2017 4th IEEE Uttar Pradesh Section International Conference on Electrical, Computer and Electronics (UPCON)
GLA University, Mathura, Oct 26-28, 2017

Database Normalization Design Pattern

Kunal Kumar
Research Scholar
University Department of Statistics and Computer Applications,
T. M. Bhagalpur University, Bhagalpur-812007, India
kunal.mishra.mca@hotmail.com

Dr. S. K. Azad
Associate Professor and Head
University Department of Statistics and Computer Applications,
T. M. Bhagalpur University, Bhagalpur-812007, India
skazad@rediffmail.com

Abstract—**Normalization is a database design technique, which is used to design the Relational Database table up to higher normal form. The main aim of the database normalization approach is to reduce data redundancy and maintain the atomicity within the database table. In this paper, The Tabular approach algorithm method is introduced to generating candidate key from set a valid set of functional dependency. Once, it determines the candidate key of a database table from a given valid set of functional dependency, and then applying normalization algorithms we can easily achieve higher level of normal form. It also shows new tabular based automatic generation of all possible super key and candidate key of a database table. A tabular approach algorithm method have table, having three columns marked as left, middle and right and one row. This column contains the valid set of given functional dependency. Once, we feed the column of the tabular approach table marked as left, middle and right with all possible valid set of functional dependency, then using proposed tabular approach algorithm method used to determine candidate key(s) and using these key(s), we can reach higher level of normal form.**

Keywords— Functional Dependency, Relational Database, key(s), Normalization, Tabular Approach.

I. Introduction

The term normalization was introduced by E.F.Codd. The major goal of relational database normalization is to maintain atomicity, remove data redundancy, remove data inconsistency from the database table and ensure that data dependence among attributes make sense [1]. Normalization is a step by step method of working that coherently organizing data in Relational database system. Normalization applies well ordered approach of break down the database tables to bring down data redundancy, inconsistency and un-desirable properties in the same manner as Insertion, Up-gradation and Deletion Anomalies. Edgar Codd, who proposed the theory of the relational database model, also developed the theory of relational database normalization design [2]. The relational database hypothesis for normalization normal forms of which provide certain rules for discover the relations degree level of permeability to conceptual inconsistencies in the database relations. Once, a relations of database reach at higher normal form then it is possible less permeability it is. In general, relational database normalization needs additional relations and some database designers find this initial difficult and then awkward [3]. It generally requires decompositions of a relational database into two or more database relations and also defining their relationships between the database relations [4][5]. The motto is to scattered the data of the relations so that Indentation, Deletion, and up gradation of a column can be done in just one relation and then spread through the rest of the relations in the database via the relationships established between them. It is a break down process that stored data into tabular form by eliminating repeated data from the relational table(s) and also preserves the dependency which is given with the relation [6]. While decomposing a table we must ensure that that new table must hold the given dependency. In the absence of Normalization, it comes to be very back-breaking to handle and insert, update and delete the database, without facing data loss [7]. DML, DDL and TCL are the database languages, that very useful.

For designing a large and good database system is a practical approach. "Relational database normalization" is the best methodology for this purpose, Relational database normalization design shows dependency among the table attributes. They the concept of finding key of the table is also make the database more robust and secure [8]. The key value of the table is always having unique value in the table due to this feature of the key all other database relations attributes are dependents on the key value. Normalization is generally a standardization technique of table designing. After, generating the key of the table using closure property of some other algorithm technique [9], designer can reach the higher normal form of the normalization; it's a step by step procedure. Functional dependency of the relation shows various relationship of attribute among attribute or some other relation [10]. To perform various operations on the database to be normalized, Let us consider the example of the table

Information

Id	Name	Dist	Pin	Email
112233	A	Pune	800000	hotmail
112234	B	Delhi	800001	gmail
112235	C	Puri	800002	outlook
112236	D	Patna	800003	live
112237	E	Pune	800004	bsnl

Figure 1: Information Table

Updating: Update is a DML Command, which always deals with data manipulation in a table not with the structure of the table. It is used to change the one or more tuple based on some condition. In, the Figure 1, if it is come to the notice that, the email address of **ID** 112235 is wrongly mentioned. The following command is used to update the email address to *live.in*. update Information set email="live.in" where ID =112235; [11]. Insertion: After creation, a table Information. It is empty i.e. there is no data in a table. So, insertion of new row in a table is done by Insert command. Insert into information values (112233,'A','PUNE', 800000,'HOTMAIL'). Deletion: If ID=112235 has only one PIN attribute but it also delete the entire dependent attribute of the database relation. To achieve higher level of normalization, we must passes through these four levels [12].

- First normal form (1NF)
- Second normal form (2NF)
- Third normal form (3NF)
- Boyce-codd normal form (BCNF)

According to First Normal Form, no tuple of relation contain group of information i.e. each set of tuple(s) in a relation must have an atomic entry; multi-valued attribute should be eliminated from the relation. To ensure that each tuple should be atomic value, then it is compulsory should form key attribute in a relation. A key(s) in a table always have unique values, there is no chance for data inconsistency. Before moving towards 2NF, it must be assumed that table is in 1NF.The Second Normal Form is based on the concept of Full Functional Dependency and there must not be any partial dependency of any attribute(s) of the key. If any attribute(s) partially depends only on one part of the concatenated key, then the table fails 2NF. It is assumed that table is in 2NF, [13]. So this *transitive functional dependency* should be removed from the table. It is assumed that table is in 3NF, Boyce and Codd Normal Form is considered as strict, then 3NF. we will use a tabular approach method to determine the key(s) of a relation. After, determining the key(s) from a given relation we moved towards higher normal form.

II. RELATED WORK

Edgar Frank "Ted" Codd in 1970 is the first who introduced the formation of relation database normalization. Codd in 1970 in introduced his classic paper on relational database normalization model. Family Functional dependencies and concepts of deriving key were also proposed by Codd in 1970. Relational database normalization is executed as a ladder of rules and regulations on a database relation to find out whether it is fulfil or violate the basic requirements of a given normal form. Three normal forms which are initially proposed are named 1NF, 2NF and 3NF was also given in Codd in 1972 [14].

Further in 1974 Codd extended the concept of third normal form to Boyce–Codd normal forms i.e. 3.5 NF. Ronald Fagin in 1977 comes with a new concept that introduced the "multi-valued dependencies" which are a most generalization form of the well-known family functional dependencies for relational database design and normalization. It also extends the understanding concepts of the conceptual design of the relational database normalization. This type's idea of multi-valued dependencies of family functional dependencies for relational database design further extends to the fourth normal form and it is strictly and stronger version of Boyce-Codd normal form or third normal form of Codd's rule. Boyce-Codd normal form was proposed by the Codd in 1974. Many algorithms and the theorems are proposed for concerning of relational database design algorithms, family functional dependencies and other dependencies in relational database and some other dependencies are explained by Ullman in 1988 , Maier in 1983 and Atzeni and De Antonellies in 1993.

In year 2016, Salvador Lucasa, Jose Meseguera also proposed a one more theoretical, and another more practical for Normal forms and normal theories in conditional rewriting a new born concepts and output on conditional term rewriting within the common framework of order-sorted rewrite theories, which is mainly support types, rewriting modulo axioms and subtypes, it also contains the more highly restricted framework of the conditional term rewriting systems as in a special case.

III. TABULAR APPROCH ALGORITHMS

In Tabular method Table can be design using following rules.

a. The table is divided into three columns and one row. First column of the table is marked as *left*, second column of the table is marked as *middle* and third column of the table is marked as *right*.

b. The attribute of the database table, on left hand side of the functional dependency arrow, are put in to the left marked column of the tabular approach table.

c. The attribute of the database table, on both side of the Functional dependency arrow left and right hand Side, are put in to the middle marked column of the Tabular approach table.

d. The attribute of the database table, on right hand side of the functional dependency arrow, are put in to the right marked column of the table the tabular Approach table.

e. The attribute of the database table which resides right marked column of the tabular approach table, is Considered as non-prime attribute i.e. not a part of Key.

f. If there is no information about the any attribute in set of given FDs, put the attribute in left marked column of the tabular approach table and merge with the left marked column attribute the tabular approach table, while computing closure of left marked column attribute.

g. Start finding closure from left marked column of the tabular approach table, if it is unable to find candidate key then, start Cartesian of left and middle marked column of the tabular approach table.

h. Here we are using GUI based Normalization Educational tool for proof the methods and finding the closure.

Example I. A database relation $R(A,B,C)$, functional dependency = $\{B \rightarrow C$ and $A \rightarrow B\}$, Find all possible candidate Key of a database Relation. Figure 2 shows the Tabular approach algorithms representation of the dependencies

Left	Middle	Right
A	B	C

Figure 2: Tabular format for the Functional Dependencies

From, Tabular approach algorithm it is clear that how to make arrangements of functional dependency in tabular format. Form, the algorithm it is clear that attribute in right marked column is not considered as key attribute of the relational database table. Now, start finding closure from attribute in left marked column

So, the attribute on left marked column is A

Closure of $\{A\}^0 = \{A\}$

Closure of $\{A\}^1 = \{A,B\}$

Closure of $\{A\}^2 = \{A,B,C\}$

Closure of $\{A\}^3 = \{A,B,C\}$

Closure of $\{A\}^2 = \{A\}^3$

Hence, Closure of $\{A\}^+ = \{A,B,C\}$

Attribute A is the candidate key of the database relation R . If left marked column of the tabular approach table is able to find the candidate key of any database relation. Then, it not compulsory to make the Cartesian of left and middle marked column of the tabular approach to finding key because this may generated super key of the relation. One of the best thing about the tabular approach algorithm is it identify non-prime during table design phase i.e. attributes holds position at right column of the tabular approach is not considered as key. The A is prime attribute. B and C, it is considered to as non-prime attribute(s)

Example II. A database relation $R(A,B,C,D)$, functional dependency = $\{A \rightarrow C$ and $BC \rightarrow D\}$, Find candidate Key of a database Relation.

Left	Middle	Right
A,B	C	D

Figure 3: Tabular format for the Functional Dependencies

In Figure 3, it shows that D is non-prime attribute of the relation i.e. It never be considered as part of the candidate key.

Closure of $\{A\}^0 = \{A\}$

Closure of $\{A\}^1 = \{A,C\}$

Closure of $\{A\}^2 = \{A,C\}$

Closure of $\{A\}^1 = \{A\}^2$ so, Attribute A is never considered as key. Now, again start finding the Cartesian closure of left and middle marked of the tabular approach table, because A is unable to generate the key attribute of a database relation.

Closure of $\{AB\}^0 = \{B,A\}$

Closure of $\{AB\}^1 = \{B,C,A\}$

Closure of $\{AB\}^2 = \{B,C,A,D\}$

Closure of $\{AB\}^3 = \{B,C,A,D\}$

Closure of $\{AB\}^2 = \{AB\}^3$

AB Candidate key of the relation of database relation R.

IV. DETERMINING HIGHEST NORMAL FORM

It is assumed, that the reader is quite familiar with the concept of normalization process. On other hand, relation of the database is assumed to be in first formal form. Our work to achieved higher normal form up to BCNF based on key(s) and functional dependencies. Here, Normalization Educational tool is used for simulation work to automate the relational database normalization of tables in Boyce–Codd Normal Form.

A. SECOND NORMAL FORM (OR 2NF)

It is mandatory that before go towards the 2NF the previous table must be in the 1NF. $R(A,B,C,D)$, functional dependency= $\{B \rightarrow D$ and $AB \rightarrow C\}$, Figure 4 is the Tabular approach representation of the dependencies

Left	Middle	Right
A,B		C,D

Figure 4: Tabular format for the Functional Dependencies

After applying tabular approach algorithms, it is found that the candidate key of the relation is *AB* and the above relation is not considered as 2NF .In 2NF each attribute of functional dependencies must be fully dependent on candidate key, but not dependent on part of the candidate key. The functional dependency $B \rightarrow D$ violet the criteria 2NF rules. To make this database relation in to 2NF, It has to be decomposed into two tables $R_1(A,B,C)$ and $R_2(B,D)$. After decomposition, database relation R_1 and R_2 are in Second normal form and the decomposition is lossless decomposition where as functional dependency preserved.

B. THIRD NORMAL FORM (or 3NF)

The above database relation $R_1(A,B,C)$ and $R_2(B,D)$ are in 2NF to moving towards 3NF it is compulsory that previous relation should be in 2NF. In order to transform this relation to

3NF. We have to find is there any dependency, where any non-prime attribute determine non-prime attribute which violet 3NF.

Now, $R_1(A,B,C)$ where, $AB \rightarrow C$ and AB is the candidate key of the database relation. So, $R_1(A,B,C)$ is reached to 3NF because no non-prime attribute functionally determine any other non-prime attribute. Now, the second relation $R_2(B,D)$ where, $B \rightarrow D$ and B is the key of the relation i.e. B is prime attribute. So, we can say that $R_2(B,D)$ is in 3NF.

C. BOYCE–CODD NORMAL FORM (OR BCNF)

Raymond Boyce and E.F. Codd the father of relational database design who proposed the theory of BCNF. The extension of third normal form which is stricter that itself known as BCNF form of normalization. A table is said to be in Boyce–Codd normal form iff, for any one of its dependencies $X \rightarrow Y$, at least one of the following conditions must be holds

- $X \rightarrow Y$ is a trivial functional dependency $Y \subseteq X$
- X is a candidate key for the relation R.

A database relation $R(A,B,C,D)$, functional dependency = $\{C \rightarrow D$ and $A \rightarrow B\}$, Figure 5 is the Tabular representation of the functional dependencies

Figure 5: Tabular format for the Functional Dependencies

Left	Middle	Right
A,C		B,D

Form Figure 5, it is clear that B, D are not prime attribute of the database relation. *AC* is the candidate key of the relation. $A \rightarrow B$ and $C \rightarrow D$ shows partial dependency in the table. So, the table is not in 2NF.Now, decompose the table into three different table $R_1(A,B)$, $R_2(C,D)$ and $R_3(A,C)$ .In the relation $R_1(A,B)$ dependency $A \rightarrow B$, where A is the key of the relation, which is in 2NF and 3NF and BCNF.The relation $R_2(C,D)$ dependency $C \rightarrow D$, where C is the key of the relation, which is in 2NF and 3NF and BCNF i.e. there is no partial, no transitive dependency and C is the super key of the relation relation $R_3(A,C)$, whenever there is no functional dependency on a relation, then all the attribute of a relation collectively form a candidate key and it is obviously in BCNF. Now, relation $R(A,B,C,D)$ is not in BCNF, after decomposing the relations into three different relations $R_1(A,B)$, $R_2(C,D)$ and $R_3(A,C)$ are in BCNF and the all functional dependency are preserved.

V. CONCLUSION

A new way of key generation method has been applied. It is built on the tabular approach algorithm method. This method is divided into one row and three columns, earlier we are using the closure based candidate key generation method. This method is automated generation of super keys and candidate key using tabular approach algorithm. The method removed the non-prime attribute of the database relation, which will be never considered as candidate key in the relation during designing phase. The attribute(s) which resides right marked side of the tabular approach table will not be the part of key. This method can also find the key of table, when there is no information about the attribute is given in the FDs. In other hand, it reduced the time complexity to generating key and human efforts.

VI. ACKNOWLEDGEMENT

This work was supported by the University Department of Statistics and Computer Applications, Tilka Manjhi Bhagalpur University, Bhagalpur.

REFERENCES

[1] S. Lucas, J. Meseguera , "Normal forms and normal theories in Conditional rewriting" , Elsevier Journal of Logical and Algebraic Methods in Programming, 85, 67–97, 2016.

[2] S. K. Singh, "Database Systems-Concepts, Design and Application," Publisher Dorling Kindesley, New Delhi India, 2009

[3] S. Lucas, J. Meseguer, "Localized operational termination in general logics, in: R.D". N. , R. Hennicker, Software, Services, and Systems: Lecture. Notes Computer. Sci., vol.8950, Springer, 2015, pp.91–114.

[4] Moussa Demba," Algorithm for relational database Normalization up to 3NF" International Journal of Database Management Systems Vol.5, No.3, June, 2013

[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] W.-D. Langeveldt, S. Link, Empirical evidence for the usefulness of Armstrong relations in the acquisition of meaningful functional dependencies,Inf. Syst. 35 (3) 352–374.2010

[7] Bourque, P. and Dupuis, R. Guide to the Software Engineering Body of Knowledge. IEEE Software, November, December, pp.35 – 44.1999

[8] Vimala, S., Khanna N, H., Saranya, G. and Kannan, A. "Applying Game Theory to Restructure PL|SQL Code", International Journal of Soft Computing, Vol.7, No.6, pp.264-270, 2012.

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

[10] R. Cowie and R. Cornelius, "Describing the Emotional States That Are Expressed in Speech," Speech Comm., vol. 40, no. 1-2, pp. 5-32, April 2003.

[11] A. H. Bahmani, Md. Naghibzadeh, B. Bahmani "Automatic database normalization and primary key generation" IEEE CCECE OR CCGEI May 5-7, Niagar Falls, Canada,2008.

[12] Gary D. Boetticher, "Computing program at UHCL changing the world bit by bit", University of Houston-Clear Lake Graduate Database", Graduate database course, Department of computer science, , 15 – 33, 2011

[13] Vimala, S., K. Nehemiah, H., Bhuvaneswaran, R.S. and Saranya, G. "Design Methodology for Relational Databases: Issues Related to Ternary Relationships in Entity-Relationship Model and Higher Normal Forms", IJDMS, Vol.5, No.3, pp.15-37, 2013.

[14] Yazici, A. and Karakaya, Z.: Normalizing Relational Database Schemas Using Mathematica, LNCS, Springer-Verlag, Vol.3992 (2006) 375-382.

Mr. Kunal Kumar is currently working as Research Scholar toward the PhD degree at the University Department of Statistics and Computer Applications. Tilka Manjhi Bhagalpur University, Bhagalpur, India. He received the Master of computer applications and Bachelor of computer applications degree from the Birla Institute Of Technology, Mesra , Ranchi, India in year 2012 and 2008. His area of research interests includes Database theory, Database Design and Big Data.

Dr. S. K. Azad is an Associate Professor and Head, in University Department of Statistics and Computer Applications at Tilka Manjhi Bhagalpur University, Bhagalpur, India.