An Efficient Extension Of Conditional Functional Dependencies Using Nested Relational Database

Suchitra Reyya[1], Sateesh Gudla[2], M.Prasanna Shivani[3]

[1]*Assistant Professor, Department of CSE,LIET,,Vizianagaram.*

[2]*Associate Professor, Department of CSE, LIET,Vizianagaram.*

[3]*Department of IT, LIET, Vizianagaram.*

Abstract

This paper propose an efficient data cleaning by using extended Conditional Functional Dependencies (eCFD's), which is an extension of Conditional Functional Dependencies(CFD's). eCFD's intend to solve the multi-valued inconsistencies to trounce drawbacks of CFD's which use pattern tableau to hold individual tuples in a table for cleaning relational data by supporting only single valued attributes. SQL techniques are used to create patterns of semantically related values for detecting single tuple CFD violations. By introducing a query and an algorithm we provide better competence for eliminating data redundancy in multi-valued attributes using nested relational database. We experimentally analyze the efficiency and performance of these eCFD based techniques in improving data quality and displays the tentative results graphically.

Keywords*: CFD's, eCFD's, Nested relational database.*

1. Introduction

Data warehouse consists of huge amount of data integrated from different database sets where, the chances of data being redundant and inconsistent are highly possible. If these warehouses are not frequently cleaned, the inconsistent data can lead to mis conceptions there by making the cleaning process more complex [7]. Data redundancy implies that different tuples may represent the same data which causes anomalies and data corruption. Thus data cleaning plays a prominent role in preprocessing. One way of achieving this is by using functional dependencies. In FD's data redundancy is reduced by enforcing integrity constraints at schema level. But the extension of FD's called CFD's were developed which aim at capturing violations for individual tuples by using the concept of pattern tables [1].When these pattern tables are implied on the entire database the tuples which violate the pattern tables are detected and hence can be easily corrected. The downside with CFD's is that they hold for only single value attributes but not for multivalued attributes [2].

This paper deals with eCFD's for improving data quality which are extended from CFD's to trounce the problem of multi-valued attributes by using nested relational database [5][6]. eCFD's are capable of capturing inconsistencies using nesting.

This paper is organized into different sections where section 1 gives a brief introduction to data cleaning concept, section 2 thrash out related work, section 3 pioneer how to purge redundancy, section 4 explains methodology to reduce redundancy by introducing algorithm, section 5 displays experimental results and section 6 holds the conclusion.

2. Related Work

Relations that contain redundant information may potentially endure from update anomalies.Functional dependencies are used to detect the redundancies by enforcing integrity constraints at schema level [1]. The constraints hold for all the tuples in the table. The main idea behind FD is that the value of a particular attribute uniquely determine the value of some other attribute [8] A relation A → B if "for every valid instance of A, that value of A uniquely determines the value of B".

Definition (Conditional Functional Dependencies): Refinement of FD's termed as CFD's capture inconsistencies that traditional FD's cannot detect. They are not expressible as traditional FD's because CFD's do not hold not entire relation, instead it holds on tuples matching the conditions specified in the pattern table [1].

A CFD ϕ on R is a pair (R : A → B, Tp), where (i) A, B are sets of attributes from relation (R), (ii) R : A → B is a customary FD, rooted in ϕ; and (iii) Tp is a tableau with all attributes in A and B , where for each X in A or B and each tuple t ∈ Tp, t[X] is either a constant 'a' , or an unnamed variable '- '.
Given an instance I of a relation schema *R* and a set Σ of CFDs on *R*, it is to find all the tuples that violate some CFD in Σ. The query below is an SQL technique for detecting violations of a CFD [2].

Select
distinct t.A
from R t, Tp tp
where t[A1] = tp[A1] AND . . .
AND t[An] = tp[An]
group by t.A **having** count(**distinct** B)> 1

This query group's tuples with the same value on attribute A and it counts the number of distinct values in B. The query returns inconsistent tuples in the database if there is more than one value for B.

3. Eliminating Redundancy using Nested Relational Database

This paper proposes eCFD's which hold for multi value data. This is achieved using nested database [6]. The overall design of the system is shown below.

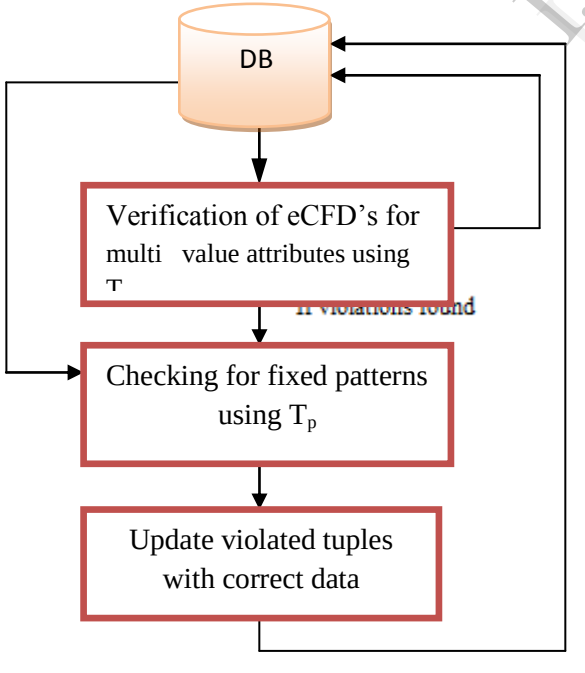

Figure 1: A model for achieving eCFd's

The model architecture describes the step by step flow of process carried to implement eCFDs. In the first step the data in the database is compared with table T_m to check for multi value inconsistencies. In the second step if the inconsistencies are found then it compares each tuple with matching fields from the database and table T_m with pattern table T_p. In the third step finally the incorrect tuple is updated with matching value from the multi tableau and saved in database.
Defnition (eCFD): Let R be some relational schema given as (R:X→Y, Tp,Tm) where (1) X,Y,Tp,Tm ∈ attr(R); (2) X→Y is an embedded FD; (3) Tp is a pattern tableau consisting of finite number of pattern tuples; (4) Tm is a multi tableau holding multiple values of Y for each X attribute.
Example 1.1: Let us consider a schema cust(AC ,PN ,NM, STR, CT, ZIP). It specifies a customer relationship in terms of the customer's phone (area code (AC), phone number (PN)), name (NM), and address (street (STR), city (CT), zip code (ZIP)). An instance D_0 of cust is shown below.

Table 1: Customer Database (CD)

AC	PN	NM	STR	CT	ZIP
718	1111111	Mik	TreeAve	Albany	12238
518	2222222	Joe	Elm Str	Colonie	12205
100	2222222	Jim	Oak Str	Troy	12181
212	3333333	Ben	5th Ave	NYC	10001
646	4444444	Ian	High St	NYC	10011
0891	2552705	Vani	Eenadu	VSP	530013
08922	2342345	Rani	Boypalm	VSP	530012
345	6784563	Pete	Troy	Ohio	23124
0890	1233211	Raja	Eenadu	VSP	530045

In the above table we can observe that the city NYC and VSP has different area codes (AC).
Φ_1: CT ∈ {NYC} → ϕ with AC ∈ {212, 646, 347, 917}.
Φ_2: CT ∈ {VIZAG} → ϕ with AC ∈ {0891, 08922}.

Hence in ϕ_1 if the city is NYC then the area codes must be one of the following values. Similarly in Φ_2 if the city is VIZAG the area code must be one of the above given values. Here CT is associated with disjunction of options rather than single value. But in the above example tuple t_3 contains CT as NYC but AC is 100 which is not one of the area codes associated with NYC and tuple t_9 has CT as VSP but AC is 0890. This problem is not overcome in CFD's as it does not hold multi-valued attributes. This multi-valued attribute problem can be solved with by less complexity by taking two tables Tm & Tp.

TABLE T_m: Tm is a multi tableau holding all the tuples consisting of multi- value of Y for a single attribute X. The Tm table for the given customer (cust) instance is shown below.

Table 2: multi tableau-T_m

CT	AC
NYC	{212,646,347,917}
VSP	{0891,08922}
Albany	{718}

Now to correct the area code in tuple t4 appropriate area code must be selected from multiple values of AC in table Tm. So we use the pattern table Tp which consists of fields, street(STR) and area code(AC) to select the area code that match the fields in the tuple. The pattern tableau for the cust relation is shown below.

Table 3: Pattern Tableau-T_p

STR	AC
5th Ave	212
8th Ave	212
High.St	646
Eenadu	0891
Boypalem	08922

From cust relation we observe that in the tuple t4, street is 8th AVE and city is NYC which uniquely determines area code as 212. Thus now the violation showed in tuple t4 can be eliminated by correcting it with area code as 212 using these two tables.

4. Methodology to Reduce Redundancy

We introduce a query for solving multi value inconsistencies in a simplified way with less complexity.
This query shows the violated tuples by considering customer relation table(C) and multi tableau(Tm) where the city's area code (AC) does not match with the disjunction of options present in the multi tableau T_m.

select ALL
from cust C, table1 T_m
where (T_m.CT < > '@'
AND T_m.AC[i….i_n] < > '@')
AND (C.CT = T_m.CT
AND C.AC < > T_m.AC[i…..i_n]);

Once the violations are displayed the area code must be corrected with the appropriate one from the multiple area codes.
The below query is used to update the area code by considering customer relation table (C), multi tableau (T_m) and pattern tableau (T_p) where the area code of T_m matches with that of T_p and the street in customer relation matches with that of T_p.

update
cust set
C.AC = (select T_p.AC
from table2 T_p, table1 T_m, cust C
where T_p.AC = T_m.AC[i….i_n]
AND C.STR = T_p.STR);

The following Algorithm can be used to explain the methodology to reduce redundancy using nesting.

Algorithm : Eliminating inconsistent tuples in multi-valued attribute using nested relational database

Input: Inconsistent multi-valued database (CD)
Output: Clean database(CCD)
Method: Solving multi-valued inconsistencies using Nested Relational database

1. **Begin**
2. Consider a table CD with different fields.
3. Take a new table multi tableau($Tm_)$ having multiple values for attribute X ie. X [i…..i_n].
4. **For** each tuple in table CD having attr X, compare it with X [i…...i_n] of table T_m.
5. **If** comparision successful then goto step 15
6. **Else**
7. Display tuples not satisfying step 4.
8. **End for**
9. Consider a pattern tableau T_p having prefixed constants and unnamed variables.
10. For updating the inconsistent values shown in step 7, for each tuple in table CD compare T_m.X [i….i_n] with T_p.X and CD.Y with T_p.Y respectively.
11. **If** comparision successful then
12. Update attr X with tp.X
13. **End if**
14. **End for**

15. Original database (CD) is updated as Clean Database (CCD)
16. **End**

Now after applying the above algorithm on TABLE 1, the consistencies in the table will be replaced by the accurate data, thereby making the database clean and error free. The below is the clean database (TABLE 4) obtained as a result of the algorithm. The AC in tuple t_4 is replaced with 212 and similarly tuple t_9 with erroneous AC is replaced with appropriate area code (AC) as 0891.

Table 4: Cleaned Customer Database (CCD)

AC	PN	NM	STR	CT	ZIP
718	1111111	Mik	TreeAve	Albany	12238
518	2222222	Joe	Elm Str	Colonie	12205
212	2222222	Jim	Oak Str	Troy	12181
212	3333333	Ben	5th Ave	NYC	10001
646	4444444	Ian	High St	NYC	10011
0891	2552705	Vani	Eenadu	VSP	530013
08922	2342345	Rani	Boypalm	VSP	530012
345	6784563	Pete	Troy	Ohio	23124
0891	1233211	Raja	Eenadu	VSP	530045

5. Experimental Analysis

In this section, we present our findings about the performance of our models for reducing redundancies over multi-value databases.

Setup: For the experiments, we used postgre SQL on Windows XP with 1.86 GHz Power PC dual CPU with 3GB of RAM.

Efficiency of eCFD's*:* The efficiency of the eCFD's is graphically displayed by considering the multi-value database. When CFD's are posted on this database all tuples having multiple values for each attribute are changed to only one value and obtains duplicate data. But in eCFD's each attribute can hold multiple data thereby preserving the data.

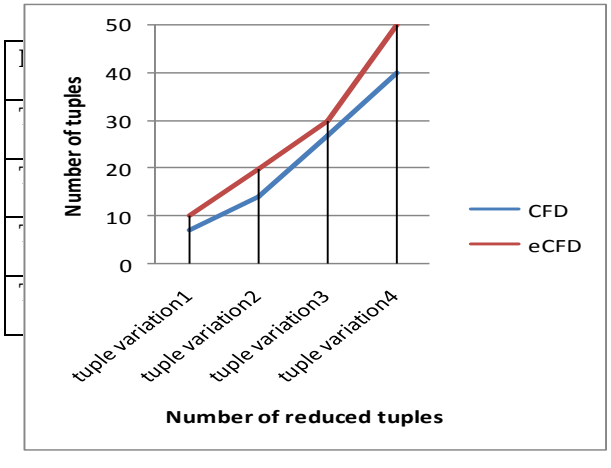

Figure 2: Graph displaying the efficiency of eCFD's

Size of the Relation*:* In this experiment we have analyzed how the redundancy is reduced by considering the space occupied by the pattern table (T_P) and multi tableau (T_m) in CFD's and eCFD's respectively. It is observed T_p can hold only one value for each fixed pattern where as T_m can efficiently hold multiple values for each attribute in the pattern thus reducing tuple size

Table 6: Comparison of size of tuples in T_p (CFD's) & T_m(eCFD's)

No. of multiple values	**T_p**	**T_m**
Pattern 1	1	2
Pattern 2	1	3
Pattern 3	1	4

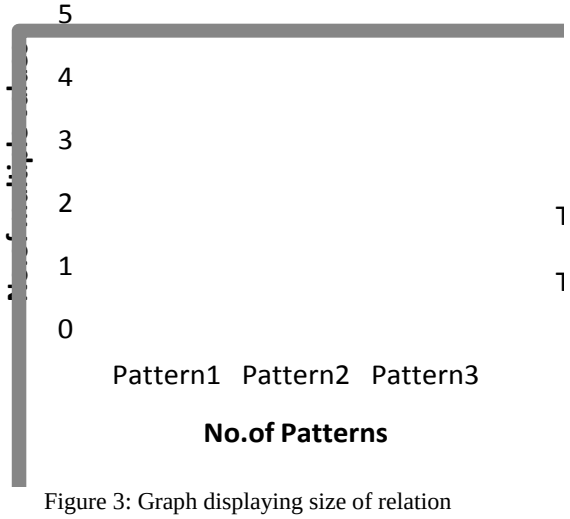

Figure 3: Graph displaying size of relation

6. CONCLUSION

We presented that nesting based eCFD's proved to be better than CFD's in eliminating redundancy in nested relational database. eCFD's acquire no extra complexity in inconsistency detection. We have developed SQL based technique and algorithm for eCFD violation detection. Our experimental results for efficiency and size of the relation proved eCFD's to be more effective compared to CFD's.

7. REFERENCES

[1] Philip Bohannon, Wenfei Fan, Floris Geerts and Xibei Jia3 Anastasios Kementsietsidis3: *Conditional Functional Dependencies for Data Cleaning*

[2] Wenfei Fan, Floris Geerts , Laks V.S. Lakshmanan and Ming Xiong *: Discovering Conditional Functional Dependencies*, IEEE conference on data engineering

[3] Sangeeta Viswanadham, V. Valli Kumari: *Eliminating Data Redundancy using Nesting based wMVD*, IEEE Conference Publicat ions, Elect ronics Computer Technology (ICECT), 2012 4th Internat ional Conference, April (2012).

[4] Sangeeta Viswanadham, V. Valli Kumari: *Eliminating Data Redundancy*, Cube (2012)

[5] Suchitra Reyya, Sangeeta Viswanadham: *Eliminating Data Redundancy through Weak Multivalued Dependencies using Nesting*

[6] Suchitra Reyya, M. Sundara Babu, Ratnam Dodda: *An Efficient Storage of Spatial Database using Nested Relational Database* , International Journal of Engineering Research & Technology (IJERT) Vol. 1 Issue 7, September – 2012

[7] Erhard Rahm, Hong Hai Do: *Data Cleaning Problems and Current Approaches*, IEEE Techn. Bulletetin on Data Engineering, Dec. (2000)

[8] Korth, H., Roth, M.: *Query languages for Nested Relational Databases, Nested Relations and Complex Objects in Databases*, Springer, (1991).