Fend off Duplication Using Hashing Techniques

Prof. A. George Louis Raja [1], Mr. P. Arulanand [2], Mr. S. Ananda Krishnan [3],Mr. M. Israel Prabu [4]

[1,2,3,4.] PG Department of computer Application,
Sacred Heart College (Autonomous), (Affiliated to Thiruvalluvar University)
Tirupattur, Vellore Dis-635601, Tamailnadu, India

***Abstract*— A large amount of data is being generated throughout the world every minute. Huge Organizations like Finance, Bank, Government Sector and others need to store and retrieve massive amount of data. There are possibilities for redundancy when the data are being transferred to database. These redundant data causes many problems in the data base such as increase the memory, increase the access time and poor and inefficient data storage. In order to avoid the problem of redundancy a specifictechnique has been invented that is called as data de-duplication. This paper providesa solution to avoid the duplications that exists in the columns of a Table. There are many techniques in practice for avoiding the duplication in the rowsof a Table such as normalization. We aim to ignore the duplication in the column usingthe "hashing techniques".**

I. INTRODUCTION

Storing data plays an important role in every organization. Efficient decision making in organization is based on data by addressing the previous records. Nowadays government agencies and several private sectors need a large amount of data and analyses data to make it efficiently and compare data with lot of resources to improve the quality of the data. Quality of data is based on accurate and complete data. But in many cases data is not quality it is just because duplicate, Dirty data, inaccurate, incomplete or erroneous data.

Record linkage is frequently used to improve the quality of data. They have lot of techniques and functions to avoid the erroneous data like indexing, hashing and clustering. Those techniques are used to avoid the redundant data in the entity only. In order to avoid redundancy some techniques have been invented such as Normalization. The normalization technique also avoids the inaccurate data in the database table and save the memory in the database table.But this technique also has some drawbacks like having redundant data in Domain.

II. DESCRIPTION TO THE PROPOSED SOLUTION:

The problem of duplication in domain can be eradicated with help of hashing technique. By using the hashing technique we store data to the database table. The data is stored in the database table if it is not duplicate,and the data could be stored in the table as it is. If the data is duplicate data will not be saved in the database table, rather duplicated data will be stored in a hash table bucket slots.The bucket slots havea reference id. Thereference id will be passed into the database table field.

III. THE PROPOSED SOLUTION

There are different types of hashing techniques such as linear hashing, static hashing, dynamic hashing, internal hashing and external hashing.Using the external hashing we may avoid the duplicate data in database.

Existing database tables have repeated data in the domain, our aim is to avoid the repeated data in the domain. Using the hashing concept we may avoid the duplicate data in the domain. In this hashing concept they have lot of buckets, the buckets have lot of slots. Using this external hashing function we can avoid the repeated data in main table and also in hash table.

Using this external hashing function the value will be stored in the hashing table.Onlythe reference value will be passed through the database table. If any value enters in the database table, first and foremost we have to check the hash table whether the value is existing or not.If the value already exists in hash table the reference value is passed from hash table to main table.If the value does not exist, the data will be stored in the hash table which is noted in a reference id, the id alone is sent to the main table to be stored.

IV. STRUCTURE OF THE HASH TABLE

The structure of the hash table contains buckets and slots.A hash table can have a lot of bucket which may have lot of slots. Each slot stores only one value.

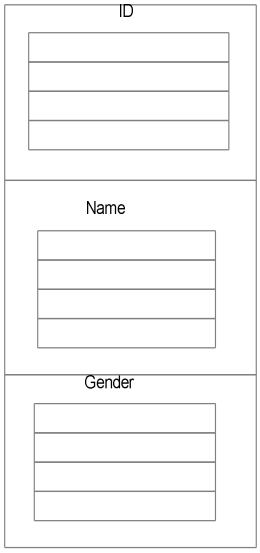
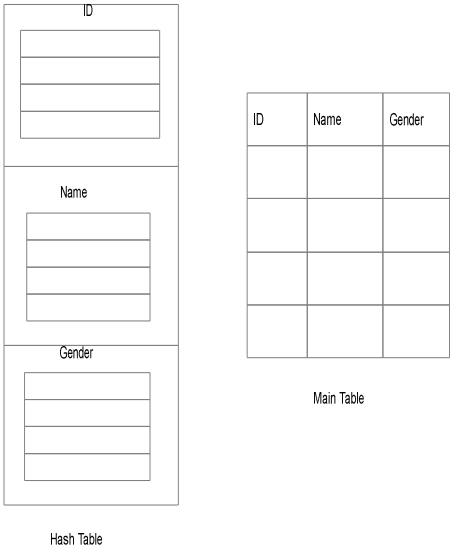


Fig 4.1

Hash Table

V. PROPOSED SOLUTION STRUCTURE

In the above two tables the first table is hashing table and the second table is reference table. If the data is given to these tables, the value will be stored in the hash table bucket and reference is stored in the main table.



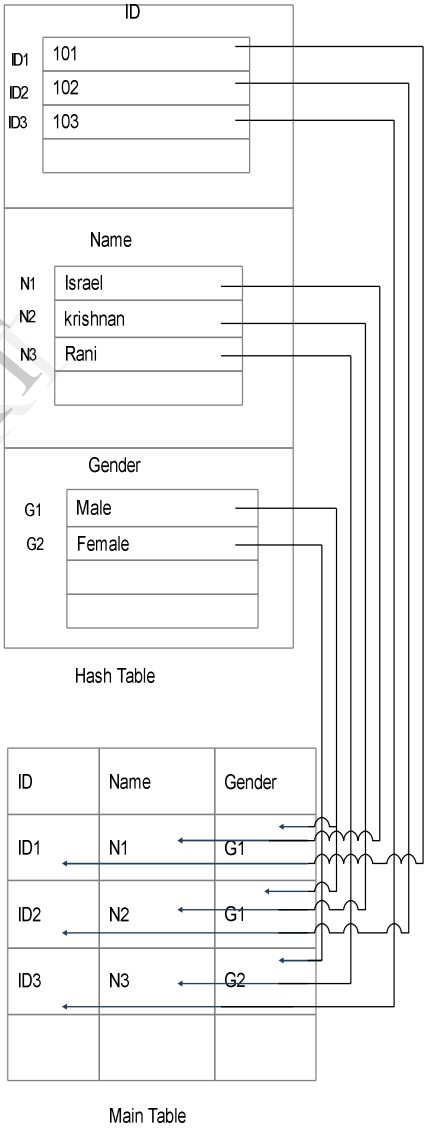
VI. INSERTING:

To insert the data into the hash table, first check whether the data is already available or not in the hash table bucket. If the data is not available in the hash table bucket means that the data will be stored in the hash table bucket.Later the data reference will passed to the main table, in the main table thereference will be stored in a certain column. Suppose the data is already available in thehash table,the data will not be stored in the hash table bucket, instead thereference is passedto the main table. Likewise the redundant data is avoided by using the hashing technique.

For example,

A main table hasthree fields like ID, Name and Gender. Hash table also hassame three bucketslikeID, Name and Gender. Bucket is referred to field in hash table.

The inserting process, first we check all the particular bucket slots if the data is redundant or not. Take this example, take



the first row ID value is 101, Name value is Israel and gender value is Male. First the ID value is checked whether the ID is already existinginhash table bucket. If the Id is not existing in the ID bucket slots, the ID value get stored in the ID bucket slots. After inserting the value of ID in the ID slots the ID of reference will be passed to ID field of the main table. Main

table consists only the references. In the same process will occur in the all the buckets. In that place memory will be saved in hash table and the main table.

In the second row the third field value Gender, the Gender value is already existing in G1. G1 is set as reference of Gender. In that place the hash table Gender bucket is checked whether the data is already existing or not, if the data is already exist means, that data will not be stored in the Gender bucket slots. Only the reference will be passedto the main table as G1.

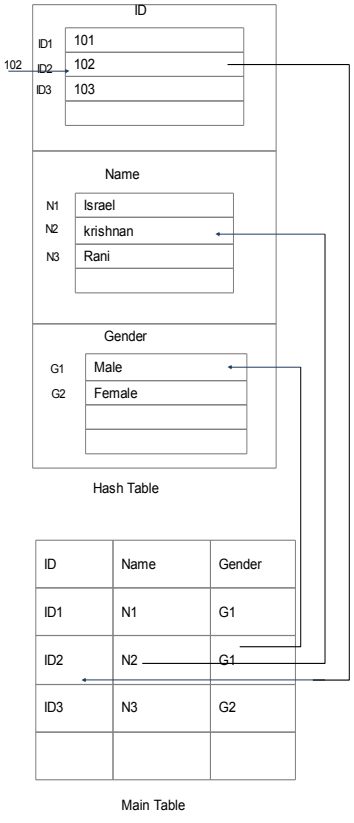
VII. RETRIEVE

If we want to search and retrieve the values of particular ID we have to enter the ID value and search it.

Once we enter the ID , the ID values goes to the hash table bucket and find where the ID bucket is available in the hash table. If the ID bucket is found, it searches whether ID is existing in the ID bucket. If ID is existing it is possible to retrieve record. If ID is not existing it is understood that the record is not present in the main table.

For example,

In the following diagram, we search the 102 ID records. So we enter the ID 102 and search it. The ID values goes to the hash table and determines the ID buckets and searches the ID value in the bucket. If the value is found out in the slot,reference the value



Of id ID2carries and looks up in the main table.As well as notices the reference id in the ID fields according to the id rows, records will be retrieved from the hash table. In this example 102 value reference id can be found in the second row main table, so that row field value is related to the reference id. So we choose the Name field and gender field in that referenceid carry on the hash table and find the correctvalue using the reference id. Name reference id is also accomplished like ID1.In ID2 Name is set as N2 and gender reference id is set as G1.These referencesare passedto the hash table bucket. The suitable buckets are found to retrieve the values.The reference id value similar Name value is Krishnan and Gender value is Male.

VIII. BENEFITS

This approach may not be efficient for small size of data. When the size of the data is large, it will become more efficient, in the case of small set of tables the fields contain less size of memory. So using this techniques avoiding the duplication in the columngives only less improvement when the database table contain small set of fields such as name, age, data of birth, college, address in particularly city, state, country and many other fields which are get duplicated in the column level. When larger memory is required in database to store up the multimedia files such as video, audio, image and if duplication occurs in the multimedia files that occupies large amount of memory. So it causes low efficient in the storage. De-duplication technique leads to avoid storage loss even in multimedia files.

Example,

For instance I store a video file, the video file is stored as binary codes. Then another person wants to store the same video file but in different name. But the content of the file is same. If this video file is stored, duplication occurs then memory is going to be wasted. If we use the de-duplication technique to check the binary codes of the video file we can identify the content of the video file is existing or not.

If there is same video file, we need not to store the full video file. Just pass the reference from reference table and store the reference of the video file. Likewise we can save more memory by using de-duplication technique.

IX. CONCLUSION

De-duplication technique is to be practised to get more efficient in data storage and to save more memory size. It will be useful when large files are managed in World Wide Web. De-duplication is very simple and more efficient technique to improve the data storage.

REFERENCES

1. Peter Christen “A Survey of Indexing Techniques for Scalable Record Linkage and De-duplication” Vol. 24, No. 9, September 2012
2. Ho Min Jung_ Sang Yong Park and Young WoongKo“Data Deduplication System Considering File Pattern”Ajumobi Udechukwu1, Christie Ezeife2, Ken Barker3 “Independent de-duplication in data cleaning”
3. Aron Culotta, Andrew McCallum “Joint Deduplication of Multiple Record Types in Relational Data”