RESEARCH ARTICLE OPEN ACCESS

Confiscation of Duplicate Tuples in The Relational Databases

Dr.K.VenkataRamana*, Dr.G.V.Ramesh Babu**
*Department of Computer Science, KMM. Institute of Post Graduate Studies, Tirupati
** Department of Computer Science, S.V.University,Tirupati

ABSTRACT
Relational database is a collection of relations. Duplicate tuple existence is common in many real time relational databases. There is no known, simple and direct technique for finding duplicated records in relational database. In a relational database, if the same real-world entity is represented by more than one tuple, then such tuples are called duplicate tuples. Finding duplicate tuples and then replacing them by one best tuple is called a fusion operation. Whenever duplicate tuples are found in the relations of any database, those tuples must be replaced with one special best approximate tuple that represents the joint information of all the duplicate tuples. Present study proposes new techniques to find duplicate tuples and then remove those duplicate tuples with the correct real world tuples. In the first step duplicate tuples in the relation are classified based on the class label and in the second step then for each set of duplicate tuples functional dependency method or union method is applied to replace duplicate tuples with the corresponding correct real world single tuple. One possibility is to replace one set of duplicate tuples with one correct real world tuple. Another possibility is to replace two or more sets of duplicate tuples in the relation by one set of correct real world tuples. Sometimes the removal of duplicate tuples in the relations of any relational database can create referential integrity violations. All such violations must be controlled and coordinated syntactically as well as semantically in relations

Keywords:– Relational Database, Duplicate Tuples, Joins, Union

I. INTRODUCTION

The purpose of this paper is to focus on duplicate record detection algorithms in relational databases. In many real time applications in current scenario the data that exists in relations in the databases are inherently associated with duplicate tuples. Finding and then removing of duplicate data tuples in the relational database is the most important and latest research topic. In the context of relational databases, dealing with duplicates comes down to (a) identifying which tuples are duplicate and (b) replacing those tuples by a single tuple [1]. Data duplication is also known as entity resolution or record linkage [1]. Duplicate records do not share a common key and/or they contain errors that make duplicate matching a difficult task [2]. Duplicate data tuples are present in the one or more relational databases when there exit multiple descriptions of the same real world entity. Errors are introduced as the result of transcription errors, incomplete information, lack of standard formats, or any combination of these factors [2]. The presence of duplicate tuples causes many database maintenance problems. Some of the reasons for the existence of duplicate tuples are presence of missing attribute values, data entry errors, typing errors and not following standards in data entry and data maintenance. Finding and then removing of duplicate data tuples in the relational database is the most important and latest research topic. In general, data tuples are duplicated in one or more relations of any relational databases when there exit multiple descriptions of the same real world entity (record). Often, in the real world, entities have two or more representations in databases [2]. Duplicate tuple detection and replacement with correct tuple is inevitable in many relations of the relational database. Data fusion is the step of actually merging multiple, duplicate tuples into a single representation of a real world object [3].

A crucial operation in the maintenance of data quality in relational databases is to remove tuples that mutually describe the same entity (i.e., duplicate tuples) and to replace them with a tuple that minimizes information loss [4]. A special procedure is needed to take care of integrity constraint violations that occur when duplicate tuples are removed from the relations. One way is to develop a general procedure that not only covers integrity constraint violation management but also manages semantic relationships among the relations in the relational database. Duplicate record detection is the process of identifying different or multiple records that refer to one unique real-world entity or object [2].

Traditional approaches to entity resolution and de-duplication use a variety of attribute similarity measures, often based on approximate string-matching Criteria [5]. Many mathematical tools or models are available for efficient management of syntax as well as semantic modifications in the relations of any relational database. A correct way of propagation is to take into account that multiple tuples are fused, meaning that linked information

should be fused accordingly [1]. Integrity constraints imposed on the relational database must be satisfied before and after deleting the original duplicate data tuples. First determine all such duplicate tuples in the relations of any relational database and then replace all such duplicate tuples by a single correct tuple. Particularly referential integrity must be considered and controlled in propagation of data fusion. Several integrity constraints management strategies such as on delete cascade, on update cascade, set null, set not null, restrict are available in database modifications. These techniques are syntactically correct but semantically incorrect.

Present study proposes a new method to eliminate duplicate tuples in the relations of a relational database. This new technique is called union fusion function technique that is applicable for attribute values. Present study also proposes another duplicate tuple replacing technique using functional dependency approach. This is a more generalized functional dependency approach that covers both the partial preservative functions and also complete (full) preservative functions.

Present study also proposes another new technique to model a technique for removal of duplicate tuples in the relations. For example, assume that a relation contains many tuples. In order to find and then remove duplicate tuples in the given relation, initially we apply a classification technique to classify all the tuples, and then based on the class labels, and duplicate tuples are identified and then these duplicate tuples are replaced by the correct real world tuple. K-nearest neighbor classifier is one best tool in machine learning as well as in data mining for finding and then remove duplicate records. Decision tree is the probably most important and highly interpretable classification technique to the data. Decision tree is used as a benchmark technique before applying any classification technique. Also the time complexity of decision tree is (n x number of attributes x log n) where n is the number of tuples in the relation. Duplicate records slow down the indexing process and significantly increase the cost for saving and managing data [6].

II. PROPOSED WORK

Data duplication is common in many real time applications particularly in the relations of any relational databases. Finding duplicate tuples and then replacing them by one best tuple is called a fusion operation. During fusion operation integrity constraint violations must be controlled carefully and relational database must be managed in a consistence way before and after database modifications as well as after removal of duplicate tuples in the relations of relational databases.

In the present research paper, a sample set of three relations viz, 1.COLLEGE, 2.CONFERENCE and 3.CONDUCTED_CONFERENCES is considered as running example for understanding purpose. In the relation COLLEGE tuples 1 and 2 are duplicated because of some reasons such as typographic errors, missing of values and lack of standard data representation procedures etc.

Both one and two duplicate tuples describe the same real world entity. These two duplicate tuples are identified and consequently replaced by one equivalent real and correct tuple. Finding and then removing these duplicate tuples with one correct and real world tuple is called a fusion operation. Present study also proposes a new fusion operation called union. Union fusion operation accepts a set of duplicate tuples and then replaces with one correct real world entity. Working principle of union fusion function operation is explained below:
Union of College_Code = {3G} U{3G} = {3G}
Union of College_Name = {KMM} U {KMM} = {KMM}
Union of Principal_Name = {Rama} U {null} = {Rama}
Union of Affiliated_University = {null} U {JNT University} = {JNT University}
In the COLLEGE relation duplicate tuples 1 and 2 are replaced by the following single tuple using proposed new union fusion function technique. The replacing function may be either partially preservative or complete preservative function. Partially preservative function is defined as follows: There exists t ε DupSet such that t[A] = REP(Dup)[A]

When A $\subseteq$ DupSet, it is called partial preservative and when A = DupSet, it is called complete preservative replace function.
For example, let A = {SNo, College_Code, College_Name, Principal_Name}
Here t[A] = Rep(Dup)[A] and
Let B = {JNT University}, then t[B] = Rep[B]

In this particular example, replacing function is partial preservative but not complete (full) preservative. Hence, 1 and 2 duplicate tuples in the COLLEGE relation are mapped with one correct real world tuple. In the COLLEGE relation, tuples 5, 6, and 7 are duplicate tuples. This is an example for complete preservative. These three duplicate tuples are shown in the FIGURE 6 and then they are replaced by the single tuple shown in the FIGURE 7.

Here, t[all attributes] = RepDup[all attributes]. Complete preservative replacing function replaces a set of tuples with another equivalent and simplified set of tuples.

Consider once again the relation COLLEGE1 with the functional dependency that holds on it, College_Code

→ {College_Name,Principal_Name,Affiliatedto}. The functional dependency states that when two values on different tuples are same on the attribute College_Code then all values of the three attributes in the right side of the functional dependency are also same. That is, if t_1[College_Code] = t_2[College_Code] then t_1[College_Name,Principal_Name,Affiliatedto] = t_2[College_Name,Principal_Name,Affiliatedto].
Therefore duplicate tuples 1 and 2 in the COLLEGE relation are replaced by tuple 1 by applying functional dependency constraints.

Sometimes it may be necessary to take multiple sets of tuples and map them into a single set of tuples. We use union operation to map multiple sets of tuples into a single set of tuples. This union operation is more generalized version of many database operations such as delete, cascading, and referential integrity etc. First step is to find and replace duplicate tuples and then remove problems that will arise after duplicate tuple replacement. First step is performed using union of values of attributes and then second point is executed by applying union of sets of tuples.
The relation ONDUCTED_CONFERENCES contains totally nine tuples. In this relation all foreign key values that contain 2 are replaced by value 1. This is due to first type of partial preservative function or functional dependency rule. Using complete (full) preservative function foreign key values 5, 6, and 7 in the relation CONDUCTED_CONFERENCES are replaced by 5.

Table-1 Tuples showing College Table Data in database

SNo	College_Code	College_Name	Principal_Name	Affiliated_University
1	3G	KMM	Rama	Null
2	3G	KMM	Null	JNT University
3	3C	Vidyanikethan	CSReddy	Null
4	2D	SHREE	Null	JNT University
5	6E	Annamacharya	Dr.MuniSwamy	JNT University
6	6E	Annamacharya	Dr.MuniSwamy	JNT University
7	6E	Annamacharya	Dr.MuniSwamy	JNT University

Table-2 Tuples showing Conference Table Data in database

Conference_Id	**Conference_Type**
Conference1	ELSEWARE
Conference2	SPRINGER
Conference3	ACM
Conference4	IEEE

Table-3 Tuples showing Conducted Conferences Table Data in database

Sr.No	**Confrence_Id**	**Number of days**	**Start_Date**
1	Conference 1	3	18/6/2009
1	Conference 2	1	10/12/2006
1	Conference 3	4	3/9/2012
2	Conference 1	3	8/6/2009
2	Conference 2	1	10/12/2006
3	Conference 1	6	3/5/2013
5	Conference 1	4	29/12/2010
6	Conference 1	5	29/12/2010
7	Conference 1	6	29/12/2010

Table-4 Duplicate Tuples originally present in the COLLEGE Table Data in database

SNo	College_Code	College_Name	Principal_Name	Affiliated_University
1	3G	KMM	Rama	Null
2	3G	KMM	Null	JNT University

Table-5 Tuple showing the correct replacement of tuples 1and 2 in COLLEGE Table Data in database

SNo	College_Code	College_Name	Principal_Name	Affiliated_University
1	3G	KMM	Rama	JNT University

Table-6 Duplicate Tuples originally present with complete preservation in the COLLEGE Table Data in database

SNo	College_Code	College_Name	Principal_Name	Affiliated_University
5	6E	Annamacharya	Dr.MuniSwamy	JNT University
6	6E	Annamacharya	Dr.MuniSwamy	JNT University
7	6E	Annamacharya	Dr.MuniSwamy	JNT University

Table-7 Tuple showing the correct replacement of tuples 5, 6and 7 in COLLEGE Table Data in database

SNo	College_Code	College_Name	Principal_Name	Affiliated_University
5	6E	Annamacharya	Dr.MuniSwamy	JNT University

Table-8 Tuples showing the COLLEGE relation after removing the duplicate tuples

SNo	College_Code	College_Name	Principal_Name	Affiliated_University
1	3G	KMM	Rama	JNT University
3	3C	Vidyanikethan	CSReddy	Null
4	2D	SHREE	Null	JNT University
5	6E	Annamacharya	Dr.MuniSwamy	JNT University

Table-9 Tuples showing the CONDUCTED_CONFERENCE relation after tuple propagation

SNo	Confrence_Id	Numberof_days	Start_Date
1	Conference1	3	18/6/2009
1	Conference2	1	10/12/2006
1	Conference3	4	3/9/2012
1	Conference 1	3	18/6/2009
1	Conference2	1	10/12/2006
3	Conference1	6	3/5/2013
5	Conference1	4	29/12/2010
5	Conference1	5	29/12/2010
5	Conference1	6	29/12/2010

Table-10 Tuples showing the CONDUCTED_CONFERENCE relation after tuple propagation with respect to the replacement tuple1

SNo	Confrence_Id	Numberof_days	Start_Date
1	Conference1	3	18/6/2009
1	Conference2	1	10/12/2006
1	Conference3	4	3/9/2012
1	Conference1	3	18/6/2009
1	Conference2	1	10/12/2006

Table-11 Tuple showing the CONDUCTED_CONFERENCE relation t is not modified in the COLLEGE

SNo	Confrence_Id	Numberof_days	Start_Date
3	Conference1	6	3/5/2013

Table-12 Tuples showing the CONDUCTED_CONFERENCE relation after tuple propagation with respect to the replacement tuple 5

SNo	Confrence_Id	Numberof_days	Start_Date
5	Con1	4	29/12/2010
5	Con1	5	29/12/2010
5	Con1	6	29/12/2010

Table-13 Tuples showing the CONDUCTED_CONFERENCE relation after removing all the duplicate tuples

SNo	Confrence_Id	Numberof_days	Start_Date
1	Con1	3	18/6/2009
1	Con2	1	10/12/2006
1	Con3	4	3/9/2012
3	Con1	6	3/5/2013
5	Con1	4	29/12/2010

First proposed method takes union among the attributes. Second proposed method takes union among the tuples. Third proposed functional dependency method is more generalized version of the above two methods. Third proposed method also takes care of partial and complete preservative functions also.

III. ALGORITHM

Let R be a relation of tuples and assume that the set of duplicate tuples are denoted by delta. That is, delta ⊆ R. Let R^' be the child relation corresponding to the parent relation, R. this algorithm will be executed in two steps. In the first step duplicate records are identified and then in the second step identified duplicated records are replaced with the correct real world records and also these changes are propagated to the dependent (referenced) relations in a semantically correct way in addition to the syntactic correctness of relations with respect to many database operations such as insert, delete, and update.

Assume that sample parent relation R = COLLEGE, and the dependent child relation of the parent relation is taken as R^' = CONDUCTED_CONFERENCES. Also assume that tuples t ϵ delta ⊆ R and tuples t* ⊆ R^'.

The relationship between parent and child relations is one to many from COLLEGE to CONDUCTED_CONFERENCES.

In the COLLEGE relation tuples 1 and 2 are duplicated and this type of duplication is deleted using union operation of between or among the attributes. Tuples 5, 6, and 7 are also duplicated and these types of duplication of records are removed by taking the union operation among the tuples but not among the attributes. In the second case sets of duplicate records are identified and then replaced with the one or more sets of real world and original or correct records.

INPUT:

Relations with duplicated tuples

OUTPUT:

Relations with duplicate tuples removed

1. For each tuple t ϵ delta do
2. In the relation R^' find a set off tuples whose foreign key matches with the primary key of the tuple t in R.
3. Let St be the set of such tuples
4 For all t* ϵ St replace foreign key values in R^' with the respective primary key of the tuple t ϵ R End for
End for
5. For each set st find projected set of tuples based on their primary key End for
6. Now apply union operation for all St Sets

IV. ANALYSIS OF RESULTS

The COLLEGE relation contains seven tuples. Tuples 1 and 2 are duplicated with respect to partial preservative function. Duplicated tuples 1 and 2 are shown in Table-4 and then these two duplicated tuples are replaced with one correct tuple shown in Table-5. Similarly, in the COLLEGE relation tuples 5, 6, and 7 are duplicated with respected to full or complete preservative function. These duplicated tuples 5, 6 and 7 are shown separately in Table-6 and then replaced with one correct tuple shown in Table-7.

The relation COLLEGE_CORRECTED is shown in Table-8 after removal of duplicate tuples with the replacement of correct tuples. The relation CONDUCTED_CONFERENCE is updated based on the updated details of the relation COLLEGE_CORRECTED and modified CONDUCTED_CONFERENCE relation is named as CONDUCTED_CONFERENCES_AFTER_PROPAGATION and is shown in Table-9. With respect to duplicate tuples 1 and 2 in the COLLEGE relation, modified tuples in the CONDUCTED_CONFERENCE relation are shown separately in the Table-10. Similarly duplicate tuples 5, 6, and 7 are replaced with tuple 5 and are shown in Table-12 separately. Tuples 3 and 4 in the COLEGE relation are not duplicated and they remain as it is. Tuple 3 is shown in Table-11.

Finally, COLLEGE relation after removal of duplicate tuples is shown in Table-8 and updated CONDUCTED_CONFERENCE relation is shown in Table-13.

V. CONCLUSIONS

When database records are duplicated, both storage and processing cost of records is very high. Future research interest is to find good algorithms to handle tuple duplication in many real applications such as catalogs, networks, Data duplication is common in many real life applications. Records are duplicated in many relational databases because of many reasons such as inclusions of null values, non-standard method representation, and typographic errors. There is no standard method for identification of duplicated records in the relations of relational databases. When there exist no specific standard method for detecting duplicate records it is very difficult to find duplicate records. Hence, there is a scope for formulating specific standard methods for duplicate record detection.

REFERENCES

[1]. Antoon Bronselaer, Daan Van Britsom, and Guy De Tre , “Propagation of Data Fusion IEEE Transactions on Knowledge and Data Enginee-ring,vol. 27, no. 5, may 2015

[2]. Ahmed K. Elmagarmid, Panagiotis G. Ipeirotis, Vassilios S. Verykios, Duplicate Record Detection: A Survey IEEE transactions on Knowledge and Data Engineering, Vol. 19, NO. 1, pp. 1–16, Jan. 2007.

[3]. Felix Naumann , Alexander Bilke, Jens Bleiholder, Melanie Weis Data Fusion in Three Steps: Research Paper Antoon Bronselaer,Marcin Szymczak, Sławomir Zadrożny, Guy De Tré, Dynamical order construction in data fusion, Published in: journal information fusion archive Volume 27 Issue C,

[4]. January2016, Pages 1-18, Elsevier Science I. Bhattacharya and L. Getoor, “Collective entity resolution in relational data,” ACM Trans.

[5]. Knowledge. Discovery Data, vol. 1, p. 2007, 2006.

[6]. Anestis Sitas, Sarantos Kapidakis,” Duplicate detection algorithms of bibliographi descriptions”, Library Hi Tech, Vol. 26 No. 2, 2008, pp. 287-301,

[7]. Emerald Group Publishing Limited, 0737-8831 DOI 10.1108/07378830810880379