[19] 中华人民共和国国家知识产权局

[51] Int. Cl.
G06F 17/30 (2006.01)
G06F 19/00 (2006.01)

[12] 发明专利说明书

专利号 ZL 00812571.6

[45] 授权公告日 2007 年 4 月 11 日

[11] 授权公告号 CN 1310173C

[22] 申请日 2000.7.31 [21] 申请号 00812571.6
[30] 优先权
[32] 1999. 7.29 [33] JP [31] 215450/99
[86] 国际申请 PCT/JP2000/005135 2000.7.31
[87] 国际公布 WO2001/009764 日 2001.2.8
[85] 进入国家阶段日期 2002.3.7
[73] 专利权人 特博数据实验室公司
地址 日本东京
[72] 发明人 古庄晋二
[56] 参考文献
US5822720A 1998.10.13
US5893899A 1999.4.13
审查员 鲍 薇

[74] 专利代理机构 中国国际贸易促进委员会专利商标事务所
代理人 付建军

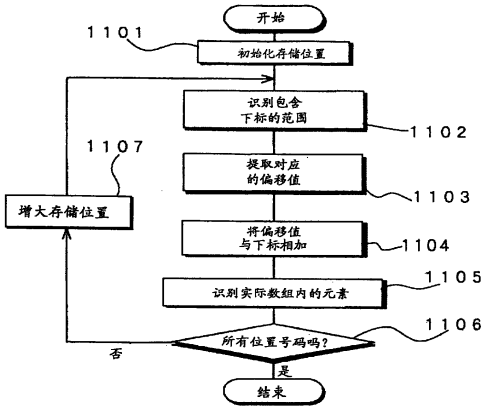
权利要求书 15 页 说明书 45 页 附图 51 页

[54] 发明名称

表格式数据显示方法、插入方法、删除方法和更新方法

[57] 摘要

本发明的目标是快速而正确地执行表格式数据的插入、删除和更新。 CPU 12 接受记录号作为下标，生成下标转换数组，用于给出对应于所述的下标的范围的偏移值，标识插入位置，该位置指出要插入的字段值的位置，在下标转换数组中，有关插入位置，给出偏移值，该偏移值定义对应的下标的范围，还标识数组的末尾，在下标转换数组中，有关那些记录号大于对应于所述的插入位置的记录号的记录，给出偏移值，该偏移值还增大对应的下标的范围，还减少能接受的下标，将要插入的所述的字段值放在规定的末尾位置，这样，根据下标转换数组内的下标的范围，给出偏移值作为所述的下标，因此，用偏移值给出的下标来标识数组的字段值。



1. 一种插入表格式数据的方法，其中，在表格式数据中的任意位置插入字段值，所述表格式数据用存储记录数组的值来表示，在存储器中包含字段和字段值，其中

所述的插入表格式数据的方法的特征在于包括下列步骤:

在所述存储器中生成下标转换数组,这种数组的构成使其能够接受记录号作为下标，并给出对应于所述的下标的范围的偏移值，

通过审阅存储数组的所述值中的字段值，标识插入位置，该位置指出要插入的字段值的位置，

在所述的下标转换数组中，有关所述的插入位置，给出偏移值，该偏移值定义对应的下标的范围，还标识在所述的数组的末尾后面的规定的位置，

在所述的下标转换数组中,有关那些记录号大于对应于所述的插入位置的记录号的记录，给出偏移值，该偏移值向上移动对应的下标范围，该偏移值还减少所接受的下标，

将要插入的所述的字段值放在存储器中存储数组的所述值的末尾后面的规定的位置，

这样，根据下标转换数组内的下标的范围，给出偏移值作为所述的下标，通过所述的偏移值给出的下标来标识具有存储数组的值的字段值。

2. 一种删除表格式数据的方法，其中，在表格式数据中的任意位置删除字段值，所述表格式数据由存储记录数组的值来表示，在存储器中包含字段和字段值，其中

所述的删除表格式数据的方法的特征在于包括下列步骤:

在所述存储器中生成下标转换数组,这种数组的构成使其能够接受记录号作为下标，并给出对应于所述的接受的下标的范围的偏移值，

通过审阅存储数组的所述值中的字段值，标识删除位置，该位置指出要删除的字段值的位置，

在所述的下标转换数组中,有关那些记录号大于对应于所述的删除位

置的记录号的记录，给出偏移值，该偏移值向下移动对应的下标范围，该偏移值还增大所接受的下标，

这样，根据下标转换数组内的下标的范围，给出偏移值作为所述的下标，通过所述的偏移值给出的下标来标识具有存储数组的值的字段值。

3. 一种更新表格式数据的方法，其中，在表格式数据中的任意位置更新字段值，所述表格式数据由存储记录数组的值来表示，在存储器中包含字段和字段值，其中

所述的更新表格式数据的方法的特征在于包括下列步骤:

在所述存储器中生成下标转换数组，这种数组的构成使其能够接受记录号作为下标，并给出对应于所述的接受的下标的范围的偏移值，

(1) 通过审阅存储数组的所述值中的字段值，假设要更新的字段值的位置是删除位置，

在所述的下标转换数组中，有关那些记录号大于对应于所述的删除位置的记录号的记录，给出偏移值，该偏移值向下移动对应的下标范围，该偏移值还增大所接受的下标，

(2) 通过审阅存储数组的所述值中的字段值，假设要更新的字段值的位置是插入位置，

在所述的下标转换数组中，有关所述的插入位置，给出偏移值，该偏移值定义对应的下标的范围，该偏移值还标识所述的数组的末尾后面的规定的位置，

在所述的下标转换数组中，有关那些记录号大于对应于所述的插入位置的记录号的记录，给出偏移值，该偏移值向上移动对应的下标范围，该偏移值还减少能接受的下标，

将要更新的所述的字段值放在所述存储器中存储数组的所述值的末尾后面规定的位置，以及

所述的 (1) 和 (2) 或 (2) 和 (1) 按顺序执行，

这样，根据下标转换数组内的下标的范围，给出偏移值作为所述的下标，通过所述的偏移值给出的下标来标识具有存储数组的值的字段值。

4. 一种针对表格式数据的值转换方法，该方法转换表格式数据中的

字段值的值，所述表格式数据由存储记录数组的值来表示，在存储器中包含字段和字段值，其中

所述的针对表格式数据的值转换方法包括下列步骤:

在所述存储器中生成值转换数组，这种数组的构成使其能: 给出对应于所述的字段值的范围的偏移值，

接受记录号作为下标，将对应于所述的字段值的范围的偏移值给予对应于所述存储器中存储数组的所述值内的所述的下标的字段值。

5. 一种插入表格式数据的方法，其中，在表格式数据中的任意位置插入字段值，所述表格式数据由这样一种数据结构来表示，所述表格式数据的数据结构的构成使对应于任意记录号的字段值由表格式数据来标识，表格式数据由记录数组来表示，包含字段，字段里面包含字段值，其中

所述的表格式数据的数据结构具有这样的特征:

所述的表格式数据以这样的方式在存储器中构成，它分为一个或多个信息块，信息块包括: 一个值列表，该列表包含第一个实际数组和一个指针数组，在所述第一个实际数组中，字段值以字段值号码的顺序存储，字段值号码对应于字段值，该字段值属于指定的字段，在所述指针数组中包含第二个实际数组，在该数组中，用于指向所述的字段值号码的指针值按唯一的记录号码顺序存储，

在所述存储器中第二个下标转换数据的构成使给出指针数组的输入、输出作为所述的值列表中的下标，给出并形成对应于所述的下标的范围的第三个偏移，第一个实际数组内的字段值通过指针数组的输出来标识，指针数组的输出由偏移通过所述的第二个下标转换数组给出，

其中，

所述的插入表格式数据的方法包括下列步骤:

(1) 有关所述的值列表:

标识插入位置，该位置指出要插入的字段值的位置，

在所述的第二个下标转换数组中，有关所述的插入位置，给出第三个偏移值，该偏移值定义对应的下标的范围，该偏移值还标识所述的第一

个数组的末尾后面的规定的位置，

在所述的第二个下标转换数组中，有关那些值大于对应于所述的插入位置的下标的记录，给出第三个偏移值，该偏移值向上移动对应的下标范围，以使定义所述的范围的值变大，该偏移值还减少能接受的下标，

将要插入的所述的字段值放在所述的第一个实际数组的末尾后面规定的位置。

(2) 有关所述的指针数组，

标识插入位置，该插入位置指出对应于要插入的记录号的指针值的位置，

在所述的第一个下标转换数组中，有关所述的插入位置，给出第一个偏移值，该偏移值定义对应的下标，还标识所述的第一个实际数组末尾后面的规定的位置，

在所述的第一个下标转换数组中，有关那些记录号大于对应于所述的插入位置的记录号的记录，给出第一个偏移值，该偏移值向上移动对应的下标范围，以使定义所述的范围的值变大，该偏移值还减少所接受的下标，

将大于现有指针值的新指针值放在所述的第二个实际数组的末尾后面规定的位置。

(3) 有关所述的指针数组，

在所述的值转换数组中，给出第二个偏移值，该偏移值增大那些值大于对应于所述的值列表中的插入位置的指针值的记录，

在所述的值转换数组中，给出第二个偏移值，以使所述的新指针值标识对应于所述的插入位置的位置。

6. 根据权利要求 5 的插入表格式数据的方法，其特征在于:

表包括 m 个记录号和 n 个字段值，

当所述的值列表内的字段值的插入位置为 i 时（0<=i<=n-1），字段值放于第一个实际数组的末尾 n 处，记录号插入位置为 j（0<=j=m-1），指针值放于第二个实际数组的末尾 m 处，

(1) 在所述的值列表的第二个下标转换数组中:

当值位于 (i-1) 或更小范围内时，给出 0 作为第三个偏移值，

当值为 i 时，给出 (n-1) 作为第三个偏移值，

当值位于 (i＋1) 或更大和 n 或更小范围内时，给出 (-1) 作为第三个偏移值，

(2) 在所述的指针数组的第一个下标转换数组中：

当下标位于 (j-1) 或更小范围内时，给出 0 作为第一个偏移值，

当下标为 j 时，给出 (m-j) 作为第一个偏移值，

当下标位于 (j＋1) 或更大和 m 或更小范围内时，给出 (-1) 作为第一个偏移值，

(3) 在所述的指针数组的值转换数组中：

当值位于 (i-1) 范围内时，给出 0 作为第二个偏移值，

当值在 i 或更大和 (n-1) 或更小范围内时，给出 1 作为第二个偏移值，

当值为 n 时，给出 (i-n) 作为第二个偏移值，

7. 根据权利要求 5 的插入表格式数据的方法，其特征在于：

表包括 m 个记录号和 n 个字段值，

当所述的值列表内的字段值的插入位置为 i (0<=i<=n-1) 时，字段值放于第一个实际数组末尾后面的规定的位置 z (z>=n) 处，记录号插入位置为 j (0<=j<=m-1) 时，指针值放于第二个实际数组末尾后面的规定位置 x (x>=m) 处，

(1) 在所述的值列表的第二个下标转换数组中：

当值位于 (i-1) 或更小范围内时，给出 0 作为第三个偏移值，

当值为 i 时，给出 (z-1) 作为第三个偏移值，

当值位于 (i＋1) 或更大和 n 或更小范围内时，给出 (-1) 作为第三个偏移值，

(2) 在所述的指针数组的第一个下标转换数组中：

当下标位于 (j-1) 或更小范围内时，给出 0 作为第一个偏移值，

当下标为 j 时，给出 (x-j) 作为第一个偏移值，

当下标位于 (j＋1) 或更大和 m 或更小范围内时，给出 (-1) 作为第

一个偏移值，

(3) 在所述的指针数组的值转换数组中：

当值位于 (i-1) 范围内时，给出 0 作为第二个偏移值，

当值在 i 或更大和 (n-1) 或更小范围内时，给出 1 作为第二个偏移值，

当值为 y 时（其中 y 是存储在第二个实际数组的位置 x 处的指针值），给出 (i-y) 作为第二个偏移值。

8. 根据权利要求 5 的插入表格式数据的方法，其特征在于，每个所述的下标转换数组包括：起始位置数组，该数组包括指出每个规定的范围内的下标的最小值的起始位置，和/或结束位置数组，该数组包括指出所述的规定的范围内的下标的最大值的结束位置，偏移数组，该数组包括对应的偏移值。

9. 根据权利要求 5 的插入表格式数据的方法，其特征在于，所述的值转换数组包括：起始位置数组，该数组包括指出每个规定的范围内的下标的最小值的起始位置，和/或结束位置数组，该数组包括指出所述的规定的范围内的值的最大值的结束位置，偏移数组，该数组包括对应的偏移值。

10. 一种删除表格式数据的方法，其中，在表格式数据中的任意位置删除字段值，所述表格式数据由这样一种数据结构来表示，所述表格式数据的数据结构的构成使对应于任意记录号的字段值由表格式数据来标识，表格式数据由记录数组来表示，包含字段，字段里面包含字段值，其中

所述的表格式数据的数据结构具有这样的特征：

所述的表格式数据以这样的方式在存储器中构成，它分为一个或多个信息块，信息块包括：一个值列表，该列表包含第一个实际数组和一个指针数组，在所述第一个实际数组中，字段值以字段值号码的顺序存储，字段值号码对应于字段值，该字段值属于指定的字段，在所述指针数组中包含第二个实际数组，在该数组中，用于指向所述的字段值号码的指针值按唯一的记录号码顺序存储，

在所述存储器中第二个下标转换数据的构成使给出指针数组的输入、输出作为所述的值列表中的下标，给出并形成对应于所述的下标的范围的第三个偏移，第一个实际数组内的字段值通过指针数组的输出来标识，指针数组的输出由偏移通过所述的第二个下标转换数组给出，

其中，

所述的删除表格式数据的方法包括下列步骤:

有关所述的指针数组，

标识删除位置，该删除位置指出对应于要删除的指针值的位置，

在所述的第一个下标转换数组中，有关所述的插入位置，定义对应的下标，

在所述的第一个下标转换数组中,有关那些记录号大于对应于所述的删除位置的记录号的记录，给出第一个偏移值，该偏移值向下移动对应的下标范围，以使定义所述的范围的值变大，该偏移值还减少所接受的下标。

11. 根据权利要求 **10** 的删除表格式数据的方法，其特征在于:

表包括 **m** 个记录号和 **n** 个字段值，

当记录号删除位置为 **j (0<=j<=m-1)** 时，

在所述的指针数组的第一个下标转换数组中:

当下标位于 **(j-1)** 或更小范围内时，给出 **0** 作为第一个偏移值，

当下标位于 **j** 或更大和 **(m-2)** 或更小范围内时，给出 **1** 作为偏移值。

12. 根据权利要求 **10** 的删除表格式数据的方法，其特征在于，每个所述的下标转换数组包括: 起始位置数组，该数组包括指出每个规定的范围内的下标的最小值的起始位置，和/或结束位置数组，该数组包括指出所述的规定的范围内的下标的最大值的结束位置，偏移数组，该数组包括对应的偏移值。

13. 根据权利要求 **10** 的删除表格式数据的方法，其特征在于，所述的值转换数组包括: 起始位置数组，该数组包括指出每个规定的范围内的下标的最小值的起始位置，和/或结束位置数组，该数组包括指出所述

的规定的范围内的值的最大值的结束位置，偏移数组，该数组包括对应的偏移值。

14. 一种更新表格式数据的方法，其中，在表格式数据中的任意位置更新字段值，所述表格式数据由这样一种数据结构来表示，所述表格式数据的数据结构的构成使对应于任意记录号的字段值由表格式数据来标识，表格式数据由记录数组来表示，包含字段，字段里面包含字段值，其中

所述的表格式数据的数据结构具有这样的特征:

所述的表格式数据以这样的方式在存储器中构成,它分为一个或多个信息块，信息块包括：一个值列表，该列表包含第一个实际数组和一个指针数组，在所述第一个实际数组中，字段值以字段值号码的顺序存储，字段值号码对应于字段值，该字段值属于指定的字段，在所述指针数组中包含第二个实际数组，在该数组中，用于指向所述的字段值号码的指针值按唯一的记录号码顺序存储，

在所述存储器中第二个下标转换数据的构成使给出指针数组的输入、输出作为所述的值列表中的下标，给出并形成对应于所述的下标的范围的第三个偏移，第一个实际数组内的字段值通过指针数组的输出来标识，指针数组的输出由偏移通过所述的第二个下标转换数组给出，

其中

所述的更新表格式数据的方法包括下列步骤:

(A) (1) 有关所述的值列表:

假设要更新的字段值的位置是插入位置，标识插入位置，该位置指出要插入的字段值的位置，

在所述的第二个下标转换数组中，有关所述的插入位置，给出第三个偏移值，该偏移值定义对应的下标的范围，还标识所述的第一个数组的末尾后面的规定的位置，

在所述的第二个下标转换数组中,有关那些值大于对应于所述的插入位置的下标的记录，给出第三个偏移值，该偏移值向上移动对应的下标范围，以使定义所述的范围的值变大，该偏移值还减少所接受的下标，

将要插入的所述的字段值放在所述的第一个实际数组的末尾后面规定的位置，

(2) 有关所述的指针数组，

假设要更新的指针值为要插入的指针值，标识插入位置，该插入位置指出要插入的指针值的位置，

在所述的第一个下标转换数组中，有关所述的插入位置，给出第一个偏移值，该偏移值定义对应的下标，还标识所述的第一个实际数组末尾后面的规定的位置，

在所述的第一个下标转换数组中，有关那些记录号大于对应于所述的插入位置的记录号的记录，给出第一个偏移值，该偏移值向上移动对应的下标范围，以使定义所述的范围的值变大，该偏移值还减少所接受的下标，

将大于现有指针值的新指针值放在所述的第二个实际数组的末尾后面规定的位置。

(3) 有关所述的指针数组，

在所述的值转换数组中，给出第二个偏移值，该偏移值增大那些值大于对应于所述的值列表中的插入位置的指针值的记录，

在所述的值转换数组中，给出第二个偏移值，以使所述的新指针值标识对应于所述的插入位置的位置。

(B) 有关所述的指针数组，

将要更新的指针值的位置标识为删除位置，考虑所述的插入位置，

在所述的第一个下标转换数组中，有关所述的删除位置，定义对应的下标，

在所述的第一个下标转换数组中，有关那些记录号大于对应于所述的删除位置的记录号的记录，给出第一个偏移值，该偏移值向下移动对应的下标范围，以使定义所述的范围的值变小，该偏移值还增大所接受的下标，以及

所述的 (A) 和 (B) 或所述的 (B) 和 (A) 按顺序执行。

15. 根据权利要求 14 的更新表格式数据的方法，其特征在于：

表包括 m 个记录号和 n 个字段值，

当所述的值列表内的字段值的更新位置为 i (0<=i<=n-1) 时，字段值放于第一个实际数组末尾 n 处，记录号更新位置为 j (0<=j<=m-1) 时，指针值放于第二个实际数组末尾 m 处，

(1) 在所述的值列表的第二个下标转换数组中：

当值位于 (i-1) 或更小范围内时，给出 0 作为第三个偏移值，

当值为 i 时，给出 (n-1) 作为第三个偏移值，

当值位于 (i＋1) 或更大和 n 或更小范围内时，给出 (-1) 作为第三个偏移值，

(2) 在所述的指针数组的第一个下标转换数组中：

当下标位于 (j-1) 或更小范围内时，给出 0 作为第一个偏移值，

当下标为 j 时，给出 (m-j) 作为第一个偏移值，

当下标位于 (j＋1) 或更大和 (m+1) 或更小范围内时，给出 0 作为第一个偏移值，

(3) 在所述的指针数组的值转换数组中：

当值位于 (i-1) 范围内时，给出 0 作为第二个偏移值，

当值在 i 或更大和 (n-1) 或更小范围内时，给出 1 作为第二个偏移值，

当值为 n 时，给出 (i-n) 作为第二个偏移。

16. 根据权利要求 14 的更新表格式数据的方法，其特征在于：

表包括 m 个记录号和 n 个字段值，

当所述的值列表内的字段值的插入位置为 i (0<=i<=n-1) 时，字段值放于第一个实际数组末尾后面的规定的位置 z (z>=n) 处，记录号插入位置为 j (0<=j<=m-1) 时，指针值放于第二个实际数组末尾后面的规定位置 x (x>=m) 处，

(1) 在所述的值列表的第二个下标转换数组中：

当值位于 (i-1) 或更小范围内时，给出 0 作为第三个偏移值，

当值为 i 时，给出 (z-1) 作为第三个偏移值，

当值位于 (i＋1) 或更大和 n 或更小范围内时，给出 (-1) 作为第三

个偏移值，

(2) 在所述的指针数组的第一个下标转换数组中：

当下标位于 (j-1) 或更小范围内时，给出 0 作为第一个偏移值，

当下标为 j 时，给出 (x-j) 作为第一个偏移值，

当下标位于 (j＋1) 或更大和 (m+1) 或更小范围内时，给出 0 作为第一个偏移值，

(3) 在所述的指针数组的值转换数组中：

当值位于 (i-1) 范围内时，给出 0 作为第二个偏移值，

当值在 i 或更大和 (n-1) 或更小范围内时，给出 1 作为第二个偏移值，

当值为 y 时（其中 y 是存储在第二个实际数组的位置 x 处的指针值），给出 (i-y) 作为第二个偏移值。

17. 根据权利要求 14 的更新表格式数据的方法，其特征在于，每个所述的下标转换数组包括：起始位置数组，该数组包括指出每个规定的范围内的下标的最小值的起始位置，和/或结束位置数组，该数组包括指出所述的规定的范围内的下标的最大值的结束位置，偏移数组，该数组包括对应的偏移值。

18. 根据权利要求 14 的更新表格式数据的方法，其特征在于，所述的值转换数组包括：起始位置数组，该数组包括指出每个规定的范围内的下标的最小值的起始位置，和/或结束位置数组，该数组包括指出所述的规定的范围内的值的最大值的结束位置，偏移数组，该数组包括对应的偏移值。

19. 一种删除表格式数据的方法，其中，在表格式数据中的任意位置删除字段值，所述表格式数据由这样一种数据结构来表示，所述表格式数据的数据结构的构成使对应于任意记录号的字段值由表格式数据来标识，表格式数据由记录数组来表示，包含字段，字段里面包含字段值，其中

所述的表格式数据的数据结构具有这样的特征：

所述的表格式数据以这样的方式在存储器中构成，它分为一个或多个

信息块，信息块包括：一个值列表，该列表包含第一个实际数组和一个指针数组，在所述第一个实际数组中，字段值以字段值号码的顺序存储，字段值号码对应于字段值，该字段值属于指定的字段，在所述指针数组中包含第二个实际数组，在该数组中，用于指向所述的字段值号码的指针值按唯一的记录号码顺序存储，

在所述存储器中第二个下标转换数据的构成使给出指针数组的输入、输出作为所述的值列表中的下标，给出并形成对应于所述的下标的范围的第三个偏移，第一个实际数组内的字段值通过指针数组的输出来标识，指针数组的输出由偏移通过所述的第二个下标转换数组给出，

其中

所述的删除表格式数据的方法包括下列步骤：

有关所述的记录号，提供第四个实际数组，在该数组中，放置记录号本身，第三个下标转换数组根据下标的范围给定的规定偏移值用于标识所述的记录号，

标识删除位置，该删除位置指出要删除的所述的记录号的位置，

在所述的第三个下标转换数组中，定义涉及所述删除的对应下标，

在所述的第三个下标转换数组中，有关那些值大于对应于所述的删除位置的下标的记录，给出第四个偏移值，该偏移值向下移动对应的下标范围，以使定义所述的范围的值变小，该偏移值还增大所接受的下标。

20. 根据权利要求 **19** 的删除表格式数据的方法，其特征在于，每个所述的下标转换数组包括：起始位置数组，该数组包括指出每个规定的范围内的下标的最小值的起始位置，和/或结束位置数组，该数组包括指出所述的规定的范围内的下标的最大值的结束位置，偏移数组，该数组包括对应的偏移值。

21. 根据权利要求 **19** 的删除表格式数据的方法，其特征在于，所述的值转换数组包括：起始位置数组，该数组包括指出每个规定的范围内的下标的最小值的起始位置，和/或结束位置数组，该数组包括指出所述的规定的范围内的值的最大值的结束位置，偏移数组，该数组包括对应的偏移值。

22. 一种删除表格式数据的方法，其中，在表格式数据中的任意位置删除字段值，所述表格式数据由记录数组来表示，包含字段，字段里面包含字段值，其中

所述的删除表格式数据的方法包括下列步骤:

有关所述的记录号，提供一个数组，在该数组中，放置记录号本身，下标转换数组根据下标的范围给定的规定偏移值用于标识所述的记录号，

标识删除位置，该删除位置指出要删除的所述的记录号的位置，

在所述的下标转换数组中，定义涉及所述删除的对应下标，

在所述的下标转换数组中，有关那些值大于对应于所述的删除位置的下标的记录，给出偏移值，该偏移值向下移动对应的下标范围，以使定义所述的范围的值变小，该偏移值还增大所接受的下标。

23. 一种事务处理的方法，其特征在于，当通过权利要求 **5-18** 中任何权利要求所述的方法执行表格式数据的插入、删除和/或更新时，执行回滚或提交，抛弃所述的下标转换数组和值转换数组。

24. 一种事务处理的方法，其特征在于，当通过权利要求 **5-18** 中任何权利要求所述的方法执行表格式数据的插入、删除和/或更新时，执行回滚，抛弃所述的下标转换数组和值转换数组。

25. 一种并行处理方法，其中，对具有这样一种数据结构的表格式数据同时执行许多过程，所述表格式数据的数据结构的构成使对应于任意记录号的字段值由表格式数据来标识，表格式数据由记录数组来表示，包含字段，字段里面包含字段值，其中

所述的表格式数据的数据结构具有这样的特征:

所述的表格式数据以这样的方式在存储器中构成，它分为一个或多个信息块，信息块包括：一个值列表，该列表包含第一个实际数组和一个指针数组，在所述第一个实际数组中，字段值以字段值号码的顺序存储，字段值号码对应于字段值，该字段值属于指定的字段，在所述指针数组中包含第二个实际数组，在该数组中，用于指向所述的字段值号码的指针值按唯一的记录号码顺序存储，

在所述存储器中第二个下标转换数据的构成使给出指针数组的输入、输出作为所述的值列表中的下标，给出并形成对应于所述的下标的范围的第三个偏移，第一个实际数组内的字段值通过指针数组的输出来标识，指针数组的输出由偏移通过所述的第二个下标转换数组给出，

其中，

在前述的过程中，那些涉及数据的插入、删除和/或更新的过程这样构成，以便在所述的指针数组中，它们经过第一个下标转换数组、第二个实际数组和值转换数组，而且在所述的值列表中，它们经过第二个下标转换数组和第一个实际数组，

另一方面，在前述的过程中，那些不涉及数据的插入、删除和/或更新的过程这样构成，以便在所述的指针数组中，它们经过第二个下标转换数组，而且，在所述的值列表中，它们经过第一个实际数组。

26. 根据权利要求 **25** 的并行处理方法，其特征在于，那些涉及数据的插入、删除和/或更新的过程可以利用根据权利要求 **5-18** 中任何权利要求的方法。

27. 一种对具有这样一种数据结构的表格式数据进行记录锁定方法，，所述表格式数据的数据结构的构成使对应于任意记录号的字段值由表格式数据来标识，表格式数据由记录数组来表示，包含字段，字段里面包含字段值，其中

所述的表格式数据的数据结构具有这样的特征:

所述的表格式数据以这样的方式在存储器中构成,它分为一个或多个信息块，信息块包括：一个值列表，该列表包含第一个实际数组和一个指针数组，在所述第一个实际数组中，字段值以字段值号码的顺序存储，字段值号码对应于字段值，该字段值属于指定的字段，在所述指针数组中包含第二个实际数组，在该数组中，用于指向所述的字段值号码的指针值按唯一的记录号码顺序存储，

在所述存储器中第二个下标转换数据的构成使给出指针数组的输入、输出作为所述的值列表中的下标，给出并形成对应于所述的下标的范围的第三个偏移，第一个实际数组内的字段值通过指针数组的输出来标识，

指针数组的输出由偏移通过所述的第二个下标转换数组给出，

其中，

提供包含用于控制锁定的数组的信息块，

对应于指出锁定类型的每个所述的记录号，字段值放在所述的信息块的数组内。

表格式数据显示方法、插入方法、删除方法和更新方法

技术领域

本发明涉及数据处理方法和数据处理装置，用于使用计算机或其它信息处理装置处理大量的数据，特别是涉及构成数据库的表格式数据的更新、删除、插入和事务处理。

本发明在处理大量数据的系统中应用，例如，数据库和数据仓库。更具体地，在大规模的科学技术计算、工厂和电力供应的控制系统中应用，以及定单管理和使命关键（如证券交易）事务性工作中。

现有技术说明

数据库可以应用于各种场合，但应用那些能够消除逻辑冲突的关系数据库(RDB)，已经成为大中型系统的主流。例如，RDB 可以用于机票预订系统或类似的系统中。在这种情况下，可以指定关键字段，以对目标执行快速搜索（通常是单个目标），或者，确认、取消或更改预订。

但是，最近几年，对数据库的需求显著地增强，这些应用不仅涉及上文所述的单个数据记录的简单管理（使命关键的工作），而且还涉及分析（基于信息的工作）。例如，前述数据库的所有者可能尝试创建交叉表格，以便分析空座率如何随着时区、航线、季节等变化。

但是，使用通常的 RDB，使用本地存储来提交事务和确认其内容，因此，尽管适合用于主要基于事务处理的使命关键工作，但是却无法用于从结构角度对大规模的数据（基于信息的工作）进行分析。例如，如果希望对任意字段执行排序，理论上，这会花时间对 n*log(n)（有 n 个记录）进行排序，那么，数据的规模越大，处理的效率就越低。（在实际产品中，处理效率的下降速度比理论上还要快）。此外，这种局限（时间 =

n*log(n)）不仅表现在排序中，而且还表现在所有的一般操作中（如对局部集合的提取（= 搜索））。

很自然，有人提出了避免这种局限和提高速度的各种方法，其中一个很好的例子是位映像索引。位映像索引是一种位映像，其中，有关每个记录是否满足条件的信息用单个的位表示。例如，对于“性别”字段，将要创建“男性”位映像和“女性”位映像。此时，如果希望只提取（搜索）“女性”的记录，则可以快速地获得该局部集合（女性记录集）。但是，不可能在任何情况下为每个值创建位映像。例如，在有十亿条记录的数据库中，对于有十亿个不同值的字段，（例如，ID 号码），就必须创建十亿个位映像，每个位映像的大小有十亿位（大约 125 MB）。可以看出，位映像索引只对于那些变化特别少的字段（例如，性别）才切合实际。此外，这种方法很显然无法快速用于局部集合，也无法用于频繁地插入和删除记录的场合。其它提高速度的方法也有这样那样的局限。由于这种局限出现在旨在提高速度的各种方法的可应用的范围内，对于用户来说似乎完全相同的操作，可能在一瞬间完成，也可能花好几个小时，因此，可能发生处理时间相当不连续的严重问题（很难预测）。

例如，如果使用在整个集合中有 1000 万个记录的表，对市场上的 RDB 来测量时间，有报告说，命中所有的记录（= 1000 万个记录）只需要花 1 秒，但只命中 737 个记录却需要花 61 秒。那么，命中 900 万个记录需要花多少处理时间呢？不必说，在执行搜索之前无法知道命中次数，因此，即使知道命中次数和进行搜索所需要的时间之间的对应关系存在某种函数，但是，却无法准确地预测所需要的时间。总之，如果假设提取 500 万个满足是男性这一条件的记录所需要的时间（搜索时间）与提取 500 万个满足是女性这一条件的记录所需要的时间（搜索时间）相同，甚至连这种理论基础都很脆弱。

这种性能的不稳定性对于实时控制系统尤其致命。例如，大规模、复杂的供电系统，其中有许多发电厂通过电网与大量的电力用户连接，或者用于大规模的化工厂的控制系统，这两者都是管理和使用大量的数据但无法使用数据库的系统的例子。

显然，即使它们的性能稳定，如果太慢，也无法使用，但是我们想强调的是，由于引入各种提高速度的方法，数据库变得十分复杂，这些方法只能在特定的有限的条件下使用，或者它们却带来了严重的问题，例如上文所提及的处理时间的严重不一致。

现在，数据库的统计应用的上述方法已经普遍地应用（如上文所述），同时，所处理的数据的量越来越大，这是不争的事实。在这种情况下，数据仓库 (DWH) 和多维数据库 (MDB) 出现了，目的在于克服上述问题。

DWH 可以看作是一种 RDB，在这种数据库中，事务处理功能等已经被删除，对其进行了调整，专门执行搜索和制表功能。相应地，n*log(n) 的处理时间这一基本限制照旧，因此性能并没有得到充分的改善，不连续问题也未得到解决。此外，由于现有的用于执行使命关键工作的 RDB 和用于执行基于信息的工作的 DWH 无法避免地混合，导致对设备的投资越来越大，而且还带来了新的问题，包括信息管理越来越复杂，运行成本越来越高。

另一方面，MDB 可以看作是这样的一个系统，其中，基于指定字段（叫做维）的组合提前准备基础交叉表，用户请求的交叉表从基础交叉表进行合成。MDB 可以令人满意地合成交叉表，但它们也存在许多问题。例如，不可能返回到原始记录（除非执行特殊操作，因为不可能从表返回到原始数据），它还无法处理诸如股票价格之类的随时更新的数据（由于交叉表必须提前准备）；而且还无法在那些提前指定的字段之外的字段（维）上准备交叉表，还有其它许多别的问题。显然，对于 DWH，也存在设备投资增加，信息管理复杂之类的问题。

看看现实，很少引入 DWH 和 MDB，除非在那些不管成本高低而更注重性能的领域，如上述控制系统和科技计算领域，金融工程等领域。显然，它们的功能和性能不能满足在上述领域的应用的需要。

为了解决这些问题，当今的发明家们已经发明了搜索、制表、排序和联接表格式数据的方法（线性筛选器方法），其处理的速度大约是现有技术的 100-1000 倍，并已经申请了专利（日本专利申请号 No. 10-

227278）。线性筛选器方法有许多优越的特性，包括很高而稳定的处理性能、处理的一致性、数据的压缩和可分割性、O(n) 体系结构，以及排序的直接连接并适合管道式和并行处理。

因此，本发明的目的是利用上述的线性筛选器方法，提供插入、删除和更新表格式数据的方法，特别是快速而正确地执行数据的插入、删除和更新。

此外，正如通常所理解的，实际的更新操作需要正确应用排他的控制（其中，控制多个过程对相同的数据进行访问）和事务处理（其中，多个过程作为单个有效处理块提交或回滚）。

为此，本发明还有一个目的，即提供插入、删除和更新表格式数据的方法，其中上述事务处理应正确地执行。

发明内容

本发明的目标通过提供一种插入表格式数据的方法来实现，其中，字段值在表格式数据中的任意位置插入，所述表格式数据用存储记录数组的值来表示，在存储器中包含字段和字段值，其中所述的插入表格式数据的方法包括下列步骤：在所述存储器中生成下标转换数组，这种数组的构成使其能够接受记录号作为下标，并给出对应于所述的下标的范围的偏移值，通过审阅存储数组的所述值中的字段值，标识插入位置，该位置指出要插入的字段值的位置，在所述的下标转换数组中，关于所述的插入位置，给出偏移值，该偏移值定义对应的下标的范围，还标识在所述的数组的末尾后面的规定的位置，在所述的下标转换数组中，关于那些记录号大于对应于所述的插入位置的记录号的记录，给出偏移值，该偏移值向上移动对应的下标范围，还减少能接受的下标，将要插入的所述的字段值放在所述的数组的末尾，这样，根据下标转换数组内的下标的范围，给出偏移值作为所述的下标，通过所述的偏移值给出的下标来标识数组的字段值。

通过本发明，使用下标转换数组来给出下标的所规定的偏移，并标识数组的字段值。相应地，在插入的情况下，通过下标转换数组，对应于

新插入的位置的下标可以标识放在数组的末尾后面所规定位置的新字段值。相应地，数据实际插入在数组中，因此没有必要移动其他数据，因此，可以将字段值逻辑地插入到数组中。此外，还可以如下文所述，执行并行处理和事务处理。

本发明的另一个目标通过提供删除表格式数据的方法来实现，其中，在表格式数据中的任意位置删除字段值，所述表格式数据由记录数组来表示，在存储器中包含字段和字段值，其中所述的删除表格式数据的方法包括下列步骤：在所述存储器中生成下标转换数组，这种数组的构成使其能够接受记录号作为下标，并给出对应于所述所接受的下标的范围的偏移值，通过审阅存储数组的所述值中的字段值，标识删除位置，该位置指出要删除的字段值的位置，在所述的下标转换数组中，关于那些记录号大于对应于所述的删除位置的记录号的记录，给出偏移值，该偏移值向下移动对应的下标范围，该偏移值还增大所接受的下标，这样，根据下标转换数组内的下标的范围，给出偏移值作为所述的下标，通过所述的偏移值给出的下标来标识数组的字段值。

在本发明中，使用下标转换数组，使对应于要删除的位置的字段值不由下标来指示。因此，实现了逻辑数据删除的过程，而不会删除实际数据和移动数据。

此外，删除表格式数据的方法可以通过按顺序或以相反的顺序执行上述的更新方法和删除方法来实现。

此外，在本发明的另外的实施例中，表格式数据的值转换方法转换由记录数组来表示的格式数据的字段值的值，记录 **OVT** 组包含字段，字段里面包含字段值，该方法包括下列步骤：生成值转换数组，这种数组的构成是这样的：给出对应于所述的字段值的范围的偏移值，它接受记录号作为下标，将对应于所述的字段值的范围的偏移值给予对应于所述的数组内的所述下标的字段值。

本发明的另一个目标通过为表格式数据提供数据结构来实现，表格式数据的构成是这样的，对应于任意记录号的字段值由表格式数据来标识，表格式数据由记录数组来表示，包含字段，字段里面包含字段值，其中

所述的表格式数据的数据结构具有这样的特征：所述的表格式数据以这样的方式构成，它分为一个或多个信息块，信息块包括：一个值列表，该列表包含第一个实际数组，在该数组中，字段值以字段值号码的顺序存储，字段值号码对应于字段值，该字段值属于指定的字段，一个指针数组，该数组包含第二个实际数组，在该数组中，用于指向所述的字段值号码的指针值按唯一的记录号码顺序存储。第二个下标转换数组的构成是这样的：输入、输出指针数组，做为所述值列表中的下标，给出并形成对应于所述下标的范围的第三个偏移，第一个实际数组内的字段值通过指针数组的输出来标识，指针数组的输出由偏移通过所述的第二个下标转换数组给出。

通过本发明，例如，某些过程可能通过下标转换数组和值转换数组形成数据流，而另一个过程可能形成数据流，该数据流只使用实际数组。如此，如果有必要，通过设置所述的转换数组的传递和非传递，就可以正确地实现并行处理和事务处理。

例如，每个所述的下标转换数组可以包括：起始位置数组（该数组包括起始位置，这些位置指出每个规定的范围内的下标的最小值），和/或结束位置数组（该数组包括结束位置，这些位置指出所述的规定的范围内的下标的最大值），偏移数组（该数组包括对应的偏移值）。此外，所述的值转换数组可以包括：起始位置数组（该数组包括起始位置，这些位置指出每个规定的范围内的下标的最小值），和/或结束位置数组（该数组包括结束位置，这些位置指出所述的规定的范围内的值的最大值），偏移数组（该数组包括对应的偏移值）。

本发明的另一个目标通过提供插入表格式数据的方法来实现，在该方法中，在任意位置插入字段值，这些字段值是表格式数据，这些数据由上面定义的数据结构来表示，其中，所述的插入表格式数据的方法包括下列步骤：

(1) 关于所述的值列表：

标识插入位置，该位置指出要插入的字段值的位置，在所述的第二个下标转换数组中，关于所述的插入位置，给出第三个偏移值，该偏移值

定义对应的下标的范围，该偏移值还标识所述的第一个数组的末尾后面的规定的位置，在所述的第二个下标转换数组中，关于那些值大于对应于所述的插入位置的下标的记录，给出第三个偏移值（该偏移值向上移动对应的下标范围，以使定义所述的范围的值变大，该偏移值还减少所接受的下标），并将要插入的所述的字段值放在所述的第一个实际数组的末尾后面规定的位置。

(2) 关于所述的指针数组，

标识插入位置，该插入位置指出对应于要插入的记录号的指针值的位置，在所述的第一个下标转换数组中，关于所述的插入位置，给出第一个偏移值，该偏移值定义对应的下标，还标识所述的第一个实际数组末尾后面的规定的位置，在所述的第一个下标转换数组中，关于那些记录号大于对应于所述的插入位置的记录号的记录，给出第一个偏移值（该偏移值向上移动对应的下标范围，以使定义所述的范围的值变大，该偏移值还减少所接受的下标），并将大于现有指针值的新指针值放在所述的第二个实际数组的末尾后面规定的位置。

(3) 关于所述的指针数组，

在所述的值转换数组中，给出第二个偏移值，该偏移值增大那些值大于对应于所述的值列表中的插入位置的指针值的记录，在所述的值转换数组中，给出第二个偏移值，以使所述的新指针值标识对应于所述的插入位置的位置。

在前述的插入表格式数据的方法的优选实施例中：表包括 m 个记录号和 n 个字段值，当所述的值列表内的字段值的插入位置为 i 时（0<=i<=n-1：下文“<=”表示左侧等于或小于右侧，而“>=”表示左侧等于或大于右侧），字段值放于第一个实际数组的末尾 n 处，记录号插入位置为 j（0<=j=m-1），指针值放于第二个实际数组的末尾 m 处，

(1) 在所述的值列表的第二个下标转换数组中：当值位于 (i-1) 或更小范围内时，给出 0 作为第三个偏移值，当值为 i 时，给出 (n-1) 作为第三个偏移值，当值位于 (i＋1) 或更大和 n 或更小范围内时，给出 (-1) 作为第三个偏移值，

(2) 在所述的指针数组的第一个下标转换数组中：当下标位于 (j-1) 或更小范围内时，给出 0 作为第一个偏移值，当下标为 j 时，给出 (m-j) 作为第一个偏移值，当下标位于 (j＋1) 或更大和 m 或更小范围内时，给出 (-1) 作为第一个偏移值，

(3) 在所述的指针数组的值转换数组中：当值位于 (i-1) 范围内时，给出 0 作为第二个偏移值，当值在 i 或更大和 (n-1) 或更小范围内时，给出 1 作为第二个偏移值，当值为 n 时，给出 (i-n) 作为第二个偏移值，

或者，表包括 m 个记录号和 n 个字段值，当所述的值列表内的字段值的插入位置为 i (0<=i<=n-1) 时，字段值放于第一个实际数组末尾后面的规定的位置 z (z>=n) 处，记录号插入位置为 j (0<=j<=m-1) 时，指针值放于第二个实际数组末尾后面的规定位置 x (x>=m) 处，

(1) 在所述的值列表的第二个下标转换数组中：当值位于 (i-1) 或更小范围内时，给出 0 作为第三个偏移值，当值为 i 时，给出 (z-1) 作为第三个偏移值，当值位于 (i＋1) 或更大和 n 或更小范围内时，给出 (-1) 作为第三个偏移值，

(2) 在所述的指针数组的第一个下标转换数组中：当下标位于 (j-1) 或更小范围内时，给出 0 作为第一个偏移值，当下标为 j 时，给出 (x-j) 作为第一个偏移值，当下标位于 (j＋1) 或更大和 m 或更小范围内时，给出 (-1) 作为第一个偏移值，

(3) 在所述的指针数组的值转换数组中：当值位于 (i-1) 范围内时，给出 0 作为第二个偏移值，当值在 i 或更大和 (n-1) 或更小范围内时，给出 1 作为第二个偏移值，当值为 y 时（其中 y 是存储在第二个实际数组的位置 x 处的指针值），给出 (i-y) 作为第二个偏移值。

此外，本发明的另一个目标通过删除表格式数据的方法来实现，其中，在表格式数据中的任意位置删除字段值，所述表格式数据由上面定义的数据结构来表示，其中，所述的删除表格式数据的方法包括下列步骤：关于所述的指针数组，标识删除位置，该删除位置指出对应于要删除的指针值的位置，

在所述的第一个下标转换数组中：关于所述的插入位置，定义对应的下标，在所述的第一个下标转换数组中，关于那些记录号大于对应于所

述的删除位置的记录号的记录，给出第一个偏移值，该偏移值向下移动对应的下标范围，以使定义所述范围的值变小，该偏移值还增加所接受的下标。

在前述的删除表格式数据的方法的优选实施例中：表包括 m 个记录号和 n 个字段值，当记录号删除位置为 j (0<=j<=m-1) 时，在所述的指针数组的第一个下标转换数组中：当下标位于 (j-1) 或更小范围内时，给出 0 作为第一个偏移值，当下标位于 j 或更大和 (m-2) 或更小范围内时，给出 1 作为偏移值，

本发明的另一个目标通过提供更新表格式数据的方法来实现，其中，在表格式数据中的任意位置更新字段值，所述表格式数据由上面定义的数据结构来表示，其中所述的更新表格式数据的方法包括下列步骤：

(A) (1) 关于所述的值列表：

假设要更新的字段值的位置是插入位置，标识插入位置，该位置指出要插入的字段值的位置，在所述的第二个下标转换数组中，关于所述的插入位置，给出第三个偏移值，该偏移值定义对应的下标的范围，还标识所述的第一个数组的末尾后面的规定的位置，在所述的第二个下标转换数组中，关于那些值大于对应于所述的插入位置的下标的记录，给出第三个偏移值（该偏移值向上移动对应的下标范围，以使定义所述的范围的值变大，该偏移值还减少能接受的下标），并将要插入的所述的字段值放在所述的第一个实际数组的末尾后面规定的位置，

(2) 关于所述的指针数组，假设要更新的指针值为要插入的指针值，标识插入位置，该插入位置指出要插入的指针值的位置，在所述的第一个下标转换数组中，关于所述的插入位置，给出第一个偏移值，该偏移值定义对应的下标，还标识所述的第一个实际数组末尾后面的规定的位置，在所述的第一个下标转换数组中，关于那些记录号大于对应于所述的插入位置的记录号的记录，给出第一个偏移值（该偏移值向上移动对应的下标范围，以使定义所述的范围的值变大，该偏移值还减少能接受的下标），并将大于现有指针值的新指针值放在所述的第二个实际数组的末尾后面规定的位置。

(3) 关于所述的指针数组，在所述的值转换数组中，给出第二个偏移

值，该偏移值增大那些值大于对应于所述的值列表中的插入位置的指针值的记录，在所述的值转换数组中，给出第二个偏移值，以使所述的新指针值标识对应于所述的插入位置的位置。

(B) 关于所述的指针数组，将要更新的指针值的位置标识为删除位置，考虑所述的插入位置，在所述的第一个下标转换数组中，关于所述的删除位置，定义对应的下标，在所述的第一个下标转换数组中，关于那些记录号大于对应于所述的删除位置的记录号的记录，给出第一个偏移值，该偏移值向下移动对应的下标范围，以使定义所述的范围的值变小，该偏移值还增大所接受的下标，以及

所述的 (A) 和 (B) 或所述的 (B) 和 (A) 按顺序执行。

在前述的更新表格式数据的方法的优选实施例中：表包括 m 个记录号和 n 个字段值，当所述的值列表内的字段值的更新位置为 i (0<=i<=n-1) 时，字段值放于第一个实际数组末尾 n 处，记录号更新位置为 j (0<=j<=m-1) 时，指针值放于第二个实际数组末尾 m 处，

(1) 在所述的值列表的第二个下标转换数组中：当值位于 (i-1) 或更小范围内时，给出 0 作为第三个偏移值，当值为 i 时，给出 (n-1) 作为第三个偏移值，当值位于 (i + 1) 或更大和 n 或更小范围内时，给出 (-1) 作为第三个偏移值，

(2) 在所述的指针数组的第一个下标转换数组中：当下标位于 (j-1) 或更小范围内时，给出 0 作为第一个偏移值，当下标为 j 时，给出 (m-j) 作为第一个偏移值，当下标位于 (j + 1) 或更大和 (m+1) 或更小范围内时，给出 0 作为第一个偏移值，

(3) 在所述的指针数组的值转换数组中：当值位于 (i-1) 范围内时，给出 0 作为第二个偏移值，当值在 i 或更大和 (n-1) 或更小范围内时，给出 1 作为第二个偏移值，当值为 n 时，给出 (i-n) 作为第二个偏移。

或者，表包括 m 个记录号和 n 个字段值，当所述的值列表内的字段值的插入位置为 i (0<=i<=n-1) 时，字段值放于第一个实际数组末尾后面的规定的位置 z (z>=n) 处，记录号插入位置为 j (0<=j<=m-1) 时，指针值放于第二个实际数组末尾后面的规定位置 x (x>=m) 处，

(1) 在所述的值列表的第二个下标转换数组中：当值位于 (i-1) 或更小范围内时，给出 0 作为第三个偏移值，当值为 i 时，给出 (z-1) 作为第三个偏移值，当值位于 (i + 1) 或更大和 n 或更小范围内时，给出 (-1) 作为第三个偏移值，

(2) 在所述的指针数组的第一个下标转换数组中：当下标位于 (j-1) 或更小范围内时，给出 0 作为第一个偏移值，当下标为 j 时，给出 (x-j) 作为第一个偏移值，当下标位于 (j + 1) 或更大和 (m+1) 或更小范围内时，给出 0 作为第一个偏移值，

(3) 在所述的指针数组的值转换数组中：当值位于 (i-1) 范围内时，给出 0 作为第二个偏移值，当值在 i 或更大和 (n-1) 或更小范围内时，给出 1 作为第二个偏移值，当值为 y 时（其中 y 是存储在第二个实际数组的位置 x 处的指针值），给出 (i-y) 作为第二个偏移值。

或者，上面定义的删除表格式数据的方法包括下列步骤：关于所述的记录号，提供第四个实际数组，在该数组中，放置记录号本身，第三个下标转换数组根据下标的范围给定的规定偏移值用于标识所述的记录号，标识删除位置，该删除位置指出要删除的所述的记录号的位置，在所述的第三个下标转换数组中，定义涉及所述的删除的对应的下标，在所述的第三个下标转换数组中，关于那些值大于对应于所述的删除位置的下标的记录，给出第四个偏移值，该偏移值向下移动对应的下标范围，以使定义所述的范围的值变小，该偏移值还增大所接受的下标。

这涉及构成包含第四个实际数组的信息块（在第四个实际数组中，放置记录号本身）和用于指定记录号的下标转换数组，从而根本删除数据。

此外，本发明的另一个目标通过事务处理方法来实现，该方法具有下列特征，当通过上述方法执行表格式数据的插入、删除和/或更新时，执行回滚或提交，抛弃所述的下标转换数组和值转换数组。在这种情况下，所述的下标转换数组和值转换数组只能在执行回滚的情况下才能抛弃。

此外，本发明的另一个目标通过并行处理方法来实现，其中，对具有上述定义的数据结构的表格式数据同时执行许多过程，其中，在前述的过程中，那些涉及数据的插入、删除和/或更新的过程这样构成以便：在

所述的指针数组中，它们经过第一个下标转换数组、第二个实际数组和值转换数组，在所述的值列表中，它们经过第二个下标转换数组和第一个实际数组，另一方面，在前述的过程中，那些不涉及数据的插入、删除和/或更新的过程这样构成以便：在所述的指针数组中，它们经过第二个下标转换数组，此外，在所述的值列表中，它们经过第一个实际数组。

那些涉及数据的插入、删除和/或更新的过程可以利用前述的插入、删除或更新表格式数据的方法。

或者，在本发明的另一个实施例中，对于具有前述的数据结构的表格式数据的记录锁定方法是这样的：其中，提供包含用于控制锁定的数组的信息块，对应于每个所述的记录号（它们指出锁定的类型）字段值放在所述的信息块的数组内。

此外，本发明的目标还通过计算机可读的记录介质来实现，该介质上记录了用于实现上面所描述的各种方法之一的程序。

附图说明

参照附图以及实施例，本发明的各种目标就会变得一目了然。这里：

图 1 是方块图，说明了能够根据本发明的实施例实现检索、制表和搜索方法的计算机系统的硬件配置。

图 2 是关系图，说明了在本发明的实施例中使用的信息块。

图 3A-3D 是关系图，说明了表格式数据的示例，以及基于此表格式数据的信息块的示例。

图 4 是关系图，说明了表格式数据的另一个示例，以及基于此表格式数据的信息块的示例。

图 5 是流程图，说明了对单个字段进行搜索的方法。

图 6 是流程图，用于描述基于表格式数据创建信息块的过程。

图 7A 和 7B 是关系图，分别说明了用于创建信息块的原始数据的示例。

图 8A 和 8B 是关系图，分别说明了将元素插入到数组中的示例和更新数组的元素的示例。

图 9A 到 9D 是关系图，用于分别描述下标、数组和数组中的值以及根据本发明的下标转换和值转换的概述。

图 10A 到 10C 是关系图，用于描述与将元素插入到数组中（根据实施例 1）相关的逻辑输入/输出关系。

图 11 是流程图，说明根据实施例 1 从下标指定数组内的元素的过程。

图 12 是关系图，用于说明在指定的位置插入特定的元素。

图 13 是流程图，说明在此实施例中为插入元素生成下标转换数组的过程。

图 14 是关系图，说明通过图 13 的过程生成下标转换数组的一个示例。

图 15A 到 15C 是关系图，用于描述与从数组中删除元素（根据实施例 1）相关的逻辑输入/输出关系。

图 16A 到 16B 是关系图，用于说明在实施例 1 中在指定的位置删除特定元素的过程。

图 17 是流程图，说明为在此实施例中删除元素生成下标转换数组的过程。

图 18A 和 18B 是关系图，说明转换数组内的数值如何随图 17 的过程而变化。

图 19 是关系图，用于描述在实施例 1 中按顺序执行插入和删除的情况下的状态。

图 20 是关系图，用于描述在实施例 1 中按顺序执行插入和删除的情况下的状态。

图 21A 到 21C 是关系图，用于描述根据本发明的实施例 2 的转换数组的结构。

图 22 是关系图，用于描述根据本发明的实施例 3 的实际数组和差异数组。

图 23A 和 23B 是关系图，用于描述根据本发明的实施例 4 的值转换数组的概述。

图 24 是流程图，说明根据本发明的实施例 4 的值转换的过程。

图 25 是流程图，说明根据本发明的实施例 4 生成值转换表的过程。

图 26A 和 26B 是关系图，用于描述在本发明的实施例 4 中在实际数组内执行值转换的情况下的状态。

图 27 是关系图，说明了根据本发明将下标转换数组和值转换数组结合的数据结构的一个示例。

图 28A 到 28D 是关系图，说明了在本发明的实施例 5 中各种信息块的结构。

图 29A 和 29B 是关系图，用于描述在本发明的实施例 5 中删除特定记录的过程。

图 30A 和 30B 是关系图，用于描述在本发明的实施例 5 中删除特定记录的过程。

图 31A 和 31B 是关系图，用于描述在本发明的实施例 5 中删除特定记录的过程。

图 32 是关系图，用于描述在本发明的实施例 5 中删除特定记录的过程。

图 33A 和 33B 是关系图，用于描述在本发明的实施例 5 中将特定记录插入到特定位置的过程。

图 34 是关系图，用于描述在本发明的实施例 5 中将特定记录插入到特定位置的过程。

图 35A 和 35B 是关系图，用于描述在本发明的实施例 5 中将特定记录插入到特定位置的过程。

图 36A 和 36B 是关系图，用于描述在本发明的实施例 5 中将特定记录插入到特定位置的过程。

图 37A 和 37B 是关系图，用于描述在本发明的实施例 5 中将特定记录插入到特定位置的过程。

图 38A 和 38B 是关系图，用于描述在本发明的实施例 5 中将特定记录插入到特定位置的过程。

图 39A 和 39B 是关系图，用于描述在本发明的实施例 5 中将特定

记录插入到特定位置的过程。

图 40A 和 40B 是关系图，用于描述在本发明的实施例 5 中将特定记录插入到特定位置的过程。

图 41 是关系图，用于描述在本发明的实施例 5 中将特定记录插入到特定位置的过程。

图 42 是关系图，用于描述在本发明的实施例 6 中删除特定记录的过程。

图 43 是流程图，说明利用本发明的事务处理。

图 44 是关系图，用于描述利用本发明对许多过程进行并行处理。

图 45 是关系图，用于描述利用本发明的排他控制。

图 46A 和 46B 是流程图，用于描述在本发明中以信息块的形式对表格式数据进行数据插入的过程。

图 47 是关系图，用于描述在本发明中以信息块的形式对表格式数据进行数据插入的过程。

图 48 是关系图，用于描述在本发明中以信息块的形式对表格式数据进行数据删除的过程。

图 49 是关系图，用于描述根据本发明的下标转换数组的另一个实例。

图 50 是关系图，用于描述根据本发明的值转换数组的另一个实例。

图 51 是关系图，用于描述根据本发明的下标转换数组和值转换数组的另一个实例。

图 52 是关系图，用于描述根据本发明以信息块的形式对表格式数据进行数据插入的过程的另一个实例。

图 53 是关系图，用于描述根据本发明以信息块的形式对表格式数据进行数据更新的过程。

图 54 是关系图，用于描述根据本发明以信息块的形式对表格式数据进行数据更新的过程的另一个示例。

具体实施方式

下面参照附图对本发明的实施例进行说明。图 1 是方块图，说明了能够根据本发明的实施例实现插入、删除和更新表格式数据的某些字段的方法的计算机系统的硬件配置。如图 1 所示，此计算机系统 10 与普通的计算机具有相同的配置，包括 CPU 12（它通过执行程序来控制整个系统以及其单个组件）、随机存取存储器 (RAM) 14（它存储工作数据等）、只读存储器 (ROM) 16（它存储程序等）、硬盘或其他固定存储设备 18、用于存取 CD-ROM 19 的 CD-ROM 驱动器 20、为 CD-ROM 驱动器 20 提供的接口 (I/F) 22、连接到外部网络的外部终端（未显示出）、键盘、鼠标和其他输入设备 24 和 CRT 显示设备 26。CPU 12、RAM 14、ROM 16、外部存储介质 18、接口 22、输入设备 24 和 CRT 显示设备 26 通过总线 28 相互连接。

根据实施例，用于在表格式数据的某些字段插入数据的程序、在某些字段删除数据的程序、在某些字段更新数据的程序、实现事务处理的程序、控制并行处理的程序以及用于实现排他控制的程序可以放在 CD-ROM 19 上，并由 CD-ROM 驱动器 20 读取，或者预先存储在 ROM 16 中。此外，一旦从 CD-ROM 19 中读取，程序还可以存储在外部存储介质 18 的特定区域。或者，前述的程序还可以通过网络（未显示）、外部终端或接口 22 从外部提供。请注意，在此说明中，插入数据是指在相邻记录之间插入数据和在记录的末尾处添加新记录。

此外，在本实施例中，为了正确而快速地实现数据的插入、删除和更新以及事务处理、并行处理和排他控制（如下文所述），必须生成规定数据格式的信息块。这种信息块生成程序也可以同样包含在 CD-ROM 19 中、存储在 ROM 16 中，或存储在外部存储介质 18 中。或者，这些程序还可以自然地通过网络（未显示）从外部提供。此外，在本实施例中，由前述的信息块生成程序（该程序生成信息块）生成的数据（信息块）存储在 RAM 14 或外部存储介质 18 的特定区域。

接下来，我们将简要地描述表格式数据的处理（这是本发明所必备的），也就是对表格式数据进行管理的形式。为了实现对表格式数据的高速搜索、制表和排序，为了按所需要的方式将许多表格式数据的表联接

起来，当今的发明家想出了具有特定数据格式的表格式数据的结构（日本专利申请 No. 10-227278）和将具有所述的特定的数据格式的表格式数据联接起来的方法（日本专利申请 No. 10-151156）。在本发明中，基本上基于在这两个申请中公开的技术将表格式数据构造为规定的信息块集，并使用它们执行搜索、制表、排序。

图 2 是关系图，说明在本发明中的实施例中使用的信息块。如图 2 所示，信息块 100 包含值列表 110 和指向值列表的指针数组 120。值列表 110 是一个表格，对于每个表格式数据的字段，该表格包含按照字段值号码顺序对应于字段值号码的字段值 111，这些字段值号码基于属于该字段的字段值进行分配（整体分配）。指向值列表的指针的数组 120 是一个数组，它包含以表格式数据表示的列（即字段）的字段值号码，或者以表格式数据记录号顺序表示的指向值列表 110 的指针。

通过将前述指向值列表的指针数组 120 和值列表 110 结合，给定某个记录号，对于某个字段，可以从指向值列表 120 的指针数组中获得对应于该记录号存储的字段值号，然后获得对应于值列表 110 内的该字段值号存储的字段值，从而通过记录号获得字段值。相应地，则可以采用与常规数据表相同的方式使用记录号（行）和字段（列）引用所有的数据（字段值）。

例如，看如图 3A 所示的表格式数据。在此例中，为客户 ID、客户名称和电话号码字段给出各种字段值。在此实施例中，这样的表格式数据存储为具有图 3B 到 3D 中指出的格式的信息块。例如，在图 3B 中，指针数组 120-1 与值列表 110-1 （该值列表包含指出客户 ID 的字段值）相关联。即，第一个记录（记录号为“0”）的指针数组的指针值为 0，则可以获得对应的指出客户 ID 的字段值 1。在图 3C 中，指针数组 120-2 与值列表 110-2（该值列表包含指出客户名称的字段值）相关联。例如，第一个记录（记录号为“0”）的指针数组的指针值为 5，则可以获得对应的指出客户名称的字段值“Smith”。此外，在图 3D 中，可以看出，指针数组 120-3 同样与值列表 110-3（该值列表包含指出电话号码的字段值）关联。此外，还可以看出，在每个值列表中，字段值按连

续的顺序应用（在本例中为升序）。

此外，在本实施例中，信息块 100 的值控制表包含值列表 110，并附有用于搜索和制表的类别号码标志数组、起始位置数组（该数组指出存储对应于字段值的指针的内存空间的起始地址），以及计数数组。类别号码标志数组的各种标志和计数数组的计数都与字段值关联。类别号码标志数组的标志值通常为“0”，但在搜索和制表时，都对应于要查找的字段值设置为“1”。此外，计数对应于具有该字段值的记录数量。请注意，起始位置对应于通过加上小于对应的指针值的指针值的计数所得出的那一个位置，不一定非得提供。

图 4A 是关系图，说明表格式数据的另一个示例，而图 4B 和 4C 是关系图，说明有关“性别”和“年龄”的信息块。如图 4B 所示，有关性别 200-1 的信息块的值控制表 210-1 包含对应于指针数组 220 的各种指针值（“男性”和“女性”）的字段值，和对应于各种字段值的类别号码、起始位置和计数。例如，指针值为“0”（即，值列表中的字段值为“男性”）的记录的数量是 632564，另一方面，指针值为“1”（即，值列表中的字段值为“女性”）的记录的数量是 367436。此外，对应于各种字段值的起始位置指出指向记录 230-1（下文介绍）的指针数组中的起始地址。可以看出，对于图 4C 也是同样的道理。

下面介绍使用具有这样的数据结构的信息块进行搜索以及生成信息块的过程的示例。图 5 是说明对单个字段进行搜索的方法的流程图。该过程由 CPU 12（参见图 1）执行规定的搜索程序来实现。在本例中，我们将搜索“年龄”字段的值为 16 或 19 的记录。首先，在有关表格式数据的信息块中，说明了有关图 4C 中所示的“年龄”200-2 的信息块（步骤 501）。

接下来，在这样说明的信息块（下文称“说明信息块”）的值列表 210-2 中，“1”设置为字段值匹配前述的搜索条件（16 或 19）的行的类别号码（步骤 502）。在本例中，“1”设置为对应于字段值号码“0”或字段值“3”的行的类别号码。接下来，获得对应于类别号码设置为“1”的行的起始位置和计数（步骤 503）。此信息叫做指针获取信息。在指向与

搜索条件匹配的记录的指针的数组中，基于在步骤 503 中获得的指针获取信息，获得指出指向与搜索条件匹配的记录的指针的记录号（记录匹配搜索条件）（步骤 504）。在本例中，可以看出，对应于字段值号“0”的记录指针，存储在指向记录的指针数组的区域，从起始位置“0”，即开始，到足以包含 45898 个指针的位置。另一方面，对应于字段值号码“3”的记录指针存储在指向记录的数组的区域中，从第 2383137 个指针，到足以包含 189653 个指针的位置。最后，为了使其用于后续处理，作为结果集创建获取的记录号数组，并存储起来（步骤 505）。

此外，还可以通过利用类别号、起始位置和计数，来实现制表和排序。

下面介绍生成用于前述的搜索过程的信息块的过程。图 6 是流程图，用于描述基于表格式数据创建信息块的过程。首先，系统 10 获取表格式原始数据，并按字段对其进行分类（步骤 601）。此原始数据可以如图 7A 所示的那样，或者如图 7B 所示的那样。这些原始数据可以从外部提供，或者可以存储在外部存储介质 18 上。包括步骤 602 到 604 的处理块 610（以后描述），生成有关单字段的信息块。相应地，当生成有关许多字段的信息块时，执行对应于处理块 610 的过程，执行的次数与字段的数量相同。下面举例说明生成有关“性别”的信息块的过程。

首先，在 RAM 14 中为有关“性别”的字段的信息块分配一个区域（步骤 602）。接下来，在此分配的区域中生成一个值控制表。更具体地，初始化该值控制表。接下来，在原始数据内，对有关“性别”的所有数据进行处理，从开始到结束，以查找出哪一种字段名称存在多少实例。在此示例中，字段名“女性”和“男性”分别找到了 367436 和 632564 次。因此字段值“女性”和“男性”在值列表中设置，相应的号码也在“计数”列中设置。此后，根据相应的标准对字段值进行排序。在排序时，在记录字段号码的同时还记录计数。接下来，确定“起始”位置列的值。这些值通过求位置在排序中高于那一行的“计数”的和取得。此外，将“起始”位置列的值分配到对应于“分类”号码列的值。此值在下一步骤中使用。在采用这种方式生成值控制表之后，生成指向记录的指针数组。此指针数组的区域的大小对应于计数的总和。

如此，可以为相应的字段创建信息块。通过提前生成信息块，这样生成的信息块用于执行搜索、制表和排序的过程。

接下来，我们将介绍在前述形式的信息块中插入、删除或更新元素的过程。

例如，考虑这样一种情况：在大小为 100 万个记录的数组（记录号 0-999,999）中的开始处的第二个位置插入一个元素。例如，有这样的情况：在对应于记录号 0-999,999 的数组中的记录号“0”和“1”之间插入一个新元素（参见图 8A）。在这种情况下，在新记录号“1”处插入新值（例如，“10”），其余的值（即，所有 999,999 个值）都必须下移一个位置。此外，在删除特定的元素的情况下，必须移动大量的元素。或者，在前述的数组中重写具有特定值的元素的情况下，例如，要增大那些值为 1 或更大的元素（参见图 8B），则必须扫描整个数组，这样就会花费大量的时间。

顺便说一句，前述的数组具有这样的优点：下标指出数组内的位置，因此，可以提取出存储在对应位置的值（即，从下标引用存储的值）。即，如果各种元素（存储值）在内存中按地址顺序存储，那么，从下标可以立即计算出地址，并可以获得对应地址的扩展值。为此，需要这样的技术：既要尽可能地减少插入和删除元素（存储值）所需要的负载，又要保持这样优点。

根据本发明，如图 9A 所示，在系统中，为数组提供下标，以获取对应的值（存储值），如图 9B 到 9D 所示，通过提供下标转换器（它将所给的下标转换成数组）和/或值转换器（它转换从数组获取的值），另外，还将数组本身分为实际数组和差异数组，则可以进行元素的插入、删除和更新，而不会改变包含在数组内的元素（存储值）的位置，也不会改变值本身。

[通过下标转换数组插入元素]

首先，我们将介绍用于插入元素等的下标转换数组（这在本发明的实施例 1 中使用）。设想如图 2-4 所示的信息块，下标可以包括记录号（用于指定指向值列表的指针的数组的元素（指针值）），或者包括指针数组

内的指针值（用于指定值列表的字段值）。此外，数组可以包括指向值列表的指针的数组、值列表和指向记录号的指针的数组。

例如，如图 10A 所示，考虑在第 0 行（下标“0”）和第 1 行（下标“1”）之间插入元素“Y_0”这种情况。在这种情况下，在插入元素“Y_0”之后，下标和数组内的元素（存储值）具有图 10B 所示的逻辑关系。

为了保持图 10B 所示的逻辑关系，同时又要尽可能地减少由插入引起的负载，考虑如图 10C 所示的下标转换数组，在图 10C 中，下标转换数组包括起始位置数组、结束位置数组和偏移数组。在这些数组内包含的元素（值）中，那些位于相同位置（例如，第 0 行、第 1 行）的元素相互关联。要关联的下标的范围由起始位置数组内的元素和结束位置数组内对应的元素来定义。即，起始位置数据内的元素指出要关联的下标的最小值，而结束位置数组内的元素指出要关联的下标的最大值。另一方面，偏移数组内的对应的元素指出偏移值，用于获得转换后的下标（该下标加到或从下标中相减，以指定真正的实际数组）。注意，在本说明中，起始位置数组、结束位置数组和偏移数组内的一组对应的元素（即，位于相同行中的一组元素）叫做“结构数组”，视情况而定，各种数组内的元素叫做“结构成员”。

下面介绍这种下标转换数组的意义。如上文所述，结构数组内的结构成员决定下标的范围和此范围内包括的下标的偏移值。因此，对于某下标，通过识别要与该下标关联的结构数组，并与偏移值相加，可以获得转换后的下标，这些转换后的下标可以用于识别数组（实际数组）的位置，并查找数组内的元素（存储值）。

例如，在图 10C 中，可以看出，下标“0”与行 0（第 1 行）的结构数组关联。在此结构数组中，偏移数组内的成员值（偏移值）为“0”，因此转换后的下标为“0+0” = 0。相应地，可以看出，实际数组内的对应于原始下标“0”的元素存储在位置“0”。此外，下标“1”与行 1（第 2 行）的结构数组关联，偏移数组内的成员值（偏移值）为“n-1”，因此，转换后的下标为“1+(n-1)=n”。相应地，可以看出，实际数组内的对应于原始下标“0”的元素存储在位置“0”。此外，对于下标“2”到“n”，

很显然，可以采用这种方式识别对应的元素所存储的位置。

更具体地，我们将参照流程图 11 介绍前述的过程。首先，初始化指出下标的位置的号码（存储位置号码）（步骤 1101），接下来，查找起始数组和结束数组的元素，根据由起始位置数组和结束位置数组的值划界的范围，识别包含下标的范围（步骤 1102）。即，此步骤识别下标所关联的结构数组。例如，在图 10C 的示例中，可以看出，下标“0”包含在由起始数组内的元素“0”和结束数组内的元素“0”标识的范围内。此外，还可以看出，下标“1”包含在由起始数组内的元素“1”和结束数组内的元素“1”标识的范围内。

接下来，基于这一范围，获取对应的偏移值（步骤 1103），获得将偏移值与下标相加所得的值（步骤 1104）。与偏移相加之后所得的值变成转换后的下标，因此，可以使用这一值来识别实际数组内的元素（步骤 1105）。例如，有关下标“1”，此下标包含在对应于行 1（第 2 行）的范围内，因此，将行 1 的偏移值“n-1”与下标“1”相加。相应地，转换后的下标变成“n”，因此可以获得“Y_0”作为实际数组内的对应元素（存储值）。此外，有关下标“2”，此下标包含在对应于行 2（第 3 行）的范围内，因此，将行 2 的偏移值“-1”与下标“2”相加。相应地，转换后的下标变成“1”，因此可以获得“X_1”作为实际数组内的对应元素（存储值）。

通过对所需要的下标重复步骤 1102 到 1105 的过程（参见步骤 1106 和 1107），可以保持图 10B 所示的逻辑关系，同时又能获得实际数组内的元素（存储值）。相应地，通过本发明，在将元素（存储值）插入数组内时，通过生成如上文所述的转换数组，可以插入元素，而不必在数组内移动元素本身，在实际数组内的规定位置（例如，对应于地址的最大值的位置）组织要插入的元素，而不会引起过多的负载。

[为进行元素插入生成下标转换数组]

接下来，我们将介绍根据实施例 1 创建前述的下标转换数组的过程。例如，如图 12 所示，设想有 5 个元素的数组（符号 1201），设想在行 0（第 1 行）和行 1（第 2 行）之间插入新元素（存储值）“Y_0”的情况

（参见符号 1202）。图 13 是流程图，用于描述为在数组中插入新元素（存储值）生成转换数组的过程。此过程由图 1 中的 CPU 12 来执行。首先，生成初始化的下标转换数组（步骤 1300）。更具体地，检查数组中的元素的数量，并确定起始位置和结束位置。例如，在图 12 所示的示例中，元素的数量是“5”，在初始的下标转换数组中，起始位置是“0”，结束位置是“4”。此外，在本例中，数组中的元素直接从下层引用，因此，初始偏移值为“0”（参见图 14 的符号 1401）。注意，在提前生成下层转换数组的情况下，即在执行一次或多次插入的情况下，或者在生成信息块时生成初始下标转换数组的情况下，忽略步骤 1300。

接下来，识别要插入的元素的位置（步骤 1301），为插入位置和有关插入位置之前的元素和插入位置之后的元素的下标转换数组分配区域。在图 14 的符号 1401 的示例中，包含在初始下标转换数组内的各种数组每个都只包含一个元素，但在步骤 1301 的处理过程中，为具有三个元素（插入位置之前的元素、插入位置的元素、插入位置之后的元素）的数组分配区域（步骤 1302）。请注意，如果已经提供具有许多元素的数组，则为元素的数量可以增加 2 的数组分配区域。

此后，有关插入位置之前的元素，确定每个数组的元素的值（步骤 1303）。更具体地，对应于起始位置数组的元素值和对应于偏移数组的元素值不发生更改，但给出“对应于插入位置-1 的值”作为结束位置数组的元素值。此外，有关插入位置，确定每个数组中的对应的元素的值（步骤 1305 和 1304）。在这种情况下，给出“对应于插入位置的值”作为起始位置数组和结束位置数组的元素值，给出“对应于实际数组的末尾的位置的值”作为偏移数组的元素值。请注意，如下文所述，如果要插入的元素将要在不同的位置设置，则可以给出对应于要插入的位置的值作为偏移值。

此外，有关插入位置之后的元素，各种数组的元素值通过下列技术来确定。即，给出“对应于插入位置+1 的值”作为对应于起始位置数组的元素值（步骤 1306），给出“对应于结束位置+1 的值”作为对应于结束位置数组的元素值（步骤 1307）。接下来，在“其初始值（原始值）-1”

中设置对应于偏移数组的元素值（步骤 1308）。

例如，在有如图 12 的符号 1201 所示的下标和数组的情况下，生成图 14 的符号 1401 所示的初始下标转换数组，可以理解，由前述的步骤 1301 和步骤 1307 的过程来生成符号 1411 所示的下标转换数组。

在生成这样的下标转换数组之后，CPU 12 将要插入的元素存储在步骤 1304 设置的实际数组的末尾之后的规定位置。因此，可以有效地（逻辑地）在实际数组的所需要的位置插入元素，而不必移动实际数组的元素（存储元素）。

[通过下标转换数组删除元素]

接下来，我们将介绍使用前述的下标转换数组删除元素的方法。例如，如图 15A 所示，设想删除数组的行 1（第 2 行）的元素的情况。在这种情况下，删除之后，数组内下标和元素（存储值）具有如图 15B 所示的逻辑关系。与插入过程相同，为了保持图 15B 所示的逻辑关系，同时又要尽可能地减少由删除产生的负载，设想如图 15C 所示的下标转换数组。在图 15C 中，与 10C 相同，下标转换数组定义下标和偏移值的范围，这样可以得出转换后的下标。更具体地，对于每个下标，可以看出，通过执行图 11 的过程来识别实际数组内的元素（存储值）。例如，可以看出，下标“1”与行 1（第 2 行）的结构数组相关联，与此结构数组关联的偏移数组内的成员值（偏移值）为“1”，这样转换后的下标“1+1=2”。相应地，可以看出，对应的元素存储在对应于实际数组中的“2”的位置。

[为进行元素删除生成下标转换数组]

例如，如图 16A 所示，设想有 5 个元素的数组（符号 1601），并设想删除行 1（第 2 行）中的元素（存储值）X_0 的情况（参见符号 1602）。请注意，在本例中，假设实际数组与数组 1601 相同。

图 17 是流程图，用于描述为从数组中删除元素（存储值）生成转换数组的过程，而图 18A 是关系图，用于说明转换数组内的值如何随处理过程而变化。

图 17 的过程也由图 1 所示的 CPU 12 来执行。首先，如果没有生成下标转换数组，则生成初始化的下标转换数组（步骤 1700）。例如，

在图 16 A 所示的示例中，生成起始位置数组、结束位置数组和偏移数组，以使元素的数量为“5”，指示起始位置的值为“0”，指示结束位置的值为“4”，偏移值为“0”（参见图 18A 的符号 1801）。

接下来，识别要删除的元素的位置（步骤 1701），针对删除位置之前的元素为下标转换数组分配区域。在图 18A 所示的示例中，如符号 1802 所示，获得一个区域，以使下标转换数组的各种数组存储插入位置之前、插入位置和插入位置之后的元素。在新区域，针对删除位置之前的元素，“指示删除位置 -1 的值”设置为结束位置数组的元素（步骤 1703），而“指示删除位置 +1 的值”设置为起始位置数组的元素（步骤 1704）。如此，关于包含删除位置的转换数组内的结构数组，如符号 1802 所示的那样，创建起始位置数组、结束位置数组和偏移数组。请注意，有关删除位置的结构数组最终不使用，因此，尽管必须计算删除位置本身，但没有必要在数组内存储元素，如图 18A 所示。

接下来，对于各种数组，有关删除位置之后的元素，执行下面的操作。首先，将起始位置数组的元素设置为“原始值-1”（步骤 1705），还将结束位置数组的元素设置为“原始值-1”（步骤 1706）。此外，将偏移数组的元素设置为“原始值+1”（步骤 1707）。例如，在图 18A 所示的示例中，位于删除位置之后的起始位置数组内的元素“2”减少，变成“1”（参见符号 1804），结束位置数组内的元素“4”也减少，变成“3”，偏移数组内的“0”增大，变成“1”。如此，可以为删除数组内的元素获得下标转换数组。请注意，甚至在删除元素时，实际数组（实际包含由下标标识的元素的数组）不会发生变化。即，只由下标转换数组执行在数组内的元素虚拟（逻辑）删除。

此外，对于图 16B 所示的其他示例，由图 17 的过程来删除数组内的元素。在此示例中，将元素“Y_0”插入数组的行 3（第 4 行）。相应地，实际数组的形式是，元素“Y_0”添加到末尾（参见图 16 的符号 1610）。此外，在前述的插入中，生成图 18B 的符号 1811 所示的下标转换数组。在这种情况下，不生成步骤 1700 所示的初始转换数组。通过对图 18B 的符号 1811 所示的下标转换数组执行步骤 1701 到步骤 1707 的过程，

可以看出，可以获得符号 1813 所示的下标转换数组。此外，可以看出，通过使用这样的下标转换数组，可以使用通过从下标（转换后的下标）获得的下标，来给出图 16 的符号 1612 所示的逻辑关系，而不会改变实际数组（图 16 的符号 1610）。

[同时插入和删除元素]

通过按顺序执行图 13 所示的前述的插入和图 17 所示的删除，可以在数组内进行元素的插入和删除。很显然，不必说，前述的处理顺序也可以颠倒过来。例如，设想数组 1901，其中，最初分别为下标“0”到“4”定义元素“0”、“100”、“200”和“400”，如图 19 所示。设想这样的情况，在此数组的行 2（第 3 行）和行 3（第 4 行）之间插入元素“201”，还删除行 1（第 2 行）的元素，以获得符号 1902 所示的数组。

在此实施例中，如图 20 所示，首先创建初始转换数组（参见符号 2001），然后，执行图 13 所示的过程，来为插入创建转换数组（参见图 20 的符号 2002），并将要插入的元素（本例为“201”）放在实际数组的末尾（参见符号 2003）。接下来，执行图 17 所示的过程，为删除创建转换数组（参见图 20 的符号 2004）。因此，此过程实现元素插入和删除。

请注意，可以看出，按顺序执行元素删除并在此被删除元素的位置插入新元素的过程，可以更新元素。在图 20 的右侧的表中（符号 2011 - 2013），存储位置是通过下标转换数组获得的下标，存储位置指出虚拟（逻辑）实际数组。最初，下标与存储位置相同（参见符号 2011），但是，通过插入和删除，创建或更新下标转换数组，因此，可以看出，按所希望的那样插入元素并从虚拟实际数组中删除（参见符号 2012 和 2013）。

[下标转换数组的其他示例]

下面是实施例 2，介绍缩小下标转换数组的大小的示例。在前述的实施例 1 中，使用起始位置数组和结束位置数组，来确定包含要删除或要插入的元素的下标或位置的范围。但是，如果在结束位置数组的某元素处的元素的值为“n”，那么，位于下一行的起始位置数组的元素的值显然是“n+1”，反之亦然。因此，在实施例 2 中，下标转换数组只包含起

始位置数组和偏移数组或结束位置数组和偏移数组，这样，下标转换数组所需要的内存的大小就会缩小。

例如，图 21A 所示的下标转换数组也可以如图 21B 那样表示。在图 21 B 中，范围 0<=i<=(3-1) 内的下标“i”对应于行 0（第 1 行），因此，这样的下标的偏移值变成“0”。此外，范围 2<=i<=(4-1) 内的下标“i”对应于行 1（第 2 行），4<=i 范围内的下标“i”对应于行 2（第 3 行）。

或者，很显然，也可以使用图 21C 中所示的下标转换数组。在这种情况下，范围 5>=i>=(3+1) 内的下标“i”对应于行 2（第 3 行），这样，范围 3>=i>=(2+1) 内的下标“i”对应于行 1（第 2 行），2>=i 范围内的下标“i”对应于行 0（第 1 行）。

请注意，如果使用图 21B 所示的下标转换数组，则可以确定下标的总数量，因此可以确定数组的大小，因此使用此下标转换数组更好。

[实际数组和差异数组]

在前述的实施例 1 中，当要将元素插入到数组中时，要插入的元素放在数组的末尾（例如，参见图 10C）。但是，显然，通过在偏移数组内设置对应的偏移值，可以将要插入的元素放在所希望的位置。在实施例 3 中，这种插入的元素放在任意的编辑工作区。这样的用于编辑的工作区（将要插入的元素放在该工作区）叫做“差异数组”。

图 22 是关系图，说明下标转换数组、实际数组和差异数组的一个示例。如图 22 所示，在此示例中，从下标转换数组 2201 中，插入的元素“Y_0”和“Y_1”放在差异数组 2203 内，与实际数组 2202 分开。在创建下标转换数组之后（参见图 13），根据要插入的元素的偏移值在该区域创建差异数组，然后将前述的元素放在所述的差异数组中，即可实现这一点。

[值转换数组]

下面根据本发明的实施例 4 介绍值转换数组。有关该数组，必须进行前述的元素插入和删除，还要更改（更新）数组本身的元素（存储值）。在此实施例中，下文所描述的值转换数组用于更新数组的元素。

例如，如图 23A 所示，设想这样的情况：必须增大某数组（记录号列表）内那些值为 1 或更大的记录。若采用常规技术，对于数组内的每个元素，必须确定数组内的元素（存储值）为 1 还是更大，若为 1 或更大，则增大该元素（存储值）。相比之下，本发明的实施例 4 采用这样的技术：将值转换数组放在数组的最后部分（下游部分），将从数组中输出的值传递到值转换数组，以提供转换后的值的输出，在此转换后的输出中实现元素（存储值）的更新。

如图 23A 所示的情况，提供如图 23B 所示的值转换数组。在此实施例中，与实施例 2 相同，值转换数组包括结束位置数组和偏移数组。显然，值转换数组还可以包括如实施例 1 那样的结束位置数组和偏移数组。

在存储在这些数组内的元素（值）中，那些位于相同位置（例如，行 0、行 1）的元素相互关联。相应地，结构数组包括放在对应位置的元素。在此实施例中，可以基于结束位置数组的值确定哪个结构数组与实际数组内的元素（存储值）所在的位置相关联，然后据此确定值的偏移。下面使用图 24，详细描述前述的示例（增大一个或多个元素）的处理过程。该过程还是由图 1 中的 CPU 执行，与上文所述的实施例相同。

首先，初始化指示元素的存储位置的号码（存储位置号）（步骤 2401），然后，检查由此存储位置号标识的数组内的元素（存储值），以确定它包含在由结束位置数组划界的哪个范围内（步骤 2402）。此步骤确定该元素与哪个结构数组关联。在图 23B 所示的示例中，存储位置号码为“0”的元素（即，行 0（第 1 行）的元素）为“1”，因此，此元素包含在结束位置数组内行 1（第 2 行）的值定义的范围内（即，1<=数组的元素<=1），因此，可以看出，该元素与行 1（第 2 行）的结构数组相关联。

接下来，获取此结构数组所属的偏移数组内的偏移值（步骤 2403），将输出偏移值与元素相加所得的值（步骤 2404）。如此，通过值转换数组获得的值可以存储在对应于输出值数组的存储位置号码的区域内（步骤 2405）。不必说，通过值转换数组获得的值还可以由另一个过程直接

利用。有关前述的存储位置号码为“0”元素，对应的偏移值为“1”，因此输出“1+1=2”作为转换数组后面的元素（存储值）。

通过对实际数组中的所需要的值重复步骤 2402 到 2405 的过程，可以在更新元素时保持图 23A 所示的逻辑关系，而不必更改实际数组的元素本身。

[生成值转换数组]

下面使用图 25 的流程图，介绍创建前述的值转换数组的过程。在图 25 的示例中，与图 23B 所示的相同，生成值转换数组（用于将规定的号码与等于或大于某值相加）。请注意，此过程还由图 1 所示的 CPU 来执行。

首先，如有必要，生成初始值转换数组（步骤 2500）。此步骤只有在没有生成值转换数组的情况下才执行。有关图 23B 所示的实际数组 2301，如图 26A 所示，获得初始值转换数组，在初始值转换数组 2601 中，分配实际数组内的元素的最大值作为结束位置数组内的元素，分配“0”作为偏移数组内的元素。

接下来，基于给出某偏移的元素的范围（存储值），确定要创建的数组中的行的数量。例如，如图 23A 所示，当给出将“1”与那些值为“1”或更大的元素相加的条件时，结束位置数组和偏移数组中的行数变成“2”。这是因为，将要生成值小于“1”的元素的行（或结构数组）和值大于“1”的元素的行（或结构数组）。

此后，每个范围内的最大值放在对应于结束位置数组的行中，其偏移值也放在对应于偏移数组的行中（步骤 2502 和 2503）。在如图 23A 所示的实际数组中，如果将“1”与值为“1”或更大的元素相加，小于 1 的值中的最大值“0”放在结束位置数组的行 0（第 1 行）中，偏移值“0”放在偏移数组的行 0（第 1 行）中。即，给出行 0 的结构数组的数值。另一方面，在结束位置数组中，在行 1（第 2 行）中给出实际数组内的元素的最大值“1”。这意味着，可以使用值转换数组的初始值。此外，偏移值“1”放在偏移数组的行 1（第 2 行）中（参见图 26A 的符号 2602）。

通过根据条件（参见步骤 2504 和 2505）对整个范围执行前述的步骤 2502 和 2503，可以根据值转换条件获得值转换数组。

请注意，在前述的实施例中，从单个条件形成两个转换数组的区域，但这不是限制，因为，很显然，可以基于许多条件创建值转换数组（例如，如果 0<=值<=1 为真，则将“1”与该值相加，如果 1<=值<=4 为真，则将“-1”与该值相加，对于其他元素，不给出偏移）（参见图 26B 的符号 2611 和 2612）。

前述的下标转换数组和值转换数组可以结合起来使用。例如，在图 27 中，在实际数组的上游部分（输入部分）（参见符号 2700）（在该部分从“0”到“99”的元素反复出现），提供下标转换数组（参见符号 2701）（该数组将偏移“0”应用到从“0”到“500,000”的下标，将偏移“1”应用到从“500,000”到“999,999”的下标）。在实际数组的下游部分（输出部分）提供值转换数组（符号 2702）（如果值为 49 或更小，则应用偏移“-1”，如果值为 50 或更大，则应用偏移“1”），因此，可以将下标传递到下标转换数组、实际数组和值转换数组，以获得输出值。

还可以使用前述的下标转换数组来转换给予某信息块内的实际数值的下标，通过转换后的下标获取实际数组内的元素（字段值），还可以使用前述的值转换数组来转换获取的元素（字段值），因此，这可以看作包括信息块的输入（下标）和输出（转换后的值）虚拟（逻辑）数组。

[对于特定的表格式数据删除数据]

下面介绍插入、删除或更新实际表格式数据的过程。在实施例 5 中，通过插入或删除所需要的成员名，或者更新这些姓或名，对包括“成员姓”和“成员名”字段的个人姓名表进行修改。这里，我们将使用根据实施例 2 的技术进行插入。即，下标转换数组和值转换数组都包括结束位置数组和偏移数组。请注意，生成下标转换数组和值转换数组的过程包括生成新转换数组和更新以前存在的转换数组，相应地，在下文中，“生成”是指生成和更新，根据具体情况，可以叫做“生成（或更新）”。

图 28A 是关系图，说明个人姓名表的示例。在此实施例中，对于此个人姓名表 2800，生成图 28B 所示的记录号列表 2810、图 28C 所示

的“成员姓”字段 2820 的信息块和图 28D 所示的“成员名”字段 2830 的信息块。在指向信息块 2820 的值列表 2821 的指针数组中，提前创建初始下标转换数组 2823 和初始转换数组 2824。此外，在创建转换数组之前，实际数组 2823 对应于指向值列表的指针数组（原始）。在值列表 2822 中，提前准备初始下标转换数组 2826。请注意，实际数组 2827 对应于包含原始字段值的值列表。

在指向信息块 2830 的值列表 2831 的指针数组中，提前准备初始下标转换数组和值转换数组，在值列表 2832 中，提前准备初始下标转换数组。

在包含具有这样的结构的信息块的表格式数据中，如图 29A 和 29B 所示，在三个记录中（参见图 29A），设想删除行 1（第 2 行）中的记录的情况（参见图 29B）。在这种情况下，如下文所述，在每个信息块中，执行[通过下标转换数组删除元素] 所描述的过程。

首先，设想“成员姓”的信息块。

为了获得如图 29A 所示的“成员姓”，最初在指向值列表的指针数组中为图 30A 所示的符号 3000 所示的下标建立输入/输出关系。即，“记录号”的输入（它是下标），将给出“指向值列表的指针数组”（它是实际数组）的指针值的输出，如符号 3000 所示。

现在，在“指向值列表的指针数组”内的下标转换数组上，由图 1 所示的 CPU 12 执行几乎与图 17 所示的相同的过程。更具体地，首先，识别在实际数组（指针数组） 3001 中要删除的元素的位置（图 17 的步骤 1701），为有关删除位置之前的元素的转换数组分配区域（步骤 1702）。接下来，有关删除位置之前的元素，放置“删除位置-1”作为结束位置数组的元素（步骤 1703）。请注意，在此示例中，不提供起始位置数组，因此，忽略步骤 1704 和步骤 1705。

此后，有关删除位置之后的元素，结束位置数组的各种元素减少（步骤 1706），偏移数组的各种元素增大（步骤 1707）。通过此过程，可以获得图 30B 所示的下标转换数组。在图 30B 中，可以看出，对于是下标的记录号，“指向值列表的指针数组”（它是实际数组）的指针值是符

号 3010 所示的输出。

此外，在“成员名”的信息块上，CPU 12 执行几乎与图 17 所示相同的过程，其执行方式与在“成员姓”的信息块上相同（参见图 28D 的符号 2830）。因此，在“成员名”的信息块内的下标转换数组 3101 变成下标转换数组 3111，如图 31B 所示。相应地，“记录号”的输入（它是下标）和“指向值列表的指针数组”（它是实际数组）的指针值从图 31A 的符号 3100 所示的那样变成图 31 B 的符号 3110 所示的那样。

在这样执行生成下标转换数组的过程之后，通过从记录号获得指针数组，可以识别值列表的实际数组内的所需要的元素（字段值）。如图 32 所示，CPU 12 使用通过下标转换数组从新记录号（下标）获得的新下标，来识别实际数组内的元素（指针值）。例如，记录号“0”的偏移值为“0”，因此，获得实际数组的行 0（第 1 行）的元素（指针值），此指针值“1”用于识别值列表内的实际数组的字段值（参见图 32 中的 (i)）。记录号“1”的偏移值为“1”，因此获得转换后的下标“1+1=2”。因此，获取实际数组内的行 2（第 3 行）的元素（指针值）。接下来，此指针值“1”用于识别值列表内的实际数组中的行 1（第 2 行）上的字段值（参见图 32 中的 (ii)）。

有关“成员名”的信息块，很显然，使用类似的过程（参见 (iii) 和 (iv)）来识别值列表内的实际数组的字段值。如此，可以获得对应于新记录号的个人姓名表的表（视图）3200。在视图 3200 中，(i) 到 (iv) 都指出在前述的过程中识别的字段值。

[插入特定的表格式数据]

下面介绍插入数据的示例。例如，设想取两个记录（参见图 33A）并在行 0（第 1 行）和行 1（第 2 行）之间插入某记录（参见图 33B）。在这种情况下，在每个信息块上执行“通过下标转换数组插入元素”，然后，对所需要的信息块执行“生成值转换数组”。

图 34 是关系图，说明在插入记录（数据）之前各种信息块的状态。首先，设想在“成员姓”信息块上更新下标转换数组和在“成员名”信息块上更新下标转换数组。

有关“成员姓”，必须在值列表的实际数组内添加元素（字段值）“Ford”。为此，CPU 12 基于值列表内的下标转换数组执行插入过程。更具体地，确定元素（字段值）的插入位置（参见图 13 的步骤 1300）。在此示例中，必须插入元素（字段值）“Ford”，但在值列表的实际数组中，元素按升序（在这种情况下为字母顺序）放置，因此元素“Ford”必须放在元素“Smith”之前。相应地，插入位置为行 0（第 1 行）。

为此，根据图 13 的步骤 1301 为各种转换数组分配区域。在此示例中，插入位置为行 0（第 1 行），因此，在插入位置之前没有元素存在，在插入位置和插入位置之后对元素执行步骤 1303 到步骤 1307。如图 35B 所示，有关插入位置（参见符号 3501），指示插入位置的元素“0”（参见符号 3511）放在结束位置数组中，指出实际数组的末尾的位置“1”作为对应于偏移数组的元素而放置（参见符号 3512）。此外，有关插入位置之后的元素，包括与“+1”相加的原始值的值放在结束位置数组中（参见符号 3513），包括与“-1”相加的原始偏移值的值作为对应于偏移数组的值放置（参见符号 3514）。此外，新元素（字段值）放在实际数组的末尾（参见符号 3520）。在将要插入的新元素实际添加到实际数组的末尾时（参见符号 3520），通过值列表内的下标转换数组，虚拟（即，逻辑）数组按升序（字母顺序）排列。

同样，元素（字段值）“Jonny”必须添加到值列表的实际数组。为此，CPU 12 通过“成员名”的信息块中的值列表内的下标转换数组执行插入过程（图 13）。更具体地，确定元素（字段值）插入位置在行 1（第 2 行）（参见 36B 的符号 3610），为各种转换数组分配区域。对于插入位置之前的元素，分配通过取插入位置（在这种情况下为“1”）和减“1”（或加“-1”）所得的值作为该结束数组的元素（参见符号 3611）。另一方面，保留对应的偏移数组中的元素（参见符号 3612）。

此外，有关插入位置（参见符号 3610），指出插入位置的元素“1”放在结束位置数组中（参见符号 3513），指出位置“2”的偏移值“1”（指出实际数组的末尾）作为偏移数组内的元素给出（参见符号 3614）。

另一方面，对于插入位置之后的元素，通过取原始值并加“1”所得

的值放在结束位置数组中（参见符号 3615）。包括与“-1”相加的原始偏移值的值“0”作为偏移数组的对应值而放置（参见符号 3516）。此外，新元素（字段值）“Jonny”放在实际数组的末尾（参见符号 3620）。请注意，在将要插入的新元素实际添加到实际数组的末尾时，通过值列表内的下标转换数组，可以看出，如符号 3621 所示，虚拟（即，逻辑）数组按升序（字母顺序）排列。

在为每个信息块的值列表生成（或更新）值转换数组之后，此外还在指针数组部分生成（或更新）下标转换数组和值转换数组。

首先，在指向值列表的数组中，必须生成（或更新）值转换数组，用于转换实际数组的元素。例如，有关指向图 37A 和 37B 中所示的“成员姓”的信息块中的值列表的指针的数组，在关联的值列表（即，“成员姓”的值列表）中，前述的过程在虚拟（逻辑）数组的起始处插入新姓“Ford”（参见图 35B 的符号 3521）。然后，在指向值列表的指针数组的指针值中（即实际数组 3701 的元素），那些值为“0”或更大的元素必须都要增大。然后，根据图 25 所示的流程图，在前述的指向值列表的指针数组中，生成（或更新）值转换数组。

如图 37A 所示的数组中，值所取的最大值为“0”，“0”以上的值必须增大，结果，“+1”与偏移数组内的值“0”相加（参见图 25 的步骤 2503 和图 37B 的符号 3711）。最初，实际包含在实际数组内的元素（指针值）与通过图 37A 所示的符号 3702 所示的值转换数组输出的元素（指针值）相同，但在生成（或更新）值转换数组之后，通过值转换数组输出的元素（指针值）比原始值增大，如图 37B 的符号 3712 所示。

此外，通过在记录号“1”处（即行 1 或第 2 行）插入新姓“Ford”，在指向值列表的指针数组中执行下标转换数组的生成（或更新）过程，姓“Ford”也插入到值列表内的虚拟（逻辑）数组的行 0（第 1 行），因此，随值列表的前述的变化，必须在指向值列表的指针数组内执行值的生成（或更新）过程。

即，在指向值列表的指针数组中，提取指针值（该值是转换为整数的记录号（下标）），相应地，使用下面的下标转换数组的生成（或更新）

过程来生成下标转换数组。因此，可以显示对应于记录号“1”的姓“Ford”，此外，可以转换指针数组中的输出（值），以使元素“Ford”和“Smith”可以按升序排列，以正确地显示虚拟（逻辑）值列表。

首先，对应于记录号（它是下标）的插入，CPU 12 对前述的指针数组的下标转换数组执行图 13 所示的过程。

更具体地，要插入的新姓“Ford”（和新名“Jonny”）的记录号为“1”，因此，插入位置为行 1（第 2 行）（参见图 13 的步骤 1301），为插入位置和之前和之后的转换数组分配区域（步骤 1302）。此后，使用步骤 1303 到步骤 1308 的过程（实际上，忽略与起始位置关联的过程）形成图 38A 所示的下标转换数组（参见符号 3810）。

因此，通过获得下标转换数组，值列表指针数组给出符号 2811 所示的输出，该输出从记录号（它是下标）的输入获得。顺便说一句，如图 37A 和 37B 所示，在前述的示例中，值列表中的虚拟（逻辑）数组按升序（字母顺序）排列，因此，可以看出元素（字段值）“Ford”位于元素（字段值）“Smith”的上方，相应地，必须使用值转换数组来在值列表指针数组内的实际数组中执行值的转换，因此，可以正确地指示值列表内的元素（字段值）。

因此 CPU 12 根据图 25 执行该过程。在此过程中，通过插入姓“Ford”，执行一个过程，以使实际数组的值的输出为“1”变成“0”，即，给出偏移值“-1”，否则，就会给出偏移值“1”。因此，可以获得图 38B 所示的值转换数组（参见符号 3820）。

在图 38B 中，通过下标转换数组、实际数组和值转换数组为新记录号（下标）输出的值在虚拟（逻辑）数组中显示（参见符号 3821），因此，可以看出，正确地识别值列表的元素（字段值）。

对于“成员名”信息块，执行与在指向“成员姓”信息块的值列表的指针数组上执行的过程类似的过程。首先，如图 39A 和 39B 所示，在“成员名”信息块内，伴随与下标转换数组的值列表生成（更新）和将元素（字段值）插入到实际数组中的过程，执行指向值列表的指针的数组的生成(或更新)（参见图 36A 和 B）。显然，让 CPU 12 执行图 13 所

示的过程，可以获得图 39B 所示的值转换数组（参见符号 3910）。

接下来，有关指针数组的前述下标转换数组，CPU 12 执行图 13 所示的过程，插入记录号（它是下标）。更具体来说，由于新成员名具有要插入的记录号“1”，执行图 13 所示的过程，以行 1（第 2 行）作为插入位置（参见图 13 的步骤 1301）。因此，可以获得如图 40A 所示的偏移数组（参见符号 4010）。

此外，必须伴随生成（或更新）前述的下标转换数组和将元素插入到实际数组的末尾的过程，更新值转换数组，因此 CPU 12 执行如图 25 所示的过程，因此，可以获得图 40B 所示的值转换数组（参见符号 4020）。

在图 40B 中，值列表指针数组（在此记录号是下标）的输出变成虚拟（逻辑）数组中所示的那样（参见符号 4021）。可以看出，通过图 40B 的虚线所示的路线，获得各种记录号中的输出。

如此，在每个信息块上生成或更新下标转换数组和/或值转换数组之后，通过将记录号在指针数组中传递，可以识别值列表的实际数组内的所需要的元素（字段值）。如图 40A 和 40B 所示，在指向值列表的指针数组中，CPU 12 从新记录号（下标）通过下标转换数组、实际数组和值转换数组获得输出，用于识别值列表的实际数组内的元素（字段值）。例如，有关“成员姓”信息块，记录号为“0”的偏移值为“0”，因此获取实际数组内的行 0（第 1 行）的元素（指针值），此外，对应于此元素（指针值）的偏移值为“1”，因此指向值列表的指针数组中的输出变成“1”（参见图 40 的 (i)）。相比之下，对于记录号“1”，下标转换数组的偏移值为“1”，因此获得转换后的下标“1+1=2”，这用于获取实际数组内的行 2（第 3 行）的元素（指针值）。此外，对应于此元素（指针值）的偏移值为“-1”，因此，指向值列表的指针数组中的输出变成“0”（参见图 40 中的 (ii)）。同样，还可以为记录号“3”获得输出“1”。此输出给予值列表，通过值列表内的下标转换数组的输出用于识别实际数组内的元素（字段值）。有关“成员名”，从记录号通过类似的路线识别值列表的实际数组内的元素（字段值）（参见图 41 的 (iv) 到 (vi)）。

如此，可以获得对应于新记录号的个人姓名表（视图）4100。在视图

4100 中，(i) 到 (vi) 指出前述的过程中标识的字段值。

通过此实施例，将向实际数组的输入定义为下标，通过转换此下标和/或转换实际数组中输出的值，可以向实际数组中插入元素或从中删除元素，而不移动实际数组的元素本身。即，结构是这样的，通过在实际数组的输入和输出上执行规定的功能，可以虚拟地（逻辑地）执行实际数组的插入/删除/更新。相应地，可以插入数据，而不必移动实际数组内的元素。由于不要求移动元素，对于表格式数据的处理速度可以显著地提高，特别是对于元素数量特别大的情况。

[更新表格式数据]

如上文所描述，采用本发明，可以插入和删除表格式数据，而不必改变实际数组（即，改变或移动数据本身）。更新表格式数据相当于按顺序删除某记录号的数据（删除记录），然后再插入相同记录号的数据。例如，在图 28 所示的表格式数据（个人姓名表）中，设想在行 1（第 2 行）进行记录更新的操作（例如，将“成员姓”“Ford”更改为“Trumen”和/或将“成员名”“Jonny”更改为“Tom”的过程）。在这种情况下，首先使用图 29A 和 29B 到图 32 所示的技术来在行 1（第 2 行）上执行记录删除的过程，然后使用图 33A 和 33B 到图 42 所示的技术来执行在行 0（第 1 行）和行 1（第 2 行）之间插入新记录的过程。

[插入以信息块形式表示的表格式数据]

我们现在介绍对以前述的信息块的形式表示的表格式数据过程处理的情况。

如前述的示例所述，如图 46A 所示，伴随将字段值插入到值列表的过程，CPU 12 执行下标转换数组的处理过程，即，生成（或更新）下标转换数组，以获得“虚拟（逻辑）数组”，其状态是，字段值按字母顺序排序（步骤 4601）。此后，由于元素（字段值）在虚拟（逻辑）实际数组（值列表）中的规定位置插入，关于指向值列表的指针数组，相应地，那些值大于对应于规定位置的值元素增大（步骤 4602）。接下来，针对指针数组内的记录号和指针值，CPU 12 执行指向值列表的指针数组的下标转换数组的处理过程（步骤 4603）。在此处理之后，最后，为指针数

组的实际数组中的最新插入的字段值（比原始实际数组内的最大值大 1 的值）生成值转换数组（步骤 4604）。

从此过程可以看出，步骤 4601 和步骤 4603 能独立地执行（还可以并行地执行，视情况而定），另一方面，步骤 4602 和步骤 4604 可以作为批处理任务来执行。相应地，如图 46B 所示，插入过程可以包括：与指针数组的下标转换数组相关的处理（步骤 4611：对应于图 46A 的步骤 4603）、与值列表的下标转换数组相关的处理（步骤 4612：对应于步骤 4601）和与指针数组的值转换相关的处理（步骤 4613：对应于步骤 4602 和 4604）。

现在设想指向包括对应于 m 个记录的 m 个元素（指针值）的值列表和指向包括 n 个元素（字段值）的值列表的指针数组。这里，假设要将元素插入到行 j 上的记录中。

如图 47 所示，可以看出，通过步骤 4611 的过程，针对指向值列表的指针数组，获得包括结束位置数组和偏移数组的下标转换数组（参见符号 4701）、在其末尾包含新指针值的实际数组（参见符号 4702）。

另一方面，有关按规定的顺序（例如，字母顺序）排列元素的值列表，甚至在基于对这些元素扫描在行 i 插入元素的情况下，通过步骤 4612 的过程，可以看出，获得包括结束位置数组和偏移数组的下标转换数组（参见符号 4703）、在其末尾包含元素（字段值）的实际数组（参见符号 4704）。这里，在指向值列表的指针的数组中，这意味着，当下标（记录号）“p”在范围 0<=p<=(j-1) 内时，增加偏移值“0”，当 j<=p<=j（即，当 p=j 时），增加偏移值“m-j”，当 (j+1)<=p<=m 时，增加偏移值“-1”。此外，在值列表中，这意味着，当输入下标（指针数组的输出）“q”在范围 0<=q<=(i-1) 内时，增加偏移值“0”，当 I<=q<=i（即当 q=i 时）时，增加偏移值“n-i”，当 (i+1)<=q<=m 时，增加偏移值“-1”。

接下来，为讨论指向值列表的指针数组中的值转换数组，可以设想，没有必要增加值列表中虚拟（逻辑）插入位置小于“i”的值的偏移（在指针值的意义上没有变化）。另一方面，对于前述的插入位置为“i”或更大的值，可以看出，这意味着，它们每个都向下（即在更大值的方向上）

移一个位置，因此，有必要向值中增加偏移值“1”。

此外，加到实际数组的末尾的指针值的值（即伴随记录号插入过程，在实际数组内位置“m”处添加的指针值）可以视为“n”。这是因为，在插入过程之前，在指针数组的实际数组中，可以取“0”或“n-1”值，因此由插入过程给出指针值“n”。因此，可以看出，此指针值“n”随值转换而变化，并指向值列表中的虚拟（逻辑）插入位置“i”，因此偏移值只需设置为“i-n”。很显然，即使给实际数组内的位置“m”处的指针值分配另一个值，值转换给出偏移值，以使将值转换为“-i”。如此，可以获得如图 47 的符号 4705 所示的值。即，有关指针数组值更改列表，当实际数组内的元素（指针值）“r”在范围 0<=r<=(i-1) 内时，增加偏移值“0”，当 I<=r<=(n-1) 时，增加偏移值“+1”，当 n<=r<=n（即当 r=n）时，增加偏移值“i-n”。

有关删除，与插入相同，设想指向包括对应于 m 个记录的 m 个元素（指针值）的值列表和指向包括 n 个元素（字段值）的值列表的指针数组，假设要删除行 j 中的记录。在这种情况下，如图 48 所示，查看指针数组内的下标转换数组（符号 4801），从图 48 中可以看出，当下标（记录号）“p”在范围 0<=p<=(j-1) 内时，增加偏移值“0”，当 j<=p<=m-2 时，增加偏移值“+1”。因此，在转换后的下标中，对应于原始（未转换的）下标“j”的指针值不再指向谁，因此标识的值列表被虚拟地（逻辑地）删除。

[采用另一种技术删除表格式数据]

下面介绍根据本发明从表格式数据中删除数据的另一个示例。这种技术在不执行数据插入的情况下很有效。此外，设想包括信息块（信息块具有图 28A-D 所示的结构）的表格式数据的情况，其中删除行 1（第 2 行）中的记录，如图 29A 和 29B 所示。

在此示例中，没有必要处理“成员姓”和“成员名”信息块，而是通过处理记录号列表来删除记录。更具体地，如图 42 所示，删除过程在记录号列表的下标转换数组上执行，以使原始记录号“1”的逻辑输出删除。如图 42 所示，记录号列表中的下标输入和记录号输出之间的关系

如符号 4201 所示。为此，为了删除行 1（第 2 行），执行图 17 所示的过程。更具体地，识别删除位置（在这种情况下行 1）（步骤 1701），接下来，为与删除位置关联的转换数组创建区域（步骤 1702），对转换数组内的元素执行规定的处理过程（参见步骤 1703 到 1707，但是，在此示例中，在起始位置不执行处理）。因此，获得如图 42 所示的下标转换数组（参见符号 4210）。在传递此下标转换数组时，输入和输出之间的关系如符号 4211 所示，在此可以看出，记录号“1”被逻辑地删除，后续的记录号都移动 1 个位置。

通过此示例，记录号本身的下标转换数组用于实现虚拟（逻辑）删除和记录号的移动。相应地，可以通过极其简单的技术删除记录。

[本发明在事务处理中的应用]

下面介绍使用前述的技术来执行适当的事务。如上文所述，使用本发明，可以插入、删除和更新元素，而不用改变实际数组中的元素的顺序或移动元素。因此，此属性可以用于正确地实现事务。图 43 是流程图，说明了事务处理过程。该处理过程也由图 1 的 CPU 12 执行。此外，假设已经事先完成了将表格式数据变成许多信息块的过程（例如，图 6 所示的过程）。

CPU 12 首先将初始转换数组应用到与事务处理相关的所有信息块（步骤 4301）。例如，在记录号列表中，为记录号数组（实际数组）生成初始下标转换数组，或在指向某信息块的值列表的指针数组中，为指针数组的实际数组生成初始下标转换数组和值转换数组，在该信息块的值列表中，为实际数组（实际值列表）生成初始转换数组。

接下来，CPU 12 执行规定的事务处理过程（步骤 4302）。这里，如果必须执行记录插入、删除或更新，如上文所述，更新下标转换数组和值转换数组，如有必要，将元素插入到实际数组的末尾（或在另一个区域生成差异数组）。例如，在操作员或其他人请求回滚（步骤 4303 的“是”的情况），CPU 12 放弃当时存在的所有转换数组（步骤 4304）。

另一方面，如果没有回滚请求（步骤 4303 的“否”的情况），且事务处理过程完成，CPU 12 就提交处理（步骤 4304）。在提交时，CPU 12

可以销毁各种转换数组或将各种转换数组返回到初始转换数组（在步骤4301 中生成）的状态，以便可以通过此状态下的转换数组执行另一个过程。此外，另一个过程还可以在处理结束时按“原样”使用转换数组。此外，还可以保存转换数组，并在必要时使用。

接下来，设想这样的情况，相同的表格式数据用于并行地执行的许多过程中。例如，设想这样的情况，某过程涉及数据更新（例如，从帐户中提取 10,000 日圓的过程），另一个过程执行表格式数据查询（例如，查找各种帐户的帐户余额的总和的过程）。

在前述的过程中（该过程涉及数据更新，叫做“更新过程”），在提交发生之前临时提取 10,000 日圓。使用本发明，更新过程使用下标转换数组和值转换数组来执行记录删除和插入（参见图 44 中的实线箭头）。另一方面，另一个过程（叫做“查询过程”）直接执行实际数组的查询，而不引用前述的下标转换数组或值转换数组（参见图 44 的虚线箭头），因此，即使并行地执行许多过程，处理过程也会正确地执行，而不会相互干扰。

下面简要介绍锁定记录的技术。在这种情况下，提供信息块，该信息块根据每个记录号码控制锁定。图 45 是描述记录锁定的关系图。如图 45 所示，以表格式数据提供用于锁定控制的信息块（符号 4501），还提供指针数组（符号 4502），在该指针数组中，可以放置与记录号码相同数量的元素。此指针数组可以包含“NON”元素（存储值），表示不可读或不可写，“R”表示只读（不可以更新和删除），“RW”表示可以读取和写入。显然，没有必要使用这些字母，当然，可以分配数值，来表示状态（例如，“0”表示不可读或不可写，“1”表示只读）。如此，通过在对应于记录号的指针数组上执行查询，该过程可以确定是否可以读取对应于该记录号的各种数据，此外，在实际处理过程中，该过程可以更新对应于某记录号的指针数组，因此获得排他的控制。

如此，通过给数组提供字段值（这些字段值包含对应于每个记录的锁定属性（类型）），首先，可以快速地执行锁定。例如，可以快速地锁定在某些字段具有特定字段值的记录（例如，在性别字段具有字段值“女

性”的记录)。即,在现有技术中,必须使用 OS 调用才能注册每个要用 OS 单独地锁定的记录,而采用本发明,只需更改信息块数组内的字段值即可执行锁定,而不涉及 OS 调用。

此外,采用前述的结构,甚至在联接表格式数据的情况下,也可以传播锁定。例如,设想根据日本专利申请 No. 11-151156 进行联接的情况,在许多表格式数据的表中,每个表都包括许多信息块,通过共享信息块的字段值(字段值具有共同的意义),来执行联接。首先,对于每个表格式数据的表,生成包含数组的信息块,用以存储与锁定相关的信息。这里,在联接过程之前,表格式数据的信息块内的数组叫做原始表。接下来,执行联接过程,但在这种情况下,基本上存储原始表的指针数组,因此从通过联接创建的表,可以查询或重写包含数组的信息块(数组中存储原始表的锁定信息)。

相应地,可以创建联接的表,可以在显示联接表时接受来自用户的数据修改请求。更具体地,下文对该过程进行描述。

(1) 首先,创建和显示联接表。

(2) 接下来,当用户发出修改联接表内的记录的请求时,程序确认没有锁定应用到请求的记录。此时,在联接生成的表中,没有控制锁定信息的信息块存在,因此,事实上,检查原始表中与记录相关的锁定信息。

(3) 接下来,程序锁定联接表内的该记录。事实上,锁定原始表中的对应的记录。

(4) 此后,程序根据用户的请求执行数据的重写。

(5) 最后,程序提交修改并释放锁定。

如此,采用前述的结构,可以将锁定传播到联接的表,因此,可以重写联接的表。

本发明决不局限于前述的实施例,因为在权利要求中叙述的本发明的范围内,各种修改都是可能的,因此,没有必要说,这些也包括在本发明的范围内。

例如,本发明不仅局限于如上面的实施例 5 到 7 所描述的许多信息块所构成的表格式数据的插入、删除和更新。如图 49 所示,可以执行

表格式数据的插入、删除和更新，这种形式的数据的特征是，记录号（参见符号 4901）与包括姓（4 个字符）、名（4 个字符）、地址（20 个字符）的字符串的结构（参见符号 4902）关联。

例如，在插入的情况下，假设包括姓名、地址等的结构数组为实际数组、假设记录号为下标，可以执行几乎与图 13 中所示的相同的过程。在删除的情况下，同样可以设想结构数组为实际数组，记录号为下标，因此，执行与图 17 中所示的几乎相同的过程。

此外，根据本发明，值转换数组也可以单独执行。例如，设想这样的情况，从数模转换器（未显示）按顺序提供 1 亿个测量数据的样本。通常情况下，1 亿个样本数据不容易包含在单独的内存中（因为即使每个样本的大小为 2 字节，也要求 2 亿个字节的数据容量（超过 100 个 MB）），因此，设想将数据分存在 10 个内存中的情况。在图 50 中，1000 万个测量数据样本（样本数据）存储在每个存储区中，从第一个存储区（No. 0）5000 到最后一个（No. 9）存储区 5009。

在这种情况下，设想查找和存储测量数据的总值的情况。这可以根据本发明按照下面的过程来执行。

将测量数据的数据值相加，从每个存储区的开始到结束。这可以由许多 CPU（在这种情况下最好是 10 个 CPU）并行地执行，因此，在存储区上执行求和过程，每个存储区都执行操作。因此，可以计算图 50 的符号 5010、5011、5019 所表示的每个存储区的总和。

接下来，获取对应于上面的相邻的存储区的存储区内的最后地址的逻辑总值，这变成该存储区的值转换数组中的偏移数组内的偏移值。在此示例中，在单个存储区内使用相同的偏移值，因此，没有必要创建结束位置数组或其他数组。例如，有关存储区 5021，偏移值为 Z (9,999,999)（其中 z 是逻辑总值），通过将偏移值 Z (9,999,999) 加到对应于此存储区 5021 的起始地址的总值 Y (10,000,000)，可以得出逻辑总值 Z (10,000,000)。此外，通过将偏移值 Z (9,999,999) 加到对应于此存储区 5021 的结束地址的总值 Y (19,999,999)，可以得出逻辑总值 Z (19,999,999)。此总值 Z (19,999,999) 可以变成在下面的相邻的存储区中

使用的偏移值。

这里，讨论计算总值所需要的时间，如果按上文所述在 10 个存储区中执行总值的计算过程，得出每个存储区的总值需要 1000 万步，设置附加偏移值大约需要 8 步。相比之下，可以看出，若使用常规技术，要获得总值，需要 1 亿步。这是因为，即使分到存储区中，位于上面的存储区的总值必须首先计算，只有在此之后，才开始对存储区进行计算，因此，各种存储区的并行处理事实上是不可能的。如此，通过根据本发明使用值转换数组，可以加速科学技术计算。

此外，根据本发明，可以采用图 51 中所示的嵌套结构。在图 51 中，在最内层嵌套中提供下标转换数组（参见符号 5101）、实际数组（参见符号 5102）和值转换数组（参见符号 5103）。另一方面，在位于外层的嵌套中，包括前述的下标转换数组、实际数组和值转换数组的集可以视为第二个实际数组（参见符号 5112），下标转换数组（参见符号 5111）放在其输入端，且值转换数组（参见符号 5113）放在其输出端。同样，在更加外层的嵌套中，内层嵌套可以视为第三个实际数组（参见符号 5212），下标转换数组（参见符号 5211）放在其输入端，而值转换数组（参见符号 5213）放在其输出端。通过采用这样的结构，可以执行分层结构的插入、删除和更新过程。例如，在多次执行插入值，以后提交更新事务的情况下，前述的结构非常有用。

此外，在上面的图 47 所示的示例中，在值列表指针数组中，指针值放在实际数组的末尾，其指针值设置为 f(m)=n（其中，n 比到目前为止指针值可取的最大值“n-1”大 1）。在值列表中，字段值放在实际数组的末尾，但本发明不仅局限于这样的结构。即，有关指向值列表的指针数组，可以将指针值放在另一个区域（差异数组），该区域与实际数组不相邻，指针值 f(m) 也可以设置为另一个值 y（其中，y>=n）。此外，有关值列表，字段值可以放在另一个区域（差异数组），该区域与实际数组不相邻。图 52 是一个示例，说明将指针值 (f(m)=y) 放在指针数组的区域“x”内，将字段值放在值列表的区域“z”内。也是在这种情况下，可以如图 47 所描述的那样，得出下标转换数组和值转换数组。

此外，前述的实施例描述了针对信息块形式的表格式数据，如何按顺序执行数据更新过程、插入过程和删除过程，或者删除过程和更新过程。对于这样的更新过程，按照下面的描述，可以得出下标转换数组和值转换数组。

例如，设想具有 m 个记录号和 n 个字段值的表格式数据的情况，其中要更新记录号“j”的字段值。这里，设想插入要更新字段值的情况下插入位置（逻辑插入位置）为“i”。

首先，我们将介绍这样的情况：在值列表指针数组中，指针值放在实际数组的末尾，其指针值设置为 f(m)=n（其中，n 比到目前为止指针值可取的最大值“n-1”大 1）。在值列表中，用于更新的字段值放在实际数组的末尾。在这种情况下，如图 53 所示，有关指针数组的下标“p”，当 0<=p<=(j-1) 时，给出偏移值“0”，当 p=j 时，给出偏移值“m-j”，当 j+1<=p<=m-1 时，给出偏移值“0”。此外，可以看出，指针数组的值转换数组和值列表的下标转换数组与图 47 中所示的相同（在删除过程中，不会触及到这些，因此插入过程的这些东西可以保留下来）。

或者，在更新过程中，在将指针值 (f(m)=y) 放在指针数组的区域“x”，将字段值放在值列表的区域“z”的情况下，可以采用几乎与图 52（参见图 54）中所描述的相同的方式得出下标转换数组和值转换数组。

此外，在前述的实施例中，执行下标转换数组、值转换数组等的生成和更新的过程，通过这样方法来实现：将规定的程序加载到典型的计算机系统 10 中，然后执行此程序，但本发明决不局限于此，而且不必把本发明说成是采用下述方法来实现：通过将专用于数据库处理的单板机连接到个人计算机或其他普通计算机系统，让前述的处理过程由所述的单板机来执行。相应地，在本说明书中，装置不一定是指物理装置，而且还包括这样的情况，各种装置的功能用软件来实现。此外，一个装置的功能可以用两个或更多的物理装置来实现，或者两个或更多装置的功能可以由一个物理装置来实现。

图 1

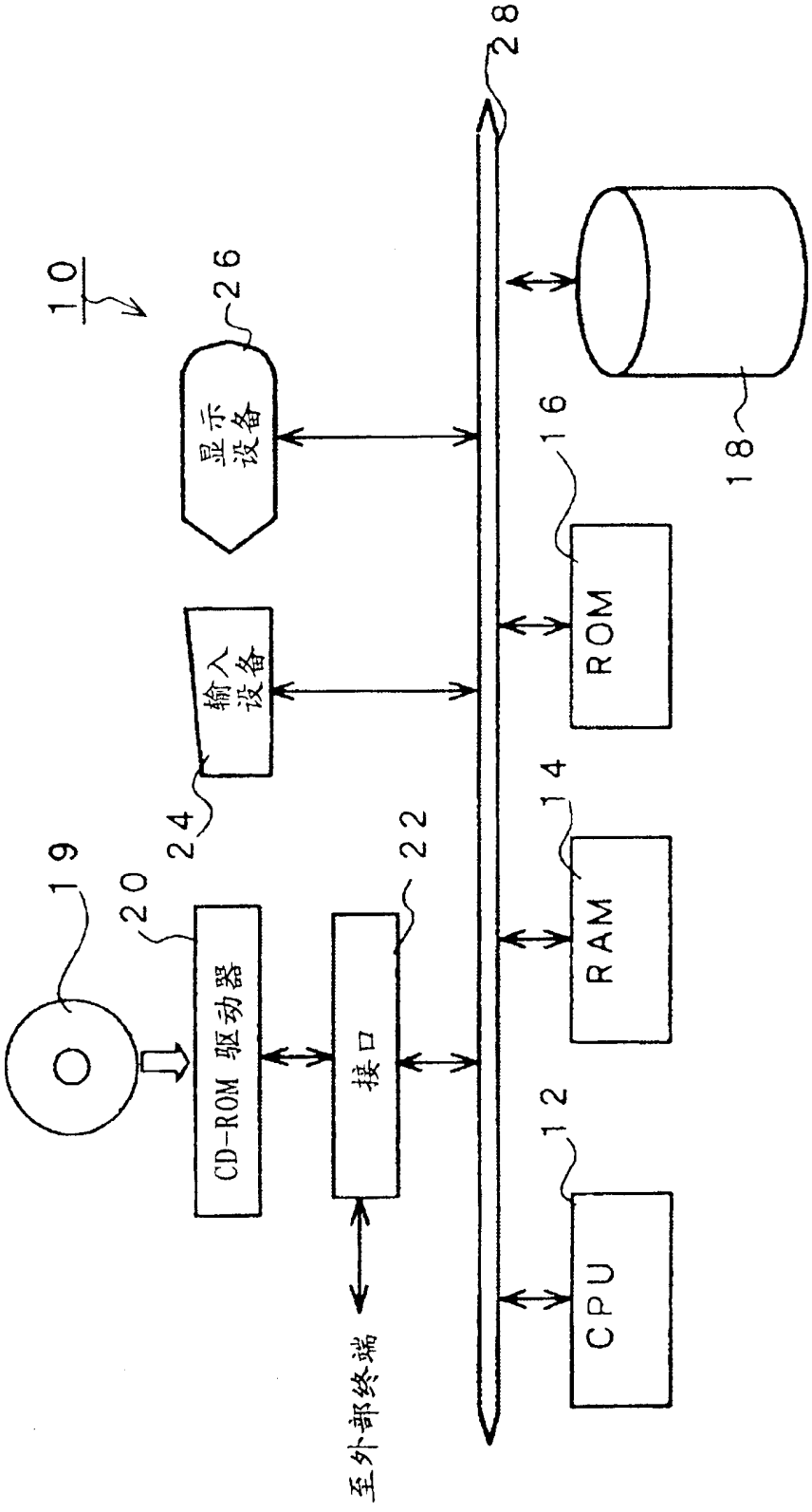

10
26
显示设备
28
18
16
ROM
输入设备
24
19
20
22
14
RAM
CD-ROM 驱动器
接口
12
CPU
至外部终端

图 2

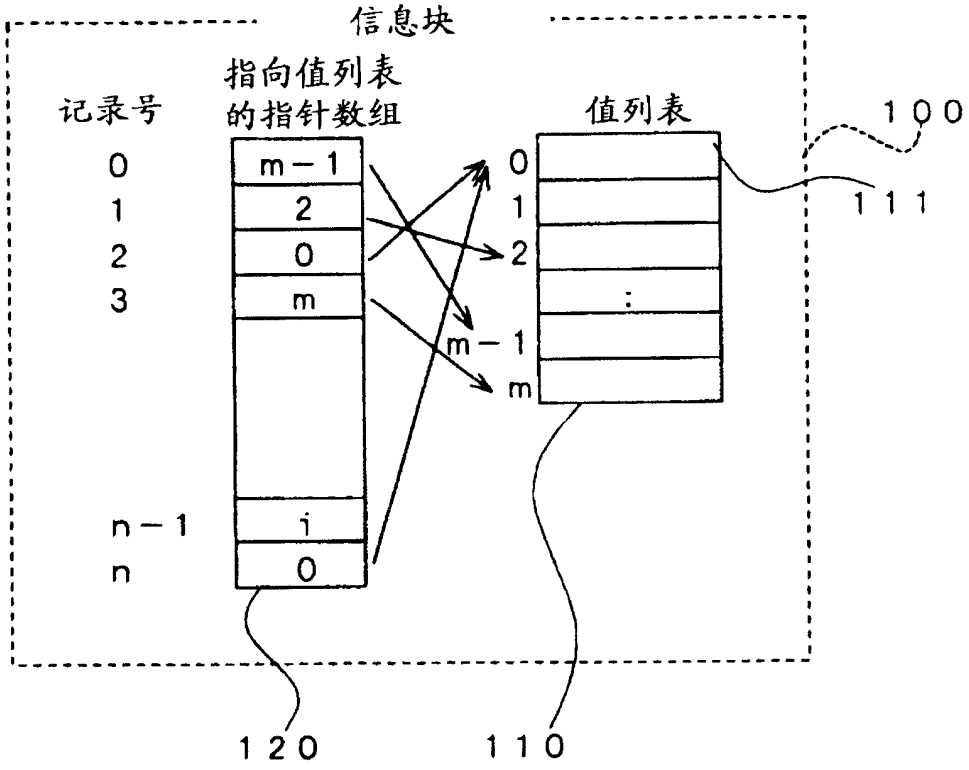

信息块
记录号
指向值列表的指针数组
值列表
100
111
0
1
2
3
n－1
n
m－1
2
0
m
i
0
0
1
2
:
m－1
m
120
110

图 3A

记录号	客户 ID	客户名	电话号码
0	1	Smith	333-aaa
1	2	Lennon	441-bbbb
2	3	Bernstein	558-cccc
3	4	Carlson	998-dddd
4	5	Ford	112-eeee
5	6	Eagle	225-ffff

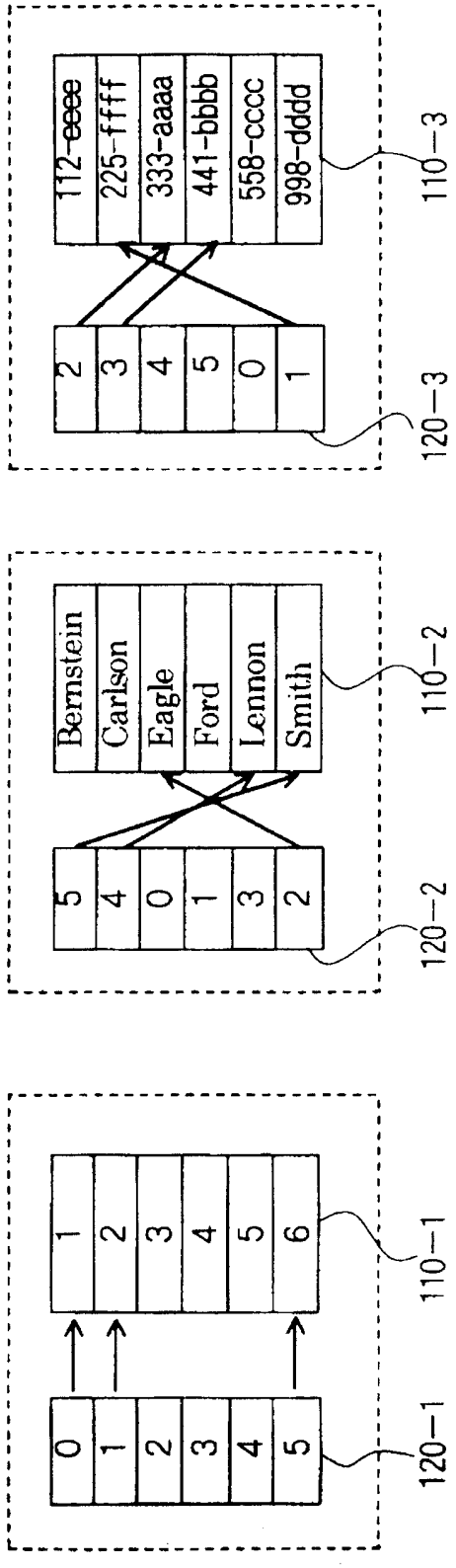


图 4A

记录号	性别	年龄	职业
0	女性	18	程序员
1	男性	21	学生
2	女性	31	教师
:	:	:	:
:	:	:	:
999999	女性	16	学生

图 4B

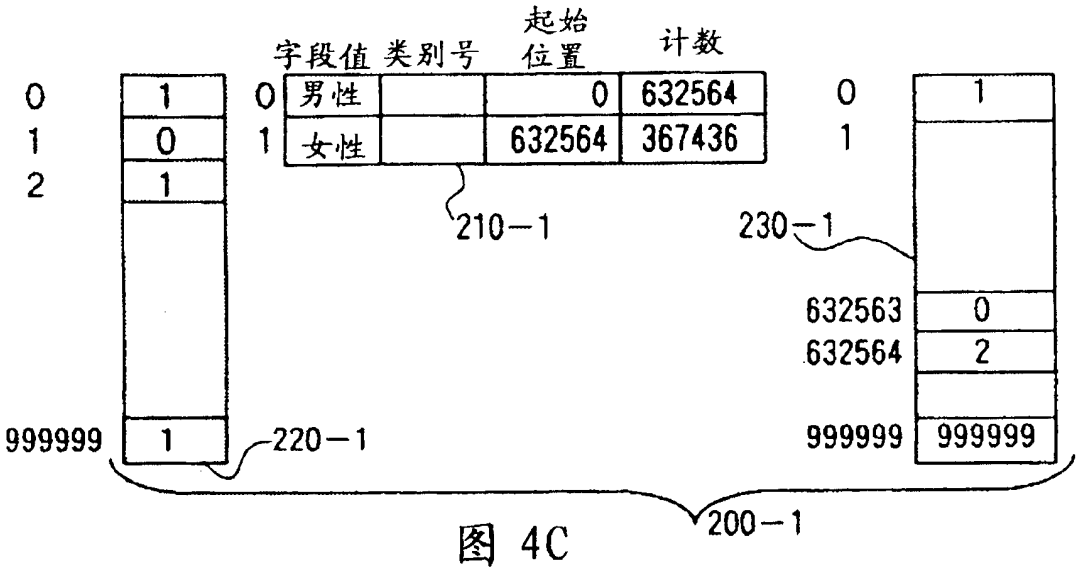


图 4C

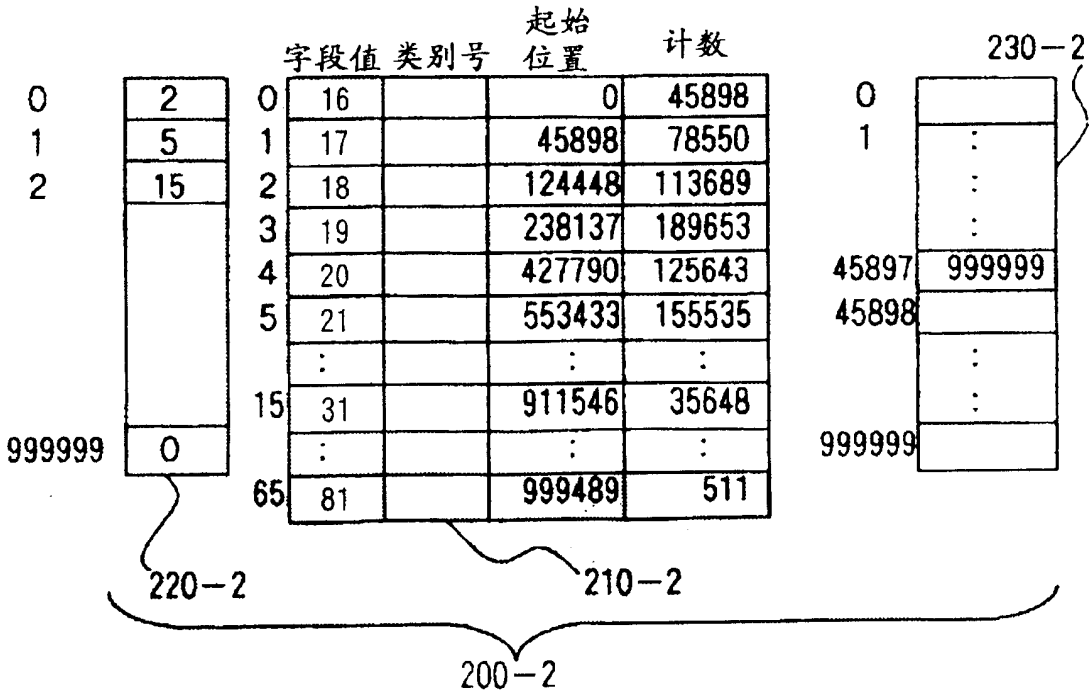


图 5

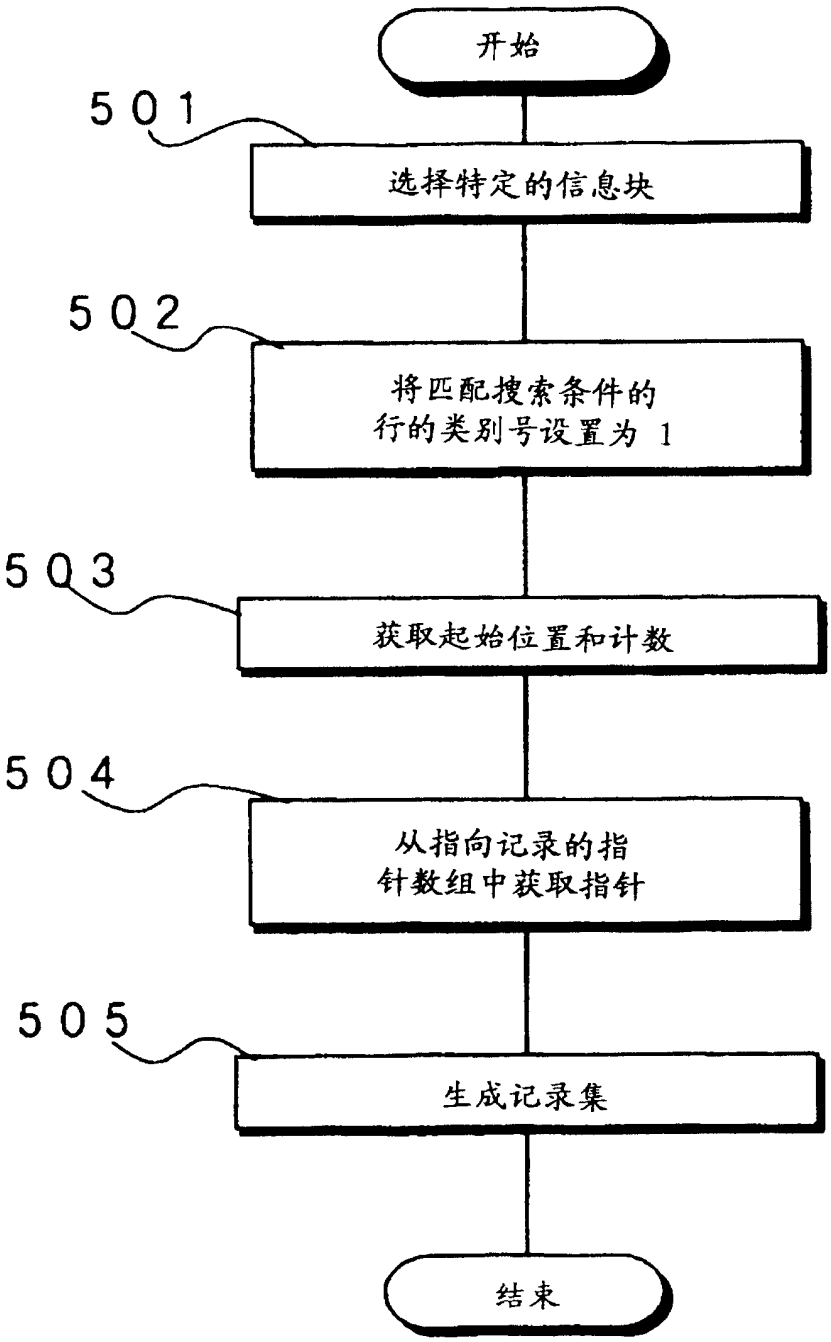

开始
501
选择特定的信息块
502
将匹配搜索条件的
行的类别号设置为 1
503
获取起始位置和计数
504
从指向记录的指
针数组中获取指针
505
生成记录集
结束

图 6

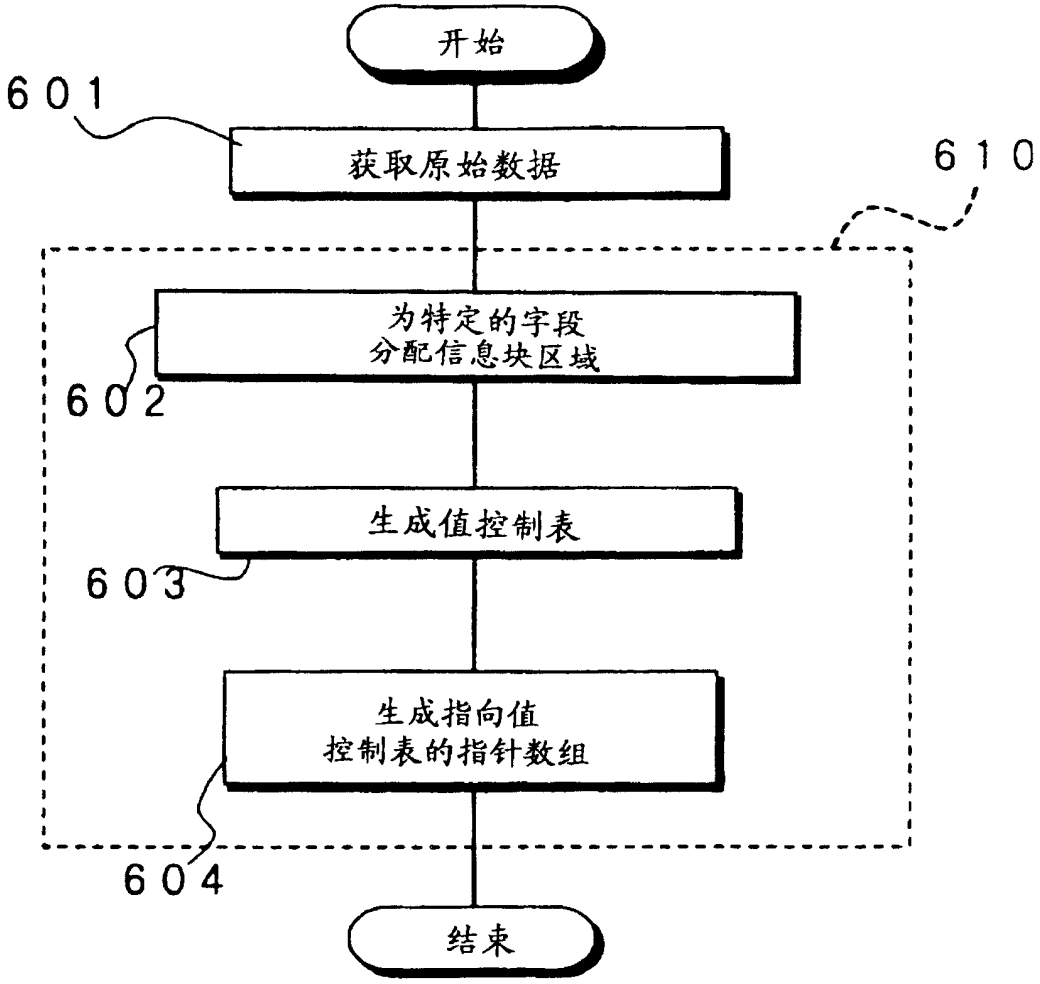

开始
601
获取原始数据
610
为特定的字段
分配信息块区域
602
生成值控制表
603
生成指向值
控制表的指针数组
604
结束

图 7A

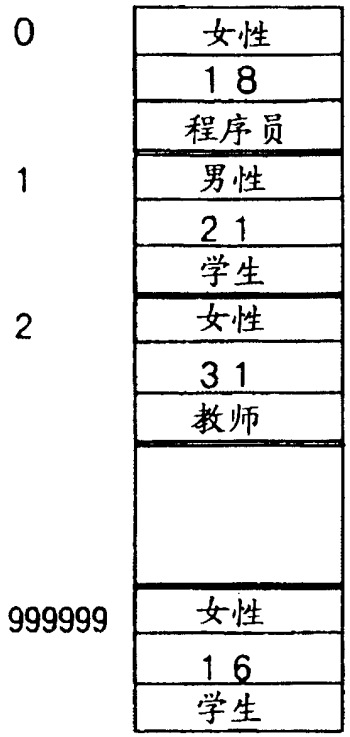

0
女性
18
程序员
1
男性
21
学生
2
女性
31
教师
999999
女性
16
学生

图 7B

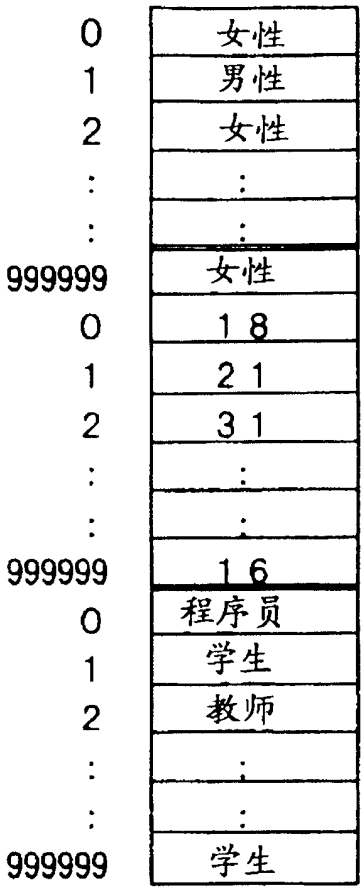

0
女性
1
男性
2
女性
999999
女性
0
18
1
21
2
31
999999
16
0
程序员
1
学生
2
教师
999999
学生

图 8A

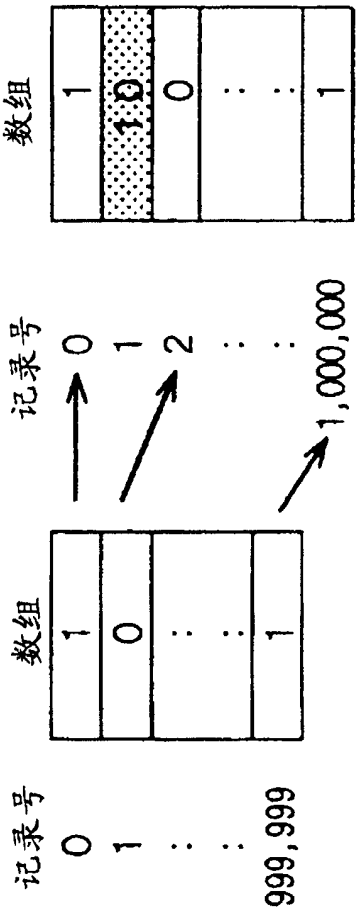

记录号
数组
0
1
:
999,999
1
0
:
1
记录号
数组
0
1
2
:
1,000,000
1
10
0
:
1

图 8B

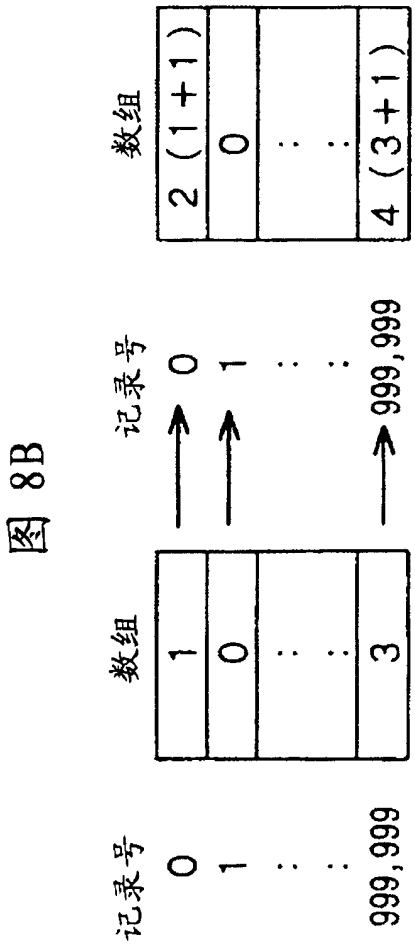
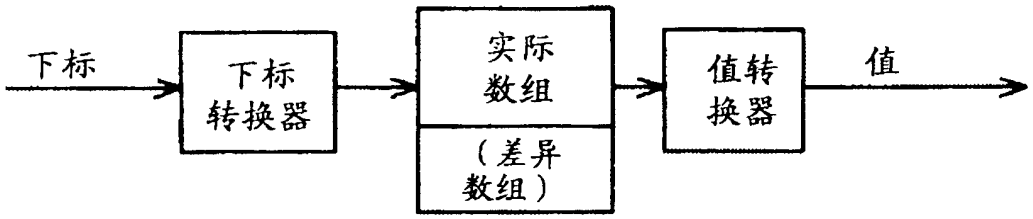

记录号
数组
0
1
:
999,999
1
0
:
3
记录号
数组
0
1
:
999,999
2 (1+1)
0
:
4 (3+1)

图 9A

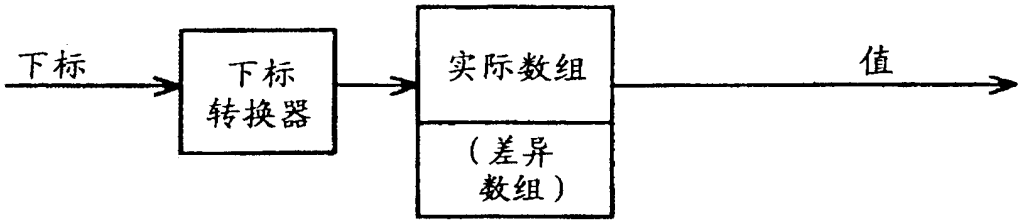


图 9B



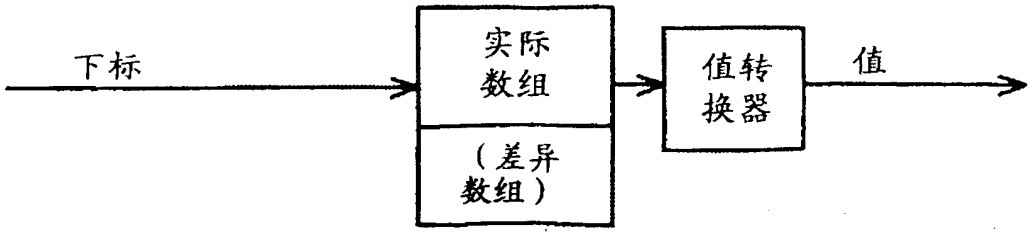
转换下标和值的示例

图 9C



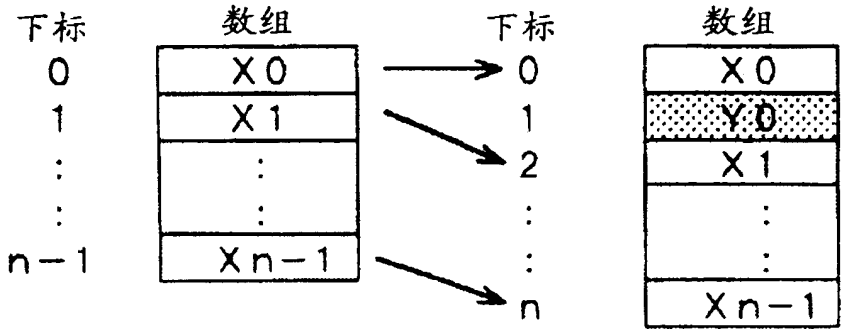
只转换下标的示例

图 9D



只转换值的示例

图 10A



将元素插入大小为“n”的数组中

图 10B

下标	存储值
0	X0
1	Y0
2	X1
:	:
n	Xn−1

下标和存储值
之间的逻辑关系

图 10C

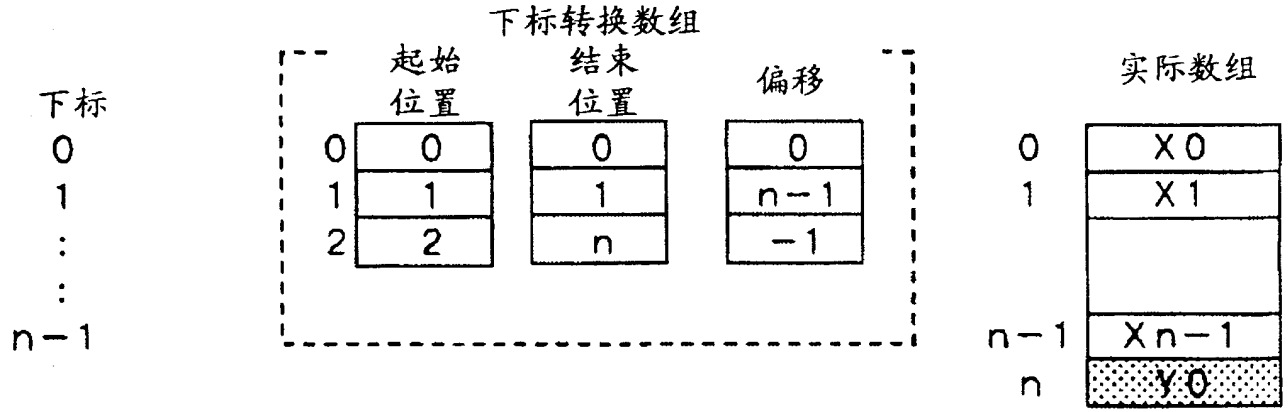


图 11

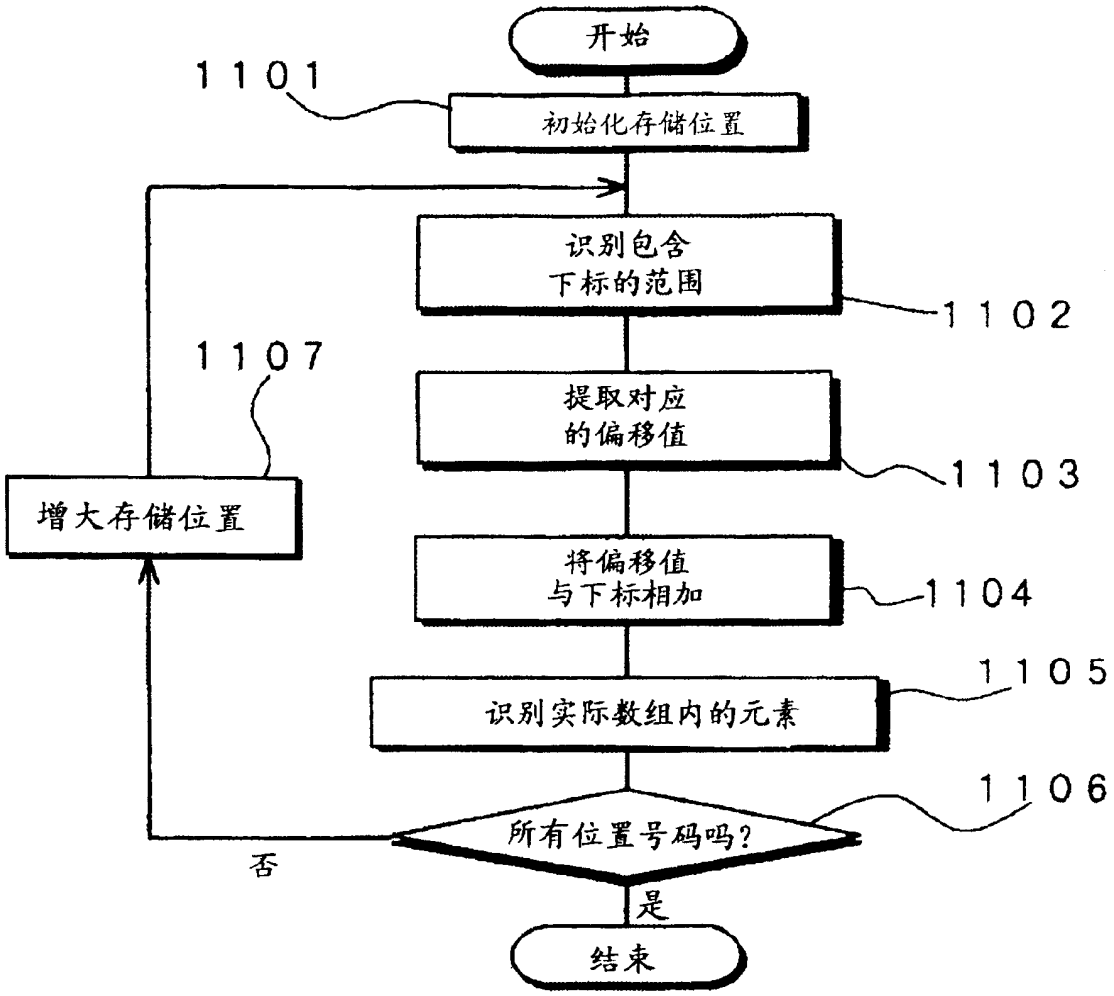

开始
1101
初始化存储位置
识别包含
下标的范围
1102
1107
提取对应
的偏移值
1103
增大存储位置
将偏移值
与下标相加
1104
1105
识别实际数组内的元素
1106
所有位置号码吗?
否
是
结束

图 12

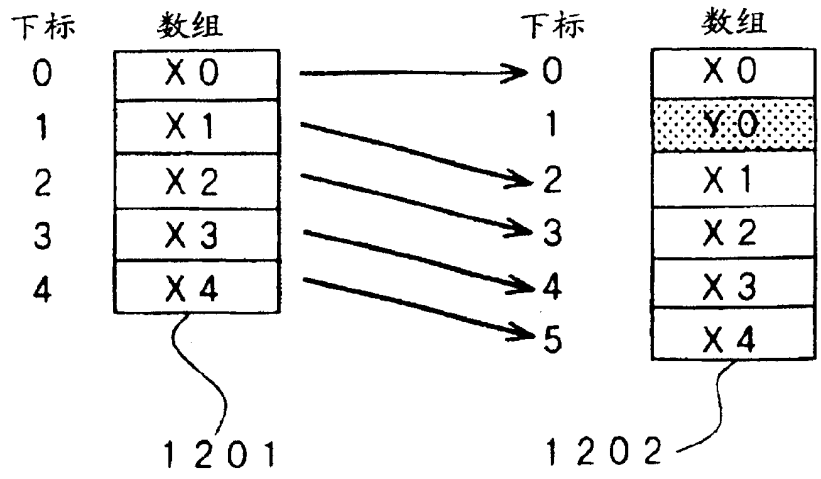

下标
数组
0
X 0
1
X 1
2
X 2
3
X 3
4
X 4
1201
下标
数组
0
X 0
1
Y 0
2
X 1
3
X 2
4
X 3
5
X 4
1202

图 13

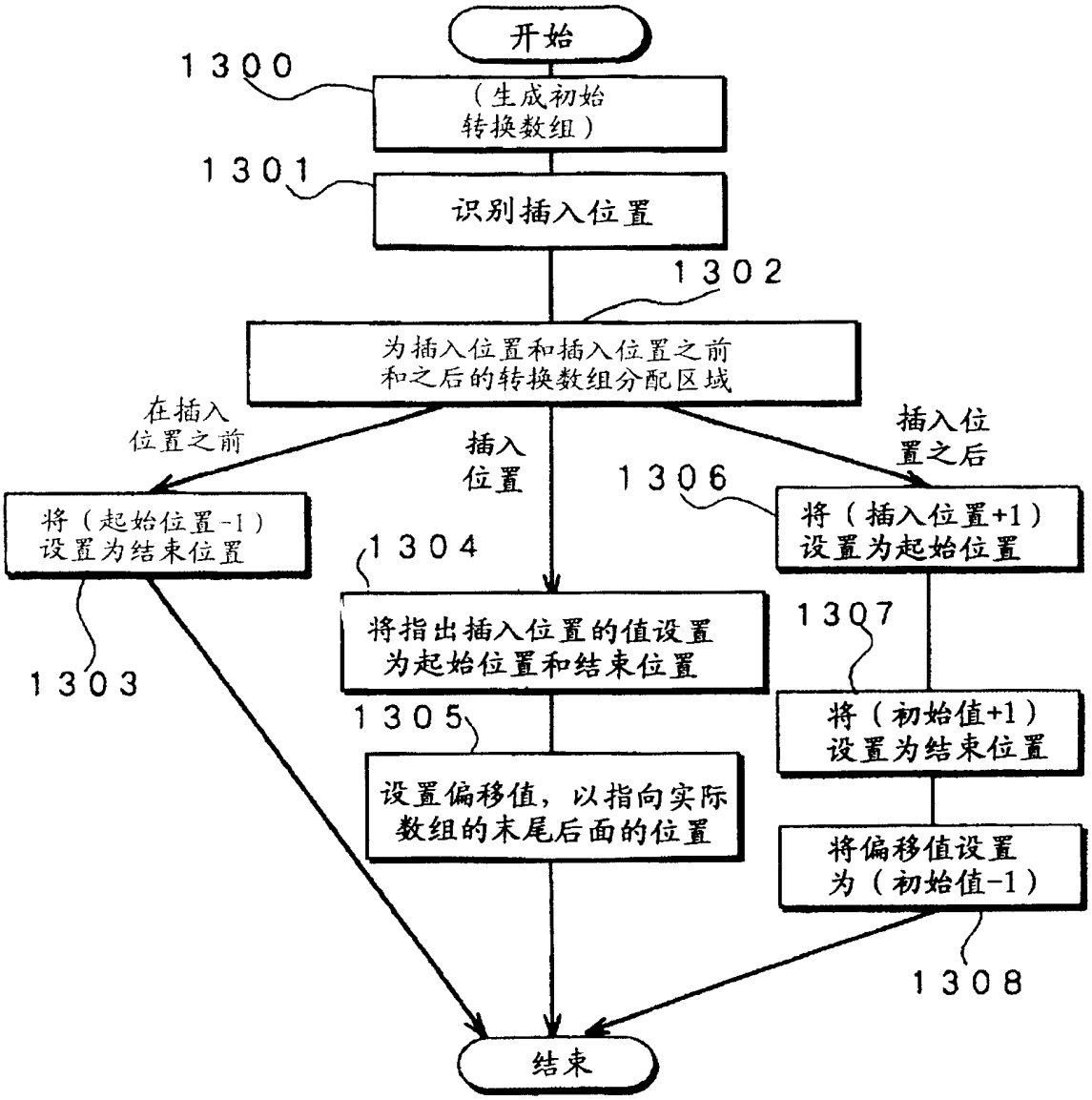

开始
1300
（生成初始转换数组）
1301
识别插入位置
1302
为插入位置和插入位置之前和之后的转换数组分配区域
在插入位置之前
插入位置
插入位置之后
将（起始位置-1）设置为结束位置
1303
1304
将指出插入位置的值设置为起始位置和结束位置
1305
设置偏移值，以指向实际数组的末尾后面的位置
1306
将（插入位置+1）设置为起始位置
1307
将（初始值+1）设置为结束位置
将偏移值设置为（初始值-1）
1308
结束

图 14

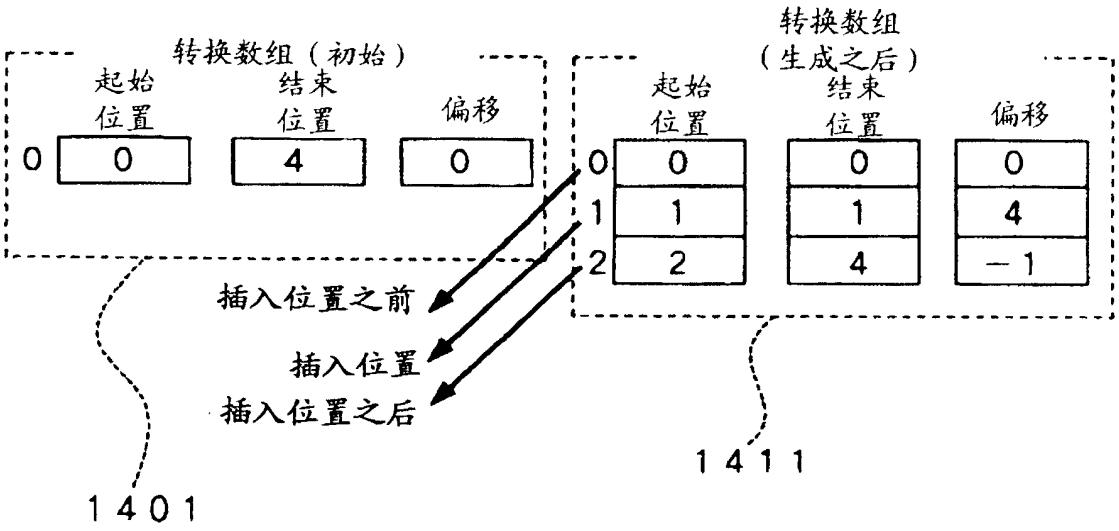
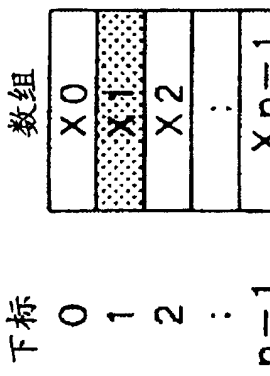
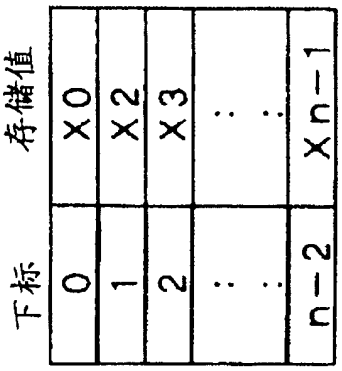

转换数组（初始）
起始位置
结束位置
偏移
0
0
4
0
转换数组（生成之后）
起始位置
结束位置
偏移
0
0
0
0
1
1
4
2
2
4
－1
插入位置之前
插入位置
插入位置之后
1401
1411

图 15A



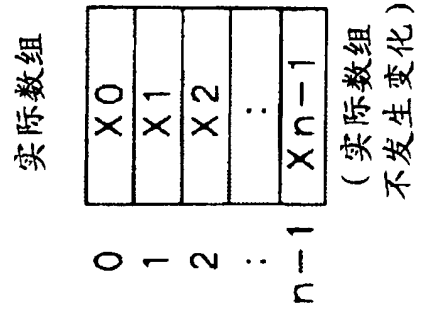
从大小为“n”的数组中删除元素

图 15B



下标和存储值之间的逻辑关系

图 15C



（实际数组不发生变化）

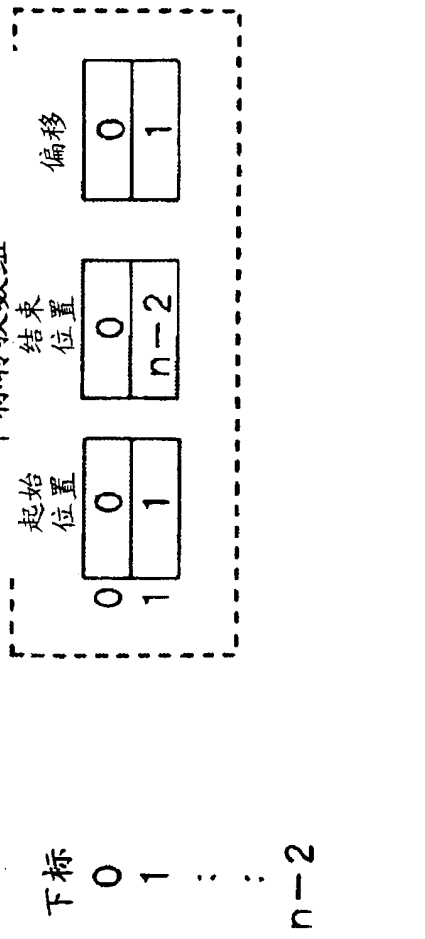


图 16A

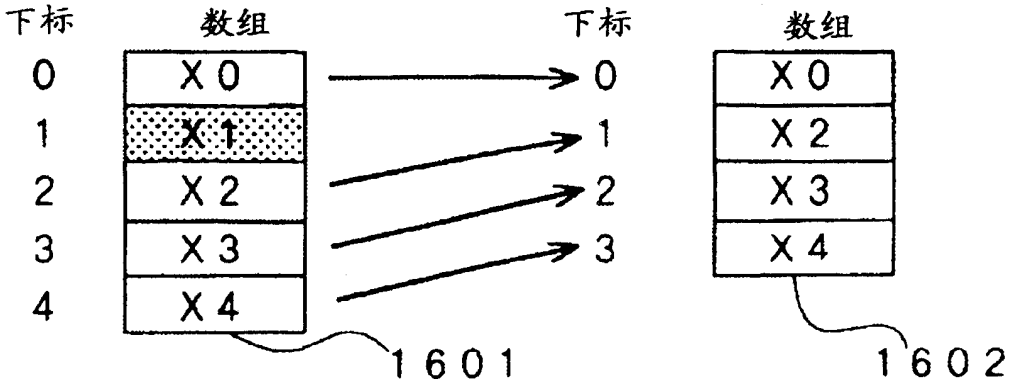
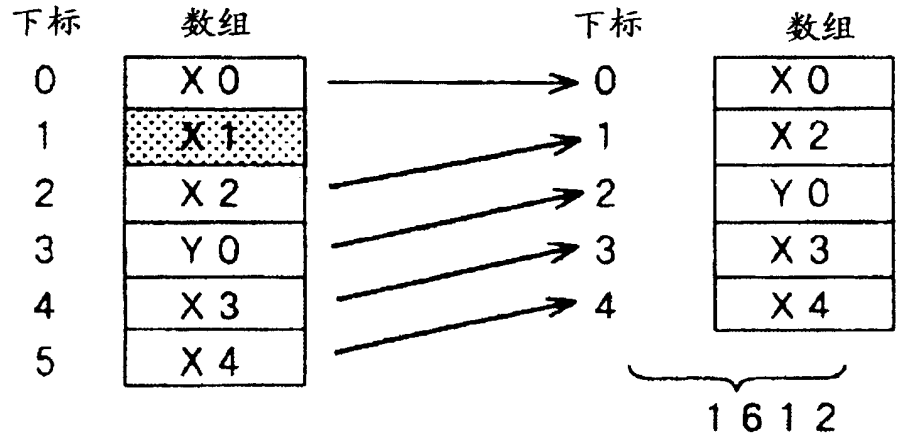

下标
数组
0
1
2
3
4
X0
X1
X2
X3
X4
1601
下标
数组
0
1
2
3
X0
X2
X3
X4
1602

图 16B


下标
数组
0
1
2
3
4
5
X0
X1
X2
Y0
X3
X4
下标
数组
0
1
2
3
4
X0
X2
Y0
X3
X4
1612

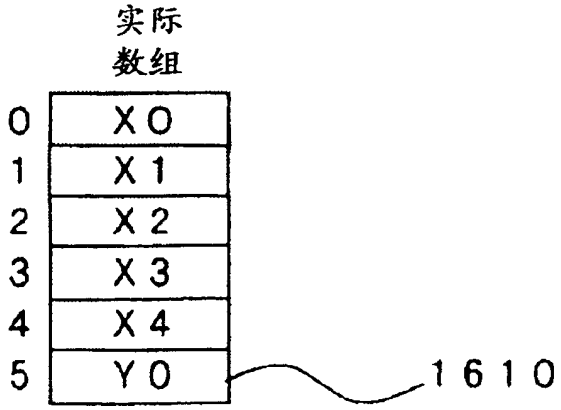
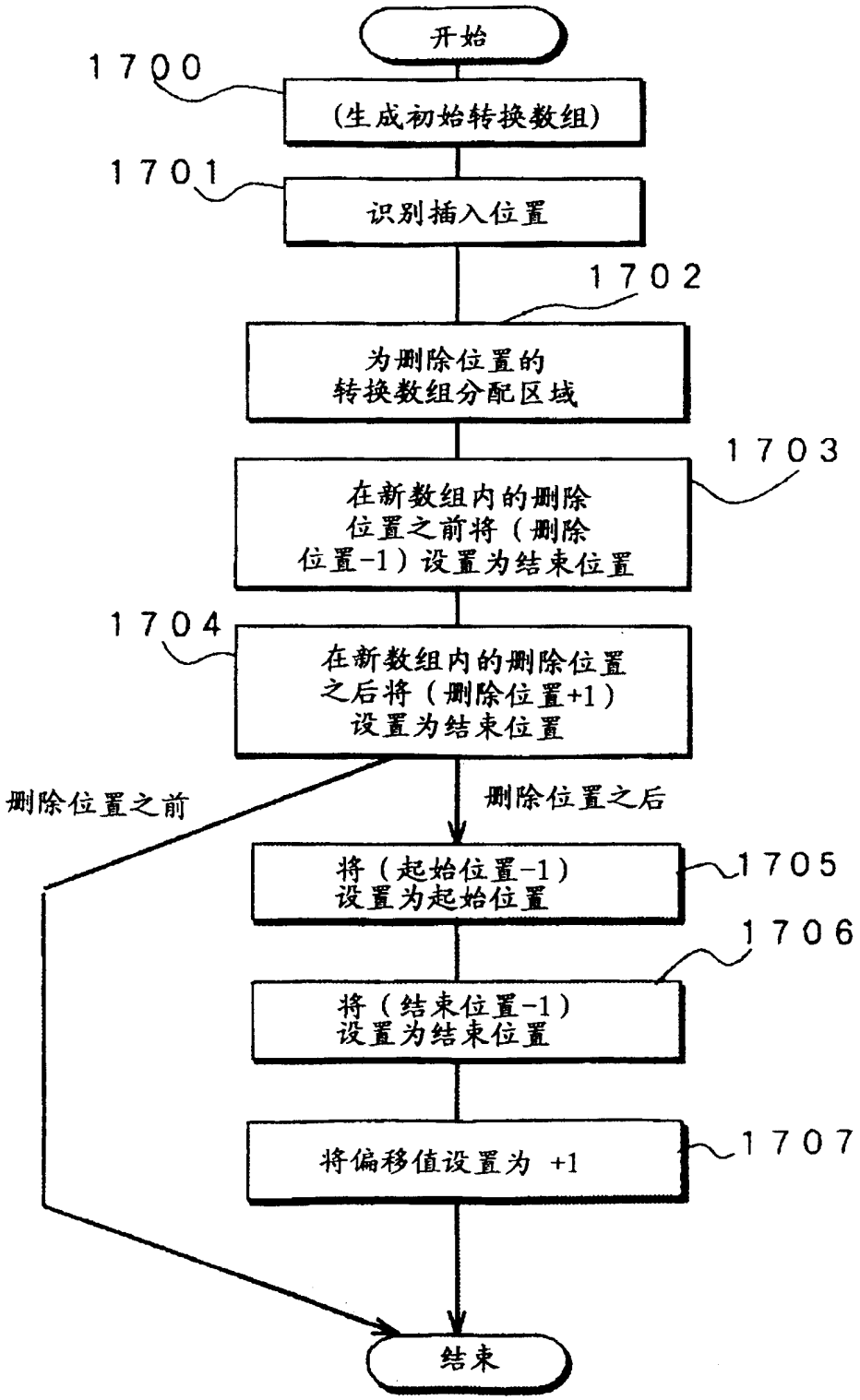

实际
数组
0
1
2
3
4
5
X0
X1
X2
X3
X4
Y0
1610

图 17


开始
1700
(生成初始转换数组)
1701
识别插入位置
1702
为删除位置的
转换数组分配区域
1703
在新数组内的删除
位置之前将（删除
位置-1）设置为结束位置
1704
在新数组内的删除位置
之后将（删除位置+1）
设置为结束位置
删除位置之前
删除位置之后
1705
将（起始位置-1）
设置为起始位置
1706
将（结束位置-1）
设置为结束位置
1707
将偏移值设置为 +1
结束

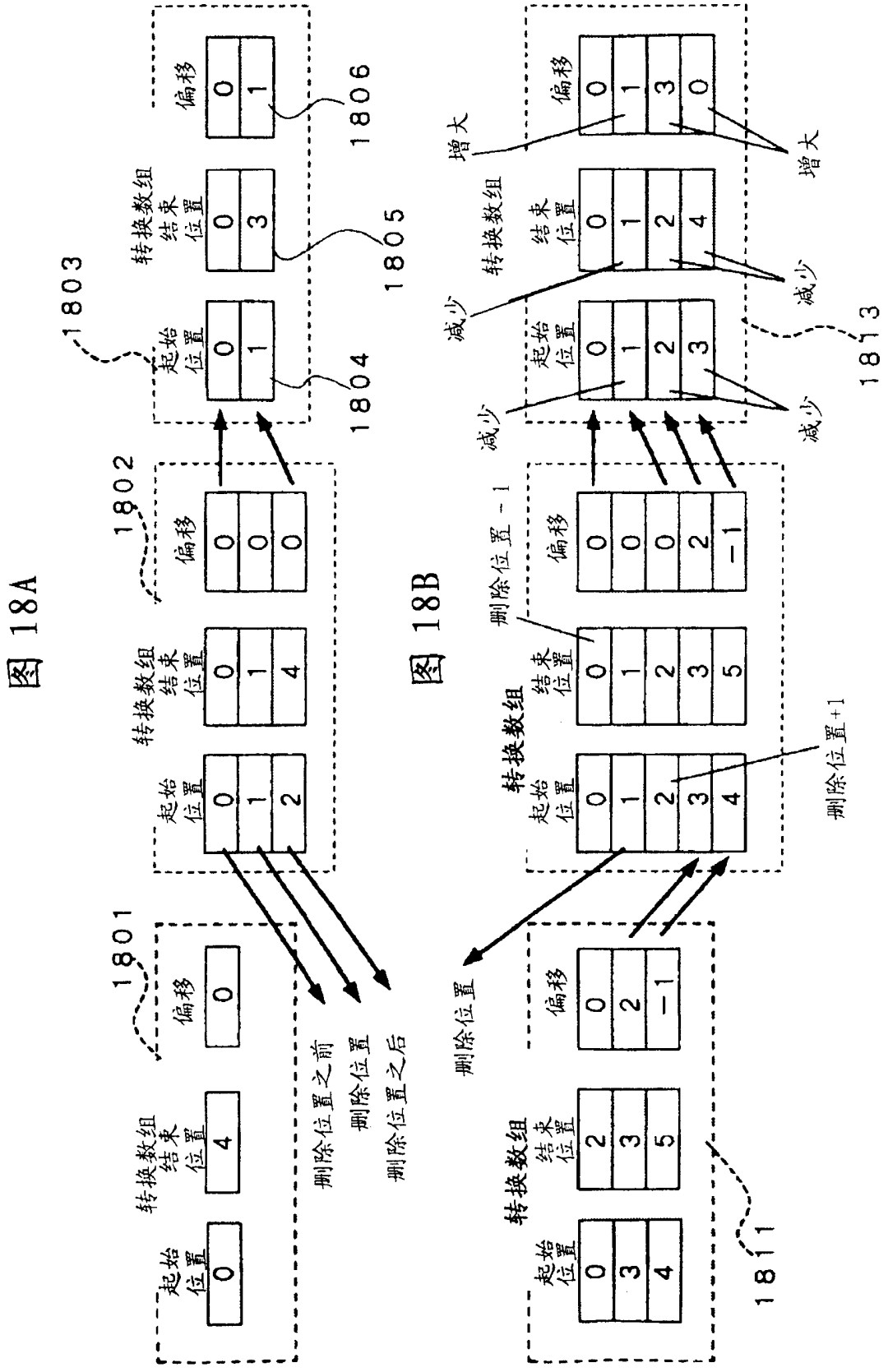


图 18A
1801
1802
1803
1804
1805
1806
起始位置
转换数组结束位置
偏移
删除位置之前
删除位置
删除位置之后
图 18B
1811
1813
转换数组
减少
增大
删除位置 - 1
删除位置 + 1

图 19

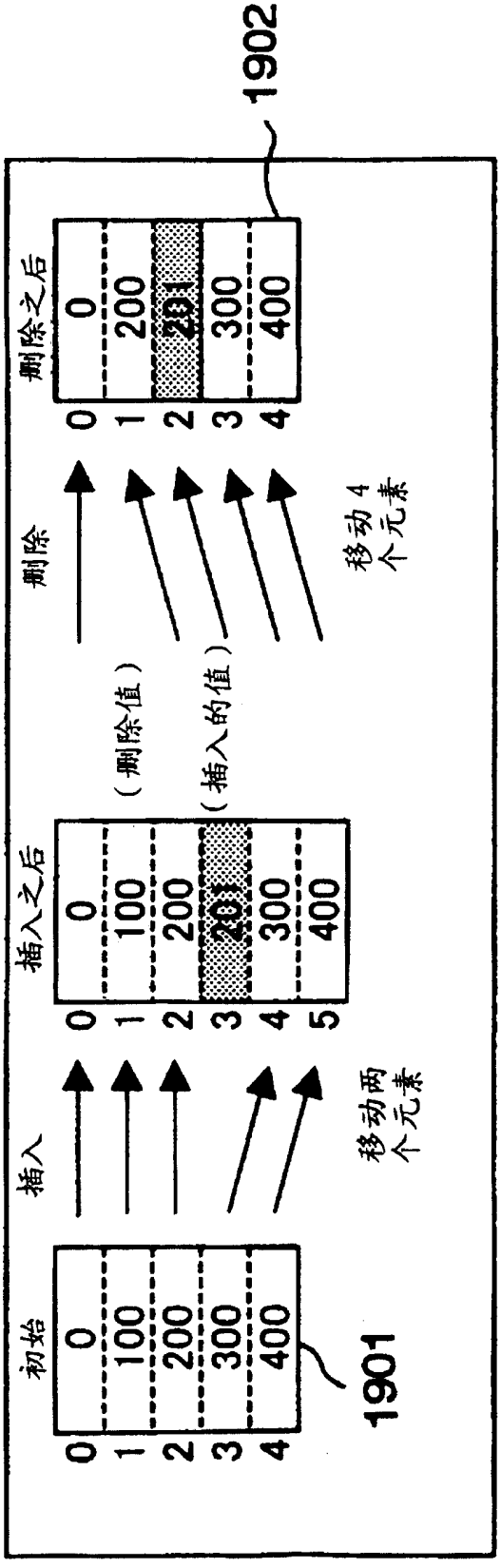

初始
插入
插入之后
删除
删除之后
0
100
200
300
400
201
(删除值)
(插入的值)
移动两个元素
移动4个元素
1901
1902

图 20

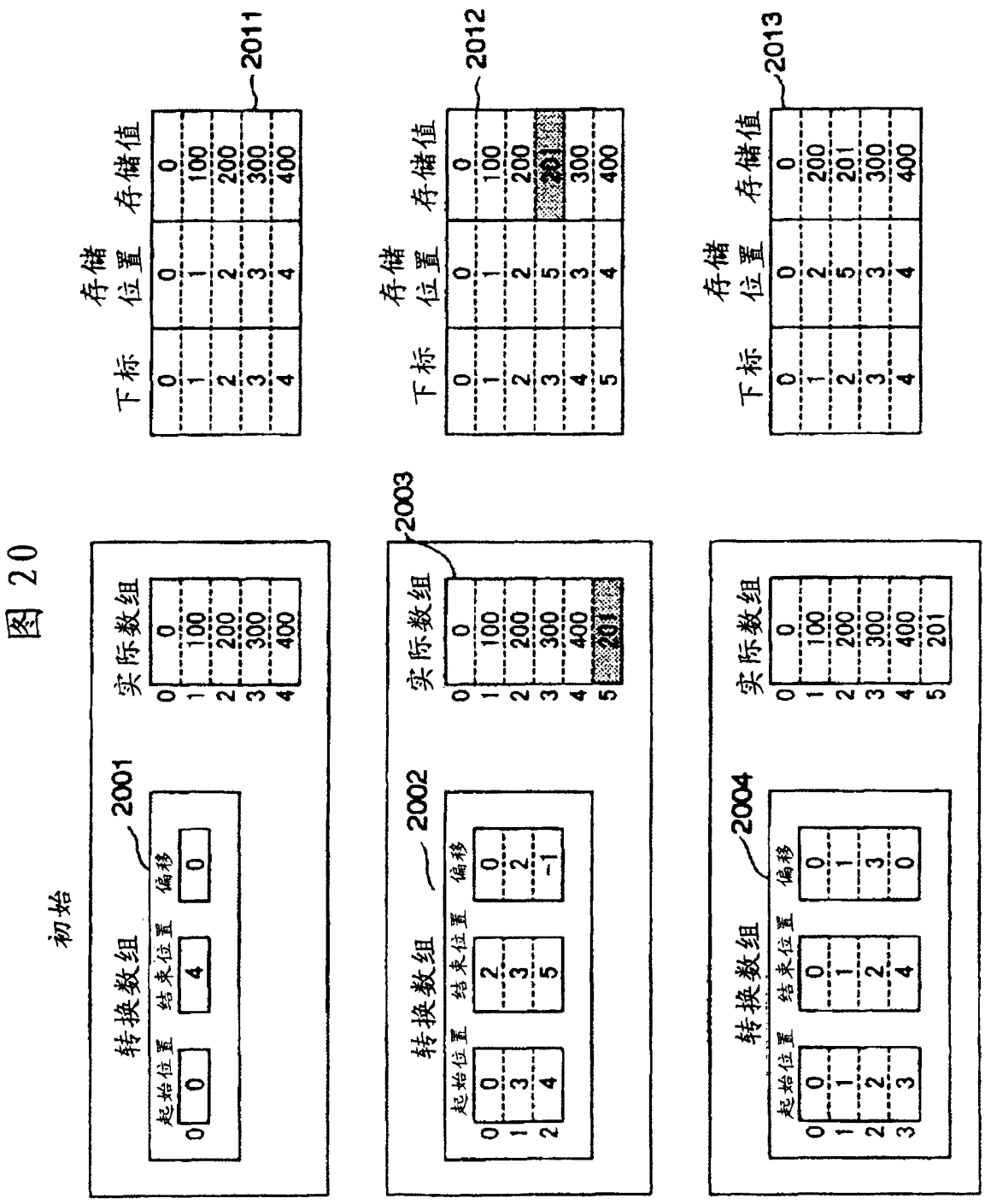

初始
转换数组
起始位置
结束位置
偏移
实际数组
存储
下标
位置
存储值
2001
2002
2003
2004
2011
2012
2013

图 21A

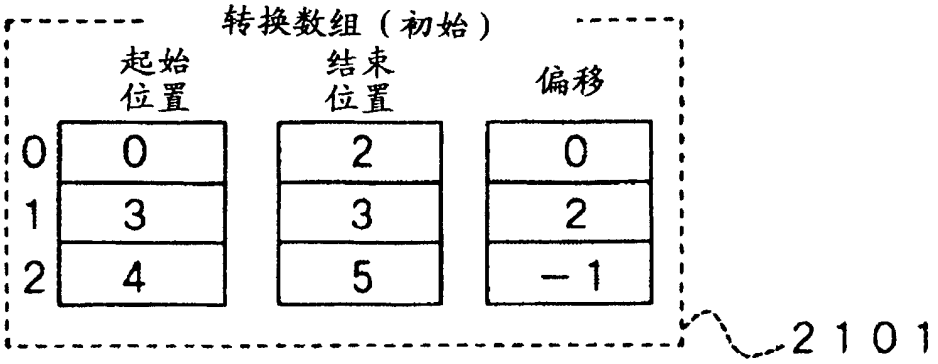
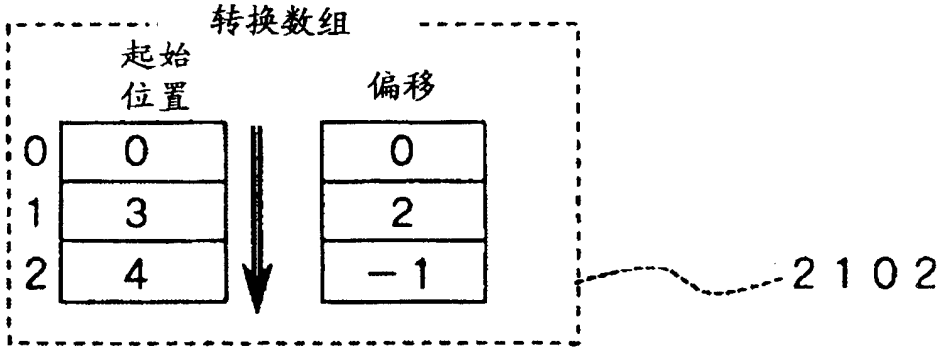
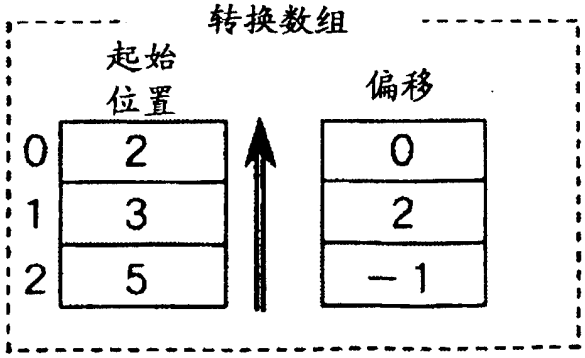


图 21B



（仅结束位置；逻辑：<=）

图 21C



（仅起始位置；逻辑：>=）

图 22

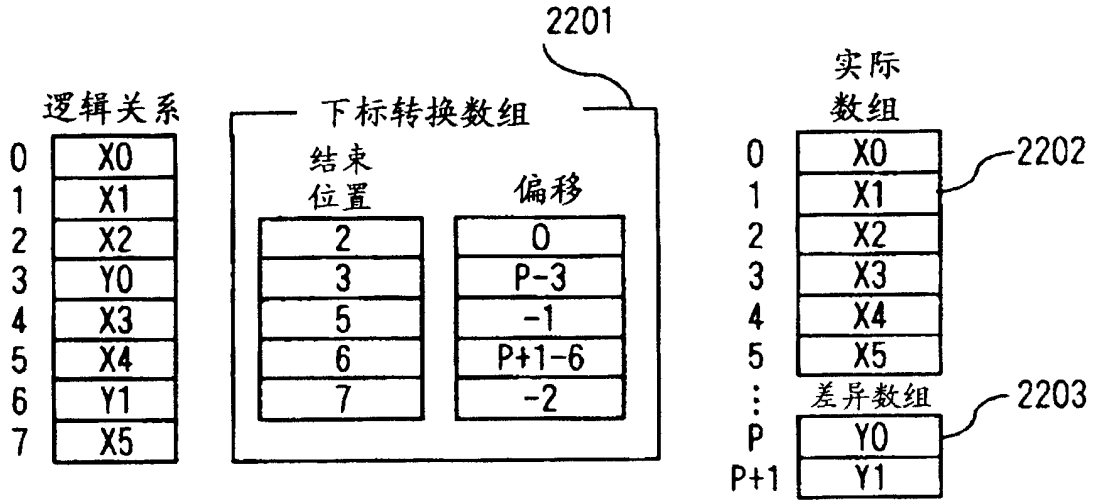
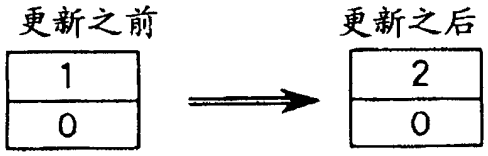


图 23A



在数组中增大值 1 或更多

图 23B

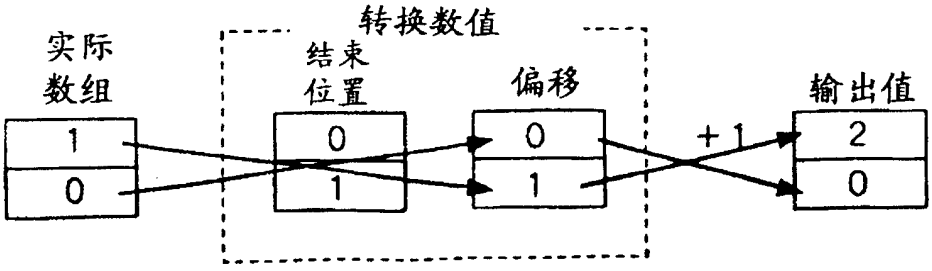


图 24

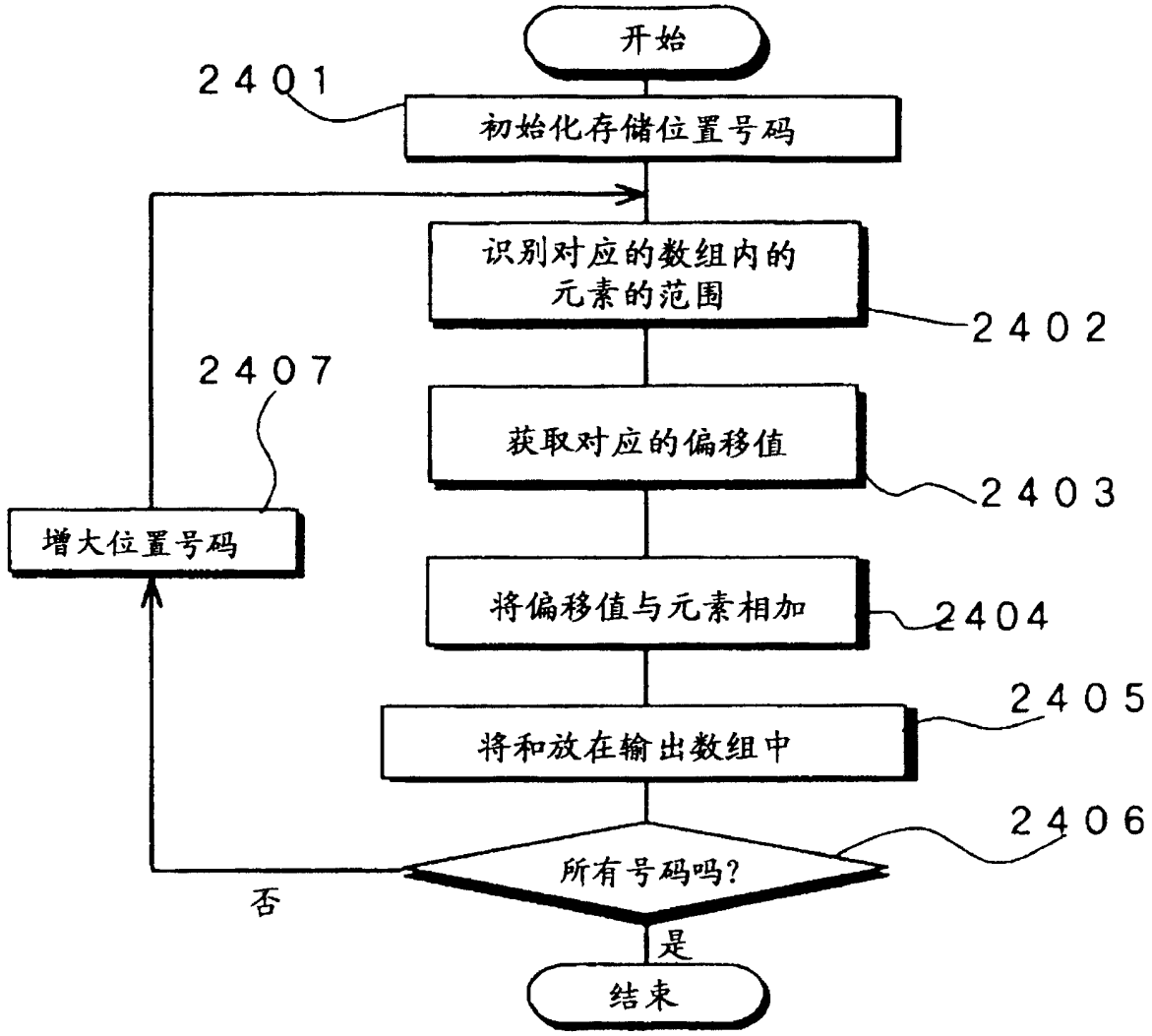

开始
2401
初始化存储位置号码
识别对应的数组内的元素的范围
2402
2407
获取对应的偏移值
2403
增大位置号码
将偏移值与元素相加
2404
2405
将和放在输出数组中
2406
所有号码吗?
否
是
结束

图 25

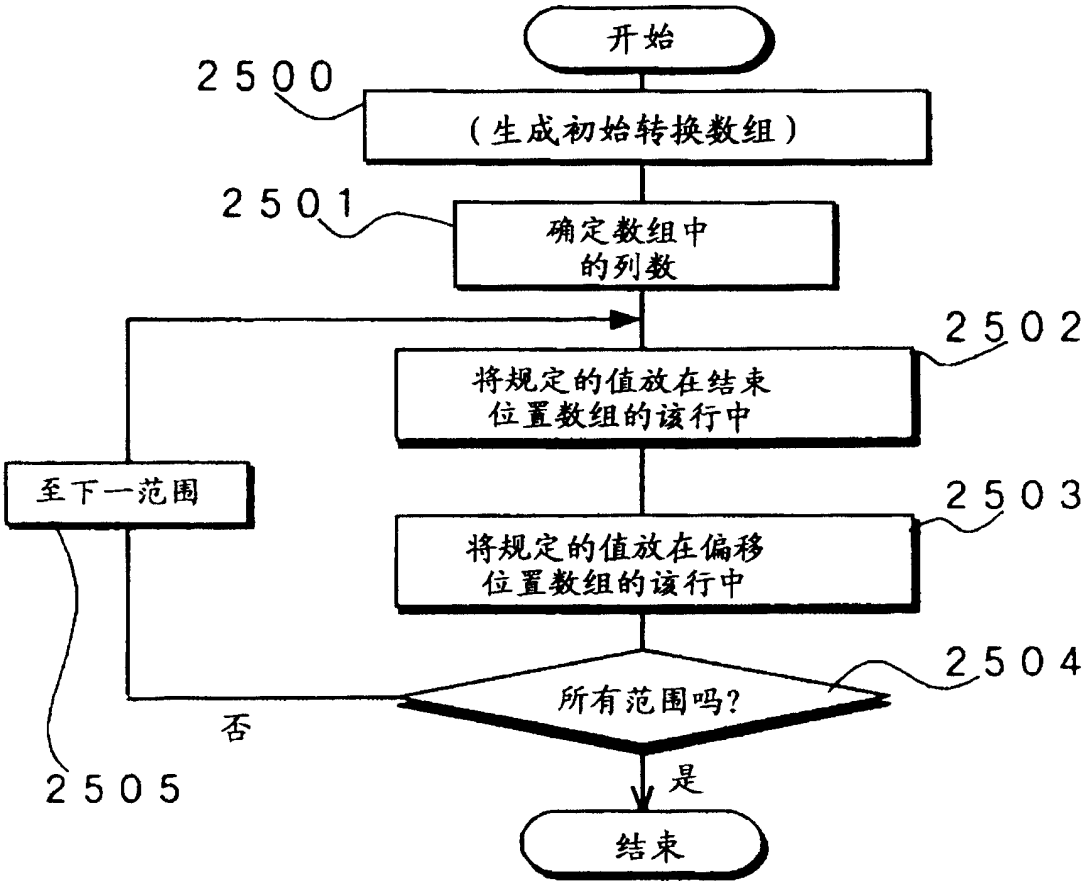

开始
2500
（生成初始转换数组）
2501
确定数组中
的列数
2502
将规定的值放在结束
位置数组的该行中
2503
将规定的值放在偏移
位置数组的该行中
2504
所有范围吗?
否
是
至下一范围
2505
结束

图 26A

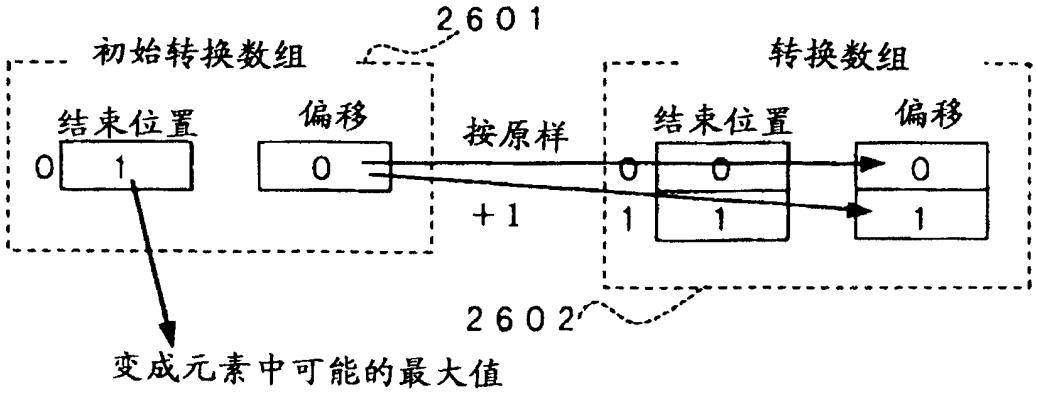

2601
初始转换数组
结束位置
偏移
0
1
0
按原样
+1
转换数组
结束位置
偏移
0
0
0
1
1
1
2602
变成元素中可能的最大值

图 26B

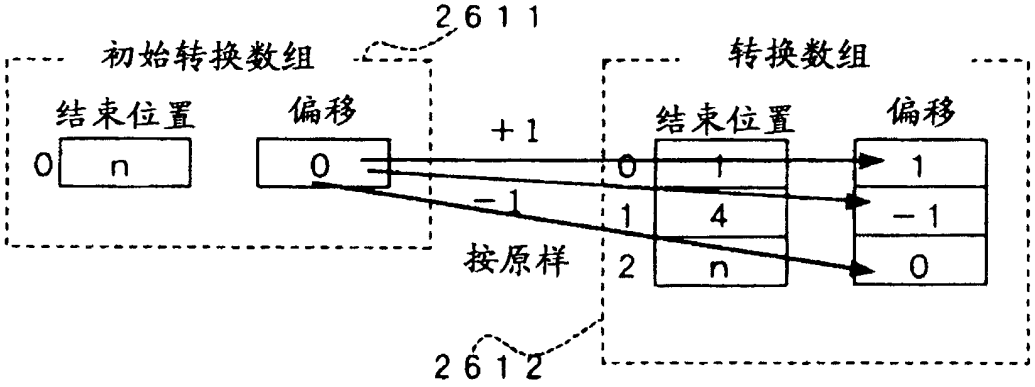
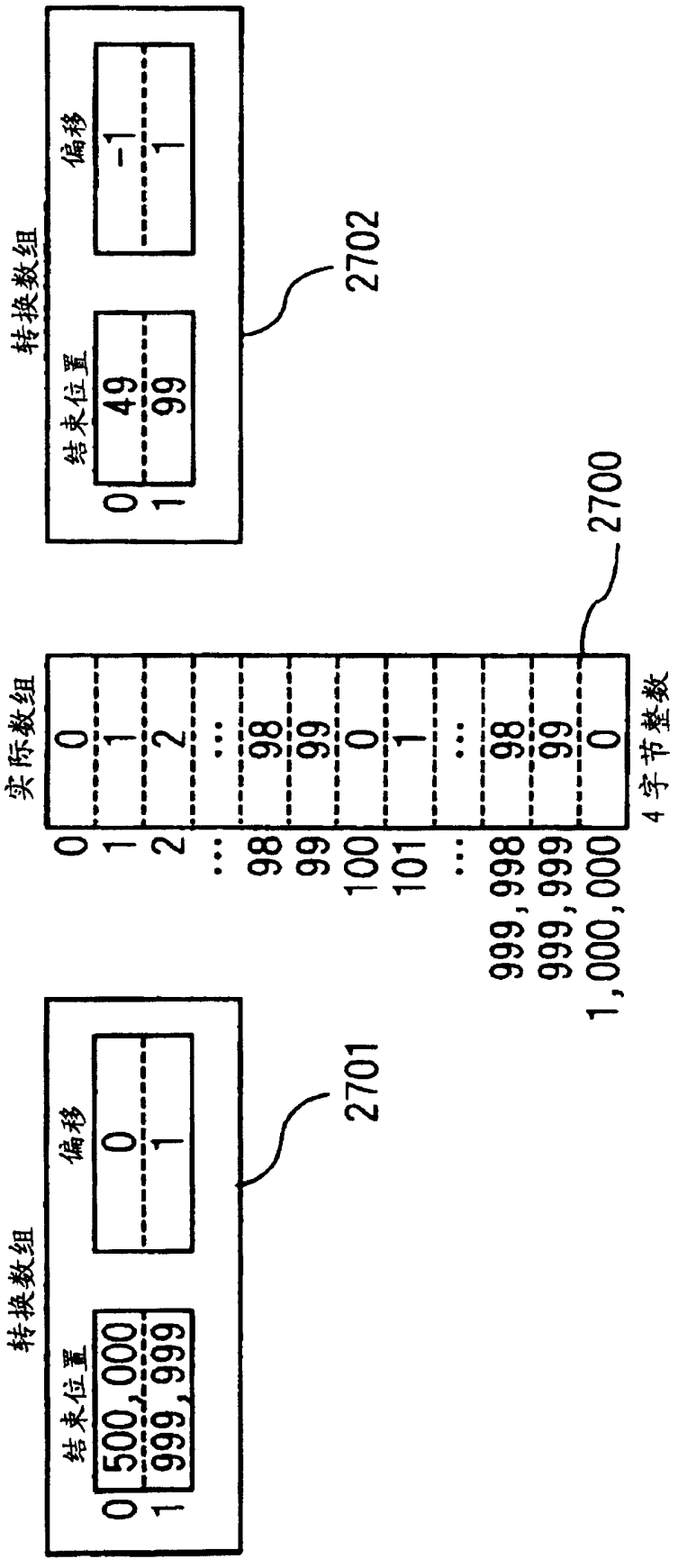

2611
初始转换数组
结束位置
偏移
0
n
0
+1
-1
按原样
转换数组
结束位置
偏移
0
1
1
1
1
4
-1
2
n
0
2612

图 27


转换数组
结束位置
偏移
0 500,000 0
1 999,999 1
2701
实际数组
0 0
1 1
2 2
... ...
98 98
99 99
100 0
101 1
... ...
999,998 98
999,999 99
1,000,000 0
4字节整数
2700
转换数组
结束位置
偏移
0 49 -1
1 99 1
2702

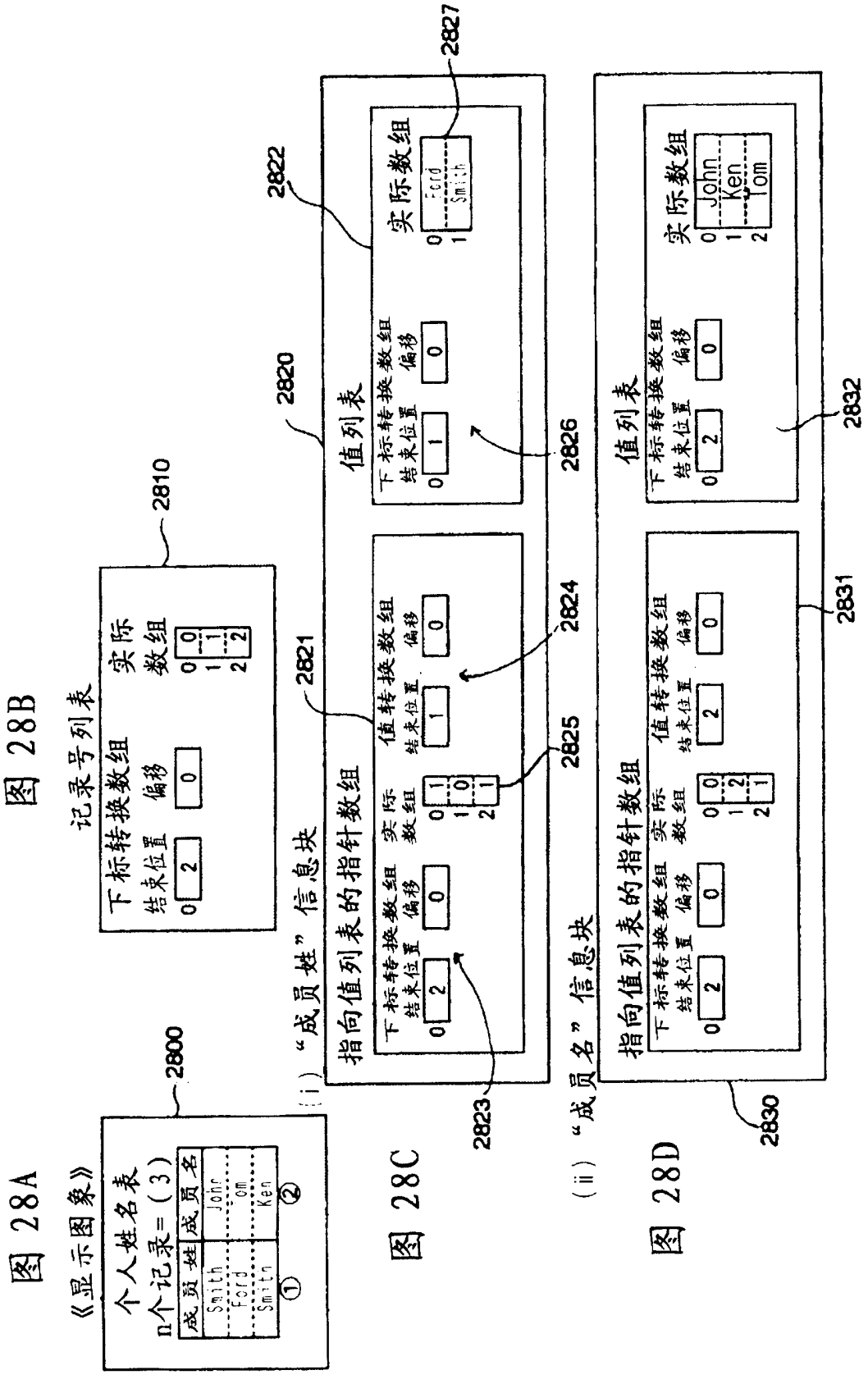
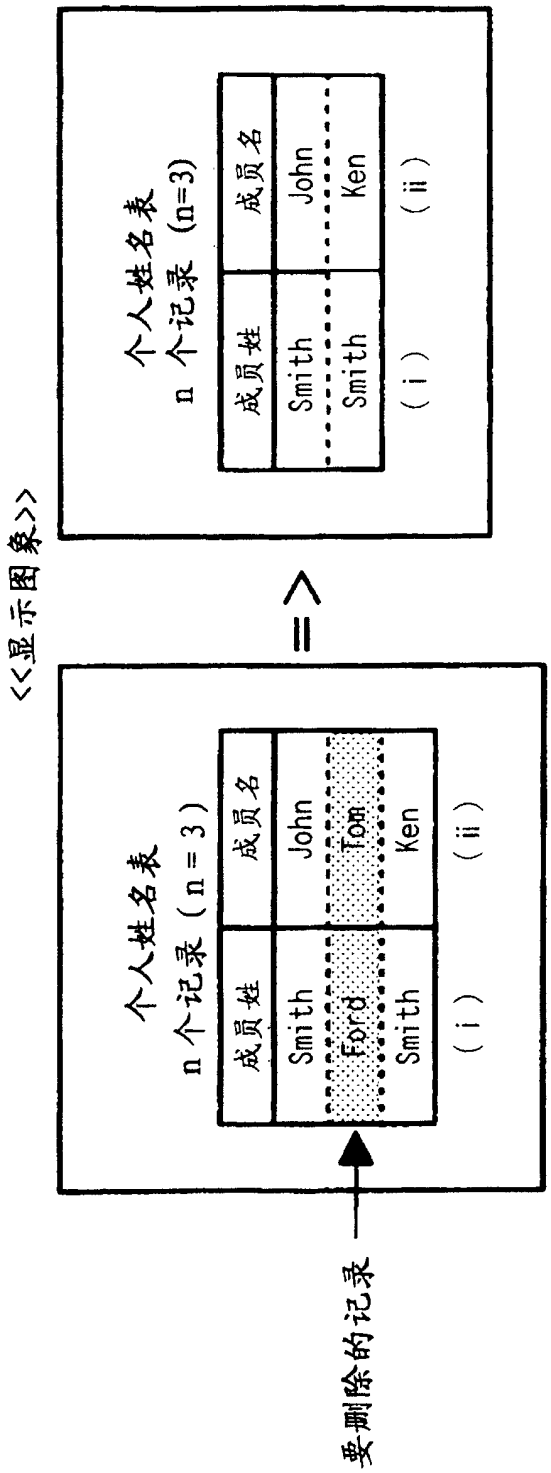


图 28A
《显示图象》
个人姓名表
n个记录=（3）
成员姓
成员名
Smith
John
Ford
Tom
Smith
Ken
①
②
2800
图 28B
记录号列表
下标转换数组
结束位置
偏移
实际数组
2810
2821
“成员姓”信息块
指向值列表的指针数组
2823
2824
2825
2820
值列表
2822
2826
2827
实际数组
Ford
Smith
图 28C
“成员名”信息块
指向值列表的指针数组
2830
2831
值列表
2832
实际数组
John
Ken
Tom
图 28D

图 29A



<<显示图象>>

图 29B

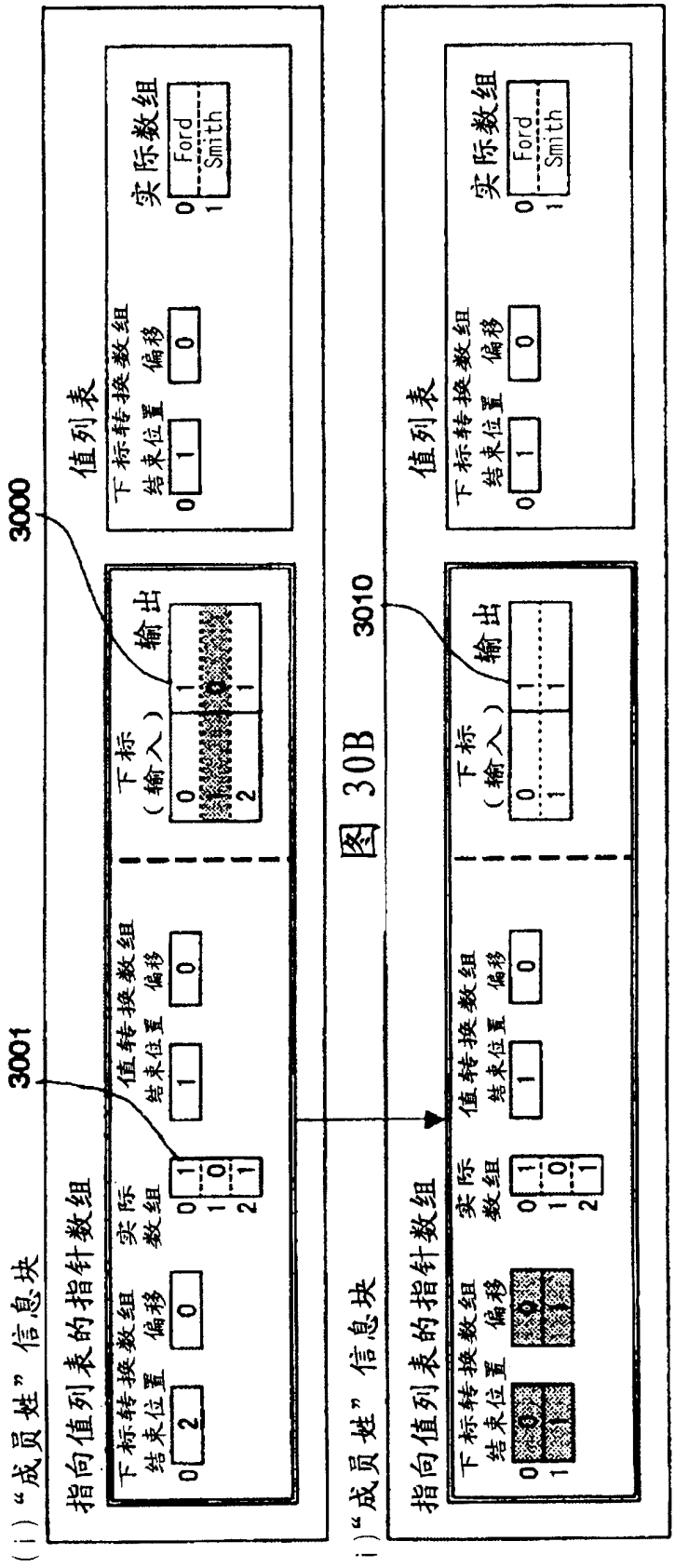

(i)“成员姓”信息块
指向值列表的指针数组
下标转换数组
结束位置
偏移
实际数组
值转换数组
结束位置
偏移
下标（输入）
输出
值列表
下标转换数组
结束位置
偏移
实际数组
Ford
Smith
3000
3001
3010

图 30A

图 30B

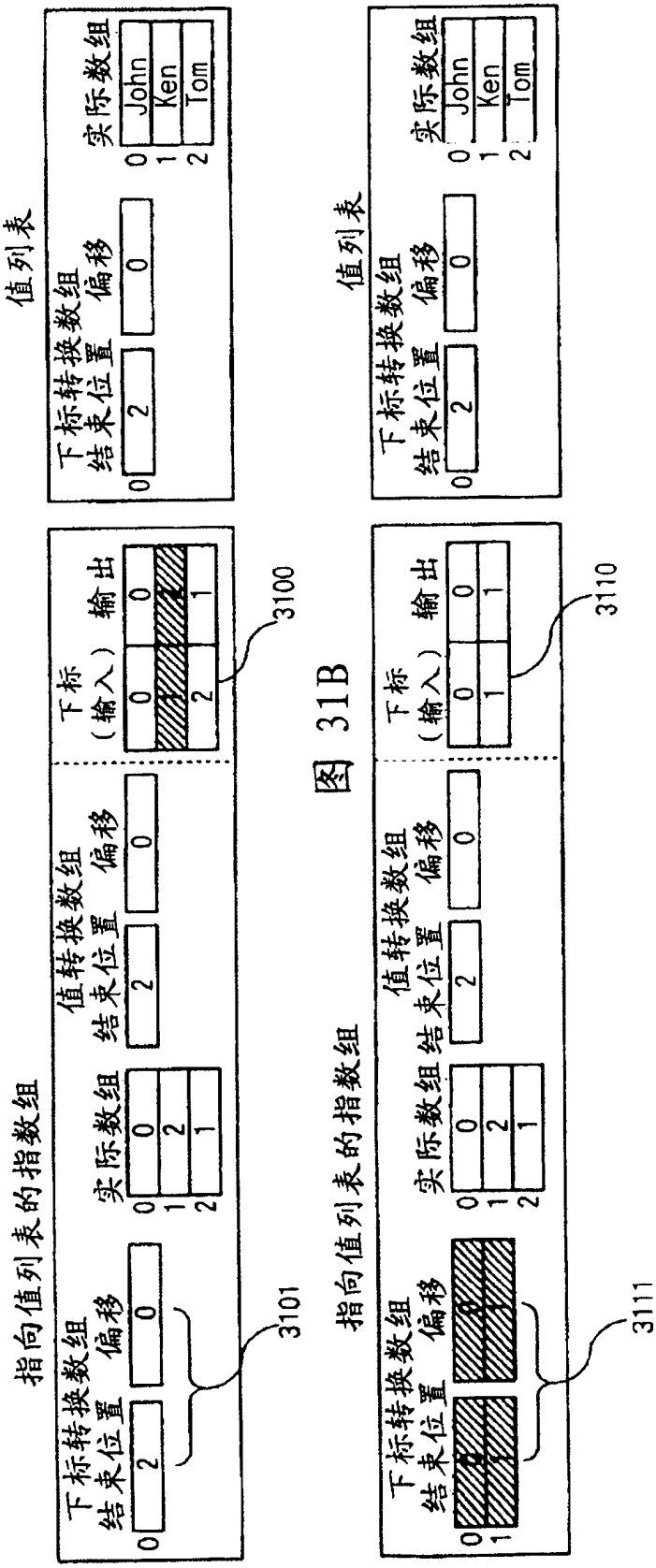

值列表
下标转换数组
结束位置 偏移
0 2 0
实际数组
0 John
1 Ken
2 Tom
指向值列表的指数组
值转换数组
结束位置 偏移
0 2 0
实际数组
0 0
1 2
2 1
下标
（输入） 输出
0 0
1 2
2 1
3100
3101

图 31A

值列表
下标转换数组
结束位置 偏移
0 2 0
实际数组
0 John
1 Ken
2 Tom
指向值列表的指数组
值转换数组
结束位置 偏移
0 2 0
实际数组
0 0
1 2
2 1
下标
（输入） 输出
0 0
1 1
3110
3111

图 31B

图 32

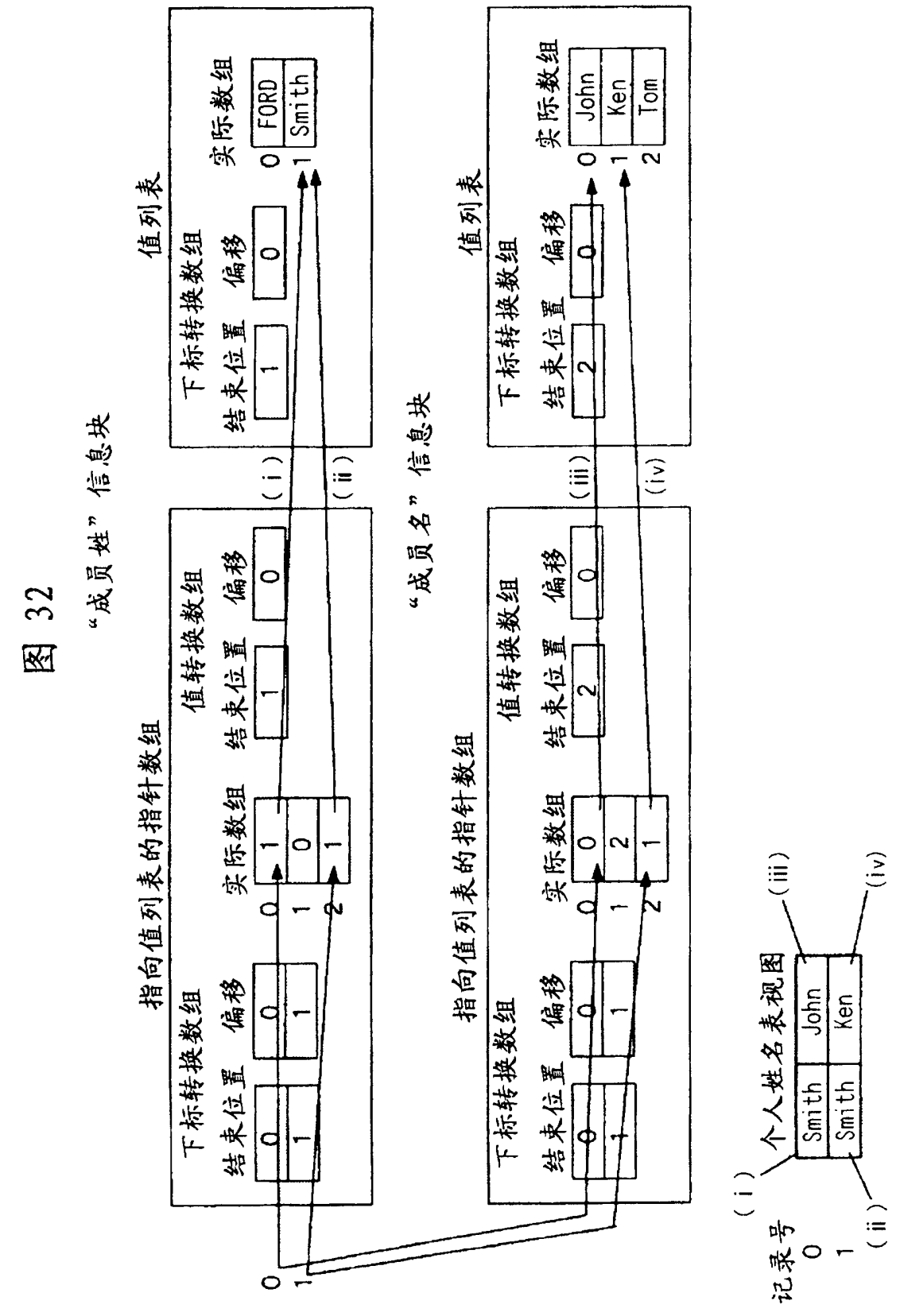
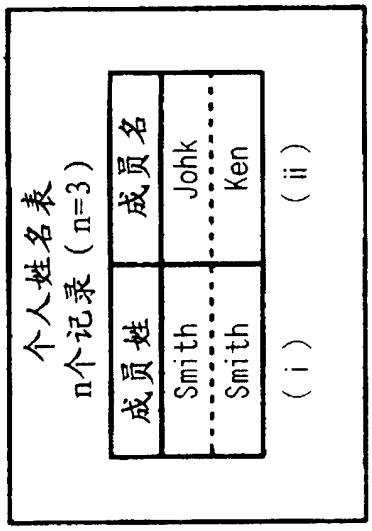
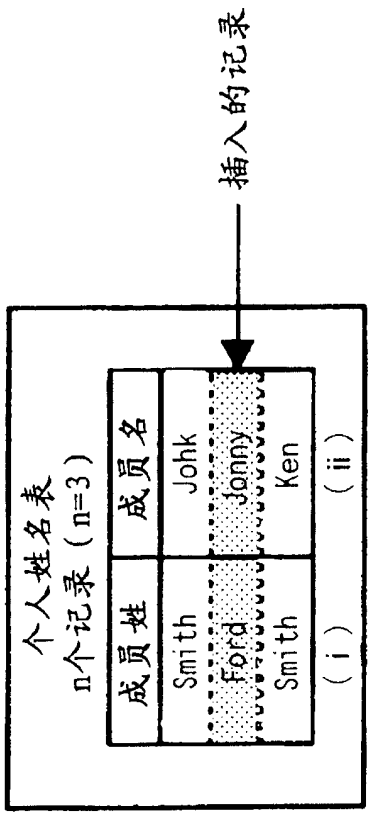

“成员姓”信息块
值列表
下标转换数组
结束位置
偏移
实际数组
FORD
Smith
(i)
(ii)
指向值列表的指针数组
值转换数组
“成员名”信息块
John
Ken
Tom
(iii)
(iv)
个人姓名表视图
记录号
Smith
John
Smith
Ken

图33A



《显示图象》

图33B



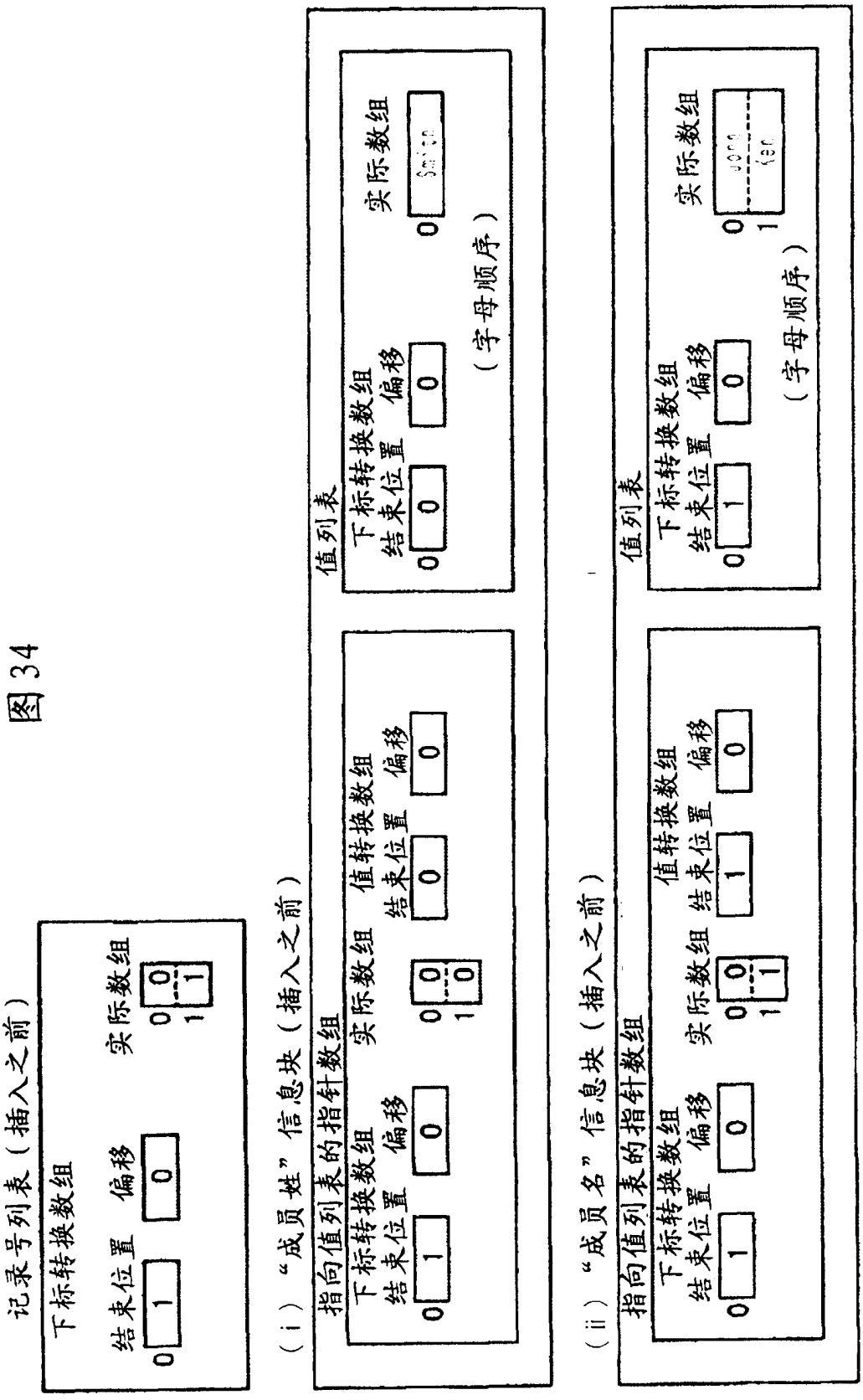

记录号列表（插入之前）
下标转换数组
结束位置
偏移
实际数组
（i）“成员姓”信息块（插入之前）
值列表
指向值列表的指针数组
下标转换数组
结束位置
偏移
实际数组
值转换数组
结束位置
偏移
（字母顺序）
（ii）“成员名”信息块（插入之前）
值列表
指向值列表的指针数组
下标转换数组
结束位置
偏移
实际数组
值转换数组
结束位置
偏移
（字母顺序）

图34

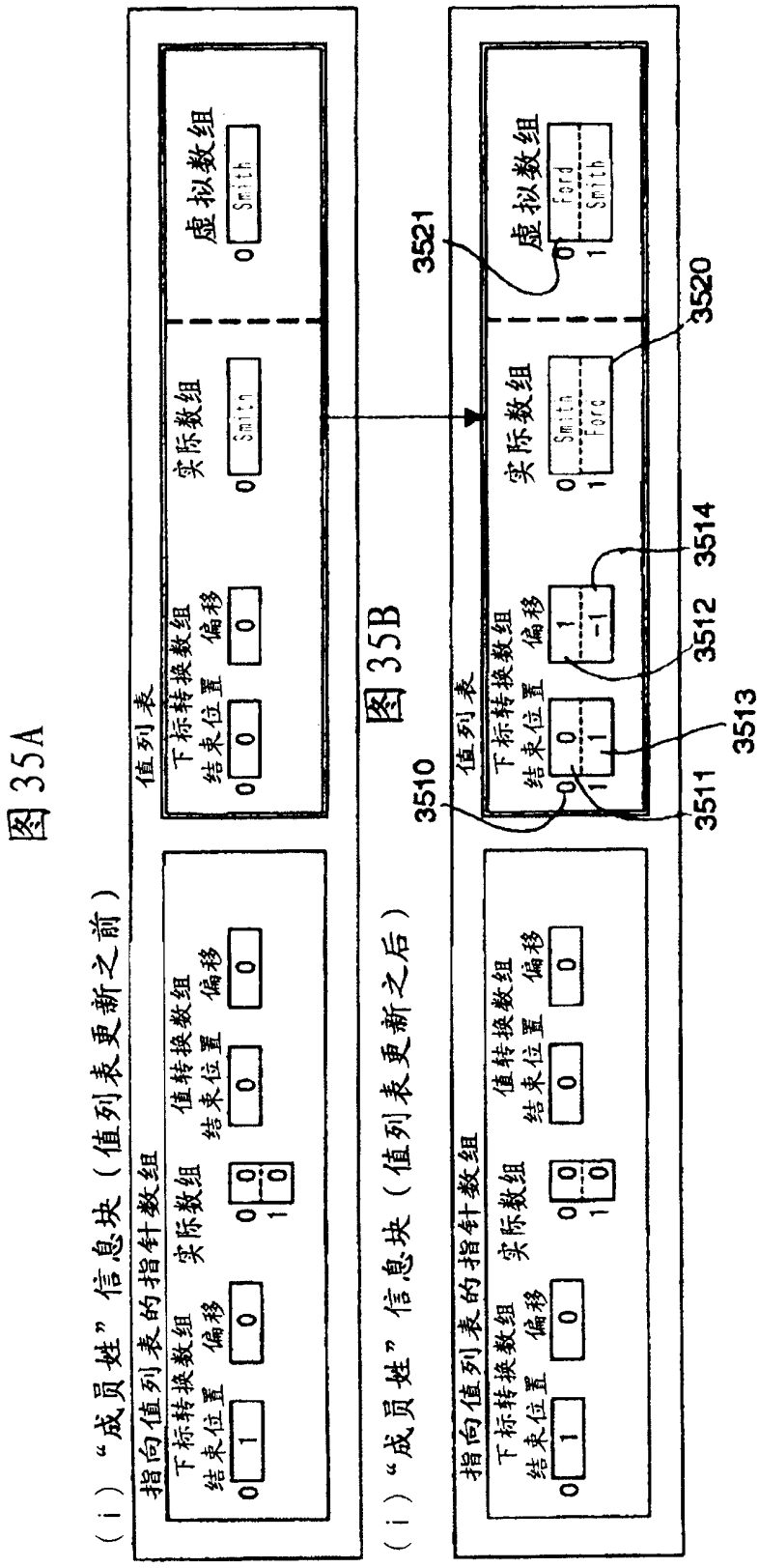


图35A
（i）“成员姓”信息块（值列表更新之前）
指向值列表的指针数组
下标转换数组
结束位置
偏移
实际数组
虚拟数组
值列表
Smith
图35B
（i）“成员姓”信息块（值列表更新之后）
3510
3511
3512
3513
3514
3520
3521
Ford

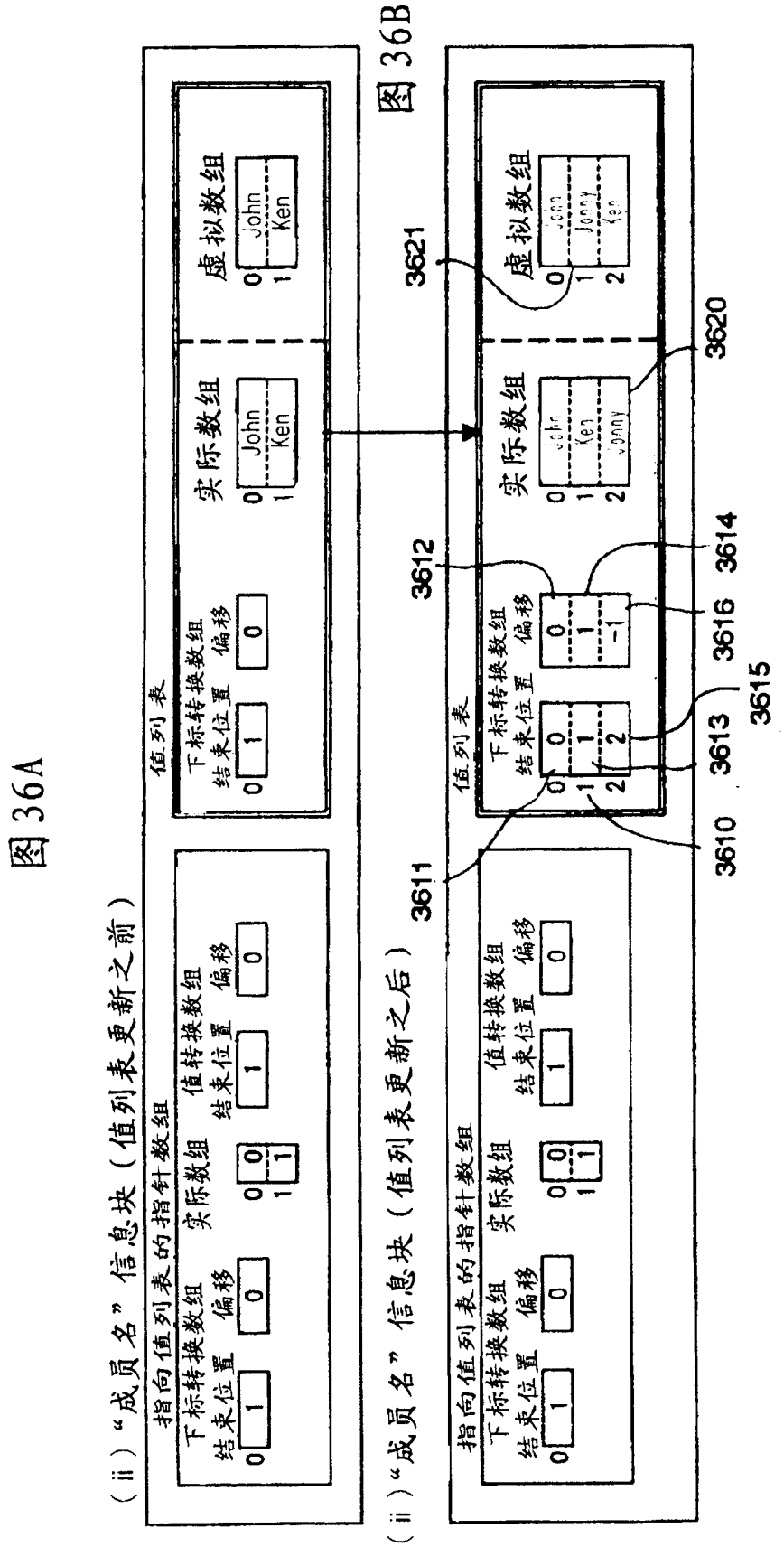


图36A
（ii）“成员名”信息块（值列表更新之前）
指向值列表的指针数组
下标转换数组
结束位置
偏移
实际数组
值转换数组
结束位置
偏移
值列表
虚拟数组
John
Ken
图36B
（ii）“成员名”信息块（值列表更新之后）
3610
3611
3612
3613
3614
3615
3616
3620
3621
Johnny

图 37A

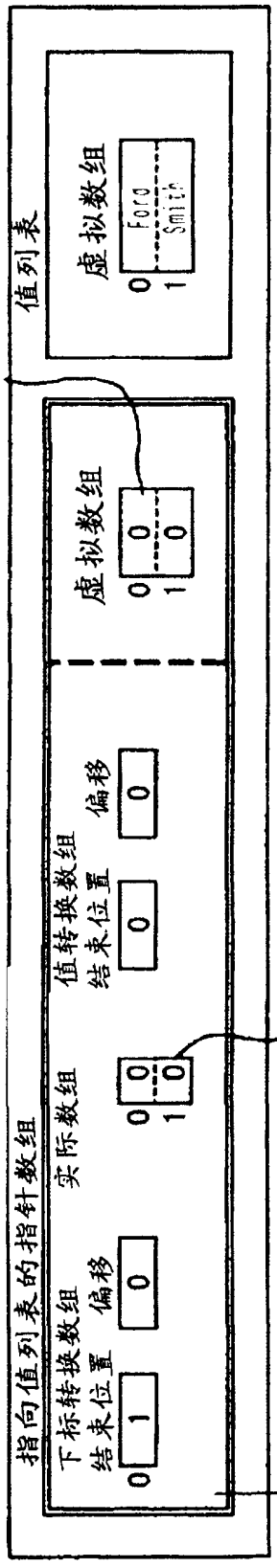

指向值列表的指针数组
下标转换数组
结束位置
偏移
实际数组
值转换数组
结束位置
偏移
虚拟数组
值列表
虚拟数组
Ford
Smith
3701
3702

图 37B

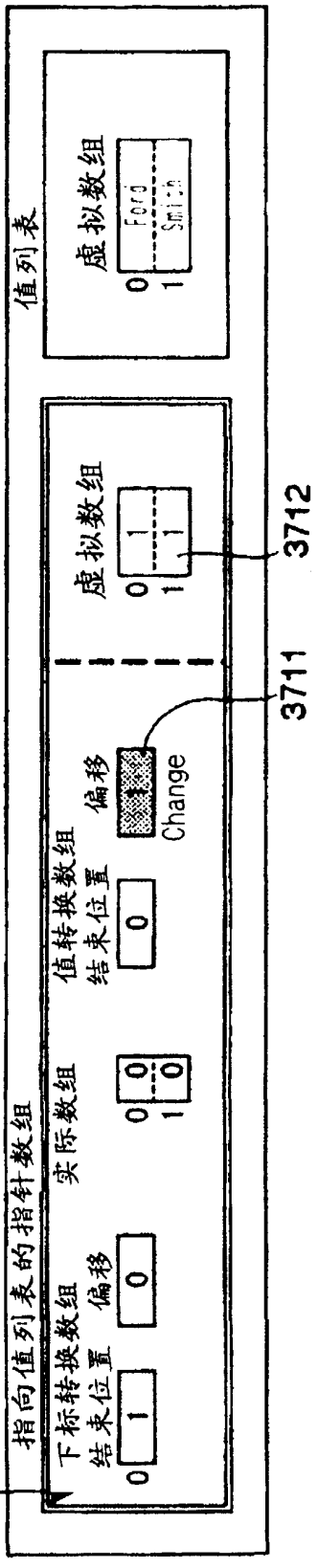

（i）“成员姓”信息块（更新一个指向值列表的指针数组之后）
指向值列表的指针数组
下标转换数组
结束位置
偏移
实际数组
值转换数组
结束位置
偏移
虚拟数组
值列表
虚拟数组
Ford
Smith
Change
3711
3712

图38A

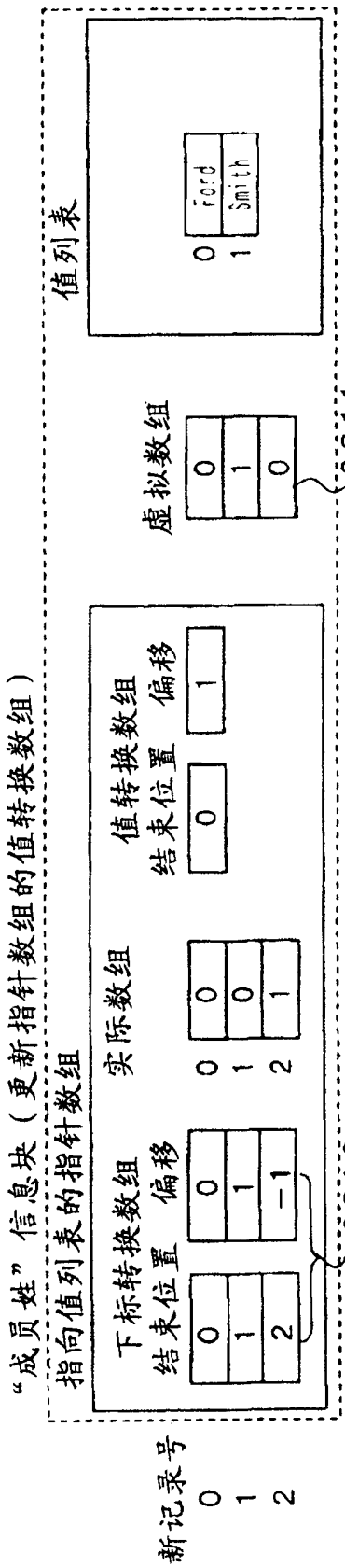

“成员姓”信息块（更新指针数组的值转换数组）
指向值列表的指针数组
下标转换数组
结束位置 偏移
0 0
1 1
2 -1
实际数组
0 0
1 0
2 1
值转换数组
结束位置 偏移
0 1
虚拟数组
0
1
0
值列表
0 Ford
1 Smith
新记录号
0
1
2
3810
3811

图38B

“成员姓”信息块（更新指针数组的值转换数组）
指向值列表的指针数组
下标转换数组
结束位置 偏移
0 0
1 1
2 -1
实际数组
0 0
1 0
2 1
值转换数组
结束位置 偏移
0 1
1 -1
虚拟数组
1
0
1
值列表
虚拟（逻辑）
组数
0 Ford
1 Smith
新记录号
0
1
2
3820
3821

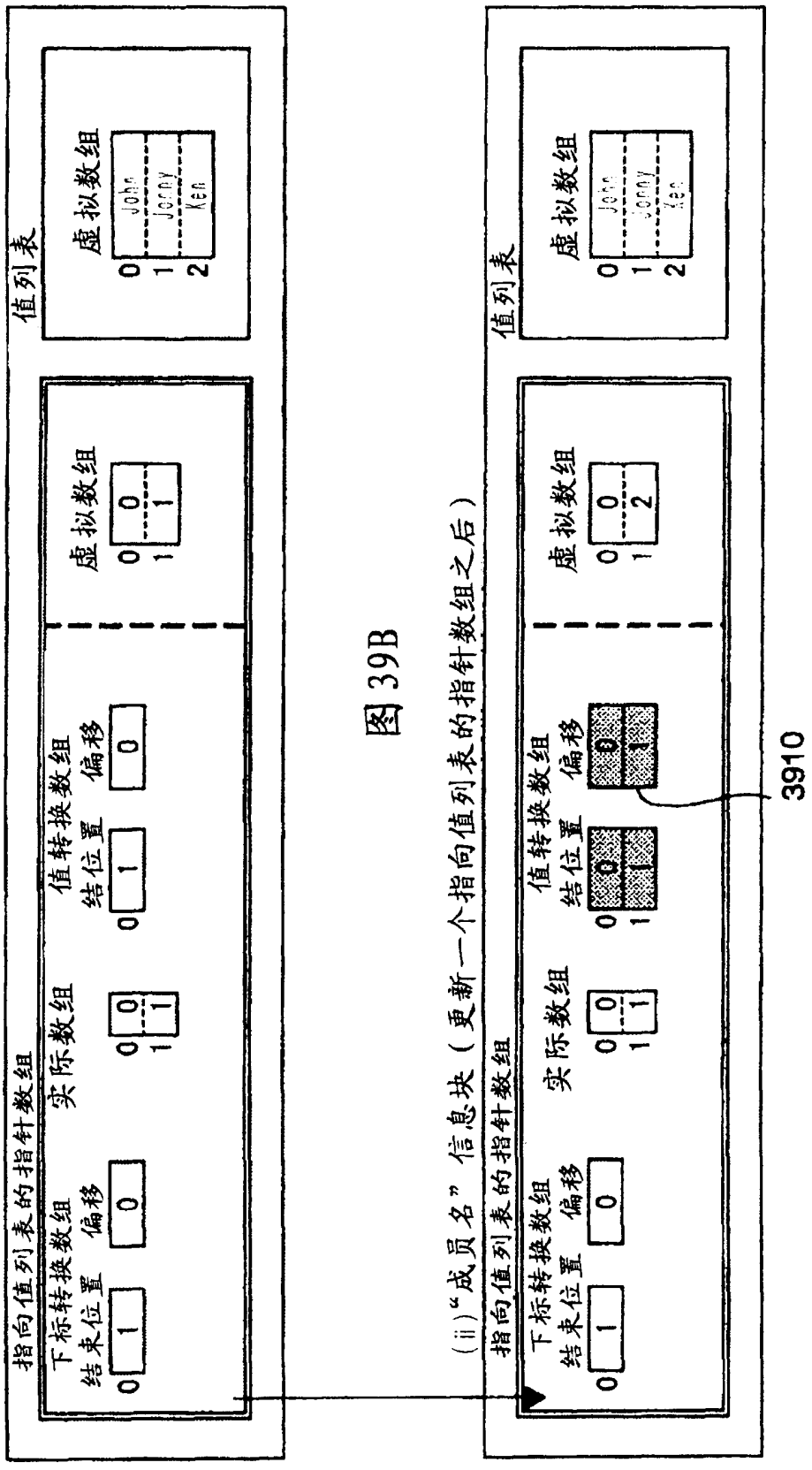

指向值列表的指针数组
下标转换数组
结束位置
偏移
实际数组
值转换数组
结束位置
偏移
虚拟数组
值列表
虚拟数组
John
Jenny
Ken
(ii)“成员名”信息块（更新一个指向值列表的指针数组之后）
3910

图 39A

图 39B

图40A

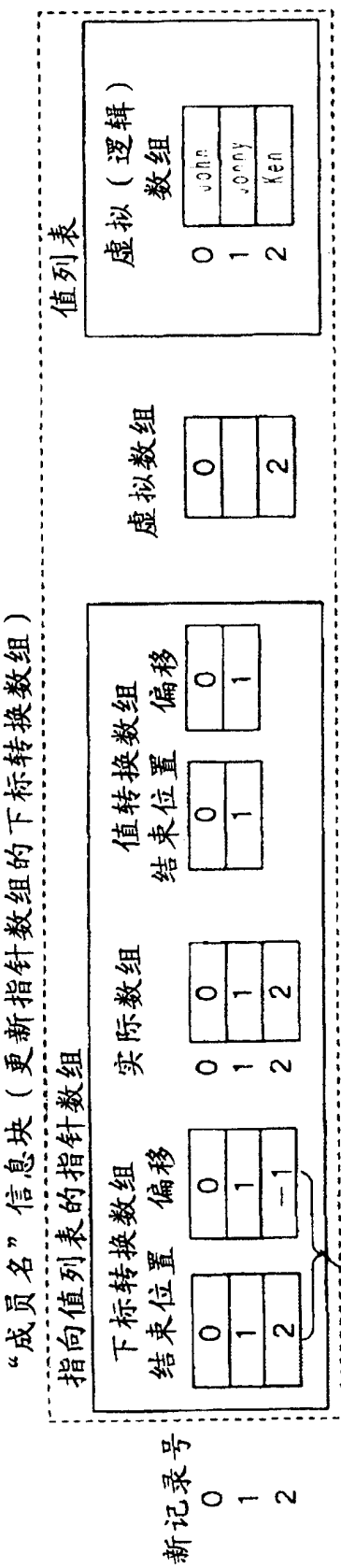

"成员名"信息块(更新指针数组的下标转换数组)
指向值列表的指针数组
下标转换数组
结束位置 偏移
实际数组
值转换数组
结束位置 偏移
虚拟数组
值列表
虚拟(逻辑)数组
John
Jonny
Ken
4010
新记录号

图40B

"成员名"信息块(更新指针数组的下标转换数组)
指向值列表的指针数组
下标转换数组
结束位置 偏移
实际数组
值转换数组
结束位置 偏移
虚拟数组
值列表
虚拟(逻辑)数组
John
Jonny
Ken
4021
4020
新记录号

图41

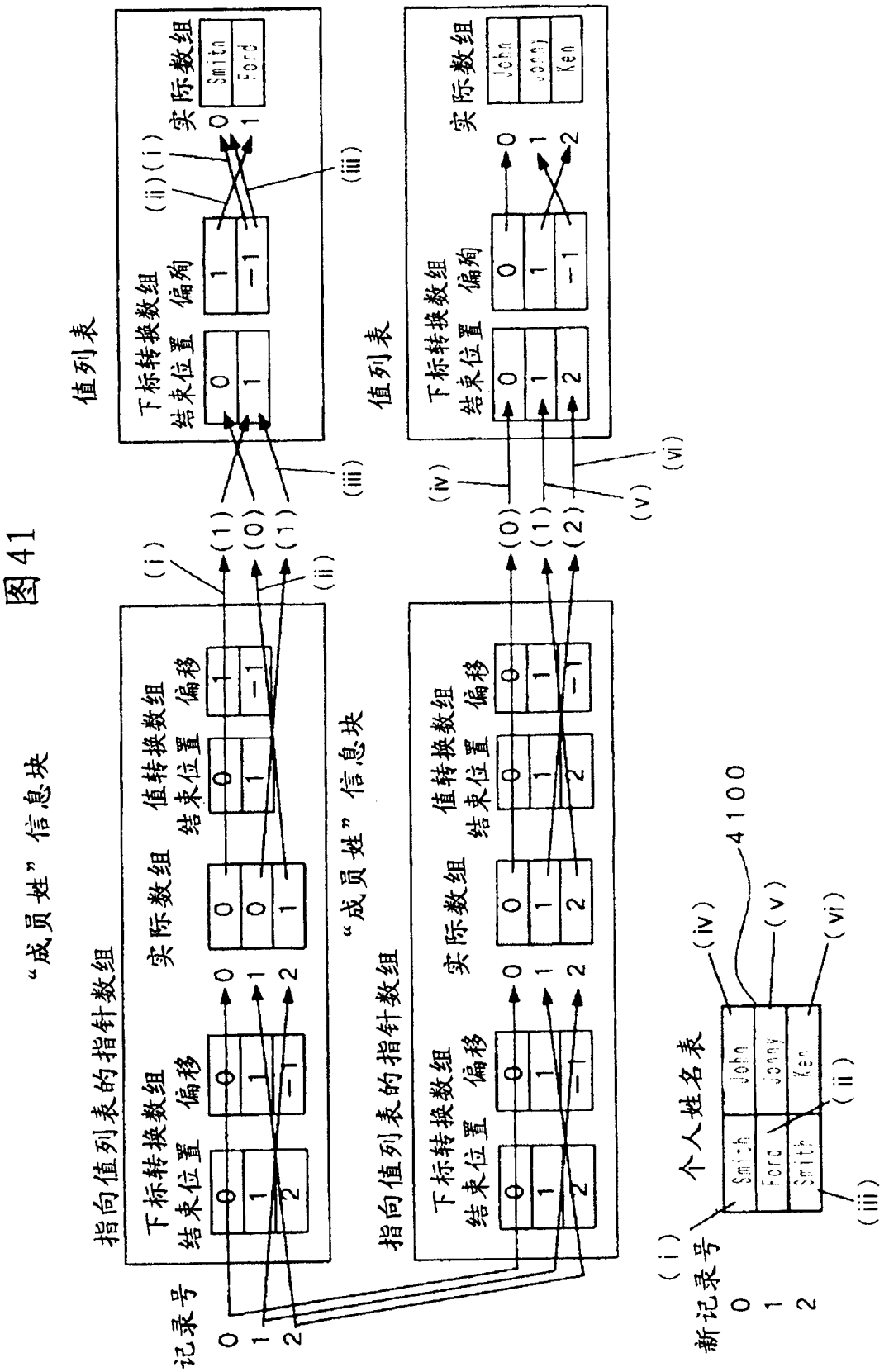

“成员姓”信息块
指向值列表的指针数组
下标转换数组
结束位置 偏移
实际数组
值转换数组
结束位置 偏移
新记录号
值列表
下标转换数组
结束位置 偏移
实际数组
Smith
Ford
“成员姓”信息块
指向值列表的指针数组
下标转换数组
结束位置 偏移
实际数组
值转换数组
结束位置 偏移
值列表
下标转换数组
结束位置 偏移
实际数组
John
Jonny
Ken
个人姓名表
新记录号
4100
Smith
Ford
Smith
John
Jonny
Ken
(i)
(ii)
(iii)
(iv)
(v)
(vi)

图42

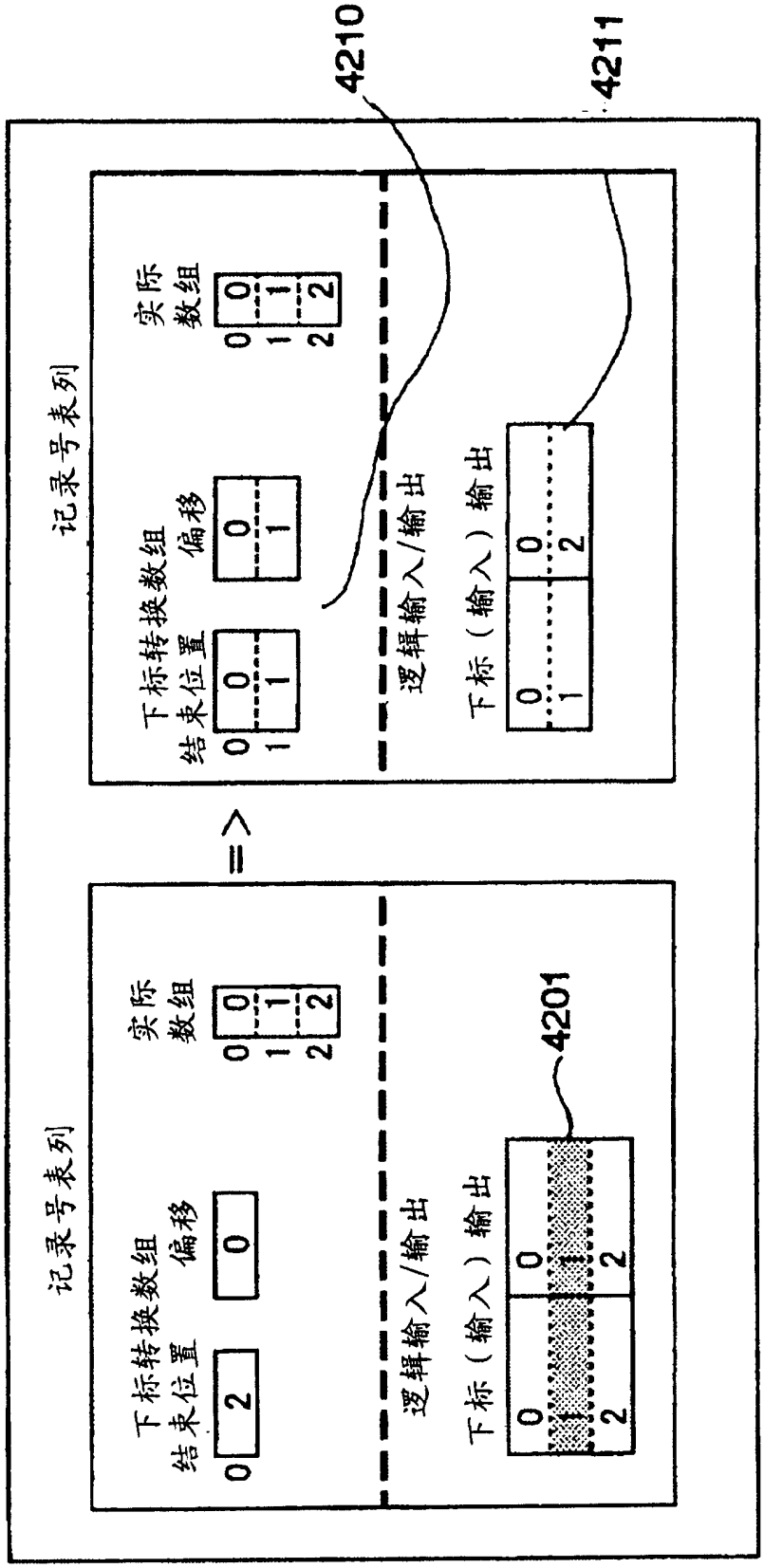


图43

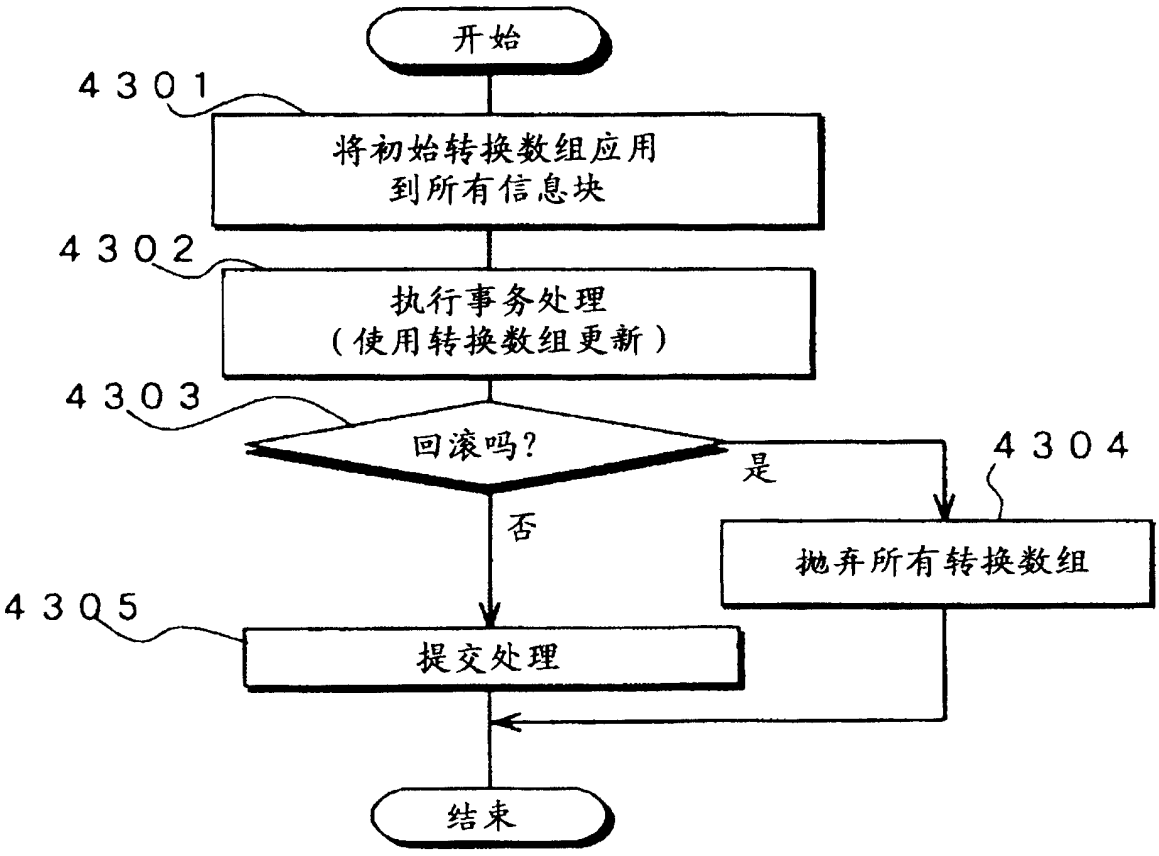

开始
4301
将初始转换数组应用
到所有信息块
4302
执行事务处理
（使用转换数组更新）
4303
回滚吗?
是
4304
抛弃所有转换数组
否
4305
提交处理
结束

图44

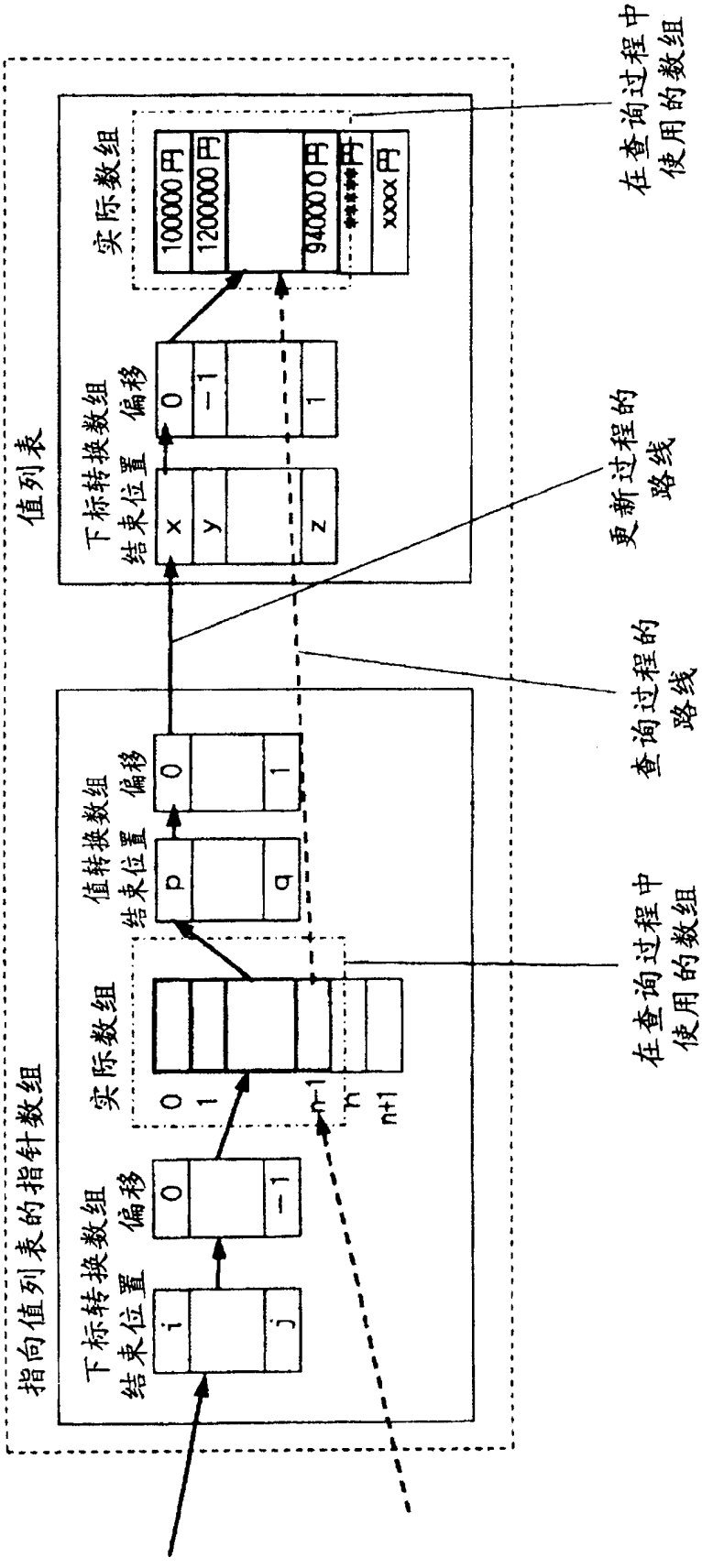
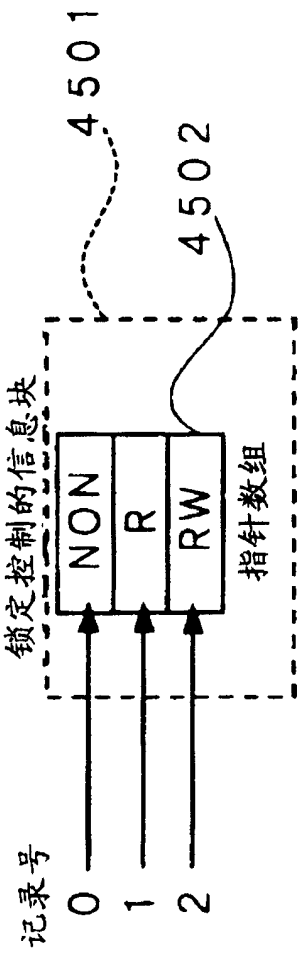

客户余额的信息块
指向值列表的指针数组
下标转换数组
结束位置
偏移
i
j
0
-1
实际数组
0
1
n-1
n
n+1
值转换数组
结束位置
偏移
p
q
0
1
值列表
下标转换数组
结束位置
偏移
x
y
z
0
-1
1
实际数组
100000円
1200000円
940000円
xxxx円
查询过程的路线
更新过程的路线
在查询过程中使用的数组
在查询过程中使用的数组

图 45



个人姓名表视图

Smith	John
Ford	Jonny
Smith	Ken

图 46A

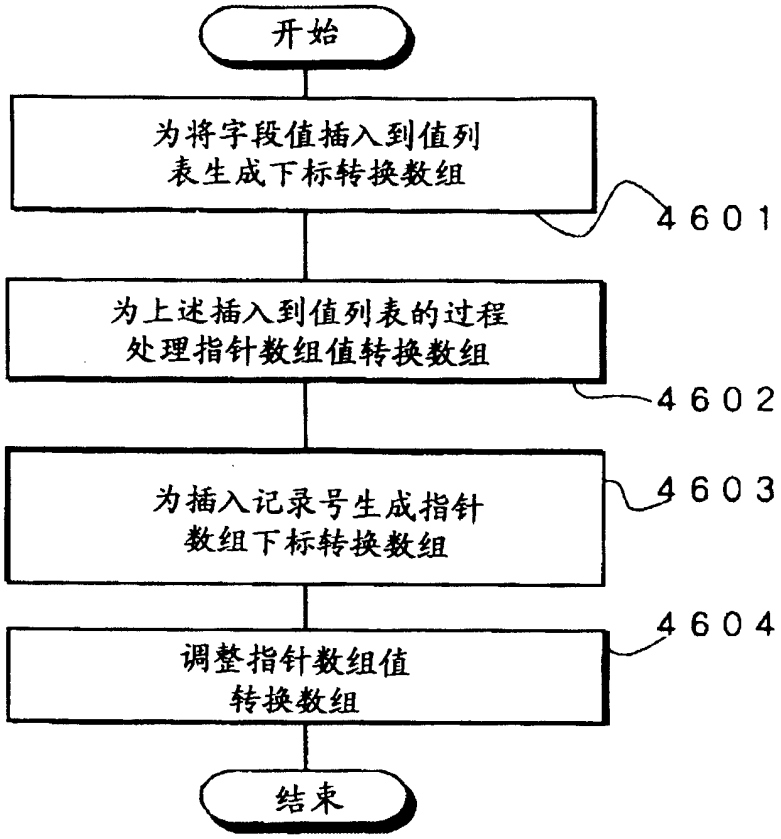

开始
为将字段值插入到值列表生成下标转换数组
4601
为上述插入到值列表的过程处理指针数组值转换数组
4602
为插入记录号生成指针数组下标转换数组
4603
调整指针数组值转换数组
4604
结束

图 46B

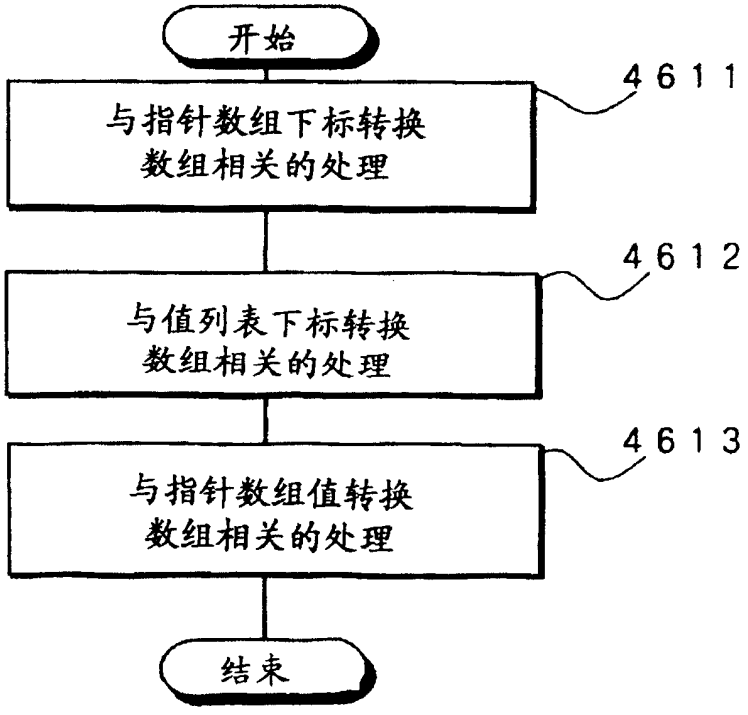

开始
与指针数组下标转换数组相关的处理
4611
与值列表下标转换数组相关的处理
4612
与指针数组值转换数组相关的处理
4613
结束

图47

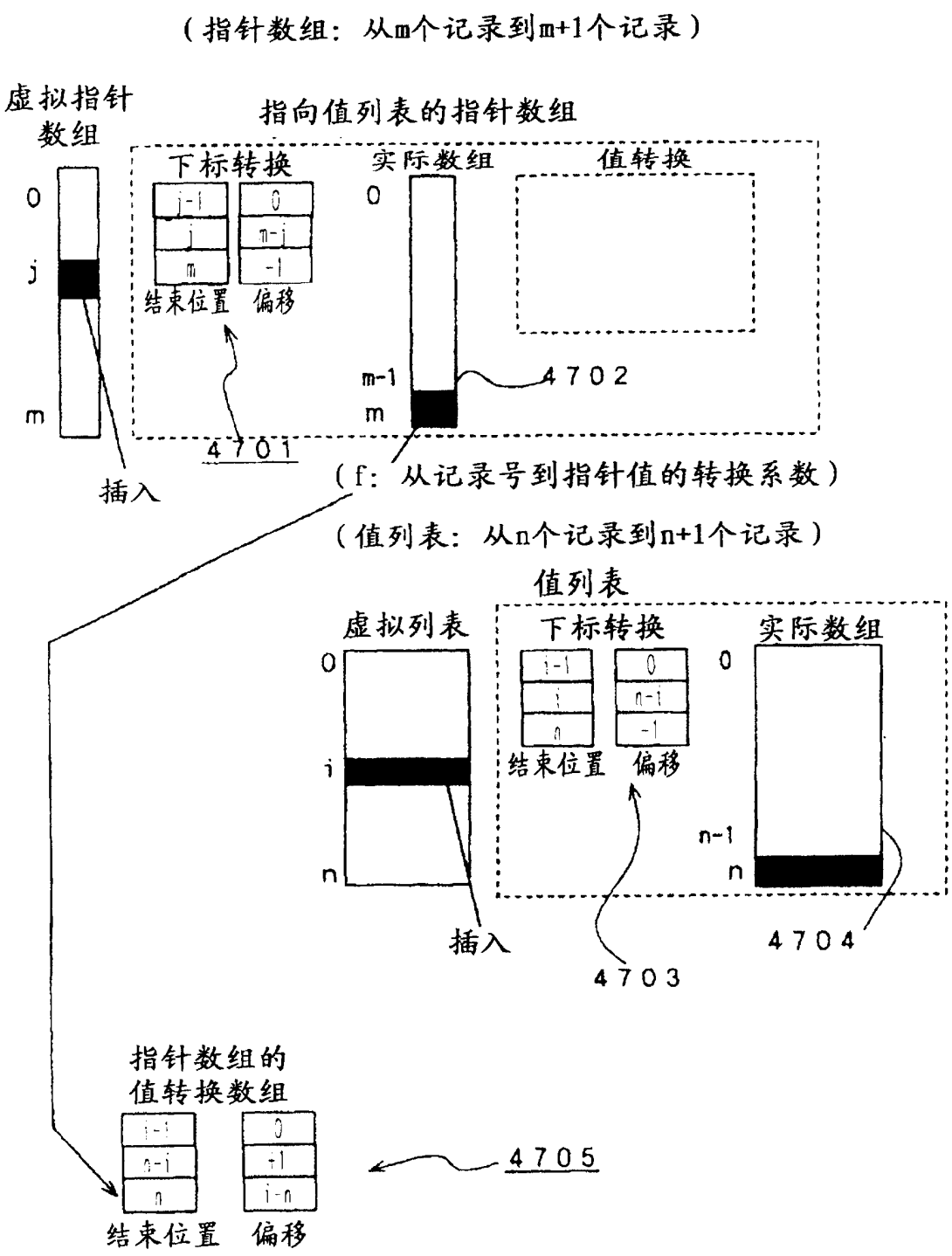

（指针数组：从m个记录到m+1个记录）
虚拟指针数组
指向值列表的指针数组
下标转换
实际数组
值转换
0
j
m
j-1
j
m
0
m-j
-1
结束位置
偏移
0
m-1
m
4702
4701
插入
（f：从记录号到指针值的转换系数）
（值列表：从n个记录到n+1个记录）
值列表
虚拟列表
下标转换
实际数组
0
i
n
i-1
i
n
0
n-1
-1
结束位置
偏移
0
n-1
n
插入
4703
4704
指针数组的值转换数组
i-1
n-i
n
0
+1
i-n
结束位置
偏移
4705

图48

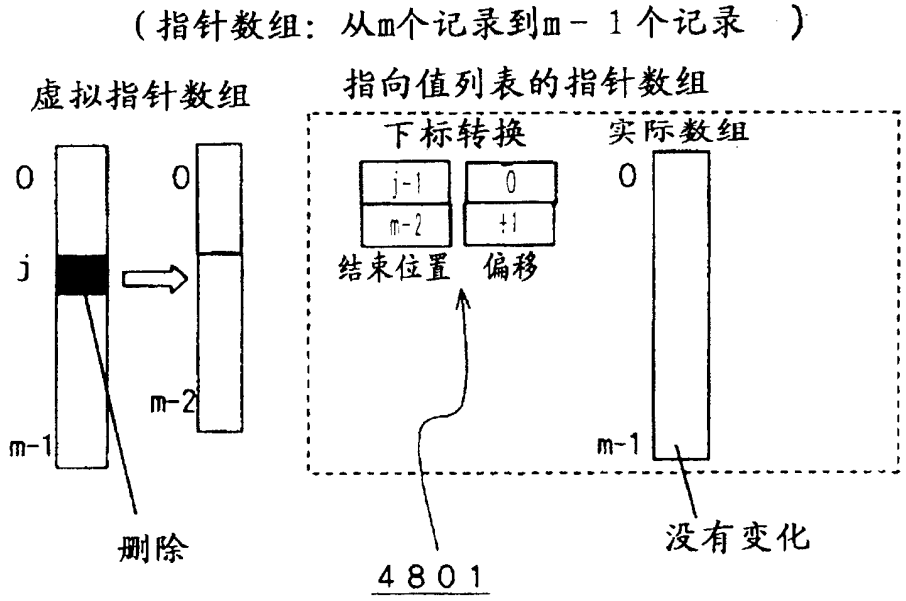

（指针数组：从m个记录到m－1个记录）
虚拟指针数组
指向值列表的指针数组
下标转换
实际数组
0
j
m-1
0
m-2
j-1
m-2
0
+1
结束位置
偏移
0
m-1
删除
没有变化
4801

图49

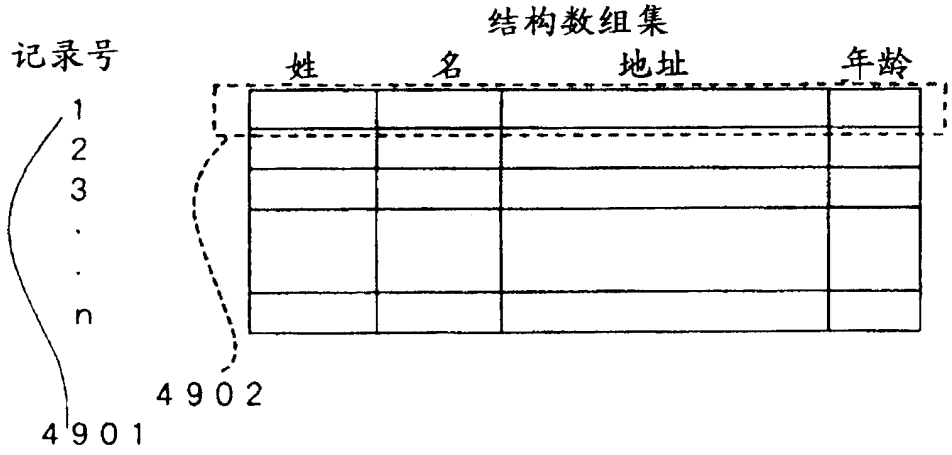

结构数组集
记录号
姓
名
地址
年龄
1
2
3
.
.
n
4902
4901

图 50

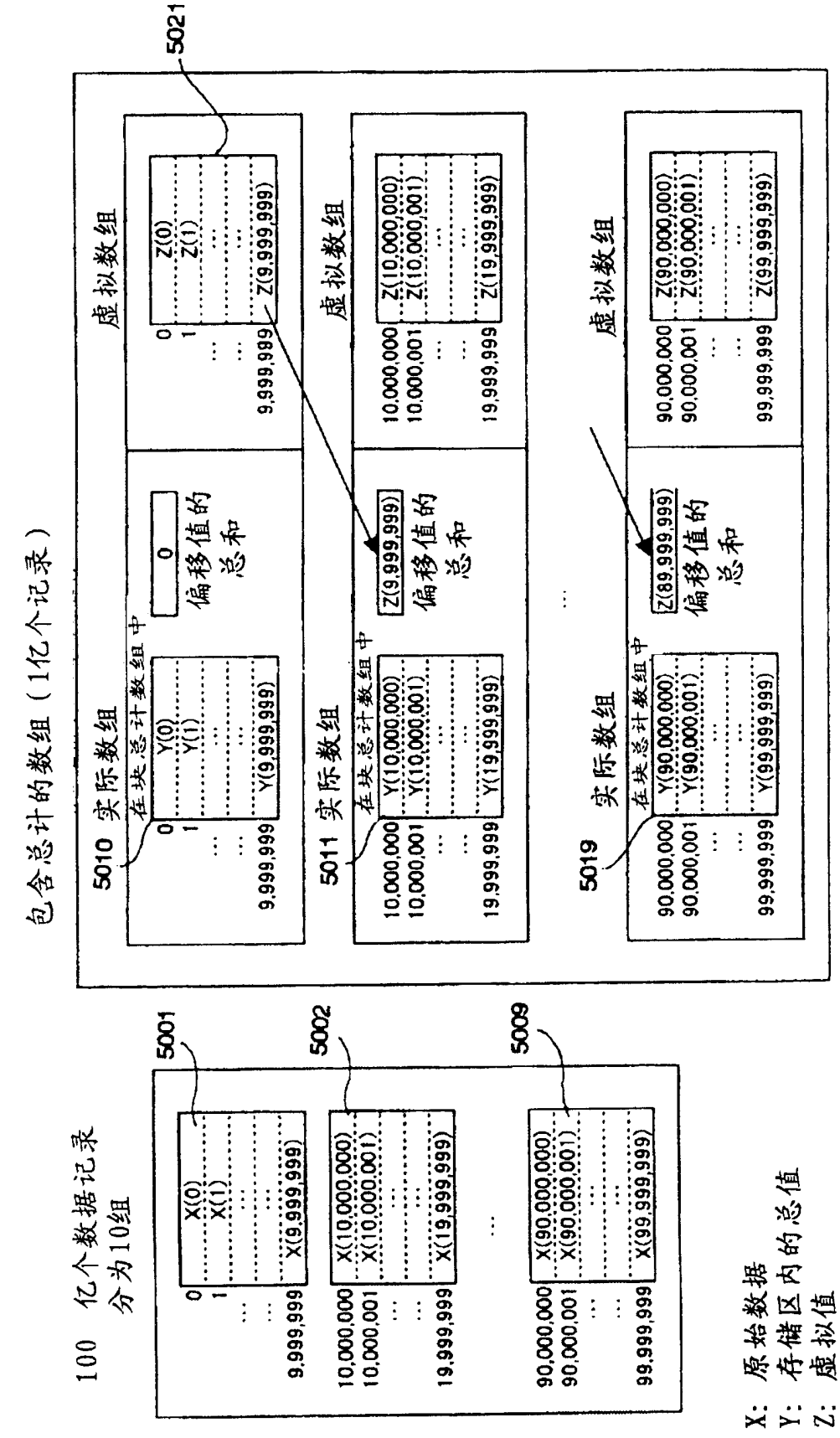

包含总计的数组（1亿个记录）
5021
5010 实际数组
在块总计数组中
虚拟数组
偏移值的总和
5011 实际数组
在块总计数组中
虚拟数组
偏移值的总和
5019 实际数组
在块总计数组中
虚拟数组
偏移值的总和
100 亿个数据记录
分为10组
5001
5002
5009
X: 原始数据
Y: 存储区内的总值
Z: 虚拟值

图51

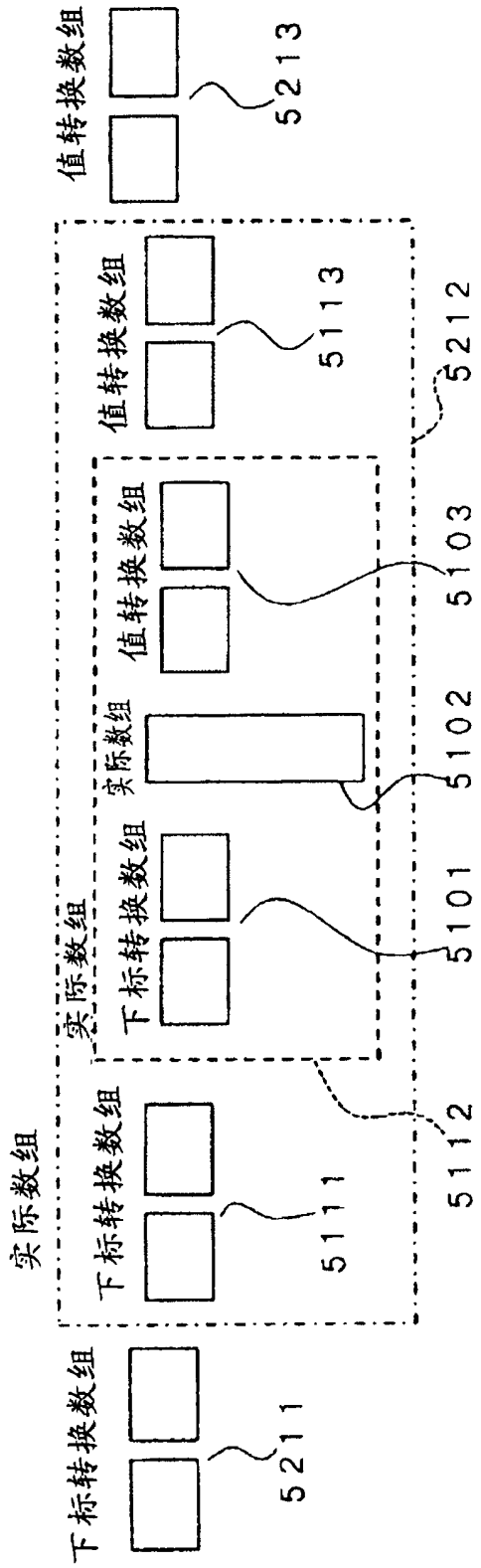

实际数组
下标转换数组
5111
5112
实际数组
下标转换数组
5101
实际数组
5102
值转换数组
5103
值转换数组
5113
5212
值转换数组
5213
下标转换数组
5211

图52

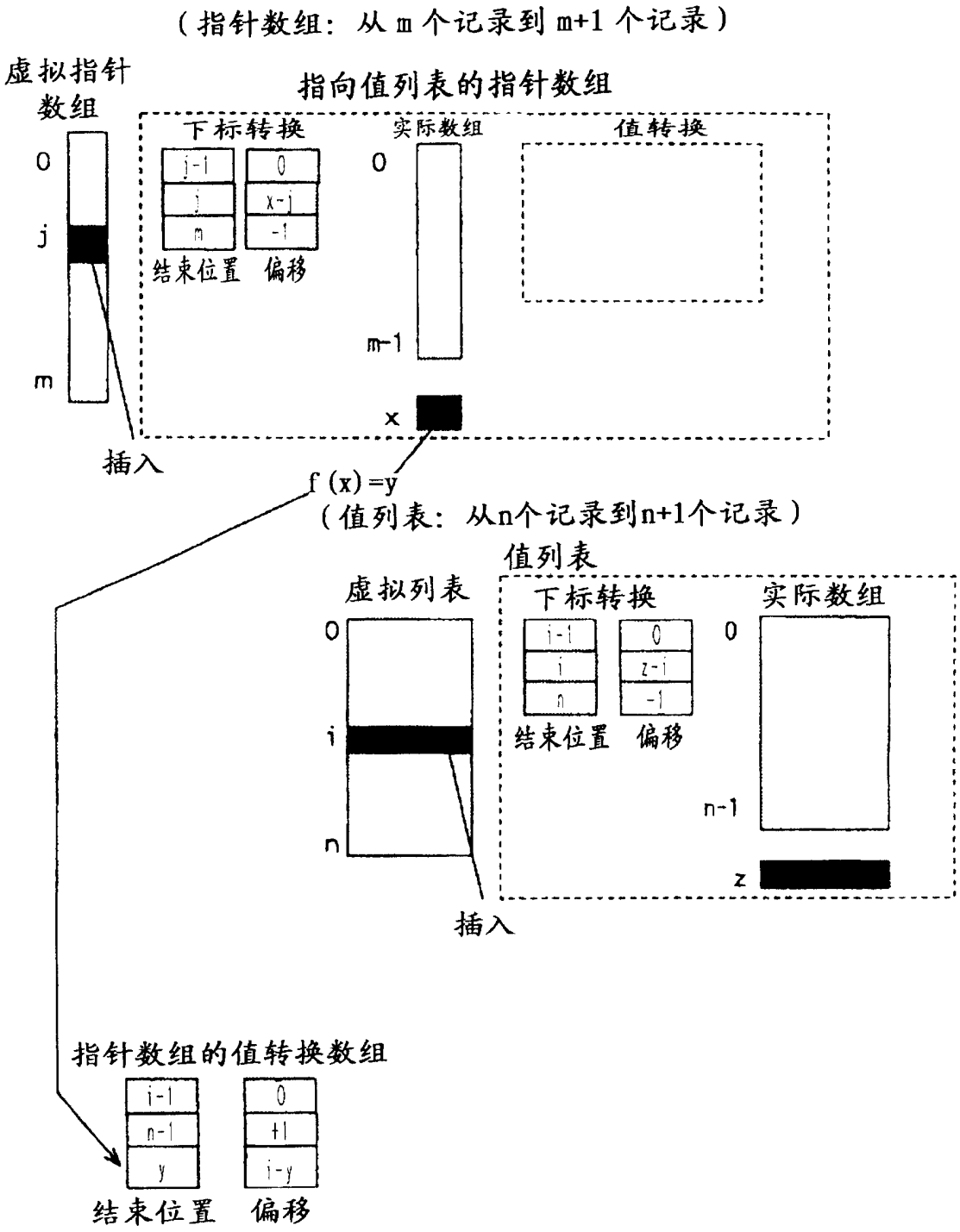

（指针数组：从 m 个记录到 m+1 个记录）
虚拟指针数组
指向值列表的指针数组
下标转换
实际数组
值转换
0
j
m
j-1
j
m
0
x-j
-1
结束位置
偏移
0
m-1
x
插入
f(x)=y
（值列表：从n个记录到n+1个记录）
值列表
虚拟列表
下标转换
实际数组
0
i
n
i-1
i
n
0
z-i
-1
结束位置
偏移
0
n-1
z
插入
指针数组的值转换数组
i-1
n-1
y
0
+1
i-y
结束位置
偏移

图53

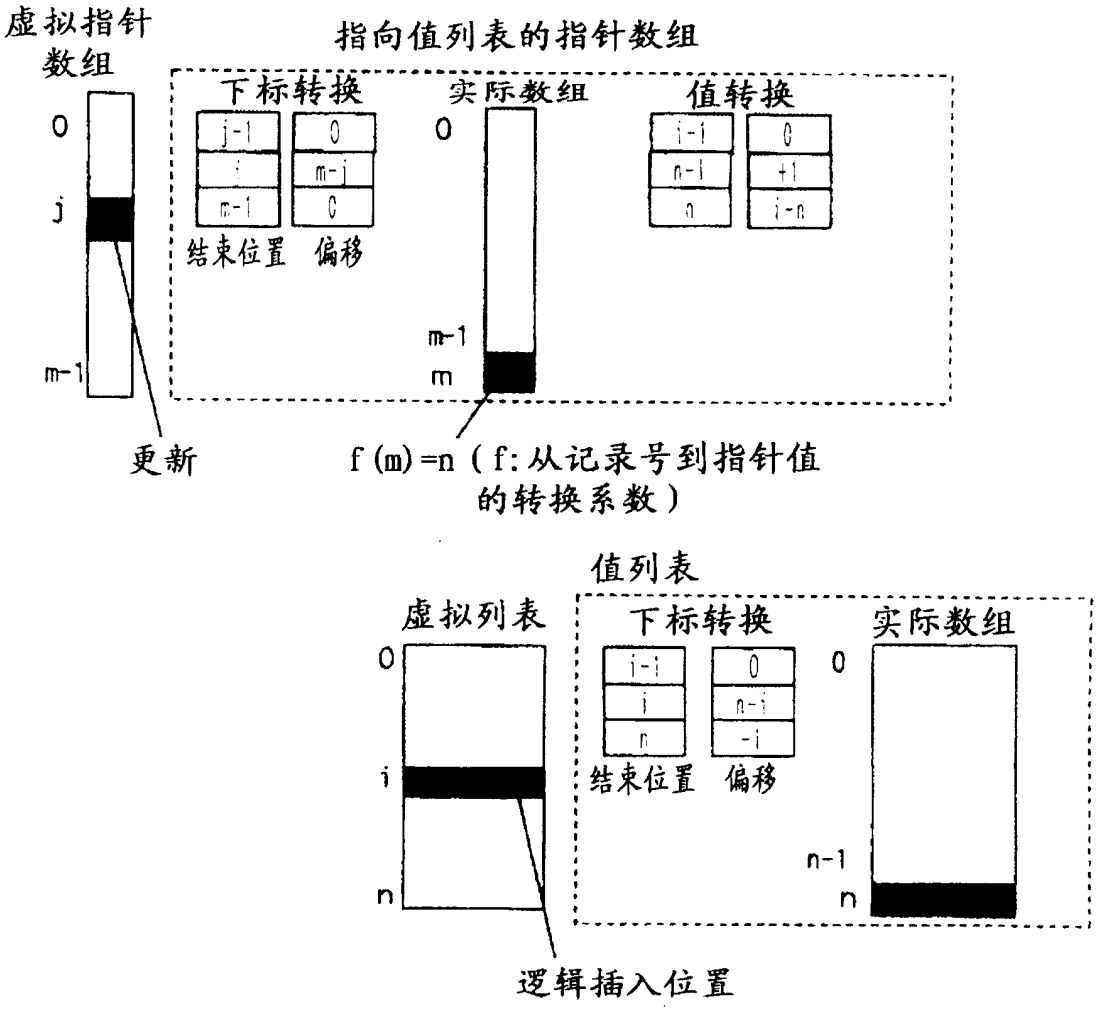
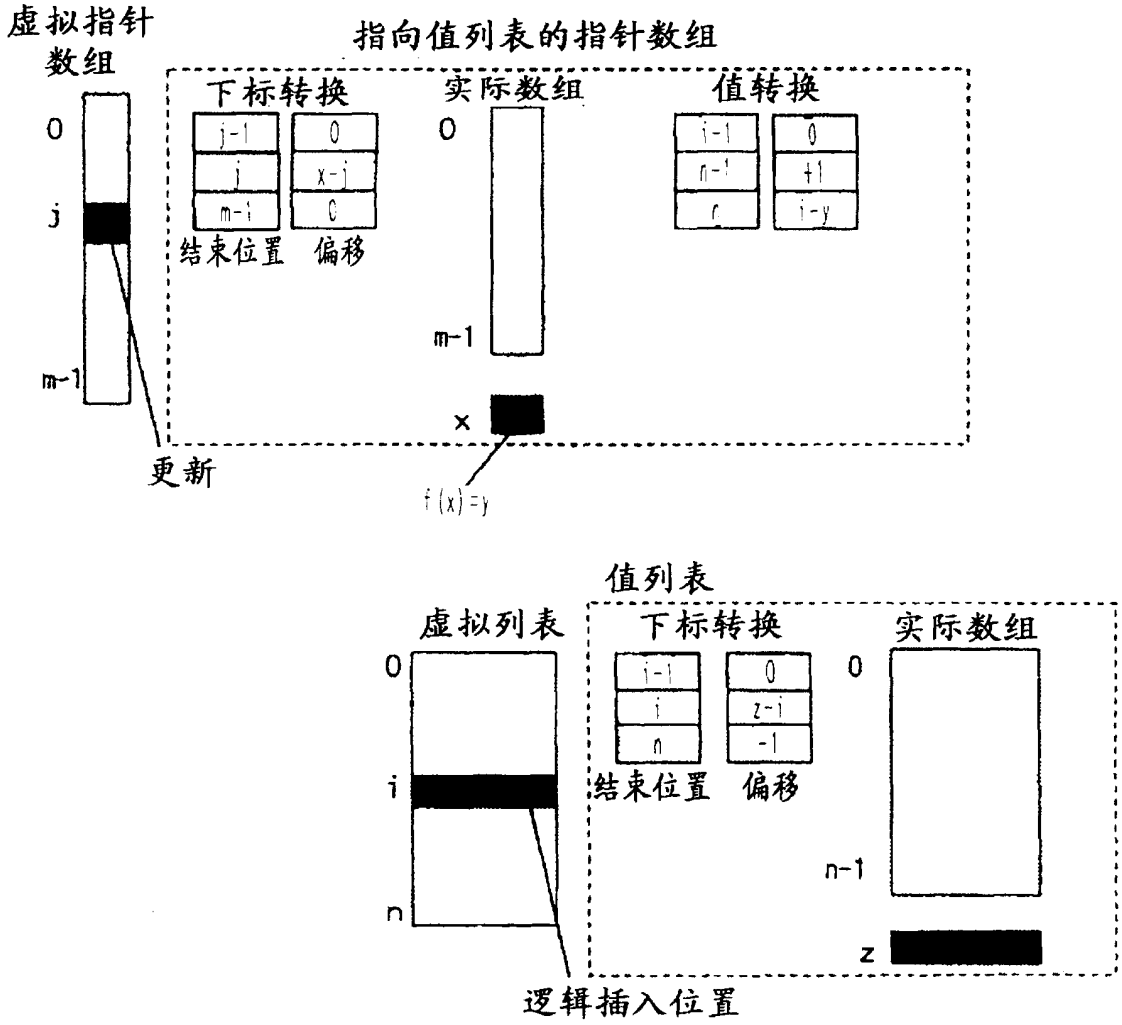

虚拟指针数组
指向值列表的指针数组
下标转换
实际数组
值转换
0
j
m-1
j-1
i
m-1
0
m-j
0
结束位置
偏移
0
m-1
m
i-1
n-1
n
0
+1
i-n
更新
f(m)=n（f:从记录号到指针值的转换系数）
值列表
虚拟列表
下标转换
实际数组
0
i
n
i-1
i
n
0
n-i
-i
结束位置
偏移
0
n-1
n
逻辑插入位置

图54


虚拟指针数组
指向值列表的指针数组
下标转换
实际数组
值转换
0
j
m-1
j-1
j
m-i
0
x-j
C
结束位置
偏移
0
m-1
x
i-1
n-1
r
0
+1
i-y
更新
f(x)=y
值列表
虚拟列表
下标转换
实际数组
0
i
n
i-1
j
n
0
z-i
-1
结束位置
偏移
0
n-1
z
逻辑插入位置