Journal of the Korean
OR/MS Society
Vol. 13 No. 1 June 1988

—Research Paper—

An Approach to Automatic Generation of Fourth Normal Form for Relational Database

Sung Joo Park,*
Young Gun Lee,**
Hyung Rae Cho*

Abstract

A new approach to the logical design of 4NF database shceme, which can be easily automated, is proposed. The main features of the approach are ; introduction of a single attribute right hand side, extension of the concept of independent relations, semantic analysis, and adoption of dpendency matrix. The underlying viewpoints of functional relationships of the approach are different from Fagin's in that we distinguish functional and multivalued dependency in terms of cardinality. An algorithm for automatic generation of fourth normal form is presented and implemented.

Key words and Phrases : database design, multivalued dependency, functional dependency, fourth normal form, 4NF, Boyce-Codd normal form, decomposition, relational database, synthesis algorithm

I. INTRODUCTION

Data relations often possess undesirable structural characteristics--data redundancy and update anomalies. Since the introduction of relational database model by Codd[12], many approaches to the design of the relational database model have been proposed to eliminate or reduce data redundancy and update anomalies[11,12,17,18,23,26].

Fagin[20] introduced a type of functional dependency named multivalued dependencies lead to fourth normal form for relational database. Roughly speaking, it is said that a relational schema is in fourth normal form if all depenencies (functional and multivalued) are the results of keys. The multivalued dependencies are similar to functional dependencies except that the former has Complementation as one of formal properties in addition to the properties of functional dependencies.

* Department of Management Science, Korea Advanced Institute of Science & Technology
** Korea Advanced Energy Research Institute

It is well known that the fourth normal form is strictly stronger than both third normal form and Boyce-Codd normal form, and every relation schema can be decomposed into a family of relational schema in fourth normal form without loss of information. Fagin's decomposition algorithm based on MVD takes recursive steps until there exists no MVD. The concept of his MVD in itself is clear and simple but the algorithm for decompostition procedure is not well established.

A new synthetic approach to eliminate or reduce the complexities in fuctional relationships is proposed in this paper. The result of the approach is fourth normal form relations. The approach has been well tailored for ease of implementation on computers by introducing several concepts, some are new and others are extension of existing ones. The features of the new approach can be summarized as follows ;

First, the approach has introduced the notion of single attribute RHSs. When we describe data dependencies(DDs), their RHSs are single attribute, where data dependency (DD) refers to all types of functional relationship, FDs and MVDs.

Second, the concept of independent relations, which was once mentioned as trifle by Fagin, has been reinterpreted from a different viewpoint in terms of cardinalities.

As the third feature of the approach, we provide a device for the analysis of semantic validity of user's logical navigation among relations of a database.

Last is the adoption of dependency matrix(DM). All the functional relationships described in a conventional manner are stored in a dependency matrix, which is a powerful mechanism for synthesizing normal forms from data dependencies.

II. OVERVIEW OF DATA DEPENDENCY

2.1 Functional Dependencies

An FD is a sentence denoted X−>Y (read "X determines Y" or "Y is functionally dependent on X").

An FD X−>Y holds in R(U) where X and Y are subsets of U(read "universe"), if for every tuple u and v of R, u[X]=v[X] implies u[Y]=v[Y] (u[X] denotes the projection of the tuple u on X). A full FD X−>Y is and FD such that there exists no proper subset $X' \subseteq X$ with X'−>Y. An FD X−>Y is trivial if $X \supseteq Y$.

Let X, Y and Z be nonempty subset of U. Then the FDs of a relation R(U) have the following properties.

FD1. Reflexivity : If $X \subseteq U$, then X−>X.
FD2. Augmentation : X−>X implies XZ−>X.

FD3. Transitivity : X−>Y and Y−>Z imply X−>Z.
FD4. Pseudotransitivity : X−>Y and YZ−>W imply XZ−>W.
FD5. Additivity(Union) : X−>Y and X−>Z imply X−>YZ.
FD5. Projectivity(Decomposition) : X−>YZ implies X−>Y and symmetrically X−>Z.

A family of FDs is full when it satisfies the FD1 through FD3.

2.2 Multivalued Dependencies

A multivalued dependency on a set of U is a statement X−>−>Y(read "X multidetermine Y" or "Y is multidependent on X") where X and Y are subsets of U.

X−>−>Y holds for the relation R(X, Y, Z) if and only if R is the join of its projection R1(X, Y) and R2(X, Z). Since the right hand side of "if and only if" in the previous sentence is symmetry in the role of Y, Z, the following rule (called "complementation") holds, too. X−>−>Y holds for R(X, Y, Z) if and only if X−>−>Z holds.

A relation R is in fourth normal form (4NF) if and only if, whenever there exists an MVD in R, say K−>−>B, then all attributes of R are also functionally dependent on K (i.e., K−>A for all attribute A of R). In other words, all the dependencies are the results of keys.

The following set of inference rules is applied to MVD.

MVD0. Complementation : If XYZ=U, Z=U−XY, and X−>−>Y then X−>−>Z holds.
MVD1. Reflexivity : If X⊆U, then X−>−>X.
MVD2. Augmentation : If X−>−>Y then XZ−>−>Y.
MVD3. Transitivity : If X−>−>Y and Y−>−>Z, then X−>−>Z−Y.
MVD4. Pseudotransitivity : If X−>−>Y and YW−>−>Z, then XW−>−>Z.
MVD5. Additivity(Union) : If X−>−>Y and X−>−>Z, then X−>−>YZ.
MVD6. Projectivity(Decomposition) : If X−>−>Y and X−>−>Z, then X−>−>Y∩Z and X−>−>Y−Z.

A family of MVDs is full when it satisfies the MVD0 through MVD3. Complementation holds for MVD alone.

III. DESIGN FEATURES OF THE PROPOSED APPROACH

3.1 Single Attribute RHS

In the conventional approaches to the logical design of relational database except the one proposed by Rissanen, RHS of a DD consists of attributes (hereafter called "composite RHS"). These apporaches cause users to suffer from the complexity or difficulties in describing a set of DDs from which they orgainize relation schema. In addition, the description of DDs by these approaches are likely to contain many redundant FDs or redundant attributes in the DDs because of the composite RHSs of DDs.

Returning to the inference rules of FDs, Additivity, which is in a sense reverse Projectity, permits an expansion of the domain of RHSs of any FD. For this reason, the generation of a family of FDs can be performed among those FDs whose RHS is small as possible, i.e., single attribute.

Single attribute RHS holds for MVDs, too. For example, let X, Y, Z be disjoint sets of attributes such that XYZ=U. Here we have two cases : one is the case in which X−>−>Y holds and the other X−>−>YZ holds and Y and Z are not independent (in this case Fagin says (Y, Z) a "cluster"). In former case, we can decompose XYZ into X−>−>Y and X−>−>Z by

Complementation. In latter case, Z is jointly dependent on X and Y so we can not decompose further but we can describe XY->->Z.

An MVD having composite RHS can be converted to an equivalent DD having single RHS following the previous procedure. For example, from the relation SIQ(SUPPLIER#, ITEM#, QTY) in Figure 1 which is said by Fagin to have an MVD SUPPLIER#->->ITEM#, QTY, we can derive a DD SUPPLIER#, QTY->->ITEM//.

SUPPLIER #	ITEM #	QTY
S-A-001	PA-1	100
S-A-001	PA-2	60
S-A-001	PA-1	60
S-A-002	PA-1	10
S-A-002	PA-2	60
S-A-003	PA-1	100
S-A-003	PA-1	50

Figure 1 An instance of Relation SIQ

In our approach, the number of distinct RHSs is clearly the same as that of attributes at most and thus scanning efforts to describe data dependencies are limited to LHS. By describing the RHSs of DDs as single attribute, we can easily describe DDs as input to the logical design of database and procedures to construct relational schema are simpler than the existing approaches. With this method, we always obtain a right-reduced set of DDs in the first step of the algorithm for synthesizing fourth normal forms.

The nature of single attribute RHS is quite different from that of right-reduced. The former, in describing DDs, always refers RHSs which are composed of only one attribute but the latter implies those having no redundant attribute(s) in RHS. Having no redundant attribute(s) doesn't mean that RHS consist of one attribute.

3.2 Independent Relations

According Fagin's MVD, MVD X->->Y holds for R(X, Y, Z) if Y and Z are independent or orthogonal. In a relation MCE(MANAGER, CHILD, EMPLOYEE), MANAGER ->-> CHILD and MANAGER->->EMPLOYEE will hold if CHILD and EMPLOYEE are independent. A relation MC(MANAGER, CHILD) and a relation ME(MANAGER, EMPLOYEE) are essentially independent relations.

We now extend the concept of independent relations by applying it to FDs, too. Suppose a relation SWH(STUDENT, WEIGHT, HEIGHT). There exist two FDs in SWH, STUDENT->WEIGHT and STUDENT->HEIGHT, and WEIGHT and HEIGHT can be considered to be independent. We, therefore, have the relation schema which consists of SW(STUDENT, WEIGHT) and SH(STUDENT, HEIGHT), where SW and SH are mutually independent. SW and SE, however, unlike MC and ME, can be merged in a relation scheme SWH(STUDENT, WEIGHT, HEIGHT) without redundancy. let | C | be cardinality of a relation C. As the cardinality | SW | and | SH | are exactly the same in nature, there can be no redundancy when we merge SW and SH in SWH(STUDENT, WEIGHT, HEIGHT). But this is not the same with MC and ME. that is MVDs. Though MC and ME are independent, | MC | and | ME | are not the same in nature and merging them in one relation scheme will cause many redundanies. This concept of independency in terms of cardinalities can be a basis for synthesizing normal forms. Consequenty, we distinguish FDs and MVDs on this basis on the contrary to Fagin who calls FD a "special case" of MVD.

Figures 2 and 3 are instances of MC and ME, respectively. If we produce MCE(MANAGER, CHILD, EMPLOYEE) by merging MC and ME, | MCE | will be 10. An instance of MCE is given in Figure 4.

MANAGER	CHILD
Smith	Tom
Smith	Greta
Gauss	Peter
Hilbert	Hubert
Hilbert	Shelly
Hilbert	John

Figure 2 An instance of T1(MANAGER, CHILD)

MANAGER	EMPLOYEE
Smith	Kennedy
Gauss	Johnson
Gauss	Hoseph
Hilbert	Pillip
Hilbert	Winston

Figure 3 An instance of T2(MANAGER, EMPLOYEE)

MANAGER	CHILD	EMPLOYEE
Smith	Tom	Kennedy
Smith	Greta	Kennedy
Gauss	Peter	Johnson
Gauss	Peter	Joseph
Hilbert	Huberet	Phillip
Hilbert	Hubert	Winston
Hilbert	Shelly	Phillip
Hilbert	Shelly	Winston
Hilbert	John	Phillip
Hilbert	John	Winston

Figure 4 An instance of T3(MANAGER, CHILD, EMPLOYEE)

3.3 Sematic Analysis

In most approaches to relational database design, semantic connection between relations are left implicitly. The users need to specify logical access paths. This tends to encourage the database users to associate two relations incorrectly by performing lossy or meaningless joins.

The proposed system constructs a database from the given set of DDs to be used for the analysis of semantic validity. The database stores infromations on the relational schema and relationships among relations constructed from the given set of DDs. In other words, interrelational semantics and functional relationships are exlicitly described. Users, therefore, are relieved of efforts to find paths for logical navigation among relations. User's logical navigations undergo the system's analysis of semantic validity to see if there are meaningless or lossy operations in user's request.

Since an attribute corresponds to distinct column of a table, its name should be unique for the description of DDs. The treatment of DDs is essentially syntactic based on Amstrong's axioms, and hence the uniqueness assumption is required also in syntactic approaches. In this regard, the synthesizing algorithm of the proposed approach is syntactic, but resulting system secures semantic connections of the relational schema, which will relieve users of the database system of logical navigation among the relations to perform proper joins and projections.

3.4 Dependency Matrix

To synthesize normal forms from a set of DDs, dependency matrix(DM) is used which stores informations representing relationships of each attribute with others in the proper cells. Interpretation of symbols stored in a cell of DM of a relation R(A1, A2, ...,An) are as follows ;

(1) If DMij has a symbol "f", then Ai−>Aj.
(2) If DMij has a symbol "m", then Ai−>−>Aj.
(3) If symbol of DMij is "c", then there are more than one cell on column j whose symbol is "c" and Aj is functionally dependent on the composite LHS composed of all Ai such that DMij's symbol is "c".
(4) If symbol of DMij is "n", FD becomes MVD in(3).
(5) If DMij blank, there is no functional relationship between Ai and Aj.

Before assigning symbols to each cell we have to determine the type of dependencies. FDs create no difficulties in determining the dependency type, but a great effort is needed to find out MVDs. Let's recall two propositions of MVD, and let X,Y and Z be disjoint sets whose union is U.

(1) For every relation R(U), the MVD X−>−>Y holds if and only if
R(U)=R[XY] * R[XZ]
(2) For every relation R[U], the MVD X−>−>Y holds if and only if Complementation, X−>−>Z holds.

If we follow the first porposition to determine an MVD, we have to actually perform the relational operations, projection and join, to see if the result of operations satisfies the proposition. For the second proposition, we have to test if Complementation holds.

However, the notion of repeation group(RG) is helpful for the determination of MVD. If an LHS K of an FD has a set of values over a domain dom(A), the MVD K−>−>A exists. This notion of RG, which is COBOL-like representation of an MVD, implies that RG is independent of other RHSs. Furthermore, we are intuitively certain that cardinality of R1(K,A) over RG is not equal to that of R2(K,B) over an K−>B.

In the relation ESD(EMPL#, SALARY, DEPENDENT) given in Figure 5, each EMPL # has a set of DEPENDENT, so DEPENDENT is an RG. In other words, each employee has one salary but several dependents. This fact is resulted from the independence between DEPENDENT and SALARY. Now we can easily determine an MVD EMPL#−>−>DEPENDENT.

EMPL #	SALARY	DEPENDENT
0001	$ 30,000	John
0001	$ 30,000	James
0002	$ 25,000	Grunt
0003	$ 28,000	Bill
0003	$ 28,000	Albert
0003	$ 28,000	Thomas

Figure 5 An instance of the relation ESD

IV. AUTOMATIC GENERATION OF 4NF RELATIONS

4.1 Construction of Dependency Matrix

Dependency matrix is a useful device for the implementation of the proposed approach. Input to the synthesizing algorithm is a set of DDs whose RHSs are all single attributes. The cardinality (or order) of DM is the same as the number of distinct attributes contained in the given set of DDs over relation R(U), U={A1, A2, ...An}. The original DM is reserved for later use when users retrieve information from the database designed by this system. A copied DM is used through the procedures for synthesizing fourth normal forms because deletions destroy the original status.

Each attribute, Ai, for i=1 to n, is assigned to i−th row and i−th column. Scanning column by column(that is, attribute by attribute) is performed to synthesize DDs whose RHS is the current attribute. This provedure guarantees the obtainmetnt of DDs having single attribute RHSs.

Our dependency matrix is somewhat different from Teorey's functional dependency matrix [13] in that Teorey's FD matrix is the transitive closure and its cardinality can be greater than the number of attributes concerned.

4.2 Sythesizing Fourth Normal Forms

Proposed synthesizing algorithm for synthesizing 4NF from the DMs is as follows ;

STEP-1. (Removal of transitive dependencies) Examine each attribute if there exist transitive dependencies (TDs). If any, remove them.

STEP-2. (Partitioning DDs) For each attribute, examine if it has data dependencies (FDs or MVDs) on other attributes. If current attribute is a RHS of a certain DD, then assign proper symbols (described in section 3.4) to the relevant cell and then label all the attributes involoved in this DD, both those contained in LHS and the one in RHS.

STEP-3. (Test of nonfunctional relationship) Search and count labelled attributes. If no attributes are labelled, constitute a relation scheme consisting of a single key relation, and go to STEP-7.

STEP-4. (Merging FDs) Merge those FDs whose LHSs are identical into a single FD.

STEP-5. (Deletion of supersets) For the FDs and MVDs whose RHSs are identical, examine if any DD exists whose LHS is superset of that of other DDs. If any, delete such superset DDs.

STEP-6. (Deletion of subset relations) Construct relations from the reduced set of DDs and test if one relation is subset of any other relations. If any, delete the subset relation.

STEP-7. (Construction of final relational schema and database) Construct final relational schema and database of information on the functional relationships among attributes and interrelational connections.

We begin synthesizing 4NF by first removing TDs in STEP-1. Removal of TDs relieves us of handling TDs in succeeding steps, which has become possible by the adoption of DM and single attribute RHSs. Let F, as an example, be the set of DDs A->->B, A->D, A->E, B->C, D->E and A,C->F. Figures 6 and 7 exhibit the contents of example DM over F before and after STEP-1, respectively.

The status of attributes can be interpreted from the DM.

1. If at least one of DMij are not null for j=1...n and fixed i, then the attribute assigned to the i-th colum is a prime attribute (in Figure 6, A, B, C and D).
2. If all of DMij are null for j=1...n and fixed i, then the attribute assigned to the i-th column is a nonprime attribute(in Figure 6, E and F).
3. If all of DMij are null for j=1...n and fixed j, then the attribute assigned to the j-th column is a prime attribute and never appears in RHSs. This type of attribute plays the role of a root attribute in the interrelational connections of final relational schema (in Figure 6, A).

A	B	C	D	E	F
	m		f	f	c
		f			
					c
				f	

Figure 6 DM for F before STEP-1

A	B	C	D	E	F
	m		f		c
		f			
					c
				f	

Figure 7 DM for F after STEP-1

In STEP-2, we partition all the data dependencies into groups of FDs and MVDs. For FDs, they are partitioned again into groups such that all FDs in each group have identical LHSs. The attributes, which are contained in either LHS, are labelled for later uses in STEP-3.

STEP-3 deals with nonfunctional relationships. If no attributes are labelled, then they are all key, and a relational schema containing single relation is constructed. In other words, if there are no data dependencies among all the attributes of a relation, they form a relational schema consisting of single relation whose candidate key is all key.

In STEP-4, we apply Additivity rule to remove redundancies which will otherwise arise in relational schema. Let F1 be A−>B and F2 A−>C. We can merge F1 and F2 in F3 A−>BC, and R1(A, B) and R2(A, C) in one relation R(A, B, C) without redundancy as well. As the cardinalities | R1 |, | R2 | and | R3 | are exactly same in nature, there can be no redundancy when we merge R1 and R2 in R(A, B, C). But the case is not the same with MVDs. Let MVDs, M1 D−>−>E, M2 D−>−>F and relations over them T1(D, E), T2(D, F) and T3(D, E, F), the | T3 | is always larger than both | T1 | and | T2 |. | T3 | is obtained by the following expression. Hence merging them in one relation scheme, T3(D, E, F), will cause many redundancies.

$$|T3| = \sum_{j=1}^{n} |T1j| * |T2j|, \quad n = |R(D)|$$

STEP-5 as well as STEP-4 contributes to eliminate redundancies from the relation schema. We examine, among the FDs whose RHSs are identical, if there exists any FD whose LHS is a superset of that of other FDs. If any, we delete it. Let F1 AB−>C and F2 ABD−>C. F2 is derived by Augmentation from F1, hence F2 has redundant attribute, D, in LHS.

In STEP-6, we construct intermediate relational schema and investigate whether there exists any relation which is subset of other relation. If we find one, we delete it because it is duplicated.

In STEP-7, we construct final relational schema in fourth normal form from the synthesized relations over the set of DDs in the preceding steps. A database of interrelational semantics and functional relationships of attributes of each relation is constructed.

4.3 Non-redundancy of Relational Schema

Our synthesizing algorithm guarantees the non-redundancy of the relational schema constructed from the give set of DDs. The proposed algorithm eliminates transitive dependenies including Pseudo-transitivity in STEP-1. In STEP-3, by applying Reflexivity, we resolve the problem of nonfunctional relationship. STEP-5 searches and removes those DDs which can be derived by applying Augmentation rule.

For MVDs, Complementation still remains to be applied. Complementaion holds only when it is based on independency of attributes. However, since description of DDs based on the single attribute RHS and the concept of independent relations in STEP-1 implies the application of Complementation, the redundancies over MVDs are eliminated.

4.4 Implementation

We have implemented a system, AGNOF(Automatic Generator of Fourth Normal Forms), of logical design of relational databases in order to verify the accuracy and efficiency of our algorithm. A sample run was executed with the case problem given in Fagin [20].

4.4.1 User Interface

The system interfaces with users for acquistion of data and for generation of documents. User's data are a set of DDs of relation R(U). Documents are intermediate process summary, the contents of database of interrelational semantics, and functional relationship among attributes. Figure 8 is an example of the conversational user interface of the AGNOF.

```
Type in attribute names one by one.
If no more attribute, enter space.

Name of Attribute #01 : CLASS
Name of Attribute #02 : SECTION
Name of Attribute #03 : STUDENT
Name of Attribute #04 : MAJOR
Name of Attribute #05 : EXAM MARK
Name of Attribute #06 : EXAM RANK
Name of Attribute #07 : YEAR
Name of Attribute #08 : INSTRUCTOR
Name of Attribute #09 : RANK
Name of Attribute #10 : SALARY
Name of Attribute #11 : TEXT
Name of Attribute #12 : DAY
Name of Attribute #13 : ROOM
Name of Attribute #14 :

Enter DDs of each attribute. Current attribute is RHS.
(Type of DDs and attribute names contained in LHS)

Current attribute is CLASS
If no more DD, then enter space.
              Type of DD :

Current attribute is SECTION
If no more DD, then enter space.
              Type of DD :

Current attribute is STUDENT
```

If no more DD, then enter space.
Type of DD : M
#s of attributes in LHS : 1 2
Type of next DD :

Current attribute is MAJOR
If no more DD, then enter space.
Type of DD : F
#s of attributes in LHS : 3
Type of next DD :

Figure 8 An example of Q-A mode

4.4.2 Document Generation

The system produces documents, on user's request, including the follwing reports.

(1) Intermediate Process Summay
- Dependency Matrices before and after the removal of transitive dependencies in STEP-1
- Partitioned groups of both FDs and MVDs
- Merged FDs from original FDs
- Deleted supersets of both FDs and MVDs
- Deleted relation of intermediate relational schema

(2) Final Relational Schema
- Scheme of each relation
  - Attribute names of each relation
  - Candidate key of each relation
- Interrelational connections

An example of generated documents are given in Figures 9 and 10.

DM AFTER REMOVAL OF TDs (TABELE-1)
('*' INDICATES DELETED TDs)

	A01	A02	A03	A04	A05	A06	A07	A08	A09	A10	A11	A12	A13
A01	0	0	N	0	C	*	0	C	0	0	M	N	C
A02	0	0	N	0	C	*	0	C	0	0	0	N	C
A03	0	0	0	F	C	*	F	0	0	0	0	0	0
A04	0	0	0	0	0	0	0	0	0	0	0	0	0
A05	0	0	0	0	0	0	0	0	0	0	0	0	0
A06	0	0	0	0	0	F	0	0	0	0	0	0	0

A07	0	0	0	0	0	0	0	0	0	0	0	0	0
A08	0	0	0	0	0	0	0	0	F	F	0	0	0
A09	0	0	0	0	0	0	0	0	0	0	0	0	0
A10	0	0	0	0	0	0	0	0	0	0	0	0	0
A11	0	0	0	0	0	0	0	0	0	0	0	0	0
A12	0	0	0	0	0	0	0	0	0	0	0	0	C

PARTITIONED SET OF FDs

STUDENT->MAJOR
CLASS, SECTION, STUDENT->EXAM_MARK
EXAM_MARK->EXAM_RANK
STUDENT->YEAR
CLASS, SECTION->INSTRUCTOR
INSTRUCTOR->SALARY
INSTRUCTOR->RANK
CLASS, SECTION, DAY->ROOM

PARTITIONED SET OF MVDs

CLASS, SECTION-->STUDENT
CLASS-->TEXT
CLASS-->DAY

Figure 9 An example of Generated Documents
(Intermediate Process Summary)

INTERRELATIONAL CONNECTIONS

	R01	R02	R03	R04	R05	R06	R07
R01	0	1	1	1	1	1	0
R02	1	0	0	0	0	0	0
R03	1	0	0	1	1	0	1
R04	1	0	1	0	1	0	0
R05	1	0	1	1	0	0	0
R06	1	0	0	0	0	0	0
R07	0	0	1	0	0	0	0

FINAL RELATIONAL SCHEMA OVER GIVEN SET OF DDs

("*" INDICATES PRIME ATTRIBUTE)

RELATION-01 *CLASS, *SECTION, *STUDENT, EXAM_MARK
RELATION-02 *EXAM_MARK EXAM_RANK
RELATION-03 *CLASS, *SECTION, *STUDENT, INSTRUCTOR
RELATION-04 *CLASS, *SECTION, *DAY, ROOM
RELATION-05 *CLASS, *TEXT
RELATION-06 *STUDENT, MAJOR, YEAR
RELATION-07 *INSTRUCTOR, SALARY, RANK

Figure An example of Generated Documents
(Final Relational Schema)

V. CONCLUSION

A new approach to the logical design for relational database model and implementation of an experimental system, AGNOF(Automatic Generator of Fouth Normal Forms) are presented in this paper. In the approach, functional relationships are essentially the only inputs. Concept of single attribute RHS through which users always obtain right-reduced set of given DDs is introduced which takes user's conveniences into consideration. We also extend the concept of independent relations by applying it to FDs as well as MVDs. These two concepts enhance the capabilities of describing the DDs for the implementation of automatic generator.

The approach presented in this paper is an effort to perform semantic analysis capable of relieving user's burden on logical navigation among relations of a relational database, which works properly and effectively through a powerful mechanism, Dependency Matrix.

REFERENCES

[1] A. K. Arora, and C. R Carlson : Normalization could be Useful. The Computer Journal 27(1), (1984).

[2] A. O. Mendelzon : On Axiomatizing Multivalued Dependencies in Relational Databases. Journal of the Association for Computing Machinery. 26(1), (1979).

[3] C. Delobel : Normalization and Hierarchical Dependencies in the Relational Data Model. ACM Transations on Database Systems. 3(3), (1978).

[4] C. J. Date : An Introduction to Database Systems. Addison-Wesley Publishing Company. (1982).

[5] C. R. Carlson, and A. K. Arora : The Application of Functional Dependency Theory to Relational Databases. The Computer Journal. 25(1), (1982).

[6] C. Zaniolo, and M. A. Melkanoff : On the Design of Relational Databases. ACM Transactions on Database Systems. 6(1), (1981).
[7] D. D. Chamberlin : Relational Data-Base Management Systems. Computing Surveys. 8(1). (1976).
[8] D. Kroenke : Database Processing. Science Research Associates, Inc. (1983).
[9] D. Maier : Minimum Covers in the Relational Database Model. Journal of the Association for Computing Machinery. 27(4), (1980).
[10] D. Maier : The Theory of Relational Databases. Computer Science Press. (1983).
[11] D. Maier, J. D. Ullman, and M. Y. Vardi : On the Foundation of the Universal Relational model, ACM Transations on Database System. 9(2), (1984).
[12] E. F. Codd : A Relational Model of Data for Large Shared Data Banks. Communications of the ACM. 13(6), (1970).
[13] E. Hevia, and T. J. Teorey : The Usage Dependency Model for Logical Database Design, CRL-TR-19-84, The University of Michigan. (1984).
[14] J. D. Ullman : Principle of Database Systems. Computer Science.
[15] J. G. Burch : Information Systems : Theory and Practice. John-wiley & sons.
[16] J. Martin : Computer Data-Base Organization. Prentice Hall. (1977).
[17] J. Rissanen : Independent Components of Relations. ACM Transactions on Database Systems. 1(4), (1977).
[18] P. A. Bernstein : Synthesizing Third Normal Form Relations from Functional Dependencies. ACM Transactions on Database Systems. 1(4), (1977).
[19] P. P. Chen : The Entity-Relationship Model-Toward a Unified View of Data. ACM Transactions on Database Systems. 1(1), (1976).
[20] R. Fagin : Multivalued Dependencies and a New Normal Form for Relational Databases. ACM Transactions on Database Systems. 2(3), (1977).
[21] S. A. Borkin : Data Models. The MIT Press.(1980).
[22] T. J. Teorey, and J. P. Fry : Design of Database Structure. Prentice Hall. (1982).
[23] T. W. Ling, F. W. Tompa, and T. Kameda : An Improved Third Normal Form for Relational Database. ACM Transactions on Database Systems. 6(2), (1981).
[24] W. W. Armstrong : Dependency Structure of Data Base Relationships. Information Processing 74, North Holland Pub. Co. (1974).
[25] W. W. Armstrong, and C. Delobel : Decompositions and Functional Dependencies in Relations. ACM Transactions on Database Systems. 5(4), (1980).
[26] Y. D. Lien : Hierarchical Schemata for Relational Databases. 6(1), (1981).