International Journal of Engineering Research-Online
A Peer Reviewed International Journal
Articles available online http://www.ijoer.in

Vol.3., Issue.5., 2015
(Sept.-Oct.)

RESEARCH ARTICLE

ISSN: 2321-7758

AN ALGORITHMIC APPROACH TO NORMALIZATION

PREETI YADAV[1], AJAY KUMAR[2]

[1] Asst. Prof. in Govt College Gurgoan(Haryana)

[2]Asst. Prof. in IGU Meerpur Rewari(Haryana)

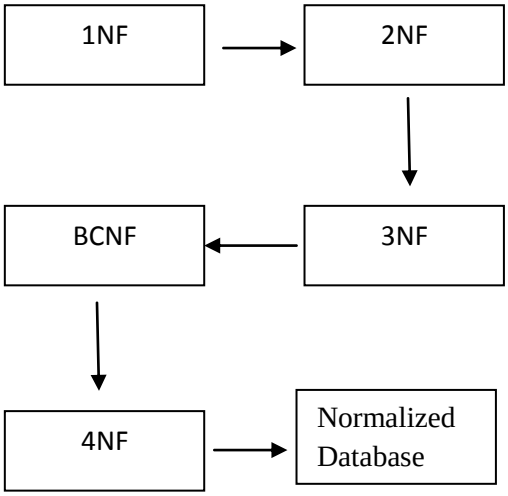
ABSTRACT

Now a days, normalization is an emerging technology that allows users to access information and services. It is the process of analyzing the given relational schemas based on the functional dependencies and using primary key to achieve minimum data redundancy. Normalizing a logical database design involves organizing the data onto more than one table. This research work aims to resolve this issue by normalization of multiple database automatically. This approach provides automatic normalization of databases up to 4NF.In this paper; we propose an algorithm for database normalization. The effectiveness of the proposed approach is saving time and reducing mind work.

Keywords— Functional dependency, keys, first normal form, Second Normal form, Third normal Form, BCNF, Forth Normal Form

I.INTRODUCTION

For developing any software system, the database normalization helps to avoid data redundancy. If the relational database is used it consumes time. If we missed out any constraint we will face problems while when we automate data base. It is easy for normalize database. The process of normalization was first developed by E.F .Codd. Normalizing a logical database design involves organizing the data into more than one table. Logical structure of the data base is designed so that basic operation on the database can be performed without any problems. It provides the performance by reducing Redundancy in database table. The basic objective of normalization are to reduce redundancy which means that information to be stored only once in a relation. Database Normalization is the process of organizing the fields and tables of a relational database to minimize redundancy and dependency. Normalization usually divides large tables into smaller tables and defining relationship between them. The objective is to isolate data so that addition, deletion and modifications of a field through the rest of the database via the defining relationship.



II. BASIC DEFINITIONS

- A super key is a set of attributes which will uniquely identify each tuple in a relation. It need not to be minimal. All candidate keys are super keys. Not all super keys are candidate keys.

- A candidate key is minimal key. It's no proper subset is a key. There can be multiple candidate keys in a table.
- Primary key is selected from candidate key by DBA. It represents table in database. Only one primary key as per table.
- An attribute A is a key attribute in a relation R if A is a part of some candidate key otherwise it is a non- key attribute in R i.e. A is not a component of any candidate key.
- Given a Relation R, a set of attribute X In R is said to be Functionally determine another set of attribute Y, also in R, written X ; if and only if each X value is associated with precisely one Y Value; R is then said to satisfy the functional dependency XY.
- A functional dependency X Y is called trivial if Y is a Subset of X.
- A set of functional dependency F in canonical form if each functional dependency X, A in F, A is a singleton Attribute.
- Extenous attribute are those attribute which even after being removed from the functional dependency do not affect its right side.
- If is an FD AB-> C and there is another FD A-> B then B can be removed From FD AB-> C to reduce it to A-> c.
- A fully functional dependency AB-->CD is said to be *Fully functional* over AB if for no pair of B and D there is a dependency of B-->D.
- Functional Dependency which do not satisfy the condition of a fully functional dependency are *partial dependency.*
- If there are two functional dependency present in the same set A-->B and B-->C then the third functional dependency A-->C is known as *transitive functional dependency.*

III. NORMAL FORMS

1 First Normal form

- It is the lowest level of normalization.
- It imposes very basic requirement over a table.
- A relation Database Table is said to be 1 NF if all the attribute in it have atomic values means have simple and single-valued both values.
- No non atomic attribute are allowed.

To check whether a table is in first normal form we need to see the table

- It is not defined by functional dependencies.
- If the table has not been provided then it is assumed that it is already in first normal form.

2. Second Normal form

- The requirement for second NF is that there should be no partial dependency on candidate key and it should be in 1NF.
- In other words. We can say that there is any functional dependency then it should be only full functional dependency over candidate key.
- Meaning of full functional dependency and partial dependency are same as explained in basic definitions.
- No part of key should be able to derive any attribute an its own.

Algorithm to check if Functional dependency (F.D.) set is in 2 NF

1. Find all candidate key and prime attribute.
2. Mark F.D. Where L.H.S. is non prime or key.
3. If combination of prime and non prime attribute are present in L.H.S.
4. Remove extraneous attribute and mark if extraneous not present.
   repeat step 2.
5. Mark the F.D. Whose R.H.S. is a prime.
6. If R.H.S. has combination of prime and non-prime , decompose and repeat step 4.
7. If any F.D. Is left unmarked then the relation is not in 2N.F.

The unmarked F.D's. is violate the 2N.F. condition.

Eg1:

F= {A-->B, A-->C, CG--> H, CG-->I, B-->H}

Step:

1) candidate key is AG.
   prime attribute=A, G.
   non Prime =B, C, H, I.
2) No F.D. is there which has AG key.

Mark F.D. B-->H because it has non -prime (B) present in L.H.S.

Now A-->B
A-->C
CG-->H
CG-->I
B-->H

3) There is a combination of prime and non prime on left,

CG-->H
CG-->I

Mark if non prime is not present in closure of the prime i.e. the non prime is not derived by that prime.

C is not derived by G, hence after marking

A-->B
A-->C
CG-->H
CG-->I
B-->H

4) No F.D. is there which has A or G in Right.

5) No combination of prime and non prime is present in R.H.S.

6) There is two unmarked F.D.

A-->B
A-->C

Which violate the 2NF condition.

Eg2:

F= {AB-->CD, B-->E, EC--> B}

Step:

1) Candidate key is AB and AEC.
prime attribute=A,B,C,E.
non Prime =D.

2) Mark F.D. where key AB or AEC is present in L.H.S.

AB-->CD
B-->E
EC-->B

3) No combination of prime and non prime attribute is present in L.H.S.

4) Mark the F.D. where prime attribute (A, B, C, E) are present in R.H.S.

AB-->CD
B-->E
EC-->B

5) No F.D. Is left unmarked then the relation is in 2N.F.

3 Third Normal form

- It requires that there should be no transitive functional dependency over candidate key present and it should also in 2 NF.
- Meaning of transitive functional dependency has been discussed in basic definitions.
- There should be no functional dependency from non-prime to non-prime attributes.

Algorithm to check if Functional dependency (F.D.) set is in 3 NF

1) Find all candidate key and prime attribute.
2) Mark F.D. Where L.H.S. is key.
3) Mark the F.D. Whose R.H.S. is a prime .
4) If R.H.S. has combination of prime and non prime , decompose and repeat step 3.
5) If any F.D. is left unmarked then the relation is not in 3N.F.

The unmarked F.D.'s Is violate the 3N.F. condition.

Eg1: F= {AF-->B, AF-->C, CF-->DE, C-->G}

Step:

1) candidate key is AF.
prime attributes = A,F.
non prime=B, C, D, E.

2) Mark F.D. where AF is present in L.H.S.

AF-->B
AF-->C
CF-->DE
C-->G

3) No F.D. is there were A or F has present

4) No F.D. is present where combination of prime and non prime in R.H.S.

5) There is two unmarked F.D.
CF-->DE
C-->G

which violate the 3NF condition.

Eg2: F= {AB-->CD, B-->E, E--> B}

Step:

1) candidate key is AB, AE.
prime attributes = A,B,E.
non prime=C, D.

2) Mark F.D. where AB is present in L.H.S.

AB-->CD
B-->E
E--> B

3) Mark the F.D. A,B,E are present in R.H.S.

AB-->CD
B-->E
E--> B

4) No F.D. is left unmarked then the relation is in 3N.F.

4 Boycee Codd Normal Form

Boycee requires that the table should be in 3NF and if there is any FD it should be over super key.

Algorithm to check if Functional dependency (F.D.) set is in BC NF

1) Find all candidate key and prime attribute.
2) Mark F.D. Where L.H.S. is key.
3) If any F.D. is left unmarked then the relation is not in BCNF.

The unmarked F.D.'s is violating the BCNF. condition.

Eg1: F= {AF-->B, AF-->D, C-->A, A-->C}

Step:

1) candidate key is AF and CF.
   prime attributes = A,F,C.
   non prime=B, D.
2) Mark two F.D. where AF is present in L.H.S.
   AF-->B
   AF-->D
   C-->A
   A-->C
3) There is two unmarked F.D.
   C-->A
   A-->C
   which violate the BCNF condition.

Eg2: F= {AB-->CD, CD-->E, E--> AB}

Step:

1) Candidate key is AB , E and CD.
   prime attributes = A,B,C,D,E.
   non prime=Null
2) Mark two F.D. where AB,CD,E are present in L.H.S.
   AB-->CD
   CD-->E
   E--> AB
3) No F.D. is left unmarked then the relation is in BC.N.F.

IV.SUMMARY

In this paper, we have presented algorithms for relational database normalization into 2NF, 3NF and BCNF using their general definitions in a step by step algorithm. The first step before performing the procedure is to understand the basic definitions which we are presented in this paper. We have tested our algorithms on many realistic examples with multiple candidate keys taken from different sources.

This work has mainly the following advantages:

- The removal of redundant dependencies
- The general and original definition of normal form is used.
- In all phases, the computation of attributes closure is minimized compared to other algorithms although using a restricted definition of normal forms.
- A primary key is determined for any general relation.

Thus we have tried to present the various normal forms and functional dependency in simple and undersatndabnle way.

V. REFRENCES

[1]. Thomas, C., Carolyn, B.: Database Systems, A Practical Approach to Design, Implementation, and Management, Pearson Fourth edition (2005).

[2]. Bahmani A., Naghibzadeh, M. and Bahmani, B.: Automatic database normalization and primary key generation, Niagara Falls Canada IEEE (2008).

[3]. Beynon-Davies, P.: Database systems Palgrave Macmillan, Third edition, ISBN 1–4039--1601–2 (2004).

[4]. . Dongare,Y. V., Dhabe,P. S. and Deshmukh, S. V.: RDBNorma: A semi-automated tool for relational database schema normalization up to third normal form, Interrnational Journal of Database Management Systems, Vol.3, No.1 (2011).

[5]. Vangipuram, R., Velputa, R., Sravya, V.: A Web Based Relational database design Tool to Perform Normalization, International Journal of Wisdom Based Computing, Vol.1(3) (2011).

[6]. Shamkant B. Navate, "Fundamentals of Database management system", Pearson Publication.

[7]. IMark Levene and Millist W. Vincent, "Justification for Inclusion Dependency Normal Form", IEEE transaction on knowledge and data engineering, VOL. 12, NO. 2, MARCH/APRIL 2000.