Serdica

Bulgariacae mathematicae publicationes

Сердика

Българско математическо списание

NEW APPROACH IN THE DESIGN OF LOGICAL DATA BASE SCHEMA*

C. DELOBEL, M. LEONARD

An important step of data base disign is to convert user's knowledge on the meaning of the informations into a logical data base schema. After a presentation of the most important concepts of the relational model we study two particular types of integrity rules, the properties of which are given; one of these types is new. Then we show the importance of integrity rules to represent a relation into a logical data base schema. At last we give one example to explain the conversion of a relation into a logical data base schema.

Introduction. How to convert user's knowledge on the meaning of the informations into a logical data base schema is an important step of data base design. This step often is called design of the logical data base schema. It has been generally accepted that there should be at least six basic levels of system abstractions namely:

(1) data semantics or user's level of abstraction, reflecting user's description of the information;

(2) conceptual model, combining the views of all users for all applications into an integrated logical data model;

(3) external model, describing one view of some users for some applications with a logical data model;

(4) internal model, reflecting all the physical implementation details of the system;

(5) data base implementation;

(6) performances measuring;

The domain of design logical data base schema is the first two levels and also converting the first one into the second one.

The purpose of the level d a t a s e m a n t i c s consists of providing formal concise description of user's knowledge on the meaning of the information without any considerations about future data manipulations. Such a description contains:

- all necessary informations;
- groups of informations which appear obvious to the users;
- properties of these informations and interrelationships among them.

We introduce new integrity constraints which are different to functional dependencies. We study how to find a set of nonredundant integrity constraints and how to avoid information redundancy during assembling informations into groups.

The purpose of the level c o n c e p t u a l m o d e l is to design a logical data base model. This step contains a choice between different solutions and

*Delivered at the Conference on Systems for Information Servicing of Professionally Linked Computer Users, May 23-29 1977, Varna.

decision making needs surveying future data manipulations. Our purpose consists only of converting the results of the first step data semantics into a particular conceptual model: this one uses as framework either relational model or network model or hierarchical model.

1. **The relational model.** This section is intended for reviewing some fundamental definitions which are presented completely in [2; 4].

● A relation R is defined over a sequence of attributes $X=\{A_1 \dots A_n\}$ with one and only one predicate whose variables are the attributes of X; this predicate denoted $\| R(A_1, A_2 \dots A_n) \|$ expresses the semantic meaning which relates these attributes. We denote R^* the name of the relation R.

● Let $a_1, a_2 \dots a_n$ be respectively objects of $A_1 \dots A_n$: if $\| R(a_1, a_2 \dots a_n) \|$ is true, then $x=(a_1, a_2 \dots a_n)$ is an entity of R. We denote: $x \in R$.

An integrity rule of R is a part of the predicate $\|R\|$ defined on a subset Y of X; it must be satisfied automatically by the result of every data base change (i. e. update insert create or delete entity), for the change to be allowed; generally at one integrity rule is associated one validation process.

A relation schema $(R^*, \mathcal{F})$ consists of one relation represented by its name R^* and the set $\mathcal{F}$ of integrity rules which are defined upon.

A data-collection C of R is a set of distinct R-entities; it may be seen as a table in which each column corresponds to a distinct attribute and each row to a distinct entity.

● A subrelation R_1 of R is a relation defined on a subset $Y=\{A_p \dots A_q\}$ of X such that: if $x=(a_1, a_2 \dots a_p \dots a_q \dots a_n)$ is an entity of R, then $g=(a_p \dots a_q)$ is an entity of R_1. We denote: $R_1=[A_p \dots A_q]\, R$.

● Let A and B two subsets of X.

B is said to be functionally dependent on A in R if $\forall a \in [A]\, R$, $\forall b$, $b_1 \in [B]\, R$, $\forall c$, $c_1 \in [C]\, R$

$\| R(a, b, c) \| = \text{true}$ and $\| R(a, b_1, c_1) \| = \text{true} \rightarrow b=b_1$

We note $A \rightarrow B$ and we name it functional dependency (FD).

$A, B \rightarrow C$ is an elementary functional dependency (EFD) if neither $A \rightarrow C$ nor $B \rightarrow C$ are functional dependencies.

● I attribute of X is a key of R iff:

$\forall A_i \in X \quad I \rightarrow A_i$

$\forall B \subset I \quad \exists A_k \in X \quad A_k$ is not functionally dependent on B in R.

● R is said nonfunctional if its key is X itself.

We denote it: $[A_1, A_2 \dots A_n]$.

● Operations on relations.

There are two basic operations which interest us, namely: projection and natural join.

The projection of a relation R over the subrelation R_1 is the restriction of R over the only attributes of R_1.

The natural join operation is used to make a new relation $T(A, B, C)$ from two relations $R(A, B)$ and $S(A, C)$, where A, B, C are disjoint sets of attributes, such that

$\| T(A, B, C) \| = \| R(A, B) \| . \| S(A, C) \|$; the new relation is so defined on the union of the attributes sets of the old relations. We denote: $T(A, B, C)= R(A, B) * S(A, C)$.

● Decomposition of a relation.

Let R be a relation. We shall say that R is decomposable into the subrelations $R_1 \dots R_n$ iff R satisfies the condition:

$$R = R_1 * \ldots * R_n.$$

We shall recall the main characterizations of when R is decomposable, which have been proved elsewhere [6].

Proposition 1. *R is decomposable if and only if for all $a \in [A]R$*

$$[a, B, C]R = [a, B]R * [a, C]R,$$

where the notation $[a, B, C]R$ stands for the set of $\{(b, c) \mid R(a, b, c) = \text{true}\}$

$$[a, B]R = \{b \mid (a, b) \in [A, B]R\},$$

$$[a, C]R = \{c \mid (a, c) \in [A, C]R\}.$$

Proposition 2. *R is decomposable iff for all $a \in [A]R$ $b \in [a, B]R$*

$$[a, b, C]R = [a, C]R.$$

Proposition 3. *If R is the relation of the relation schema $(R^*(A, B, C)$ $A \to B)$ then*

$$R = [A, B]R * [A, C]R.$$

2. Particular integrity rules: functional and relational dependencies.

2.1. Functional dependencies. The functional dependencies have the following properties: let $R(A, B, C, D)$ be a relation and $\mathcal{F}$ the set of FDs for R then

(4) reflexivity $A \to A \in \mathcal{F}$

(5) projection if $A \to B, C \in \mathcal{F}$ then $A \to B$ and $A \to C \in \mathcal{F}$

(6) augmentation if $A \to B \in \mathcal{F}$ then $A, C \to B \in \mathcal{F}$

(7) additivity if $A \to B$ and $A \to C \in \mathcal{F}$ then $A \to B, C \in \mathcal{F}$

(8) transitivity if $A \to B$ and $B \to C \in \mathcal{F}$ then $A \to C \in \mathcal{F}$

(9) preudo-transitivity if $A \to B$ and $B, C \to D \in \mathcal{F}$ then $A, C \to D \in \mathcal{F}$.

There are several axiomatizations of FDs from these properties [7]. The one we will use is based on properties (4), (6), (9).

● Graphical representation of a set of FDs

Let $\mathcal{F}$ be a set of FDs $\{f_1, f_2, \ldots, f_n\}$ over an attribute set $A = \{A_1, A_2, \ldots, A_k\}$. We define a graph, denoted $G(\mathcal{F})$, as follows. The graph includes two types of nodes: the round nodes represent the attribute nodes, the square nodes represent the FDs. An edge is directed from a round node A_i to a square node f_j if $f_j: A_i, B \to C$, where B and C are attributes of A exists;

An edge is directed from a square node f_i to a round node A_m if $f_k: B \to A_m$ exists.

Example 2. 0. Let $\mathcal{F}$ be the set of FR's over the attributes $\{A, B, C, D, E, F, G\}$

$$\mathcal{F} = \{f1, f2, f3, f4, f5\}$$

$$f1: ACG \to E,\ f2: F \to E,\ f3: AD \to F,\ f4: BC \to D,\ f5: AC \to B.$$

We give the corresponding graph in Fig. 1.

We say $\mathcal{F}$ is circuitless if $G(\mathcal{F})$ is itself circuitless.

● We can consider all the previous properties of FDs as a set of rules for obtaining new FDs from a given set $\mathcal{F}$. The closure of $\mathcal{F}$, denoted $\mathcal{F}+$, is defined as the set of all FDs that are obtainable by successive applications of these properties. An elementary closure of $\mathcal{F}$, denoted $\mathcal{F}^{\oplus}$, is a subset of $\mathcal{F}+$ such every FD is a EFD. An elementary minimum covering of $\mathcal{F}$, denoted $\mathcal{F}^*$, is a part of $\mathcal{F}^{\oplus}$ such that:

(a) $$(\mathcal{F}^*)^+ = \mathcal{F}^+$$

(b) $$\forall \mathcal{G} \subset \mathcal{F}^* \ \mathcal{G}^+ \neq \mathcal{F}^+.$$

● The concept of elementary minimum covering $\mathcal{F}^*$ of an initial set of DFs is very important for the validation of integrity rules. Indeed the valida-

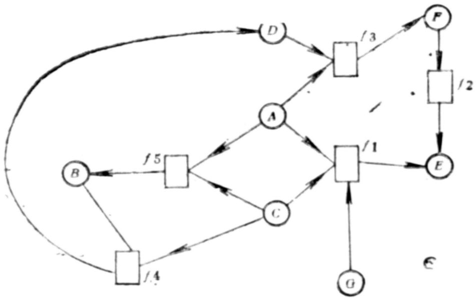

Fig. 1

tion of all FDs of $\mathcal{F}$ is obtained by the validation of only all FDs of $\mathcal{F}^*$. Therefore, it is very useful to build an algorithm to obtain one $\mathcal{F}^*$ from $\mathcal{F}$.

● [5] proved the strong analogy beween a set of FDs and a boolean function.

Example 2. 1.

$$\mathcal{F} = \{A \to B;\ B, C \to D;\ A \to E\} \iff f = a\bar{b} + bc\bar{d} + a\bar{e}.$$

This result permits to prove that $\mathcal{F}^{\oplus}$ is unique and if $\mathcal{F}$ is circuitless, $\mathcal{F}^*$ is unique [14].

Other mathematical results have been proved and efficient algorithms [12; 15; 17] have been built to obtain the elementary minimum covering $\mathcal{F}^*$.

2. 2. Relational dependencies. A relational dependency (RD) of a relation R is an integrity rule: it is a decomposition of some projection of R or R itself.

Example 2. 2. R(SEMINAR, STUDENT, INSTRUCTOR) is a relation of an education program planning.

The relational dependency:

R = [SEMINAR, STUDENT]R * [SEMINAR, INSTRUCTOR]R

expresses all students attending a seminar are teached by all instructors of this seminar.

We study now two types of RD: first order hierarchical decomposition (FOHD), multivalued dependency (MD).

2.2.1. First order hierarchical decomposition (FOHD) [6].

● Let $R(X, Y, Z, U)$ be relation where X, Y, Z, U are disjoint attributes. $X: Y \mid Z \mid U$ is a first order hierarchical decomposition of R iff $R = [X, Y]R *$ $[X, Z]R * [X, U]R$.

X is the root of the FOHD, Y, Z, U the branches.

● Properties of the FOHD

It is not possible to give here all the proofs for the properties. These proofs can be found in [6].

(10) — Clustering of branches

Let $X: Y|Z|U$ be a FOHD, then if Y and Z are two chosen branches $X: Y, Z|U$ is a FOHD.

(11) — Deletion of a branch

Let $X: Y|Z|U$ be FOHD, then if Y and Z are two chosen branches $X: Y|Z$ is a FOHD.

(12) — Projection of a branch

Let $X: Y|Z$ be a FOHD and T be a subset of Y, then $X: T|Z$ is a FOHD.

(13) — Projection of the root

Let $X, Y, Z: U|V$ and $X, Y: Z|U, V$ be two FOHDs, then $X, Y: Z|U|V$ is a FOHD. Let $X, Y, Z: U|V$ be a FOHD such that $X, Y \rightarrow Z$, then $X, Y: Z|U|V$ is a FOHD.

(14) — Root modification

Let $X: Y, Z|U, V$ be a FOHD and Y, U be two chosen subsets among Y, Z, U, V, then $X, Y, U: Z|V$ is a FOHD.

(15) — Decomposition of a branch

Let $X: U|V$

$X: Y|Z$ be FOHDs such that $Y \subset U$ and $Z \subset U$,

then $X: Y|Z|V$ is a FOHD.

(16) — Complex generation of FOHD

Let $X_1: Y_1|Z_1|T_1|U_1$

$X_2: Y_2|Z_2|T_2|U_2$ be two FOHDs such that

$Y_1, Z_1, T_1, U_1 \supset X_2 \wedge X_1'$

$Y_2, Z_2, T_2, U_2 \supset X_1 \wedge X_2'$

where X_1' and X_2' denote the complement of X_1 and X_2 in $\mathcal{A}$ (where $\mathcal{A} = \{X_1, Y_1, Z_1, T_1, U_1, X_2, Y_2, Z_2, T_2, U_2\}$),

if Y_1 and Z_1 are two chosen branches then

$$X_1, (Y_1, Z_1) \wedge X_2: Y_1 \wedge Y_2 | Y_1 \wedge Z_2 | Y_1 \wedge T_2 | Y_1 \wedge U_2 | Z_1 \wedge Y_2 | Z_1 \wedge Z_2 | Z_1 \wedge T_2 | Z_1 \wedge U_2 | T_1 | U_1$$

is a FOHD.

(17) ● All these properties of FOHDs can be observed as a set of rules for obtaining new FOHDs from a given set of FOHDs. Interactions between a set $\mathcal{F}$ of FDs and a set $\mathcal{G}$ of FOHDs over the same relation were proved like:

— Let $X: Y|Z \in \mathcal{G}$ and $Y \rightarrow Z \in \mathcal{F}$
  then $X \rightarrow Z \in \mathcal{F}$

— Let $X: Y|Z \in \mathcal{G}$ and $Y \rightarrow U \in \mathcal{F}$
  then $X: Y, U|Z \in \mathcal{G}$

— Let $X: Y|Z \in \mathcal{G}$ and $X, Y \rightarrow U \in \mathcal{F}$ and $X, Z \rightarrow U \in \mathcal{F}$
  then $X \rightarrow U \in \mathcal{F}$.

2.2.2. **Multivalued dependencies** (MD) [13]

● Let $R(A_1, A_2, \ldots, A_n)$ be a relation over the set of attributes: $\mathcal{A} = \{A_1, A_2 \ldots A_n\}$,
then we say A_1 multi-determines A_2 in R iff $\forall B \subset \mathcal{A}, B \wedge (A_1, A_2) = \emptyset$, $A_1: A_2|B$ is a FOHD in R.
We write the multivalued dependency $A_1 \rightarrow\rightarrow A_2$.

(18) ● Let $R(A_1, A_2 \ldots A_n)$ be a relation over the set of attributes $\mathcal{A}$ and let $A_1 \rightarrow A_2$ be a functional dependency,
then [6] proved simply that:

$$\forall B \subset \mathcal{A},\ B \wedge (A_1, A_2) = \emptyset \quad A_1 : A_2 \mid B.$$

Therefore a FD of R is a particular MD of R.

● Properties of MD ([13, 16], and simplier proved in [18])

Let $R(A, B, C, D, E, F)$ be a relation and let $\mathcal{H}$ be a set of MDs on R

— full MD

Let $A \rightarrow\rightarrow A, B$ be a MD of $\mathcal{H}$, then $A \rightarrow\rightarrow B \in \mathcal{H}$ and can replace the first one in $\mathcal{H}$

$A \rightarrow\rightarrow B$ with A and B disjoint attributes is called full MD. We shall consider only such MDs.

(19) — reflexivity $A \rightarrow\rightarrow A \in \mathcal{H}$

(20) — augmentation if $A \rightarrow\rightarrow B \in \mathcal{H}$ then $A, C \rightarrow\rightarrow B \in \mathcal{H}$

(21) — additivity if $A \rightarrow\rightarrow B \in \mathcal{H}$ and $A \rightarrow\rightarrow C \in \mathcal{H}$ then $A \rightarrow\rightarrow (B, C) \in \mathcal{H}$

(22) — transitivity if $A \rightarrow\rightarrow B \in \mathcal{H}$ and $B \rightarrow\rightarrow C \in \mathcal{H}$ then $A \rightarrow\rightarrow C \in \mathcal{H}$

(23) — pseudo-transitivity if $A \rightarrow\rightarrow B \in \mathcal{H}$ and $B, C \rightarrow\rightarrow D \in \mathcal{H}$ then $A, C \rightarrow\rightarrow D \in \mathcal{H}$.

Note that we give none conditions on the attributes about transitivity and pseudo-transitivity. The new results concern these cases:

— if $A, B \rightarrow\rightarrow C \in \mathcal{H}$ and $C, D \rightarrow\rightarrow A, E \in \mathcal{H}$, where A, B, C, D, E are disjoint attributes and F is their complement, then

$$A, B, D \rightarrow\rightarrow A, E \text{ and so } A, B, D \rightarrow\rightarrow E \in \mathcal{H}.$$

Proof. (1) $A, B : C \mid D, E, F$

(2) $C, D : A, E \mid B, F$

$\Rightarrow$ (3) $A, B, D : C \mid E, F$ (by application of (14))

(4) $A, B, C, D : E \mid F$

(5) $A, B, D : C \mid E \mid F$ (by application of (13))

and so $A, B, D \rightarrow\rightarrow E$

— if $A \rightarrow\rightarrow B \in \mathcal{H}$ and $B \rightarrow\rightarrow A, C \in \mathcal{H}$, where A, B, C are disjoint attributes then $A \rightarrow\rightarrow A, C$ and so $A \rightarrow\rightarrow C \in \mathcal{H}$.

The proof is the same as the last one if $A, B \rightarrow\rightarrow C \in \mathcal{H}$ and $C \rightarrow\rightarrow A, D \in \mathcal{H}$ then $A, B \rightarrow\rightarrow D \in \mathcal{H}$.

Example 2.3. We consider an education program planning with attributes: SEMINAR, TYPE, MONTH, DAY, ROOM, INSTRUCTOR, STUDENT, SCORE, BOOK, RANK, SALARY. A SEMINAR is characterized by an identification number, each SEMINAR corresponds to a certain TYPE and is scheduled for every month. Each SEMINAR has various INSTRUCTORS and STUDENTS, but has only one location characterized by the attribute SCORE. Each TYPE of SEMINAR has a given set of BOOKS, which are used by all the STUDENTS of the SEMINAR as references.

So here is the relation R (SEMINAR, TYPE, MONTH, ROOM, INSTRUCTOR, STUDENT, SCORE, BOOK, RANK, SALARY) with the following integrity rules:

SEMINAR $\rightarrow$ TYPE

INSTRUCTOR, MONTH $\rightarrow$ SEMINAR

SEMINAR $\rightarrow$ ROOM

STUDENT, MONTH $\rightarrow$ SEMINAR

INSTRUCTOR $\rightarrow$ RANK, SALARY

STUDENT, MONTH, SEMINAR $\rightarrow$ SCORE

Φ : MONTH | TYPE, INSTRUCTOR

which expresses that the organisation is the same every month; one can notice that the root of the FOHD is empty.

SEMINAR: STUDENT | INSTRUCTOR

which expresses that all students attending a seminar are teached by all instructors of the seminar.

TYPE →→ BOOK

which expresses that the knowledge of the type of a seminar determined the set of books.

3. Representation of a relation into a data base schema. We examinate now how to introduce a relation into a data base schema. We shall prove that several solutions are available but they are more or less efficient with regard of only information validation.

3.1. Decomposition process. Let R be a relation and I the set of integrity rules which are defined over R. The problem of the decomposition of R is to find a set of subrelations of $R(R_1, R_2 \ldots R_n)$ such $R = R_1 * R_2 \ldots * R_n$.

A decomposition of a relation R is obtained by considering the properties of FOHD as rules of decomposition.

Example 3.1. Let R' (STUDENT, INSTRUCTOR, SEMINAR, MONTH) be a subrelation of the relation R (example 2.3).

I contains the FDs (i_1) INSTRUCTOR, MONTH → SEMINAR
(i_2) STUDENT, MONTH → SEMINAR
and the FOHDS (i_3) SEMINAR, MONTH: STUDENT | INSTRUCTOR
(i_4) Φ: MONTH | INSTRUCTOR.

The following decompositions of R are available:

(D_1) R' = [INSTRUCTOR, MONTH, SEMINAR]R' * [INSTRUCTOR, MONTH, STUDENT]R' (by application of (18))

(D_2) = [STUDENT, MONTH, SEMINAR]R' * [STUDENT, MONTH, INSTRUCTOR]R'.

There are generally several available decompositions of the same relations R. The choice of one of them is the beginning step of designing data base schema. The chosen one is called the representation of R into the data base schema.

3.2. Malfunctions due to the data base schema. Malfunctions in the manipulation of a data base can result from the representation of a relation into the data base schema. They concern the integrity rules and the main operations create, delete, update an entity; they induce either information anomalies (like redundance or lost of information) or long executions of validation processes.

There are two types of such malfunctions for the FD, MD and FOHD integrity rules: within-relation malfunctions and between-relations malfunctions.

The first results concern only the FD integrity-rules [3]. We extend these results by considering MDs and FOHDs [16].

3.2.1. Within-relation malfunctions. Let us show them with an example.

Example 3.2.

R_1 (TYPE, BOOK, STUDENT) as defined in the example and the integrity rule TYPE →→ BOOK.

Furthermore we consider the following data-collection of R_1:

TYPE	BOOK	STUDENT
t_1	b_1	s_1
t_1	b_1	s_2
t_1	b_2	s_1
t_1	b_2	s_2
t_2	b_1	s_1
t_2	b_1	s_3

If the chosen representation (D_1) of R into the data base schema is R itself, so happen the following malfunctions about:

— creating entities.

● Indeed the only key of R is the whole combination of the three attributes. Therefore no attribute composing the primary key can have an undefined value; so is the information on students who read a book for a seminar, available only when, currently, at least one type for that student and that book is active.

● Otherwise, if we create the new entity (t_2, b_2, s_1), the validation process of the rule TYPE →→ BOOK must add the new entity (t_2, b_2, s_3) or reject the first one.

● At last, if we create the new entity (t_1, s_3) of the subrelation [TYPE, STUDENT]R_1, the validation process must immediately generate these new entities of R: (t_1, b_1, s_3) and (t_1, b_2, s_3).

— deleting entities

● When for instance the type t_1 is eliminated so are deleted all the entities composed with t_1 and consequently the following entities of the subrelations [BOOK, STUDENT]R_1 are definitively lost:

$$(b_1, s_2), (b_2, s_1) \text{ and } (b_2, s_2).$$

● Otherwise if we delete (t_1, b_1, s_1), the validation process must delete as well either the entity (t_1, b_1, s_2) or the other one (t_1, b_2, s_1).

— updating entities

● The updating of (t_1, b_1) into (t_1, b_4) induces the updating of as many entities of R as students who attend a seminar of type t_1.

● The other one of (b_1, s_1) into (b_1, s_4) induces a more complex validation process.

Remark. Another representation of R_1 is (D_2) [TYPE, BOOK]R_1 and [TYPE, STUDENT]R_1 because it is a decomposition of R_1. Thus, two data-collections are considered:

TYPE	BOOK
t_1	b_1
t_1	b_2
t_2	b_1

TYPE	STUDENT
t_1	s_1
t_1	s_2
t_2	s_1
t_2	s_3

In this case, all the previous malfunctions disappear; indeed the integrity rule TYPE →→ BOOK automatically is checked and no validation process is needed.

3.2.2. Between-relation malfunctions. Let us show them with an example.

Example 3.3. R_2(SEMINAR, TYPE, BOOK) as defined in the example 2.3 and the integrity rules (r_2) : SEMINAR → TYPE
(r_3) : TYPE →→ BOOK.

Furthermore we consider the following data-collection of R_1 which satisfies the integrity rules:

SEMINAR	TYPE	BOOK
s_1	t_1	b_1
s_2	t_1	b_2
s_1	t_1	b_2
s_2	t_1	b_1

The chosen representation (D_3) of R_2 into the data base schema consists of the two subrelations [SEMINAR, TYPE]R_2 and [SEMINAR, BOOK]R_2 which compose a decomposition of R_2. Here are their two data-collections:

SEMINAR	TYPE
s_1	t_1
s_2	t_1

SEMINAR	BOOK
s_1	b_1
s_2	b_2
s_1	b_2
s_2	b_1

Such a representation of R_2 induces malfunctions about
— creating new entities like
(t_1, b_3) creating the entities (s_1, b_3) and (s_2, b_3)
(s_3, t_1) also creating the entities (s_3, b_1) and (s_3, b_2)
— deleting entities like
(s_1, b_2) also deleting either (s_1, b_1) or (s_2, b_2)
(t_1, b_2) also deleting (s_1, b_2) and (s_2, b_2)
— updating entities like
(s_1, b_1) into (s_1, b_3) also updating (s_2, b_1) into (s_2, b_3)
(t_1, b_1) into (t_1, b_3) also updating (s_1, b_1) and (s_2, b_1) respectively into (s_1, b_3) and (s_2, b_3).

All these operations may be very long to be executed; they are induced by the validation process of the integrity rule (r_3) TYPE →→ BOOK which is defined through the two subrelations of R_2.

Remark. Another representation of R_2 could eliminate all these between-relations malfunctions:
(D_4) [SEMINAR, TYPE]R_2 and [TYPE, BOOK]R_2.
Indeed every integrity rule is no more defined through several subrelations but on only one subrelation; thus, no more validation process of (r_3) is needed.

3.3. Minimal representation of a relation into a data base schema. The previous examples show how the choice of a representation of R induces malfunctions and how one representation may eliminate all them

and does not need any validation process. We study now some relations with a set F of integrity rules which are supposed to be only FDs.
May a malfunction-free representation of such a relation exist?
[5] gave the answer when F is circuitless.

Definition. *a FD of $F: A$ of $\rightarrow B$ is transitive iff no FDs of F^+ are such as: $A \rightarrow C$ and $C \rightarrow B$.*

Theorem 1. *All FDs of F^* are transitive.*

Theorem 2. *Let $\{f_1, f_2 \ldots f_n\}$ be the FDs of F^*, I the key of R, $\{R_1, R_2 \ldots R_n\}$ the subrelations of R which are defined respectively on the attributes of $f_1, f_2 \ldots f_n$.*

Then either $R = R_1 * R_2 * \ldots R_n$ when I is included in some R_i or $R = I * R_1 * R_2 \ldots * R_n$ when not.

This decomposition is unique and called minimal decomposition.

Theorem 3. *When the minimal decomposition is chosen to represent a relation R into a data base schema, it includes no malfunctions about the FDs integrity rules.*

Remark. To obtain the minimal decomposition of R is equivalent to obtain the key of R and the elementary minimal covering F^* of F.

Example 3.4. Let us consider always the same relation about education program planning with the set F of only FDs integrity-rules.

F^*: SEMINAR $\rightarrow$ TYPE
INSTRUCTOR, MONTH $\rightarrow$ SEMINAR
SEMINAR $\rightarrow$ ROOM
STUDENT, MONTH $\rightarrow$ SEMINAR
INSTRUCTOR $\rightarrow$ RANK, SALARY
STUDENT, MONTH $\rightarrow$ SCORE

The only change concerns the last FD f; indeed if f_1 = (STUDENT, MONTH $\rightarrow$ SEMINAR) and f_2 = (STUDENT, MONTH, SEMINAR $\rightarrow$ SCORE) belong to F, then f may be formed and removes f_2 from F^*. The key of R is composed by (INSTRUCTOR, STUDENT, MONTH) and here is the minimal decomposition of R:

R = [INSTRUCTOR, STUDENT, MONTH]R * [SEMINAR, TYPE, ROOM]R * [INSTRUCTOR, MONTH, SEMINAR]R * [STUDENT, MONTH, SEMINAR, SCORE]R * [INSTRUCTOR, RANK, SALARY]R.

4. Conversion of a relation into a conceptual schema. There are several conceptual models but we introduce one [12] which uses as framework either relational or network or hierarchical model.

4.1. Conceptual model. Let $G(\mathcal{B}, \mathcal{L}, N, \gamma, \mu)$ be a graph where

— $\mathcal{B}$ is the set of block names: a block the name of which is B is a set of attributes $A(A_1, A_2, \ldots, A_n)$; B is always associated with one (or several) other set of attributes denoted $k(B)$ which is the key of B. It means that one instance of $k(B)$ corresponds to only one instance of B.

On one hand, $k(B)$ may not be included; in A and in this case the attributes which belong to $k(B) - (k(B) \wedge A)$ are named foreign key attributes ([3]). On the other hand, if $k(B)$ is included in A, then B is a name of a relation.

— $\mathcal{L}$ is a set of names which correspond to the names of links between blocks. The set $\mathcal{L}$ contains the element H to denote hierarchical link.

— N is the following set: $\{F, H, B\}$ (F functional, H hierarchical, B binary).

— γ is a partial application and μ a partial function:

$$\gamma : \mathcal{B} \times \mathcal{L} \rightarrow \mathcal{B}$$

$$\mu : \mathcal{B} \times \mathcal{B} \times \mathcal{L} \rightarrow N,$$

which are defined by the following table. Let B_1 and B_2 be the names of two blocks:

Conditions	γ	μ
$L_1 : k(B_1) \rightarrow k(B_2)$ elementary FD	$\gamma(B_1, L_1)=B_2$	$\mu(B_1, B_2, L_1)=F$
$k(B_1) \subset k(B_2)$	$\gamma(B_2, H)=B_1$	$\mu(B_2, B_1, H)=H$
$L_2[k(B_1), k(B_2)]$ •	$\gamma(B_1, L_2)=B_2$ $\gamma(B_2, L_2)=B_1$	$\mu(B_1, B_2, L_2)=B$ $\mu(B_2, B_1, L_2)=B$

Then the graph of a data-structure includes nodes which correspond to blocks edges which correspond to links.

An edge is directed from B_1 to B_2 iff $\gamma(B_1, L_1)=B_2$ and $\mu(B_1, B_2, L_1)=F$ or H; it is not directed iff $\gamma(B_1, L_1)=B_2$ and $\mu(B_1, B_2, L_1)=B$.

4.2. Conversion process of a relation representation into a conceptual schema. We will now show it with the same example of education program planning. The minimal decomposition was: $R=R_1 * R_2 * \ldots R_5$

R_1 : [INSTRUCTOR, STUDENT, MONTH]

R_2 : SEMINAR → TYPE, ROOM

R_3 : INSTRUCTOR, MONTH → SEMINAR

R_4 : STUDENT, MONTH → SEMINAR, SCORE

R_5 : INSTRUCTOR → RANK, SALARY

According to the previous table we can form the following graph where each node corresponds to a relation:

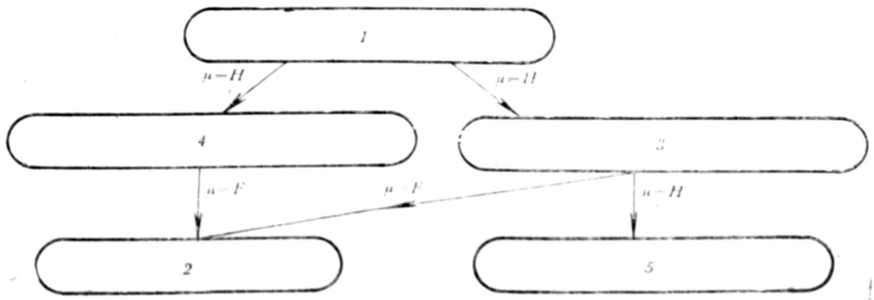

Fig. 2

1 — $\overset{*}{R}1$; INSTRUCTOR STUDENT MONTH ; 2— $\overset{*}{R_2}$: **SEMINAR** TYPE—ROOM 3— $\overset{*}{R3}$: **INSTRUKTOR MONTH** SEMINAR; 4— $\overset{*}{R_4}$: **STUDENT MONTH** SEMINAR SCORE; 5— $\overset{*}{R_5}$. **INSTRUKTOR** RANK SALARY

REFERENCES

1. J. Boitieaux. Etude mathématique d'un ensemble de notions. *Contrat DGRST* 67. 01. 015, 1969.
2. E. F. Codd. A relational model of data for large shared data banks. *Communs. ACM* **13**, 1970, 377—387.
3. E. F. Codd. Further normalization of the data base relational model. *Courant Computer Science Symposium 6, Data Base Systems*. New York, 1971, pp. 65—98.
4. C. Delobel. Aspects théoriques sur la structure de l'information dans une base de données. *RIRO*, **3**, 1971, p. 37—64.
5. C. Delobel, R. G. Casey. Decomposition of a data base and the theory of boolean switching functions. IBM Journal of Research and Development, **17**, 1973, 374—386.
6. C. Delobel, M. Leonard. The decomposition process in a relational model. *International Workshop on Data Structures, IRIA*, Namur (Belgium), May 1974.
7. W. W. Armstrong. Depending structure of data base relationship. *IFIP*, 1974, pp. 582—583.
8. C. P. Wang, H. H. Wedekind. Segment synthesis in logical data base design. IBM, Journal of Research and Development, **19**, 1975, 71—77.
9. H. A. Schmid, J. R. Swenson. On the semantics of the relational data model. *Proceeding ACM SIGMOD*, W. K. King (Ed.), San José, Calif., May 1975, 211—223.
10. J. M. Cadiou. On semantic issues in the relational model of data. *Proceeding International Symposium on Mathematical Foundations of Computer Science*, Gdánsk, Poland, September 1975.
11. P. A. Bernstein. Normalization and functional dependencies in the relational data base model. Ph. D. Thesis, University of Toronto, 1975.
12. M. Adiba, C. Delobel, M. Leonard. A unified approach for modelling data in logical data base design. *IFIP Workshop*, Freudenstadt (Germany), January 1976.
13. R. Fagin. Multivalued dependencies and a new normal form for relational data bases. IBM Research Report RJ (to appear in TODS).
14. J. Rissanen. Independent components of relation. *IBM Research Laboratory*, San José, October 1976.
15. M. Leonard, F. Reynaud. Existence du consensus et caractérisation des couvertures, et bases irrédondantes d'une fonction $\Sigma_i \mu_i A'_i$. *Discrete Mathematics* (to appear).
16. C. Zaniolo. Analysis and design of relational schemata for data base systems. Ph. D. Dissertation, UCLA, 1976.
17. P. A. Bernstein. Synthesizing third normal form relations from functional dependencies. *ACM TODS*, January 1977.
18. R. Fagin, J. H. Howard. A complete axiomatization for multivalued dependencies in a relational data base. *IBM Research Report, 1976*.
19. C. Delobel Sémantique des relations et processus de décomposition dans le modèle relationnel, *Research Report*. University I of Grenoble, September 1976.

Université Scientifique et
Médicale de Grenoble
BP 53 38041 Grenoble — Cedex

Received 11. 9 1977