

(19) World Intellectual Property
Organization
International Bureau



(43) International Publication Date
3 March 2005 (03.03.2005)

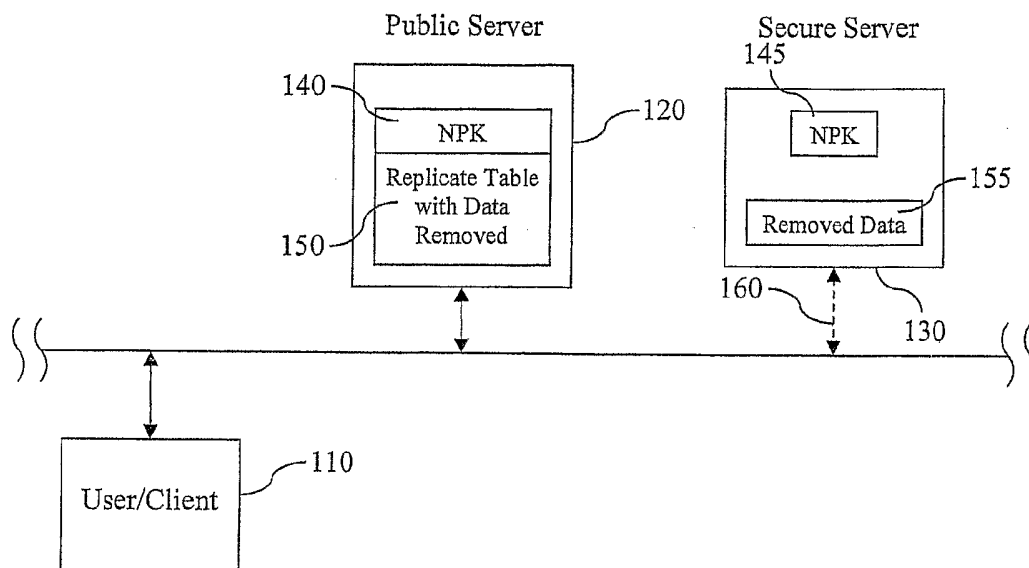
PCT

(10) International Publication Number
WO 2005/020008 A2

- (51) International Patent Classification⁷: **G06F**
- (21) International Application Number:
PCT/US2004/026687
- (22) International Filing Date: 17 August 2004 (17.08.2004)
- (25) Filing Language: English
- (26) Publication Language: English
- (30) Priority Data:
60/496,090 18 August 2003 (18.08.2003) US
- (71) Applicant (for all designated States except US): **EAGLE-HAWK LIMITED**; Ram Re House, 2nd Floor, 46 Reid Street, Hamilton, Bermuda HM12 (BM).
- (72) Inventors; and
- (75) Inventors/Applicants (for US only): **PERCIVAL, Marvin** [NZ/US]; 15 Charlotte Drive, Andover, MA 01810 (US). **TICE, William, C.** [US/US]; 47 Prospect Street, Georgetown, MA 01833 (US).
- (74) Agents: **THIBODEAU, David, J., Jr.** et al.; Hamilton, Brook, Smith & Reynolds, P.C., 530 Virginia Road, P.O. Box 9133, Concord, MA 01742-9133 (US).
- (81) Designated States (unless otherwise indicated, for every kind of national protection available): AE, AG, AL, AM, AT, AU, AZ, BA, BB, BG, BR, BW, BY, BZ, CA, CH, CN, CO, CR, CU, CZ, DE, DK, DM, DZ, EC, EE, EG, ES, FI, GB, GD, GE, GH, GM, HR, HU, ID, IL, IN, IS, JP, KE, KG, KP, KR, KZ, LC, LK, LR, LS, LT, LU, LV, MA, MD, MG, MK, MN, MW, MX, MZ, NA, NI, NO, NZ, OM, PG, PH, PL, PT, RO, RU, SC, SD, SE, SG, SK, SL, SY, TJ, TM, TN, TR, TT, TZ, UA, UG, US, UZ, VC, VN, YU, ZA, ZM, ZW.
- (84) Designated States (unless otherwise indicated, for every kind of regional protection available): ARIPO (BW, GH, GM, KE, LS, MW, MZ, NA, SD, SL, SZ, TZ, UG, ZM, ZW), Eurasian (AM, AZ, BY, KG, KZ, MD, RU, TJ, TM), European (AT, BE, BG, CH, CY, CZ, DE, DK, EE, ES, FI, FR, GB, GR, HU, IE, IT, LU, MC, NL, PL, PT, RO, SE, SI, SK, TR), OAPI (BF, BJ, CF, CG, CI, CM, GA, GN, GQ, GW, ML, MR, NE, SN, TD, TG).
- Published:**
— without international search report and to be republished upon receipt of that report

[Continued on next page]

(54) Title: DATA SECURITY THROUGH DISSEMBLY OF DATA ELEMENTS OR CONNECTIONS BETWEEN ELEMENTS



(57) Abstract: A technique for providing data security through the removal of elements of a database table, be they the data fields themselves, or structural components, such as connection information that defines how, for example, the elements of a database are structured. The removed elements are placed in a secure location, retrievable only by having knowledge of a new primary key stored with the removed elements.



For two-letter codes and other abbreviations, refer to the "Guidance Notes on Codes and Abbreviations" appearing at the beginning of each regular issue of the PCT Gazette.

DATA SECURITY THROUGH DISSEMBLY OF DATA ELEMENTS OR CONNECTIONS BETWEEN ELEMENTS

RELATED APPLICATIONS

This application claims the benefit of U.S. Provisional Application No. 60/496,090, filed August 18, 2003. The entire teachings of the above application are incorporated herein by reference.

BACKGROUND OF THE INVENTION

The problem of protecting an organization's sensitive data from access by unauthorized users is an increasingly important issue for essentially all businesses, government and agency organizations, as well as consumers. In response to the problem of unauthorized access, organizations often adopt a multi-layered approach to security, using some combination of perimeter systems (e.g., firewalls or authentication) and encryption technologies designed to mathematically encode data within a database.

Unfortunately, years of experience have demonstrated that even the most sophisticated perimeter and encryption technologies can be penetrated by determined attackers, with incidents of such vulnerabilities being both continuous and well-documented. Once such penetration occurs, unauthorized access has been gained to the actual data being protected, and the confidentiality of this data is at risk.

SUMMARY OF THE INVENTION

The present invention involves the concept of providing data security through the removal of elements of a data store, whether they be the data fields themselves or structural components, such as connection information that defines how the tables of a database are related. The removed elements are placed in a secure remote location,

- 2 -

retrievable only by those having authenticated permission to access the removed information.

In the case of data element removal, individual data elements are removed from the database tables and secured via access control mechanisms. As determined by the access control mechanisms, the removed elements are then accessible only by authorized users.

The database structure is disassembled into discrete tables by removing the linkages that bind the database tables together and securing the linkages via access control mechanisms. The individual tables of the database structure are then stored as separate, unrelated tables on either the same, or different, computers. Then, as determined by the access control mechanisms, the linkages are retrieved and employed to reassemble data from the individual database tables for authorized users.

The method may involve assigning new primary keys and/or linkage fields to the table to allow for secure retrieval of the removed data elements and their association with the appropriate table data elements.

The invention has advantages over techniques that remove data elements but then rely simply on encryption and/or access control for security. Specifically, the present invention depends on deconstruction, removal of elements (data elements or structural elements), and inserting a new primary key. The present invention can still utilize encryption in the data tables that have elements removed or in the secure location where removed elements are stored, but the invention is not dependent on the use of such encryption to achieve data security.

BRIEF DESCRIPTION OF THE DRAWINGS

The foregoing and other objects, features and advantages of the invention will be apparent from the following more particular description of preferred embodiments of the invention, as illustrated in the accompanying drawings in which

- 3 -

like reference characters refer to the same parts throughout the different views. The drawings are not necessarily to scale, emphasis instead being placed upon illustrating the principles of the invention.

Fig. 1 is a block diagram of a network in which the present invention may be employed;

Fig. 2 is a flow diagram of a process to remove and secure an element from the replicate table;

Fig. 3 is a flow diagram of a process to reconstruct the replicate table;

Fig. 4 is a diagram of the original or replicate table;

Fig. 5 is a diagram of the replicate table with the new primary key field;

Fig. 6 is a diagram of the replicate table with the new primary key field and a data field removed;

Fig. 7 is a diagram of the secure location where the replicate new primary key and the removed element are stored;

Fig. 8 is a flow diagram of a process to secure the link between replicate tables;

Fig. 9 is a diagram of original or replicate parent and child tables linked together;

Fig. 10 is a diagram of the replicate parent and child tables with new primary keys and new link fields;

Fig. 11 is a diagram of the parent table with the new link field removed and the child table with the link field removed; and

Fig. 12 is a diagram of the secure location where the removed link fields are stored.

DETAILED DESCRIPTION OF THE INVENTION

In general, determined attackers can and will penetrate perimeter security (e.g., firewalls) and encryption technologies. Thus, sensitive data that resides in a database protected only by these techniques is potentially at risk to unauthorized access.

- 4 -

Fig. 1 shows a network in which the present invention may be employed. A user/client 110 can access a public server 120 containing database tables. An authorized user/client 110 can further access a secure server 130 via secure network connections 160.

To understand a first aspect of the present invention, consider the structure of a typical relational database. In such a structure, data elements reside in rows and columns. Each row contains the potential for multiple fields, with each field in the row being a separate column. Within this structure, if data contained in a field, row, column, or table is corrupted or removed, that data is no longer available for access in the table.

To that end, the present invention provides security by preventing unauthorized access to sensitive data through the removal of certain elements from the database. In such an instance, even a successful attacker cannot access data that is no longer there. In addition, the removal process occurs in such a way as to enable the programmatic reconstruction of the database for authorized users.

Programmatic removal and relocation of a data element from a table to a separate, secure location having memory of the original location of the removed element in a database table allows for, first, security of the removed data element and, second, programmatic reconstruction of the removed data element to the database table.

In a first embodiment, the invention can be carried out by a data processing system that is adapted for relational database processing. Such a system may include an appropriately programmed data processor with associated secure storage 130 that is accessible via secure network connections 160.

A method that secures data through element removal proceeds as a sequence of processing steps shown in the flowchart in Fig. 2. The method begins in step 205. In step 210, the original table is replicated to produce a replicate table 400 shown in Fig. 4. The replicate table 400 (and the original table) consists of a Primary Key 405 field and any number of fields for data such as Field Two 410. All of the data in the replicate table (and original table) is available for retrieval. However, if the data in

- 5 -

Field Two 410 is corrupted, such as by its complete removal, it is no longer available for retrieval and the table is of limited value to someone who accesses it.

To reach that result, in step 220, a new primary key (NPK) field is added to the replicate table, with the NPK value having no relationship to the original table data (such an NPK value can be generated randomly or in other ways). The resulting table is shown in Fig. 5 where the NPK field 510 has a value, for example, of X4S7.

In step 230, a secure location is created for the storage of objects removed from the replicate table. The secure location should be separate from the originating database containing the table data to be removed, and is ideally an object oriented data environment with very strong access control attributes. An example of an object oriented database suitable for this repository would be the StrongBox™ product available from Eaglehawk Limited or the ObjectStore® product available from Progress Software. Alternatively, the secure location could be a relational or other database type, utilizing very strong access control and possibly encryption of data.

Next, in step 240, the New Primary Key is replicated and stored into the secure location as an individual object. Fig. 7 shows a secure location 700 where the NPK value 710 is stored.

Finally, in step 250, (user specified) data from the replicate table is replicated, removed, and stored as objects in the secure location. For example, Fig. 6 shows a replicate table 600 with the data from Field Two 610 removed leaving a NULL or specious value. As illustrated in Fig. 7, the removed data (BBBB) is stored as an object in the secure location 700. In the end, the contents of the secure location 700 will appear as a disjointed set of values to an unauthorized user. The removed data of the replicate table is thus replicated in the secure location, ensuring that no obvious relationship exists with the NPK value to a user who does not have appropriate access rights. The original table can then be deleted or removed from user access.

- 6 -

When a user or client needs access to the replicate table data including the removed data elements the flowchart steps shown in Fig. 3 are followed. The method begins in step 305. In step 310, an authorized user or client makes a reconstruction request for removed and/or table-resident data elements. Any table-resident elements are processed based on selection criteria of the request. In step 320, NPK values of rows meeting the selection criteria are retrieved. In step 330, the secure location is accessed using retrieved NPK values and removed data elements are located and processed according to selection criteria applying to the NPK of these data objects. Finally, in step 340, using matching NPK values, the previously removed and table-resident data elements meeting selection criteria of the request are combined.

A second important aspect of the present invention relates to the fact that in a traditional relational database structure separate data tables can be, and typically are, connected using parent-child relationships. Each such table must have a primary key (PK) field. A row in each child table must have a field with a value matching the value of its parent row in the parent table PK field. For example, in Fig. 9, a child table 920 has a value in Field One 925 matching the value in the PK field 915 of a parent table 910.

However, if the parent table PK field in such a structure is corrupted or removed, the child table is no longer able to relate to the parent table. As such, a secure method for storing the table can be provided through programmatic removal and relocation of a parent table PK to a secure location. As described above, this allows for, first, security through the inability of the child table to relate to the parent table and, second, programmatic reestablishment of the parent/child relationship.

In one embodiment, the invention can secure the contents of a structural table by the processing steps shown in Fig. 8. The process begins in step 805. In step 810, the original parent and child data tables are replicated producing replicate parent and child tables. In step 820, two fields are added to each replicate table: a New Primary Key (NPK) field and a New Linking (NL) field. Fig. 10 illustrates the addition of these new fields in the replicate parent table 910 and the replicate child

- 7 -

table 920. In step 830, the NPK field in the replicate parent and child tables are populated with different values having no relationship to the original data. For example, in Fig. 10, the NPK field 1030 of the child table 920 has a different value than the value of the NPK field 1010 of the parent table 910.

In step 840, the NL field in the parent and child replicate tables are populated with the same values, but again, having no relationship to the original data. For example, in Fig. 10, the NL field 1040 of the child table 920 has the same value as the NL field 1020 of its parent row in the parent table 910.

In step 850, the NPK field value from the Parent Table is replicated and stored in a secure location.

In the remaining steps the data structure is secured by disconnecting the replicate parent and child tables. First, in step 860, the replicate parent table NL field is removed, secured in a secure location, and set to NULL. Then, in step 870, the replicate child table field matching the replicate parent table PK is set to NULL.

At the end of this process, the replicate parent and child tables appear as they do in Fig. 11 with the NL field 1020 in the Parent Table 910 set to NULL and Field One 925 in the Child Table 920 set to NULL. Fig. 12 illustrates the secure location 1200 at this point containing the replicate NPK value of the Parent Table 1220 and the NL field value of the replicate parent and child tables 1210.

The data structure is then reestablished by reconstructing the parent/child relationship. A method of the secure location when supplied with the replicate parent table NPK will return the replicate child table NL, thus providing the ability to reestablish the relationship between tables. Similarly, a method of the secure location when supplied with the child table NL will return the replicate parent table NPK, thus providing the ability to reestablish the relationship between tables.

While this invention has been particularly shown and described with references to preferred embodiments thereof, it will be understood by those skilled in the art that various changes in form and details may be made therein without departing from the scope of the invention encompassed by the appended claims.

- 8 -

CLAIMS

What is claimed is:

1. A method for providing data security comprising the steps of:
identifying an original table having a primary key field and multiple data fields;
replicating the original table in a replicate table;
adding a new primary key to the replicate table;
removing an element from the replicate table; and
storing the element removed in a secure location, with no linkage between the new primary key and the element removed.
2. A method as in claim 1 wherein the original table resides in a database structure.
3. A method as in claim 1 wherein the removed elements are data fields.
4. A method as in claim 3 wherein the removed elements comprise link information.
5. A method as in claim 1 wherein the secure location is an object oriented database with strong access control.
6. A method as in claim 1 wherein the data elements in the secure location appear as a disjointed set of values.
7. A method as in claim 1 wherein the replicate table is stored in the secure location.

- 9 -

8. A method as in claim 1 wherein the replicate table is stored in a remote location.
9. A method as in claim 1 wherein the replicate table with elements removed is generally accessible.
10. A method as in claim 1 and further providing for reconstruction of the replicate table comprising the steps of:
 - retrieving the new primary key associated with a requested removed element;
 - presenting the new primary key to the secure location to retrieve the requested removed element; and...
 - combining the requested removed element with the replicate table.
11. A method as in claim 1 wherein ~~the~~ the new primary key has no relationship to the original table data.
12. A method as in claim 1 wherein the original table has parent and child portions that replicate into respective parent replicate and a child replicate portions.
13. A method as in claim 12 wherein a new primary key field is added to the parent replicate and the child replicate.
14. A method as in claim 12 wherein a new link field is added to the parent replicate and the child replicate.
15. A method as in claim 12 wherein the data value in a link field in the child portion is removed and stored in the secure location.

- 10 -

16. A method as in claim 14 wherein the data value in a new link field in the parent portion is removed and stored in the secure location.
17. A method as in claim 12 wherein a new primary key is added to the parent replicate and a different new primary key is added to the child replicate.
18. An apparatus for providing data security comprising:
 - an original data store, for storing an original table;
 - a replicate data store, for storing a new primary key and a replicate of the original table in which a selected removed element of the original table is not stored;
 - a secure location, for storing the removed element and a replicate of the new primary key, with no apparent linkage between the removed element and the new primary key.
19. An apparatus as in claim 18 wherein the original table resides in a database structure.
20. An apparatus as in claim 18 wherein the removed elements are data fields.
21. An apparatus as in claim 20 wherein the removed elements comprise link information.
22. An apparatus as in claim 18 wherein the secure location is an object oriented database with strong access control.
23. An apparatus as in claim 18 wherein the data elements in the secure location appear as a disjointed set of values.

- 11 -

24. An apparatus as in claim 18 wherein the replicate table is stored in the secure location.
25. An apparatus as in claim 18 wherein the replicate table is stored in a remote location.
26. An apparatus as in claim 18 wherein the replicate table with elements removed is generally accessible.
27. An apparatus as in claim 18 additionally comprising a reconstructed data store for storing a reconstructed data table which originated from portions of the replicate data store and at least one element retrieved from the secure location.
28. An apparatus as in claim 18 wherein the new primary key has no relationship to the original table data.
29. An apparatus as in claim 18 wherein the original table has parent and child portions that replicate into respective parent replicate and child replicate portions.
30. An apparatus as in claim 29 wherein the new primary key field is added to the parent replicate and the child replicate.
31. An apparatus as in claim 29 wherein a new link field is added to the parent replicate and the child replicate.
32. An apparatus as in claim 29 wherein the data value in a link field in the child portion is removed and stored in a secure location.

- 12 -

33. An apparatus as in claim 31 wherein the data value in a new link field in the parent portion is removed and stored in a secure location.
34. An apparatus as in claim 29 wherein a new primary key is added to the parent replicate and a different new primary key is added to the child replicate.

1/12

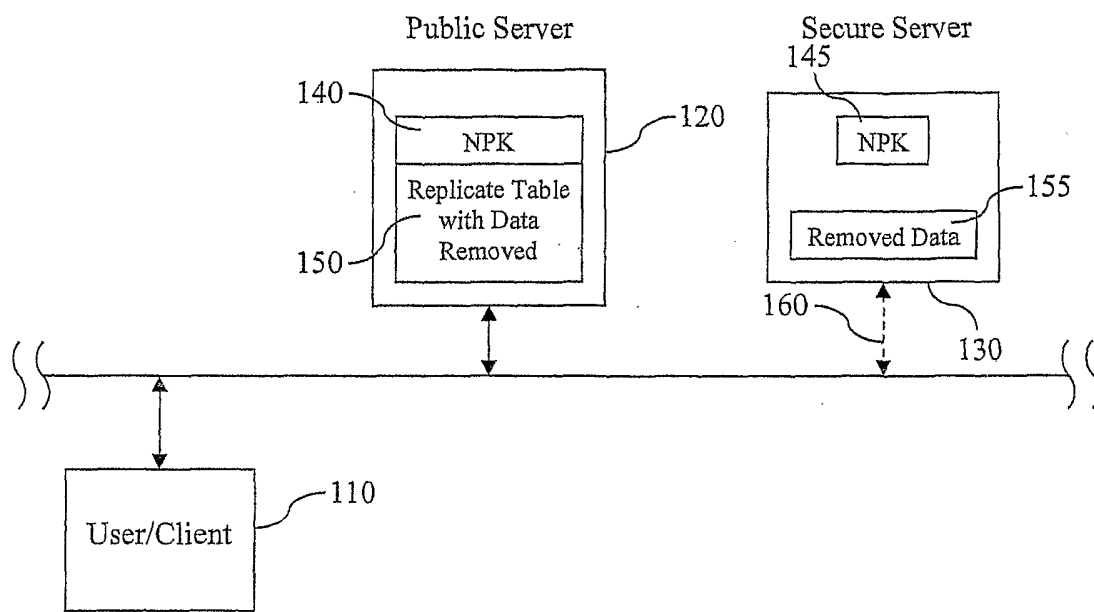


Fig. 1.

2/12

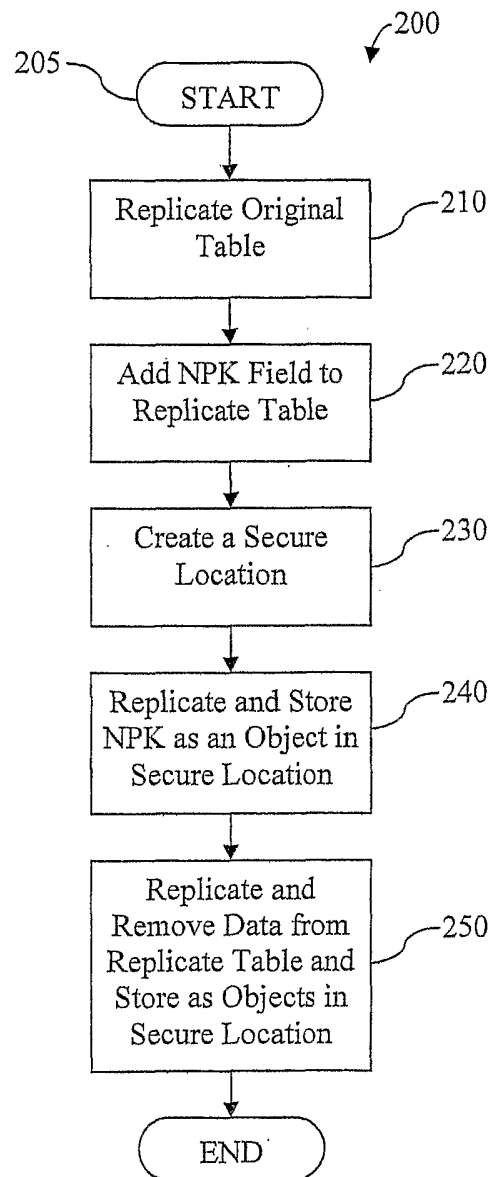


Fig. 2.

3/12

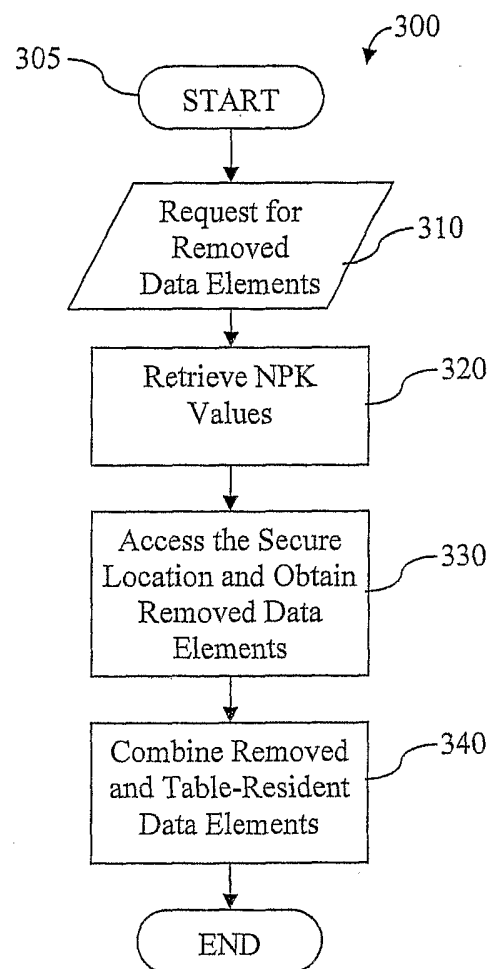


Fig. 3.

The diagram shows a table structure. A bracket on the right side of the table is labeled 400, indicating the entire table. A bracket pointing to the first row is labeled 405. A bracket pointing to the remaining rows is labeled 410.

Table
Primary Key – PK – value = 1234
Field One – value = AAAA
Field Two – value = BBBB
Field Three – value = CCCC
Etc.

Fig. 4.

500

Table
New Primary Key – NPK – value = X4S7
Primary Key – PK – 1234
Field One – value = AAAA
Field Two – value = BBBB
Field Three – value = CCCC
Etc

510

Fig. 5.

600

Table
New Primary Key – NPK – value = X4S7
Primary Key – PK – 1234
Field One – value = AAAA
Field Two – value = <i>NULL</i>
Field Three – value = CCCC
Etc

610

Fig. 6.

7/12

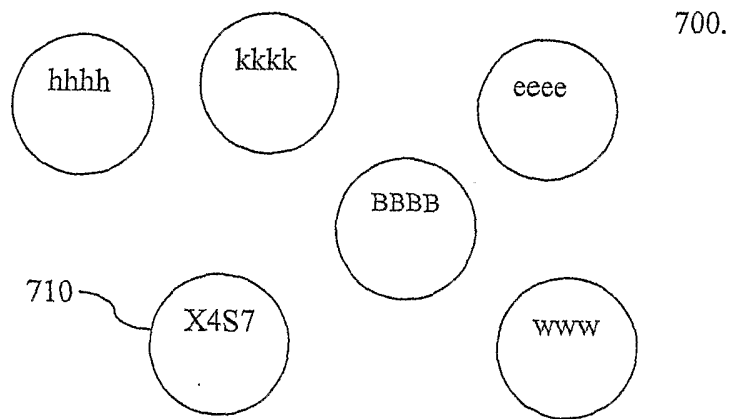


Fig. 7.

8/12

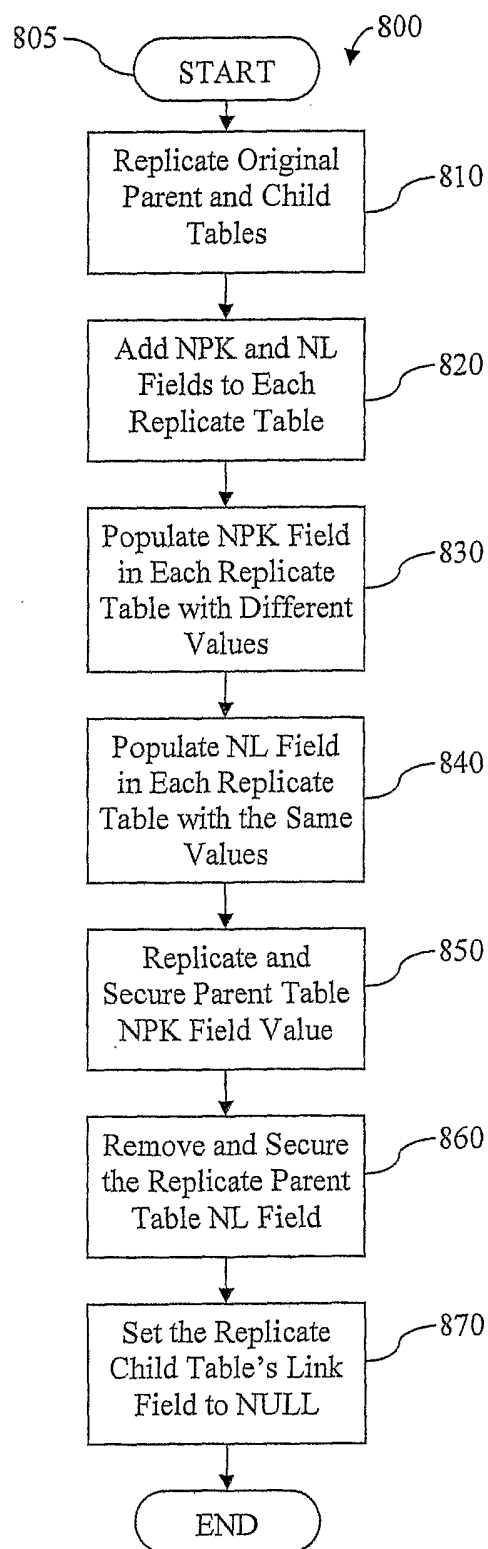


Fig. 8.

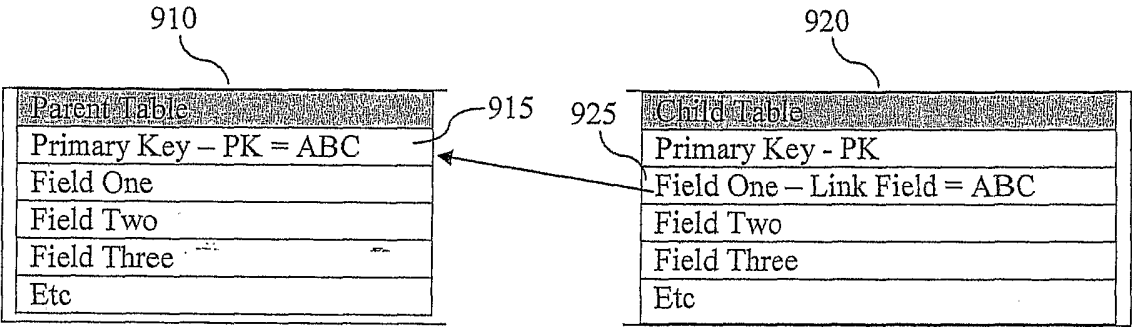


Fig. 9.

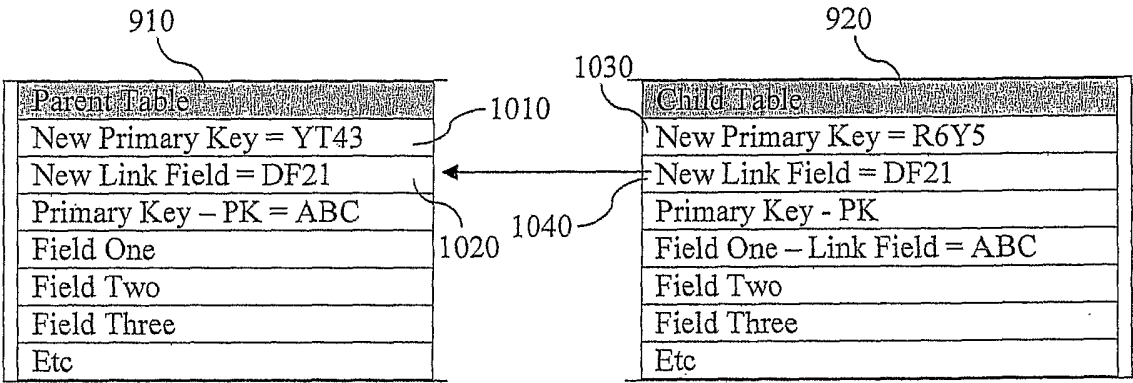


Fig. 10.

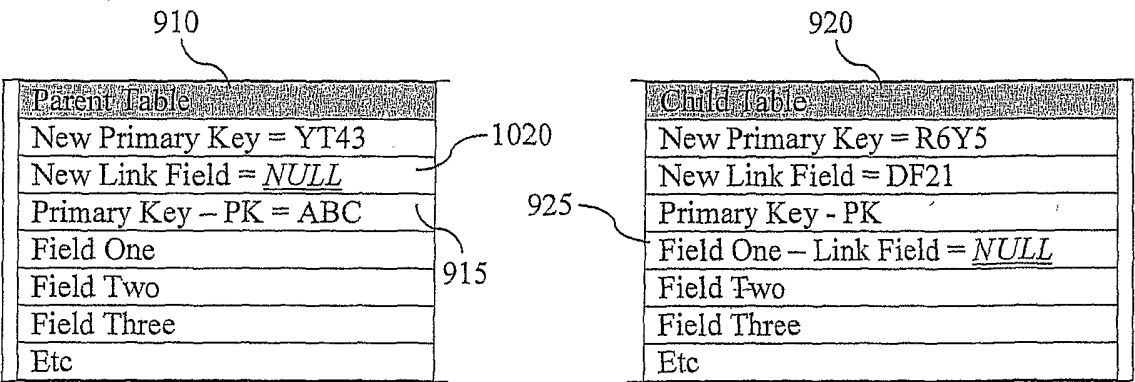


Fig. 11.

12/12

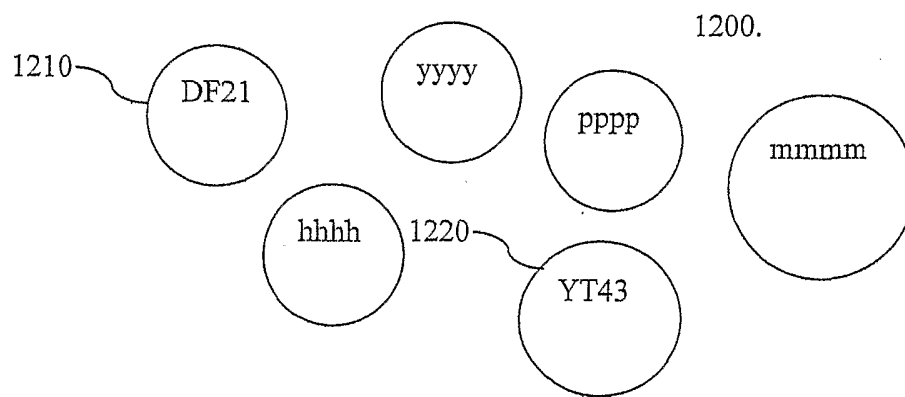


Fig. 12.