(12) INTERNATIONAL APPLICATION PUBLISHED UNDER THE PATENT COOPERATION TREATY (PCT)

(19) World Intellectual Property Organization
International Bureau

(43) International Publication Date
1 December 2005 (01.12.2005)

PCT

(10) International Publication Number
WO 2005/114480 A2

(51) International Patent Classification[7]: G06F 17/30

(21) International Application Number:
PCT/US2005/017602

(22) International Filing Date: 20 May 2005 (20.05.2005)

(25) Filing Language: English

(26) Publication Language: English

(30) Priority Data:
60/573,670 21 May 2004 (21.05.2004) US

(71) Applicant *(for all designated States except US)*: **COMPUTER ASSOCIATES THINK, INC.** [US/US]; One Computer Associates Plaza, Islandia, NY 11749 (US).

(72) Inventor; and
(75) Inventor/Applicant *(for US only)*: **WITT, Russell** [US/US]; 4524 Fremont Lane, Plano, TX 75093 (US).

(74) Agent: **BHAVSAR, Samir, A.**; Baker & Botts, LLP, 2001 Ross Avenue, Dallas, TX 75201 (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, KM, KP, KR, KZ, LC, LK, LR, LS, LT, LU, LV, MA, MD, MG, MK, MN, MW, MX, MZ, NA, NG, NI, NO, NZ, OM, PG, PH, PL, PT, RO, RU, SC, SD, SE, SG, SK, SL, SM, 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, IS, IT, LT, 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*

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.

(54) Title: UNINTERRUPTED DATABASE INDEX REORGANIZATION/MOVEMENT

(57) Abstract: A method for reorganizing a first database includes copying data form the first database to a second database having an index structure different from that of the first database, updating the second database to reflect changes that have been made to the first database since the commencement of the copying, and switching the second database for the first database.

WO 2005/114480 A2

UNINTERRUPTED DATABASE INDEX REORGANIZATION / MOVEMENT

BACKGROUND

Reference to Related Application

The present disclosure is based on and claims the benefit of Provisional application Serial No. 60/573,670 filed May 21, 2004, the entire contents of which are herein incorporated by reference.

TECHNICAL FIELD

The present disclosure relates to databases and, more specifically, to uninterrupted reorganization and movement of databases.

DESCRIPTION OF THE RELATED ART

A database is an organized collection of data. Most databases are computerized and are stored on computer-readable storage devices, such as hard disks. Computerized databases are frequently built, maintained and called upon to store, organize, and retrieve useful information as needed. A database management system (DBMS) is generally a computer program that is designed to store, organize, and retrieve computerized database information.

Many databases organize information as a collection of tables. Each table may include one or more entries and each entry may have one or more attributes. Each attribute may then be assigned one or more values. Each table may include one or more columns; a single column generally stores a single attribute for each entry. Each table may include of one or more rows, a single row generally stores a single entry with all of its attributes.

In certain databases, for example relational databases, two or more tables may be related to each other by virtue of sharing common entries and/or common elements. A DBMS for relational databases may utilize various indexes in responding to queries.

Database entries may be hierarchically arranged such that some entries are parents

of other entries which are children.

Indexing is a technique used by most current DBMSs to speed up particular kinds of queries. For example, redundant information may be internally generated and stored in the database tables to more quickly locate table entries. The most common kind of index is a sorted list of the contents of some particular table column, with pointers to the row associated with the value.

There are multiple different forms of indexed databases. One such form is a fixed-indexed database. The fixed-index database includes an index for assisting in quickly responding to queries that has been specifically organized for a particular database. While the fixed-indexed database may be able to quickly respond to queries, it may be necessary to re-organize the fixed indexes every time the database is modified. For example, it may be necessary to re-organize fixed indexes as new records are added to the database. Re-organizing an index is the process of updating the index to reflect a modified database, for example to reflect the addition of a new database entry. For example, where a fixed-index is the sorted list described above, re-organization may be the re-sorting of the list after a new entry has been added.

Another form of indexed database is the dynamically indexed database. Dynamically indexed databases may utilize indexes that may be modified without having to re-organize the index every time the database is modified. While the dynamically indexed database may not be as fast as the fixed-indexed database, the dynamically indexed database may require less frequent re-organization.

However, with either form of indexed database, periodic re-organization may be required. Re-organization of database indexes often requires that the database be made temporary unavailable. While re-organization is underway, I/O activity, such as for example queries, reads, and updates may be prevented.

There may be other reasons why a database may be made temporarily unavailable. For example, it may be desirable to move a database from one device to another. For example, a database may be moved from one server to another. Moving a database may require that the database be made temporarily unavailable.

Users of modern DBMS are becoming increasingly less tolerant of DBMS downtime. There is therefore a need to utilize a system and method for re-organizing

database tables without impeding database I/O activity.

SUMMARY

A method for reorganizing a first database includes copying data from the first database to a second database having an index structure different from that of the first database, updating the second database to reflect changes that have been made to the first database since the commencement of the copying, and switching the second database for the first database.

A method for reorganizing a first database includes preparing a second database, identifying a most recent transaction from a first database log, copying data from the first database to the second database, updating the second database with transactions performed to the first database since the identified most recent transaction, and switching the second database for the first database.

A method for reorganizing a first database includes preparing a second database, identifying a most recent transaction from a first database log, copying data from the first database to the second database, updating the first database and the second database with transactions performed to the first database since the identified most recent transaction, and switching the second database for the first database.

A method for moving a first database from a first device to a second device includes copying data from the first database on the first device to a second database on the second device, updating second database to reflect changes that have been made to the first database since the commencement of the copying, and switching the second database for the first database.

A system for reorganizing a first database includes a copying unit for copying data from the first database to a second database having an index structure different from that of the first database, an updating unit for updating the second database to reflect changes that have been made to the first database since the commencement of the copying, and a switching unit for switching the second database for the first database.

A system for reorganizing a first database includes a preparing unit for preparing a second database, an identifying unit for identifying a most recent transaction from a first database log, a copying unit for copying data from the first database to the second

database, an updating unit for updating the second database with transactions performed to the first database since the identified most recent transaction, and a switching unit for switching the second database for the first database.

A system for moving a first database from a first device to a second device includes a copying unit for copying data from the first database on the first device to a second database on the second device, an updating unit for updating second database to reflect changes that have been made to the first database since the commencement of the copying, and a switching unit for switching the second database for the first database.

A computer system includes a processor and a program storage device readable by the computer system, embodying a program of instructions executable by the processor to perform method steps for reorganizing a first database. The method includes copying data from the first database to a second database having an index structure different from that of the first database, updating the second database to reflect changes that have been made to the first database since the commencement of the copying, and switching the second database for the first database.

A computer system includes a processor and a program storage device readable by the computer system, embodying a program of instructions executable by the processor to perform method steps for reorganizing a first database. The method includes preparing a second database, identifying a most recent transaction from a first database log, copying data from the first database to the second database, updating the second database with transactions performed to the first database since the identified most recent transaction, and switching the second database for the first database.

A computer system includes a processor and a program storage device readable by the computer system, embodying a program of instructions executable by the processor to perform method steps for reorganizing a first database. The method includes preparing a second database, identifying a most recent transaction from a first database log, copying data from the first database to the second database, updating the first database and the second database with transactions performed to the first database since the identified most recent transaction, and switching the second database for the first database.

A computer system includes a processor and a program storage device readable

by the computer system, embodying a program of instructions executable by the processor to perform method steps for moving a first database from a first device to a second device, the method includes copying data from the first database on the first device to a second database on the second device, updating second database to reflect changes that have been made to the first database since the commencement of the copying, and switching the second database for the first database.

BRIEF DESCRIPTION OF THE DRAWINGS

A more complete appreciation of the present disclosure and many of the attendant advantages thereof will be readily obtained as the same becomes better understood by reference to the following detailed description when considered in connection with the accompanying drawings, wherein:

FIG. 1 is a block diagram showing an embodiment of the present disclosure;

FIG. 2 shows a is a flow chart showing a method for re-organizing a database according to an embodiment of the present disclosure;

FIG. 3 shows a is a flow chart showing a method for copying a database according to an embodiment of the present disclosure;

FIG. 4 is a flow chart showing a method for re-organizing a database according to another embodiment of the present disclosure;

FIG. 5 is a block diagram showing a system according to an embodiment of the present disclosure; and

FIG. 6 shows an example of a computer system capable of implementing the method and apparatus according to embodiments of the present disclosure.

DETAILED DESCRIPTION

In describing the preferred embodiments of the present disclosure illustrated in the drawings, specific terminology is employed for sake of clarity. However, the present disclosure is not intended to be limited to the specific terminology so selected, and it is to be understood that each specific element includes all technical equivalents which operate in a similar manner.

Embodiments of the present disclosure allow for the re-organization of database

indexes and/or the moving of a database from one device to another without interrupting database functionality for extended periods of time.

Fig. 1 is a block diagram showing an embodiment of the present disclosure. An original database 11 may handle I/O activity such as, for example, query requests and/or database updates. The original database 11 may be an indexed database, for example a fixed-indexed database or a dynamically indexed database, containing one or more indexes 12. The original database 11 may additionally contain a transaction log 13 (also referred to as an audit log). The transaction log 13 may be part of the index itself or a separate file on the same or different data storage device. The transaction log 13 may be a log that records every modification that is made to the original database 11. The transaction log 13 may therefore be used to back out database modifications and/or re-apply database modifications in the event that there is a problem with the integrity of the data stored on the database.

Embodiments of the present disclosure utilize a second database 14 that may be located either on the same data storage device as the original database 11 or on a different data storage device. This second database 14 may be set up to store the data from the first database 11. The second database 14 may contain an index 15 that has been re-organized. The second database 14 may be used to handle I/O activity intended for the first database 11 allowing for access to the database 11 data without significant interruption. The second database 14 may then continue to function as the primary database handling I/O activity and maintaining a second database transaction log 16.

Fig. 2 is a flow chart showing a method for re-organizing a database according to an embodiment of the present disclosure. It may first be determined whether a new index structure is desired (Step S21). A new index structure may be desired, for example, when a database index requires re-organization. Examining the control statements of the DBMS may help determine whether a new index structure is required. If it is determined that a new index structure is not required (No, Step S21) then the determination may be made again (Step S21) at a later point.

If it is determined that a new index structure is required (Yes, Step S21), then a second database may be prepared, for example, the second database may be created or an existing second database may be initialized (Step S22). It may be beneficial to check to

see that the appropriate requirements for creating the second database are satisfied before the second database is created. For example, it may be beneficial to check to see that enough free space remains on the device that the second database is to be created on. There should be enough free space to accommodate the existing data in the original database plus the new index structure. Depending on the file-structure of the platform upon which the second database is to be located, the check for free space may not be necessary.

Next, the transaction log of the original database may be examined to determine the identity of the last transaction to be logged to the transaction log (Step S23). Then, the data content of the original database may be copied to the created second database (Step S24). Since the index structure of the new database might be different from the index structure of the old database, a "get" command used to read data from the original database and a "put" command used to write data to the second database may each refer to their own index structure. For example, data taken from a particular address on the original database may not occupy the same address once copied to the second database.

After all data has been copied to the second database (Step S24), all transactions that have occurred on the original database since the copying began may be applied to the second database (Step S25). These transactions may be determined by examining the transaction log of the original database and appliying all transactions that occurred after the last transaction as identified in Step S23. Care may be given to ensure that the transactions are applied to the second database in a way that accounts for the new index structure of the second database.

Next an exclusive lock may be placed on the original database to ensure that no further changes may be made to the original database (Step S26). The transaction log may then be reexamined to ensure that the second database is properly synchronized with the original database (Step S27). This may entail applying updates to the second database that were made on the original database while second database was updated in Step S25. This step is likely to be very short and therefore the length of time the database is locked for is minimized.

While the database is locked, the second database may be switched for the original database (Step S28). This may entail updating the I/O control blocks of the

applications that utilize the database, for example the DBMS, so that they point to the second database rather than the original database. The exclusive lock may then be removed allowing I/O activity to be directed to the second database (Step S29).

After the above method has been completed, the original database may be disabled to prevent applications from accessing the original database. This will help to ensure that all applications using the database have successfully been switched to the second database.

This process may be repeated every time a new index structure is desired. For example, the resources of the original database may be used as a subsequent database when it is determined that the index structure of the second database is to be renewed.

Fig. 3 is a flow chart showing a method for copying a database according to an embodiment of the present disclosure. Aspects of this embodiment of the present disclosure closely follow the embodiment described above and shown in Fig. 2. Details described above may therefore be used in carrying out corresponding steps in the current embodiment. First, a second database may be created (Step S32). Next, the transaction log of the original database may be examined to determine the identity of the last transaction to be logged to the transaction log (Step S33). Then the data content of the original database may be copied to the created second database (Step S34). After all data has been copied to the second database (Step S34), all transactions that have occurred on the original database since the copying began may be applied to the second database (Step S35). Next an exclusive lock may be placed on the original database to ensure that no further changes may be made to the original database (Step S36). The transaction log may then be reexamined to ensure that the second database is properly synchronized with the original database (Step S37). While the database is locked, the second database is switched for the original database (Step S38). The exclusive lock may then be removed allowing I/O activity to be directed to the second database (Step S39).

Fig. 4 is a flow chart showing a method for re-organizing a database according to another embodiment of the present disclosure. Aspects of this embodiment of the present disclosure closely follow the embodiment described above and shown in Fig. 2. Details described above may therefore be used in carrying out corresponding steps in the current embodiment.

It may first be determined whether a new index structure is desired (Step S41). If it is determined that a new index structure is not required (No, Step S41) then the determination may be made again (Step S41) at a later point. If it is determined that a new index structure is required (Yes, Step S41), then a second database may be created (Step S42). The transaction log of the original database may be examined to determine the identity of the last transaction to be logged to the transaction log (Step S43). Then the data content of the original database may be copied to the created second database (Step S44) while subsequent database transactions are applied to both the original database as well as the second database (Step S45). These steps (S44 and S45) may be carried out simultaneously, for example, using parallel processes. There may be multiple ways to apply the transactions to each database simultaneously. For example, if the transaction relates to data that has already been copied, the transaction may be successfully applied to each database while if the transaction relates to data that has yet to be copied, the transaction may be successfully applied to the original database where it will be later copied to the second database. After all data has been copied to the second database (Step S44), the second database may be switched for the original database (Step S46).

Fig. 5 is a block diagram showing a system according to an embodiment of the present disclosure. A copying unit 53 may copy data from a first database 51 on a first device 52 to a second database 55 on a second device 56. The second database 55 may have a different index structure than the first database 51. An updating unit 54 may then update the second database 55 to reflect changes that may have been made to the fist database 51 since the commencement of the copying. A switching unit 57 may then switch the second database 55 for the first database 51.

Fig. 6 shows an example of a computer system which may implement the method and system of the present disclosure. The system and method of the present disclosure may be implemented in the form of a software application running on a computer system, for example, a mainframe, personal computer (PC), handheld computer, server, etc. The software application may be stored on a recording media locally accessible by the computer system and accessible via a hard wired or wireless connection to a network, for example, a local area network, or the Internet.

The computer system referred to generally as system 1000 may include, for

example, a central processing unit (CPU) 1001, random access memory (RAM) 1004, a printer interface 1010, a display unit 1011, a local area network (LAN) data transmission controller 1005, a LAN interface 1006, a network controller 1003, an internal bus 1002, and one or more input devices 1009, for example, a keyboard, mouse etc. As shown, the system 1000 may be connected to a data storage device, for example, a hard disk, 1008 via a link 1007.

The above specific embodiments are illustrative, and many variations can be introduced on these embodiments without departing from the spirit of the disclosure or from the scope of the appended claims. For example, elements and/or features of different illustrative embodiments may be combined with each other and/or substituted for each other within the scope of this disclosure and appended claims.

What is claimed is:

1. A method for reorganizing a first database, comprising:

copying data from the first database to a second database having an index structure different from that of the first database;

updating the second database to reflect changes that have been made to the first database since the commencement of the copying; and

switching the second database for the first database.

2. The method of claim 1, wherein prior to copying the data from the first database to the second database, a most recent transaction made to the first database is identified and when updating the second database to reflect changes that have been made to the first database since the commencement of the copying, those transactions that were made to the first database since the identified most recent transaction are applied to the second database.

3. The method of claim 1, additionally comprising:

locking the first database from further transactions, and

applying any transactions that have been applied to the first database but not the second database to the second database, after updating the second database to reflect changes that have been made to the first database since the commencement of the copying and before switching the second database for the first database, and

unlocking the first database after switching the second database for the first database.

4. The method of claim 1, wherein the index structure of the second database is a reorganization of the index structure of the first database.

5. A method for reorganizing a first database, comprising:

preparing a second database;

identifying a most recent transaction from a first database log;

copying data from the first database to the second database;

updating the second database with transactions performed to the first database since the identified most recent transaction; and

switching the second database for the first database.

6. The method of claim 5, further comprising:

locking the first database after updating the second database with transactions performed to the first database since the identified most recent transaction; and

synchronizing the second database with the first database.

7. A method for reorganizing a first database, comprising:

preparing a second database;

identifying a most recent transaction from a first database log;

copying data from the first database to the second database;

updating the first database and the second database with transactions performed to the first database since the identified most recent transaction; and

switching the second database for the first database.

8. A method for moving a first database from a first device to a second device, comprising:

copying data from the first database on the first device to a second database on the second device;

updating the second database to reflect changes that have been made to the first database since the commencement of the copying; and

switching the second database for the first database.

9. A system for reorganizing a first database, comprising:

a copying unit for copying data from the first database to a second database having an index structure different from that of the first database;

an updating unit for updating the second database to reflect changes that have been made to the first database since the commencement of the copying; and

a switching unit for switching the second database for the first database.

10. The system of claim 9, wherein prior to copying the data from the first database to the second database, a most recent transaction made to the first database is identified and when updating the second database to reflect changes that have been made to the first database since the commencement of the copying, those transactions that were made to the first database since the identified most recent transaction are applied to the second database.

11. The system of claim 9, additionally comprising:

a locking unit for locking the first database from further transactions, and

an applying unit for applying any transactions that have been applied to the first database but not the second database to the second database, after updating the second database to reflect changes that have been made to the first database since the commencement of the copying and before switching the second database for the first database, and

an unlocking unit for unlocking the first database after switching the second database for the first database.

12. The system of claim 9, wherein the index structure of the second database is a reorganization of the index structure of the first database.

13. A system for reorganizing a first database, comprising:

a preparing unit for preparing a second database;

an identifying unit for identifying a most recent transaction from a first database log;

a copying unit for copying data from the first database to the second database;

an updating unit for updating the second database with transactions performed to the first database since the identified most recent transaction; and

a switching unit for switching the second database for the first database.

14. The system of claim 13, further comprising:

a locking unit for locking the first database after updating the second database with transactions performed to the first database since the identified most recent transaction; and

a synchronizing unit for synchronizing the second database with the first database.

15. A system for reorganizing a first database, comprising:

a preparing unit for preparing a second database;

an identifying unit for identifying a most recent transaction from a first database log;

a copying unit for copying data from the first database to the second database;

an updating unit for updating the first database and the second database with transactions performed to the first database since the identified most recent transaction; and

a switching unit for switching the second database for the first database.

16. A system for moving a first database from a first device to a second device, comprising:

a copying unit for copying data from the first database on the first device to a second database on the second device;

an updating unit for updating second database to reflect changes that have been made to the first database since the commencement of the copying; and

a switching unit for switching the second database for the first database.

17. A computer system comprising:

a processor; and

a program storage device readable by the computer system, including computer executable code executable by the processor for reorganizing a first database, the computer executable code comprising:

code for copying data from the first database to a second database having an index structure different from that of the first database;

code for updating the second database to reflect changes that have been made to the first database since the commencement of the copying; and

code for switching the second database for the first database.

18. The computer system of claim 17, wherein prior to copying the data from the first database to the second database, a most recent transaction made to the first database is identified and when updating the second database to reflect changes that have been made to the first database since the commencement of the copying, those transactions that were made to the first database since the identified most recent transaction are applied to the second database.

19. The computer system of claim 17, additionally comprising:

code for locking the first database from further transactions, and

code for applying any transactions that have been applied to the first database but not the second database to the second database, after updating the second database to reflect changes that have been made to the first database since the commencement of the copying and before switching the second database for the first database, and

code for unlocking the first database after switching the second database for the first database.

20. The computer system of claim 17, wherein the index structure of the second database is a reorganization of the index structure of the first database.

21. A computer system comprising:

a processor; and

a program storage device readable by the computer system, including computer executable code executable by the processor for reorganizing a first database, the computer executable code comprising:

code for preparing a second database;

code for identifying a most recent transaction from a first database log;

code for copying data from the first database to the second database;

code for updating the second database with transactions performed to the first database since the identified most recent transaction; and

code for switching the second database for the first database.

22. The computer system of claim 21, further comprising:

code for locking the first database after updating the second database with transactions performed to the first database since the identified most recent transaction; and

code for synchronizing the second database with the first database.

23. A computer system comprising:

a processor; and

a program storage device readable by the computer system, including computer executable code executable by the processor for reorganizing a first database, the computer executable code comprising:

code for preparing a second database;

code for identifying a most recent transaction from a first database log;

code for copying data from the first database to the second database;

code for updating the first database and the second database with transactions performed to the first database since the identified most recent transaction; and

code for switching the second database for the first database.

24. A computer system comprising:

a processor; and

a program storage device readable by the computer system, including computer executable code executable by the processor for moving a first database from a first device to a second device, the computer executable code comprising:

code for copying data from the first database on the first device to a second database on the second device;

code for updating second database to reflect changes that have been made to the first database since the commencement of the copying; and

code for switching the second database for the first database.

Fig. 1

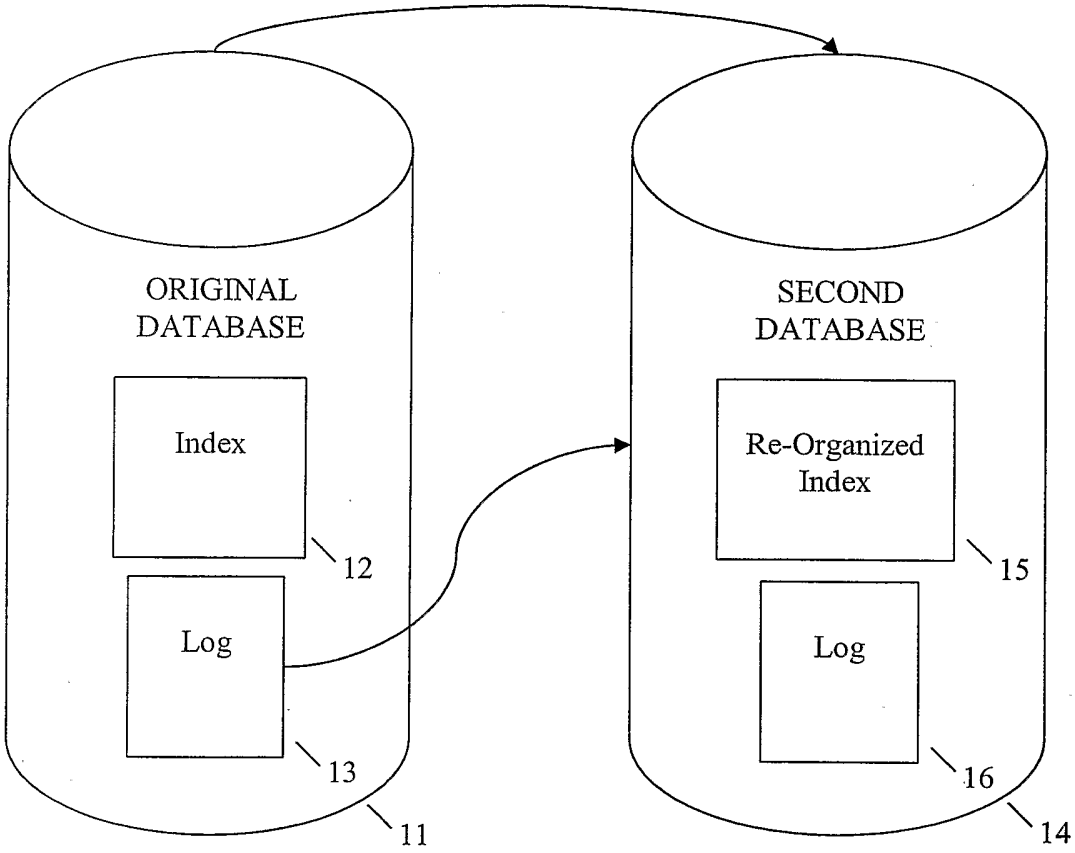

ORIGINAL
DATABASE
Index
12
Log
13
11
SECOND
DATABASE
Re-Organized
Index
15
Log
16
14

Fig. 2

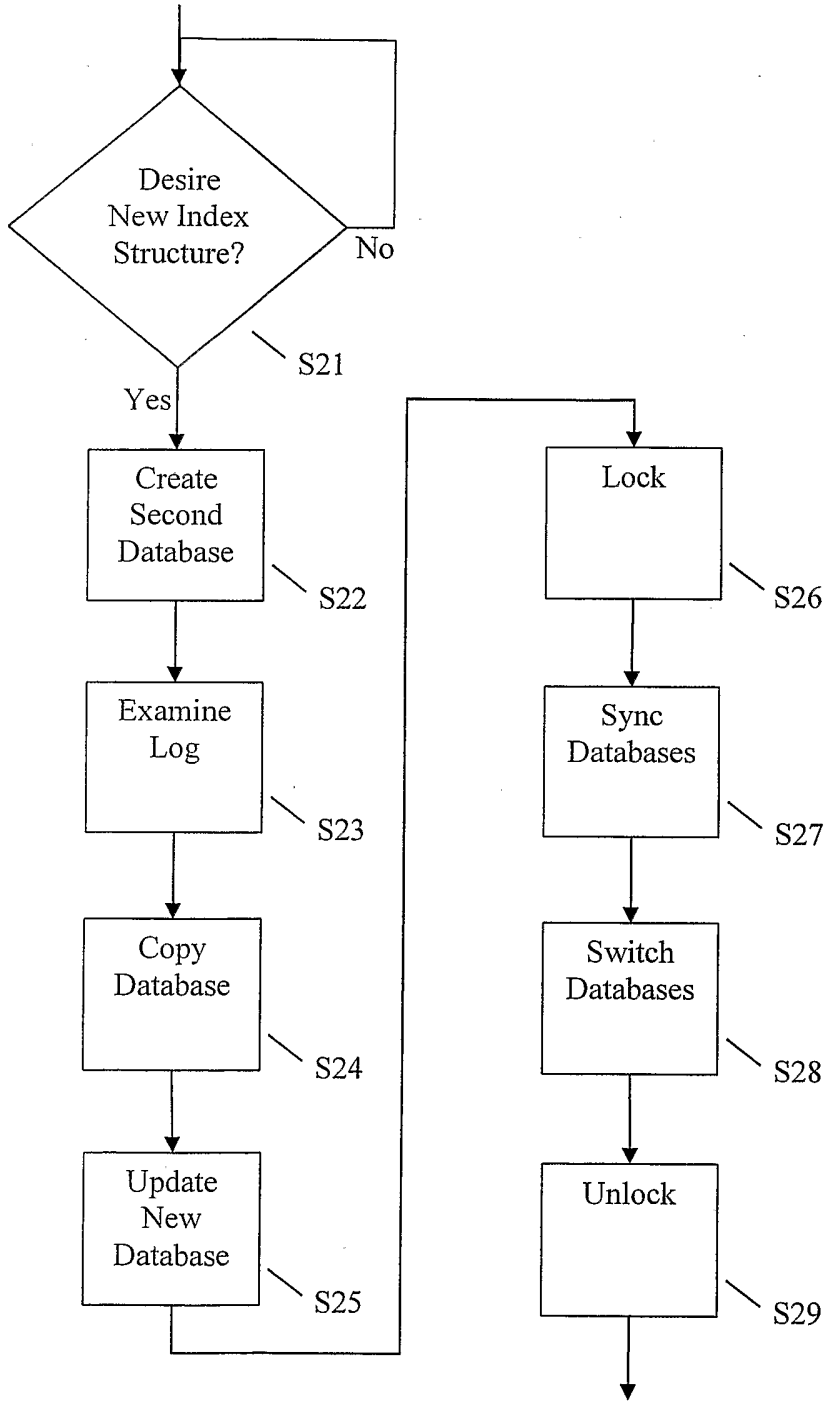

Desire New Index Structure?
No
S21
Yes
Create Second Database
S22
Examine Log
S23
Copy Database
S24
Update New Database
S25
Lock
S26
Sync Databases
S27
Switch Databases
S28
Unlock
S29

Fig. 3

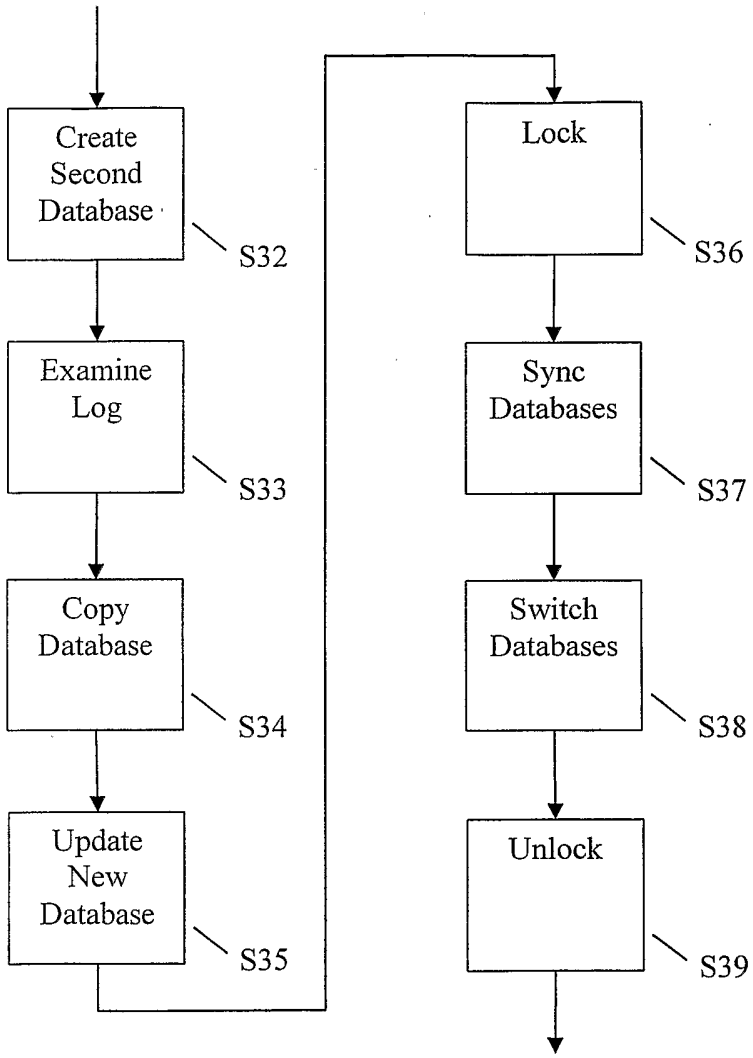

Create Second Database
S32
Examine Log
S33
Copy Database
S34
Update New Database
S35
Lock
S36
Sync Databases
S37
Switch Databases
S38
Unlock
S39

Fig. 4

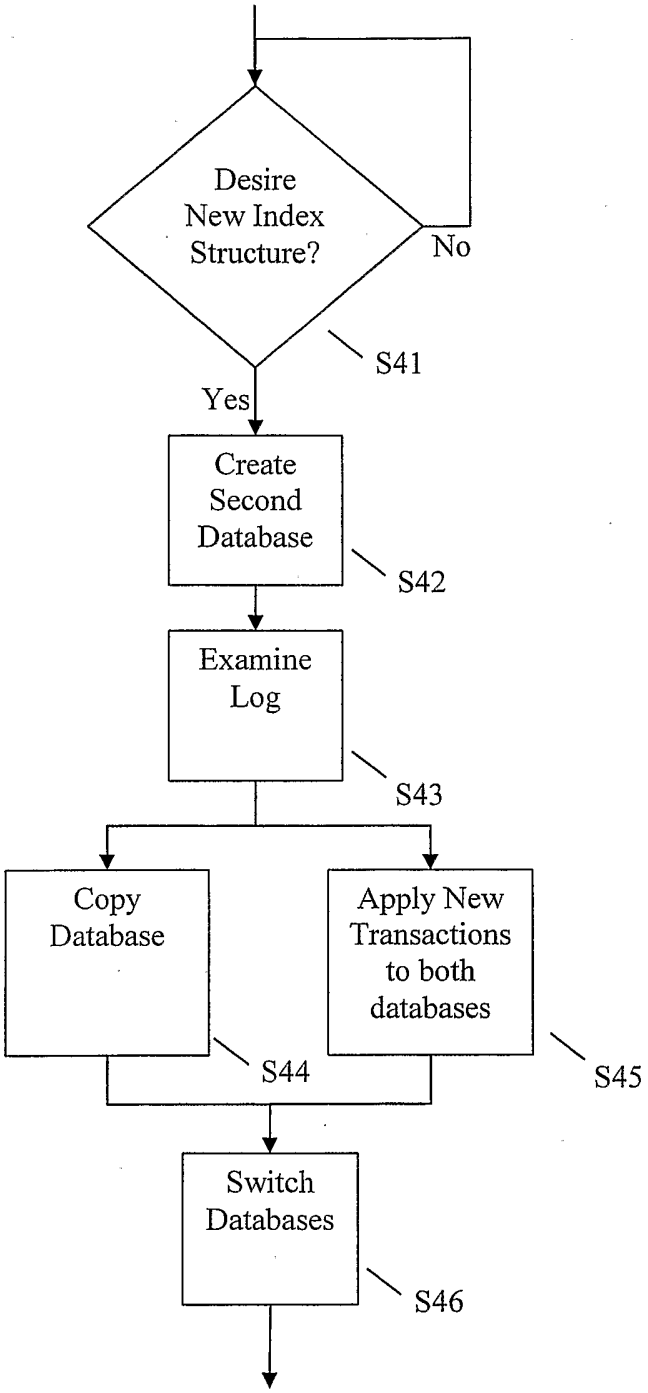

Desire New Index Structure?
No
S41
Yes
Create Second Database
S42
Examine Log
S43
Copy Database
Apply New Transactions to both databases
S44
S45
Switch Databases
S46

Fig. 5

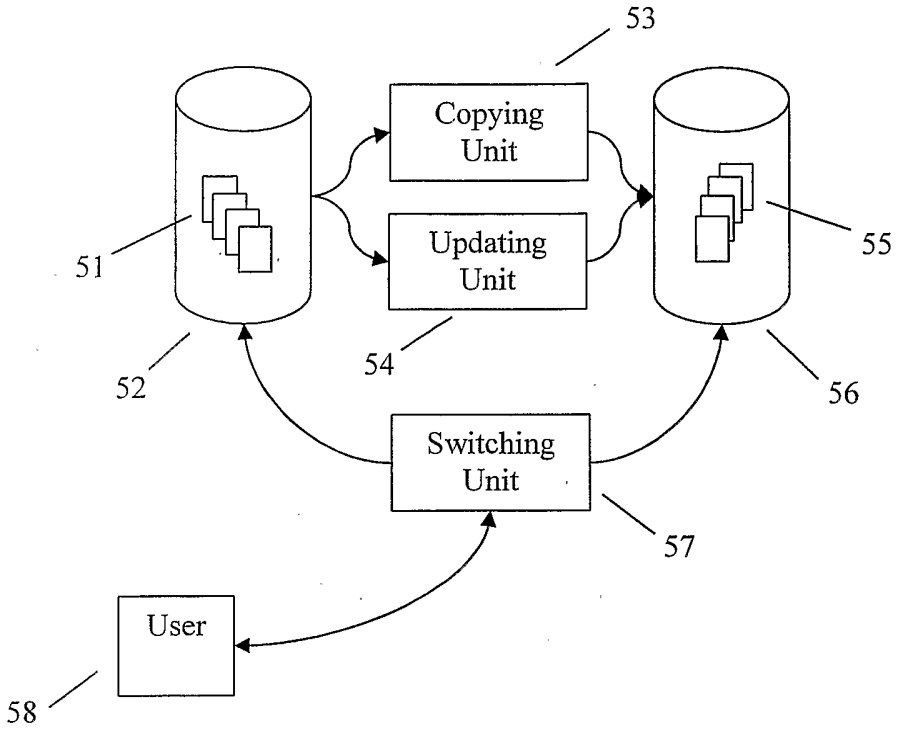
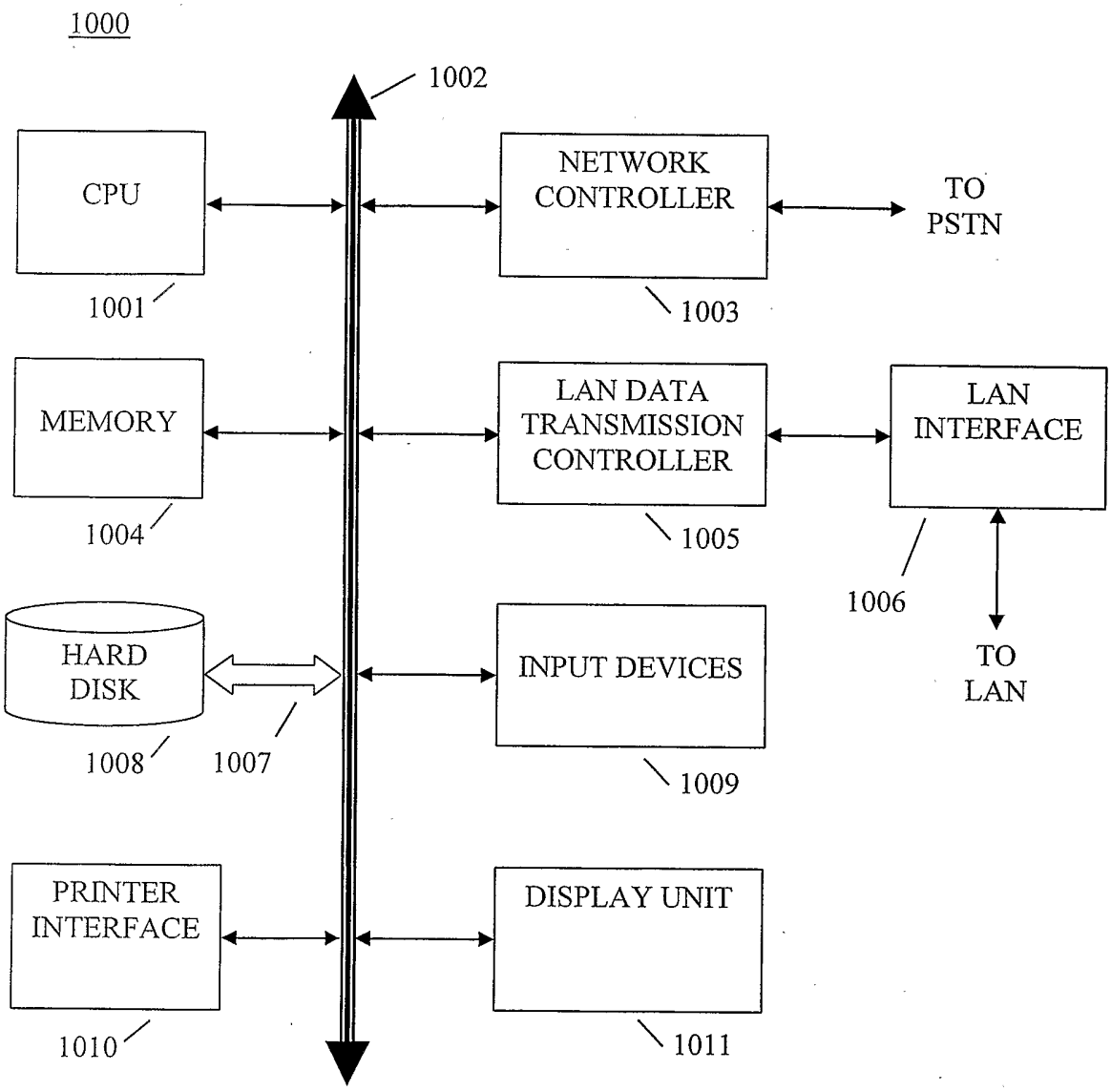

53
Copying Unit
Updating Unit
51
55
52
54
56
Switching Unit
57
User
58

Fig. 6


1000
1002
CPU
1001
NETWORK CONTROLLER
1003
TO PSTN
MEMORY
1004
LAN DATA TRANSMISSION CONTROLLER
1005
LAN INTERFACE
1006
TO LAN
HARD DISK
1008
1007
INPUT DEVICES
1009
PRINTER INTERFACE
1010
DISPLAY UNIT
1011