Griffeth, Nancy D.

Working Paper

Decomposition and Recoverability of Relations

Discussion Paper, No. 345

Provided in Cooperation with:
Kellogg School of Management - Center for Mathematical Studies in Economics and Management Science, Northwestern University

Suggested Citation: Griffeth, Nancy D. (1978) : Decomposition and Recoverability of Relations, Discussion Paper, No. 345, Northwestern University, Kellogg School of Management, Center for Mathematical Studies in Economics and Management Science, Evanston, IL

This Version is available at:
https://hdl.handle.net/10419/220705

DISCUSSION PAPER NO. 345

Decomposition and Recoverability of Relations

by

Nancy D. Griffeth

Northwestern University
Graduate School of Management
Evanston, Illinois 60201

November, 1978

1. INTRODUCTION

The theory of relational databases owes its power to the ability to define and manipulate database integrity constraints, such as functional dependencies [C1, B1, W1], multi-valued dependencies [F1], and first-order hierarchical dependencies [D1]. The theory of dependencies is connected to the design of database systems by the following observation: each dependency corresponds to one or more possible decompositions of the database. Since each decomposition defines a different logical structure of the database, the dependencies determine the class of possible logical structures.

As an example, consider the following personnel database: the four attributes employee number (E), project number (P), project leader employee number (L) and department (D) are stored. Each employee works in a single department (in the notation of functional dependencies, $E \rightarrow D$ and $L \rightarrow D$) and all the employees in a department report to the same project leader ($DP \rightarrow L$). There are many possible decompositions of this database. Some of them are:

$R_1 = (E,D)$ $R_2 = (E,P,L)$

$R_1 = (L,D)$ $R_2 = (E,P,L)$

$R_1 = (D,L)$ $R_2 = (P,L)$ $R_3 = (E,D)$ $R_4 = (E,P)$

This list is not inclusive, of course. Generating an inclusive list of possible decompositions, would require the necessary and sufficient conditions under which a relation can be recovered from a decomposition

Rissanen has proved the necessary and sufficient conditions for a decomposition into two relations [R1]. In this paper, the necessary and sufficient conditions are given for decomposition into an arbitrary number of relations.

2. THE RELATIONAL MODEL

Let $\alpha = \{A, B, C, \ldots\}$ be a finite set of symbols, called attributes. Associated with each symbol is a set dom (A), dom (B), dom (C), etc., called a domain. A subset R(ABC...) of the Cartesian product of these domains is called a relation over α.

Let R denote a relation over α. For a subset β of α the relation R_β over β, called the projection of R on β, is defined as the image of R under the projection operator mapping the Cartesian product of the domains listed in α onto the Cartesian product of the domains listed in β.

If the projection $R_\alpha \rightarrow R_\beta$ has an inverse $R_\beta \rightarrow R_\alpha$, which maps each element of R_β to the unique element in R_α that projects to it, then $R_\beta \rightarrow R_\alpha$ and $R_\beta \rightarrow R_{\alpha-\beta}$ are functions. When the relations stored in a database are restricted to those such that $R_\beta \rightarrow R_\gamma$ is a function, then there is a functional dependency from β to γ, written

$$\beta \rightarrow \gamma$$

Armstrong [A1] calls a family $\mathcal{F}$ of functional dependencies full when it satisfies the following three statements: For all nonempty subsets β, γ, and δ of α:

(1) if $\beta \rightarrow \gamma$ and $\gamma \rightarrow \delta$ are in $\mathcal{F}$, then so is $\beta \rightarrow \delta$

(2) $\beta \subseteq \alpha$ implies $\beta \rightarrow \beta$ is in $\mathcal{F}$

(3) $\beta \rightarrow \gamma$ and $\beta \rightarrow \delta$ are in $\mathcal{F}$ if and only if $\beta \rightarrow \gamma \cup \delta$ is in $\mathcal{F}$

A full family of dependencies defines a database by defining the set of relations that may be stored in it. We denote by $\mathcal{R}(\mathcal{F})$ the set of all relations R such that $R_\beta \rightarrow R_\gamma$ is a function for every functional dependency $\beta \rightarrow \gamma$ in $\mathcal{F}$. A database Δ is a triple $\langle \alpha, \mathcal{F}, \mathcal{A} \rangle$ where α is a set of attributes, $\mathcal{F}$ a full family of functional dependencies, and $\mathcal{A}$ is a database schema. A database schema

is a set of subsets of the attribute set. If the database schema $\mathcal{A}$ is equal to $\{\alpha(1), \ldots, \alpha(n)\}$ then the stored database corresponding to relation $\mathcal{R}$ in $\mathcal{R}(\mathcal{T})$ is the set $\{R_{\alpha(1)}, \ldots, R_{\alpha(n)}\}$ of relations. The relation R is ideally recoverable from the stored database.

A relation is recovered from a set of stored relations by using the natural join defined by Codd [C1]. This is a product defined for any two "joinable" relations R and R′ over attribute sets α and α', respectively. The relations are joinable if $\beta = \alpha \cap \alpha'$ and $R_\beta = R'_\beta$. Let $\beta = \{B_1, \ldots, B_k\}$, $\alpha = \beta \cup \{A_1, \ldots, A_m\}$, and $\alpha' = \beta \cup \{A'_1, \ldots, A'_n\}$. Then the natural join $R*R'$ is the relation over $\alpha \cup \alpha'$ containing all tuples $(a_1, \ldots, a_m, b_1, \ldots, b_k, a'_1, \ldots, a'_n)$ for which $(a_1, \ldots, a_m, b_1, \ldots, b_k)$ is in R and $(b_1, \ldots, b_k, a'_1, \ldots, a'_n)$ is in R′.

A database schema $\mathcal{A} = \{\alpha(1), \ldots, \alpha(n)\}$ over attribute set α will be called a lossless schema for a full family $\mathcal{F}$ of functional dependencies over α if, for every relation R in $\mathcal{R}(\mathcal{F})$,

$$R = R_{\alpha(1)} * \cdots * R_{\alpha(n)}.$$

The following theorem, proved by Rissanen [R1], gives necessary and sufficient conditions for a database schema. $A = \{\beta_1, \beta_2\}$ to be a lossless schema.

THEOREM 1. Let α be a set of attributes. Then for all full families $\mathcal{T}$ over α and for all subsets β_1, β_2 of α such that $\alpha = \beta_1 \cup \beta_2$,

$\beta_1 \cap \beta_2 \rightarrow \beta_1$ or $\beta_1 \cap \beta_2 \rightarrow \beta_2$ is in $\mathcal{T}$ if and only if for all R in $\mathcal{R}(\mathcal{T})$

$$R = R_{\beta_1} * R_{\beta_2}.$$

To extend this result to arbitrary schemas $\mathcal{A}$, Rissanen points out that we can define a decomposition series consisting of pairwise decompositions of a relation into components. For example:

$$R = R_{01} * R_{02} = (R_{11} * R_{12}) * (R_{21} * R_{22})$$

where

$$R_{01} = R_{11} * R_{12}$$

$$R_{02} = R_{21} * R_{22}$$

Clearly, the original relation can be recovered from any such decomposition. However, as the example of section 3 shows, it is not true that all decompositions arise from successive pairwise decompositions.

3. A DECOMPOSITION WHICH IS NOT A SERIES OF PAIRWISE DECOMPOSITIONS

Consider the relation R(ABCDEFGHIJKL) together with the full family $\mathcal{F} = \{A \rightarrow CI,\ D \rightarrow EL,\ F \rightarrow BGH,\ BD \rightarrow AIJ\}^*$

It is shown below that there is no series of pairwise decompositions such that $R = R_{01} * R_{02} = (R_{11} * R_{12}) * (R_{21} * R_{22})$

with $R_{01} = R_{11} * R_{12}$, $R_{02} = R_{21} * R_{22}$

but in fact, for every R in $\mathcal{R}(\mathcal{F})$, it is nonetheless true that

$$R = RABCDIJ) * R(ABEFGH) * R(CDEFKL)$$

To see that there is no series of pairwise decompositions, it suffices to examine every natural join involving the three relations. These are:

$$R(ABCDEFGHIJ) \subsetneq R(ABCDIJ) * R(ABEFGH)$$

because $AB \rightarrow \beta$ in $\mathcal{J}$ iff $\beta \subseteq ABCI$

$$R(ABCDEFIJKL) \subsetneq R(ABCDIJ) * R(CDEFKL)$$

because $CD \rightarrow \beta$ in $\mathcal{J}$ iff $\beta \subseteq CDEL$

$$R(ABCDEFGHKL) \subsetneq R(ABEFGH) * R(CDEFKL)$$

because $EF \rightarrow \beta$ in $\mathcal{J}$ iff $\beta \subseteq BEFGH$

To see that R = R(ABCDIJ) * R(ABEFGH) * R(CDEFKL), take elements x_1, x_2, x_3 belonging to some R in $\mathcal{R}(\mathcal{F})$. If $\{X_1\}_{ABCDIJ} * \{X_2\}_{ABEFGH} * \{X_3\}_{CDEFKL} \neq \emptyset$, then the following equalities must hold:

$$\{X_1\}_{AB} = \{X_2\}_{AB}$$

$$\{X_1\}_{CD} = \{X_3\}_{CD}$$

$$\{X_2\}_{EF} = \{X_3\}_{EF}$$

Applying the functional dependencies A → CI, D → EL, and F → BGH gives the following equalities:

$$\{X_1\}_{ABCI} = \{X_2\}_{ABCI}$$

$$\{X_1\}_{CDEL} = \{X_3\}_{CDEL}$$

$$\{X_2\}_{BEFGH} = \{X_3\}_{BEFGH}$$

Using transitivity of equality, we find that:

$$\{X_1\}_{ABCEI} = \{X_2\}_{ABCEI}$$

$$\{X_1\}_{BCDEL} = \{X_3\}_{BCDEL}$$

$$\{X_2\}_{BCEFGH} = \{X_3\}_{BCEFGH}$$

Now applying the functional dependency BD → AIJ gives:

$$\{X_1\}_{ABCEI} = \{X_2\}_{ABCEI}$$

$$\{X_1\}_{ABCDEIJL} = \{X_3\}_{ABCDEIJL}$$

$$\{X_2\}_{BCEFGH} = \{X_3\}_{BCEFGH}$$

And finally applying transitivity one more time gives:

$$\{X_1\}_{ABCEI} = \{X_2\}_{ABCEI}$$

$$\{X_1\}_{ABCDEIJL} = \{X_3\}_{ABCDEIJL}$$

$$\{X_2\}_{ABCEFGHI} = \{X_3\}_{ABCEFGHI}$$

Thus, $\{X_3\}_{ABCDIJ} = \{X_1\}_{ABCDIJ}$ and $\{X_3\}_{ABEFGH} = \{X_2\}_{ABEFGH}$, so that

$$\{X_1\}_{ABCDIJ} * \{X_2\}_{ABEFGH} * \{X_3\}_{CDEFKL} = \{X_3\}.$$

Since this holds for arbitrary R in $\mathcal{R}(\mathcal{F})$ and for arbitrary elements x_1, x_2, and x_3

of R, it follows that for every R in $\mathcal{R}(\mathcal{J})$,

$$R(ABCDEFGHIJKL) = R(ABCDIJ) * R(ABEFGH) * R(CDEFKL)$$

for $\mathcal{F} = \{A \rightarrow CI,\ D \rightarrow EL,\ F \rightarrow BGH,\ BD \rightarrow AIJ\}$ * Note that the three relations defined are in fact independent components of R.

4. NECESSARY AND SUFFICIENT CONDITIONS FOR A DECOMPOSITION

To determine whether a set $\{\alpha(1), \ldots, \alpha(n)\}$ of subsets of the attribute set α gives a decomposition of $\mathcal{R}(\mathcal{F})$, we define the smallest family of equivalence relations $\{S_\beta \mid \beta \subseteq \alpha\}$ over the set $\{1, \ldots, n\}$ which satisfies the following conditions:

A1. For each subset β of α, if $\alpha(i) \cap \alpha(j) \rightarrow \beta$ is in $\mathcal{J}$, then (i,j) is in S_β.

A2. For each subset β of α, if (i,j) is in $S_{\{A\}}$ for all A in β, then $(i,j) \in S_\beta$

A3. For all subsets β and γ of α, if $\beta \rightarrow \gamma$ is in $\mathcal{F}$ and (i,j) is in S_β, then (i,j) is in S_γ.

A4. For each subset β of α, S_β is an equivalence relation.

This family will be called the <u>decomposition family</u> of equivalence relations for the set $\{\alpha(1), \ldots, \alpha(n)\}$ of subsets of α and the full family $\mathcal{J}$ of functional dependencies over α.

Then the necessary and sufficient conditions for $\{\alpha(1), \ldots, \alpha(n)\}$ to be a decomposition of $\mathcal{J}$ are given in Theorem 2.

THEOREM 2. Let $\mathcal{A} = \{\alpha(1), \ldots, \alpha(n)\}$ be a set of subsets of an attribute set α; let $\mathcal{F}$ be a full family of functional dependencies over α; and let $\{S_\beta \mid \beta \subseteq \alpha\}$ be the decomposition family of equivalence relations over $\mathcal{F}$ and $\mathcal{A}$. Then $\mathcal{A}$ is a lossless schema for α and $\mathcal{F}$ iff there is some $\alpha(i) \in \mathcal{A}$ such that for every $\alpha(j) \in \mathcal{A}$, (i,j) belongs to $S_{\alpha(j)}$.

Remark: For the case n = α, the decomposition family can be described as follows:

$S_\beta = \begin{matrix}10\\01\end{matrix}$ if $\alpha(1) \cap \alpha(2) \rightarrow \beta$ is not in $\mathcal{F}$

$S_\beta = \begin{matrix}11\\11\end{matrix}$ if $\alpha(1) \cap \alpha(2) \rightarrow \beta$ is in $\mathcal{F}$

Then the theorem states that $\{\alpha(1), \alpha(2)\}$ is a lossless schema for α and $\mathcal{F}$ if and only if (1,2) belongs to $S_{\alpha(1)}$ or (1,2) belongs to $S_{\alpha(2)}$. This condition holds if and only if $\alpha(1) \cap \alpha(2) \rightarrow \alpha(1)$ is in $\mathcal{F}$ or $\alpha(1) \cap \alpha\ (2) \rightarrow \alpha\ (2)$ is in $\mathcal{F}$

Proof of Theorem 2:

First, let us suppose that $\mathcal{A}$ is a lossless schema for α and $\mathcal{F}$, and prove that there must be some $\alpha(i)$ in $\mathcal{A}$ such that, for every $\alpha(j)$ in $\mathcal{A}$, (i,j) belongs to $S_{\alpha(j)}$.

By definition, $\mathcal{A} = \{\alpha(1), \ldots \alpha(n)\}$ is a lossless schema for α and $\mathcal{F}$ means that for every R in $\mathcal{R}(\mathcal{F})$,

$$R = R_{\alpha(1)} *\ldots* R_{\alpha(n)}.$$

Also for every subset S of R

$$S = S_{\alpha(1)} *\ldots* S_{\alpha(n)}$$

[This holds if $\mathcal{F}$ is a set of functional dependencies but need not hold if it is a set of multivalued dependencies or first-order hierarchical dependencies.]

The proof proceeds by constructing a relation $R = \{x_1,\ldots,x_n\}$ from the decomposition family for $\mathcal{A}$ and $\mathcal{F}$ and showing that R belongs to $\mathcal{R}(\mathcal{F})$. R is constructed so that $\{x_i\} = \{x_1\}_{\alpha(1)} *\ldots* \{x_n\}_{\alpha(n)}$; the theorem follows immediately.

Let $\mathcal{S} = \{S_\beta \mid \beta \subseteq \alpha\}$ be the decomposition family for $\mathcal{A} = \{\alpha(1), \ldots, \alpha(n)\}$ and $\mathcal{F}$. Define functions f_A: $\{1,\ldots, n\} \rightarrow$ dom (A) so that $f_A\ (i) = f_A(j)$ if and only if (i,j) is in $S_{\{A\}}$. (f_A is well-defined for each A as long as |dom (A)| is greater than or equal to the number of partitions induced by $S_{\{A\}}$.)

Let $R = \{x_1,\ldots, x_n\}$ be defined so that $\{x_i\}_{\{A\}} = \{f_A(i)\}$. Then the following conditions hold:

(1) $\{x_1\}_{\alpha(1)} *\cdots* \{x_n\}_{\alpha(n)} \neq \emptyset$, because by condition [A1] on $\mathscr{S}$, for all A in $\alpha(i) \cap \alpha(j)$, (i,j) is in $S_{\{A\}}$ and therefore $\{x_i\}_A = \{f_A(i)\} = \{f_A(j)\} = \{x_j\}_A$.

(2) $\{x_1, \ldots, x_n\}$ is in $\mathcal{R}(\mathcal{F})$, because by condition [A3] on $\mathscr{S}$, if $\beta \rightarrow \gamma$ is in $\mathcal{T}$ and if $\{x_i\}_\beta = \{x_j\}_\beta$, then $f_A(i) = f_A(j)$ for all A in β (implying that (i,j) is in S_β) and therefore (i,j) is in S_γ and consequently $\{x_i\}_\gamma = \{x_j\}_\gamma$.

It follows from (1), (2), and the hypothesis that $\mathcal{A}$ is a lossless schema for α and $\mathcal{F}$, that the following must hold: for some i

$$\{x_i\} = \{x_1\}_{\alpha(1)} *\cdots* \{x_n\}_{\alpha(n)}.$$

But this implies that for each j, $\{x_i\}_{\alpha(j)} = \{x_j\}_{\alpha(j)}$, and therefore $f_A(i) = f_A(j)$ for all $A \in \alpha(j)$ and therefore $(i,j) \in S_{\alpha(j)}$.

Conversely, let us suppose that there is some $\alpha(i)$ in $\mathcal{A}$ such that (i,j) belongs to $S_{\alpha(j)}$ for every $\alpha(j)$ in $\mathcal{A}$. We wish to show that $\mathcal{A}$ is a lossless schema for α and $\mathcal{F}$, i.e., that for every R in $\mathcal{R}(\mathcal{F})$

$$R = R_{\alpha(1)} *\cdots* R_{\alpha(n)}.$$

It suffices to show that for every R and every $x_1, \ldots, x_n$ in R, either $\{x_1\}_{\alpha(1)} *\cdots* \{x_n\}_{\alpha(n)} = \emptyset$ or $\{x_1\}_{\alpha(1)} *\cdots* \{x_A\}_{\alpha(n)} = \{x_i\}$ for some i.

If $\{x_1\}_{\alpha(1)} *\cdots* \{x_n\}_{\alpha(n)} \neq \emptyset$, then $\{x_i\}_{\alpha(i)\cap\alpha(j)} = \{x_j\}_{\alpha(i)\cap\alpha(j)}$ for all i,j.

Define a class $\mathcal{T} = \{T_\beta \mid \beta \subseteq \alpha\}$ of equivalence relations as follows:

(i,j) is in T_β if and only if

$$\{x_i\}_\beta = \{x_j\}_\beta.$$

$\mathcal{T}$ satisfies conditions A1 - A4, as shown below:

[A1]: If $\alpha(i) \cap \alpha(j) \rightarrow \beta$ is in $\mathcal{T}$, then $(x_i)_{\alpha(i)\cap\alpha(j)} = (x_j)_{\alpha(i)\cap\alpha(j)}$ and therefore $(x_i)_\beta = (x_j)_\beta$

[A2]: Trivial

[A3]: If $\beta \rightarrow \gamma$ is in $\mathcal{T}$ and if $(x_i)_\beta = (x_j)_\beta$ then $(x_i)_\gamma = (x_j)_\gamma$.

[A4]: By definition.

Since $\mathcal{S}$ is the smallest family of such relations, each relation S_β refines the corresponding relation T_β. Thus for i such that (i,j) is in $S_{\alpha(j)}$ for all j,

$$(x_i)_{\alpha(j)} = (x_j)_{\alpha(j)}.$$

Q.E.D.

$\alpha(1)$ = ABCDIJ

$\alpha(2)$ = ABEFGH

$\alpha(3)$ = CDEFKL

A.

	1	2	3
1	1^h	1^a	1^i
2	1^a	1^h	1^g
3	1^i	1^g	1^h

B.

	1	2	3
1	1^h	1^a	1^g
2	1^a	1^h	1^f
3	1^g	1^f	1^h

C.

	1	2	3
1	1^h	1^d	1^b
2	1^d	1^h	1^g
3	1^b	1^g	1^h

D.

	1	2	3
1	1^h	0	1^b
2	0	1^h	0
3	1^b	0	1^h

E.

	1	2	3
1	1^h	1^g	1^e
2	1^g	1^h	1^c
3	1^e	1^c	1^h

F.

	1	2	3
1	1^h	0	0
2	0	1^h	1^c
3	0	1^c	1^h

G.

	1	2	3
1	1^h	0	0
2	0	1^h	1^f
3	0	1^f	1^h

H.

	1	2	3
1	1^h	0	0
2	0	1^h	1^f
3	0	1^f	1^h

I.

	1	2	3
1	1^h	1^d	1^i
2	1^d	1^h	1^g
3	1^i	1^g	1^h

J.

	1	2	3
1	1^h	0	1^i
2	0	1^h	0
3	1^i	0	1^h

K.

	1	2	3
1	1^h	0	0
2	0	1^h	0
3	0	0	1^h

L.

	1	2	3
1	1^h	0	1^e
2	0	1^h	0
3	1^c	0	1^h

a. $\alpha(1) \cap \alpha(z) = \{A,B\}$

b. $\alpha(1) \cap \alpha(3) = \{C,D\}$

c. $\alpha(2) \cap \alpha(3) = \{E,F\}$

d. $A \rightarrow CI$ and $A \in \alpha(1) \cap \alpha(2)$

e. $D \rightarrow EL$ and $D \in \alpha(1) \cap \alpha(3)$

f. $F \rightarrow BGH$ and $F \in \alpha(2) \cap \alpha(3)$

g. Transitivity

h. Reflexivity

i. $BD \rightarrow AIJ$ and $(1,3) \in S_{BD}$

	1	2	3
1	1*	1*	0
2	1*	1*	0
3	0	0	1*

$\beta \subseteq$ ABCEI

	1	2	3
1	1*	0	0
2	0	1*	1*
3	0	1*	1*

$\beta \subseteq$ ABEFGHI

	1	2	3
1	1*	0	1*
2	0	1*	0
3	1*	0	1*

$\beta \subseteq$ ABCDEIJL

* For all β, if $(i,j) \in R_B$ for all $B \leftarrow \beta$ then $(i,j) \in R_\beta$

Point: $S_{\alpha(1)} = S_{ABCDIJ} = \begin{matrix} 101 \\ 010 \\ 101 \end{matrix}$

$S_{\alpha(2)} = S_{ABEFGH} = \begin{matrix} 100 \\ 011 \\ 011 \end{matrix}$

$S_{\alpha(3)} = S_{CDEFKL} = \begin{matrix} 100 \\ 010 \\ 001 \end{matrix}$

There exists some j such that for every i, $(i,j) \in S_{\alpha(i)}$.

References

[1] Bernstein, P. A. "Synthesizing third normal form relations from functional dependencies", ACM Trans on Database Systems 1, 4 (December, 1976) 277-298

[2] Codd, E. F. "Further normalization of the relational model", in Data-base systems, ed. by Randall Rustin. Prentice-Hall, 1975

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

[4] Fagin, R. "Multivalued dependencies and a new mormal form for relational databases", ACM Trans Database Systems 2,3 (September, 1977) 262-278

[5] Rissanen, J. "Independent Components of Relations", ACM Trans Database Systems 2,4 (December, 1977) 317-325

[6] Wang, C. P., and Wedekind, H. H., "Segment systhesis in logical database design", IBM J Research and Development 19,1 (January, 1975), 71-77