadrightarrow \Delta$.

M1. *Complementation:*[4] $\Gamma \twoheadrightarrow \Delta$ iff $\Gamma \twoheadrightarrow \Lambda$, where $\Lambda = \Omega - (\Gamma \cup \Delta)$.

From M0 and M1 we derive that all dependencies of the form $\Gamma \twoheadrightarrow \Delta$, where either $\varnothing \subseteq \Delta \subseteq \Gamma$ or $\Gamma \cup \Delta = \Omega$, are valid in any $R(\Omega)$, (*trivial* MDs). Moreover, we derive that $\Gamma \twoheadrightarrow \Delta$ iff $\Gamma \twoheadrightarrow (\Delta - \Gamma)$. As was proved in [27] and [2], MDs also have the following properties.

M2. *Augmentation*: If $\Gamma \twoheadrightarrow \Delta$ and $\Lambda \supseteq \Psi$, then $(\Gamma \cup \Lambda) \twoheadrightarrow (\Delta \cup \Psi)$.
M3. *Transitivity*: If $\Gamma \twoheadrightarrow \Delta$ and $\Delta \twoheadrightarrow \Psi$, then $\Gamma \twoheadrightarrow (\Psi - \Delta)$.

Since $\Gamma \twoheadrightarrow \Psi$ is not always valid the analogy to FDs is not complete here. Indeed the more precise term, such as "restricted transitivity," could be used [27]. The validity of $\Gamma \twoheadrightarrow \Psi$ can, however, be guaranteed under supplementary assumptions. For instance, if $\Delta \cap \Psi = \varnothing$, then $\Gamma \twoheadrightarrow \Psi$. Also, $\Gamma \twoheadrightarrow \Psi$ when $\Gamma \rightarrow \Delta$. In this case $\Gamma \rightarrow (\Delta \cap \Psi)$ by distributivity and $\Gamma \twoheadrightarrow \Psi$ by the additivity property of MDs (additivity is proved in M7). Similar statements can be made for the pseudotransitivity property.

M4. *Pseudotransitivity*: If $\Gamma \twoheadrightarrow \Delta$ and $(\Delta \cup \Lambda) \twoheadrightarrow \Psi$, then $(\Gamma \cup \Lambda) \twoheadrightarrow (\Psi - \Delta)$.
M5. *General Composition*: If $\Gamma \twoheadrightarrow \Delta$ and $\Lambda \twoheadrightarrow \Psi$ where $\Lambda \subseteq (\Gamma \cup \Delta)$, then $\Gamma \twoheadrightarrow (\Psi - \Delta)$.

PROOF. $(\Gamma \cup \Delta) \twoheadrightarrow \Psi$ by augmentation on $\Lambda \twoheadrightarrow \Psi$. But since $\Gamma \twoheadrightarrow \Delta$, then $\Gamma \twoheadrightarrow (\Psi - \Delta)$ by pseudotransitivity. Q.E.D.

MDs do not have the distributivity property with respect to union; however, they have the following property.

M6. *Partitionability*: If $\Gamma \twoheadrightarrow \Delta$ and $\Gamma \twoheadrightarrow \Lambda$, then

(1) $\Gamma \twoheadrightarrow (\Delta - \Lambda)$,
(2) $\Gamma \twoheadrightarrow (\Lambda - \Delta)$,
(3) $\Gamma \twoheadrightarrow (\Delta \cap \Lambda)$.

Another important property establishes a pattern of implication from the MDs in a relation to the MDs in its projections.

M7. *Projectability*: If $\Gamma \twoheadrightarrow \Delta$ in $R(\Omega)$ and $\Gamma \subseteq \Psi \subseteq \Omega$, then $\Gamma \twoheadrightarrow (\Delta \cap \Psi)$ in $\Pi R(\Psi)$.

This property can be derived quite easily from the definitions. A detailed proof can be found in [27].

In contradiction to FDs the implication from a projection to the original relation does not hold. Thus MDs do not have the reverse projectability property, a fact which is more easily clarified by an example. Consider the following relation:

PS (PD,	SUP,	P#,	PRICE)
8JKFF	MOT	4267	1.15
8JKFF	FC	4267	0.97
4KRAM	TI	1135	3.75

(3.4)

[4] Although this statement of complementation is weaker than the one used in [2], the two rules are actually equivalent. Furthermore, Biskup [7] has proved that replacing complementation by the rule "$\varnothing \rightarrow \Omega$" still obtains a complete set.

The attributes are the part numbers (P#), the suppliers of those parts (SUP), the part description (PD), and the wholesale price of the parts (PRICE). Furthermore, we find that P# $\rightarrow$ PD but {P#, SUP} $\not\rightarrow$ PRICE, since different suppliers may charge a different price for the same part, as seen from the snaphsot given above. Therefore, in relation (3.4) PRICE is not MD on P# alone, that is, P# $\not\twoheadrightarrow$ PRICE. Thus, for example,

$$M_{PRICE}(8JKFF, MOT, 4267) \neq M_{PRICE}(8JKFF, FC, 4267).$$

However, in the projection

ΠPS (PD,	P#,	PRICE)
8JKFF	4267	1.15
8JKFF	4267	0.97
4KRAM	1135	3.75

(3.5)

we have P# $\rightarrow$ PD. Also, P# $\twoheadrightarrow$ PRICE in (3.5) while P# $\not\twoheadrightarrow$ PRICE in (3.4). This example shows that an MD in a projection does not imply a corresponding MD in the original relation. It also demonstrates that the distributive property is not valid for MDs. Thus in relation (3.4), for example, P# $\rightarrow$ PD; therefore P# $\twoheadrightarrow$ {SUP, PRICE}. Yet P# $\not\twoheadrightarrow$ PRICE. Intuitively speaking this means that the MD expresses an irreducible association between P#, SUP, and PRICE and not two independent relationships between P# and SUP and between P# and PRICE.

3.4 A Complete Axiomatization for FDs and MDs

Reflexivity, augmentation, and transitivity form a complete set of inference rules for FDs. The closure of a set F of FDs, denoted F^+, is the set of all FDs derivable from F using these rules. The strong completeness theorem which follows from Armstrong's results [1] can be stated as follows: *For every set F of FDs, there exists a relation such that the set of FDs that are valid in it is exactly* F^+.

A similar strong completeness theorem for MDs was proved in [2]. Complementation, reflexivity, augmentation, and transitivity supply a complete set of inference rules for MDs. If G is a set of MDs valid in $R(\Omega)$, its closure set, denoted G^+, is the set of all MDs derivable from G using these rules.

Mixed Rules. Assume that a set F of FDs of a relation $R(\Omega)$ is given together with a set G of MDs. Additional dependencies which could not be derived from F or from G separately, might be derived from F and G jointly.

Some MDs might be deduced by the very fact that every FD is also an MD.

MX1. If $\Gamma \rightarrow \Delta$, then $\Gamma \twoheadrightarrow \Delta$.

$\Gamma \twoheadrightarrow \Delta$ will be called the *MD counterpart* of $\Gamma \rightarrow \Delta$. Moreover, some new FDs might be inferable by the following.

MX2. If $\Gamma \twoheadrightarrow \Lambda$ and $\Delta \rightarrow \Psi$, where $\Lambda \supseteq \Psi$ and $\Lambda \cap \Delta = \emptyset$, then $\Gamma \rightarrow \Psi$.

The proof of this second rule can be found in [2] along with the proof that {F1, F2, F3, M1, M2, M3, MX1, MX2}[5] is a complete set of inference rules for the

[5] This set does not contain M0(MD reflexivity) since M0 may be derived from F1 and MX1.

Table I. Formal Properties of FDs and MDs

Property	FD	MD
Reflexivity	Yes	Yes
Augmentation	Yes	Yes
Additivity	Yes	Yes
Distributivity	Yes	No[a]
Transitivity	Yes	Yes
Pseudotransitivity	Yes	Yes
General composition	Yes	Yes
Complementation	No	Yes
Projectability	Yes	Yes
Reverse projectability	Yes	No

[a] A weaker property called partitionability holds for MDs.

family of functional and multivalued dependencies. The closure of $F \cup G$, denoted $(F, G)^+$, consists of (1) the set of all FDs and (2) the set of all MDs which are derivable from F and G by repeated applications of those eight rules. The strong completeness theorem by Beeri et al. can be stated as follows: *For any sets F of FDs and G of MDs (on the attribute set Ω) there exists a relation $R(\Omega)$ such that the set of dependencies valid in R is exactly $(F, G)^+$.*

Let G and G' be sets of dependencies of a relation, while $P_1, P_2, \ldots, P_r$ are some inference rules or properties for these dependencies. The set G' is said to be a *cover for G*, according to these properties, if every dependency in G is inferable from G' by the application of these rules. When no proper subset of G' is a cover, then G' is said to be a *minimal cover.*

Table I summarizes the formal properties of FDs and MDs.

4. ELEMENTARY DEPENDENCIES

Following the analytical approach, we assume that the database is modeled as a set of relations, each accompanied by an unambiguous statement of its intension and by a typical sample of its content. From these, the designer must detect the dependencies of each relation and, using this information, design the right schema for this database. In this section we introduce the concept of *elementary FDs* and of *multiple elementary MDs* which (1) simplify the designer task of detecting the dependencies of a relation, and (2) supply the foundation for the schema design algorithm discussed in the following sections.

Clearly the quality of the schema produced by an algorithm driven by the dependency structure of a relation depends upon the correctness and completeness with which the designer initially characterized these dependencies. While no absolute guarantee against human error can ever be offered, it is clear that the designer is apt to perform most reliably when he need only be concerned with few and simple dependencies appealing directly to his intuition. As we shall see, elementary FDs and multiple elementary MDs have the following properties: (1) they constitute a small subset of all the FDs and MDs valid in a relation; (2) they have a simple nondecomposable structure which makes them appeal to intuition; (3) they have a number of formal properties which are very useful for schema design; (4) they provide the same information as all the FDs and MDs of the relation (in fact we show that all the FDs and MDs of a relation are inferable from its elementary FDs and its multiple elementary MDs by the inference rules discussed in the previous section).

In the discussion which follows we shall find it useful to refer to the following example.

The field service for a computer manufacturer is interested in the following attributes

Customer number:	CUST#
Customer name:	CUSTN
Computer model:	MODEL
Quantity of a given model:	MODQ
Technician number:	TEC#
Technician name:	TECN

and in the following information:

(1) the name of each customer;
(2) the name of each technician;
(3) the computer models, and their quantity, used by each customer;
(4) the set of technicians assigned to a customer (assume that every technician assigned to a customer can repair all the computer models owned or rented by that customer).

Thus a composite relation defined on those attributes permits the following snapshot of its content:

FIELD (CUST#,	CUSTN,	MODEL,	MODQ,	TEC#,	TECN)
351	WOODS	A	3	4562	SMITH
351	WOODS	A	3	3333	FISHER
351	WOODS	B	1	4562	SMITH
351	WOODS	B	1	3333	FISHER
552	IRON	C	1	7532	FISHER
891	LEAD	B	1	4562	SMITH

(4.1)

The statement of intension and the sample content of our relation yield the following dependencies (which are not the only ones, however).

D1. CUST# $\rightarrow$ CUSTN
D2. CUST# $\rightarrow\!\!\!\rightarrow$ {MODEL, MODQ}
D3. CUST# $\rightarrow\!\!\!\rightarrow$ {TEC#, TECN}
D4. {CUST#, MODEL} $\rightarrow$ MODQ
D5. TEC# $\rightarrow$ TECN
D6. TEC# $\rightarrow\!\!\!\rightarrow$ {CUST#, CUSTN, MODEL, MODQ}.

(4.2)

4.1 Elementary Dependencies

A uniform treatment of MDs and FDs can be developed on the basis of their augmentation and additivity properties. Let G denote the set of all MDs of $R(\Omega)$ where the right side of each MD is not empty and is disjoint from the left side. Thus G is a set of *ordered* pairs (Γ, Δ) where $\Gamma \rightarrow\!\!\!\rightarrow \Delta$ in $R(\Omega)$ and $\Delta - \Gamma = \Delta$,

$\Delta \neq \varnothing$. Moreover, let F be the set of ordered pairs denoting the FDs of R with disjoint left and right sides. Obviously, $F \subseteq G$.

Let us next define the following partial order among ordered pairs of subsets of Ω:

$$(\Gamma, \Delta) \leq (\Lambda, \Psi) \quad \text{if} \quad \Gamma \subseteq \Lambda \quad \text{and} \quad \Delta \subseteq \Psi. \tag{4.3}$$

The minimum members of G according to (4.3) are called the *elementary* MDs of $R(\Omega)$. They form a set denoted by $\bar{G}$ (the maximum members of G are the various dichotomies of Ω). Thus "$\Gamma \twoheadrightarrow \Delta$" is elementary iff $\Gamma \cap \Delta = \varnothing$ and there exists no distinct "$\Gamma' \twoheadrightarrow \Delta'$" where $\Gamma' \subseteq \Gamma$ and $\Delta' \subseteq \Delta$. The set F is also ordered by (4.3) in a completely similar fashion. The minimum members of F are called *elementary* FDs.[6] They form a set denoted by $\bar{F}$. It follows from the distributivity property that elementary FDs have the form "$\Gamma \rightarrow A$," where $A \notin \Gamma$ and $\Gamma' \not\rightarrow A$ for every $\Gamma' \subset \Gamma$.

We can now state a theorem which can be regarded as an extension to MX1.

PROPOSITION 4.1. *The MD counterpart of an elementary FD is an elementary MD.*

PROOF. By contradiction assume that $\Gamma \rightarrow A$ is an elementary FD while $\Gamma \twoheadrightarrow A$ is not an elementary MD. Then there must exist a $\Gamma' \twoheadrightarrow A$ where $\Gamma' \subset \Gamma$. But then it follows from MX2 that $\Gamma' \rightarrow A$. That would contradict our original assumption. Q.E.D.

PROPOSITION 4.2. *$\bar{F}$ and $\bar{G}$ are, respectively, the minimal covers for F and G according to additivity and augmentation.*

PROOF. Given a $(\Gamma, \Delta) \in G$, we can construct $G' \subseteq \bar{G}$ as follows:

$$G' = \{(\Gamma', \Delta') \in \bar{G} \mid (\Gamma', \Delta') \leq (\Gamma, \Delta)\}.$$

If Ψ denotes the union of the right side of the MDs in G', then by augmentation and additivity we infer $\Gamma \twoheadrightarrow \Psi$. We want to show that $\Psi = \Delta$. Clearly, $\Psi \subseteq \Delta$. We know that the MD $\Gamma \twoheadrightarrow (\Delta - \Psi)$ is in G according to M6. If $\Psi \subset \Delta$, then there exists an elementary $\Gamma'' \twoheadrightarrow \Delta''$ where $(\Gamma'', \Delta'') \leq (\Gamma, \Delta - \Psi)$. But (Γ'', Δ'') must then belong to G' and $\Psi \supseteq \Delta''$. But $\Delta'' \subseteq \Delta - \Psi$, a contradiction. Thus $\bar{G}$ is a cover for G according to additivity and augmentation. To prove minimality and uniqueness, we only need to observe that in M2 and M7 the implied dependency is always greater in terms of (4.3) than its implicant or implicants. But since every MD in $\bar{G}$ is minimal in terms of (4.3), it cannot be derived from other MDs using M2 and M7. Thus not only $\bar{G}$ is the minimal cover of G by additivity and an augmentation, but it is also the only subset of G which has this property. In a similar fashion we can prove that $\bar{F}$ is the minimal cover of F by additivity and augmentation utilizing, however, the distributivity property instead of M6. Q.E.D.

If we consider all the dependencies of a relation (including those where the left and the right sides overlap), then it follows from Proposition 4.2 that $\bar{F}$ and $\bar{G}$

[6] The concept of elementary FD, unlike that of elementary MD, is not new. For instance, in [13] we find them defined as "elementary functional relationships."

are, respectively, the minimal covers for these FDs and MDs by reflexivity, additivity, and augmentation. Since these properties do not constitute a complete set of inference rules, however, smaller covers can be constructed.[7] The problem of including complementation for MDs is considered in the next section.

Elementary dependencies can be regarded as noncompound associations involving a minimal set of attributes. Any other nontrivial dependency is, in fact, obtained by adding some attributes to an elementary one and/or by combining them according to the additivity law. For this reason, when considering the various dependencies of a relation, we think more naturally in terms of elementary dependencies. All six dependencies listed in (4.2) are elementary.

4.2 Dependencies with Nested Left Sides

The properties of MDs with nested left side are very important for understanding and analyzing the logical structure of relations. Let us denote by $D(\Gamma)$ the family of *all* nonempty sets Δ for which $\Gamma \twoheadrightarrow \Delta$. The minimum elements of this family according to the ordering defined by set containment is denoted $\bar{D}(\Gamma)$.[8] It follows from M6 and M7 that $D(\Gamma)$ is closed under union and intersection. Thus $\bar{D}(\Gamma)$ is a partition of Ω. A singleton set $\{A\}$ appears in $\bar{D}(\Gamma)$ for every $\Gamma \rightarrow A$. Actually if Γ^* (where $\Gamma^* \supseteq \Gamma$) denotes the set of all attributes in Ω which are FD on Γ, then $D(\Gamma^*) = D(\Gamma)$ [2].

In the case of the relation FIELD (4.1) we have

$$\bar{D}(\{\text{CUST\#, MODEL}\}) = \{\{\text{CUST\#}\}, \{\text{MODEL}\}, \{\text{CUSTN}\}, \{\text{MODQ}\}, \{\text{TEC\#, TECN}\}\}$$

while

$$\bar{D}(\{\text{CUST\#}\}) = \{\{\text{CUST\#}\}, \{\text{CUSTN}\}, \{\text{MODEL, MODQ}\}, \{\text{TEC\#, TECN}\}\}.$$

If A and B are two partitions of the same set, A is said to be *at least as fine as* B when, for each $a \in A$ there exists an element $b \in B$ such that $b \supseteq a$. If, furthermore, $A \neq B$, then A is said to be *finer* than B. That is to say that B was obtained by merging some of the elements of A. We can now state the following.

PROPOSITION 4.3. *If* $\Gamma \supseteq \Gamma'$, *then* $\bar{D}(\Gamma)$ *is at least as fine a partition of* Ω *as* $\bar{D}(\Gamma')$.

PROOF. We prove that for any pair $\Delta \in \bar{D}(\Gamma)$ and $\Delta' \in \bar{D}(\Gamma')$ either $\Delta' \supseteq \Delta$ or $\Delta' \cap \Delta = \varnothing$. Indeed, augmenting $\Gamma' \twoheadrightarrow \Delta'$ and combining into $\Gamma \twoheadrightarrow \Delta$ according to M6, we obtain $\Gamma \twoheadrightarrow (\Delta \cap \Delta')$. Thus either $\Delta \cap \Delta' = \Delta$ or $\Delta \cap \Delta' = \varnothing$, or we contradict the assumption that $\Delta \in \bar{D}(\Gamma)$. Q.E.D.

The family of all nonempty sets Δ for which $\Gamma \twoheadrightarrow \Delta$ is an elementary MD is denoted $E(\Gamma)$.

The union of all sets in $E(\Gamma)$ is denoted Γ^-. In general we have $\varnothing \subseteq E(\Gamma) \subseteq \bar{D}(\Gamma)$. In particular, if Γ is such that every MD with left side properly contained in Γ is trivial, then $E(\Gamma) = \bar{D}(\Gamma) - U(\Gamma)$, where $U(\Gamma)$ denotes the unit partition of Γ. In relation (4.1), for example, we find that $E(\{\text{CUST\#}\})$ contains all the

[7] For instance, in the set of elementary FDs, $\{g_1: A \rightarrow B, g_2: B \rightarrow C, g_3: A \rightarrow C\}$, g_3 is derivable from g_1 and g_2 by transitivity. The analogous property holds for the MD counterpart of these FDs.

[8] These concepts are taken from [2] where $\bar{D}(\Gamma)$ is called the dependency basis of Γ.

elements of $\bar{D}(\{\text{CUST\#}\})$ except, of course $\{\text{CUST\#}\}$. Thus $\{\text{CUST\#}\}^- = \{\text{CUSTN, MODEL, MODQ, TEC\#, TECN}\}$ whereas $E(\{\text{CUST\#, MODEL}\}) = \{\{\text{MODQ}\}\}$ and $\{\text{CUST\#, MODEL}\}^- = \{\text{MODQ}\}$.

We now have the following important theorem.

PROPOSITION 4.4. *Let $\Gamma \supset \Gamma'$ and let $\Lambda = \Gamma - \Gamma'$. Then, if $E(\Gamma)$ is not empty, there exists a $\Psi \in E(\Gamma')$ such that $\Psi \supseteq (\Gamma^- \cup \Lambda)$.*

PROOF. Given $\Delta \in E(\Gamma)$, by Proposition 4.3, there exists a $\Psi \in \bar{D}(\Gamma')$ such that $\Psi \supseteq \Delta$. We prove that $\Psi \supseteq (\Gamma^- \cup \Lambda)$ and that $\Gamma' \twoheadrightarrow \Psi$ is elementary. In order to prove that $\Psi \supseteq \Lambda$, we can augment and complement the last MD to yield $[\Gamma' \cup (\Lambda \cap \Psi)] \twoheadrightarrow \bar{\Psi}$ where $\bar{\Psi}$ $(\Omega - \Psi)$. Since $[\Gamma' \cup (\Lambda \cap \Psi) \cup \bar{\Psi}] \supseteq [\Gamma' \cup \Lambda]$, we can compose this last MD with $(\Gamma' \cup \Lambda) \twoheadrightarrow \Delta$, according to M5. Thus we obtain $[\Gamma' \cup (\Lambda \cap \Psi)] \twoheadrightarrow (\Delta - \bar{\Psi})$. Since $\Psi \supseteq \Delta$, we have $\Delta - \bar{\Psi} = \text{D}$. Then either $\Lambda \cap \Psi = \Lambda$ (i.e., $\Psi \supseteq \Lambda$), or the assumption that $(\Gamma' \cup \Lambda) \twoheadrightarrow \Delta$ is elementary is violated. In order to prove that $\Psi \supseteq \Gamma^-$, observe that if there exists another $\Delta' \in E(\Gamma)$ where $\Delta \neq \Delta'$, then by the previous reasoning there exists a $\Psi' \in \bar{D}(\Gamma')$ where $\Psi' \supseteq (\Delta' \cup \Lambda)$. Thus $\Gamma' \twoheadrightarrow (\Psi \cap \Psi')$ where $(\Psi \cap \Psi') \neq \varnothing$. This contradicts the assumption that $\Psi \in \bar{D}(\Gamma')$ unless $\Psi = \Psi'$. Hence Ψ contains each class in $E(\Gamma')$; therefore it also contains Γ^-. To prove that $\Gamma' \twoheadrightarrow \Psi$ is elementary, we can use what already has been proved, namely, that if $\Gamma'' \twoheadrightarrow \Psi$, with $\Gamma'' \subset \Gamma'$, then $\Psi \supseteq \Gamma - \Gamma''$. Hence it follows that $\Psi \supseteq \Gamma' - \Gamma'' \supset \varnothing$, which is a contradiction since Γ' and Ψ are disjoint. Q.E.D.

Proposition 4.4 has some important implications in the development of a methodology for analyzing the dependency structure of a relation. The following is a corollary of Proposition 4.4.

PROPOSITION 4.5. *If $E(\Gamma)$ is empty, so is every $E(\Psi)$ with $\Psi \supseteq \Gamma$.*

We can also state the following.

PROPOSITION 4.6. *Let $\bar{G}$ be the set of elementary MDs of a relation R. For each nonempty $E(\Gamma)$ choose an arbitrary $\Delta \in E(\Gamma)$ and remove $\Gamma \twoheadrightarrow \Delta$ from $\bar{G}$. Any subset of $\bar{G}$ so constructed is a minimal cover for the MDs of R according to reflexivity, augmentation, additivity, and complementation.*

PROOF. Consider Γ^- where $E(\Gamma)$ is not empty. If $\Gamma^- = \Omega$, then $\Gamma \twoheadrightarrow \Gamma^-$ is trivial. Otherwise, by augmentation and additivity on the elementary MDs with the left side properly contained in Γ, we derive $\Gamma \twoheadrightarrow (\Omega - \Gamma^-)$. This yields $\Gamma \twoheadrightarrow \Gamma^-$ by complementation. Now $E(\Gamma)$ is a partition of Γ^-; thus one set in $E(\Gamma)$, but not more than one, is inferable by Boolean algebra from the remaining $|E(\Gamma)| - 1$ sets and Γ^-. To prove minimality, say that $g_1 : \Gamma \twoheadrightarrow \Delta_1$ and $g_2 : \Gamma \twoheadrightarrow \Delta_2$ are two elementary MDs and that G_{12} denotes the set of elementary MDs of R with the left side $\Gamma' \subseteq \Gamma$, g_1 and g_2 excluded. By Proposition 4.4, the right side, say Ψ, of each MD in G_{12} must satisfy the following condition:

$$\Psi \supseteq (\Delta_1 \cup \Delta_2) \qquad or \qquad \Psi \cap (\Delta_1 \cup \Delta_2) = \varnothing. \tag{4.4}$$

Now if we complement an MD which satisfies (4.4) or we augment it to an MD with the left side not greater than Γ, we obtain an MD whose right side satisfies (4.4). Also if we add two MDs where the right side satisfies (4.4), we obtain an

MD whose right side satisfies (4.4). Thus it is impossible to construct g_1 or g_2 from G_{12} by these rules. Q.E.D

4.3 Multiple Elementary MDs

The elementary MDs of a relation can be further subdivided into *single* elementary MDs and *multiple* elementary MDs. Single elementary MDs are defined such that no other elementary MD with the same left side exists. Multiple elementary MDs are those for which there exist one or more elementary MDs with the same left side. Thus multiple elementary MDs always come in groups of two or more. For instance, we find that D1, D2, and D3 in (4.2) define a first group of multiple elementary MDs for FIELD. D5 and D6 define a second such group. On the other hand D4 is a single elementary MD.

The distinction between single and multiple elementary MDs is important for schema design and for the analysis of the dependencies of relations. Indeed we present later a decomposition algorithm for schema design which is driven by the elementary FDs and the multiple elementary MDs of the relation, whereas single elementary MDs are not used in the decomposition. We also show that this method produces decompositions having desirable properties which could be lost otherwise. For the present, however, we shall only point out the difference between single and multiple MDs in terms of minimal covers.

Starting with the set of all the MDs of a relation, or any superset of its elementary MDs, the following algorithm can be used to obtain the minimal covers for the MDs of this relation according to reflexivity, additivity, augmentation, and complementation.

Step 1. Eliminate those MDs wherein the right side is empty or has some attribute in common with the left side.
Step 2. Find the minimum MDs in terms of the partial order defined in (4.3). Eliminate all other MDs.
Step 3. Partition the remaining MDs into groups such that all the members of a group have the same left side.
Step 4. Eliminate an arbitrary member from each group.

According to Proposition 4.2 the set obtained at the end of step 2 consists of all the elementary MDs of the relation. Thus by Proposition 4.6, at the end of step 4, we obtain indeed a minimal cover of the MDs of the relation according to reflexivity, additivity, augmentation, and complementation. Now while the results obtained are unique up to step 3, step 4 is clearly nondeterministic: For each group of n elementary MDs of identical left side there are n possible choices as to which member should be eliminated. We present criteria for the proper choice of such elimination within the framework of the decomposition algorithm presented in the next section. Yet for groups containing only one elementary MD ($n = 1$), no choice is necessary. These elementary MDs will always be eliminated no matter which cover is chosen. Thus we can carry out step 4 to the extent that all single elementary MDs are eliminated while all the multiple elementary MDs are retained. In a sense this is equivalent to using the minimal cover approach as far as possible, while preserving uniqueness and avoiding premature commitment.

The concepts of single and multiple MDs can also help the designer obtain more easily a complete characterization of the dependency structure of the relation at hand by encouraging him to search for the fewest and simplest dependencies rather than have him list a plethora of dependencies and then eliminate the redundant ones by minimal cover algorithms. The fact that the designer can neglect the single elementary MDs introduces a great simplification. In particular, every MD of $R(\Omega)$ of the type $\Gamma \twoheadrightarrow (\Omega - \Gamma)$ can be discarded. (Although this is a trivial MD it can be elementary according to the definition.) Moreover, Proposition 4.4 can be particularly useful in the search for multiple elementary MDs. Assume, for instance, that we want to find the multiple elementary MDs with the left side Γ after all the elementary MDs with a certain left side Γ', where $\Gamma' \subset \Gamma$ have been found; let these MDs form the set $\{\Gamma' \twoheadrightarrow \Delta_i \mid 1 \le i \le n\}$. Assume also that elementary MDs with the left side Γ form the set $\{\Gamma \twoheadrightarrow \Theta_j \mid 1 \le j \le m\}$. Then by Proposition 4.4 there must exist an i for which (1) $\Delta_i \supseteq \Lambda$ where $\Lambda = \Gamma - \Gamma'$, and (2) $\Delta_i - \Lambda \supseteq \Theta_j$, for every $1 \le j \le m$.

Moreover, if $\Gamma \twoheadrightarrow \Theta_j$ is multiple, then Θ_j is properly contained in $\Delta_i - \Lambda$. Therefore, all multiple elementary MDs with the left side $\Gamma \supset \Gamma'$ can be generated as follows.

(1) Select the elementary MD, say $\Gamma' \twoheadrightarrow \Delta$, with the left side Γ' and the right side Δ containing $\Lambda = \Gamma - \Gamma'$.
(2) Move the Λ attributes from the right side of this MD into its left side. (Thus set $\Gamma = \Gamma' \cup \Lambda$ and $\Delta = \Delta - \Lambda$.)
(3) Partition Δ into subsets $\Delta_1, \ldots, \Delta_p$ such that for each $1 \le k \le p$, $\Gamma \twoheadrightarrow \Delta_k$ and for no $\Delta'_k \subset \Delta_k$, $\Gamma \twoheadrightarrow \Delta'_k$. Determine which of these MDs is elementary (i.e., for no $\Gamma'' \subset \Gamma$, $\Gamma'' \twoheadrightarrow \Delta_k$).

Therefore, if there does not exist any elementary $\Gamma' \twoheadrightarrow \Delta$ with $|\Delta| > 2$, then no multiple elementary MD with the left side $\Gamma \supset \Gamma'$ can exist either.

Consider our previous example FIELD (4.1), where D1, D2, D3 are the elementary dependencies with the left side {CUST#}. Since the right sides of D1, D2, and D3 contain fewer than three attributes, no multiple elementary MD can exist having as the left side a superset of {CUST#}. Dependency D6 instead has more than two attributes at the right side. Thus MDs constructed by moving some of these attributes into the left side and partitioning the others could be multiple and elementary. From the statement of intension of FIELD (4.1), however, we conclude that none of the MDs so constructed holds.

The set of elementary FDs and multiple elementary MDs for FIELD are given in (4.2). Notice that whenever an elementary FD such as "CUST# $\rightarrow$ CUSTN" appears, we do not list explicitly its MD counterpart, "CUST# $\twoheadrightarrow$ CUSTN," although the latter is a multiple elementary MD and belongs to the set as attested by Proposition 4.1 and by the presence in (4.2) of D1 and D2 having the left side CUST#. This notational convention is used for conciseness in the remainder of this paper.

5. DECOMPOSITION OF A RELATION

A general decomposition algorithm for designing relational schemata for databases is presented in this section. The generality of the algorithm is such that it can be used for designing both normal form relations and other types of schemata, such as Chen's E-R diagrams [8]. Section 6 of this paper discusses the application of the algorithm to the design of third normal form schemata. A companion paper will discuss its application to the design of graphical schemata, including E-R diagrams.

The decomposition algorithm implements three new concepts: (1) complete relatability, (2) admissibility of covers, and (3) validation of results. This section introduces the algorithm and discusses points (1) and (2). Point (3) is discussed in Section 6.

5.1 Dependencies in Projections

The decomposition algorithm is recursive in nature: Once the original relation, say $R(\Omega)$, is decomposed into a pair of projections on smaller attribute sets, these must be decomposed in turn. At each step the next decomposition is chosen on the basis of the elementary FDs and of the multiple elementary MDs in the relation. We have discussed above the problem of determining these dependencies in the original relation(s). We now consider the problem of determining the same dependencies in the projections successively generated by the decomposition algorithm. For FDs, which have both the projectability and reverse projectability property, the rule is simple: The set of elementary FDs of $\Pi R(\Omega')$, $\Omega' \subseteq \Omega$, simply consists of all the elementary FDs of $R(\Omega)$ which have both the left and right sides contained in Ω'. For MDs, the rule is more complex since MDs other than those inferable by projectability might appear in a projection. Any MD of $\Pi R(\Omega')$ which is not derivable by projectability from the MDs of $R(\Omega)$ is said to be *latent* in $R(\Omega)$. Latent MDs, which represent an interesting, but less common type of semantic constraint, are discussed in Section 5.2.

The algorithm used for constructing the set of multiple elementary MDs in a projection is based on the following proposition.

PROPOSITION 5.1. *Let G' denote the set of MDs of $\Pi R(\Omega')$ derived by projectability from the set of multiple elementary MDs of $R(\Omega)$, $\Omega' \subseteq \Omega$. If $R(\Omega)$ contains no latent MD, G' has the following properties*:

(1) *G' contains all the multiple elementary MDs of $R(\Omega')$.*
(2) *Every member of G' is an elementary MD for $\Pi R(\Omega')$.*

PROOF. To prove (1), let $\Gamma \twoheadrightarrow \Delta$ be a multiple elementary MD of $\Pi R(\Omega')$. Since $R(\Omega)$ has no latent MD, it must contain one or more MD from which $\Gamma \twoheadrightarrow \Delta$ can be constructed by projectability. Out of these let us select one, say $\Gamma \twoheadrightarrow \Lambda$, which has a minimal right side (i.e., $\Lambda \cap \Omega' = \Delta$ and $R(\Omega)$ contains no $\Gamma \twoheadrightarrow \Lambda'$ such that $\Lambda' \subset \Lambda$ and $\Lambda' \cap \Omega' = \Delta$). We want to prove that $\Gamma \twoheadrightarrow \Lambda$ is elementary. By contradiction assume that $\Gamma' \twoheadrightarrow \Lambda'$ exists in $R(\Omega)$ with $(\Gamma', \Lambda') < (\Gamma, \Lambda)$. This last MD projects into $\Pi R(\Omega')$ as $\Gamma' \twoheadrightarrow (\Lambda' \cap \Omega')$. Now $(\Gamma', \Lambda' \cap \Omega') \leq (\Gamma, \Delta)$ by the previous definition. But since $\Gamma \twoheadrightarrow \Delta$ is elementary, then $\Gamma' = \Gamma$ and $\Lambda' \cap \Omega' = \Delta$. This contradicts the assumption that $\Gamma \twoheadrightarrow \Lambda$ has a minimal right side. Thus we find that every elementary MD in $\Pi R(\Omega')$ can be obtained by projectability from some elementary MD in $R(\Omega)$. Thus if the former is also multiple, so is the latter.

To prove (2), let $\Gamma \twoheadrightarrow \Delta$ be the projection of an elementary MD of $R(\Omega)$, say $\Gamma \twoheadrightarrow \Lambda$ ($\Delta = \Lambda \cap \Omega'$). We want to prove that $\Pi R(\Omega')$ contains no elementary $\Gamma' \twoheadrightarrow \Delta'$ with $(\Gamma', \Delta') < (\Gamma, \Delta)$. We proved above that each elementary MD of $\Pi R(\Omega')$ can be derived by projectability from an elementary MD of $R(\Omega)$. Thus let $\Gamma' \rightarrow \Lambda'$ be an elementary MD in $R(\Omega)$, such that $\Delta' = \Lambda' \cap \Omega'$. Since $\Gamma \twoheadrightarrow \Lambda$ and $\Gamma' \twoheadrightarrow \Lambda'$ are both elementary and their right sides are not disjoint, it follows from Proposition 4.4 that $\Lambda' \supseteq \Lambda \cup (\Gamma - \Gamma')$. Thus $\Delta' = (\Lambda' \cap \Omega') \supseteq \Delta \cup (\Gamma - \Gamma')$, which contradicts the assumption $(\Gamma', \Delta') < (\Gamma, \Delta)$. Q.E.D.

In the absence of latent MDs, the following algorithm can be used to derive the set of multiple elementary MDs of $\Pi R(\Omega')$ (G' is a set variable, and $\bar{G}_m$ is the set of multiple elementary MDs of $R(\Omega)$):

S1. Set G' to empty.
S2. For each $(\Gamma \twoheadrightarrow \Delta) \in \bar{G}_m$ where $\Gamma \subset \Omega'$ and $\{\Delta \cap \Omega'\} \neq \varnothing$ add $\Gamma \twoheadrightarrow (\Delta \cap \Omega')$ to G'.
S3. Eliminate all the single MDs from G'.

At the end of S2, G' coincides with the set of MDs of $\Pi R(\Omega')$ derivable by projectability from $\bar{G}_m$; thus it has the two properties enunciated in Proposition 5.1. Therefore, after the final elimination of single MDs at step S3, G' coincides with the multiple elementary MDs of $\Pi R(\Omega')$.

As an example of decomposition according to multiple elementary MDs, consider our example FIELD (4.1) whose dependencies are defined by (4.2) (no latent MD occurs here). Decomposing according to D1, we obtain

$$\Pi\text{FIELD(CUST\#, CUSTN)}, \tag{5.1}$$

$$\Pi\text{FIELD(CUST\#, MODEL, MODQ, TEC\#, TECN)}. \tag{5.2}$$

By the previous rules, we find that the set $\bar{F}$ for (5.1) consists only of D1. The set $\bar{G}_m$ is instead empty; thus (5.1) is atomic. For (5.2) we find $\bar{F}$ = {D4, D5} and $\bar{G}_m = G'$ = {D2, D3, D5, D6'}, where D6' is TEC# $\twoheadrightarrow$ {CUST#, MODEL, MODQ}. Applying D2, (5.2) is next decomposed into

$$\Pi\text{FIELD(CUST\#, MODEL, MODQ)}, \tag{5.3}$$

$$\Pi\text{FIELD(CUST\#, TEC\#, TECN)}. \tag{5.4}$$

Relation (5.3) is atomic. For (5.4) we have $\bar{G}_m$ = {D5, D6″} where D6″ is "TEC# $\twoheadrightarrow$ CUST#." Note that D3 was eliminated since it is a single elementary MD for (5.4). Thus out of the three elementary MDs of the original relation with the left side {CUST#}, only two are effectively used in the decomposition. In general, out of n elementary MDs with the same left side, $n - 1$ at most will be actually used in the decomposition. The last MD always projects into a single MD and is thus eliminated.

In conclusion, decomposing by multiple MDs is equivalent to decomposing by the MDs of a minimal cover obtained according to reflexivity, augmentation, additivity, and complementation (see Proposition 4.6).

The set of relations generated by the previous decomposition is therefore

$$\begin{aligned}&\Pi \text{FIELD}(\text{CUST\#}, \text{CUSTN}),\\ &\Pi \text{FIELD}(\text{CUST\#}, \text{MODEL}, \text{MODQ}),\\ &\Pi \text{FIELD}(\text{TEC\#}, \text{TECN}),\\ &\Pi \text{FIELD}(\text{CUST\#}, \text{TEC\#}).\end{aligned} \tag{5.5}$$

It should be noted that redundancy problems will occur when MDs other than multiple elementary ones are used in the decomposition. Assume, for example, that the single MD "{CUST#, MODEL} $\rightarrow\!\!\rightarrow$ MODQ," is used in the decomposition of FIELD; whether this MD is used to decompose (4.1) directly or to decompose its projection (5.2), the subrelation ΠFIELD(CUST#, MODEL) is obtained in addition to the four subrelations in (5.5). But since this new subrelation is a projection of ΠFIELD(CUST#, MODEL, MODQ), already contained in (5.5), it adds no new information.

5.2 Latent MDs

Every MD of $\Pi R(\Omega')$, $\Omega' \subset \Omega$ which cannot be obtained by projectability from some MDs of $R(\Omega)$ is said to be *latent* in $R(\Omega)$. This concept is related to, but much more restrictive than, that of the *embedded* MD proposed in [16].[9] Since latent MDs occur in practice and possess a well-defined semantic interpretation, they should be considered by the database designer along with the other MDs. Let us first clarify their nature by an example.

Assume the existence of a relation characterized by the following attributes:

(1) Students identified by their student number (ST#).
(2) College courses (COURSE).
(3) Teaching assistants (TA).
(4) Number of hours that a TA has spent with a student (HRS).

Moreover, assume the following constraints:

(a) A course is attended by many students, and one student may attend many courses.
(b) More than one TA may be assisting in a course, and the same TA may be helping in more than one course.
(c) A student may turn for help to any TA assigned to the course.
(d) The up-to-date total number of hours that a TA has spent with a student is inclusive of any course at which the TA meets the student and not subdivided by course.

[9] Every $\Gamma \rightarrow\!\!\rightarrow \Delta$ which holds for some projection $\Pi R(\Omega')$ is said to be an embedded MD for $R(\Omega)$. Thus embedded MDs include both latent MDs and MDs derived by projectability.

A small sample of the content of such a relation is given as follows:

HELP(COURSE,	ST#,	TA,	HRS)
MATH 10	5512	SMITH	0
MATH 10	9732	SMITH	12
MATH 10	4008	SMITH	15
MATH 10	5512	MARIN	0
MATH 10	9732	MARIN	0
MATH 10	4008	MARIN	12
PHYS 20	4008	MARIN	12
PHYS 20	7532	MARIN	6

(5.6)

Notice, for instance, that MARIN has tutored 4008 in MATH 10 and/or in PHYS 20 for a total of 12 hours. Thus in (5.6) we have

$$\{\mathrm{ST\#}, \mathrm{TA}\} \rightarrow \mathrm{HRS}$$

$$\{\mathrm{ST\#}, \mathrm{TA}\} \twoheadrightarrow \mathrm{COURSE}.$$

No other elementary FD nor multiple elementary MD exists. Decomposing relation (5.6) according to the above dependencies, we find

$$\Pi\mathrm{HELP}(\mathrm{ST\#}, \mathrm{TA}, \mathrm{HOURS}) \tag{5.7}$$

$$\Pi\mathrm{HELP}(\mathrm{COURSE}, \mathrm{ST\#}, \mathrm{TA}). \tag{5.8}$$

Now in (5.8) we witness the appearance of the pair of MDs

$$\mathrm{COURSE} \twoheadrightarrow \mathrm{ST\#}$$

$$\mathrm{COURSE} \twoheadrightarrow \mathrm{TA},$$

which were latent in (5.6).

Thus (5.8) can further be decomposed into the pair

$$\Pi\mathrm{HELP}(\mathrm{COURSE}, \mathrm{ST\#})$$

$$\Pi\mathrm{HELP}(\mathrm{COURSE}, \mathrm{TA}).$$

The designer can either list the latent MDs of the initial relation or wait until they become explicit in the course of the decomposition. The second approach would require the designer's intervention to reassess the dependencies of the projections after each decomposition step. Thus the first approach appears preferable, since it requires the designer's intervention only once. After both the usual and the latent multiple elementary MDs have been listed along with the elementary FDs, it is easy to determine $\bar{G}_m$ and $\bar{F}$ for any projection. This method is a reasonable one since latent MDs are the expression of very particular semantic constraints which could hardly escape the attention of a careful designer. The relation HELP, for instance, is characterized by the obvious constraint that HRS must be specified for each TA and student who, respectively, teach and take the same course.

5.3 Decomposing for Complete Relatability

The decomposition algorithm proposed in this paper has three (main) objectives:

(1) decomposing complex relations into simple well-defined primitives,
(2) preserving information,
(3) minimizing redundancy.

The decomposition algorithm is recursive in nature. At each step a relation is decomposed into a pair of subprojections according to a multiple elementary MD. Then these projections are decomposed in turn, and the process continues until a set of atomic relations is obtained. (An atomic relation is one which only contains trivial MDs—Section 2.2.) Proposition 3.1 and the associative and commutative property of natural joins ensure that the original relation is reconstructible as the natural join of this final set of atomic projections. Relation (4.1), for instance, is reconstructible as the natural join of the projections indicated in (5.5). Thus every decomposition according to MDs is content-preserving. Complete data relatability, however, demands that the "structure" of the relation be preserved along with its content. The structure considered in this paper includes three components: (1) the composition of the various relations (i.e., their attribute sets); (2) the functional dependencies; (3) the multivalued dependencies. Now structural information is preserved when the dependencies of the original relations are inferable from the composition of the relations in the resulting decomposition and from the dependencies in these relations. Thus a reversibility test must be used to verify that no loss of structural information has occurred. (The formal rules for performing this test are discussed in the next section.) The algorithm selects among the multiple elementary MDs at hand, those which ensure the preservation of structural information and uses only these in the decomposition.

The subprojections obtained at the end of a decomposition are atomic relations (i.e., they contain only trivial MDs). Thus MDs do not play any visible role in the final decomposition; they exhaust their function in the generation of the (proper) decomposition. Only the following two elements are visible in the final schema: (1) the attribute sets of the resulting atomic subrelations, and (2) a set of elementary FDs. Following [27], we call (1) and (2) A-structure and Z-structure, respectively. The importance of these two structural components and their mutual relationships have been discussed in [27]. In a companion paper we will show how their combined representation in the form of special diagrams finds application as conceptual database schemata [28]. In this paper we concentrate instead on the applications of our decomposition algorithm to improve on the state-of-the art approaches to the design of 3NF schemata. These applications utilize two novel concepts: (1) the notion of *scope of elementary* FDs, and (2) the notion of *admissibility of FD covers*. The scope of an elementary FD, $\Gamma \rightarrow A$, is the set $\Gamma \cup \{A\}$, that is, the set of attributes appearing at either side of this elementary FD.

To define the concept of admissibility, let A and Z denote, respectively, a set of atomic projections and a set of elementary FDs for a relation $R(\Omega)$. The set Z is said to be *admissible* with respect to A when the following two conditions are satisfied:

(1) If Z contains an elementary FD with scope Δ, it must contain every other elementary FD of $R(\Omega)$ having scope Δ. Moreover, if $\Pi R(\Delta)$ for such a Δ is atomic, then A must contain it as a member.
(2) If A contains an atomic projection $\Pi R(\Delta)$, then Z must contain every elementary FD of $R(\Omega)$ having scope Δ.

The final product of the decomposition algorithm discussed in this section is a set of atomic relations called *ACOVER* and a set of elementary FDs called *ZCOVER*. The algorithm is designed so that *ZCOVER* is admissible with respect to *ACOVER*. The advantages of this approach are discussed in detail in Section 6. Basically, however, the admissibility condition guarantees uniformity of treatment over the A- and the Z-structure to capture the natural relationships between attributes. For instance, we like to regard a one-to-one correspondence, say $B \leftrightarrow C$, as a semantically elementary relationship although syntactically it is modeled as a composite object (i.e., by the pair of elementary FDs $B \to C$ and $C \to B$). Thus condition (1) guarantees that $B \to C$ and $C \to B$ are treated uniformly: If either of these FDs appears in *ZCOVER*, so does the other; and the atomic subprojection $\Pi R(B, C)$ is included in *ACOVER*. Conversely, condition (2) guarantees that once an atomic subprojection is included in *ACOVER*, all its elementary FDs are included in *ZCOVER* (every elementary FD in an atomic relation has as scope the complete attribute set of the relation). In our example FIELD, for instance, inclusion of the atomic subprojection ΠFIELD(CUST#, MODEL, MODQ) in *ACOVER* implies that {CUST#, MODEL} → MODQ must be included in *ZCOVER*.

The previous example shows the simple primitives in terms of which the original relations are reduced: Our basic primitives are atomic subprojections with all the elementary FDs contained therein. A subprojection with n attributes contains at the most n such FDs. In addition to elementary FDs having as scope atomic subprojections, the admissibility condition allows us to include in *ZCOVER* elementary FDs with scope, say Δ, where $\Pi R(\Delta)$ is decomposable. These FDs will, in fact, be included as needed to ensure that the complete relatability condition is satisfied.

A final objective of Algorithm 5.1 is minimizing redundancy both in terms of atomic subprojections and elementary FDs resulting from the decomposition. For this purpose, only multiple elementary MDs are used by Algorithm 5.1.

This policy is very helpful in minimizing redundancy in the A-structure. The main support for this statement comes from empirical evidence. For instance, in Section 5.1, we have seen that violation of this policy produces a redundant set of atomic subrelations (i.e., one where a relation is derivable from another by projection). A number of other examples can be produced to show that this is a general behavior not restricted to the particular example FIELD. A formal proof of the minimum redundancy property is instead available for the Z-structure: In Section 5.5 it is proved that the set *ZCOVER* produced by Algorithm 5.1 supplies an absolute minimum cover for the FDs of the relation.

5.4 The Algorithm

Let us consider next the problem of decomposing a relation according to the complete relatability criterion. We solve this problem by selecting among the possible decompositions at hand one which ensures preservation of both the FD and the MD structure of the given relation. In particular, to preserve the FD structure, we construct, through the decomposition, set *ZCOVER* which constitutes a (minimal) cover for the FDs of the relation.

Say that we consider the decomposition of $R(\Omega)$ into $R(\Omega_1)$ and $R(\Omega_2)$, where

$$R(\Omega) = R(\Omega_1) \cdot R(\Omega_2), \tag{5.9}$$

Ω_1 and Ω_2 denote proper subsets of Ω, $\bar{F}$ denotes the elementary FDs of $R(\Omega)$, and F_1 and F_2 denote the elementary FDs of $\Pi R(\Omega_1)$ and $\Pi R(\Omega_2)$.

The problem of preserving the FD structure of R is solved in two phases. First we deal with the elementary FDs having as scope the whole attribute set, Ω. As we prove later (Proposition 5.3), these FDs cannot be implied by F_1 and F_2, no matter how Ω_1 and Ω_2 are chosen. Thus we preserve them by explicitly entering them in *ZCOVER*. Then we subtract all the elementary FDs with scope Ω from $\bar{F}$ and we address the problem of preserving the remaining FDs. This is done by selecting a decomposition where the following condition is satisfied:

$$\text{CRC1}: \bar{F} \subseteq (F_1 \cup F_2)^+. \tag{5.10}$$

As discussed in the previous section, to avoid redundancy, we decompose according to multiple elementary MDs. Thus the algorithm selects among these a $\Gamma \twoheadrightarrow \Delta$, such that if $\Omega_1 = \Gamma \cup \Delta$ and $\Omega_2 = \Omega - \Gamma$, then both CRC1 and a similar complete relatability condition for MDs are satisifed. This complete relatability condition for MDs is discussed next.

The reverse projectability property does not hold for MDs since a $\Gamma \twoheadrightarrow \Delta$ in $\Pi R(\Omega_1)$ does not imply $\Gamma \twoheadrightarrow \Delta$ in $R(\Omega)$. However, the following weaker property holds.[10]

PROPOSITION 5.2. *Joinability*: *If*

(1) $R(\Omega \cup \Psi) = S(\Omega) \cdot P(\Psi)$,
(2) $\Gamma \twoheadrightarrow \Delta$ *in* $S(\Omega)$,
(3) $\Delta \cap \Psi = \emptyset$,

then $\Gamma \twoheadrightarrow \Delta$ *in* $R(\Omega \cup \Psi)$.

PROOF. Let us define first $\Lambda = \Omega - (\Gamma \cup \Delta)$. According to (2) we have

$$R(\Omega \cup \Psi) = (\Pi S(\Gamma \cup \Delta) \cdot \Pi S(\Gamma \cup \Lambda)) \cdot P(\Psi).$$

By the properties of the join we obtain

$$R(\Omega \cup \Psi) = \Pi S(\Gamma \cup \Delta) \cdot [\Pi S(\Gamma \cup \Lambda) \cdot P(\Psi)].$$

[10] A related result defining the relationship between MDs and embedded MDs was presented in [25]. An issue yet unresolved is whether there exist inference rules (from the projections to the join) stronger than these. However, see [25] on this topic.

By computing the join in the brackets immediately above, we see that R is the natural join of the relations with respective attribute sets $(\Gamma \cup \Delta)$ and $X = (\Gamma \cup \Lambda \cup \Psi)$. But by assumption (3), $(\Gamma \cup \Delta) \cap X = \Gamma$. Thus $\Gamma \twoheadrightarrow \Delta$ in R. (Note that the join in the brackets need not be a projection of R for this to be true.) Q.E.D.

Now let G_1 and G_2 denote the elementary MDs of $\Pi R(\Omega_1)$ and $\Pi R(\Omega_2)$. Moreover, let G_{11} denote the set of MDs in G_1 which have a right side disjoint from Ω_2, and G_{22} denote the set of those MDs in G_2 which have a right side disjoint from Ω_1. Then it is easy to see that the set of MDs of $R(\Omega)$ which we can infer by joinability from $G_1 \cup G_2$ is $G_{11} \cup G_{22}$. Moreover, we can use the information previously found about the preservation of certain FDs. Assuming that condition (5.10) is satisfied, we can add to $G_{11} \cup G_{22}$ the MDs obtained from $\bar{F}$ using MX1 and denoted by $\tilde{G}_F$. Then we can compute the MD closure of this combined set and check whether every element of $\tilde{G}_m$ is contained in the closure. This condition can be expressed as

$$\text{CRC2: } \tilde{G}_m \subseteq (\tilde{G}_F \cup G_{11} \cup G_{22})^+. \tag{5.11}$$

The conjunction of condition CRC1 and CRC2 supplies our complete relatability condition, which we denote by CRC. Note that CRC2 ensures that the MDs of $R(\Omega)$ can be derived from the combined FDs and MDs of $\Pi R(\Omega_1)$ and $\Pi R(\Omega_2)$. By contrast, CRC1 ensures that the FDs of $R(\Omega)$ are derivable from the FDs of $\Pi R(\Omega_1)$ and $\Pi R(\Omega_2)$ alone (i.e., without using the G_{11}, G_{22}, and rule MX2). This policy was chosen to ensure that the set *ZCOVER* generated by Algorithm 5.1 can be proved to be a minimal cover for the FDs of the relation (a property used in the design of 3NF schemata discussed in Section 6). This policy also reflects the notion that FDs are "stronger" constraints than MDs and that they should stand alone independently of MDs (which do not even appear explicitly in the final decomposition). Moreover, it appears doubtful that the utilization of G_{11}, G_{22}, and MX2 would yield any further FD not already included in $(F_1 \cup F_2)^+$. We make this conjecture on the basis of the structure of the MDs of G_{11} and G_{22}, although we cannot offer a formal proof at this time.

The decomposition Algorithm 5.1 constructs the two sets *ZCOVER* and *ACOVER* for a given relation $R(\Psi)$. Elements of *ZCOVER* have the form $(\ell{:}\Gamma \to A)$ where $\Gamma \to A$ is an elementary FD and ℓ is a label. The elements of *ACOVER* have the form $(\ell'{:}\ \Delta)$ where $\Pi R(\Delta)$ is an atomic subprojection of $R(\Psi)$ and ℓ' is a label. If $\Delta = \Gamma \cup \{A\}$ then $\ell' = \ell$.

Algorithm 5.1: Decomposition of a Relation $R(\Psi)$

A1. Determine (a) $\bar{F}_0$, the elementary FDs of R;
(b) $\tilde{G}_0$, the multiple elementary MDs of R;
(c) $\tilde{G}_L$, the multiple elementary MDs latent in R.

A2. Initialize the set variables *ZCOVER*, *ACOVER* to the empty set and the integer variable L to 1.

A3. DECOMPOSE (Ψ) (Invoke the procedure of Figure 1).

A4. Print *ZCOVER* and *ACOVER*.

The recursive procedure DECOMPOSE (Ω) has been formalized in Figure 1 using Algol-like syntactic constructs. The various declarations were omitted for

```
procedure DECOMPOSE (Ω) comment a recursive procedure to decompose ΠR(Ω);
begin STEP1: DETERMINE (F̄, Ḡm);
      STEP2: FLAG ← false;
             for each Γ → A ∈ F̄ do
                 if Γ ∪ {A} = Ω then begin FLAG ← true;
                                           ZCOVER ← ZCOVER ∪ {L:Γ → A};
                                           F̄ ← F̄ − {Γ → A};
                                     end STEP2;
   if Ḡm = ∅ then STEP3: begin ACOVER ← ACOVER ∪ {L:Ω}; L ← L + 1; end STEP 3
             else STEP4: begin NOTFOUND ← true;
                               for each Γ →→ Δ ∈ Ḡm while NOTFOUND do
                               begin Ω1 ← Γ ∪ Δ; Ω2 ← Ω − Δ;
                               COMPUTE (F1, F2, ḠF, G11, G22);
                 if (F̄ ⊆ (F1 ∪ F2)+ ∧ Ḡm ⊆ (ḠF ∪ G11 ∪ G22)+) then
                                  begin if FLAG then L ← L + 1;
                                        DECOMPOSE (Ω1);
                                        DECOMPOSE (Ω2);
                                        NOTFOUND ← false;
                                  end
                               end;
                               if NOTFOUND then REPORTFAILURE
                         end STEP4
end DECOMPOSE;
```

Fig. 1. An Algol-like description for the recursive procedure DECOMPOSE. (Declarations are omitted. All variables but *ZCOVER*, *ACOVER*, *L*, $\bar{F}_0$, $\bar{G}_0$, $\bar{G}_L$, and Ω are local to this procedure.) Note that the complete relatability conditions,

$$\bar{F} \subseteq (F_1 \cup F_2)^+ \wedge \bar{G}_m \subseteq (\bar{G}_F \cup G_{11} \cup G_{22})^+,$$

represent the core of the procedure.

brevity. It is understood that all the variables except *ACOVER*, *ZCOVER*, L, $\bar{F}_0$, $\bar{G}_0$, $\bar{G}_L$, and Ω are local to DECOMPOSE. The procedure consists of four steps. At STEP1 the elementary FDs and the multiple elementary MDs of $\Pi R(\Omega)$, $\Omega \subseteq \Psi$, are determined according to the rules presented in previous sections. (As we recall, once the designer has specified the latent MDs along with the multiple elementary MDs and the elementary FDs of the original $R(\Psi)$, then STEP1 can be performed without his further intervention.) At STEP2 all the FDs in $\bar{F}$ with scope Ω are removed from $\bar{F}$ and added to *ZCOVER* using the current value of L as label. Next, if $\Pi R(\Omega)$ is atomic (i.e., $\bar{G}_m = \emptyset$), then STEP3 is executed; otherwise STEP4 is executed. At STEP3, Ω is entered in the *ACOVER* with the current value of L as label. At STEP4 instead, $\bar{G}_m$ is searched for a $\Gamma \twoheadrightarrow \Delta$ which ensures complete relatability. Thus after setting Ω_1 and Ω_2, respectively, equal to $\Gamma \cup \Delta$ and $\Omega - \Delta$, the sets F_1, F_2, $\bar{G}_F$, G_{11}, and G_{22} are constructed by a procedure called here COMPUTE. Finally, the validity of CRC1 and CRC2 are tested; if they are verified, then the procedure is recursively invoked to decompose $\Pi R(\Omega_1)$ and $\Pi R(\Omega_2)$. If not, such $\Gamma \twoheadrightarrow \Delta$ is found, failure is reported, and execution halts.

The value of the global integer variable L is incremented after a new atomic component is formed (STEP3). It is also incremented when $\Pi R(\Omega)$ is found decomposable after some elementary FD with scope Ω has been detected. (This is done at STEP4 on the basis of the information passed down from STEP2 by means of FLAG.) This discipline guarantees that all the FDs of identical scope, say Δ, are entered in *ZCOVER* under the same label, say ℓ. Moreover, if $\Pi R(\Omega)$ is next found to be atomic, then Δ is entered in *ACOVER* under the same label ℓ. If $\Pi R(\Omega)$ is instead found decomposable, then its atomic components will receive label values $\ell + 1$, $\ell + 2, \ldots$.

Let us consider now the problem of determining G_{11} and G_{22}. By definition G_{11} is the set of MDs in G_1 which have a right side disjoint from Ω_2. Now if we assume that G_1 is the set of multiple MDs of $\Pi R(\Omega_1)$, then it follows directly from the definition that each MD in G_1 having its right side disjoint from Ω_2 must also belong to $\bar{G}_m$. Thus it is possible to evaluate G_{11} and G_{22} without having to construct the actual dependencies in $\Pi R(\Omega_1)$ and $\Pi R(\Omega_2)$ explicitly. G_{11} can be obtained by taking those dependencies in $\bar{G}_m$ which have their left side contained in Ω_1 and their right side contained in $\Omega_1 - \Omega_2$. Symmetrically, G_{22} consists of those dependencies in $\bar{G}_m$ which have their left side contained in Ω_2 and their right side contained in $\Omega_2 - \Omega_1$.

A linear time algorithm to decide whether one FD belongs to the closure of a set of FDs, F, was given in [5]. This algorithm is linear in the length of the representation of F. Efficient algorithms to decide whether a given MD belongs to the closure of a set of MDs have been proposed in [3, 17, 19, 23]. In particular, the time complexity of the algorithm presented in [17] is $n \log n$. Thus Algorithm 5.1 has an obvious upper bound of $O(n^4 \log n)$, where n denotes the cardinality of $\bar{G}_0 \cup \bar{G}_L$. This polynomial bound, however, only applies after $\bar{F}_0$, $\bar{G}_0$, and $\bar{G}_L$ have been computed, since the size of these sets may actually be exponential. Fortunately this seems not to be the case in realistic situations. Also, refinements to Algorithm 5.1 have been proposed [24] to improve its performance and to eliminate the need for listing all the elementary FDs and multiple elementary MDs of R. A discussion of these refinements is outside the scope of this paper.

5.5 Examples

We shall now apply Algorithm 5.1 to the previous example FIELD. At step A1 the designer enters the elementary FDs and the multiple elementary MDs of set (4.2) (no latent MD exists here). After the initialization at A2 the DECOMPOSE routine is entered. Here execution of STEP1 simply returns the dependency set (4.2) and STEP2 is of no consequence. Then STEP4 is executed where the multiple MD counterpart of D1, that is, $\overline{\text{D1}}$: CUST# $\twoheadrightarrow$ CUSTN, is tested first.[11] For brevity say that F_p denotes $F_1 \cup F_2$ and G_p denotes $\bar{G}_F \cup G_{11} \cup G_{22}$. Thus for $\overline{\text{D1}}$ we have

$$F_p = \{\text{D1, D4, D5}\},$$
$$G_p = \{\overline{\text{D1}}, \overline{\text{D4}}, \overline{\text{D5}}, \text{D2, D3}\}.$$

The only dependency missing is D6, which is directly derivable from $\overline{\text{D5}}$ when we form G_p^+. Thus the complete relatability conditions are satisfied and the algorithm next attempts to decompose ΠFIELD(Ω_1) and the ΠFIELD(Ω_2) where Ω_1 =

[11] Whenever we have an FD denoted Dn, we denote its MD counterpart (obtained from it by rule MX1) by $\overline{\text{Dn}}$. For the sake of brevity the MD counterparts are usually omitted in the original lists of dependencies.

{CUST#, CUSTN} and Ω_2 = {CUST#, MODEL, MODQ, TEC#, TECN}. $\Pi R(\Omega_1)$ contains D1 with scope Ω_1; thus "1:CUST# → CUSTN" is entered in *ZCOVER*. Moreover, ΠFIELD(Ω_1) contains no multiple MD; thus STEP3 is executed, "1: {CUST#, CUSTN}" is also entered in *ACOVER*, and the decomposition of $\Pi R(\Omega_1)$ is complete. As ΠFIELD(Ω_2) is considered next for decomposition and DETERMINE ($\bar{F}$, $\bar{G}_m$) is executed, the following dependencies are found:

$$\bar{F} = \{\mathrm{D4, D5}\}, \qquad \bar{G}_m = \{\mathrm{D2, D3}, \overline{\mathrm{D5}}, \mathrm{D6'}\}$$

where D6′ is obtained from D6 by striking out CUSTN, as per the projectability rule. As D2:CUST# →→ {MODEL, MODQ} is tested in STEP4, we find that $\underline{F_p} = \bar{F}$. Also G_p contains all the MDs of $\bar{G}_m$ except D6′, which is obtained from $\overline{\mathrm{D5}}$ when G_p^+ is constructed. Thus the complete relatability condition is once again met, and the decomposition of ΠFIELD(CUST#, MODEL, MODQ) is invoked next. The overall decomposition sequence is therefore the one discussed in Section 5.1 leading to (5.5). At the end, *ZCOVER* and *ACOVER* are

ACOVER = {1: {CUST#, CUSTN}; 2: {CUST#, MODEL, MODQ};
3: {TEC#, TECN}; 4: {CUST#, TEC#}}
ZCOVER = {1: CUST# → CUSTN; 2: {CUST#, MODEL} → MODQ;
3: {TEC# → TECN}}.

Let us now consider a relation having multiple atomic decompositions. In this instance we have data concerning privately owned motor vehicles and their owners. This is the sort of information which the California Department of Motor Vehicles may want to maintain. The attributes of interest are

LIC: license numbers of motor vehicles;
MAKE: manufacturers of motor vehicles;
MODEL: models of vehicles;
YEAR: year in which the vehicle was manufactured;
VALUE: current value of the vehicle;
OWNER: unique identifier of a person (for simplicity we give only the family name, in reality, a composite of ID which includes the first name, birth date, and location, etc., may be needed);
DRVL: driving license numbers;
VIOL: code number for traffic violations;
DATE: month, day, and year of violation.

The following information is required:

(1) make, model, and year of any licensed vehicle;
(2) current (blue book) value of a given type of vehicle;
(3) legal owner of a given vehicle;
(4) driving license number of a given person (and the person having a certain license number);
(5) traffic violation history of any driver; the records consist of pairs: violation code and date of warrant.

Thus one may start with the sample content shown in Table II for the DMV

Table II. A Sample Situation for the Relation DMV

DMV (	LIC,	MAKE	MODEL	YEAR	VALUE	OWNER	DRVL	VIOL	DATE)
	2LL 235	FORD	CAPRI	1972	1550	OWEN	A771235	22520	4/5/75
								22351	1/9/75
	AFF 255	GM	NOVA	1975	2300	OWEN	A771235	22520	4/5/75
								22351	1/9/75
	QPR 937	FIAT	128	1973	1900	WEST	C991837	21803	9/5/74
								22100	2/12/75
	LFV 222	GM	VEGA	1976	2550	MANN	D331973	0000	0/00/00

relation. Please note that in Table II the various violations on each driver's record have been grouped together. Such a representation is more concise and expressive than a true first normal form representation. Notice, for instance, how the MDs, "DRVL $\twoheadrightarrow$ {VIOL, DATE}" and "OWNER $\twoheadrightarrow$ {VIOL, DATE}" are depicted in Table II. Anyone who is accustomed to identifying the patterns of transitive FDs in relations would soon recognize its generalized version as it appears in the table. In fact we have

$$\text{LIC} \rightarrow \text{OWNER}$$

and

$$\text{OWNER} \twoheadrightarrow \{\text{VIOL, DATE}\},\ \text{LIC} \twoheadrightarrow \{\text{VIOL, DATE}\}$$

according to the transitivity property of MDs. The set of elementary FDs and multiple elementary MDs characterizing DMV is therefore

D1: LIC → MAKE
D2: LIC → YEAR
D3: LIC → MODEL
D4: LIC → VALUE
D5: LIC → OWNER
D6: LIC → DRVL
D7: LIC $\twoheadrightarrow$ {VIOL, DATE}
D8: {MAKE, YEAR, MODEL} → VALUE
D9: {MAKE, YEAR, MODEL} $\twoheadrightarrow$ {LIC, OWNER, DRVL, VIOL, DATE}
D10: OWNER → DRVL
D11: OWNER $\twoheadrightarrow$ {VIOL, DATE}
D12: OWNER $\twoheadrightarrow$ {LIC, MAKE, YEAR, MODEL, VALUE}
D13: DRVL → OWNER
D14: DRVL $\twoheadrightarrow$ {VIOL, DATE},
D15: DRVL $\twoheadrightarrow$ {LIC, MAKE, YEAR, MODEL, VALUE}.

Assume that $\overline{\text{D1}}$: LIC $\twoheadrightarrow$ MAKE is tested first. There we find that D8 is missing from $F_p = F_1 \cup F_2$ and D9, D12, D15 are missing from $G_p = (G_F \cup G_{11} \cup G_{22})$. Thus both CRC1 and CRC2 fail since D8 $\notin F_p^+$ and D9 $\notin G_p^+$ (indeed it is always true that a nontrivial FD or MD with left side Γ can only be inferred from a set of dependencies if that set contains some other dependency having as the left side either Γ or a subset of it). The case of $\overline{\text{D2}}$ and $\overline{\text{D3}}$ is analogous[12] to

[12] This may be seen immediately from the fact that MAKE, YEAR, and MODEL behave precisely in the same way within the dependencies of DMV.

Table III. A Decomposition for Relation DMV

Label	*ACOVER*	*ZCOVER*
1	{VALUE, MAKE, MODEL, YEAR}	{MAKE, YEAR, MODEL} → VALUE
2	{LIC, MAKE}	LIC → MAKE
3	{LIC, YEAR}	LIC → YEAR
4	{LIC, MODEL}	LIC → MODEL
5	{OWNER, DRVL}	OWNER → DRVL
5	{OWNER, DRVL}	DRVL → OWNER
6	{OWNER, VIOL, DATE}	None
7	{LIC, OWNER}	LIC → OWNER

the case of $\overline{\mathrm{D1}}$: They do not pass the complete relatability test either. If $\overline{\mathrm{D4}}$ is used, D8 and D9 are lost again. If $\overline{\mathrm{D5}}$ is used, D10, D13 and D11, D12 are lost. Symmetrically, with $\overline{\mathrm{D6}}$ we lose D10, D13 and D14, D15. With D7 we lose D11, D12, D14, and D15. In each of the previous cases these dependencies cannot be reconstructed by closure. Using $\overline{\mathrm{D8}}$, however, we find that only D12 and D15 are missing from G_p; however, they are derivable by complementation and partitionality from the remaining ones. After DMV is decomposed according to $\overline{\mathrm{D8}}$, we find that DECOMPOSE(Ω) applied to ΠDMV(MAKE, YEAR, MODEL, VALUE) generates the elementary FD, "1: {MAKE, YEAR, MODEL} → VALUE" and the atomic component "1: {MAKE, YEAR, MODEL, VALUE}."

Thus the decomposition of ΠDMV(LIC, MAKE, MODEL, YEAR, OWNER, DRV, VIOL, DATE) is attempted next. $\bar{F}$ and $\bar{G}_m$ for this subrelation consists of D1, D2, D3, D5, D6, D7, D10, D11, D12′, D13, D14, and D15′, where D12′ and D15′ denote the MDs constructed from D12 and D15 by projecting out VALUE. While $\overline{\mathrm{D1}}$ could not be used to decompose the original DMV, it can be used now in this projection without violating complete relatability. Indeed the only dependencies missing are D12′ and D15′, which are inferable by complementation from the remaining ones. Thus $\overline{\mathrm{D1}}$ is used and so are $\overline{\mathrm{D2}}$ and $\overline{\mathrm{D3}}$ at the following steps. We shall skip these steps and resume our decomposition algorithm after the atomic component "4: {LIC, MODEL}" has been generated. The subrelation at hand is then

ΠDMV(LIC, OWNER, DRVL, VIOL, DATE)

characterized by dependencies D5, D6, D7, D10, D11, D12″, D13, D14, D15″ where D12″ is "OWNER →→ LIC" and D15″ is "DRVL →→ LIC." If "$\overline{\mathrm{D5}}$: LIC →→ OWNER" is used, then D10, D13 and D11, D12″ are lost. If $\overline{\mathrm{D6}}$ is used, D10, D13 and D14, D15″ are lost. If D7 is used, we lose D11, D12″, D14, and D15″. Thus we find that $\overline{\mathrm{D10}}$ yields the first acceptable decomposition in the sequence. The procedure DECOMPOSE (Ω), where Ω = {OWNER, DRVL}, first finds the two elementary FDs with scope Ω (i.e., OWNER → DRVL and DRVL → OWNER) and enters them in *ZCOVER* under the same label "5," next it adds "5: {OWNER, DRVL}" to *ACOVER*.

As the decomposition of ΠDMV(LIC, OWNER, VIOL, DATE) with dependencies D5, D7, D11, D12″ is attempted next, $\overline{\mathrm{D5}}$ and D7 again fail the complete relatability test, while D11 passes it. Thus the decomposition ends with the two atomic components "6: {OWNER, VIOL, DATE}" and "7: {LIC, OWNER}." Table III summarizes the complete decomposition thus obtained.

A relation sometimes possesses more than one *ACOVER*, *ZCOVER* pair. This typically occurs in the presence of one-to-one correspondence between attributes. In relation DMV, for instance, DRVL and OWNER play an interchangeable role. Thus the attributes VIOL, DATE on the one hand and the attributes MAKE, MODEL, YEAR, VALUE on the other hand can be associated with either one of them. Considering all the possible combinations, we find four different cover graphs for DMV. The particular cover graph generated by the decomposition algorithm depends on the order in which the dependencies are listed. The designer can operate accordingly to steer the algorithm into a particular cover. For instance, the decomposition shown in Table III is the result of having listed all the dependencies with the left side OWNER before the dependencies with the left side DRVL.

5.6 Minimal FD Cover

The set *ZCOVER* obtained upon successful completion of Algorithm 5.1 has some important properties which are discussed next.

Let F be a set of FDs and Γ be a set of attributes. The *closure set of* Γ *relative to* F, defined Γ^*, is defined recursively as follows:

(1) $\Gamma^* \supseteq \Gamma$.
(2) If F contains an FD $\Delta \to A$ where $\Delta \subseteq \Gamma^*$, then $A \in \Gamma^*$.

Thus to construct Γ^*, we start with Γ and then keep adding to it the right side of those FDs in F which satisfy (2) until no new attribute can be added. An important result obtained in [6] states that an FD $\Gamma \to B$ *belongs to* F^+ *iff* $B \in \Gamma^*$. This property allows us to prove the following propositions.

PROPOSITION 5.3. *Let F denote the set of FDs in $R(\Omega)$. If $f \in F$ is elementary and has Ω as scope, then $f \notin (F - \{f\})^+$.*

PROOF. Let $\Gamma \to A$ with $\Gamma = \Omega - \{A\}$ be our elementary FD. The closure set of Γ relative to $F - \{f\}$ is equal to Γ, since the existence in $F - \{f\}$ of an FD $\Delta \to B$, with $B \notin \Delta$ and $\Delta \subset \Gamma$, would contradict the assumption that $\Gamma \to A$ is elementary. Q.E.D.

This proposition justifies the need to enter into *ZCOVER* every elementary FD with scope Ω when a relation $\Pi R(\Omega)$ is decomposed as started in Section 5.4. A different discipline would not guarantee that *ZCOVER* at the end of Algorithm 5.1 is in fact a cover for the FDs of the relation.

Consider now a decomposition step taken according to a multiple elementary MD whereby $\Pi R(\Omega)$ has been decomposed into $\Pi R(\Omega 1)$ and $\Pi R(\Omega 2)$. Let $\bar{F}$, F_1, and F_2 denote the elementary FDs of $\Pi R(\Omega)$, $\Pi R(\Omega_1)$, and $\Pi R(\Omega_2)$, respectively; F_0 denotes the set of elementary FDs of $\bar{F}$ having Ω as scope. It turns out that these three classes of FDs behave independently in terms of the minimal covers. In fact, the minimal cover membership problem for an FD in any of these classes can be resolved solely on the basis of its own classmates disregarding the FDs in the other two classes. This property is trivially true for the FDs in F_0. For F_1 and F_2 we have the following proposition.

PROPOSITION 5.4. *If F_1, F_2, and F_0 are produced by a decomposition step according to a multiple elementary MD, then every $f \in F_1$ belonging to $(F_1 - \{f\} \cup F_2 \cup F_0)^+$ also belongs to $(F_1 - \{f\})^+$.*

PROOF. Let $f = \Gamma \rightarrow A$ and Γ^* be its closure set relative to $(F_1 - \{f\} \cup F_2 \cup F_0)$.

As a first step in constructing Γ^* we only use the FDs in $F_1 - \{f\}$. Thus we construct the closure set of Γ relative to $F_1 - \{f\}$, which we denote Γ'. As the second step we only use FDs of F_2 and compute the closure set of Γ' relative to F_2, denoted Γ''. Γ'' can be partitioned into the following three subsets: $\Gamma_{22} = \Gamma'' - \Gamma'$, $\Gamma_{12} = \Gamma' \cap \Omega_2$, and $\Gamma_{11} = \Gamma' - \Omega_2$. Since we have only used the FDs of F_2 in the second step, we have $\Gamma_{22} \subseteq \Omega_2$ and $\Gamma_{12} \rightarrow \Gamma_{22}$. Now if $\Gamma_{22} \cap \Omega_1$ is not empty we can write $\Gamma_{12} \rightarrow (\Gamma_{22} \cap \Omega_1)$ and conclude that no MD with left side $\Omega_1 \cap \Omega_2$ is elementary, a contradiction. Thus $\Gamma_{22} \subseteq \Omega_2 - \Omega_1$; that is, every Ω_1 attribute of Γ'' was already in Γ'. Thus $A \in \Gamma'' \cap \Omega_1$ iff $A \in \Gamma'$. To complete the proof we need to show that $\Gamma'' = \Gamma^*$. Observe that since $\Gamma'' \cap \Omega_1 = \Gamma'$, Γ'' is also the closure set of Γ relative to $F_2 \cup F_1 - \{f\}$. Moreover, consideration of the FDs in F_0 will not add any new attribute. Indeed, for each $\Delta \rightarrow B \in F_0$, $B \notin \Gamma^*$ because otherwise $\Gamma \rightarrow B$, and "$\Delta \rightarrow B$" would not be elementary. Thus $\Gamma'' = \Gamma^*$. Q.E.D.

We can now state the following important property of Algorithm 5.1.

PROPOSITION 5.5. *The set ZCOVER obtained at successful completion of Algorithm* 5.1 *constitutes a minimal cover for the FDs of* $R(\Psi)$.

PROOF. Since CRCl is verified at each decomposition step, *ZCOVER* is clearly a cover. We need to show that it is minimal. Let $f: \Gamma \rightarrow A$ be an arbitrary member of *ZCOVER*. In Algorithm 5.1 this elementary FD was inserted into *ZCOVER* at STEP2, that is, before $\Pi R(\Gamma \cup \{A\})$ was either recognized as being atomic (STEP3) or decomposed (STEP4). In either case if $\bar{F}$ denotes the set of elementary FDs of $\Pi R(\Gamma \cup \{A\})$, then by Proposition 5.3 we know that $f \notin (\bar{F} - \{f\})^+$. Thus if $\Pi R(\Gamma \cup \{A\})$ is the given initial relation, then the proof is complete. Otherwise consider an ancestor decomposition step where say $\Pi R(\Omega)$ was decomposed into $\Pi R(\Omega_1)$ and $\Pi R(\Omega_2)$. Without loss of generality we assume that $\Pi R(\Omega_1)$ either is $\Pi R(\Gamma \cup \{A\})$ or it was further decomposed into a set of subprojections one of which is $\Pi R(\Gamma \cup \{A\})$. Let F_1 and F_2 be the elementary FDs, respectively, of $\Pi R(\Omega_1)$ and $\Pi R(\Omega_2)$, while F_0 denotes the elementary FDs with scope Ω. According to Proposition 5.4, if $f \notin (F_1 - \{f\})^+$, then $f \notin (F_1 - \{f\} \cup F_2 \cup F_0)$. In other words, a minimal cover of $\bar{F}$ is the union of F_0, a minimal cover of F_1, and a minimal cover of F_2. The proof now proceeds by induction. Q.E.D.

Also, it should be clear from the previous discussion that the admissible covers generated by the decomposition algorithm, when no elementary FD of nonatomic scope exists, define the "independent components" of the original relation [22]. However, given an elementary FD of nonatomic scope Δ, [22] does not pursue any further decomposition of the Δ-projection, while we do. Thus the "atomic relations" of [22] may be further decomposed in our algorithm.

6. DESIGN OF NORMAL FORM SCHEMATA

An interesting application of Algorithm 5.1 is discussed next: We show how it can be used to improve the results produced by Bernstein's algorithm for the design of third normal form schemata [6]. This is but one of the many useful applications of the concepts and the procedures presented in the previous sections. A companion paper discusses their application to the analysis and design of graphical schema, including, in particular, Chen's entity-relationship diagrams [8].

6.1 Bernstein's Algorithm for the Design of 3NF Schemata

A relational schema consists of a set of database relations and the specification of one or more candidate keys for each relation. If X is a key of a relation R, and A is an attribute of R that is not in X, then "$X \to A$" is said to be *embodied* in R.

Bernstein's approach to the design of 3NF schemata is what is called "synthetic." It assumes that the initial description of the database can be formulated directly in terms of functional relationships. These are then used to synthesize algorithmically a relational schema. A first problem encountered in this approach concerns nonfunctional relationships, such as a many-to-many relationship between two attributes. This problem is resolved by introducing new attributes. Thus a nonfunctional connection f among a group of attributes $A_1, A_2, \ldots, A_n$ is represented as the following FD: f: $\{A_1, A_2, \ldots, A_n\} \to \Theta$ where Θ is a newly introduced attribute that is unique to f and does not appear in any othr FD. Each FD representing a nonfunctional relationship has its own private Θ attribute. For instance, a many-to-many relationship between DRIVER and AUTOMOBILE is represented by the FD, f_1: {DRIVER, AUTOMOBILE} $\to \Theta_1$. The domain underlying these Θ attributes is the pair $\{0, 1\}$. Thus corresponding to the pair (driver-1, automobile-1), Θ_1 has value 1 if driver-1 actually drives automobile-1, and the value of 0 otherwise. This technique allows the designer to produce a set of FDs, denoted H_0, which characterizes both functional and nonfunctional relationships of interest. Given H_0, Bernstein's algorithm for designing a 3NF schema is stated below. (For simplicity, we have assumed that the FDs in H_0 are elementary.)

Algorithm 6.1: Bernstein's Algorithm to Design 3NF Schemata

(1) (Find covering.) Find a minimal cover H for H_0.
(2) (Partition.) Partition H into groups such that all of the FDs in each group have identical left sides.
(3) (Merge equivalent keys.) Let $J = \varnothing$. For each pair of groups, say H_1 and H_2, with left sides X and Y, respectively, merge H_1 and H_2 together if there is a bijection $X \leftrightarrow Y$ in H^+. For each such bijection, add $X \to Y$ and $Y \to X$ to J. For each $A \in Y$, if $X \to A$ is in H, then delete it from H. Do the same for each $Y \to B$ in H with $B \in X$.
(4) (Eliminate transitive dependencies.) Find an $H' \subseteq H$ such that $(H' + J)^+ = (H + J)^+$ and no proper subset of H' has this property. Add each FD of J into its corresponding group of H'.
(5) (Construct relations). For each group, construct a relation consisting of all the attributes appearing in that group. Each set of attributes that appears on the left side of any FD in group is a key of the relation.

In [6] it is proved that every schema produced by this algorithm has the following properties:

(a) Every relation in the schema is 3NF.
(b) If F denotes the set of all FDs embodied in the schema relations, then $F^+ = H_0^+$.
(c) The total number of relations in the schema is minimal (i.e., no schema with fewer relations has property (b)).

These three properties are closely reminiscent of the three objectives presented at the beginning of Section 5.3. Property (a) guarantees that the components have a well-defined form. Property (b) guarantees that the structural information contained in H_0 is preserved in the final schema. Property (c) suggests that the schema is minimal in the sense that no relation can be dropped without losing the last property. Thus the schemata obtained using Bernstein's algorithm have some very desirable properties. Nevertheless, some problems remain. The next section illustrates these problems and their resolution through the application of Algorithm 5.1.

6.2 Problems and Solutions

A first problem in the approach taken in [6] is the treatment of nonfunctional relationships. Consider our previous relation DMV. Here we find a pair of nonfunctional relationships: one which associates the traffic violation history (VIOL, DATE) of a driver with his drivers license number (DRVL), the other which associates (VIOL, DATE) with the ID of this person (OWNER). Although there may be other nonfunctional relationships, we shall only consider these to illustrate our point. As previously explained, these two will be modeled by the pair of FDs:

$$\text{DX1: \{VIOL, DATE, DRVL\}} \rightarrow \Theta_1,$$

$$\text{DX2: \{VIOL, DATE, OWNER\}} \rightarrow \Theta_2.$$

In addition to these, the H_0 will include the previous FDs, that is, D1, D2, D3, D4, D5, D6, D8, D10, and D13. Thus a minimum cover should be computed next. One minimal cover consists of D1, D2, D3, D5, D8, D10, D13, DX1, and DX2. Using this minimal cover, Algorithm 6.1 produces the following schema (different keys within the same relation are denoted by different styles of underlining):

$$\text{DMV237}(\underline{\text{LIC}}, \text{MAKE, YEAR, MODEL, OWNER}) \tag{6.1}$$

$$\text{DMV8}(\underline{\text{MAKE}}, \underline{\text{YEAR}}, \underline{\text{MODEL}}, \text{VALUE}) \tag{6.2}$$

$$\text{DMV10}(\underline{\text{OWNER}}, \underline{\text{DRVL}}) \tag{6.3}$$

$$\text{DMVX}(\underline{\text{VIOL}}, \underline{\text{DATE}}, \underline{\text{DRVL}}, \underline{\text{OWNER}}, \Theta_1, \Theta_2). \tag{6.4}$$

This schema presents two obvious redundancy problems. The first is that (6.4) has (6.3) as projection; yet neither of these two relations can be omitted without losing information about FDs (property (b)). The second problem is that the columns Θ_1 and Θ_2 in (6.4) are logically equivalent since $\Theta_1 = 1$ implies $\Theta_2 = 1$, and vice versa. Thus they could (both) be dropped from this relation. This second problem, in particular, occurs independently of the minimal cover selected for the previous FDs, since it follows from the treatment of "nonfunctional" relationships used in [6].

Here, however, by using MDs, we can treat nonfunctional relationships in a formal and consistent fashion. It is easy to see how Algorithm 5.1 can be used in conjunction with Algorithm 6.1 to eliminate the previous problems in the framework of the decomposition approach. The designer works with functional and multivalued dependencies, as described in the previous sections. Then, using Algorithm 5.1, he obtains the set of atomic subprojections *ACOVER* and the set of elementary FDs *ZCOVER*. Next, Algorithm 6.1 is applied using *ZCOVER* as the minimum FD cover H. This produces a minimal set of 3NF relations which completely characterize the functional relationships of the initial relation. Indeed, each member of *ZCOVER*, say $(i:\Gamma \rightarrow A)$, is embodied in a 3NF relation thus obtained. The same relation also embodies every other FD of equal scope (i.e., having the same label i). Finally, we need to include in the schema the nonfunctional relationships. These are found by looking in *ACOVER* for those members, say $(j:\Delta)$, for which no FD with label j is found in *ZCOVER*. For each of these a separate relation, say $R_j(\Delta)$, with key Δ, will be generated. Applying this discipline to our example DMV, per Table III, we first obtain relations (6.1)–(6.3); finally, to represent the nonfunctional relationship for label 6, we use

$$\text{DMV6}(\underline{\text{VIOL}}, \underline{\text{DATE}}, \underline{\text{OWNER}}). \tag{6.5}$$

The previous redundancy problem has been cured.

Since the treatment of nonfunctional relationships was a major motivation of our research, the benefits realizable in this area were not too surprising. On the other hand the benefits obtainable in the treatment of functional relationships were a pleasant surprise. We can illustrate this assertion by means of the following example which describes the rank order of students at the completion of their courses:

RANK(CT, C#, SN, P) (6.6)

CT	C#	SN	P
MATH	443	JONES	1
MATH	443	SMITH	2
ENGL	012	WANG	1
ENGL	012	JONES	2
HIST	179	SMITH	1
HIST	179	DOE	2
HIST	179	BROWN	3

We assume that a student is identified by his name (SN). Both the course title (CT) and the course number (C#) are sufficient to identify a course (thus C# $\leftrightarrow$ CT). The column denoted (P) shows the position obtained by a student in the courses he completed. Assume now that no two students can have the same position in any given course (no ties). The set of elementary FDs characterizing this relation is as follows (in the synthetic approach the designer would omit (6.6) and start here):

D1: CT → C#

D2: C# → CT

D3: {CT, SN} → P (6.7)

D4: {C#, SN} → P

D5: {CT, P} → SN

D6: {C#, P} → SN.

This set has four minimal covers. Assuming that {D1, D2, D3, D6} is found in the first step of Algorithm 6.1 then at the end of the algorithm we obtain the following pairs:

R12(CT, C#) (6.8)

R36(CT, C#, SN, P). (6.9)

The candidate keys of (6.8) are

{CT} and {C#}.

The candidate keys of (6.9) are

{CT, SN}, {C#, SN}, {CT, P} and {C#, P}.

Indeed the set of FDs embodied in this schema is exactly (6.7).

A serious redundancy problem characterizes the extension of this pair of relations: (6.8) is a projection of (6.9). However, we cannot simply eliminate (6.8) from the schema since this relation embodies the two FDs D1 and D2 which are not inferable from the keys of (6.9). Thus elimination of (6.8) would cause the loss of very important structural information. This observation only restates the general property proved in [6] that every schema produced by Algorithm 6.1 has a minimal number of relations. Thus no relation can be dropped without losing structural information. Apparently, therefore, we are locked in the dilemma of either accepting the previous redundancy or losing structural information. Fortunately, this is not the case, and a simple solution to this problem is available: to use only those FD covers which satisfy the admissibility condition. For instance, if the admissible FD cover {D1, D2, D4, D6} is used, Algorithm 6.1 produces the schema:[13]

R12(CT, C#) (6.10)

R46(SN, C#, P). (6.11)

Combinations {CT} and {C#} are the keys for (6.10). The pairs {SN, C#} and {C#, P} are the keys for (6.11). The elementary FDs D1, D2, D4, and D6 are embodied in the schema, while D3 and D5 are derived from them by transitivity. Thus this schema completely characterizes the intension of the case at hand. Moreover, while (6.10) is in fact our old (6.8), (6.11) is a proper subprojection of (6.9) since the column CT is missing. Thus the content of (6.10) cannot be derived

[13] We indicate here the various candidate keys by different underlines (______ and ~~~~~~).

by projection from (6.11). The redundancy problems affecting the previous schema have been removed. Relation (6.7) has a second nonadmissible cover: {D1, D2, D4, D5}. This produces the old pair of relations (6.8), (6.9) with their well-known problems. Moreover, there exists a second admissible cover: {D1, D2, D3, D5}. From this, Algorithm 6.1 produces

$$R12(\underline{CT}, C\#)$$
$$R35(\underline{SN, CT}, P).$$

This schema is free of redundancy problems.

The above example has illustrated the benefits of using only admissible FD cover in designing 3NF relations. These benefits are not restricted to the design of 3NF schemata, but they extend to other forms of schemata. A companion paper discusses their application to the design of graphical schemata including E-R diagrams.

6.3 Validation of Results

A major effort in current software research is directed toward producing results which are verifiable and may be validated in some formal fashion. The state-of-the-art methods to design database schemata present some problems in this respect. In the framework of the relational model, for instance, we find Codd's pioneering work on 3NF. Codd's approach was inductive. Some carefully chosen examples were first used to illustrate the anomalies effecting unnormalized relations; then the definition of 3NF was proposed to remove the anomalies affecting those relations. But it was later realized that a number of anomalies were not eliminated by 3NF, and BCNF was proposed to cure these further anomalies. It was later learned that this second definition did not remove the important class of anomalies connected with multivalued dependencies. Again a new definition was proposed as a remedy: 4NF, the fourth normal form definition [16]. Unfortunately, no proof that the fourth normal form removes all anomalies was given; indeed there does not even exist a formal definition of anomaly.[14] One wonders whether the fourth normal form will also be superseded by a new definition. Thus the current concepts of normal forms are somehow beset by the dilemma of being a highly formal means to obtain an objective which cannot be formalized or verified. Moreover, the range of possible semantic structures and constraints is so wide, that there might not exist a single normal form capable of generating anomaly-free schemata in every situation. Consider, for example, a dictionary of corresponding technical terms in three different languages, say French, German, and English:

DICT(F,	G,	E)
f1	g1	e1
f2	g1	e1
f1	g1	e2
f2	g1	e2
f3	g2	e3
f3	g3	e3

[14] A precise definition of anomaly has recently been proposed in [21].

Assume that one object or concept might be described by more than one synonym in each language. One term, however, never describes more than one concept. Then the triples of the relation can be partitioned into groups, one per concept, each group being in turn a separable relation. For example, the first four tuples above form a group which is the Cartesian product {f1, f2} × {g1} × {e1, e2}. This relation is characterized by the following dependencies:

$$\begin{aligned} &F \twoheadrightarrow G,\ G \twoheadrightarrow F,\ E \twoheadrightarrow F \\ &F \twoheadrightarrow E,\ G \twoheadrightarrow E,\ E \twoheadrightarrow G \end{aligned} \tag{6.12}$$

with all the anomalies arising thereby (e.g., the addition of a new term as synonym of g1 requires the insertion of four tuples into the first group).

None of the three possible decompositions into 4NF cures the previous anomalies. For instance, if we decompose into ΠDICT(F, G) and ΠDICT(G, E), we find that the addition of a synonym of g1 requires the insertion of two pairs in the first subrelation and two pairs in the second. (Another example of similar structures can be constructed by listing in a flat table the various terms which describe comparable database concepts under the relational network and hierarchical approach jargon.)

In conclusion, although a given normal form definition might work well for the vast majority of cases, there always remains the unusual or unpredictable case which is recalcitrant to the canonical treatment. In these cases, the formal algorithms for the design of normal form schemata might do more harm than good since they replace the designer's guiding intuition by the illusory assurance of mathematical formalism. Indeed, ensuring that the schema is in normal form constitutes only a facet of the larger and more complex problem of capturing the semantics of the database through the schema. The decomposition algorithm proposed in this paper offers a meaningful contribution to the solution of this problem. First, we have noted the elusive nature of update anomalies whose interpretations depend upon the usage of database relations, as well as the syntactic structure of their dependencies. Complete relatability was therefore proposed as a design criterion more liable to be verified and consonant with logical database design.

The decomposition procedure presented here validates each decomposition step according to the complete relatability criterion. Only when this is met is a final decomposition returned to the designer. In every other case the procedure halts and through the REPORTFAILURE routine reports and returns control to the designer. He can then analyze the unusual circumstances at hand, verify that they are not the result of previous oversights on his part, and finally decide on the particular solution that the case demands. For instance, each dependency of the relation DICT fails to satisfy the CRC2 condition (5.11); thus REPORTFAILURE is invoked. Thereby the designer is made aware that no decomposition is acceptable and that a different solution is needed. This could, for instance, be the introduction of a new domain, say C for concept, whereby each group of DICT receives a unique identifier. The decomposition algorithm can then be used to decompose successfully the expanded relation (characterized by the dependencies $F \rightarrow C$, $G \rightarrow C$, $E \rightarrow C$, $C \twoheadrightarrow F$, $C \twoheadrightarrow G$, and $C \twoheadrightarrow E$ in addition to those of set (6.12)) into the schema of Figure 2.

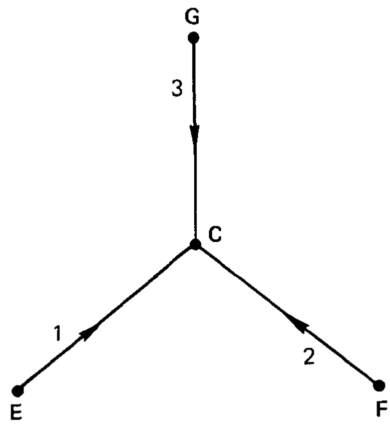

Fig. 2. The structure of a dictionary.

A second case in which the decomposition algorithm will fail concerns the relation:

$$\text{DEP}(\text{E}\#, \text{D}\#, \text{PJ}),$$

with dependencies:

$$\text{D1: E}\# \rightarrow \text{D}\#$$

$$\text{D2: E}\# \twoheadrightarrow \text{PJ}$$

$$\text{D3: PJ} \rightarrow \text{D}\#$$

$$\text{D4: PJ} \twoheadrightarrow \text{E}\#.$$

This relation describes the employees working in a department and the projects assigned to this department, and also the employees working on the various projects. A decomposition according to D1 or D2 causes the loss of D3; symmetrically, a decomposition according to D3 or D4 results in a loss of D1. Only a decomposition into the three atomic subrelations (each with the FDs therein contained)

$$\begin{aligned} &\Pi\text{DEP}(\text{E}\#, \text{D}\#)\\ &\Pi\text{DEP}(\text{E}\#, \text{PJ})\\ &\Pi\text{DEP}(\text{PJ}, \text{D}\#)\end{aligned} \tag{6.13}$$

will ensure the preservation of both FD structure and content of the original relation. There is a content redundancy, however, associated with this solution, since the first (the last) in (6.13) is a projection of the join of the last (the first) two. Moreover, these three relations obey an integrity constraint that does not follow from the definition of the FDs in (6.13) although it is implied by the FDs in the original DEP: An employee can be assigned to a project only if he and the project belong to the same department. Thus if (6.13) were to become a schema (e.g., with the addition of keys to define FDs), there would be an additional integrity constraint of which the user had better be aware. In this case, too, therefore, REPORTFAILURE can be seen as a desirable occurrence which forewarns the designer of an uncommon situation of which he had better be aware. The problem of when and how interrelational constraints, such as the redundancy constraint, should be used to ensure that a decomposition preserves both content and structure of the original relation is not totally understood yet and deserves further investigation.

There is also some concern regarding the potentially nondeterministic nature of our decomposition algorithm with respect to REPORTFAILURE. The question is whether it is possible, once REPORTFAILURE has occurred, to backtrack and apply the dependencies in a different order to obtain a successful completion. For the examples we have considered, REPORTFAILURE occurred not because of the inadequacy of the algorithm to find the proper solution but rather for an objective problem with the situation at hand. For these examples, at least, backtracking and retrying does not change the final result since it does not remove the problem. In general, however, it is clear that we need a better understanding of the failure situations. In particular the issue of whether rules stronger than joinability can be used is an important one and deserves further investigation.

7. CONCLUSION

This paper has presented a formal approach to the analysis of the dependency structure of database relations and to the design of relational schemata using these dependencies. Our investigation of the state of the art has indicated that the concept of complete relatability constitutes a more rigorous criterion for designing logical schemata than the removal of anomalies. We also have recognized that dependencies and minimal covers supply a powerful tool for the analysis and design of relational schemata. Since the minimal cover technique can only be correctly applied to the dependencies of a given relation, we have adopted the decomposition approach whereby the initial relations are refined into smaller subcomponents. We have also developed a unified treatment of functional and multivalued dependencies. This has led to the concepts of elementary FDs and multiple elementary MDs which help the designer in characterizing the dependency structure of relations and supply the basis for our decomposition algorithm. This algorithm, by combining search for a minimal cover and decomposition, generates schemata which ensure complete relatability. The concepts of admissibility of covers and validation of results implemented by the algorithm have also been discussed and their importance in schema design underscored. As a first application, it has been shown how these results can be used to improve the design of 3NF schemata.

ACKNOWLEDGMENT

The authors would like to express their sincere appreciation to the referees whose incisive critiques and comments have played an important role in the development of this paper.

REFERENCES

1. ARMSTRONG, W.W. Dependency structures of data base relationships. Information Processing, vol. 74. North-Holland, Amsterdam, 1974, pp. 580-583.
2. BEERI, C., FAGIN, R., AND HOWARD, J.H. A complete axiomatization for functional and multivalued dependencies in database relations. *Proc. ACM SIGMOD Int. Conf. Management of Data*, Toronto, Canada, 1977, pp. 47-61.
3. BEERI, C. On the membership problem for multivalued dependencies in relational databases. TR-229, Princeton Univ., Princeton, N.J., Sept. 1977.
4. BEERI, C., BERNSTEIN, P.A., AND GOODMAN, N. A sophisticate's introduction to database normalization theory. *Proc. Int. Conf. Very Large Data Bases*, West Berlin, Germany, 1978, pp. 113-124.
5. BEERI, C., AND BERNSTEIN, P.A. Computational problems related to the design of normal form relational schemas. *ACM Trans. Database Syst. 4*, 1 (March 1979), 30-59.
6. BERNSTEIN, P.A. Synthesizing third normal form relations from functional dependencies. *ACM Trans. Database Syst. 1*, 4 (Dec. 1976), 272-298.
7. BISKUP, J. On the complementation rule for multivalued dependencies on database relations. *Acta Inf. 10*, 3 (1978), 297-305.
8. CHEN, P.P. The entity-relationship model toward a unified view of data. *ACM Trans. Database Syst. 1*, 1 (March 1976), 9-36.
9. CODD, E.F. A relational model of data for large shared data banks. *Commun. ACM 13*, 6 (June 1970), 377-387.
10. CODD, E.F. Further normalization of the data base relational model. In *Data Base Systems*, Courant Computer Science Series, vol. 6, R. Rustin, Ed., Prentice-Hall, Englewood Cliffs, N.J., 1972, pp. 33-64.
11. CODD, E.F. Recent investigations in relational database systems. Information Processing, vol. 74. North-Holland, Amsterdam, 1974, pp. 1017-1021.
12. DATE, C.J. *An Introduction to Database Systems*, 2nd ed. Addison-Wesley, Reading, Mass., 1977.
13. DELOBEL, C., AND CASEY, R.G. Decomposition of a data base and the theory of Boolean switching functions. *IBM J. Res. Dev. 17*, 5 (Sept. 1972), 374-386.
14. DELOBEL, C., AND LÉONARD, M. The decomposition process in a relational model. *Proc. Int. Workshop on Data Structure Models for Information Systems*, Namur, Belgium, May 1974, pp. 57-80.
15. FAGIN, R. Multivalued dependencies and a new normal form for relational databases. *ACM Trans. Database Syst. 2*, 3 (Sept. 1977), 262-278.
16. FAGIN, R. The decomposition versus synthetic approach to relational database design. *Proc. Third Int. Conf. Very Large Data Bases*, Tokyo, Japan, Oct. 1977, pp. 441-446.
17. GALIL, Z. An almost linear time algorithm for computing a dependency basis in a relational database. IBM Res. Rep. RJ2656, IBM Research Labs., San Jose, Calif., Oct. 1979.
18. JOINT GUIDE AND SHARE DATABASE REQUIREMENT GROUP. Requirements for a Database Management System, November 1970. (Available from SHARE, Suite 750, 25 Broadway, New York, NY 10004.)
19. HAGIHARA, K., ITO, M., TANIGUCHI, K., AND KASAMI, T. Decision problems for multivalued dependencies in relational databases. *SIAM J. Comput. 8*, 2, 247-264.
20. MENDELZON, A.O. On axiomatizing multivalued dependencies in relational databases. *J. ACM 26*, 1 (Jan. 1979), 37-44.
21. MONAHAN, J., AND MELKANOFF, M.A. To be published.
22. RISSANEN, J. Independent components of relations. *ACM Trans. Database Syst. 2*, 4 (Dec. 1977), 317-325.
23. SAGIV, Y. An algorithm for inferring multivalued dependencies that works also for a subclass of proportional logic. VIVCDCS-R-79-954, Dep. Computer Science, Univ. Illinois, Urbana, Jan. 1979.
24. SILVA, A.M. Derivation of relational dependencies through case-study analyses. M.S. Thesis, Computer Science Dep., Univ. California, Los Angeles, 1978.
25. TANAKA, K., KAMBAYSHI, Y., AND YAJIMA, S. On the representability of decompositional schema design with multivalued dependencies. Rep. ER79-01, Dep. Information Science, Kyoto Univ., Japan, Jan. 1979.

26. WANG, C.P., AND WEDEKIND, H.H. Segment synthesis in logical data base design. *IBM J. Res. Dev. 19*, 1 (Jan. 1975), 71–77.
27. ZANIOLO, C. Analysis and design of relational schemata for database systems. Rep. UCLA-ENG-7669, Computer Science Dep., Univ. California, Los Angeles, July 1976.
28. ZANIOLO, C., AND MELKANOFF, M.A. A formal approach to the definition and the design of conceptual schemata for database systems. To appear in *ACM Trans. Database Syst.*

Received November 1977; revised December 1979; accepted March 1980