Design of Normalized Relation: An Ameliorated Tool

Ajeet A. Chikkamannur[1] and Shivanand M. Handigund[2]
[1] Department of Computer Science and Engineering
REVA Institute of Technology and Management
Bangalore 560064, INDIA
[2] Department of Computer Science and Engg.
Bangalore Institute of Technology
Bangalore 560004, INDIA
ac.ajeet@gmail.com, smhandigund@gmail.com,

Abstract: *The "normalization" is a practice used to design the relation(s) for a good database eliminating undesirable functional dependencies amongst that exist amongst attributes of the relation. The complexities involved in the normalization of relations, have mowed down vendors from automating the normalization processes. Although the keyword normalization is existing in the data manipulation language of Structured Query Language (SQL) standard*

This paper unravels the complexities involved in the normalization process and proposes an automatic methodology for refining the relations with normalization. The primary key for each relation is designed based on the superset of minimum attribute(s), which uniquely determines other attribute values of the tuple in the relation. Utilizing the blend of analytical and synthetic approaches, the proposed implementation process forms and refines the relations grouping (with the use of axioms) the desirable functional dependencies of the relation to satisfy the first, second and third normal form rules.

Keywords: *dependency matrix, functional dependency, attributes, pseudo-transitivity axiom, relation, normalization*

Received: July 03, 2010 | ***Revised***: October 08, 2010 | ***Accepted***: May 2011

1. Introduction

Relational theory was originally proposed by Codd [1, 2, 3] and subsequently refined through many years of research. But today in the software market, the products are designed by deviating relational theory in its entirety [7] and the design of new relational database is made either by evolution of existing product or by the experience of designers [8]. Presently, research is carrying on the application of relational theory for XML data [19, 20] in web applications.

In relational theory, "normalization" is a procedure used to implement good database design, regulating the functional dependencies amongst the attributes. The normalization process eliminates the implicit and undesirable dependencies. But the vendors are mowed t over the automation of normalization procedures in their products, though the keyword "normalization" exists in SQL standard.

This paper proposes a realizable design procedure to design relation(s) from a set of functional dependencies and attributes by incorporating the constraints laid by the first normal form, second normal form, third normal form, with identification of candidate key attribute(s).

2. Background

A relationship between the abstracted attributes from a problem statement is represented by "functional dependency". Such dependencies are used to group the attributes for a relation. A functional dependency over a set of attributes U is represented as $X \rightarrow Y$ where $X, Y \subset U$. If any relation R holds the functional dependency then the two tuples t_1 and t_2 of relation R with $t_1[X] = t_2[X]$ then $t_1[Y] = t_2[Y]$.

Normalization is a procedure to group the attributes for a relation [11] depending on the fulfillment of normal forms constraints. Originally three normal forms: first normal form, second normal form and third normal form of relations were proposed based on the functional dependencies. Further, the fourth, fifth normal forms were proposed based on the

multivalued and join dependencies. Recently the sixth normal form i.e. the Domain Key Normal Form [7] is added but it is applicable for a specific application. However, this paper restricts the discussion for those normal forms which are on functional dependencies i.e. the first three normal forms.

The first normal form [1] of a relation stresses on the atomicity of attribute values. It focuses on the structure of an attribute with unique and single value rather than a set of values and dependency of other attributes on partial values. Here, the same attribute's atomicity is application dependent. For example, in an employee relation, the attribute *date* (comprising date, month and year) of joining may be atomic with respect to that application and the same is not atomic when it is included in a zodiac application. Hence, the structure of an attribute is designed to maintain atomicity based on the application.

The second normal form [2, 3] of a relation is structured based on the dependency relationship between the part of key (sub key) and non-key attributes. This is acquired by eliminating a subset connection between determinant attributes of functional dependencies participating in the construction of second normal form relation.

The third normal form [2, 3] of a relation is relationship between the non-key attributes themselves. The elimination of dependencies between non-key attributes further trims the second normal form relation to the third normal form relation.

3. Frame work

The database schema design commences with abstraction of attributes and functional dependencies from an application by forward engineering process [4] from the software requirement specification (SRS). The abstracted attributes from a requirements specification in a forward engineering are descriptive in nature and hence, the attributes are structured with a single value, in a separate relation [8] or with relation valued attributes [7] depending on the application requirements.

The structured attributes and minimal covered attributes group[5, 6] represented in canonical form of functional dependencies are represented as a column and row of a dependency matrix with each element value depending on the following condition.

$$a_{ij} = \begin{cases} 1 & \text{determinant attribute of fd} \\ 0 & \text{dependent attribute of fd.} \\ x & \text{otherwise} \end{cases}$$

3.1First Normal Form Relation

The design of first normal form of a relation commences with identification of candidate key(s) from a set of attributes and fds. *The criterion for selection of a candidate key is based on "dominating set theory" with minimum number of attributes i.e. minimal super set of attributes is proposed* below:

P = {X | as a determinant attribute ∈ number of fds}
A = {Y | number of determinant attributes ∈ fds}

The determinant attributes of a row corresponding to the maximum P and minimum A, are selected as a candidate key.

Candidate key = {
key | determinant attributes of fd corresponding to maximum (P) and minimum(A)}

The procedure of identifying a candidate key from a dependency matrix representation of attributes and functional dependencies is as follows:

1) Count the number of 1's in each column (ccount)
2) Count the number of 1's in each row (rcount)
3) while (Value (maximum ccount, minimum rcount value) != 1)
   minimum rcount ← next minimum rcount
4) Candidate key ← determinant attributes of row selected

3.2 Second Normal Form Relation

The second normal form relation(s) is designed for preserving the transitivity dependencies between a part of the primary and non-key attribute, but the *subset link between key and non-key attributes* is eliminated. The pseudo-transitivity axiom [9] (which is inferred from the three axioms reflexivity, augmentation and transitivity) is employed for identifying the transitivity link among functional dependencies and the rule is given below.

If A → B and WA → C then WB → C

The pseudo-transitivity relation between functional dependencies is identified by comparing the element values with 0 and 1 of an attribute in a different row and same column and if it is true, then the dependent attributes (elements with 0 values) of the row with element value 1 is merged with the row having the element value 0. The row with element value 1 is discarded. This procedure is repeated until all the functional dependencies are completed. A procedure to design a second normal form relation is given below.

1) Select a row which has value (max rcount, min account)== 1
2) Merge the attributes of rows having pseudo-transitivity link and delete the linked row
   Repeat this step until there is no link.
3) If there are rows without pseudo-transitivity Links then select the row with value (max rcount, next min ccount) == 1 and
   go to step 2.
4) Merge the rows with exclusively identical determinant attributes.
5) For all rows
   If (determinant attribute(s) of row i is subset of row i+1)

Delete the dependent attribute of row i
in accord with dependent attributes of row i+1

6) Construct relation corresponding to each row and revamp the determinant attribute(s) to key attribute(s).

3.3 Third Normal Form Relation

The third normal form relation(s) is constituted by identifying the pseudo-transitive link path among the different rows and the path link is tested for termination. If a path terminates at the dependent attribute(s) of a starting row then the terminating attribute(s) is deleted. The procedure to constitute a third normal form relation(s) from a set of functional dependencies is given below:

1) Merge the rows with exclusively identical determinant attributes
2) Select a row which has a value (max rcount, min ccount) == 1
3) If (path ends at dependent attribute(s) of selected row)
4) Delete the attribute(s) of selected row at which path ends.
5) Repeat the step 3 by selecting non deleted rows one by one
6) Construct relation corresponding to each row and revamp the determinant attribute(s) to key attribute(s)

4. Case study

In this section, three case studies from simplest to complex are considered for the demonstration. The execution of tool is shown by the graphical approach and the implementation is done with amelioration in the depth first search (DFS) source code.

Case 1: Consider the functional dependencies **A → BC, E → AD, G→ AEJK, GH→ FI. K→ AL, J →K** [22]. The dependency matrix representation is depicted in figure 1.

	A	B	C	D	E	F	G	H	I	J	K	L	
1	1	0	x	x	x	x	x	x	x	x	x	x	1
2	1	x	0	x	x	x	x	x	x	x	x	x	1
3	0	x	x	x	1	x	x	x	x	x	x	x	1
4	x	x	x	0	1	x	x	x	x	x	x	x	1
5	0	x	x	x	x	x	1	x	x	x	x	x	1
6	x	x	x	x	0	x	1	x	x	x	x	x	1
7	x	x	x	x	x	x	1	x	x	0	x	x	1
8	x	x	x	x	x	x	1	x	x	x	0	x	1
9	x	x	x	x	x	0	1	1	x	x	x	x	2
10	x	x	x	x	x	x	1	1	0	x	x	x	2
11	0	x	x	x	x	x	x	x	x	x	1	x	1
12	x	x	x	x	x	x	x	x	x	x	1	0	1
13	x	x	x	x	x	x	x	x	x	1	0	x	1
	2	0	0	0	2	0	6	2	0	1	2	0	

Figure 1. Dependency matrix

The bottom row depicts the number of participation of each attribute in a set of functional dependencies and the last column shows the number of determinant attributes in each functional dependency. The attribute G has the maximum number of participation and minimum number of association with other determinant attributes. Hence, the attribute G is determined as a candidate key and is revamped as a primary key in a relation. The resulting first normal form relation is shown in figure 2.

G	A	B	C	D	E	F	H	I	J	K	L

Figure 2. First Normal Form Relation

Case 2: Consider the functional dependencies **AB→CEFGH, A→D, F→G, BF→H, BCH→ADEFG** and **BCF→ADE** [22].

The dependency matrix representation is shown in figure 3.

	A	B	C	D	E	F	G	H	
1	1	1	0	x	x	x	x	x	**2**
2	1	1	x	x	0	x	x	x	2
3	1	1	x	x	x	0	x	x	2
4	1	1	x	x	x	x	0	x	2
5	1	1	x	x	x	x	x	0	2
6	1	x	x	0	x	x	x	x	1
7	x	x	x	x	x	1	0	x	1
8	x	1	x	x	x	1	x	0	2
9	0	1	1	x	x	x	x	1	3
10	x	1	1	0	x	x	x	1	3
11	x	1	1	x	0	x	x	1	3
12	x	1	1	x	x	0	x	1	3
13	x	1	1	x	x	x	0	1	3
14	0	1	1	x	x	1	x	x	3
15	x	1	1	0	x	1	x	x	3
16	x	1	1	x	0	1	x	x	3
	6	**14**	8	0	0	5	0	5	

Figure 3. Dependency matrix representation

Rows 6 and 7 are ignored since the value for maximum column count (14) and minimum row count (1) is x. Hence, the row 1 is selected as an initial functional dependency for the design. The pseudo-transitive linked rows are depicted by darkening the elements and the dependent attributes are merged to form a single row. The resulted rows from figure 3 are shown in figure 4.

	A	B	C	D	E	F	G	H
1	1	1	0	0	0	0	0	x
2	1	1	x	x	0	x	x	x
3	1	1	x	x	x	0	0	0
4	1	1	x	x	x	x	0	x
5	1	1	x	x	x	x	x	0
6	1	x	x	0	x	x	x	x

Figure 4. Result rows

The attributes of rows 1, 2, 3, 4 and 5 of result rows are merged to form a relation because of exclusive identical determinant attributes and the relation corresponding to attributes of row 6 is constituted. The second normal form relations are shown in figure 5.

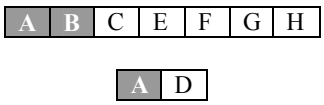

Figure 5. Second Normal Form Relations

The attribute D is eliminated from relation 1 because of $A \in D$ and the determinant attributes are revamped to the primary key of a relation, which is depicted by darkening the attributes.

Case 3: Consider the functional dependencies **AB→CEFGH, A→D, F→G, BF→H, BCH→ADEFG** and **BCF→ADE** [21]**.** Minimal cover [5, 6] eliminates the **BCF→ADE**. The dependency matrix representation is shown in the figure 6.

	A	B	C	D	E	F	G	H
1	1	1	0	x	x	x	x	x
2	1	1	x	x	0	x	x	x
3	1	1	x	x	x	0	x	x
4	1	1	x	x	x	x	0	x
5	1	1	x	x	x	x	x	0
6	1	x	x	0	x	x	x	x
7	x	x	x	x	x	1	0	x
8	x	1	x	x	x	1	x	0
9	0	1	1	x	x	x	x	1
10	x	1	1	0	x	x	x	1
11	x	1	1	x	0	x	x	1
12	x	1	1	x	x	0	x	1
13	x	1	1	x	x	x	0	1

Figure 6. Dependency matrix representation

In the first step, the rows with exclusively identical determinant attributes are merged and the resulting matrix is shown in figure 7.

	A	B	C	D	E	F	G	H	
1	1	1	0	x	0	0	**0d**	**0d**	2
2	1	x	x	0	x	x	x	x	1
3	x	x	x	x	x	1	0	x	1
4	x	1	x	x	x	1	x	0	2
5	0	1	1	**0d**	**0d**	**0d**	**0d**	1	3
	2	3	1	0	0	2	0	1	

Figure 7 Result Matrix

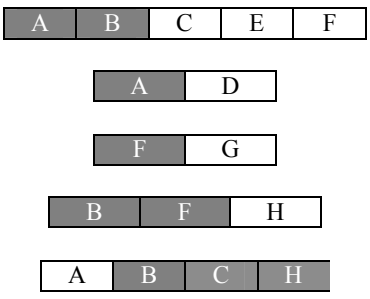

Figure 8. Third Normal Form Relations

Since there is a pseudo-transitive link with rows 3, 4 and path terminates at G, H attributes of row 1 in result for third normal form shown in the figure 9, Hence the attributes G, H are deleted from row 1. Similarly the attributes D, E, F and H are deleted because of pseudo-transitive link of rows 1, 2, 3 and 4. Then the relation corresponding to each row attributes is created. The third normal form relations with their keys are shown in figure 8.

5. Related work

Database management books [7, 8, 9, 10, 11, 14] provide normalization as a topic with theory of decomposing the universal relation to a normalized relation with examples. In a paper George C. Philip [12] given the practical role of normalization in database design, addressing the confusion in representing attributes with repeating values, discussed the removal of inconsistencies in defining relations with first normal form and simplified process of identifying the candidate keys. W. Kent [13] has given the guidelines for a record design. These guidelines are corresponding from first to the fifth normal forms, without referring to the concept of a relational model for generality, are easier to understand.

Fangjie Xu et al [16] developed a Computer Application Interface tool to demonstrate normalization or relation in step by step format for easier understanding of the key points of the theory. Nijsse et al [17] have developed a step by step method to develop a logical relational data model with non-mathematical terms. But the method uses an adapted version of Normalization by Synthesis.

Tauqeer Hussain et al [18] have proposed the elimination of the normalization process from database design and the relations are created by Entity-Relationship Diagram with a set of rules derived from the functional dependencies. Amir Hassan Bahmani et al [21] have developed an automatic tool with the aid of graph theory and identification of a key as a side outcome.

Recently the "normalization theory" is utilized in the XML data relationship [19, 20] for web applications. This paper identifies the candidate key attributes with a simple approach and the design of first normal form, second normal form and third normal form relation(s).

6. Conclusion

This paper automated a methodology that blends the analytical [1, 2, 3] and synthetic approaches [15] for the design of normalized relations by the dependency matrix representation corresponding to the attributes and functional dependencies.

The criterion for identification of a candidate key based on the *"dominating set theory" with minimum number of determinant attributes* is proposed. This is realized by the procedure that counts the participation of an attribute as a determinant attribute in a number of functional dependencies and the minimum number of associations with other determinant attributes.

Further, the second normal form, third normal form relations are designed by preserving, eliminating pseudo-transitivity link between various functional dependencies respectively. Then, candidate key(s) is revamped as the super key for a relation(s) at the end of each normal form design procedure.

The tool is refined with amelioration of DFS algorithm in its source code. The tool is simple, efficient for identification of key attribute(s), to design relations satisfying the first, second, third, normal form relation(s). This is demonstrated with case study. Further the correctness and completeness of a designed relation(s) is to be realized.

7. Acknowledgement

We acknowledge the grant provided by All Indian Council of Technical Education (AICTE) under Research Promotion Scheme through it's F. NO.: 8023 / BOR / RID / RPS – 99 / 2007-08

References

[1] E.F. Codd, "A Relational Model of Data for Large Shared Data Banks", *Comm. ACM 12 (6)*, June 1970, page 377-387.

[2] E.F. Codd, "Normalized Data Base Structure: A Brief Tutorial", ACM SIGFIDET *Workshop on Data Description, Access, and Control*, San Diego, California, 1971

[3] E.F. Codd, "Further Normalization of the Data Base Relational Model", *IBM Research Report* RJ909.

[4] S M. Handigund, "Reverse Engineering of Legacy COBOL systems", Ph. D. thesis Indian Institute of Technology Bombay, 2001

[5] A. A. Chikkamanur, S. M. Handigund "Categorization of Functional Dependencies for a Minimal Cover", *ICSTC, San Diego USA*. page 213-217, 2008.

[6] A A. Chikkamanur, S. M. Handigund, "An efficient Methodology for Determining a Minimal Cover of Functional Dependencies", *ICISTM 2008*, Dubai. 2008. unpublished

[7] C. J. Date, A. Kannan, S. Swaminathan, "An Introduction to Database Systems", 8th Edition, Pearson Education (Dorling Kindersley (India) Pvt. Ltd.), 2008.

[8] Elmasri, Navathe, "Fundamentals of Database Systems", 5th Edition, Pearson Education, 2008

[9] Silberschatz, Korth, Sudarshan, "Database System Concepts", 5th Edition, McGraw-Hill International Edition, 2004

[10] Jeffrey D Ullman, "Principles of Database Systems", Second Edition, Galgotia Publications (P) Ltd, New Delhi, 1984.

[11] Rob, Cornel, "Database Systems Design, Implementation and Management", 5th edition, Thomson Asia Pvt. Ltd., Singapore, 2003

[12] George C. Philip "Teaching Database Modeling and Design: Areas of Confusion and Helpful Hints" *Journal of Information Technology Education* Volume 6, page 481-497, 2007

[13] W. Kent, "A Simple Guide to Five Normal Forms in Relational Database Theory", *Communications of the ACM, Vol. 26, No. 2,* page 110-114, 1983.

[14] Patric O'Neil, Elizabeth O'Neil, "Database: principles, programming and performance", 2nd Edition, Morgan Kaufmann, 2001.

[15] Philip A. Bernstein, " Synthesizing Third Normal form Relations from Functional Dependencies", *ACM transactions on database systems, Vol. 1, No. 4,* page 277-298, 1976.

[16] Fangjie Xu, Huichuan Duan, "A CAI Tool for the Theory of Relation Normalization", *1-4244-1285-0/07 IEEE* page 532-534, 2007

[17] J. P. Nijsse, R. J. Whiddett, C. F. Atkins "Logical Relational Datamodelling through Normalization by Synthesis" *Proceedings of the 15th Annual NACCQ, Palmerston North, New Zealand*, page 133-138, 2003

[18] Tauqeer Hussain, Shafay Shamail, Mian M. Awais, "Eliminating process of Normalization in Relational Database design" *Proceedings IEEE INMIC,* page 408-412, 2003.

[19] M Arenas, L Libkin, "An Information-Theoretic Approach to Normal Forms for Relational and XML Data" *Journal of the ACM (JACM), Vol. 52(2)*, page 246-283, 2005.

[20] Kolahi, S., "Dependency-Preserving Normalization of Relational and XML Data"

Journal of Computer System Science, Vol. 73(4): page 636-647, 2007.

[21] Amir Hassan Bahmani, Mahmoud Naghibzadeh, Behnam Bahmani "Automatic database normalization and primary key generation" *IEEE CCECE/CCGEI May 5-7*, Niagara Falls, Canada, 2008.

[22] Thomos Connolly, Carolyn Begg, "Database Systems: A practical approach to design, implementation and management", Third Edition, Pearson Education, 2005.

Ajeet A. Chikkamannur received his M. Tech. degree

in Computer Science and Engineering in 2001 from the Visvesvaraya Technological University, India. Currently pursuing the Ph.D. and the research is focused on Design of Fourth Generation Languages. His research interests are Object Oriented System Development, Database Management Systems, System Simulation and Modeling. Presently working as Professor, Department of Computer Science and Engineering and teaching for graduate courses for last twenty one years

Prof. Shivanand M. Handigund received his Ph.D.

degree from Department of Computer Science & Engineering, Indian Institute of Technology, Bombay in 2001. Currently, he is working as a full time Professor and Head, Super Computer, M. Tech. CSE Programme and Research Centre at Department of Computer Science and Engineering, Bangalore Institute of Technology, Bangalore. His research interests are Software Engineering, Reverse Engineering, Database Management Systems, Object Technology and Computer Graphics. He teaches several courses to Academia and Industry engineers. He has organized number of conferences and delivered keynote addresses & invited talks at several conferences. He is a Ph. D. referee and IEEE technical papers reviewer.