Data base mappings-Part 1 : theory

François Bancilhon, N. Spyratos

INRIA

Institut National
de Recherche
en Informatique
et en Automatique

Domaine de Voluceau
Rocquencourt
B.P.105
78153 Le Chesnay Cedex
France
Tél. 954 90 20

Rapports de Recherche

N°62

fait le 30-04-81

DATA BASE MAPPINGS
PART I : THEORY

Paru le 28.04.81

Note 319

François BANCILHON
Nicolas SPYRATOS

Avril 1981

DATA BASE MAPPINGS

PART I : THEORY

François Bancilhon - Nicolas Spyratos
INRIA
Domaine de Voluceau
B.P. 105
78153 Le Chesnay - France

Abstract

Based on simple observations of the organization and functioning of data base systems, we give formal definitions for data base mappings. Then, we define a natural structure on the set of data base mappings and we study the notion of independence. In part II of this work (appearing as a separate INRIA research report n° 63) we apply our theory of data base mappings to two important areas of data base systems : data base decomposition and view updating.

Résumé

Nous donnons une définition formelle des opérateurs relationnels, basée sur l'observation de l'organisation et du fonctionnement des bases de données. Puis nous définissons une structure naturelle de l'ensemble des opérateurs relationnels et nous étudions la notion d'indépendence.
Dans la 2ème partie de ce travail (qui constitue un 2ème rapport INRIA n° 63) nous appliquons les résultats de notre théorie à deux domaines importants des Bases de données : la décomposition et les mises à jour dans les vues.

INTRODUCTION

A data base is an amount of information about facts of the real world. This information is coded and stored on memory devices as a set of data. We can view the data base as an object, or a variable, with a name, say BASE, and a state, or value at time t, say $BASE_t$. The value $BASE_t$ is the set of data in the data base at time t.

The facts of the real world represented in a data base, usually follow certain rules. These can be

- physical rules, for example, "every human being is either male or female"
- human rules, for example, "salaries never decrease"

The translation of these rules in the data base context, gives rise to integrity constraints, that is, rules that the data must satisfy in order to represent accurately the real world. Therefore, the integrity constraints define a set of possible values of the data base, that is, a state space for the variable BASE.

Let us take a closer look at some of the usual ways in which a data base is used :

Querying : A query is a mapping which associates to each value of the state space the answer (for that value)

Updating : An update is a mapping which associates to each value of the state space a new value of the state space (the modified data base value).

View definition : The view is a new data base whose state space is defined by a mapping on the state space of the original data base.

Distribution : Data base distribution sets up a mapping which associates to each value of the state space the set of values of the local data bases.

We shall call a data base mapping any mapping from a data base state space into another. We have just seen that such mappings appear in a number of different situations. We feel therefore justified in studying the set of data base mappings from a general point of view. We would like to point out that the notion of a data base mapping is not related to any specific data model. However, in order to avoid excessive formalisms, we have chosen a specific model, namely, the relational model, as the context of our discussion. An additional reason for this choice is the number of available theoretical results. We would like to use these results in order to illustrate our theory.

The paper is organized as follows. In Section 2, we recall briefly the basic definitions and notation from the relational model. In Section 3, we define formally a data base mapping and a lattice structure on the set of these mappings as follows. A data base mapping is seen as an "information carrier". Given two mappings carrying comparable information, we say that the one that carries more information dominates the other. Dominance being a partial order relation, we define a maximal and a minimal element in the set of data base mappings. Finally, we study the suppremum and the infimum of finite sets of mappings. In Section 4, we study the notion of independence of data base mappings. Independence is defined in a way similar to the one used for random variables.

We believe that a theory of data base mappings in itself is of little interest (in fact the mathematical tools used are relatively simple). It becomes important if it can solve concrete data base problems. In a companion paper , we apply our theory to two such problems, namely, data base decomposition and view updating. The solution to these problems is a (a posteriori) justification for a theory of data base mappings.

THE RELATIONAL MODEL

Let $U = \{A_1, A_2, \ldots, A_n\}$ be a set of names called attributes. With each attribute A_i we associate a set of values V_i (these sets of values need not be distinct). Any mapping on U which assigns to each attribute A_i a value in V_i is called a tuple over U. A set of tuples over U is called a relation over U. It is not hard to see how the tuples of a relation can be represented as lines in a table where each column is associated one-to-one with an attribute. The notation

$$R(A_1:V_1;A_2:V_2;\ldots;A_n:V_n)$$

is referred to as a relation schema. It is used to represent a variable R whose values are (finite) relations over A. We shall call R a relational variable. An example of a relation schema is the following

PAY(EMPLOYEE : NAME ; SALARY : INTEGER)

PAY is the relational variable, EMPLOYEE and SALARY are the attributes and NAME and INTEGER are the associated sets of values. For example, NAME could be a set of character strings and INTEGER a set of integers. Whenever the attributes imply clearly the corresponding sets of values, we can simplify further the notation by dropping the sets of values. Thus, the previous relation schema can be written as follows

PAY(EMPLOYEE,SALARY)

We shall use the symbol R_t to denote the value of the relational variable R at time t. Also, we shall use the symbol x.A to denote the restriction of a tuple x over U to a subset A of U.

A data base variable D is a set of relational variables. The value of D at time t, denoted D_t, is the set of values of its relational variables at time t. A data base schema consists of

1 - a data base variable D

2 - a predicate C on the values of D expressing the integrity contraints

It will be denoted by $<D|C>$. The data base state space, associated to $<D|C>$, and denoted $S<D|C>$, is defined as follows

$$S<D|C> = \{D_t | C(D_t) = \text{true}\}$$

The symbol λ will stand for the empty predicate, i.e., $\lambda(D_t)$ is always true. Therefore, $S<D|\lambda>$ is simply the set of all possible values of D. It follows that

$$S<D|C> \subset S<D|\lambda> \quad \forall\ C$$

Let us now define the relational operations that we shall be using in our examples.

Let R(U) be a relation schema. The projection of a value R_t over a subset A of U, denoted $R_t[A]$, is defined as follows

$$R_t[A] = \{x.A | x \in R_t\}$$

Let R1(U1), R2(U2) be two schemas such that $U1 \cap U2 = \emptyset$.

Let $A1 \in U1$, $A2 \in U2$ be two attributes and V1, V2 their associated sets of values. Let θ be a binary relation on V1 and V2, i.e., $\theta \subset V1 \times V2$ (usually, θ is one of the following : $=, \neq, \geq, \leq, >, <$).

The θ-join of $R1_t$ and $R2_t$ over A1 and A2, denoted $R1_t[A1\theta A2]R2_t$, is a relation over $U1 \cup U2$, defined by

$$R1_t[A1\theta A2]R2_t = \{(x1,x2) | x1 \in R1_t,\ x2 \in R2_t,\ x1.A1\ \theta\ x2.A2\}$$

An equijoin is a θ-join where θ is the equality.

The natural join is a θ-join when $U1 \cap U2 \neq \emptyset$. It is denoted by $R1_t * R2_t$ and it is defined as follows

$$R1_t * R2_t = \{x | x.U1 \in R1_t,\ x.U2 \in R2_t\}$$

where x denotes tuples over U1 ∪ U2.

Given a relation schema R(U), an elementary condition is an expression of the form $(A_i \theta a)$, where $A_i \in U$, $\theta \subset V_i \times V_i$ and $a \in V_i$. For example, given the relation schema

R(STUD,COURSE,GRADE,PROF)

the expressions (GRADE > 70) and (PROF ≠ John) are elementary conditions. A selection condition C on R(U) is a Boolean expression of elementary conditions (i.e., a predicate on tuples of R_t). The selection of R_t with respect to C, denoted $R_t|C$, is defined by

$$(R_t|C) = \{x | x \in R_t,\ C(x) = \text{true}\}$$

Finally, we define the two kind of integrity constraints that we shall use in this paper :

Let R(U) be a relation schema. A functional dependency is syntactically denoted $X \rightarrow Y$, where X and Y are (not necessarily disjoint) subsets of U. The semantics is as follows. We say that R_t satisfies $X \rightarrow Y$ if

$$\forall\ a, b \in R_t \quad a.X = b.X \Rightarrow a.Y = b.Y$$

A multivalued dependency is denoted syntactically by $X \twoheadrightarrow Y$. The semantics is as follows. We say that R_t satisfies $X \twoheadrightarrow Y$, iff $(R_t|X=x)[Y] = (R_t|X=x \text{ and } Z=z)[Y]$, ∀ x, ∀ z such that $xz \in R_t[XZ]$.

Given a relation schema U we shall use the letters X, Y, Z,... to denote subsets of U. We shall denote unions of subsets of U by concatenation of their symbols, for example, XYZ stands for X ∪ Y ∪ Z. However, we shall use this notation only when the subsets are disjoint.

DATA BASE MAPPINGS : DEFINITION AND STRUCTURE

A data base mapping transforms a given data base into a new one. Therefore, in order to define such a mapping, we need

- a data base schema $<D|C>$
- a data base variable D'
- a mapping f that associates to each value in $S<D|C>$, a value in $S<D'|\lambda>$

Definition 1 - Given a data base schema $<D|C>$ and a data base variable D', any mapping

$$f : S<D|C> \rightarrow S<D'|\lambda>$$

is called a data base mapping. □

The image $f(S<D|C>)$ being a subset of $S<D'|\lambda>$, it defines a state space on D'. This state space can also be defined by a predicate C' such that :

$$f(S<D|C>) = S<D'|C'>$$

Whenever there is no confusion possible, we shall use the symbol S to denote the set $S<D|C>$. Also, we shall use the symbol MAP to denote the set of all mappings on S. We shall not talk here about the definition of the set MAP nor about the computability of its mappings. Questions of this nature are treated elsewhere[2].

Let us now turn into the question of structuring the set MAP. Of course, we would like to define a structure with an intuitive meaning to it. So, let us see first what a data base mapping can do for us. In the introduction, we discussed several situations (querying, view definition, etc..) where a data base mapping is used to extract part of the information in the data base in the desired format. Therefore,

we can consider a data base mapping as a "communication channel" between the data base and the user. The user turns his channel on whenever he needs information from the data base. The quantity and format of the information that he gets depend on the "channel specifications". Therefore, we should be able to associate with each data base mapping a measure of the information it can carry. Rather than defining a quantitative measure for each mapping we shall try to

(i) compare data base mappings based on their "capacity", i.e., the amount of information they can carry

(ii) compose them somehow so that their capacity is added up.

Definition 2 - Let $f,g \in MAP$. We say that f dominates g, denoted $f \geq g$, iff

$$f(D_{t1}) = f(D_{t2}) \Rightarrow g(D_{t1}) = g(D_{t2}) \quad \forall D_{t1}, D_{t2} \in S \qquad \square$$

Intuitively, this means that if two data base values cannot be distinguished by f then they cannot be distinguished by g either. Let us note that this comparison between f and g depends only on the common domain of the two mappings, namely S, and not on their co-domains which are different in general.

Example 1

Relation schemas : R1(EMPL,SAL,DEPT), R2(DEPT,MGR,#EMPL)
R3(EMPL,DEPT,MGR), R4(EMPL,DEPT,#EMPL)

Integrity constraints :

c1 : EMPL → SAL,DEPT in R1
c2 : DEPT → MGR,#EMPL in R2
c3 : The attribute #EMPL in R2 is the number of employees in the corresponding department computed from R2
C : c1 ∧ c2 ∧ c3

Data base variables : $D = \{R1,R2\}$, $D1 = \{R3\}$, $D2 = \{R4\}$

Data base mappings :

$$\left.\begin{array}{l} f1 : S<D|C> \rightarrow S<D1|\lambda> \text{ such that} \\ \qquad f1(D_t) = (R1_t * R2_t)[EMPL,DEPT,MGR], \quad \forall\, D_t \in S<D|C> \\ f2 : S<D|C> \rightarrow S<D2|\lambda> \text{ such that} \\ \qquad f2(D_t) = (R1_t * R2_t)[EMPL,DEPT,\#EMPL], \quad \forall\, D_t \in S<D|C> \end{array}\right\} \quad (1)$$

It follows from Definition 2 that $f1 \geq f2$ □

From now on, when defining data base mappings, we shall use a simplified notation as follows. Instead of the definitions (1) in the previous example, we shall write

$$\left.\begin{array}{l} f1 = (R1 * R2)[EMPL,DEPT,MGR] \\ f2 = (R1 * R2)[EMPL,DEPT,\#EMPL] \end{array}\right\} \quad (1')$$

An immediate consequence of Definition 2 is the following theorem.

Theorem 1 - Let $f,g \in MAP$ such that $f \geq g$. Then, there exists $h \in MAP$ such that $g = hf$ □

(Juxtaposition of mappings denotes composition throughout this paper). It should be emphasized that this theorem guarantees existence but not computability of the mapping h.

Let us see some more examples to further illustrate Definition 2.

Example 2

Relation schemas : R(STUD,COURSE,GRADE,PROF), R1(STUD,COURSE,GRADE)
R2(STUD,COURSE)

Integrity constraints : $C = \lambda$

Data base variables : $D = \{R\}$, $D1 = \{R1\}$, $D2 = \{R2\}$

Mappings : $f1 = R[STUD,COURSE,GRADE]$, $f2 = R[STUD,COURSE]$

Applying Definition 2 we obtain $f1 \geq f2$ □

Example 3

Relation schemas : R(NAME,ADDR,SAL), R1(NAME,ADDR,SAL), R2(NAME,ADDR,SAL)

Integrity constraints : C : NAME → ADDR,SAL in R

Data base variables : D = {R}, D1 = {R1}, D2 = {R2}

Mappings : f1 = (R|SAL < 15000)[NAME,ADDR,SAL]

f2 = (R|1000 < SAL < 12000)[NAME,ADDR,SAL]

Applying Definition 2 we obtain : f1 ≥ f2 □

According to Definition 2, the more data base values f can distinguish, the more information it can carry. Therefore, it is natural to consider the partition on S induced by f. We shall denote this partition by S/f. Each member of S/f is a set of data base values that cannot be distinguished by f. So the smaller these members are the more information f can carry. We shall denote by S/f ≥ S/g the fact that the partition S/f is a refinement of S/g (i.e., each member of S/f is contained in some member of S/g). It follows that : f ≥ g <=> S/f ≥ S/g. Similarly, let $\equiv_f$ denote the equivalence relation induced on S by f (i.e., two data base values are equivalent iff they have the same image under f). It follows from Definition 2 that : f ≥ g <=> $\equiv_f \subset \equiv_g$. We state these results formally in the following theorem.

Theorem 2 - Let f,g ∈ MAP. Then

f ≥ g <=> S/f ≥ S/g <=> $\equiv_f \subset \equiv_g$ □

It follows from Definition 2 that the relation "dominates" is reflexive and transitive but not antisymmetric (as f ≥ g and g ≤ f implies S/f = S/g, but not necessarily f=g). Therefore, it is not a partial order on MAP. In order to define a partial order we need an equivalence relation on MAP such that mappings inducing the same partition on S are equivalent.

Definition 3 - Let $f,g \in MAP$. Then, f and g are equivalent, denoted $f \equiv g$ iff $f \geq g$ and $g \geq f$ □

It can be easily verified that $\equiv$ is an equivalence relation on MAP and that $\geq$ becomes a partial order on the set of equivalence classes in MAP.

Next, we shall look for maximal and minimal elements (up to equivalence) in the set MAP. First, let us see an example of equivalent data base mappings.

Example 4

Relation schemas : R(STUD,COURSE,GRADE,PROF), R1(STUD,COURSE), R2(COURSE,PROF), R3(STUD,COURSE,PROF)

Integrity constraints : c1 : STUD,COURSE $\rightarrow$ GRADE in R, c2 : COURSE $\rightarrow$ PROF in R, $C = c1 \wedge c2$

Data base variables : D = {R}, D1 = {R1,R2}, D3 = {R3}

Mappings : f1 = (R[STUD,COURSE], R[COURSE,PROF]) f2 = R[STUD,COURSE,PROF]

We have : $f1 \geq f2$ and $f2 \geq f1$, therefore, $f1 \equiv f2$ □

Let us now go back to our interpretation of a data base mapping as a communication channel. Intuitively, such a channel has a maximum capacity if it lets the whole data base information go through. It has a minimum capacity if it lets no information go through, that is, if it transmits the same message no matter what the data base value is. This leads to the following definition.

Definition 4 - Let $D_{to} \in S<D|C>$ be a fixed data base value. Define

$1_S : S<D|C> \rightarrow S<D|\lambda>$ such that $1_S(D_t) = D_t \quad \forall D_t \in S<D|C>$

$0_S : S<D|C> \rightarrow S<D|\lambda>$ such that $0_S(D_t) = D_{to} \quad \forall D_t \in S<D|C>$ □

The following theorem is an immediate consequence of Definition 4.

Theorem 3 - The following is true $\forall$ f $\in$ MAP

(i) $0_S \leq f \leq 1_S$

(ii) $=_S \subset \Xi_f \subset S \times S$ □

This theorem implies that 1_S and 0_S are the largest and the smallest element of MAP, respectively (up to equivalence).

Let us now look for the infimum and the supremum of two data base mappings f and g. We shall work with the equivalence relations Ξ_f and Ξ_g. Recall first (Theorem 2), that $f \geq g \iff \Xi_f \subset \Xi_g$. The relation

$$\alpha = (\Xi_f \cup \Xi_g)$$

is a first candidate, as $\Xi_f \subset \alpha$ and $\Xi_g \subset \alpha$. Unfortunately, α is not necessarily transitive i.e., it is not an equivalence relation. However, the transitive closure

$$\beta = \alpha^* = (\Xi_f \cup \Xi_g)^*$$

is always an equivalence relation and the following theorem shows that it is the required one.

Theorem 4 - Let f,g $\in$ MAP. Let $\beta = (\Xi_f \cup \Xi_g)^*$. Then S/$\beta$ is the infimum of S/f and S/g □

Proof : We must show that (1) S/$\beta \leq$ S/f and S/$\beta \leq$ S/g ; (2) for every γ such that : S/$\gamma \leq$ S/f and S/$\gamma \leq$ S/g we have S/$\gamma \leq$ S/β .

(1) Let $\alpha = (\Xi_f \cup \Xi_g)$. Then $\Xi_f \subset \alpha \subset \alpha^* = \beta$ and $\Xi_g \subset \alpha \subset \alpha^* = \beta$. Therefore, S/$\beta \leq$ S/f and S/$\beta \leq$ S/g Q.E.D.

(2) Let γ be an equivalence relation such that S/$\gamma \leq$ S/f and S/$\gamma \leq$ S/g. Then $\Xi_f \subset \gamma$ and $\Xi_g \subset \gamma$. Therefore, we obtain, $\alpha = (\Xi_f \cup \Xi_g) \subset \gamma$.
As α^* is the smallest equivalence relation containing α we conclude that $\beta = \alpha^* \subset \gamma$, i.e., S/$\gamma \leq$ S/β Q.E.D.

This theorem shows the existence of an infimum for f and g (up to equivalence). We shall denote this infimum by $f \wedge g$. For the supremum of f and g the task is easier as the relation $\equiv_f \cap \equiv_g$ is an equivalence relation.

Theorem 5 - Let $f,g \in MAP$. Then $S/\equiv_f \cap \equiv_g$ is the infimum of S/f and S/g □

Proof : Let $\alpha = \equiv_f \cap \equiv_g$. Then α is an equivalence relation. Now, $\alpha \subset \equiv_f$ and $\alpha \subset \equiv_g$ implies that $S/f \leq S/\alpha$ and $S/g \leq S/\alpha$. On the other hand let $h \geq f$ and $h \geq g$. Then $\equiv_h \subset \equiv_f$ and $\equiv_h \subset \equiv_g$. Therefore, $\equiv_h \subset (\equiv_f \cap \equiv_g) = \alpha$. It follows that $S/h \geq S/\alpha$ Q.E.D.

We shall denote the supremum (up to equivalence) of f and g by $f \vee g$. The mapping $f \vee g$ consists of "adding up" the information carried by f and g. A natural way to express this operation seems to be the juxtaposition of the values of f and g.

Definition 7 - Let f1 be a mapping from S into $S<D1|\emptyset>$. Let f2 be a mapping from S into $S<D2|\emptyset>$. Let $D1 \cap D2 = \emptyset$. The juxtaposition of f1 and f2, denoted by $f1 \times f2$ is a mapping from S into $S<D1 \cup D2|\emptyset>$, defined by

$$(f1 \times f2)(D_t) = (f1(D_t), f_2(D_t)) \quad \forall D_t \in S \qquad \square$$

Example 5

Relation schemas : R(NAME,ADDR,TEL), R1(NAME,ADDR), R2(NAME,TEL)

Integrity constraints : C : NAME → ADDR,TEL in R

Data base variables : D = {R}, D1 = {R1}, D2 = {R2}

Mappings : f1 = R[NAME,ADDR], f2 = R[NAME,TEL]

The juxtaposition of f1 and f2 is the mapping

$$f1 \times f2 = (R[NAME,ADDR], R[NAME,TEL])$$

It is interesting to note that in this example we have $f1 \times f2 \equiv 1_S$ □

Theorem 6 - Let $f,g \in MAP$. Then $f \times g \equiv f \vee g$ □

Proof : Let $D_1, D_2 \in S<D|C>$. Then,

$$D_1 \equiv_{f \times g} D_2 \iff f \times g(D_1) = f \times g(D_2)$$
$$\iff (f(D_1), g(D_1)) = (f(D_2), g(D_2))$$
$$\iff f(D_1) = f(D_2) \text{ and } g(D_1) = g(D_2)$$
$$\iff D_1 \equiv_f D_2 \text{ and } D_1 \equiv_g D_2$$

Therefore, $\equiv_{f \times g} = (\equiv_f \cap \equiv_g)$ that is, $S/_{f \times g} = S/_{f \vee g}$ Q.E.D.

This theorem implies that $f \times g$ is a representative of the equivalence class of $f \vee g$. Let us now study the case of $f \wedge g$ through an example.

Example 6

Relation schemas : R(STUD,EXAM,HOUR), R1(STUD,EXAM), R2(STUD,HOUR), R3(STUD)

Integrity constraints : c1 : STUD,HOUR → EXAM, c2 : EXAM → HOUR
C = c1 ∧ c2

Data base variables : D = {R}, D1 = {R1}, D2 = {R2}, D3 = {R3}

Mappings : f1 = R[STUD,EXAM], f2 = R[STUD,HOUR], f3 = R[STUD]

We have : f1 ∧ f2 = R[STUD] = f3. That is, the projection on the attribute STUD represents the smallest common part of the two mappings. □

We can generalize this last example in the case of two projections R[XY] and R[XZ] of the same relation. We can see that the projection R[X] is always the smallest common part, that is, it defines the mapping $f \wedge g$. However, it looks like, apart from this special case (projection and functional dependencies), there is no simple expression for $f \wedge g$, in terms of mappings. So, although we have found a simple expression for $f \vee g$, no such expression is in sight for $f \wedge g$.

As g is a complement of f and $f(D_{t1}) = f(D_{t1})$, we have : $g(D_{t1}) \neq g(D_{t2})$.

Therefore, one of the following two assertions is true

$$g(D_{t3}) \neq g(D_{t1}) \text{ or } g(D_{t3}) \neq g(D_{t2})$$

Suppose the following is true

$$g(D_{t3}) \neq g(D_{t1}) \qquad (2)$$

Let us define h by

$$\begin{cases} h(D_{t1}) = h(D_{t3}) = x \quad [\text{where x is a value not in } g(S<D|C>)] \\ h(D_t) = g(D_t), \quad \forall\, D_t \in S \quad \text{such that : } D_t \neq D_{t1},\ D_t \neq D_{t3} \end{cases} \qquad (3)$$

It follows from this definition that h is a complement of f and, because of (2) and (3) h is not greater than g, a contradiction to (1) Q.E.D.

Of course, this result does not exclude the existence of minimal complements.

Definition 9 - Let $f,g \in MAP$. Then, g is a minimal complement of f iff

(i) $f \times g \equiv 1_S$

(ii) $h \leq g$ and $f \times h \equiv 1_S \Rightarrow h \equiv g$ □

In the proof of Theorem 8 we have already seen how, given a minimal complement, we could construct a new one. This suggests that, in general, a data base mapping has more than one minimal complement. Can we give an intuitive meaning to these complements ? Let us see an example.

Example 8

Relation schemas : R(PART,COST,SALEPRICE,PROFIT,PROFITRATE), R1(PART,COST)
R2(PART,PROFIT), R3(PART,SALEPRICE), R4(PART,PROFITRATE)

Integrity constraints :

c1 : PART → COST,SALEPRICE,PROFIT,PROFITRATE in R

c2 : COST ≥ 0 in R, c3 : SALEPRICE > COST in R,

c4 : PROFIT = SALEPRICE - COST in R, c5 : PROFITRATE = PROFIT/COST in R

C = c1 ∧ c2 ∧ c3 ∧ c4 ∧ c5

Data base variables : D = {R}, $D_i = \{R_i\}$, i=1,2,3,4,5

Data base mappings : f1 = R[PART,COST], f2 = R[PART,PROFIT],
f3 = R[PART,COST]

One can verify easily that each of the pairs (f1,f2), (f1,f3) (f1,f4) suffices to reconstruct the data base. Therefore, each of the mappings f2, f3, f4 is a complement of f1. Furthermore, each of them is a minimal complement of f1 and no two of them are comparable (as none of them alone suffices to compute any other). □

In the previous example, we relied heavily on the fact that the attributes were strongly related by the integrity constraints (see, for example, constraints c4 and c5). Let us consider an example with only functional dependencies as integrity constraints.

Example 9

Relation schemas : R(MGR,SECR,DEPT,SAL), R1(MGR,DEPT)
R2(MGR,SECR,SAL), R3(MGR,SAL), R4(MGR,SECR), R5(DEPT,SECR)
R6(DEPT,SAL)

Integrity constraints :

c1 : MGR ↔ SECR ↔ DEPT in R, c2 : MGR → SAL, C = c1 ∧ c2

Data base variables : D = {R}, D1 = {R1}, D2 = {R2}, D3 = {R3,R5},
D4 = {R4,R6}

Data base mappings : f1 = R[MGR,DEPT], f2 = R[MGR,SECR,SAL],
f3 = (R[MGR,SAL], R[DEPT,SECR]), f4 = (R[MGR,SAL], R[DEPT,SECR])

The mappings f2, f3 and f4 are each a minimal complement of f1. Furthermore, no two of them are comparable . □

This example might suggest that the multiplicity of minimal complements is related to functional dependencies. So, let us conclude this section by a simple example that shows that the multiplicity of complements has nothing to de with functional dependencies either.

Example 10

Relational schemas : R1(MALE), R2(FEMALE), R3(PERSON)

Integrity constraints : C : $R1 \cap R2 = \emptyset$

Data base variables : $D = \{R1,R2\}$, $D1 = \{R1\}$, $D2 = \{R2\}$, $D3 = \{R3\}$

Data base mappings : $f1 = R1$, $f2 = R2$, $f3 = R1 \cup R2$

The mappings f2 and f3 are complements of f1 and they are not comparable. That is, with the information provided in this example, we cannot produce the list of females, starting from the list of persons and vice versa. □

INDEPENDENCE OF DATA BASE MAPPINGS

The concept of independence is well known in diverse branches of mathematics such as linear algebra and probability theory. The interest in this concept lies in the fact that it allows to act independently on two objects. Also it allows the decomposition of a complex system into simple and independent parts that we can study separately.

In this section, we study the concept of independence between two data base mappings. Viewing mappings as information carriers we can see, intuitively, the following situations arising.

1 - One of the mappings determines the other

2 - The two mappings are somehow related but none of them determines the other

3 - The two mappings have nothing to do with each other : they are independent.

We have already studied the first situation through the "dominance" relation in the previous section. We set now to study the other two.

Let us start by an example of logical propositions about a data base

P : John makes more than $ 1000
P1 : John makes $ 1200
P2 : John makes between $ 800 and $ 4000
P3 : Peter makes $ 5000

We see that P1 determines P. P2 and P are related but none of them determines the other. And, it looks like P3 and P are independent. Put differently, p determines p', if knowing p we know p' ; p is related to p', if knowing one of the two gives us some information about the other ; p is independent of p', if knowing p does not affect our knowledge of p'. This last approach is the one taken when defining the

independence of random variables in probability theory.

The concept of independence has been studied in the context of data base decomposition[16,24]. Decomposition is done through projections and the data base is reconstructed by a join. Furthermore, the integrity ocnstraints are restricted to be functional dependencies. In such a context, if we consider a relation schema R(EMP,DEPT,MGR) with EMP → DEPT ; DEPT → MGR , the two parts of the decomposition

$$\begin{cases} R[EMP,DEPT] & \text{with EMP} \rightarrow \text{DEPT} \\ R[DEPT,MGR] & \text{with DEPT} \rightarrow \text{MGR} \end{cases}$$

are considered independent, essentially because the set of the data base integrity constraints is preserved in the decomposition. On the other hand, if we consider the relation R(STUD,EXAM,HOUR) with STUD,HOUR → EXAM ; EXAM → HOUR then, the two parts of the decomposition

$$\begin{cases} R[STUD,EXAM] & \\ R[EXAM,HOUR] & \text{with EXAM} \rightarrow \text{HOUR} \end{cases}$$

are not considered independent because the set of data base integrity constraints is not preserved (in order to preserve it we must introduce constraints linking the two parts of the decomposition).

Our objective is to give a general definition of independence, applicable to any kind of data base mappings and any kind of integrity constraints and then test our definition on examples. In doing so, we shall start by the intuitive definition that we discussed earlier.

> "Two data base mappings f and g are independent if knowing the value of f does not affect our knowledge of the value of g, and vice versa".

Suppose that the data base is at state D_t at some time t. Think of an observer whose only knowledge of the data base is : $<D|C>$, f and g. That is, his knowledge consists of

$1 : D_t \in S<D|C>$

$2 : f(D_t) \in f(S<D|C>)$

$3 : g(D_t) \in g(S<D|C>)$

Suppose now that the observer sees the value $f(D_t) = D'_t$. His knowledge now about D_t has become

$1' : D_t \in f^{-1}(D'_t)$

That is, his knowledge about $f(D_t)$ is now complete

$2' : f(D_t) = D'_t$

Finally, his knowledge about $g(D_t)$ has become

$3' : g(D_t) \in g(f^{-1}(D'_t))$

This knowledge about $g(D_t)$ has not changed, compared to 3 above, if

$4 : g(f^{-1}(D'_t)) = g(S<D|C>)$

A similar reasoning for g gives

$5 : f(g^{-1}(D''_t)) = f(S<D|C>)$

We can rewrite 4 and 5 above as follows :

$\forall\ D'_t \in f(S)\ \ \forall\ D''_t \in g(S)\ \ \exists D_t \in S$ such that $f(D_t) = D'_t$ and $g(D_t) = D''_t$

which is equivalent to the following

$(f \times g)(S) = f(S) \times g(S)$

that is, every value in f(S) is "compatible" with every value in g(S).

Definition 10 - Let $f,g \in MAP$. Then f and g are independent, denoted $f \sim g$, iff $(f \times g)(S) = f(S) \times g(S)$ □

Let us test this definition on an example.

Example 11

Relation schemas : R1(EMPL,SAL), R2(PART #,COST)

Integrity constraints : c1 : EMPL → SAL in R1, c2 : PART# → COST in R2
C = c1 ∧ c2

Data base variables : D = {R1,R2}, D1 = {R1}, D2 = {R2}

Data base mappings : f1 = R1, f2 = R2

Clearly, (f1 × f2)(S) = f1(S) × f2(S). Therefore, f1 and f2 are independent. □

Let us now see an example of two mappings that are not independent.

Example 12

Relational schemas : R(STUD,EXAM,HOUR), R1(STUD,EXAM), R2(EXAM,HOUR)

Integrity constraints : c1 : EXAM → HOUR in R, c2 : STUD,HOUR → EXAM in R , C = c1 ∧ c2

Data base variables : D = {R}, D1 = {R1}, D2 = {R2}

Data base mappings : f1 = R[STUD,EXAM], f2 = R[EXAM,HOUR]

We have now (f1 × f2)(S) ≠ f1(S) × f2(S), since the following values of f1 and f2 are not compatible.

$D1_t$	STUD	EXAM
	John	Math
	John	Physics

$D2_t$	EXAM	HOUR
	Math	9 a.m
	Physics	9 a.m

Therefore, f1 and f2 are not independent (i.e., a user who lookds at $D1_t$ knows, about $D2_t$, that Math and Physics exams do not take place at the same hour). □

Let us now see an example of independence which does not seem to fit into our definition.

Example 13

Relational schemas : R(NAME,ADDR,AGE), R1(NAME,ADDR), R2(NAME,AGE)

Integrity constraints : C : NAME → ADDR,AGE in R

Data base variables : D = {R}, D1 = {R1}, D2 = {R2}

Data base mappings : f1 = R[NAME,ADDR], f2 = R[NAME,AGE]

Following Rissanen's definition[6], these two mappings are independent. However, the following values are not compatible.

$D1_t$	NAME	ADDR
	John	Paris
	Peter	London

$D2_t$	NAME	AGE
	John	25
	Jack	30

Therefore, according to our definition, the mappings f_1 and f_2 are not independent. This is so because looking at value $D1_t$ we learn something about the value $D2_t$. Namely, by looking at $D1_t$ we know the list of names in $D2_t$ (because $D1_t$ and $D2_t$ are projections of the same D_t). Nevertheless, there is some sort of independence between f_1 and f_2, that we could state as follows

> "given that the list of names in $D1_t$ and $D2_t$ is the same, knowing $D1_t$ does not change our knowledge about $D2_t$".

That is, f_1 and f_2 are independent up to a list of names. Notice, that the list of names can be represented by a third mapping $f3(D_t)$ = R[NAME] □

Definition 11 - Let $f,g,h \in MAP$. Then f and g are independent modulo h, denoted $f \sim g \bmod h$, iff

$$(f \times g)(B) = f(B) \times g(B) \quad \forall B \in S/h \qquad \square$$

Following again the natural structure that we have defined on MAP we define the complement of a given mapping f as any mapping g whose information "added" to that of f reconstructs the data base.

Definition 8 - Let f, g $\in$ MAP. Then g is a complement of f iff $f \times g \equiv 1_S$ □

The following theorem is an immediate consequence of Definition 8.

Theorem 7 - Let f,g $\in$ MAP. The following statements are equivalent

(i) g is a complement of f

(ii) $\equiv_{f \times g} = '=_S'$

(iii) $f \times g$ is injective

(iv) g distinguishes all pairs of data base values that cannot be distinguished by f. □

Example 7

Relation schemas : R(NAME,ADDR,AGE)

Integrity constraints : C : NAME $\rightarrow$ ADDR,AGE

Data base variables : D = {R}, D1 = D2 = D

f1 = (R|AGE $\geq$ 25), f2 = (R|AGE $\leq$ 30)

It follows from Definition 8 that f1 and f2 are complementary mappings. □

Some important consequences of Definition 8 are the following.

1 - The mapping 1_S is the complement of every f, as $f \times 1_S \equiv 1_S$. This implies that every mapping f has at least one complement.

2 - If g is a complement of f and $g_1 \equiv g$ then g_1 is a complement of f. That is, a complement is defined up to equivalence.

3 - If g is a complement of f and $g_1 \geq g$ then g_1 is a complement of f (to see this observe that $g_1 \geq g \Rightarrow f \times g_1 = f \times g$ and, as $f \times g \equiv 1_S$ we obtain $f \times g_1 \equiv 1_S$)

It follows from Definition 8 that D_t and $(f(D_t),g(D_t))$ are equivalent representations of the data base, in the sense that they contain the same information. The second representation however, in light of point 3 above, may contain redundant information. Therefore, we would like to find a complement of f which is unique and minimal Such a complement would contain the smallest amount of information necessary to add to f in order to reconstruct the data base. Let us then define the set

$$COMP(f) = \{h | h \in MAP \text{ and } h \text{ is complement of } f\}$$

The question we are asking rephrased in mathematical terms, is the following : if the infimum of COMP(f) exists, does it belong to this set ? Apart from a trivial case, the answer is no.

Theorem 8 - The infimum of COMP(f) belongs to COMP(f) iff $f \equiv 1_S$ or $f \equiv 0_S$ □

Proof : If part :

1 - Suppose $f \equiv 0_S$. Then $f \times g \equiv 1_S \Rightarrow g \equiv 1_S$. Therefore, $COMP(f) = \{1_S\}$ and $\inf(COMP(f)) = 1_S \in COMP(f)$

2 - Suppose $f \equiv 1_S$. Then $g \in COMP(f)$ $\forall$ $g \in MAP$. That is, $COMP(f) = MAP$ and, therefore, $\inf(COMP(f)) = 0_S \in COMP(f)$

Only if part : We shall assume that the infimum of COMP(f) exists and that it belongs to COMP(f) and we shall derive a contradiction. That is, we suppose that $f \not\equiv 0_S$, $f \not\equiv 1_S$ and
$\exists$ g such that (i) $f \times g \equiv 1_S$ and (ii) $f \times h \equiv 1_S \Rightarrow h \geq g$ (1)
We have that :

$$f \not\equiv 1_S \Rightarrow \exists A \in S/f \text{ such that : } |A| > 1$$
$$\Rightarrow \exists A \in S/f, D_{t1}, D_{t2} \in A, D_{t1} \neq D_{t2}$$
$$f \not\equiv 0_S \Rightarrow \exists B \in S/f \text{ such that : } B \neq A \text{ and } B \neq \emptyset$$
$$\Rightarrow \exists D_{t3} \in B \text{ such that : } D_{t3} \neq D_{t1} \text{ and } D_{t3} \neq D_{t2}$$

That is, $f \sim g \bmod h$ iff $f \sim g$ on every $B \in S/h$. Looking back at the previous example, we see that $f1 \sim f2 \bmod f3$. It is interesting to note that for $h \equiv 0_S$, Definition 11 reduces to Definition 10. That is,

$$f \sim g \iff f \sim g \bmod 0_S$$

Let us now study the properties of independence. First, let us see an equivalent characterization of independence.

Definition 12 - Let $f1, f2 \in MAP$. Let $f1(S) = S1$, $f2(S) = S2$. The values $D1_t \in S1$, $D2_t \in S2$ are called compatible iff there exists $D_t \in S$ such that $f1(D_t) = D1_t$ and $f2(D_t) = D2_t$ □

It follows from this definition that

$$(D1_t, D2_t) \text{ is compatible iff } f1^{-1}(D1_t) \cap f2^{-1}(D2_t) \neq \emptyset$$

This leads to an equivalent definition of independence.

Theorem 9 - Let $f1, f2 \in MAP$. Then $f1 \sim f2$ iff

$$\forall D1_t \in S1 \quad \forall D2_t \in S2 \quad f1^{-1}(D1_t) \cap f2^{-1}(D2_t) \neq \emptyset \quad \square$$

As an immediate consequence of this theorem we obtain the following properties of independence.

Theorem 10 - Let $f,g,h \in MAP$. Then

(i) $f \sim 1_S \iff f \equiv 0_S$

(ii) $f \sim 0_S$

(iii) $f \sim f \iff f \equiv 0_S$

(iv) $f \sim g$ and $h \leq f \Rightarrow h \sim g$ □

Let us now look at properties of conditional independence.

Theorem 11 - Let $f,g,h \in MAP$. Then

$$h \geq f \Rightarrow f \sim g \bmod h \quad \square$$

Proof : Let $B \in S/h$. As $h \geq f$, there exists $A \in S/f$ such that $B \subset A$. Therefore, $f(B) = f(A) = \{y\}$, for some $y \in f(S)$. We have :

$$\begin{aligned}(f \times g)(B) &= \{(f(D_t), g(D_t)) \mid D_t \in B\} \\ &= \{(y, g(D_t) \mid D_t \in B\} \\ &= \{y\} \times \{g(D_t) \mid D_t \in B\} \\ &= \{y\} \times g(B) \\ &= f(B) \times g(B) \qquad \text{Q.E.D.}\end{aligned}$$

The following are immediate consequences of Theorem 11.

$f \sim g \bmod f$

$f \sim g \bmod g$

$f \sim g \bmod f \vee g$

$f \sim g \bmod 1_S$

The following theorem translates Theorem 10 to the case of conditional independence. Its proof is trivial.

Theorem 12 - Let $f,h \in MAP$. Then

(i) $\quad f \sim 1_S \bmod h \iff f \leq h$

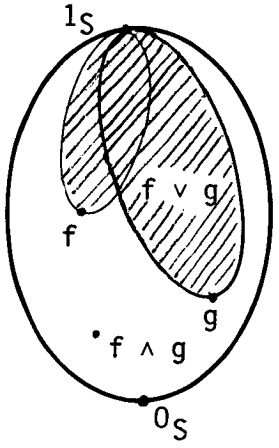
(ii) $\quad f \sim 0_S \bmod h$

(iii) $\quad f \sim f \bmod h \iff f \leq h$ □

Let us look again into the meaning of $f \sim g \bmod h$. The mappings f and g are independent "up to h" and, for $h \equiv 0_S$, we have total independence. So, h expresses somehow a "link" between f and g.
Of course, we would like to see what is the "weakest link", if it exists, up to which f and g are independent. To do this, we must study the set

$$MOD(f,g) = \{h \mid f \sim g \bmod h\}$$

Theorem 11 already gives some information on this set as the following figure shows. The hachured part is included in the set MOD(f,g), but what we are mainly interested in is the lower part, i.e., the part lying under f and g. To study this part we need the following lemmas.



Lemma 1 - Let $f,g,h \in MAP$. Let $T \subset S$. Then

$$\left.\begin{array}{ll} (1) & (f \times g)(T) = f(T) \times g(T) \\ (2) & h \leq f \text{ and } h \leq g \end{array}\right\} \Rightarrow |h(T)| = 1 \qquad \square$$

Proof : Let f',g',h' denote the three mappings restricted to the set T. Clearly, (1) and (2) still hold for the restrictions, and it follows from (1) that $f' \sim g'$. Then, Theorem 9 implies that

$$\forall A' \in T/f' \quad \forall B' \in T/g' \quad A' \cap B' \neq \emptyset \qquad (3)$$

It follows from (2) that

$$\forall A' \in T/f' \quad \exists C_1 \in T/h' \text{ such that } A' \subset C_1$$
$$\forall B' \in T/g' \quad \exists C_2 \in T/h' \text{ such that } B' \subset C_2$$

It follows from (3) that $C_1 = C_2 = T$ Q.E.D.

Lemma 2 - Let $f,g,h,k \in MAP$. Then,

$$\left.\begin{array}{l} f \sim g \bmod k \\ h \leq f \text{ and } h \leq g \end{array}\right\} \Rightarrow k \geq h \qquad \square$$

Proof : $f \sim g \bmod k$ implies that

$$(f \times g)(T) = f(T) \times g(T) \qquad \forall\, T \in S/h$$

Applying Lemma 1 we obtain that $|h(T)| = 1$. Therefore, there exists $C \in S/h$ such that $T \subset C$, i.e., $k \geq h$ Q.E.D.

The following theorem is an immediate consequence of Lemma 2.

Theorem 13 - Let $f,g,h \in MAP$. Then,

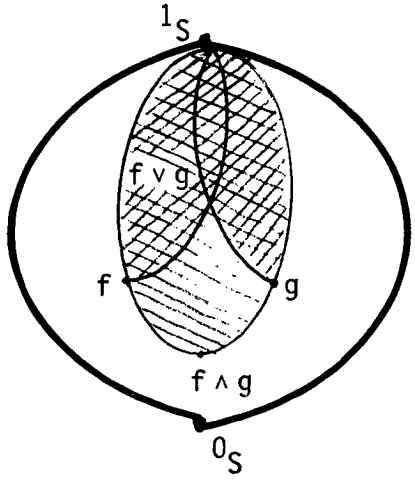
$$f \sim g \bmod h \Rightarrow h \geq f \wedge g \qquad \square$$

A corollary of this theorem is the following

Theorem 14 - Let $f,g,h \in MAP$. Then,

$$f \sim g \Rightarrow f \wedge g \equiv 0_S \quad \text{(but the converse is not true)} \qquad \square$$

It follows from Theorem 13 that every mapping in the set MOD(f,g) dominates $f \wedge g$ as shown in the following figure (the set MOD(f,g) corresponds to the shaded area).



Two negative results about independence are the following

1 - It is not always true that : $f \sim g \bmod (f \wedge g)$. Looking back at Example 12, we have : $f_1 \wedge f_2 = R[HOUR]$ and $f_1 \sim f_2 \bmod f_1 \wedge f_2$ is not true.

2 - The set MOD(f,g) is not closed under "dominance", i.e., it is possible to find mappings f,g,k,h_1 such that $f \sim g \bmod h$, $h_1 \geq h$ and $f \sim g \bmod h_1$ is false. One simply has to add to h some information that "correlates" f and g as the following example shows.

Example 14

Relation schemas : R(PART,COST,SALEPR), R1(PART,COST), R2(PART,SALEPR) R3(PART), R4(PART)

Integrity constraints : C : PART → COST,SALEPR

Data base variables : D = {R}, D1 = {R1}, D2 = {R2}, D3 = {R3} D4 = {R3,R4}

Mappings : f1 = R[PART,COST], f2 = R[PART,SALEPR], f3 = R[PART] f4 = (R[PART], (R|COST < SALEPR)[PART])

Then we have :

f1 ∿ f2 mod f3, f4 ≥ f3 but f1 ∿ f2 mod f4 is false □

In conclusion, we have seen two notions of independence

- Total independence : f ∿ g
- Conditional independence : f ∿ g mod h

We have shown that the best we can do is : f ∿ g mod (f ∧ g). We shall call this "optimal" case weak independence.

CONCLUSION

Viweing data base mappings as "information carriers", we have been able to define formally the following key concepts

1 - Dominance : Given two data base mappings carrying comparable information the one that carries more information dominates the other.

2 - Independence : Two data base mappings are independent if knowing the value of the one does not affect our knowledge of the value of the other.

3 - Weak independence : Two data base mappings are weakly independent if they are independent "up to their common part".

4 - Complement : Two data base mappings are complementary if the "sum" of the information they carry is sufficient to recompute the data base.

It is important to note that our theory does not depend on any specific data model. However, we have chosen to present it in the context of a specific model, namely, the relational model, in order to avoid excessive formalism.

In the second part of this paper we apply the above concepts to two specific data base problems : data base decomposition and view updating. We also discuss some parallels that exist between the theory of data base mappings, on the one hand, and the theory of relations and probability theory, on the other hand.

REFERENCES

[1] Aho A.V., C. Beeri et J.D. Ullman (1978). "The theory of joins in relational database", submitted to TODS. A preliminary version appeared in Proc. Eighteenth Annual IEEE Symposium on Foundations of Computer Science, pp. 107-113.

[2] Armstrong W.W. (1974). "Dependency structures of data base relationships", Proc. 1974 IFIP Congress, pp. 580-583, North Holland, Amsterdam.

[3] Bancilhon F. (1978). "On the completeness of query languages for relational databases", Proc. Seventh Symp. on Mathematical Foundations of Computer Science, Springer Verlag.

[4] Bancilhon F., N. Spyratos. "Update Semantics of Relational Views", To appear in ACM TODS.

[5] Beeri C., P.A. Bernstein et N. Goodman (1978). "A sophisticate's introduction to database normalization theory", Proc. ACM Intl. Conf. on Very Large Data Bases, pp. 113-124.

[6] Beeri C., A.O. Mendelzon, Y. Sagiv et J.D. Ullman (1979). "Equivalence of relational database schemes", Proc. Eleventh Annual ACM Symposium on the Theory of Computing, pp. 319-329.

[7] Beeri C., J. Rissanen (1979). "On the decomposition of relational Data Base Schemas", Acte des Journées d'étude "Bases Formelles pour Bases de Données", Toulouse.

[8] Biskup J., U. Dayal et P.A. Bernstein (1979). "Synthesizing independent database schemas", ACM/SIGMOD International Symposium on Management of Data, pp. 143-152.

[9] Chamberlain D.D., J.N. Gray et D.D. Traiger (1975). "Views, authorization and locking in a relational data base system". Proceedings of AFIPS NCC, Vol. 44.

[10] Chandra A.K. et D. Harel (1978). "Computable queries for relational databases", Proc. Eleventh Annual ACM Symposium on the Theory of Computing, pp. 309-319.

[11] Codd E.F. (1970). "A relational model for large shared data banks", Comm. ACM 13 : 6, pp. 377-387.

[12] Codd E.F. (1972a). "Further normalization of the data base relational model", in Data Base Systems (R. Rustin, ed.) Prentice Hall, Englewood Cliffs, N.J., pp. 33-64.

[13] Codd E.F. (1972b). "Relational completeness of data base sub-languages". ibid, pp. 65-98.

[14] Dayal U. et P.A. Bernstein (1978). "On the updatability of Relational views", Proceedings of the 4th International Conference on Very Large Data Bases, West Berlin.

[15] Dayal U. (1979). "Schema-Mapping problems in data base systems", Ph. Thesis, Harvard University.

[16] Delobel C. et R.C. Casey (1972). "Decomposition of a database and the theory of Boolean switching functions", IBM J. Res. 17 : 5, pp. 370-386.

[17] Delobel C. (1978). "Normalization and hierarchical dependencies in the relation data model", ACM Trans. on Database Systems 3 : 3, pp. 201-222.

[18] Fagin R. (1977). "Multivalued dependencies and a new normal form for relational data bases", ACM Trans. on Database Systems 2 : 3, pp. 262-278.

[19] Makinouchi A. (1977). "A consideration on Normal Form of Not-necessarily Normalized Relation in the Relation Data Model". Proceedings Very Large Data Bases, Third International Conf., Tokyo, Japan, October 6-8, pp. 447-453.

[20] Nicolas J.M. (1978). "Mutual dependencies and some results on undecomposable relations", Proc. ACM Intl. Conf. on Very Large Data Bases, pp. 360-367.

[21] Paolini P. et G. Pelegatti (1977). "Formal definition of mappings in a data base", ACM/SIGMOD, Proceedings of the Int. Conf. on Management of Data, August, pp. 40-46.

[22] Paolini P. (1979). "Formal definition of views and verification of applications programs". Acte des journées d'étude "Bases Formelles pour Bases de Données", Toulouse.

[23] Paredaens J. (1978). "On the expressive power of relational algebra", Information Processing Letters 7 : 2, pp. 107-111.

[24] Rissanen J. (1977). "Independent components of relations", ACM Trans. on Database Systems 2 : 4, pp. 317-325.

[25] Rissanen J. (1978). "Relations with functional and join dependencies and their representation by independent components", Unpublished manuscript, IBM, San Jose, California.

[26] Sevcik K.C., A.L. Furtado (1978). "Complete and Compatible sets of Update Operations", Proc. ICMOD 78, Milan, Italie, Juin, pp. 247-260.

[27] Spyratos N. (1980). "Translation Structures of Relational Views", Proc. 6th Intl. Conf. on Very Large Data Bases, Montreal, Canada.

Imprimé en France

par

l'Institut National de Recherche en Informatique et en Automatique