

[Name of the Document] Specification

[Title of the Invention]

INFORMATION PROCESSING SYSTEM AND
INFORMATION PROCESSING METHOD

5

[Technical Field]

[0001]

The disclosure discussed herein relates to
an information processing system and an information
10 processing method.

[Background Art]

[0002]

Use of indexes such as B-tree and Hash
15 indexes are known techniques to search for data
stored in a database and display the search results
(e.g., Patent Documents 1 and 2).

[RELATED-ART DOCUMENTS]

20 [NON-PATENT DOCUMENTS]

[0003]

[Patent Document 1] JP-A-2007-25735

[Patent Document 2] JP-A-6-335165

25

[Summary of the Invention]

[Problem to be Solved by the Invention]

[0004]

Recent development of various sensor
30 devices and observation devices have made it
possible to obtain a large amount of data
representing sensing results and observation results.
This has increased a demand for searching for

desired data from such a large amount of data (so-called big data) to display the search results.

[0005]

5 However, to transfer or aggregate big data
is difficult due to its data size. Further, in
order to search for desired data from big data,
search servers to be used need to have high-
performance hardware resources, which will result in
high installation cost of such search servers. The
10 installation of the search servers provided with
high-performance hardware resources will also result
in high maintenance cost (e.g., maintenance cost,
power consumption, etc.).

[0006]

15 Providers of big data are required to
provide big data as files, using simple file servers
with inexpensive hardware/software relating to
search processing. Use of such simple file servers
will simplify the maintenance of the servers and
20 lower the maintenance cost.

[0007]

 Users of big data desire to flexibly
combine search files of the big data providers to
display the search results, using general-purpose
25 PCs (personal computers) or the like.

[0008]

 The present invention has been made in
light of the above-mentioned points, and an object
of the present invention is to provide an
30 information processing system and an information
processing method that enables users of big data to
flexibly combine a plurality of big data files
distributed over a network, such as the Internet, to

easily find and display desired data.

[Means to Solve the Problem]

[0009]

In order to solve the above-described
5 problem, an information processing system according
to an embodiment of the present invention includes:
a storage storing a table including a
plurality of records each including first array data
having an item value that is predetermined; second
10 array data having a record number for identifying
each of the records; and third array data relating
the first array data with the second array data; and
an acquisition unit configured to acquire,
in response to specification of the item value or
15 the record number, one or more records corresponding
to the specified item value or record number from
among the plurality of records stored in the table,
using the first array data, the second array data,
and the third array data.

20

[Advantageous Effect of the Invention]

[0010]

According to the disclosed techniques,
users of big data are enabled to flexibly combine
25 multiple big data files distributed over a network,
such as the Internet, to easily find and display
desired data.

[Brief Description of the Drawings]

30 **[0011]**

FIG. 1 is a diagram illustrating an
overall configuration example of a data management
system according to an embodiment of the present

invention;

FIG. 2 is a diagram illustrating a hardware configuration example of a computer;

FIG. 3 illustrates examples of tables;

5 FIG. 4 is a diagram illustrating examples of a record order view;

FIG. 5 is a diagram illustrating examples of a value order view;

10 FIG. 6 is a diagram illustrating an example of a record order UNION view;

FIG. 7 is a diagram illustrating an example of a value order UNION view;

FIG. 8 is a diagram (part 1) illustrating an example of an internal data structure;

15 FIG. 9 is a diagram (part 2) illustrating an example of an internal data structure;

FIG. 10 is a diagram illustrating an example of a method of generating SVL, IND, and INV;

20 FIG. 11 is a diagram illustrating an application example 1; and

FIG. 12 is a diagram for illustrating an application example 2.

[Mode for Carrying Out the Invention]

25 **[0012]**

The following illustrates embodiments of the present invention (hereinafter also referred to as "the present embodiment"). In the following, a description is given of a data management system 1
30 that displays and searches for data stored in a database.

[0013]

<OVERALL CONFIGURATION>

First, an overall configuration of a data management system 1 according to a present embodiment will be described with reference to FIG. 1. FIG. 1 is a diagram illustrating an overall configuration example of a data management system according to present embodiment.

[0014]

As illustrated in FIG. 1, the data management system 1 according to the present embodiment includes one or more terminal devices 10 and one or more data management devices 20. The terminal device 10 and the data management device 20 are communicably connected via a network N, such as the Internet, a LAN (Local Area Network), or a telephone network.

[0015]

The terminal device 10 is an information processor (computer) such as a personal computer (PC) operated by a user who is a user of big data. A user may operate the terminal device 10 to display and search for data managed by the data management device 20.

[0016]

The terminal device 10 includes a display processor 100 for displaying and searching for data managed by the data management device 20. The display processor 100 is configured to function according to one or more programs installed in the terminal device 10, which are executed by a CPU (Central Processing Unit), for example.

[0017]

The display processor 100 generates and displays views of data managed by the data

management device 20 according to a user's operation. The display processor 100 also searches for desired data from a certain view, and displays the search result. A view is display information representing data that are voluntarily selected from one or more tables (or other views). A table (or list) is information including one or more data. Data search indicates specification of a range of values with respect to a certain item (e.g., =, <, >, between, etc.) of data forming a certain view to extract desired data from the certain view. The resulting data retrieved from the certain view also forms a view. Hence, a record order view, a value order view, a record order UNION view, and a value order UNION view described later may also be referred to as a view generated by all the records constituting one or more tables 1000. However, the search result of the data search may also be referred to as a view of some of displayed records of these views.

20 **[0018]**

The terminal device 10 is not necessarily a PC, and may be, for example, various information processing devices such as a smartphone and a tablet terminal.

25 **[0019]**

The data management device 20 is an information processor (computer) such as a database server or the like used by a big data provider to manage various data. The data management device 20 includes a data processor 200 configured to manage various data, and a database 300 storing various data. The data processor 200 is configured to function according to one or more programs installed

in the data management device 20, which are executed by a CPU (Central Processing Unit), for example. The database 300 may be provided by, for example, an auxiliary storage or the like.

5 **[0020]**

The data processor 200 transmits corresponding data from among the data stored in the database 300 to the terminal device 10 in response to a request from the terminal device 10. As a result, the terminal device 10 is enabled to generate and display a view, and also display a search result.

[0021]

The database 300 includes a table 1000 composed of one or more data, and SVL (Sorted Value List) 1100, IND (Indirect Record Index) 1200 and INV (Inverted Record Index) 1300 that are used for view generation and data search. The SVL 1100, the IND 1200, and the INV 1300 are internal data generated for a certain item of each of the records constituting the table 1000. Details of these data will be described later.

[0022]

The views displayed by the display processor 100 of the terminal device 10 according to the present embodiment include a record order view, a value order view, a record order UNION view, and a value order UNION view. The record order view is a view that displays respective data (this data is also referred to as a "record") constituting the table 1000 in the order of record number. The value order view is a view that displays respective records constituting the table 1000 in the value

order of a specified item (this item is also referred to as a "column") of these records. The record order UNION view is a view that displays respective records constituting multiple tables 1000
5 in a record order. The value order UNION view is a view that displays respective records constituting multiple tables 1000 in the value order of a certain item of these records. Details of these views will be described later. Note that generating one view
10 from multiple tables 1000 (or views) is also referred to as "UNION" or "Virtual UNION".

[0023]

In this case, the record order UNION view and the value order UNION view may be generated from
15 the records constituting the tables 1000 managed by the same data management device 20, or may be generated from records constituting the tables 1000 separately managed by the different data management devices 20. Hereinafter, to distinguish the
20 different data management devices 20, one of the data management devices 20 is referred to as a data management device 20-1, and the other is referred to as a data management device 20-2, and the like. Similarly, to distinguish the different tables 1000,
25 one of the tables 1000 is referred to as a table 1000-1, and the other is referred to as a table 1000-2, and the like. The same applies to distinguish each of the different SVLs 1100, each of the different INDs 1200, and each of the different
30 INVs 1300.

[0024]

The data management system 1 according to the present embodiment is enabled to generate and

display various views easily and quickly, even when the number of records constituting tables 1000 is huge, by using the SVL 1100, the IND 1200, and the INV 1300 for generating and displaying various views.

5 Further, the data management system 1 according to the present embodiment is enabled to perform a search at high speed and retain search results with a small data capacity, by using the SVL 1100, the IND 1200, and the INV 1300 for performing data
10 search.

[0025]

Note that FIG. 1 illustrates an example in which the terminal device 10 and the data management device 20 are different devices; however, the
15 present embodiment may be applied in a similar fashion even when these devices are integrally formed. Further, the data management device 20 may be provided by, for example, a NAS (Network Attached Storage) or the like.

20 **[0026]**

<HARDWARE CONFIGURATION>

Next, a hardware configuration of one or more computers for providing the terminal device 10 and the data management device 20 according to the
25 present embodiment will be described with reference to FIG. 2. FIG. 2 is a diagram illustrating an example of a hardware configuration of a computer 500.

[0027]

30 The computer 500 illustrated in FIG. 2 includes an input device 501, a display device 502, an external I/F 503, a random access memory (RAM) 504, a read only memory (ROM) 505, a CPU 506, a

communication I/F 507, and an auxiliary storage 508. These hardware components are communicably connected via the bus B.

[0028]

5 The input device 501 is a keyboard, a mouse, a touch panel, or the like, which is used by a user to perform various input operations. The display device 502 is a display, or the like, which is configured to display various processing results
10 (e.g., various views). Note that the data management device 20 may not necessarily include at least one of the input device 501 and the display device 502.

[0029]

15 The external I/F 503 is an interface with an external device. The external device includes a recording medium 503a, and the like. The terminal device 10 and the data management device 20 may read and write the recording medium 503a, and the like
20 via the external I/F 503.

[0030]

 Examples of the recording medium 503a include a flexible disk, a CD (Compact Disc), a DVD (Digital Versatile Disk), an SD memory card (Secure
25 Digital memory card), and a USB (Universal Serial Bus) memory card.

[0031]

 The communication I/F 507 is an interface for connecting the computer 500 to the network N.
30 The terminal device 10 and the data management device 20 may perform data communication with another device via the communication I/F 507.

[0032]

The RAM 504 is a volatile semiconductor memory that temporarily retains programs and data. The ROM 505 is a non-volatile semiconductor memory capable of retaining programs and data even when the power is turned off. The ROM 505 stores, for example, OS (Operating System) settings, network settings, and the like.

[0033]

The CPU 506 is an arithmetic-logic device that loads programs and data from the ROM 505, the auxiliary storage 508, and the like onto the RAM 504 and executes processes.

[0034]

The auxiliary storage 508 is, for example, a hard disk drive (HDD) or a solid state drive (SSD), which is a non-volatile storage storing programs and data. The programs and data stored in the auxiliary storage 508 include, for example, an OS, application software for providing various functions on the OS, and the like. The auxiliary storage 508 of the computer 500 configured to serve as the terminal device 10 stores one or more programs for providing a function of the display processor 100 described above. Likewise, the auxiliary storage 508 of the computer 500 configured to serve as the data management device 20 stores one or more programs for providing a function of the data processor 200 described above.

[0035]

The terminal device 10 and the data management device 20 according to the present embodiment having the hardware configuration of the computer 500 illustrated in FIG. 2 are enabled to

perform various processes according to the present embodiment.

[0036]

<TABLES>

5 The following illustrates examples of displaying respective views with reference to a table 1000-1 illustrated in FIG. 3(a) and a table 1000-2 illustrated in FIG. 3(b), according to an embodiment. Hereinafter, a table number of the table 1000-1 illustrated in FIG. 3(a) is "0", and a table number of the table 1000-2 illustrated in FIG. 3(b) is "1". Note that the table 1000 is an electronic file or data composed of one or more data (records). Note also that the table number is
10 identification information for identifying at least a table 1000 used for generating and displaying a view.
15

[0037]

 The table 1000-1 illustrated in FIG. 3(a)
20 is stored, for example, in a database 300 of the data management device 20-1, and the table 1000-2 illustrated in FIG. 3(b) is stored, for example, in a database 300 of the data management device 20-1. However, the table 1000-1 illustrated in FIG. 3(a)
25 and the table 1000-2 illustrated in FIG. 3(b) may be stored in the database 300 of the same data management device 20.

[0038]

 The table 1000-1 illustrated in FIG. 3(a)
30 includes four records, and each record includes an item "Name" and an item "Age". The table 1000-1 illustrated in FIG. 3(b) includes four records, and each record includes an item "Name" and an item

"Age". A position is a number (e.g., location (an address) of a storage area in which each record is stored, etc.) indicating a relative position from the top record in each table 1000.

5 **[0039]**

Tables 1000-1 and 1000-2 respectively illustrated in FIGS. 3(a) and 3(b) are each composed of four records; however, these tables are merely examples. Each table 1000 may include any desired number of records. A typical number of records forming each table 1000, which is handled as big data, may be tens of thousands to hundreds of millions or more. Moreover, each of the records forming the table 1000-1 and the table 1000-2 illustrated in FIGS. 3(a) and 3(b) has two items; however, this is merely an example, and each record may also have any desired number of items.

[0040]

<DIFFERENT VIEWS>

20 Next, an illustration is given of respective views in a case of using the table 1000-1 and the table 1000-2 illustrated in FIGS. 3(a) and 3(b).

[0041]

25 «RECORD ORDER VIEWS»

FIG. 4(a) illustrates a record order view VW1-1 of the table 1000-1 illustrated in FIG. 3(a). FIG. 4(b) illustrates a record order view VW1-1 of the table 1000-1 illustrated in FIG. 3(b).

30 **[0042]**

As illustrated, the record order view is a view that displays records forming the table 1000 in the record order. Note that the record order view

may be a view that displays, for example, the records forming the table 1000 in reverse order.

[0043]

5 Note that each record displayed in the record order view VW1 illustrated in FIG. 4 includes all the items of the record constituting the table 1000; however, the present invention is not limited to this example. The items of each record displayed in the record order view may be some of the items of
10 the record constituting the table 1000. The same applies to other views such as value order view, record order UNION view, and value order UNION view.

[0044]

«VALUE ORDER VIEWS»

15 FIG. 5(a) is a value order view VW2-1 with respect to the item "Name" of the table 1000-1 illustrated in FIG. 3(a). Further, FIG. 5(b) is a value order view VW2-2 with respect to the item "Name" of the table 1000-2 illustrated in FIG. 3(b).

20 **[0045]**

As illustrated in FIGS. 5(a) and 5(b), the item "RecNo" is added to each record of the value order views VW2-1 and VW2-2. The value of the item "RecNo" is a position of a record in the table 1000.
25 In the value order view, the item "RecNo" is added to each record as illustrated. As a result, the position of each record in the table 1000, which acts as the basis of the value order view, is also managed in the value order view. Thus, when a user
30 of the terminal device 10, for example, selects a desired record from the value order view, the display processor 100 is enabled to acquire a corresponding record from the table 1000, which acts

as the basis of the value order view, based on the value of the item "RecNo" of the selected record.

[0046]

For example, "1" is added as a value of the item "RecNo" to the record at position "0" in the value order view VW 2-1 illustrated in FIG. 5(a). This indicates that the record at position "0" in the value order view VW2-1 illustrated in FIG. 5(a) corresponds to a record at position "1" in the table 1000-1 illustrated in FIG. 3(a). The same applies to values of the item "RecNo" of other records.

[0047]

In the value order view VW2-1 and the value order view VW2-2, values of the item "Name" are displayed in alphabetical order (in ascending order of the alphabet).

[0048]

As described above, the value order view displays values in the value order with respect to a specified item (e.g., ascending order (or descending order) or alphabetical order) of the records constituting the table 1000. Note that FIG. 5(a) and FIG. 5(b) illustrate the value order view with respect to the item "Name"; however, the value order view may be displayed with respect to the item "Age" in a similar fashion.

[0049]

«RECORD ORDER UNION VIEW»

FIG. 6 illustrates a record order UNION view VW3 composed of records constituting the table 1000-1 illustrated in FIG. 3(a) and records constituting the table 1000-2 illustrated in FIG. 3(b).

[0050]

As illustrated in FIG. 6, the record order UNION view VW3 displays respective records constituting the table 1000-2 following the records constituting the table 1000-1. In the record order UNION view VW3, the position of each record constituting the table 1000-2 is replaced. Specifically, when the position of the record is m and the number of records constituting table 1000-1 is N , the position of the record is replaced with " $m + N$ " for each of the records constituting table 1000-2 in the record order UNION view VW 3. Following this method, positions "0" to "3" of the records of the table 1000-2 illustrated in FIG. 3(b) are replaced with positions "4" to "7" in the record order UNION view VW3 illustrated in FIG 6.

[0051]

In the record order UNION view VW3 illustrated in FIG. 6, the records constituting the table 1000-2 with the table number "1" are displayed following the records constituting the table 1000-1 with the table number "0". However, this order of displaying the tables may be reversed.

[0052]

As described above, the record order UNION view displays the records constituting the respective tables 1000 in the table number order and in the record order. Note that FIG. 6 illustrates an example of the record order UNION view that is composed of two tables 1000; however, the record order UNION view may be composed of three or more tables 1000 in a similar fashion.

[0053]

«VALUE ORDER UNION VIEW»

FIG. 7 is a value order UNION view VW4 with respect to the item "Name" of records constituting the table 1000-1 illustrated in FIG. 3(a) and records constituting the table 1000-2 illustrated in FIG. 3(b).

[0054]

As illustrated in FIG. 7, an item "RecNo" is added to each record of the value order UNION view VW4. The item "RecNo" is already described above; however, the position of each record constituting the table 1000-2 is replaced in FIG. 7. Specifically, when the position of the record is m and the number of records constituting table 1000-1 is N , the position of the record is replaced with " $m + N$ " for each of the records constituting table 1000-2 in the record order UNION view VW 3.

[0055]

Thus, in response to selection of a desired record from the value order UNION view by a user of the terminal device 10, the display processor 100 is enabled to acquire a corresponding record from the table 1000 based on a value of the item "RecNo" of the selected record. Note that the table 1000 acts as the basis of the value order UNION view. Details of this method will be described later.

[0056]

Further, in the value order UNION view VW4, values of the item "Name" are displayed in alphabetical order (ascending order of the alphabet). More specifically, in the value order UNION view VW 4, from highest order of priority values of the item

"Name", the table number, and the position of the table 1000 are displayed.

[0057]

As described above, the value order UNION
5 view displays values in the value order with respect
to a specified item (e.g., ascending order (or
descending order) or alphabetical order) of the
records constituting a plurality of tables 1000.
Note that FIG. 7 illustrates the value order UNION
10 view with respect to the item "Name"; however, the
value order UNION view may be displayed with respect
to the item "Age" in a similar fashion.

[0058]

<INTERNAL DATA STRUCTURE>

15 The following illustrates an internal data
structure of the database 300 of the data management
device 20. As described above, the database 300
stores the SVL 1100, the IND 1200, and the INV 1300
as internal data generated from the table 1000.
20 Hereinafter, the SVL 1100 the IND 1200, and the INV
1300 generated from each of the table 1000-1 and the
table 1000-2 illustrated in FIG. 3(a) and FIG. 3(b)
will be described.

[0059]

25 FIG. 8(a) depicts the table 1000-1 with
the table number "0", and the SVL 1100-1, IND 1200-1,
and INV 1300-1 generated with respect to the item
"Name" of each of the records constituting the table
1000-1.

30 **[0060]**

Likewise, FIG. 8(b) depicts the table
1000-2 with the table number "1", and the SVL 1100-2,
IND 1200-2, and INV 1300-2 generated with respect to

the item "Name" of each of the records constituting the table 1000-2. Note that when replacement of a record position occurs (i.e., use "UNION" to be unified with another table 1000), a position of the
5 INV 1300-2 is also replaced. That is, when positions "0" to "3" of table 1000-2 are replaced with positions "4" to "7" respectively, record numbers "3", "1", "0" and "2" stored at positions "0" to "3" of INV 1300-2 are also replaced with "7",
10 "5", "4" and "6" as illustrated in FIG. 8(b).

[0061]

The SVL 1100 illustrated in FIGS. 8 (a) and 8 (b) is obtained by making the value of the item "Name" of each of the records constituting the
15 table 1000 unique, then sorting the records in ascending order (in ascending order of the alphabet), and finally adding a value " ∞ " indicating the maximum value.

[0062]

20 The IND 1200 illustrated in FIGS. 8(a) and 8(b) stores a start position of the INV 1300 corresponding to each element of the SVL 1100. For an element of SVL 1100, positions of INV 1300 corresponding to the element of SVL 1100 indicate a
25 range from a start position corresponding to the element to a position immediately preceding the start position corresponding to a next element of SVL 1100. For example, for "Cathy" in SVL 1100-1, an element corresponding to "Cathy" in IND 1200-1 is
30 "1". For a next element "Emily" after "Cathy" in SVL1100-1, an element corresponding to "Emily" in IND 1200 is "3". Thus, for "Cathy" in SVL 1100, a position of a corresponding element of INV 1300-1

indicates from "1" to "2".

[0063]

Each element of IND 1200 indicates the total number of occurrences for listings of a lower (alphabetic) value than the corresponding element in SVL 1100. For example, an element of IND 1200-1 corresponding to "Emily" illustrated in FIG. 8(a) is "3". This indicates that in the table 1000-1, the number of records having a value of the item "Name" smaller than the value of "Emily" is "3".

[0064]

INV 1300 illustrated in FIGS. 8(a) and 8(b) is of transposed record numbers. For example, in the INV 1300-1 illustrated in FIG. 8(a), an element corresponding to "Emily" of the SVL 1100-1 is "2". This "2" indicates that, in the table 1000-1, a record having a value of the item "Name" being "Emily" is stored at a position "2". Likewise, in the INV 1300-1 illustrated in FIG. 8(a), elements corresponding to "Cathy" of the SVL 1100-1 are "0" and "3". These "0" and "3" indicate that, in the table 1000-1, a record having a value of the item "Name" being "Cathy" is stored at positions "0" and "3".

[0065]

As illustrated in FIG. 8, the SVL 1100, the IND 1200, and the INV 1300 are provided by array data, a list, vector data, and the like.

[0066]

Note that FIG. 8 illustrates the SVL 1100, IND 1200 and INV 1300 generated with respect to the item "Name", and FIG. 9 illustrates, as another example, SVL 1100, IND 1200 and INV 1300 generated

with respect to the item "Age".

[0067]

FIG. 9(a) depicts the table 1000-1 with the table number "0", and the SVL 1100-1, IND 1200-1, and INV 1300-1 generated with respect to the item "Age" of each of the records constituting the table 1000-1. Likewise, FIG. 9(b) depicts the table 1000-2 with the table number "1", and the SVL 1100-2, IND 1200-2, and INV 1300-2 generated with respect to the item "Age" of each of the records constituting the table 1000-2. Note that, when replacement of a record position occurs in FIG. 9(b), as described above, a position of the INV 1300-2 is also replaced. That is, when positions "0" to "3" of table 1000-2 are replaced with positions "4" to "7" respectively, positions "0" to "3" of INV 1300-2 are also replaced with "5", "7", "4" and "6" as illustrated in FIG. 9(b).

[0068]

In FIG. 9, values of elements of SVL 1100, IND 1200 and INV 1300 differ from those of FIG. 8; however, the meanings represented by these data are the same.

[0069]

<METHOD OF GENERATING SVL 1100, IND 1200, AND INV 1300>

The following illustrates, with reference to FIG. 10, a case where SVL 1100-1, IND 1200-1, and INV 1300-1 illustrated in FIG. 8(a) are generated from the table 1000-1 illustrated in FIG. 3(a). Each process of S1 to S5 in FIG. 10 is performed by the data processor 200.

[0070]

S1. Add RecNo: The data processor 200 extracts each element of the item "Name" from the table 1000-1, and adds to each of the extracted elements an item "RecNo" having a record number as a value. Note that the record number may be the same number as a position of each element.

[0071]

S2. Sort By Value: The data processor 200 sorts the elements (e.g., sorts in ascending order) by the value of the item "Name".

[0072]

S3. Separate: The data processor 200 separates the item "Name" from the item "RecNo" for each element. Each element with the item "RecNo" after this separation becomes INV 1300.

[0073]

S4. Unique & Count: The data processor 200 creates an item "Count" having a value to be the number of occurrences of each value of the item "Name", and eliminates a duplicated value from the item "Name".

[0074]

S5. Aggregate: The data processor 200 adds an element having a value of "0" at the top of the item "Count", and then converts the number of occurrences into an accumulated number of occurrences. In addition, the data processor 200 adds an element having a value "∞" at the end of the item "Name". The SVL 1100 is obtained from the above-described processes of respective elements of the item "Name", and the IND 1200 is obtained from the above-described processes of the elements of the item "Count".

[0075]

Note that when each element of the item "Count" is $c[i]$ ($i = 1, \dots, I$), to convert the number of occurrences into the accumulated number of occurrences may indicate to set the element of the item "Count" to $c[j]$ (i.e., $c[1] + \dots + c[j]$) in the order from $j=2$ to $j=I$.

[0076]

In FIG. 10, SVL 1100, IND 1200, and INV 1300 are generated with respect to the item "Name" as examples; however, the SVL 1100, IND 1200 and INV 1300 may also be generated with respect to the item "Age" in the same fashion as described above.

[0077]

Note that the above-described processes from S1 to S5 may be performed when a new table 1000 is added to the database 300, or when a new record is added to the table 1000 in which SVL 1100, IND 1200, and INV 1300 are already generated.

[0078]

Further, illustration of the present embodiment is given on the basis that the data processor 200 performs all the processes of S1 to S5; however, one or more of the processes of S1 to S5 may be executed by a device differing from the data management device 20. In particular, after a device differing from the data management device 20 has executed all the processes of S1 to S5, the data management device 20 may store the processing results of this device.

[0079]

<SEARCH METHOD>

Here, when a value order view or a value

order UNION view is obtained, a known bisection method (binary search method) may be used for searching for desired data from the value order view or the value order UNION view. The data search may
5 be performed by a user of the terminal device 10 specifying a search condition (e.g., =, <, >, between, negation of these, and the like). Note that the data search may be performed by the display processor 100 or may be performed by the data
10 processor 200.

[0080]

In this case, desired data may be searched for at high speed from the value order view or the value order UNION view, and search results may be
15 retained with small data capacity by the following two reasons.

[0081]

Reason 1: The top position and the bottom position of a hit area (an area hit by the known
20 bisection method) can be specified by the known bisection method or the like. Accordingly, records other than the record at the top position and the record at the bottom position do not need to be evaluated for the search condition.

25 **[0082]**

Reason 2: When the number of hits (hit records) is huge, the related art technique needs to retain all the records in the hit area as search results; however, according to the present
30 embodiment, it is sufficient to retain only respective records at the top position and at the bottom position of the hit area as search results.

[0083]

The above reason 2 will be described in more detail. For example, to search for records having a value of the item "Name" being "Cathy" to "Dorothy" in the value order UNION view VW4
5 illustrated in FIG. 7, the records corresponding to these values are records at positions "2" to "6". Accordingly, in the present embodiment, it is sufficient to retain respective records only at the top position "2" and the bottom position "6" of the
10 hit area. In this case, the length of the hit area is $6 - 2 + 1 = 5$, and the number of hits is thus 5. In the following, retrieving (obtaining) of all the five records will be described.

[0084]

15 Prior to further illustration, it should be noted that the value order UNION view VW4 illustrated in FIG. 7 is virtual. Hence, in order to extract each record from the value order UNION view VW4 illustrated in FIG. 7, it is necessary to
20 call a routine for extracting the specified rows of the value order UNION view VW4. Since a value of "RecNo" returned from this routine is a replaced one, the table 1000 is identified by determining one of the table 1000-1 and the table 1000-2, based on the
25 value of RecNo, and the value of RecNo resulting from replacement is returned to the value before the replacement. Note that a routine may be used to return the table number and RecNo before replacement, without a process of returning the replaced value of
30 RecNo, identifying the table 1000 using the replaced value of the RecNo, and replacing the replaced value of the RecNo with the value of the RecNo before replacement.

[0085]

(1) RecNo of the first hit record is 0. RecNo = 0 belongs to the table 1000-1 (because RecNo = 0 is less than 4). Offset = 0 of the table 1000-1 is subtracted from RecNo = 0, and RecNo = 0 - 0 = 0 in the table is obtained. As a result, a record at position "0" in the table 1000-1 is obtained.

[0086]

(2) RecNo of the second hit record is 3. RecNo = 3 belongs to the table 1000-1 (because RecNo = 3 is less than 4). Offset = 0 of the table 1000-1 is subtracted from RecNo = 3, and RecNo = 3 - 0 = 3 in the table is obtained. As a result, a record at position "3" in the table 1000-1 is obtained.

[0087]

(3) RecNo of the third hit record is 5. RecNo = 5 belongs to the table 1000-2 (because RecNo = 5 is 4 or more). Offset = 4 of the table 1000-2 is subtracted from RecNo = 5, and RecNo = 5 - 4 = 1 in the table is obtained. As a result, a record at position "1" in the table 1000-2 is obtained.

[0088]

(4) RecNo of the fourth hit record is 4. RecNo = 4 belongs to the table 1000-2 (because RecNo = 4 is 4 or more). Offset = 4 of the table 1000-2 is subtracted from RecNo = 4, and RecNo = 4 - 4 = 0 in the table is obtained. As a result, a record at position "0" in the table 1000-2 is obtained.

[0089]

(5) RecNo of the fifth hit record is 6. RecNo = 6 belongs to the table 1000-2 (because RecNo = 6 is 4 or more). Offset = 4 of the table 1000-2 is subtracted from RecNo = 6, and RecNo = 6 - 4 = 2 in

the table is obtained. As a result, a record at position "2" in the table 1000-2 is obtained.

[0090]

5 Note that as a view representing search results, a view may display the records in the hit area, from among the records displayed in the value order view or the value order UNION view. The view representing the search results is displayed by the display processor 100.

10 **[0091]**

<METHOD OF GENERATING DIFFERENT VIEWS>

15 Next, a method of generating each view will be described. By generating each view, the display processor 100 is enabled to display these views. Note that the display processor 100 may generate and display a view by acquiring data from the data management device 20, or the display processor 100 may display a view by generating a view by acquiring data from the database 300 and
20 transmitting the generated view to the display processor 100. However, in a case of generating a view where the tables 1000 managed by different data management devices 20 are unified together, it is preferable that the display processor 100 generates
25 a view.

[0092]

«METHOD OF GENERATING RECORD ORDER VIEW»

30 For example, to generate a record order view VW1-1 with table number "0", records at positions "0" to "3" of the table 1000-1 illustrated in FIG. 8(a) may be sequentially acquired. Likewise, to generate a record order view VW1-2 of table 1000-2 with table number "1", records at positions "0" to

"3" of the table 1000-2 illustrated in FIG. 8(b) may be sequentially acquired.

[0093]

<METHOD OF GENERATING VALUE ORDER VIEW>

5 For example, to generate a value order view VW2-1 with respect to the item "Name" of the table 1000-1 with the table number "0", records whose positions are values of respective elements at positions "0" to "3" of the INV 1300-1 illustrated in FIG. 8(a) may be acquired from the record order view VW1-1. More specifically, values of respective elements at positions "0" to "3" of INV 1300-1 illustrated in FIG. 8(a) are INV[0] = 1, INV[1] = 0, INV[2] = 3, and INV[3] = 2. Accordingly, records at positions "1", "0", "3", and "2" are sequentially acquired in the record order from the record order view VW1-1 illustrated in FIG. 4(a), and these positions of the acquired records are added to the acquired records as respective values of the item "RecNo". As a result, the value order view VW2-1 illustrated in FIG. 5(a) is obtained.

[0094]

25 Likewise, to generate a value order view VW2-2 with respect to the item "Name" of the table 1000-2 with the table number "1", records whose positions are values of respective elements at positions "0" to "3" of the INV 1300-2 illustrated in FIG. 8(b) may be acquired from the record order view VW1-2. More specifically, values of respective elements at positions "0" to "3" of INV 1300-2 illustrated in FIG. 8(b) are INV[0] = 3, INV[1] = 1, INV[2] = 0, and INV[3] = 2. Accordingly, records at positions "3", "1", "0", and "2" are sequentially

acquired in the record order from the record order view VW1-2 illustrated in FIG. 4(b), and these positions of the acquired records are added to the acquired records as respective values of the item "RecNo". As a result, the value order view VW2-2 illustrated in FIG. 5(b) is obtained.

[0095]

«METHOD OF GENERATING RECORD ORDER UNION VIEW»

For example, to generate a record order UNION view VW3 illustrated in FIG. 6, records from the position "0" to "3" of the table 1000-1 illustrated in FIG. 8(a) may be sequentially acquired, and subsequently, records from the position "0" to "3" in the table 1000-2 illustrated in FIG. 8(b) may be sequentially acquired.

[0096]

However, when positions "4" to "7" after replacement are specified for acquiring records at positions "0" to "3" of the table 1000-2 illustrated in FIG. 8(b), these positions "4" to "7" need to be converted into positions "0" to "3" before replacement.

[0097]

<METHOD OF GENERATING VALUE ORDER UNION VIEW>

For example, to generate a value order UNION view VW4 illustrated in FIG. 7, a value of the item "Name", the table number, and the record number in the table may be acquired in descending order of priority, from the value order views VW2-1 and VW2-2, illustrated in FIGS. 5(a) and 5(b).

[0098]

In this case, respective records are acquired from the value order view VW2-1 illustrated

in FIG. 5(a) and the value order view VW2-2 illustrated in FIG. 5(b) in the following order

[0099]

- 5 (1) Record at position "0" of the value order view VW2-1 illustrated in FIG. 5(a);
- (2) Record at position "0" of the value order view VW2-2 illustrated in FIG. 5(b);
- (3) Record at position "1" of the value order view VW2-1 illustrated in FIG. 5(a);
- 10 (4) Record at position "2" of the value order view VW2-1 illustrated in FIG. 5(a);
- (5) Record at position "1" of the value order view VW2-2 illustrated in FIG. 5(b);
- (6) Record at position "2" of the value order view
- 15 VW2-2 illustrated in FIG. 5(b);
- (7) Record at position "3" of the value order view VW2-2 illustrated in FIG. 5(b); and
- (8) Record at position "3" of the value order view VW2-1 illustrated in FIG. 5(a).

20 In this case, values of the item "RecNo" are replaced in the (2) and (5) to (7) above.

[0100]

As described above, the data management system 1 according to the present embodiment uses

25 the SVL 1100, the IND 1200, and the INV 1300 for generating and displaying various views. As a result, the data management system 1 may generate and display various views easily and quickly even when the number of records constituting the table

30 1000 is huge. Further, the data management system 1 according to the present embodiment uses the SVL 1100, the IND 1200, and the INV 1300 for performing data search. As a result, the data management

system 1 may perform a search at high speed, and may retain search results with a small data capacity.

[0101]

Note that UNION views composed of a plurality of tables 1000 that are unified by using "UNION" have been described in the present embodiment; however, "JOIN" may also be used instead of "UNION". For example, when calculating the number of occurrences of a value of the item in S4 "Unique & Count" above, the number of occurrences may be multiplied by the number of occurrences of a value that matches a value in the table 1000 subject to "JOIN" to generate IND 1200 and INV 1300.

[0102]

<VARIOUS FUNCTIONS>

The following illustrates various functions defined with respect to the table 1000 having the above-described data structure.

[0103]

• Count_LT(table number, item name, value)
This is a function to obtain the number of records with a value smaller than the specified value for the item of the specified item name in a table with the specified table number.

[0104]

For example, when the function Count_LT searches the SVL 1100 for the item of the specified item name, and finds an element that matches the specified value, the function Count_LT may simply obtain IND[i] as a position i of that element. However, when the function Count_LT searches SVL 1100 for the item of the specified item name, and finds no element that matches the specified value,

the function Count_LT may simply obtain, from among the values of the elements of the SVL 1100, IND[i] as a position i of the element having a value greater than the specified value and the smallest value.

5

[0105]

• Count_LE(table number, item name, value)

This is a function to obtain the number of records with a value smaller than or equal to the specified value for the item of the specified item name in a table with the specified table number.

10

[0106]

For example, when the function Count_LE searches the SVL 1100 for the item of the specified item name, and finds an element that matches the specified value, the function Count_LE may simply obtain IND[i+1] as a position i of that element. However, when the function Count_LE searches SVL 1100 for the item of the specified item name, and finds no element that matches the specified value, the function Count_LE may simply obtain, from among the values of the elements of the SVL 1100, IND[i] as a position i of the element having a value greater than the specified value and the smallest value.

15

20

25

[0107]

• Count_EQ(table number, item name, value)

This is a function to obtain the number of records with a value equal to the specified value for the item of the specified item name in a table with the specified table number.

30

[0108]

For example, the function Count_EQ may be

Count_EQ() = Count_LE() - Count_LT() .

[0109]

• Pos(table number, item name, value)

This is a function to obtain a position at which the
5 specified value appears in the SVL 1100 for the item
of the specified item name in a table with the
specified table number.

[0110]

For example, when the function Pos
10 searches the SVL 1100 for the item of the specified
item name, and finds an element that matches the
specified value, the function Pos may simply obtain
a position of that element.

[0111]

15 • Total_Count_LT(item name, value)

This is a function to obtain the sum of records of
the number of records with value smaller than the
specified value of all the tables for the item of
the specified item name.

20 **[0112]**

The function Total_Count_LT may simply
apply a table number to Table[i] to be
 Σ Count_LT(Table[i], item name, value).

[0113]

25 • Total_Count_LE(item name, value)

This is a function to obtain the sum of records of
the number of records with value smaller than or
equal to the specified value of all the tables for
the item of the specified item name.

30 **[0114]**

The function Total_Count_LE may simply
apply a table number to Table[i] to be
 Σ Count_LE(Table[i], item name, value).

[0115]

<METHOD OF OBTAINING RECORD NUMBER OF RECORD ORDER VIEW FROM VALUE ORDER UNION VIEW>

As described above, the item "RecNo" is
5 added to each record of the value order UNION view.
Hence, when a user of the terminal device 10 selects
a desired record from the value order UNION view,
and a position (record number) of the record order
view of the corresponding table 1000 is obtained
10 from the value of the item "RecNo" of the selected
record, a corresponding record may be obtained from
the table 1000 according to the obtained position
(record number).

[0116]

15 The following illustrates a method of
obtaining a table number and a record number of the
record order view relating to the table 1000 with
this table number (relative position from the first
record) from the record number of the value order
20 UNION view (value of the item "RecNo"), using the
functions defined above. The method described below
may be performed by either the display processor 100
or the data processor 200 to which the record number
r in the value order UNION view is given.
25 Hereinafter, the SVL 1100 relates to the item "Name".
Accordingly, the following illustration is given, on
the basis of a parameter "item name" of the above
function being "Name", and the description thereof
is omitted. Specifically, when the value of the
30 item "Name" is v, the function Total_Count_LT(Name,
v) is simply expressed as Total_Count_LT(v).

[0117]

In step S11, find v to which r belongs.

[0118]

The v is obtained by simply searching for v that satisfies the following formula.

[0119]

5 $Total_Count_LT(v) \leq r < Total_Count_LE(v)$
Since each SVL 1100 is arranged in ascending order and has a unique value, the above v may be efficiently searched for using, for example, a known bisection method (binary search method).

10 **[0120]**

As an example, Table 1 illustrates values v corresponding to values r for the item "Name" in the tables 1000-1 and 1000-2 illustrated in FIGS. 3A and 3B.

15 **[0121]**

[Table 1]

| r | Total_Count_LT(v) | Total_Count_LE(v) | v |
|-----|-----------------------|-----------------------|---------|
| 0 | 0 | 2 | Ann |
| 1 | 0 | 2 | Ann |
| 2 | 2 | 5 | Cathy |
| 3 | 2 | 5 | Cathy |
| 4 | 2 | 5 | Cathy |
| 5 | 5 | 7 | Dorothy |
| 6 | 5 | 7 | Dorothy |
| 7 | 7 | 8 | Emily |

20 In step S12, the table number j is obtained from r and v .

[0122]

As can also be seen from Table 1 above, Total_Count_LT(v) indicates a position at which the

value v starts within the value order UNION view.
Hence, when $offset = r - Total_Count_LT(v)$, $offset$ indicates the number of records from the start position of value v to the record of value r in the value order UNION view.

5

[0123]

Accordingly the table number j may be obtained by calculating the minimum j satisfying the following formula 1.

10

[0124]

[Formula 1]

$$\sum_{i=0}^j Count_EQ(Table[i], Item_Name, v) > offset \quad \dots(\text{Formula 1})$$

As an example, Table 2 illustrates the table number j calculated from the formula 1 that is added with respect to Table 1.

15

[0125]

[Table 2]

| r | Total_Count_LT(v) | Total_Count_LE(v) | v | offset | Table_Number j |
|---|-------------------|-------------------|---------|--------|----------------|
| 0 | 0 | 2 | Ann | 0 | 0 |
| 1 | 0 | 2 | Ann | 1 | 1 |
| 2 | 2 | 5 | Cathy | 0 | 0 |
| 3 | 2 | 5 | Cathy | 1 | 0 |
| 4 | 2 | 5 | Cathy | 2 | 1 |
| 5 | 5 | 7 | Dorothy | 0 | 1 |
| 6 | 5 | 7 | Dorothy | 1 | 1 |
| 7 | 7 | 8 | Emily | 0 | 0 |

20 In step S13, the record number r_0 of the record order view is obtained from r , v , and j .

[0126]

First, when the number of all the records in the table 1000 with a table number i that satisfies $i < j$ is "a 1st number of records", the 1st number of records may be obtained by the following formula 2.

[0127]

[Formula 2]

$$\sum_{i=0}^{i < j} \text{Count_LE}(Table[i], Item_Name, v) \quad \dots(\text{Formula 2})$$

Likewise, when the number of all the records in the table 1000 with a table number i that satisfies $i > j$ is "2nd number of records", the 2nd number of record number may be obtained by the following formula 3.

[0128]

[Formula 3]

$$\sum_{i=j+1} \text{Count_LT}(Table[i], Item_Name, v) \quad \dots(\text{Formula 3})$$

Hence, when $r_j = (r - \text{"1st number of records"} - \text{"2nd number of records"})$ is obtained, a position in the value order view relating to the table 1000 with the table number j may be obtained. The record number of the table 1000 may be acquired by $\text{INV}[r_j]$ of the table 1000 with the table number j . Note that this record number is the same as the relative position from the top record in the record order relating to this table 1000.

[0129]

Note that the above record number $\text{INV}[r_j]$

may optionally be converted into the record number r_0 of the value order UNION view by the following method. The record number $INV[r_j]$ of the j th table 1000 (table 1000 with table number j), which acts as
5 the basis of the value order UNION view, may be converted into a record number in value order UNION view (value of item "RecNo") by adding the sum of the number of records constituting each of the 0th to $(j-1)$ th tables 1000.

10 **[0130]**

As described above, the position (record number) of the record order view relating to the corresponding table 1000 may be obtained from the value (record number) of the item "RecNo" of the
15 value order UNION view. Accordingly, when a desired record is selected by a user of the terminal device 10 from the value order UNION view, the selected record may be obtained from the corresponding table 1000 (the table 1000 that stores the selected
20 record).

[0131]

<APPLICATION EXAMPLE 1>

The following describes an application example 1 of the present embodiment. In the
25 application example 1, when there are records with the same value of a certain item in the same table 1000 for displaying a value order UNION view for the certain item, only one of such records is displayed for each table 1000.

30 **[0132]**

For example, when displaying a value order UNION view for the item "Name", the value order UNION view may be generated by deleting the records

having the same value of the item "Name" in the value order view of each table 1000 except one record.

[0133]

5 In the example illustrated in FIG. 11, in the value order view of the table 1000-1 with the table number "0", there are two records with the item "Name" being "Cathy". Hence, one of these two records is deleted.

10 **[0134]**

 Likewise, in the value order view of the table 1000-2 with the table number "1", there are two records with the item "Name" being "Dorothy". Hence, one of these two records is deleted.

15 **[0135]**

 Here, among the records with the same value, a method of selecting one of the records to be left may be determined in advance. For example, among records with the same value, a record having a
20 smallest value for position in the value order view may be left, or a record with the smallest value (record number) of the item "RecNo" may be left.

[0136]

 When only one of the records having the
25 same value is displayed in the value order UNION view in each table 1000 as in the application example 1, the functions Count_LT and Count_LE defined above may be modified as follows.

[0137]

30 • Count_LT(table number, item name, value)
 For example, when the function Count_LT searches the SVL 1100 for the item of the specified item name, and finds an element that matches the specified

value, the function Count_LT may simply obtain i as a position of that element. However, when the function Count_LT searches SVL 1100 for the item of the specified item name, and finds no element that matches the specified value, the function Count_LT may simply obtain, from among the values of the elements of the SVL 1100, i as a position of the element having a value greater than the specified value and the smallest value.

10 **[0138]**

• Count_LE(table number, item name, value)

For example, when the function Count_LE searches the SVL 1100 for the item of the specified item name, and finds an element that matches the specified value, the function Count_LE may simply obtain i+1 as a position of that element. However, when the function Count_LE searches SVL 1100 for the item of the specified item name, and finds no element that matches the specified value, the function Count_LE may simply obtain, from among the values of the elements of the SVL 1100, i as a position of the element having a value greater than the specified value and the smallest value.

20 **[0139]**

25 <APPLICATION EXAMPLE 2>

The following describes an application example 2 of the present embodiment. The application example 2 is an example where the present embodiment is applied to a keyword search.

30 **[0140]**

For example, as illustrated in FIGS. 12(a) and 12(b), records constituting the table 1000-1 and the table 1000-2 each include the item "Text", the

item "Keyword", and the item "RecNo". Here, the item "Text" is a column that stores the text including the search target keyword. Further, the item "Keyword" is a keyword extracted in advance
5 from the text stored in the item "Text".

[0141]

In this case, as described in the present embodiment, it is possible to search for text that includes the specified keyword, from among the text
10 stored in the item "Text", by generating SVL 1100, IND 1200, and INV 1300 with respect to the item "Keyword". In addition, it is also possible to display the value order UNION views of the table 1000-1 and the table 1000-2. As a result, keyword
15 search using a plurality of tables 1000 may be performed easily and quickly.

[0142]

For example, when a user desires to search
20 for text that contains the keyword "car" from the value order view of table 1000-1, INV[3] = 1, INV[4] = 2 and INV[5] = 3 are obtained using SVL 1100-1, IND 1200-1 and INV 1300-1. As a result, the record with record number "1", the record with record
25 number "2", and the record with record number "3" may be acquired from the table 1000-1 as text including the specified keyword. The keyword search in the value order UNION views of the table 1000-1 and the table 1000-2 is similarly performed.

30 **[0143]**

The present invention is not limited to the embodiments specifically disclosed above, and various alterations or modifications may be made

without departing from the spirit and the scope of the invention.

[Description of Reference Signs]

- 5 **[0144]**
- 1 data management system
- 10 terminal device
- 20 data management device
- 100 display processor
- 10 200 data processor
- 1000 table
- 1100 SVL 1100
- 1200 IND 1200
- 1300 INV 1300

15

[Name of Document] Claims

5 1. An information processing system
comprising:
 a storage storing a table including a
plurality of records each including first array data
having an item value that is predetermined; second
10 array data having a record number for identifying
each of the records; and third array data relating
the first array data with the second array data; and
 an acquisition unit configured to acquire,
in response to specification of the item value or
15 the record number, one or more records corresponding
to the specified item value or record number from
among the plurality of records stored in the table,
using the first array data, the second array data,
and the third array data.

20

 2. The information processing system
according to claim 1, wherein
25 the first array data includes respective
predetermined item values stored in the plurality of
records without being duplication in an order of the
item values in an array.

30

 3. The information processing system
according to claim 1 or 2, wherein

the second array data includes the record number stored in an order of the item values in an array.

5

4. The information processing system according to any one of claims 1 to 3, wherein

the third array data includes an
10 accumulated number of occurrences of the item value, obtained by accumulating the number of occurrences of the item value in an order of records stored in the table.

15

5. The information processing system according to any one of claims 1 to 4, further comprising:

20 a plurality of data management devices each having the storage, wherein

the acquisition unit acquires, in response to specification of the item value or the record number, one or more records corresponding to the
25 specified item value or record number, from among the plurality of records stored in the table in the storage included in each of the plurality of data management devices, using the first array data, the second array data, and the third array data stored
30 in the storage included in each of the plurality of data management devices.

6. The information processing system according to claim 5, wherein

5 the acquisition unit acquires, in response to specification of the record number in a unified view of a first table and a second table that are unified in the order of the item values, one or more records corresponding to the specified record number from the second table using the first array data, 10 the second array data, and the third array data corresponding to the second table with replacement of the specified record number.

15

7. The information processing system according to claim 6, wherein

the acquisition unit replaces the record number by subtracting a number of records in the 20 first table from the record number.

8. The information processing system 25 according to any one of claims 1 to 7, wherein

the acquisition unit acquires first information specifying a top record and second information specifying a bottom record from among one or more records corresponding to the specified 30 item value or record number, and retains the acquired first information and second information as search results in a predetermined storage area.

9. An information processing method performed by an information processing device, the information processing device including a storage storing a table including a plurality of records each including first array data having an item value that is predetermined; second array data having a record number for identifying each of the records; and third array data relating the first array data with the second array data, the information processing method comprising:

acquiring one or more records corresponding to a specified item value or record number using the first array data, the second array data, and the third array data, from among the plurality of records stored in the table, in response to the specification of the item value or the record number.

20

[Name of the Document] Abstract

[Abstract]

[Object] To allow users of big data to flexibly
combine multiple big data files distributed over a
5 network such as the Internet to easily search for
desired data and display the search result.

[Solution Means] An information processing system
includes a storage storing a table including a
plurality of records each including first array data
10 having an item value that is predetermined; second
array data having a record number for identifying
each of the records; and third array data relating
the first array data with the second array data; and
an acquisition unit configured to acquire, in
15 response to specification of the item value or the
record number, one or more records corresponding to
the specified item value or record number from among
the plurality of records stored in the table, using
the first array data, the second array data, and the
20 third array data.

[Selected Figure] FIG. 1

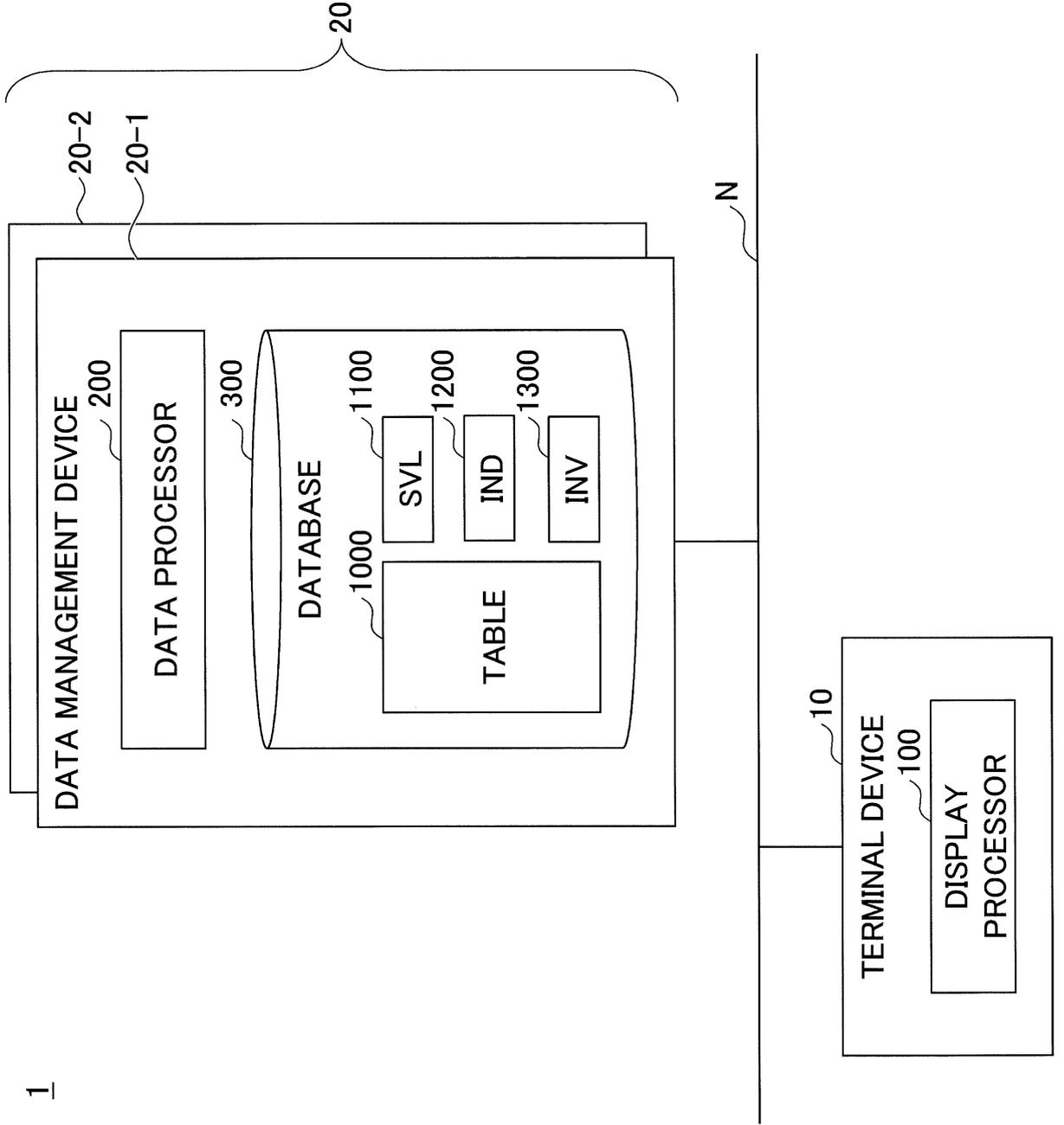


FIG.1

FIG.2

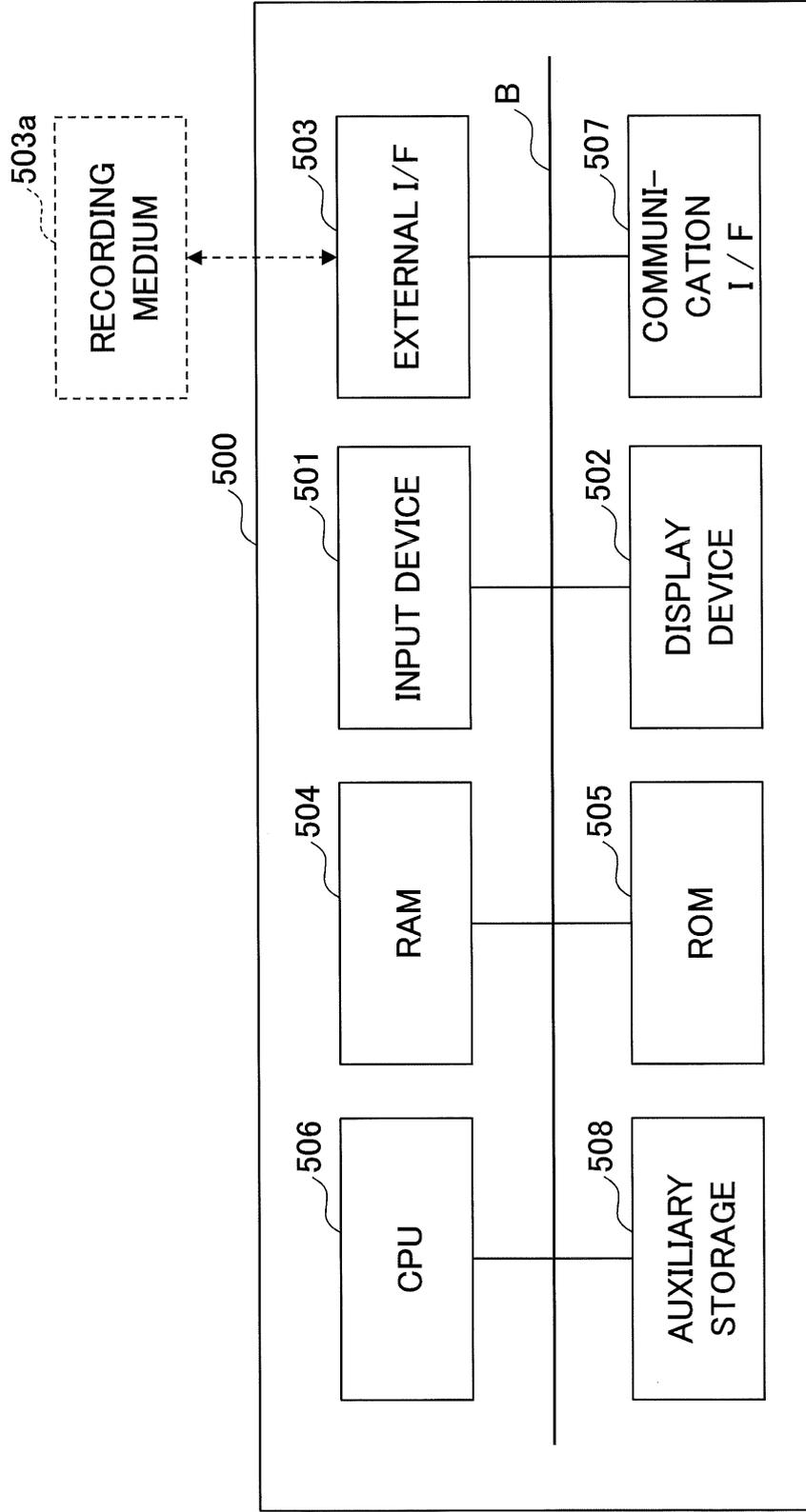


FIG.3

(a) TABLE NUMBER=0

| POSITION | Name | Age |
|----------|-------|-----|
| 0 | Cathy | 8 |
| 1 | Ann | 9 |
| 2 | Emily | 7 |
| 3 | Cathy | 9 |

1000-1

(b) TABLE NUMBER=1

| POSITION | Name | Age |
|----------|---------|-----|
| 0 | Dorothy | 7 |
| 1 | Cathy | 6 |
| 2 | Dorothy | 8 |
| 3 | Ann | 6 |

1000-2

FIG.4

(a)

| POSITION | Name | Age |
|----------|-------|-----|
| 0 | Cathy | 8 |
| 1 | Ann | 9 |
| 2 | Emily | 7 |
| 3 | Cathy | 9 |

VW1-1

(b)

| POSITION | Name | Age |
|----------|---------|-----|
| 0 | Dorothy | 7 |
| 1 | Cathy | 6 |
| 2 | Dorothy | 8 |
| 3 | Ann | 6 |

VW1-2

FIG.5

(a)

| POSITION | Name | Age | RecNo |
|----------|-------|-----|-------|
| 0 | Ann | 9 | 1 |
| 1 | Cathy | 8 | 0 |
| 2 | Cathy | 9 | 3 |
| 3 | Emily | 7 | 2 |

VW2-1

(b)

| POSITION | Name | Age | RecNo |
|----------|---------|-----|-------|
| 0 | Ann | 6 | 3 |
| 1 | Cathy | 6 | 1 |
| 2 | Dorothy | 7 | 0 |
| 3 | Dorothy | 8 | 2 |

VW2-2

FIG.6

| POSITION | Name | Age |
|----------|---------|-----|
| 0 | Cathy | 8 |
| 1 | Ann | 9 |
| 2 | Emily | 7 |
| 3 | Cathy | 9 |
| 4 | Dorothy | 7 |
| 5 | Cathy | 6 |
| 6 | Dorothy | 8 |
| 7 | Ann | 6 |

VW3

RE-PLACE

FIG.7

| POSITION | Name | Age | RecNo |
|----------|---------|-----|-------|
| 0 | Ann | 9 | 1 |
| 1 | Ann | 6 | 7 |
| 2 | Cathy | 8 | 0 |
| 3 | Cathy | 9 | 3 |
| 4 | Cathy | 6 | 5 |
| 5 | Dorothy | 7 | 4 |
| 6 | Dorothy | 8 | 6 |
| 7 | Emily | 7 | 2 |

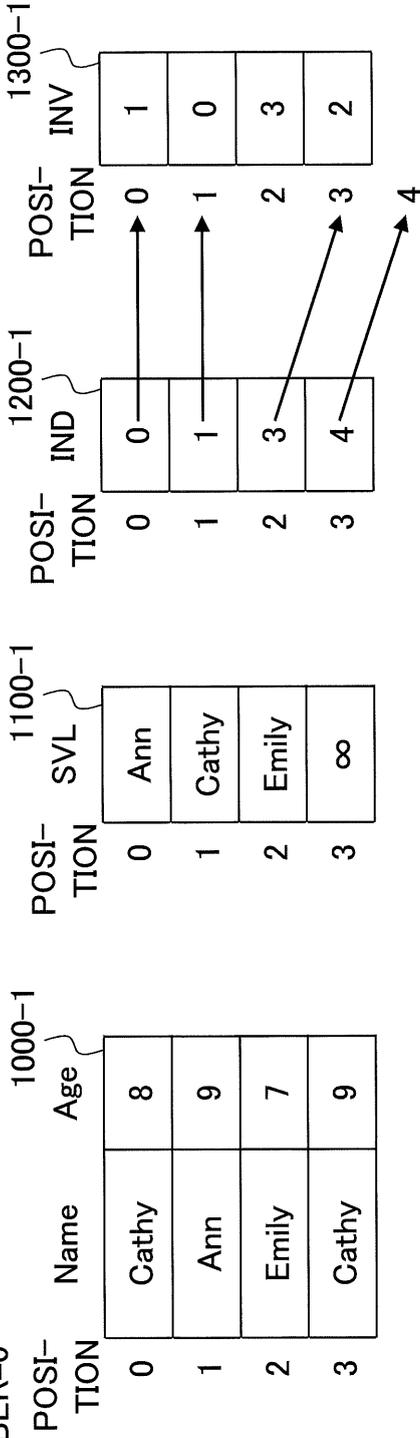
VW4

RE-PLACE

RE-PLACE

FIG.8

(a) TABLE NUMBER=0



(b) TABLE NUMBER=1

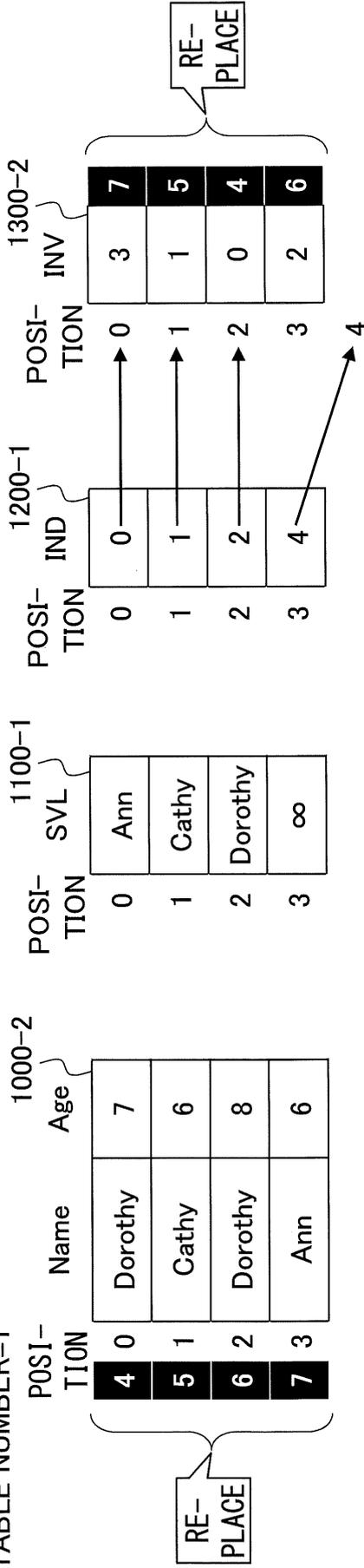
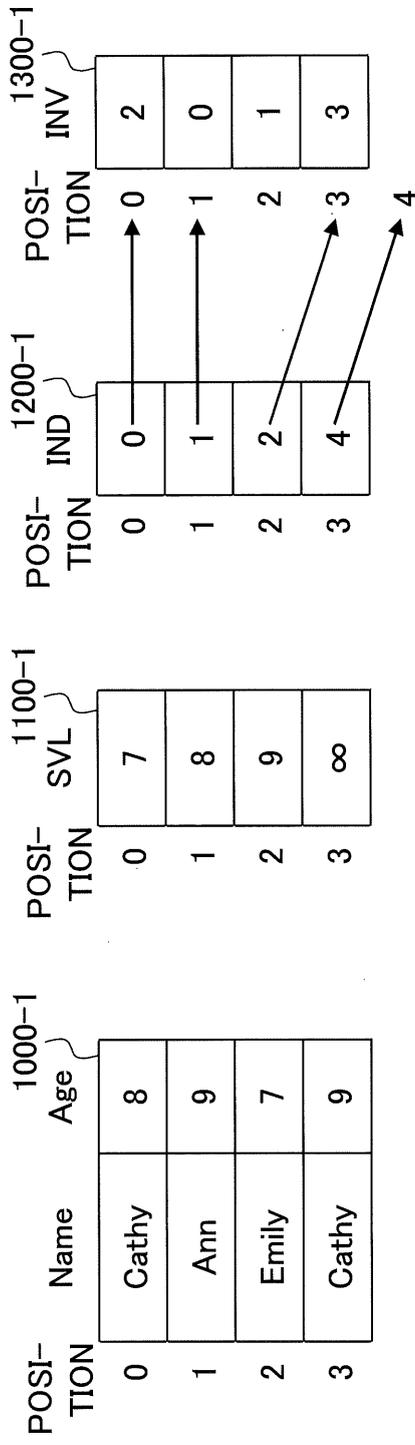
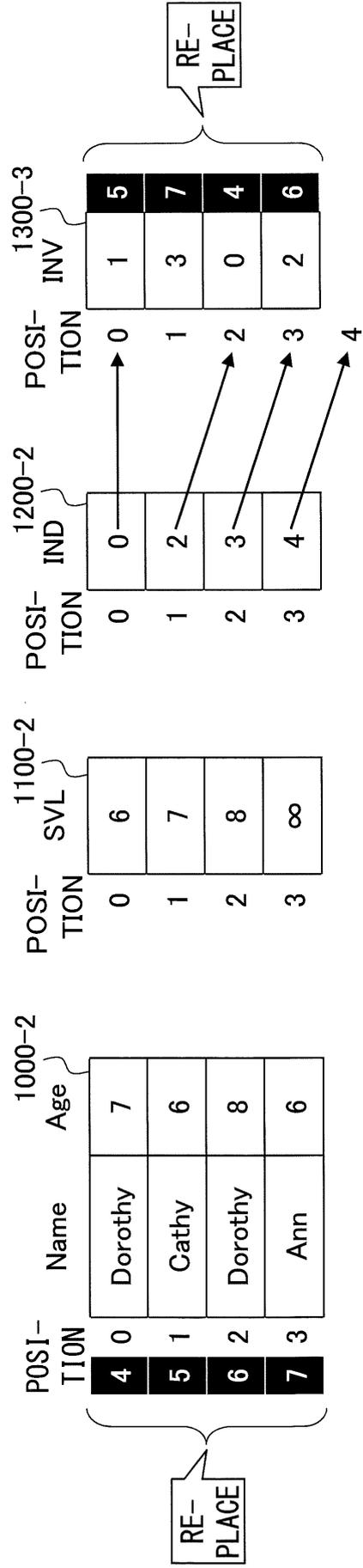


FIG.9

(a) TABLE NUMBER=0



(b) TABLE NUMBER=1



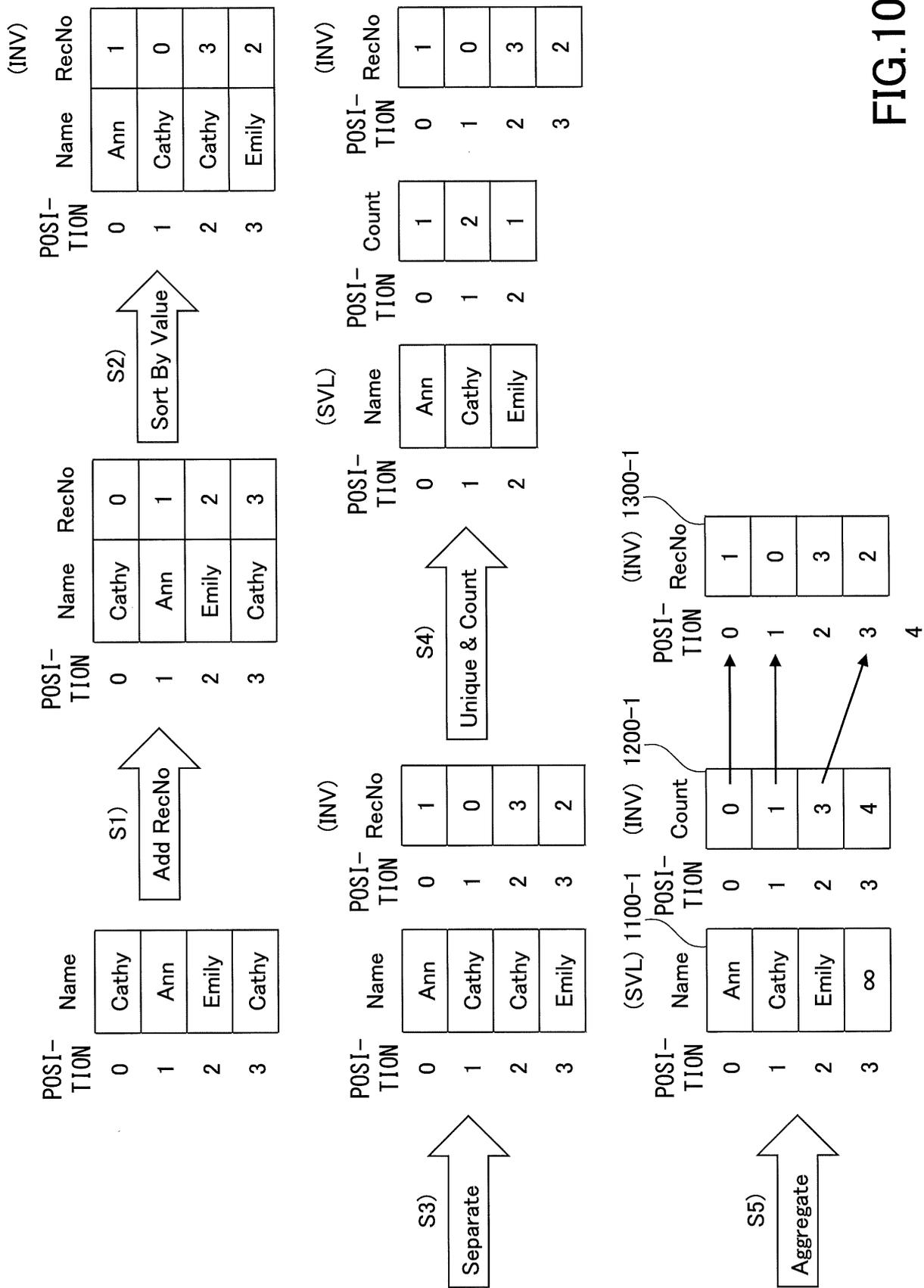


FIG.10

FIG.11

<VALUE ORDER UNION VIEW WITH RESPECT TO NAME>

<VALUE ORDER UNION VIEW WITH RESPECT TO NAME>

TABLE NUMBER=0

| POSITION | Name | Age | RecNo |
|----------|-------|-----|-------|
| 0 | Ann | 9 | 1 |
| 1 | Cathy | 8 | 0 |
| 2 | Cathy | 9 | 3 |
| 3 | Emily | 7 | 2 |



| POSITION | Name | Age | RecNo |
|----------|-------|-----|-------|
| 0 | Ann | 9 | 1 |
| 1 | Cathy | 8 | 0 |
| 2 | Emily | 7 | 2 |

TABLE NUMBER=1

| POSITION | Name | Age | RecNo |
|----------|---------|-----|-------|
| 0 | Ann | 6 | 3 |
| 1 | Cathy | 6 | 1 |
| 2 | Dorothy | 7 | 0 |
| 3 | Dorothy | 8 | 2 |



| POSITION | Name | Age | RecNo |
|----------|---------|-----|-------|
| 0 | Ann | 6 | 3 |
| 1 | Cathy | 6 | 1 |
| 2 | Dorothy | 7 | 0 |



| POSITION | Name | Age | RecNo |
|----------|---------|-----|-------|
| 0 | Ann | 9 | 1 |
| 1 | Ann | 6 | 7 |
| 2 | Cathy | 8 | 0 |
| 3 | Cathy | 6 | 5 |
| 4 | Dorothy | 7 | 4 |
| 5 | Emily | 7 | 2 |

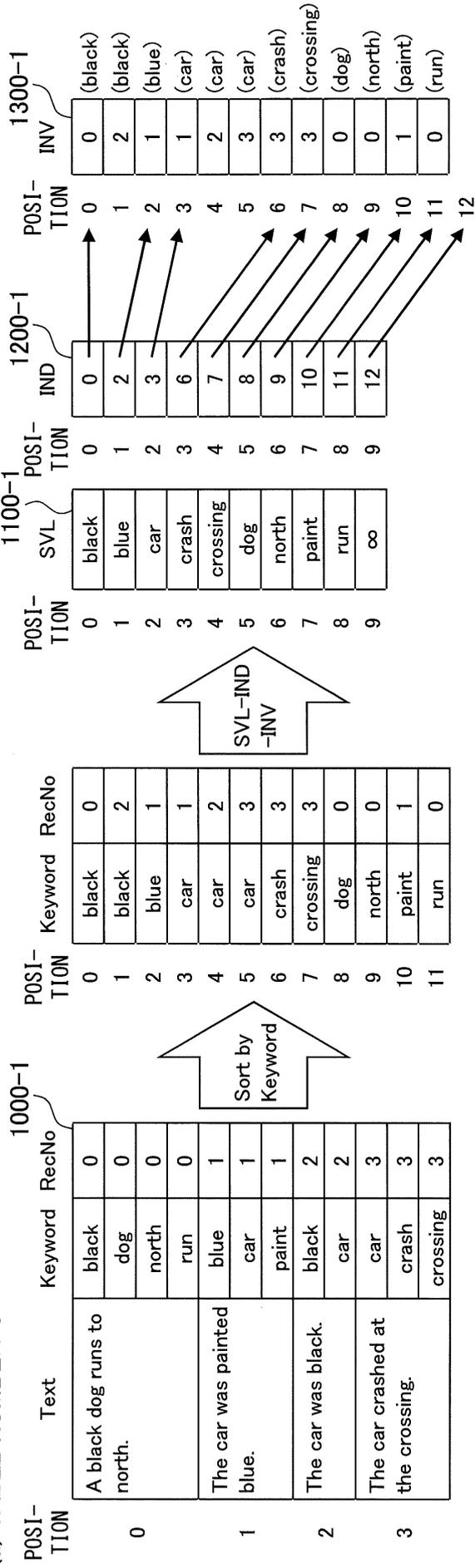
RE-PLACE

RE-PLACE



FIG.12

(a) TABLE NUMBER=0



(b) TABLE NUMBER=1

