CD-001

New data structures to reduce data size and search time

Tsuneo Kuwabara†

1. Introduction

It is an important problem for databases to reduce search time. Indexing methods have been commonly used to reduce search time. However, indexing methods do not reduce the data size. Here, I propose data structures and searching methods to reduce both search time and data size[1],[2],[3]. The proposed methods maintain data normalizations and integrity. The proposed methods are also independent from indexing methods, so the two methods can be used simultaneously.

In this paper, the principles of the proposed data structures are described in Section 2. Some applicable templates and the simulation results of reduction rates are shown in Sections 3 and 4, respectively. The updating methods, which are also important processes, are shown in Section 5[3], [4]. Section 6 gives the conclusions.

2. Principles

The principles of the proposed data structures are shown in Fig. 1. Table A in Fig. 1 is an example of a conventional data structure. Here, the values of Items A and B have multiple relations with each other.

Tables B, C, and D are examples of the proposed data structures. Multiple values of Item A and Item B that are related to each other in Table A are assigned to the same group in Tables B and C. The remaining relations between Item A and Item B, those that cannot be stored in Table B nor Table C, are recorded in Table D.

In the example shown in Fig. 1, the total size of the data is reduced by about 25%, relative to the size with conventional data structures, by using the proposed data structures. The reduction of total data size effects a reduction in search time.

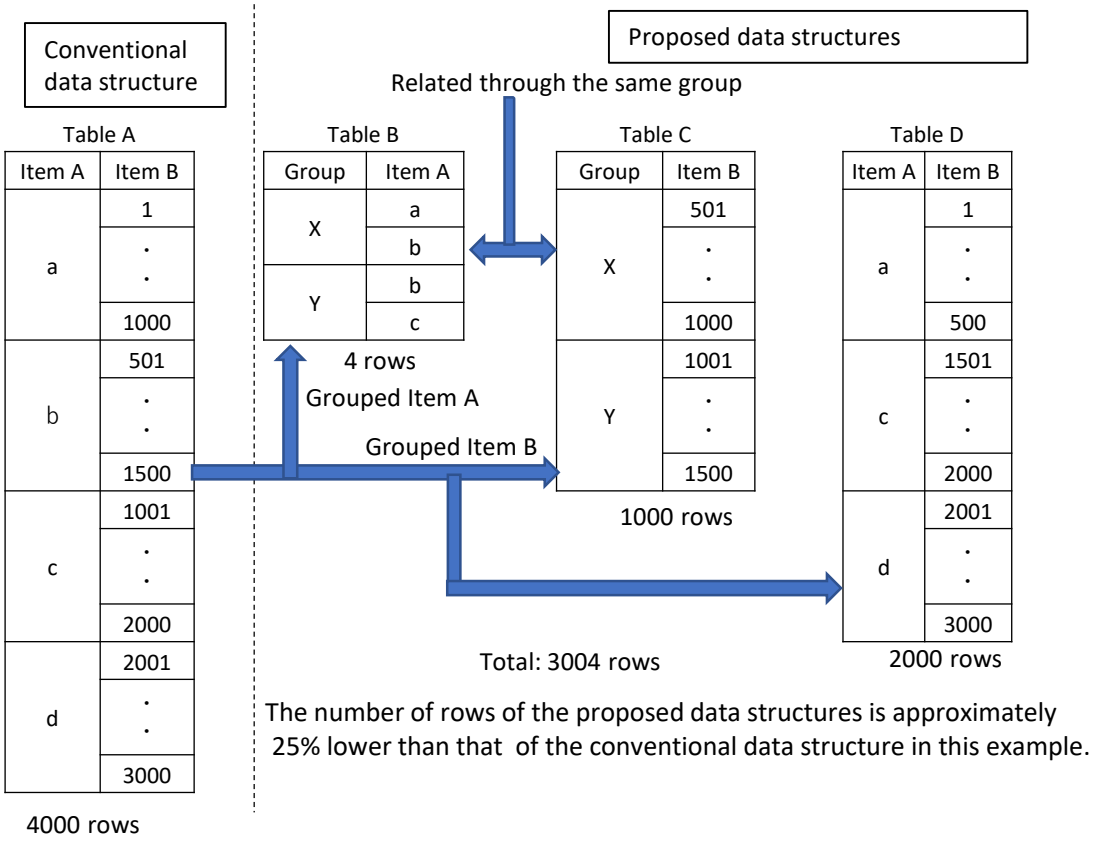

Fig. 1 Principles of the proposed data structures

†Department of Information Sciences, Faculty of Science, Kanagawa University, Hiratsuka-shi, Japan

Moreover, in some cases, some tables can be omitted from the search. For example, only Table D needs to be searched when the values of Item B related to value d of Item A are to be checked, because the value d of item A is not related to any group in Table B.

3. Applicable Template

Fig. 2 shows another template for a conventional data structure. Table F corresponds to Table A in Fig. 1. Attributes of Item A are recorded in Table E, and those of Item B in Table G. As a specific example, if Item A were merchandise, then Item B might be purchasers of the merchandise.

The attributes of Item B related to some item A, whose attributes have certain values, can be found from the data structures in Fig. 2 by appropriately using subqueries if necessary.

Considering the same data structures as in Fig. 2, the number of rows in Table F is much higher than the numbers in Table E and Table G. Because of this, the time to search Table F may be much longer than the times to search Tables E and G.

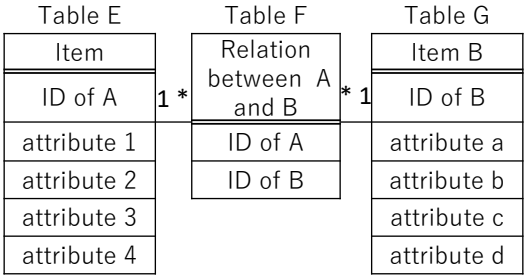

Fig. 2 Template for conventional data structures

Fig. 3 shows the proposed data structures corresponding to the conventional data structures shown in Fig. 2. Table F in Fig. 2 is replaced with Table H, Table I, and Table J in Fig. 3. Values of Item A and Item B related each other in Table F are assigned to the same group in Tables H and I. The relations between Item A and Item B that cannot be recorded in Table H or Table I are recorded in Table J.

As mentioned in Section 2, the expected collective size of Table H, Table I, and Table J is smaller than that of Table F. As a result, the search time using the data structures in Fig. 3 is expected to be less than that using the data structures in Fig. 2.

4. Simulations of Time and Size Reduction

Experimental results of time and size reduction by using the proposed methods have been previously reported, in reference [2]. In this paper, the theoretical reduction rate of data size, D, and the reduction rate of expected search time, T, are introduced. These rates are based on the data structures shown in Fig. 2 and Fig. 3, calculated by comparing the size of Table H, Table I, and Table J collectively with the size of Table F. Moreover, it is assumed that the number of records in Table H is equal to that in Table I. This is the worst-case assumption for the proposed method in terms of search time.

D and T are given by Eq. (1) and Eq. (2), respectively.

$$\mathrm{D} = 1 - \frac{\sum_{i=1}^{n}(M_i+N_i)+S}{\sum_{i=1}^{n} M_i \cdot N_i + S} = \frac{1-X}{1+\mathrm{Y}} \qquad (1)$$

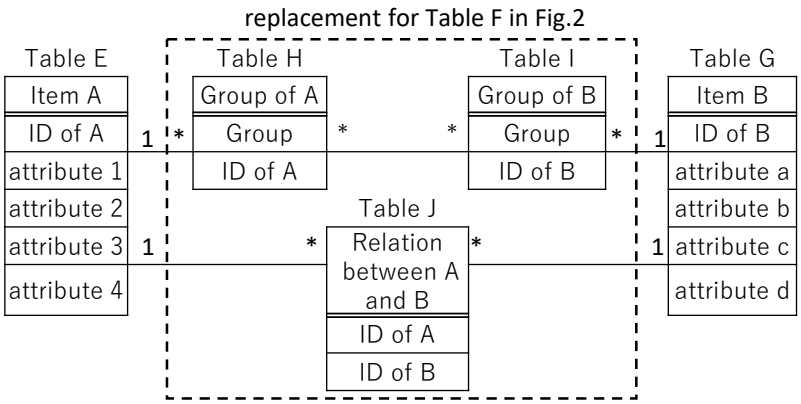

Fig. 3 Template for proposed data structures

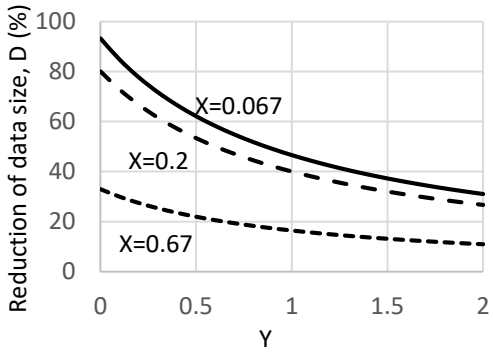

Fig. 4 Simulated reduction of data size

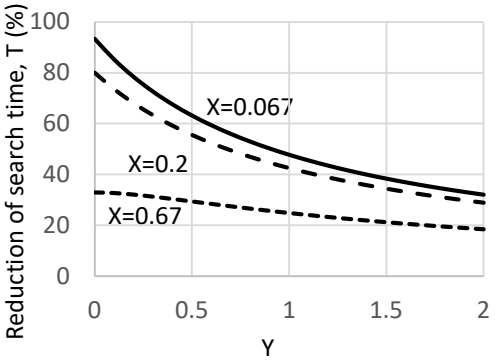

Fig. 5 Simulated reduction of search time

$$\mathrm{T} = \mathrm{D} \cdot \frac{\sum_{i=1}^{n}(M_i \cdot N_i)}{\sum_{i=1}^{n} M_i \cdot N_i + S} + \left(1 - \frac{\sum_{i=1}^{n} M_i + S}{\sum_{i=1}^{n} M_i \cdot N_i + S}\right) \cdot \frac{S}{\sum_{i=1}^{n} M_i \cdot N_i + S} = \frac{1 - X + Y - XY/2}{(1+Y)^2} \quad (2)$$

where,

$$X = \frac{\sum_{i=1}^{n}(M_i + N_i)}{\sum_{i=1}^{n} M_i \cdot N_i}$$

$$\mathrm{Y} = \frac{S}{\sum_{i=1}^{n} M_i \cdot N_i}.$$

with the following symbols.
i: group number
n: the number of groups
M_i: the number of records of group *i* in Table H
N_i: the number of records of group *i* in Table I
S: the number of records in Table J

Here, *X* is the rate of compression achieved by refactoring Table F to Tables H and J. *Y* is the ratio of uncompressed data to compressed data in the proposed data structures.

D and *T* are numerically calculated by Eq. (1) and Eq. (2), respectively, under the following conditions.

(1) $M_i = N_i = 30$ for all *i*.
In this case, $X \approx 0.067$.
(2) $M_i = N_i = 10$ for all *i*. In this case, X = 0.2.
(3) $M_i = N_i = 3$ for all *i*. In this case, X ≈ 0.67.

The calculation results for *D* and *T* are shown in Fig. 4 and Fig. 5, respectively.

Fig. 4 and Fig. 5 show that the proposed methods can effectively reduce total data s

size and search times. In case (1) with *Y* = 0, both *D* and *T* are approximately 93%. Even in case (3) with *Y* = 2, *D* is approximately 11% and *T* is approximately 18%.

5. Data-updating Algorithm

In this section, the algorithm to update data for the proposed data structures is described. Here, the data structures in Fig. 3 are assumed to have the basic structure shown, and newly inputted data are only relations between the ID of item A and the ID of item B. Though Table H and Table I are logically equivalent in Fig 3., they cannot be treated equally in practical applications. For example, if Item A represents a piece of merchandise and Item B is the purchaser, the data size of Item B in Table I may be much larger than that of Item A in Table H. Here, it is assumed that the data in Table I is bigger than that in Table H. If the relations between Item A and Item B are supplemented then, new data may need to be added to Table H or Table I.

However, it is possible to handle new data by updating only one of Table H and Table I. The algorithm described here handles new data, adding records to Table I only.

Dividing Table J into two tables, Table J-1 and Table J-2, allows efficient updates. Table J-1 contains the relations between the IDs of Item A and Item B, which are not related in any groups of Table H and Table I. In contrast, Table J-2 contains the IDs of Items A and B, which are related through being in the same group in Table H or Table I. By dividing Table J into Tables J-1 and J-2, it becomes easy to examine whether a newly inputted record forms a new relation between a group and

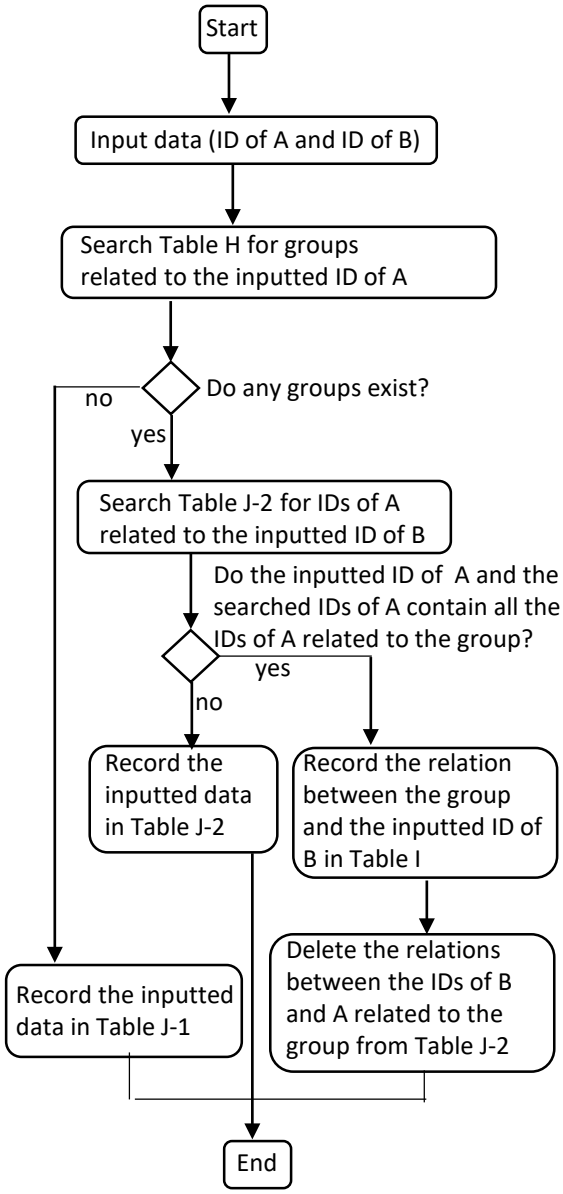

Fig. 6 Algorithm to update data with the proposed method

the inputted ID of Item B by searching only Table J-2.

Fig. 6 shows the update algorithm for the proposed method. In this algorithm, Table H is searched first for the inputted ID of Item A, based on the assumption that the size of Table H is smaller than that of Table I. If there is no group related to the inputted ID of Item A, then the inputted data are recorded on Table J-1. When there exist some groups related to the inputted ID of Item A, then Table J-2 is searched for the IDs of Item A related to the inputted ID of Item B.

If all the IDs of Item A related to the group without the inputted ID of A are searched, then the relation between the group and the inputted ID of Item B are added in Table I, and the relations between the inputted ID of Item B and all the IDs of Item A related to the group newly recorded on Table I are deleted from Table J-2. In this case, the size of Table J-2 and the total data size are reduced.

If some IDs of Item A related to the group without the newly inputted ID of item A do not exist, then the inputted data are recorded in Table J-2.

As described above, Table J-2 contains the IDs of items A and B only as related to the groups. Moreover, some records of Table J-2 are sometimes deleted. As a result, the data size of Table J-2, which will be searched in this algorithm, is expected to remain relatively small even after newly inputting data.

It takes more time for the updating than on the conventional methods, mainly in the process of deleting data from Table J-2. One of the relaxation methods of this problem is to proceed the updating or only the deleting for multiply inputs at one time[3],[4], conveniently in idle times for search task.

6. Conclusion

New data structures and a method of searching databases are proposed. In the proposed method, the data size and search times can be reduced. As well, data normalization and integrity can be perfectly maintained. The proposed methods are independent from indexing methods, so both methods can be used simultaneously.

References

[1] T. Kuwabara : JapanesePatentNo.6269884, Kanagawa University(patentee) (2017.5.19 application, 2018.1.12 registration)
[2]T. Kuwabara : New Data Structures to Reduce Searching Time on Databases: IEICE General Conference 2018,D-4-7,p28(2018).
[3]T. Kuwabara : PCT application JP2018/018419, Kanagawa University (applicant), (2018.5)
[4] T. Kuwabara : Japanese Patent application No.2018-090308, Kanagawa University (applicant), (2018.5)