THE DESIGN OF RELATIONAL INFORMATION SYSTEM
ACCORDING TO DIFFERENT KINDS OF DEPENDENCIES

C. DELOBEL⁽¹⁾ & E. PICHAT⁽²⁾

ABSTRACT

The purpose of this paper is to present

- (a) a survey of different approaches proposed for the conceptual design of logical schemas for relational data base systems,
and
- (b) a new approach based upon the study of the decomposition structure of a relation which ensures the complete joinability of data.

*(1) - Laboratoire I.M.A.G., Université Scientifique et Médicale de Grenoble,
B.P. 53, 38041 GRENOBLE Cedex (France).*

*(2) - Institut d'Informatique d'Entreprise, C.N.A.M., 292 Rue Saint Martin,
75141 PARIS Cedex 03 (France).*

1. INTRODUCTION

The purpose of this paper is to present

- (a) a survey of different approaches proposed for the conceptual design of logical schemas for relational data base systems, and
- (b) a new approach based upon the study of the decomposition structure of a relation which ensures the complete joinability of data.

In the design of data base schema there are basically two approaches : the decomposition and the synthetic approach.

The decomposition approach has been proposed by Codd [COD 70] [COD 71] in the presentation of the relational data model. In this model, the concept of functional dependency is an important element when one is considering how to group attributes to form relations. The properties of functional dependencies, studied in [BOI 69], [COD 71], [DEL 71], [DC 73], [ARM 74], are the basic elements used to define the three CODD's normal forms and the normalization process. This normalization process can be viewed as the decomposition of a relation into two sub-relations (or more), so that the original one can be regenerated by the composition operation of the two sub-relations.

The synthetic approach attacked the same problem. The objective is that a designer should synthesize the data base schema algorithmically from a given set of semantic properties of data.

The first attempts based only on functional dependencies [DC 73], [WW 75], [SS 75] led to some imperfections and do not guarantee that the relations are in CODD's third normal form. The work of Bernstein [BER 75], [BER 77] develops an effective procedure for synthesizing relations satisfying CODD's third normal form. A similar approach is proposed in [LR 76], [LRP 77] with efficient algorithms to derive both the closure and a minimum covering set of functional dependencies by using the equivalence between the operations on functional relationships and the operations in boolean algebra.

If the functional relationship concept plays probably the most important role in the process of defining relational schema, it is not the only one. From an idea contained in [BOI 69] we have defined [DL 74] another concept : the first order hierarchical decomposition which allows to decompose relations independently of functional relationships. More recently, FAGIN [FAG 76] and ZANILOLO [ZAN 76], [ZAM 77] have introduced the notion of multivalued dependency which includes, as a special case, the functional relationship.

The following section, section 2 of the paper, reviews the definition of a relation and the condition for a relation to be decomposable.

Section 3 presents the different types of constraints describing a relation :

functional dependency, multivalued dependency, first order hierarchical decomposition. The properties of this type of constraints are reviewed and their role in the process design is illustrated through examples.

In section 4, the basic properties of the decompositions of a relation are studied and it is shown how it is possible, from these properties, to develop a new approach for the design of logical schemas. In this approach, the concept of maximal decompositions for a family of decompositions is an important element of the process.

2. THE RELATIONAL MODEL

The intent of this section is to review some definitions. Familiarity with the fundamental concepts of the relational model as presented in [COD 70] and [BER 77], and especially with the concept of functional dependency in [ARM 74] is assumed. We use throughout the section the terminology found in [BFH 77].

2.1. Relations and constraints

In the database terminology, the word relation is sometimes confusing because it is used to denote a set of tuples and a structural description of sets of tuples. It is important to make the distinction clear: when we talk about set of tuples we shall use the word relation and in the other case we shall use the word relational schema. With this approach a relation is an instance of a relational schema.

Attributes are identifiers taken from a finite set $A_1, A_2, \dots, A_n$. Each attribute A_i has associated with it a domain, denoted by $\text{DOM}(A_i)$, which is the set of possible values for that attribute. We shall use the letters $A, B, \dots$, for single attributes and the letters $X, Y, \dots$, for sets of attributes.

An *X-value* is an assignment of values to the attributes of X from their domains.

Also, if X and Y are sets of attributes (not necessarily disjoint), then we write $X \cup Y$, $X \cap Y$ for the union and the intersection of X and Y , but if X and Y are disjoint, we shall use (X, Y) for the union.

A *relation* on the set of attributes $\{A_1, A_2, \dots, A_n\}$ is a subset of the cartesian product $\text{DOM}(A_1) \times \text{DOM}(A_2) \times \dots \times \text{DOM}(A_n)$. The elements of the relation are called *tuples*. A relation R on $\{A_1, A_2, \dots, A_n\}$ will be denoted $R(A_1, A_2, \dots, A_n)$. If one wants to distinguish among the attributes the sets of disjoint attributes X and Y , we shall use the notation $R(X, Y, \dots)$. In this case if $X = \{A_1, A_2\}$ and $Y = \{A_3, A_4, A_5\}$ the tuple $(x, y, \dots)$ stands for $(a_1, a_2, a_3, a_4, a_5, \dots)$.

A *constraint* involving the set of attributes $A_1, A_2, \dots, A_n$ is a predicate on the collection of all relations on this set. A relation $R(A_1, A_2, \dots, A_n)$ obeys the constraint if the value of the predicate for R is "true". A constraint is defined by giving a notation or a language for expressing it and the condition under which a relation obeys it.

A constraint can be seen as an intrinsic property of the data; for example, suppose that parts in an inventory are described by a relation $R(\text{PART-NUMBER}, \text{COLOR}, \text{PRICE}, \dots)$. A priori, any relation of this form can exist in the database. However if one specifies a constraint: the PRICE must range between \$0 and \$100 a piece, then only relations in which this constraint is valid can exist in the database. Similarly the specification that the knowledge of the PART-NUMBER-value implies the COLOR-value is also another type of constraint. In this case, this type of constraint is called

a functional dependency and denoted $\text{PART-NUMBER} \rightarrow \text{COLOR}$.

In this paper, we are investigating only some constraints : functional dependency, multivalued dependency, first order hierarchical decomposition which are presented in section 3.

2.2. Operations on relations

In his original presentation of the relation model [COD 70], Codd introduced the relational algebra as a data manipulation language. There are two basic operations that will be of some interest of us : projection and natural join.

The *projection* of a relation $R(X,Y,Z)$ over a set of attributes X is the restriction of R over the attributes in X ; this operation will be denoted $R[X]$, and defined by $R[X] = \{x | \exists y \exists z : (x,y,z) \in R\}$.

As a special case of projection we denote by $R[x,Y]$ the projection of R over Y from an X -value x , $R[x,Y] = \{y | \exists z : (x,y,z) \in R\}$.

The *natural join* operation is used to make a connection between attributes that appear in different relations. Let $R(X,Y)$ and $S(X,Z)$ be two relations: then the natural join $R \Join S$ is the set of $\{(x,y,z) | (x,y) \in R \text{ and } (x,z) \in S\}$; $R \Join S$ is a relation defined over the attributes $\{X,Y,Z\}$.

2.3. Decomposition of a relation

Let $R(X,Y,Z)$ be a relation ; we shall say that R is *decomposable* if there exists two relations S and T such that :

- (a) S and T are projections of R : $S = R[X,Y]$, $T = R[X,Z]$
- (b) the natural join of S and T is R : $R = S \Join T$.

In other way, we shall say that the pair $((X,Y), (X,Z))$ constitutes a *decomposition* of R .

This concept of decomposition is very important in the process for designing a database schema, because in place of storing the relation R in the database, we can store only the projections of R . The purpose of this paper is to study under what conditions we can decompose a relation.

First, we shall recall basic characterizations for a relation to be decomposable ; the proofs have been given elsewhere.

Proposition 1 : $R(X,Y,Z)$ is decomposable iff for all X -values x which are in R

$$R[x,Y,Z] = R[x,Y] \times R[x,Z]$$

where the operation $\times$ denotes the cartesian product.

Proposition_2 : $R(X,Y,Z)$ is decomposable iff for all X-value and Y-value which are in R :

$$R[x,y,Z] = R[x,Z].$$

Example_1 : Let $R(\text{BOOK}, \text{CLASS}, \text{STUDENT}, \text{PROFESSOR})$ be a relation where the interpretation can be given intuitively as follows. Each CLASS has various STUDENTS, but a STUDENT is in one CLASS, a CLASS has one PROFESSOR. Each CLASS has a given set of BOOKS as reference's books. A sample of the relation is given by the table below.

R :	BOOK	CLASS	STUDENT	PROFESSOR
	b_1	math	peter	mike
	b_1	math	john	mike
	b_3	programming	jane	mike
	b_2	math	peter	mike
	b_3	programming	james	mike
	b_2	math	john	mike

One can notice that by application of proposition 1, the relation R is decomposable into two relations $S = R[\text{BOOK}, \text{CLASS}]$ and $T = R[\text{CLASS}, \text{STUDENT}, \text{PROFESSOR}]$.

S :	BOOK	CLASS	T :	CLASS	STUDENT	PROFESSOR
	b_1	math		math	peter	mike
	b_2	math		math	john	mike
	b_3	programming		programming	jane	mike
				programming	james	mike

At the present time, we only make a constatation that relation R is decomposable ; we shall see in the next chapter that the decomposition property is related to the type of constraints.

3. THE TYPE OF CONSTRAINTS

3.1. Functional dependencies

3.1.1. Definition

Functional dependencies form a family of constraints. The properties of functional dependencies have been studied extensively in [DC 73], [ARM 74], [BER 77] ; therefore we recall here only the definition.

A *functional dependency* (abbr. FD) is a sentence denoted $f : X \rightarrow Y$ where f is the name of the functional dependency and X and Y are sets of attributes. A functional dependency $f : X \rightarrow Y$ holds in $R(U)$ where X and Y are subsets of U , if for every tuple u and v of R $u[X] = v[X]$ implies $u[Y] = v[Y]$ ($u[X]$ denotes the projection of the tuple u on X). According to the definition f can be seen as the unique application from $R[X]$ to $R[Y]$, therefore we can omit the name of the FD and write only $X \rightarrow Y$.

A *full functional dependency* $X \twoheadrightarrow Y$ is an FD such that there exists no proper subset $X' \subset X$ with $X' \rightarrow Y$.

We can relate the concept of functional dependency to the concept of decomposition according to proposition 3 which is easy to prove.

Proposition 3 : If $R(X,Y,Z)$ is a relation such that functional dependency $X \rightarrow Y$ holds then R is decomposable and we have $R = R[X,Y] * R[X,Z]$.

The proposition 3 is the first proposition which establishes an association between a constraint and the decomposition property.

3.1.2. Properties of functional dependencies

The properties of FD's are important in the design of relational schemas and we shall use them later. These properties can be seen as inference rules for FDs, that is rules that deal with the implication of new FDs from a given set of FDs.

Let $R(U)$ be a relation defined over a set U of attributes, and X,Y,Z,W be subsets of U .

F1 - Reflexivity : if $X \supseteq Y$ then $X \rightarrow Y$

F2 - Augmentation : if $X \rightarrow Y$, then for all Z $X \cup Z \rightarrow Y \cup Z$

F3 - Transitivity : if $X \rightarrow Y$ and $Y \rightarrow Z$ then $X \rightarrow Z$

F4 - Pseudo-transitivity : if $X \rightarrow Y$ and $Y \cup Z \rightarrow W$ then $X \cup Z \rightarrow W$

F5 - Additivity : if $X \rightarrow Y$ and $X \rightarrow Z$ then $X \rightarrow Y \cup Z$.

Example_2 : We can define for the relation R given in example 1, the FDs according to the relationships properties between attributes

STUDENT $\rightarrow$ CLASS

CLASS $\rightarrow$ PROFESSOR.

According to F3 we can derive STUDENT $\rightarrow$ PROFESSOR by transitivity.

3.1.3. Decomposition based upon functional dependencies

It is well known that if someone adopts for the relational schema the relation R only, this relation suffers from a lot of anomalies. These anomalies occur when one wants to make an update, insertion or delete operation. The problem of designing relational schemas is equivalent to replace the relation R by a set of relations which give the same information without destroying the relationships between attributes. This problem is equivalent to find for the relation R a decomposition such each relation in the decomposition contains less anomalies than the original one. This problem has been, first, identified by CODD and leads him to the definitions of normal's form of a relation. Two main approaches have been recognized as valuable to produce suitable normal forms : the decomposition approach and the synthetic approach related with the irredundant covering technique.

The decomposition approach consists to apply successively the proposition 3 to different functional dependencies. For example, the application of STUDENT $\rightarrow$ CLASS in example 1 gives the decomposition of the relation R defined over the attributes STUDENT, CLASS, PROFESSOR, BOOK into two relations R1 defined over the attributes STUDENT, CLASS and R2 defined over STUDENT, PROFESSOR, BOOK. It is still again possible to decompose R2, because the FD STUDENT $\rightarrow$ PROFESSOR holds also in R2, into two relations R21 defined over STUDENT, PROFESSOR and R22 defined over STUDENT, BOOK.

The overall decomposition process can be represented by a tree as the one shown in figure 1. The terminal nodes describe the attribute sets of each relation obtained at the end of the decomposition process. To each non-terminal node we can associate an attribute set equal to the union of the attribute sets of their successors. Each non-terminal node corresponds a decomposition step which is labelled according to a FD.

Fig. 1 - A decomposition of a relation

The difficulty of this approach is the possibility of multiple decompositions. It may be seen that the relation R has in addition the following decomposition as shown by figure 2.

Fig. 2 - A decomposition of a relation

These two decompositions possess an anomaly because both they generate the relation $R[STUDENT, BOOK]$. The meaning of this relationship between the attributes is : $s \in STUDENT$ and $b \in BOOK$ are related if there exists a $CLASS$ c such the $STUDENT$ s is in the $CLASS$ c and the $BOOK$ b is used by the $CLASS$ c . In other way one can say that $STUDENT$ and $BOOK$ are not directly related.

The irredundant cover technique is more systematic. It proceeds in different steps. The first step is to construct the closure and an irredundant cover.

Let F be a given set of FDs, the *closure* of F denoted F^+ is the set of FDs which can be derived from F through the inference rules of FDs.

A *cover* C is a set of FDs from which all others can be derived, that means that : $C^+ = F^+$. An *irredundant cover* F_m is a set of FD_s such the closure of F_m is equal to closure of F , i.e. $(F_m)^+ = F^+$, and there exists no proper subset F' of F such $(F')^+ = F^+$.

The problem of obtaining an efficient irredundant cover algorithm has received wide attention [DC 73], [LR 76], [LRP 77].

As we can see from our previous example 2, the irredundant cover of $F = \{STUDENT \rightarrow CLASS, CLASS \rightarrow PROFESSOR\}$ is equal to F . So, the two relations built from the two FDs in the irredundant cover, say R_1 defined over $STUDENT, CLASS$ and R_2 defined over $CLASS, PROFESSOR$, are not enough to regenerate R .

The second step is to synthesize valuable relational schemas from an irredundant cover set of FDs regarded as semantic primitives. Different algorithms have been proposed to obtain optimal set of relations in third normal form from an irredundant cover [BER 75], [BER 77], [FLO 77], [OSB 77].

3.2. Multivalued dependencies

3.2.1. Definition

In a functional dependency $X \rightarrow Y$, the knowledge of the X -value determines a unique

Y-value. In a multivalued dependency the X-value determines a set of Y-values. According to the definition given by Fagin [FAG 76] and Zaniolo [ZAN 76], a *multivalued dependency* (abbr. MVD) is a sentence denoted $g : X \twoheadrightarrow Y$ where g is the name of the multivalued dependency and X and Y are sets of attributes. A multivalued dependency $g : X \twoheadrightarrow Y$ holds in $R(X,Y,Z)$, where X,Y,Z are disjoint sets of attributes, if for every X,Z -value, (x,z) that appears in R , we have

$$R[x,z,Y] = R[x,Y].$$

As we do for FDs, here also we usually omit the name g and write only $X \twoheadrightarrow Y$.

One can notice that the definition of an MVD uses the same condition as proposition 2.

Example_3 : For the relation R defined in example 1, one can notice that the MVD $CLASS \twoheadrightarrow BOOK$ holds in R , because for every $CLASS, STUDENT, PROFESSOR$ -value (c, t, p) that appears in R we have :

$$R[c,t,p,BOOK] = R[c,BOOK].$$

Proposition_4 : The relation $R(X,Y,Z)$ obeys the MVD $X \twoheadrightarrow Y$ iff R is decomposable into two parts $R[X,Y]$ and $R[X,Z]$.

This proposition is a direct consequence of the definition and proposition 2.

□

The decomposition given in example 1 is based upon proposition 4 applied to the MVD $CLASS \twoheadrightarrow BOOK$.

In the definition of MVD given here, as in Fagin's paper [FAG 76], we require the left and right sides of an MVD be disjoint. There are two reasons for that : first, a transitivity property does not always hold if the restriction is lifted, and second, in all practical situations a database designer will define this type of constraint with disjoint sets of attributes. In [BFH 77] the reader can find a complete study of MVD properties; nevertheless we shall recall the basic properties without proof.

3.2.2. Properties of MVDs

The properties of MVDs are very similar to FDs. In this subsection we discuss inference rules for MVDs, that is rules that deal with the implication of new MVDs from a given set of MVDs.

Let $R(U)$ be a relation defined over a set U of attributes, where X,Y,Z are disjoint sets of attributes contained in U .

M0 reflexivity : if $X \supseteq Y$ then $X \twoheadrightarrow Y$

M1 complementation : if $X \twoheadrightarrow Y$ then $X \twoheadrightarrow U - (X \cup Y)$

We have to note that this complementation rule has no equivalent for the FDs.

M2 augmentation : if $X \twoheadrightarrow Y$ then for all Z then $X \cup Z \twoheadrightarrow Y \cup Z$

M3 transitivity : if $X \twoheadrightarrow Y$ and $Y \twoheadrightarrow Z$ then $X \twoheadrightarrow Z$

M4 pseudo-transitivity : if $X \twoheadrightarrow Y$ and $Y, Z \twoheadrightarrow W$ then $X, Z \twoheadrightarrow W$

M5 additivity : if $X \twoheadrightarrow Y$ and $X \twoheadrightarrow Z$ then $X \twoheadrightarrow Y \cup Z$

M6 decomposition : if $X \twoheadrightarrow Y$ and $X \twoheadrightarrow Z$ then $X \twoheadrightarrow Y \cap Z$.

3.2.3. Decomposition of relation with MVDs

We have seen in section 3.1.3 that the decomposition of a relation R structured only by FDs can lead to some anomalies. Now we can repeat the decomposition process of the relation.

$R(\text{BOOK}, \text{STUDENT}, \text{CLASS}, \text{PROFESSOR})$ structured by the FDs

$\text{STUDENT} \twoheadrightarrow \text{CLASS}$, $\text{CLASS} \twoheadrightarrow \text{PROFESSOR}$ and the MVD $\text{CLASS} \twoheadrightarrow \text{BOOK}$.

It is easy to obtain the decomposition illustrated by figure 3.

Figure 3

We can see that in this decomposition (figure 3), we obtain a relation where the attributes CLASS and BOOK are directly related.

3.2.4. MVDs defined on a subset of attributes

Sometimes it is possible to interpret the concept of MVDs as an extension of the concept of FDs by saying that the MVD $X \twoheadrightarrow Y$ holds in R means that the X-value determines a unique set of Y-value. This interpretation can be misleading. This fact is more easily clarified by an example.

Example_4 : Consider the following relation R over the attributes BOOK, CLASS, STUDENT, PROFESSOR with a different semantic from example 1. Each CLASS has different STUDENTS and PROFESSORS but a STUDENT is only in one CLASS. The BOOK is determined by a PROFESSOR and a CLASS. Then we have the FDs : $\text{STUDENT} \twoheadrightarrow \text{CLASS}$ and $\text{CLASS}, \text{PROFESSOR} \twoheadrightarrow \text{BOOK}$.

An instance of the relational schema is given by the table below.

R :	BOOK	CLASS	STUDENT	PROFESSOR
	b ₁	math	peter	mike
	b ₁	math	john	mike
	b ₂	programming	jane	mike
	b ₂	math	peter	ronald
	b ₂	math	john	ronald
	b ₂	programming	james	mike

In the relation R given in example 4, the knowledge of CLASS-value determines the set of PROFESSOR-value, but is not true that $CLASS \twoheadrightarrow PROFESSOR$ because :

$R[\text{math}, \text{PROFESSOR}] = \{\text{mike}, \text{ronald}\}$

$R[b_1, \text{math}, \text{peter}, \text{PROFESSOR}] = \{\text{mike}\}.$

This is due to the fact that the validity of the MVD, $X \twoheadrightarrow Y$ in $R(X,Y,Z)$ depends not only on the values of X and Y, but also on the values of Z. It is possible that $X \twoheadrightarrow Y$ is not valid in $R(X,Y,Z)$, but that $X \twoheadrightarrow Y$ is valid in the projection $R[X,Y,Z']$ where Z' is a subset of Z.

For example, the constraint $CLASS \twoheadrightarrow PROFESSOR$ is only valid on $R[CLASS, STUDENT, PROFESSOR]$. Furthermore we can interpret this constraint as "all the students attending a class are taught by all instructors".

One can check that the MVD $CLASS \twoheadrightarrow STUDENT$ is valid in R ; then the relation R is structured by the FDs : $STUDENT \rightarrow CLASS$, $CLASS, PROFESSOR \rightarrow BOOK$ and the MVDs $CLASS \twoheadrightarrow STUDENT$ and $CLASS \twoheadrightarrow PROFESSOR$ which are only valid on the set of attributes CLASS, PROFESSOR, STUDENT.

To capture this new type of constraints, Fagin calls this case *embedded multivalued dependencies*. To denote an embedded MVD we need to specify precisely the sets of attributes which occur.

This concept of embedded multivalued dependencies is similar to the concept of first order hierarchical decomposition which has been previously defined in [DL 74] and which is more general than the concept of MVDs.

3.3. First order hierarchical decomposition

3.3.1. Definition

A *first order hierarchical decomposition* (abbr. FOHD) [DEL 77] is a sentence denoted $X : Y_1|Y_2|\dots|Y_k$ where $X, Y_1, Y_2, \dots, Y_k$ are disjoint sets of attributes.

A relation $R(X, Y_1, Y_2, \dots, Y_k, W)$ where $X, Y_1, Y_2, \dots, Y_k, W$ are disjoint sets of attributes, obeys the FOHD if for every X-value we have

$R[x, Y_1, Y_2, \dots, Y_k] = R[x, Y_1] \times R[x, Y_2] \times \dots \times R[x, Y_k].$

According to proposition 1, it is easy to prove that this condition is equivalent to the condition :

$$R[X, Y_1, Y_2, \dots, Y_k] = R[X, Y_1] * R[X, Y_2] * \dots * R[X, Y_k]$$

which expresses the decomposability of the projection R over $\{X, Y_1, \dots, Y_k\}$ into k projections.

The definition of FOHD given here is more general than the definition of decomposition that we have introduced in section 2.3.

In the FOHD, $X : Y_1|Y_2|\dots|Y_k$, X is called the *root segment*, $Y_1, Y_2, \dots, Y_k$ are called *segments* and the pair (X, Y_i) is called a *branch*. We shall say that a FOHD is a *full first order hierarchical decomposition* or a *generalized multivalued decomposition* (abbr. GMVD) if W is the empty set. We shall denote a GMVD as $X \twoheadrightarrow Y_1|Y_2|\dots|Y_k$. In the case where the root X is empty, this means that the projection of R over the set of attributes $\{Y_1, Y_2, \dots, Y_k\}$ is the cartesian product of the projections of R respectively on $Y_1, Y_2, \dots$, and Y_k .

For the relation R given in example 4, the FOHD

CLASS : STUDENT|PROFESSOR

holds. We have also seen that the MVD CLASS $\twoheadrightarrow$ STUDENT holds in R , thus proposition 4 can be applied to give the FOHD :

CLASS : PROFESSOR, BOOK|STUDENT.

As one can see, the definitions of FOHDs and MVDs are two facets of the same problem : under what conditions is a relation decomposable ? Effectively, each time that a constraint on a relation can be paraphrased by a sentence like "all the students attending a seminar are taught by all instructors", we have a constraint of the family of FOHDs, and this sentence refers to the condition of proposition 1. On the other hand, a sentence like "the type of seminar determines uniquely the set of books" can generate a constraint of the family of MVDs, and this sentence refers to the condition of proposition 2.

From the definition of FOHD and MVD, it is clear that we have the following relationships : a GMVD represents a set of MVDs, for example if $X \twoheadrightarrow Y|Z|U$ holds in $R(X, Y, Z, U)$ then we have $X \twoheadrightarrow Y$, $X \twoheadrightarrow Z$ and $X \twoheadrightarrow U$ and the converse. Nevertheless if $X \twoheadrightarrow Y$ holds in $R(X, Y, Z, U)$ we have $X \twoheadrightarrow Y|Z, U$. And if $X : Y|Z$ holds in $R(X, Y, Z, U)$ we cannot deduce any MVD.

3.3.2. Properties of FOHDs

In this section, we discuss inference properties for FOHDs from a given set of FOHDs. The properties are only reviewed here since they have been presented in [DEL 77]. Throughout this section, we assume that the relation R is defined over a given disjoint set of attributes $X, Y_1, Y_2, \dots, Y_k, W$.

- H0 permutation : if $X : Y_1|Y_2|\dots|Y_k$ holds in R and π a permutation of $1,2,\dots,k$,
 then $X : Y_{\pi(1)}|Y_{\pi(2)}|\dots|Y_{\pi(k)}$ holds in R.
- H1 addition of branches : if $X : Y_1|Y_2|\dots|Y_k$ holds in R then $X : Y_1 \cup Y_2|\dots|Y_k$
 holds also in R.
- H2 deletion of a branch : if $X : Y_1|Y_2|\dots|Y_k$ holds in R then $X : Y_2|\dots|Y_k$ holds
 in R.
- H3 projection of a branch : if $X : Y_1|Y_2|\dots|Y_k$ holds in R then $X : Y_1^i|Y_2|\dots|Y_k$ holds
 in R where Y_1^i is any subset of Y_1 .
- H4 root augmentation : if $X : Y_1|Y_2|\dots|Y_k$ holds in R then $X \cup Y_1^i : Y_1 - Y_1^i|Y_2|\dots|Y_k$
 holds in R, where Y_1^i is any subset of Y_1 .
- H5 projection of the root and decomposition of a branch :
 if $X : Y_1|Y_2 \cup Y_3|\dots|Y_k$ and $X \cup Y_1 : Y_2|Y_3$ holds in R then
 $X : Y_1|Y_2|Y_3|\dots|Y_k$ holds in R.
- H6 decomposition : if $X : Y_1|Y_2|\dots|Y_k$ and $X : Z_1|Z_2|\dots|Z_r$ hold in R with
 $Y_1 \cap Z_1 \neq \emptyset$ and $Y_1 \cap Z_2 \neq \emptyset$ then $X : Y_1 \cap Z_1|Y_1 \cap Z_2|Y_2|\dots|Y_k$ holds
 in R.

3.4. Axiomatization for FDs and MVDs

Armstrong [ARM 74] proved that the properties of reflexivity, augmentation and transitivity constitute a complete set of inference rules for FDs. The completeness theorem can be stated as follows : for every set F of FDs there exists a relation R such that the FDs which are valid in R is exactly F^+ .

This property is only valid for the FDs ; we now turn our attention to a more general problem. What could be a set of inference rules for a set F of FDs and a set M of MVDs ? In the previous section, we have seen properties for MVDs but these properties are not enough to define a complete axiomatization since certain combinations of FDs and MVDs imply additional FDs that cannot be derived by the use of the above rules. We need to introduce three additional properties :

FM1 : if $X \rightarrow Y$ then $X \twoheadrightarrow Y$

FM2 : if $X \twoheadrightarrow Y$ and $W \rightarrow Z$ where $Y \supseteq W$ with $Y \cap Z = \emptyset$ then $X \rightarrow Z$.

Berri and al [BFH 77] proved that the rules $\{F1,F2,F3,M1,M2,M3,FM1,FM2\}$ are a complete set of inference rules for the family of functional and multivalued dependencies. The completeness theorem can be stated as follows : for every set F of FDs and M of MVDs there exists a relation R such that the set of FDs and MVDs valid in R is exactly $(F \cup M)^+$.

The problem of determining a complete set of inference rules for FOHDs is an open research problem. It has been solved only in the special case of full FOHD.

4. DECOMPOSITION PROCESS USING DECOMPOSITION STRUCTURE

4.1. General analysis of the decomposition approach

The different approaches for the design of logical schemas presented in section 3 are based on the properties of FDs and MVDs. In all of these approaches, the process starts with a list of dependencies given by the designer. From a designer view point, it is not necessary to consider all the given dependencies. An irredundant cover is adequate for this purpose, since the remaining dependencies are derivable by the inference rules. In the previous section (section 3.4), we have remembered that we have to consider 8 basic inference rules.

Instead to look at the dependencies, there exists an alternative approach where the family of decompositions of a relation is directly studied. This approach presents the following advantages :

- to study the decomposition properties of a relation we need only 4 basic inference rules
 - the family of decompositions for the relation R can be ordered by an order relation in order to simplify the designer's task for finding a minimal cover. To achieve this goal, first we shall define a *-operation that generates from two decompositions a new one, and second to avoid the multiplicity of decompositions we introduce an order relation denoted $\subseteq$ and according to which the *-operation is isotonus. This property is very important when one is looking for a minimal cover
 - to consider only decompositions which ensures the complete joinability of data.
- A similar idea has been introduced by Rissanen [RIS 77] with the notion of independent component and by Zaniolo and Melkanoff [ZAM 77] with the concept of complete relatibility.

4.2. The decomposition of a relation

Let R be a relation over U. A *decomposition* of R is a pair (X,Y) of subsets of U with $X \cup Y = U$ such that

$$R = R[X] * R[Y]$$

A decomposition can be considered as a GMVD : $X \cap Y \twoheadrightarrow X-Y | Y-X$ with two branches.

By similarities the sets X-Y and Y-X will be called the segments of the decomposition and $X \cap Y$ the root segment.

Using the definition, Armstrong and Delobel [AD 77] proved that the following inference rules (or axioms) for decompositions are complete.

D1 empty decomposition : $(\emptyset, U)$ holds in R

D2 symetry : if (X,Y) holds in R then (Y,X)

D3 augmentation : if (X,Y) holds in R and $Z \subseteq U$ then $(X \cup Z, Y \cup Z)$

D4 intersection : if (X,Y) and (Z,W) hold in R with $Z \cap W = Y$ then $(X \cap Z, W)$.

The completeness theorem can be stated as follows : for every set D of decompositions there exists a relation R such that the decompositions which are valid in R are exactly D^+ . The very important point with this axiomatization is that it captures directly the decomposition structure of the relation R .

We introduced now some additional rules that we have found to be useful for the manipulation of decomposition :

D5 absorption : if $X \subseteq U$ then (X,U) holds in R

D6 : if (X,Y) and (Z,W) hold in R with $Z \cap W \subseteq Y$ then $(X \cap (Y \cup Z), Y \cup W)$ holds also in R .

The proofs are given in [AD 77] and they are direct consequence of properties D1-D4.

4.3. Analysis of an initial set of decompositions

To study the decomposition structure of the relation R we need at the initial stage of the process to have a set of decompositions. This can be done only by the designer in interpreting, from his understanding of the case at hand, the dependencies in terms of decompositions. The following example will explain the initial stage.

Example_5 : Let R be a relation over the attributes Student, Class, Professor, Book and Hour. On an intuitive background the relation R expresses a time table. A student is only in one class. A class has different students and professors. A professor at one hour is teaching to only one class. A class uses different books. Then we can consider that the relation R is structured by the following set F of dependencies (FDs and MVDs), $F = \{f1, f2, f3, f4\}$

- f1 $S \rightarrow C$
- f2 $C \twoheadrightarrow S$
- f3 $C \twoheadrightarrow B$
- f4 $H, P \rightarrow C$

By application of proposition 3 successively to f1-f4, we derive the following decompositions d1-d4 :

- d1 $S \twoheadrightarrow C | B, P, H$
- d2 $C \twoheadrightarrow S | B, P, H$
- d3 $C \twoheadrightarrow B | S, H, P$
- d4 $H, P \twoheadrightarrow C | S, B$

It is from d1, d2, d3 and d4 that the designer will study the various decompositions of the relation R .

We now present an effective methodology for the design of relational schema using the decomposition approach. The different stages of this methodology are :

- starting from an initial set of decompositions, derive all the possible decompositions
- construct a minimal cover for the initial set of decompositions
- select from the minimal cover a decomposition that ensures complete joinability of data.

4.4. An algorithm for finding the maximal decompositions

4.4.1. The $\star$ -operation

The reference rule D4 for decomposition can be generalized into two directions by deleting the condition and by considering a decomposition in more than two branches. This can be done by considering the proposition 5 applied to GMVDs.

Proposition_5 : If $Y \twoheadrightarrow B_1|B_2|\dots|B_J$ and $Z \twoheadrightarrow A_1|A_2|\dots|A_I$ hold in R then :

- (1) $Y \cup (B_k \cap Z) \twoheadrightarrow B_1|B_2|\dots|B_{k-1}|B_k \cap A_1|B_k \cap A_2|\dots|B_k \cap A_I|B_{k+1}|\dots|B_J$,
with $k \leq J$, holds also in R.

Proof :

By setting $B_k \cap Z = C$, $B_k \cap A_1 = C_1$, ..., $B_k \cap A_I = C_I$ then we have to prove that :

- (2) $Y \cup C \twoheadrightarrow B_1|B_2|\dots|B_{k-1}|C_1|C_2|\dots|C_I|B_{k+1}|\dots|B_J$

Let

- (3) $(y, c, b_1, b_2, \dots, b_{k-1}, c_1, c_2, \dots, c_I, b_{k+1}, \dots, b_J)$

- (4) $(y, c, b'_1, b'_2, \dots, b'_{k-1}, c'_1, c'_2, \dots, c'_I, b'_{k+1}, \dots, b'_J)$

be two tuples into R.

$Y \twoheadrightarrow B_1|B_2|\dots|B_J$ implies $Y \twoheadrightarrow B_k|B_1, \dots, B_{k-1}, B_{k+1}, \dots, B_J$ according to H1 ; then we can infer the existence of the tuples :

- (5) $(y, c, b'_1, b'_2, \dots, b'_{k-1}, c_1, c_2, \dots, c_I, b'_{k+1}, \dots, b'_J)$

- (6) $(y, c, b_1, b_2, \dots, b_{k-1}, c'_1, c'_2, \dots, c'_I, b_{k+1}, \dots, b_J)$

$Z \twoheadrightarrow A_1|A_2|\dots|A_I$ implies $Z \twoheadrightarrow A_1|A_2, \dots, A_I$ according to H1 ; then we can infer from (3) and (6) the tuple :

- (7) $(y, c, b_1, b_2, \dots, b_{k-1}, c'_1, c_2, \dots, c_I, b_{k+1}, \dots, b_J)$

and from (4) and (5) the tuple :

- (8) $(y, c, b'_1, b'_2, \dots, b'_{k-1}, c_1, c'_2, \dots, c'_I, b'_{k+1}, \dots, b'_J)$;

the tuples (3), (4), (7) and (8) imply the existence of the GMVD

$$(9) \quad Y \cup C \rightarrow C_1 | B_1, B_2, \dots, B_{k-1}, C_2, \dots, C_I, B_{k+1}, \dots, B_J.$$

It is easy to prove that we have also by replacing successively C_1 by C_2 , until C_1 by C_I :

$$(10) \quad \begin{aligned} Y \cup C &\rightarrow C_2 | B_1, B_2, \dots, B_{k-1}, C_1, C_3, \dots, C_I, B_{k+1}, \dots, B_J \\ &\vdots \\ Y \cup C &\rightarrow C_I | B_1, B_2, \dots, B_{k-1}, C_1, C_2, \dots, C_{I-1}, B_{k+1}, \dots, B_J. \end{aligned}$$

From GMVDs (9) and (10) we can derive :

$$(11) \quad Y \cup C \rightarrow B_1, B_2, \dots, B_{k-1} | C_1 | C_2 | \dots | C_I | B_{k+1}, \dots, B_J$$

according to inference rule H6. But the two branches $B_1, B_2, \dots, B_{k-1}$ and $B_{k+1}, \dots, B_J$ in (11) can be decomposed since $Y \rightarrow B_1 | B_2 | \dots | B_J$; then we get finally

$$Y \cup C \rightarrow B_1 | B_2 | \dots | B_{k-1} | C_1 | C_2 | \dots | C_I | B_{k+1} | \dots | B_J$$

which is equal to (2).

□

We shall denote by $\underset{k}{*}$ the operation :

$$\begin{aligned} (Y \rightarrow B_1 | B_2 | \dots | B_J) \underset{k}{*} (Z \rightarrow A_1 | A_2 | \dots | A_I) = \\ (Y \cup (B_k \cap Z) \rightarrow B_k \cap A_1 | \dots | B_k \cap A_I | B_1 | B_2 | \dots | B_{k-1} | B_{k+1} | \dots | B_J). \end{aligned}$$

The $*$ -operation can be considered as an application from $G \times G$ to G , such the image by the $*$ -operation of two GMVDs g_1 and g_2 given with respectively i and j branches is a GMVD with $i+j-1$ branches. This operation looks like the consensus operation in boolean algebra defined in [KP 77].

Remark : Let g_1 and g_2 be GMVDs and $g = g_1 \underset{k}{*} g_2$; then the root of g contains at least one of the roots of g_1 or g_2 .

4.4.2. Definition of an order relation

We shall define an order relation, denoted $\subseteq$, on the decompositions of a relation as follows :

$$\begin{aligned} (Y \rightarrow B_1 | B_2 | \dots | B_J) \subseteq (Z \rightarrow A_1 | A_2 | \dots | A_I) \iff \\ (Y \subseteq Z) \wedge (\forall i = 1, 2, \dots, I, \exists j \in \{1, 2, \dots, J\} : B_j \cup Y \subseteq A_i) \end{aligned}$$

Proposition 6 : The relation $\subseteq$ is a partial order.

The proof is given in appendix.

Examples : $A, D \twoheadrightarrow B|C, E \supseteq A \twoheadrightarrow B, D|C|E$
 $A, D \twoheadrightarrow B|C, E \supseteq A \twoheadrightarrow B|D|C, E.$

Remarks :

- (1) if an GMVD g' is derivable from g by addition of branches (inference rule H1) then $g' \supseteq g$.
- (2) if an GMVD g' is derivable from g by augmentation (inference rule D3 or H4) then $g' \supseteq g$.

These two remarks show that to obtain the closure of a given set of decompositions it is not necessary to consider the addition of branches and the augmentation inference rules. We shall prove in the next proposition an important property that relates the $*$ -operation and the order relation $\supseteq$.

Proposition 7 : The $*$ -operation is isotonus according to $\supseteq$.
 The proof is given in appendix.

□

This property means that if $(Y \twoheadrightarrow B_1|B_2|\dots|B_J) \supseteq (Z \twoheadrightarrow A_1|A_2|\dots|A_I)$ then
 $(X \twoheadrightarrow C_1|C_2|\dots|C_K) * (Y \twoheadrightarrow B_1|B_2|\dots|B_J) \supseteq (X \twoheadrightarrow C_1|C_2|\dots|C_K) * (Z \twoheadrightarrow A_1|A_2|\dots|A_I).$

4.4.3. Maximal decompositions algorithm

The determination of a maximal covering of decompositions is greatly simplified according to proposition 7 because we have only to consider the maximal decompositions according to the order relation $\supseteq$. This will be illustrated on an example.

Let G be an initial set of decompositions as in our example 5, $G = \{d1, d2, d3, d4\}$

d1 $S \twoheadrightarrow C|B, P, H$
 d2 $C \twoheadrightarrow S|B, H, P$
 d3 $C \twoheadrightarrow B|S, H, P$
 d4 $H, P \twoheadrightarrow C|S, B$

From d2 and d3, we derive according to proposition 5

d5 $C \twoheadrightarrow S|B|HP.$

As $d2 \supseteq d5$ and $d3 \supseteq d5$ we can delete d2 and d3 from G and we introduce d5 in G . At the end of the process, we have only the maximal decompositions $MAX(G^*)$. In the present example, we have $MAX(G^*) = \{d5, d6, d7\}$

d5 $C \twoheadrightarrow S|B|H, P$
 d6 $S \twoheadrightarrow C|B|H, P$
 d7 $H, P \twoheadrightarrow C|S|B.$

Now we have only to search an irredundant covering among the maximal decompositions. In the present case it is $MAX(G^*)$.

The designer has the choice between three different decompositions d5, d6 and d7 ; we have to choose the ones which satisfy a special condition called "complete joinability of data".

4.5. Complete joinability of data

The notion of normal forms introduced by CODD provide a guideline to study the decomposition of a relation.

Many authors have modified the definition of the third normal form by looking at the type of anomalies in a relation, so we have Boyce, Codd, Kent, Irreducible and fourth normal forms. We propose a more general criterion for a decomposition, the *complete joinability of data*. We shall say that a decomposition has the complete joinability of data, intuitively, if this decomposition preserves all the joins.

Definition : Let F be a given set of dependencies (FDs and MVDs) for a relation R and D the initial set of decompositions derived from F as explained in subsection 4.4. Let $Y \twoheadrightarrow A_1 | A_2 | \dots | A_I$ be a generalized decomposition of R belonging to D ; then we consider

$F_{Y \cup A_k}$ $k = 1, 2, \dots, I$ as the set of dependencies which hold over the attribute set $Y \cup A_k$. To each $F_{Y \cup A_k}$ $k = 1, 2, \dots, I$, we associate $D_{Y \cup A_k}$ which is the set of decompositions derived from $F_{Y \cup A_k}$.

We shall say that the generalized decomposition $Y \twoheadrightarrow A_1 | A_2 | \dots | A_I$ has the complete joinability property iff :

$$\left(\bigcup_{k=1}^I D_{Y \cup A_k} \right)^+ = D^+$$

Example : If we consider decomposition d6 $S \twoheadrightarrow C | B | H, P$, we have

$$F_{S,C} = \{S \rightarrow C, C \twoheadrightarrow S\}$$

$$F_{S,B} = \emptyset$$

$$F_{S,H,P} = \emptyset$$

and

$$D_{S,C} = \{S \twoheadrightarrow C | B, P, H, C \twoheadrightarrow S | B, H, P\}$$

$$D_{S,B} = \emptyset$$

$$D_{S,H,P} = \emptyset.$$

One can check $(D_{S,C} \cup D_{S,B} \cup D_{S,H,P})^+ \subset D^+$, then d6 has not the complete joinability property of data.

Now, if we consider $d5 \ C \twoheadrightarrow S|B|H,P$, we have

$$F_{S,C} = \{S \rightarrow C, C \twoheadrightarrow S\}$$

$$F_{C,B} = \{C \twoheadrightarrow B\}$$

$$F_{P,H,C} = \{P, H \rightarrow C\}$$

$$\text{and } D_{S,C} = \{d1, d2\}$$

$$D_{C,B} = \{d3\}$$

$$D_{P,H,C} = \{d4\}$$

then as $D_{S,C} \cup D_{C,B} \cup D_{P,H,C} = D$, the decomposition $d5$ has the complete joinability of data.

5. CONCLUSIONS

The main objective of this paper was to establish a rigorous foundation for the design of conceptual schemas for a database system. Because of its formal nature the relational data model was used as a framework.

In a first part, we introduced an analysis of the state of the art, by considering the different kind of dependencies : functional dependencies, multivalued dependencies and hierarchical dependencies. We also recognized that dependencies and **irredundant** covers supply a power tool for the analysis and design of relational schemas.

In a second part, we proposed the decomposition approach where the initial relations are broken into subcomponents. This decomposition approach is based upon the concept of generalized multivalued dependency. The formal properties of these dependencies yields a number of new concepts, such the $*$ -operation which offers the possibility of deriving new decompositions from an initial set of decompositions ; a technique is proposed for finding the maximal decompositions among the whole set of decompositions.

APPENDIX

Proposition 6 : The relation $\subseteq$ is a partial order.

1. $\subseteq$ is reflexive.
2. Let us show that $\subseteq$ is transitive, that is :
 $(X \rightarrow C_1|C_2|\dots|C_K) \subseteq (Y \rightarrow B_1|B_2|\dots|B_J)$ and
 $(Y \rightarrow B_1|B_2|\dots|B_J) \subseteq (Z \rightarrow A_1|A_2|\dots|A_I)$
 $\Rightarrow (X \rightarrow C_1|C_2|\dots|C_K) \subseteq (Z \rightarrow A_1|A_2|\dots|A_I)$.

Indeed $X \subseteq Y$ and $Y \subseteq Z \Rightarrow X \subseteq Z$

$$\forall i = 1, 2, \dots, I, \exists j \in \{1, 2, \dots, J\} : B_j \cup Y \supseteq A_i ; \forall k \in \{1, 2, \dots, K\} : \\ C_k \cup X \supseteq B_j.$$

Hence : $\forall i = 1, 2, \dots, I, \exists k \in \{1, 2, \dots, K\} : C_k \cup X \supseteq B_j \cup Y \supseteq A_i$.

3. $\subseteq$ is antisymmetric because

$Y \rightarrow B_1|B_2|\dots|B_J \subseteq Z \rightarrow A_1|A_2|\dots|A_I$ and $Z \rightarrow A_1|A_2|\dots|A_I \subseteq Y \rightarrow B_1|B_2|\dots|B_J$
 give, thanks to the antisymmetry of $\subseteq$: $Y = Z$ and

$$\forall i = 1, 2, \dots, I, \exists j \in \{1, 2, \dots, J\} : B_j \cup Y \supseteq A_i$$

or $B_j \supseteq A_i$ because A_i and Y are disjoint,

that is by symmetry : $\forall i = 1, 2, \dots, I, \exists j \in \{1, 2, \dots, J\} : B_j = A_i$.

□

Proposition 7 : The $*$ -operation is isotonus according to $\subseteq$.

Let us show that

$$(Y \rightarrow B_1|B_2|\dots|B_J) \subseteq (Z \rightarrow A_1|A_2|\dots|A_I) \Rightarrow (X \rightarrow C_1|C_2|\dots|C_K) \overset{*}{\underset{k}{\subseteq}} \\ (Y \rightarrow B_1|B_2|\dots|B_J) \\ (X \rightarrow C_1|C_2|\dots|C_K) \overset{*}{\underset{k}{\subseteq}} \\ (Z \rightarrow A_1|A_2|\dots|A_I).$$

At first $\forall k = 1, 2, \dots, K$, isotonus $\cap$ and $\cup$ relatively to $\subseteq$ give :

$$Y \subseteq Z \Rightarrow X \cup (C_k \cap Y) \supseteq X \cup (C_k \cap Z).$$

Secondly $\forall i = 1, 2, \dots, I, \exists j \in \{1, 2, \dots, J\} : B_j \cup Y \supseteq A_i$,

hence $\forall k = 1, 2, \dots, K : C_k \cap (B_j \cup Y) \supseteq C_k \cap A_i$,

a fortiori : $(C_k \cap B_j) \cup X \cup (C_k \cap Y) \supseteq C_k \cap A_i$.

REFERENCES

- [BOI 69] J. BOITTIAUX, Etude mathématique d'un ensemble de notions. Contrat DGRST 67.01.015 (1969).
- [COD 70] E.F. CODD, A relational model of data for large shared data banks, CACM, June 1970, pp. 377-387.
- [COD 71] E.F. CODD, Further normalization of the database relational model, Courant Computer Science Symposium 6, Database Systems, Prentice Hall, N.Y. (May 1971), pp. 65-98.
- [DEL 71] C. DELOBEL, Aspects théoriques sur la structure de l'information dans une base de données, RIRO, B.3, 1971, pp. 37-64.
- [DC 73] C. DELOBEL & R.G. CASEY, Decomposition of a data base and the theory of boolean switching functions. IBM Journal of Research and Development, Vol. 17, n° 5, 1973.
- [DL 74] C. DELOBEL & M. LEONARD, The decomposition process in a relational model. International Workshop on Data Structures, IRIA, Namur (Belgique), May 1974.
- [ARM 74] W.W. ARMSTRONG, Dependency structures of database relationships. Proc. IFIP 1974, pp. 580-583, North Holland, 1974.
- [WW 75] C.P. WANG & H.H. WEDEKIND, Segment synthesis in logical data base design. IBM Journal of Research and Development, January 1975.
- [SS 75] H.A. SCHMID & J.R. SWENSON, On the semantics of the relational data model. Proceedings ACM SIGMOD, W.K. King (Ed.), San José, Calif., May 1975, pp. 211-223.
- [BER 75] P.A. BERNSTEIN, Normalization and functional dependencies in the relational data base model. Ph.D Thesis, Departement of Computer Science, University of Toronto, 1975.
- [FAG 76] R. FAGIN, Multivalued dependencies and a new normal form for relational data bases. TODS 23 (Sept. 1977), pp. 262-278.
- [RIS 76] J. RISSANEN, Independent components of relation. IBM Research Laboratory, San José, October 1976.

- [LR 76] M. LEONARD & F. REYNAUD, Existence du consensus et caractérisation des couvertures et bases irrédundantes d'une fonction $\sum_i \mu_i A_i^1$.
Rapport de recherche, September 1976, Computing Laboratory, University of Grenoble (submitted to "Discrete Mathematics").
- [ZAN 76] C. ZANIOLO, Analysis and design of relational schemata for data base systems. Ph.D Dissertation, UCLA, 1976.
- [BER 77] P.A. BERNSTEIN, Synthesizing third normal form relations from functional dependencies. ACM TODS, January 1977.
- [BFH 77] C. BEERI, R. FAGIN & J. HOWARD, A complete axiomatization for functional and multivalued dependencies in database relations, Proc. 1977 ACM SIGMOD, D.C.P. Smith (ed.), Toronto, pp. 47-61.
- [RLP 77] F. REYNAUD, M. LEONARD & E. PICHAT, Calculation of the prime implications and their irredundant covers in data bases. Research Report #3, June 1977, CNAM.
- [KP 77] A. KAUFMANN & E. PICHAT, Méthodes mathématiques non numériques et leurs algorithmes. Tome I : algorithmes de recherche des éléments maximaux. Masson, Paris, 1977.
- [OSB 77] S. OSBORN, Normal forms for relational data bases. Ph.D Thesis, University of Waterloo, November 1977.
- [FLO 77] A. FLORY, Un modèle et une méthode pour la conception logique d'une base de données. Thèse d'Etat, Université de Lyon, France, Novembre 1977.
- [AD 77] W. ARMSTRONG, C. DELOBEL, Decompositions and Functional dependencies in Relations. Département d'Informatique et Recherche Opérationnelle, Université de Montréal, October 1977 (submit to ACM-TODS).
- [DEL 78] C. DELOBEL, Normalization and hierarchical dependencies in the relational data model (to appear in ACM-TODS).
- [ZAM 77] C. ZANIOLO, M. MELKANOFF, Relational schemas for data base systems. Technical Report, UCLA, November 1977.