US 20080046487A1

(19) United States

(12) Patent Application Publication (10) Pub. No.: US 2008/0046487 A1

Kamada et al. (43) Pub. Date: Feb. 21, 2008

(54) DATA PROCESSING SYSTEM AND METHOD WITH DATA SHARING FOR THE SAME

(76) Inventors: **Yasuhiro Kamada**, Yokohama (JP); **Nobuo Kawamura**, Atsugi (JP); **Junko Katsumata**, Yokohama (JP); **Kota Yamaguchi**, Yamato (JP); **Keisuke Toda**, Sagamihara (JP)

Correspondence Address:
**ANTONELLI, TERRY, STOUT & KRAUS, LLP**
**1300 NORTH SEVENTEENTH STREET**
**SUITE 1800**
**ARLINGTON, VA 22209-3873 (US)**

(21) Appl. No.: **11/840,476**

(22) Filed: **Aug. 17, 2007**

**Related U.S. Application Data**

(63) Continuation of application No. 10/930,765, filed on Sep. 1, 2004, now Pat. No. 7,275,072.

(30) **Foreign Application Priority Data**

May 27, 2004 (JP) ..... 2004-157619

**Publication Classification**

(51) **Int. Cl.**
*G06F 7/00* (2006.01)

(52) **U.S. Cl.** ..... **707/205**

(57) **ABSTRACT**

If relocation of a table with a high reference frequency is made possible, the database management method and system are arranged to improve performance according to the number of increased nodes and accumulate a reference frequency of each external storage unit allocated to each database management node. A data storage area for storing the table with a high reference frequency, located in the external storage unit, is specified as a shared data area to be referenced by plural nodes. The system is controlled to cause the nodes to refer to the shared table stored in the shared data area on any timing in parallel. This prevents load from being concentrated onto the node allocated to the data area.

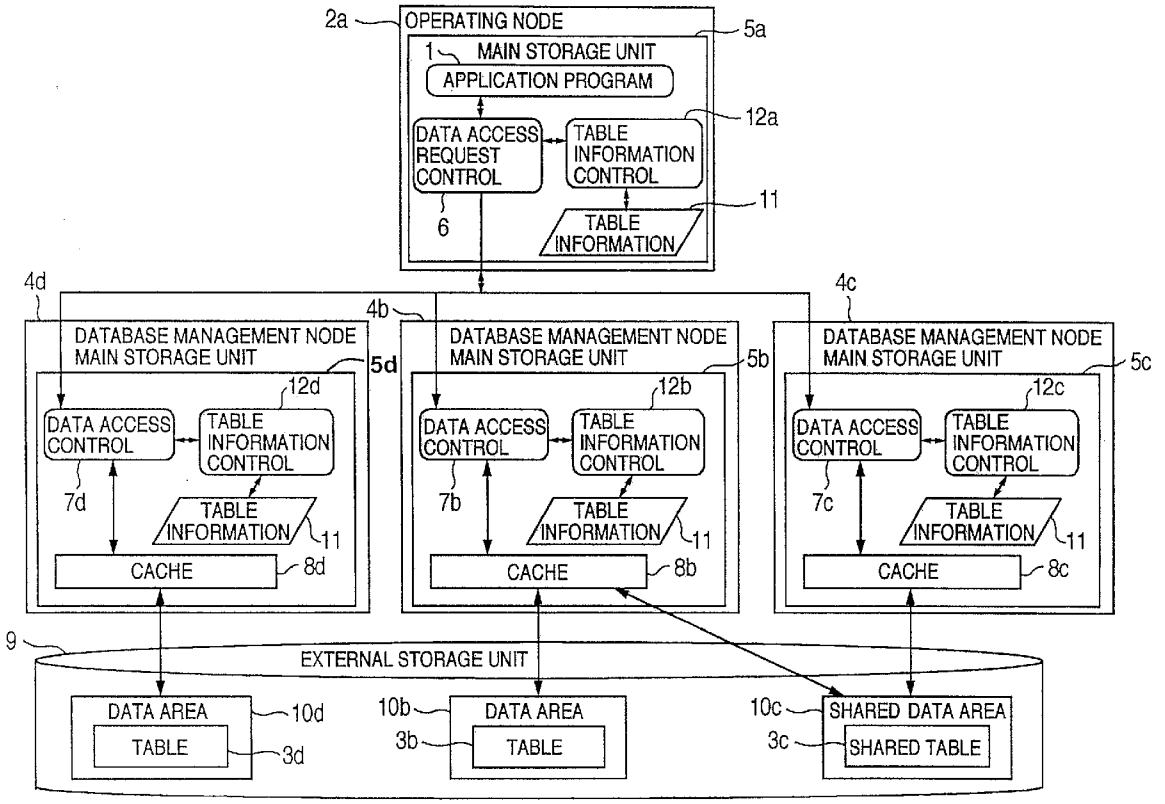


FIG.1

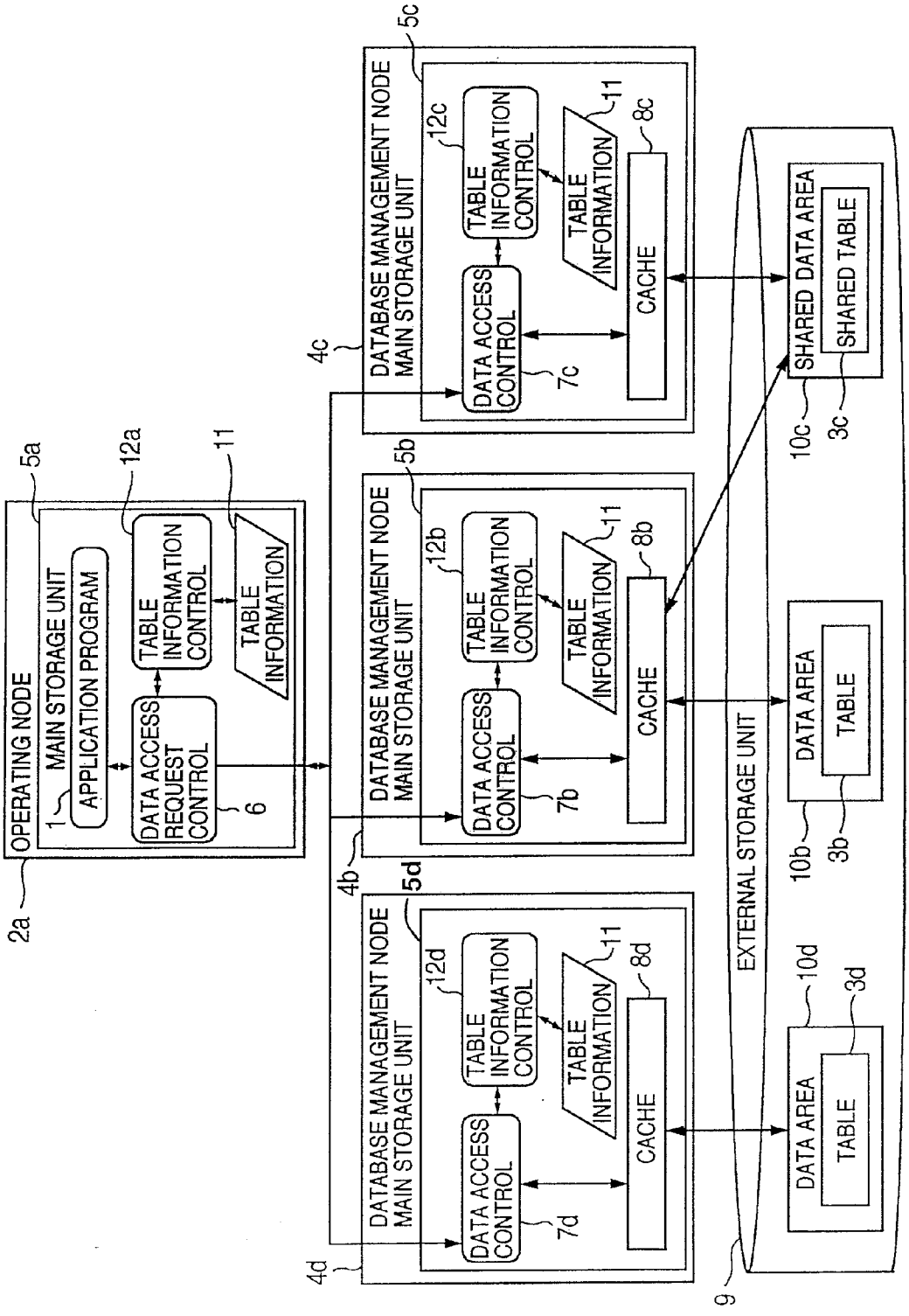

OPERATING NODE
MAIN STORAGE UNIT
APPLICATION PROGRAM
DATA ACCESS REQUEST CONTROL
TABLE INFORMATION CONTROL
TABLE INFORMATION
DATABASE MANAGEMENT NODE MAIN STORAGE UNIT
DATA ACCESS CONTROL
TABLE INFORMATION CONTROL
TABLE INFORMATION
CACHE
EXTERNAL STORAGE UNIT
DATA AREA
TABLE
SHARED DATA AREA
SHARED TABLE
1
2a
4b
4c
4d
5a
5b
5c
5d
6
7b
7c
7d
8b
8c
8d
9
10b
10c
10d
11
12a
12b
12c
12d
3b
3c
3d

FIG.2

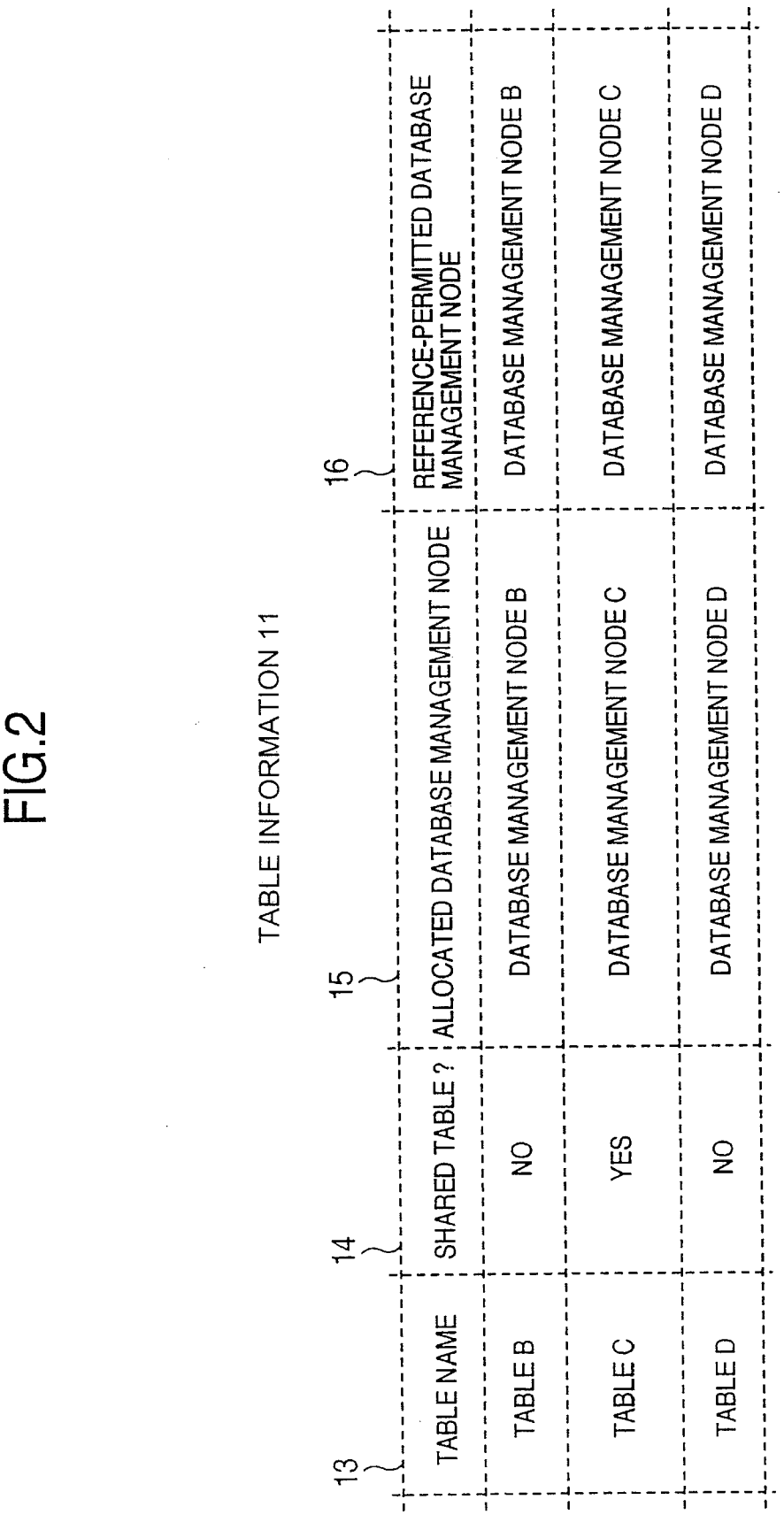


TABLE INFORMATION 11
13
14
15
16
TABLE NAME
SHARED TABLE ?
ALLOCATED DATABASE MANAGEMENT NODE
REFERENCE-PERMITTED DATABASE MANAGEMENT NODE
TABLE B
NO
DATABASE MANAGEMENT NODE B
DATABASE MANAGEMENT NODE B
TABLE C
YES
DATABASE MANAGEMENT NODE C
DATABASE MANAGEMENT NODE C
TABLE D
NO
DATABASE MANAGEMENT NODE D
DATABASE MANAGEMENT NODE D

FIG.3

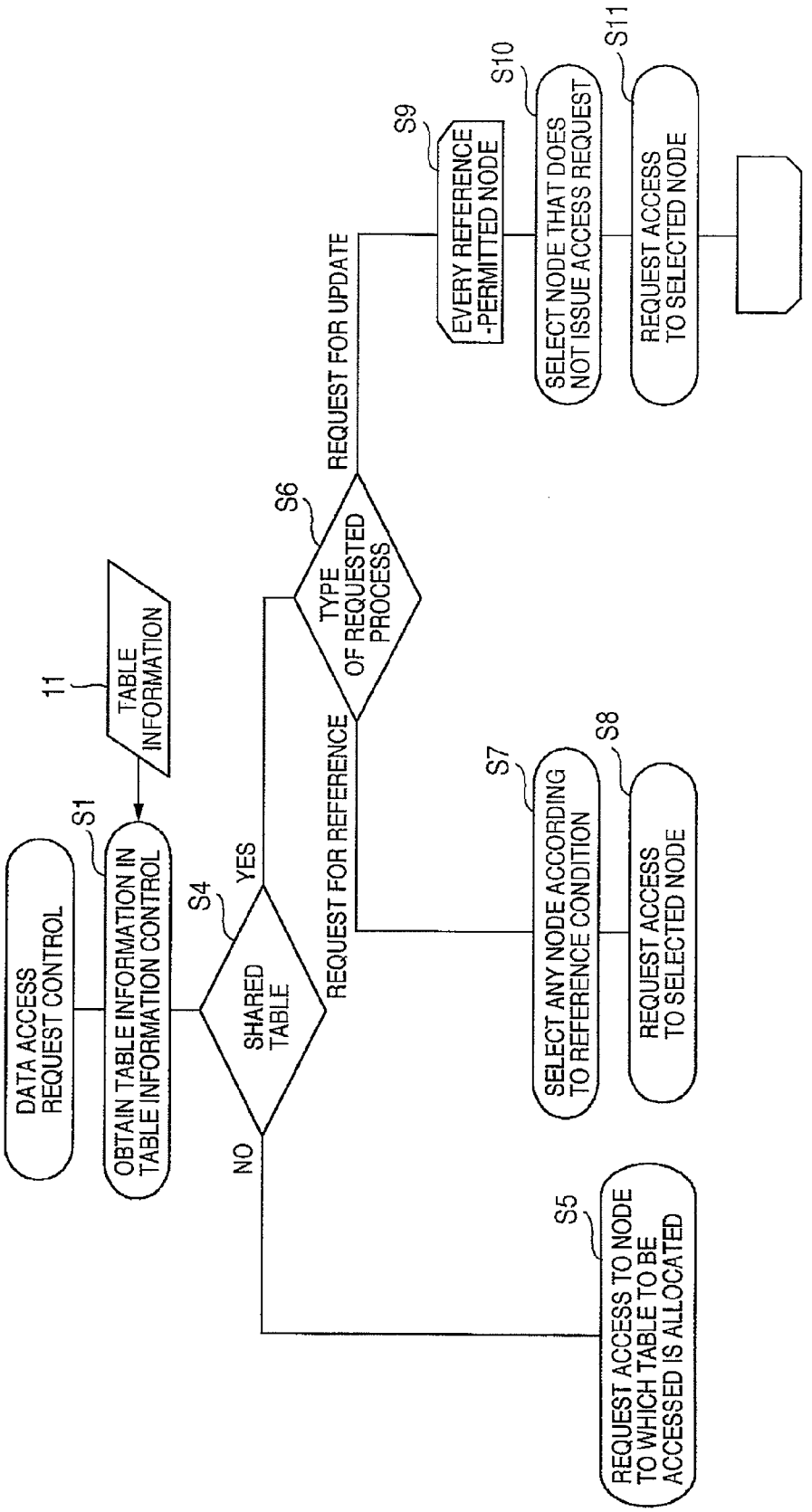

DATA ACCESS REQUEST CONTROL
S1
OBTAIN TABLE INFORMATION IN TABLE INFORMATION CONTROL
11
TABLE INFORMATION
S4
SHARED TABLE
YES
NO
S6
TYPE OF REQUESTED PROCESS
REQUEST FOR UPDATE
REQUEST FOR REFERENCE
S9
EVERY REFERENCE -PERMITTED NODE
S10
SELECT NODE THAT DOES NOT ISSUE ACCESS REQUEST
S11
REQUEST ACCESS TO SELECTED NODE
S7
SELECT ANY NODE ACCORDING TO REFERENCE CONDITION
S8
REQUEST ACCESS TO SELECTED NODE
S5
REQUEST ACCESS TO NODE TO WHICH TABLE TO BE ACCESSED IS ALLOCATED

FIG.4

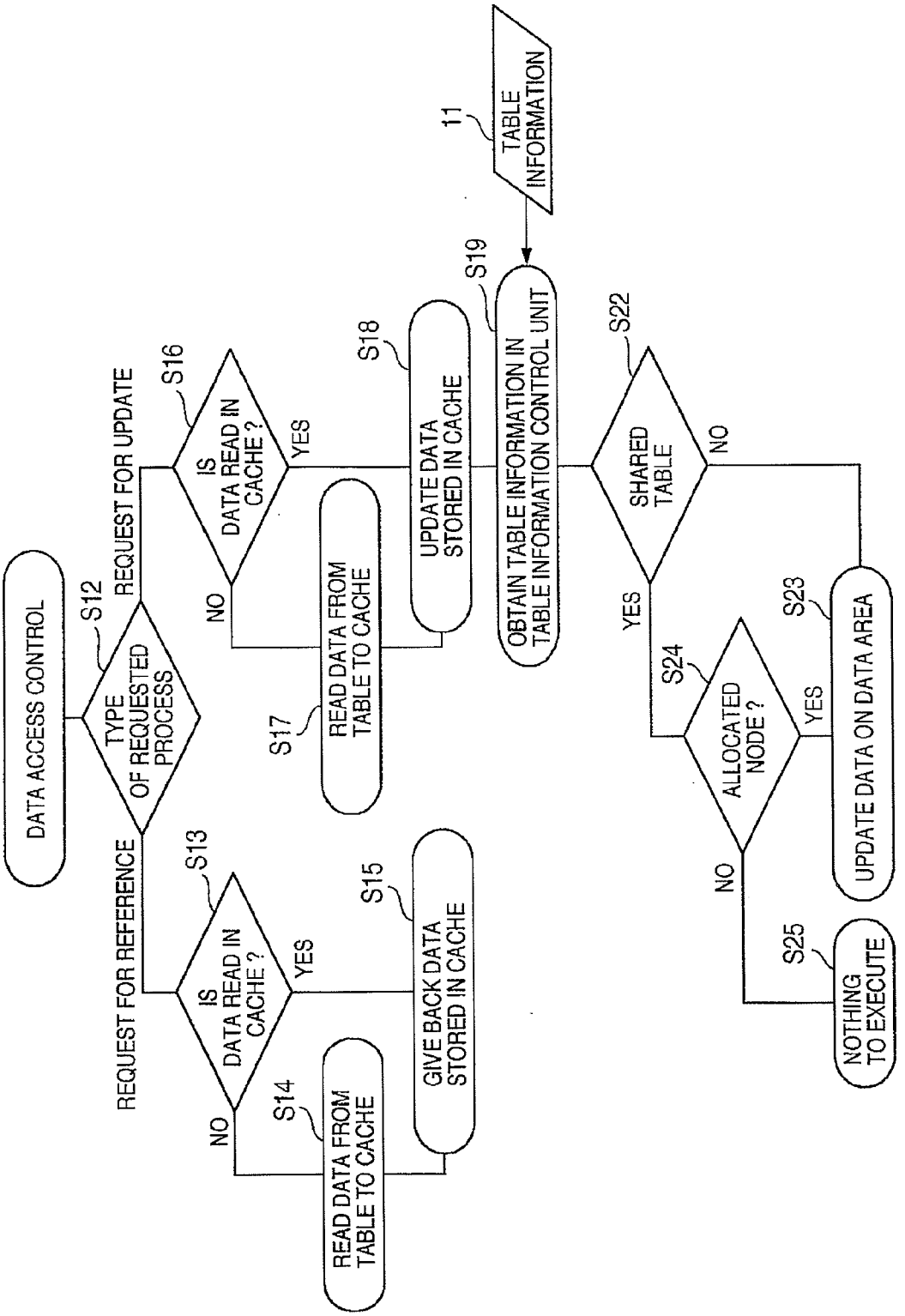

DATA ACCESS CONTROL
S12
TYPE OF REQUESTED PROCESS
REQUEST FOR REFERENCE
REQUEST FOR UPDATE
S13
IS DATA READ IN CACHE ?
YES
NO
S14
READ DATA FROM TABLE TO CACHE
S15
GIVE BACK DATA STORED IN CACHE
S16
IS DATA READ IN CACHE ?
YES
NO
S17
READ DATA FROM TABLE TO CACHE
S18
UPDATE DATA STORED IN CACHE
11
TABLE INFORMATION
S19
OBTAIN TABLE INFORMATION IN TABLE INFORMATION CONTROL UNIT
S22
SHARED TABLE
YES
NO
S24
ALLOCATED NODE ?
YES
NO
S23
UPDATE DATA ON DATA AREA
S25
NOTHING TO EXECUTE

FIG.5

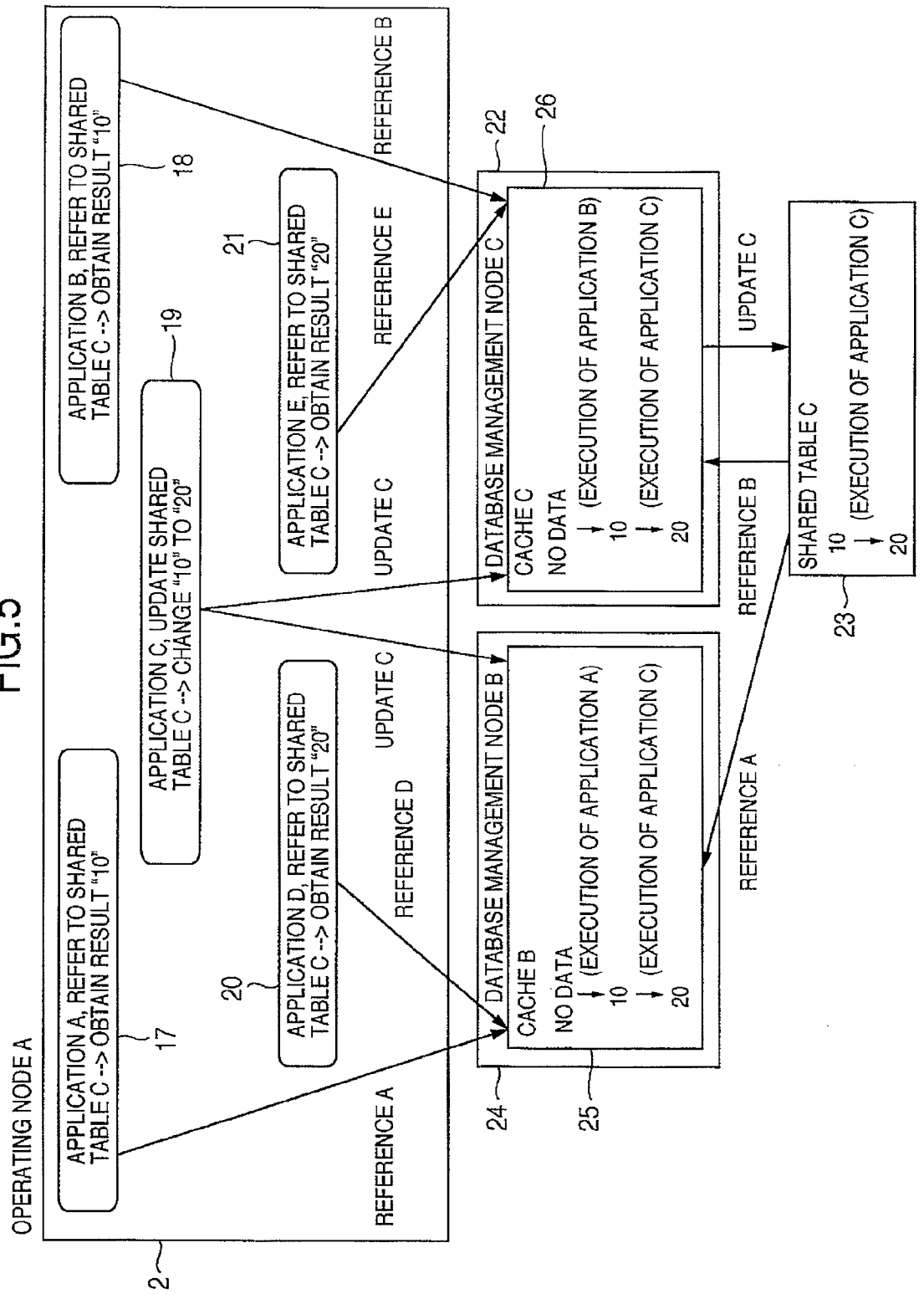

OPERATING NODE A
APPLICATION A, REFER TO SHARED TABLE C --> OBTAIN RESULT "10"
17
APPLICATION B, REFER TO SHARED TABLE C --> OBTAIN RESULT "10"
18
APPLICATION C, UPDATE SHARED TABLE C --> CHANGE "10" TO "20"
19
APPLICATION D, REFER TO SHARED TABLE C --> OBTAIN RESULT "20"
20
APPLICATION E, REFER TO SHARED TABLE C --> OBTAIN RESULT "20"
21
REFERENCE A
REFERENCE B
REFERENCE D
REFERENCE E
UPDATE C
UPDATE C
UPDATE C
2
DATABASE MANAGEMENT NODE B
CACHE B
NO DATA
(EXECUTION OF APPLICATION A)
10
(EXECUTION OF APPLICATION C)
20
24
25
DATABASE MANAGEMENT NODE C
CACHE C
NO DATA
(EXECUTION OF APPLICATION B)
10
(EXECUTION OF APPLICATION C)
20
22
26
SHARED TABLE C
10
(EXECUTION OF APPLICATION C)
20
23
REFERENCE A
REFERENCE B
UPDATE C

FIG.6

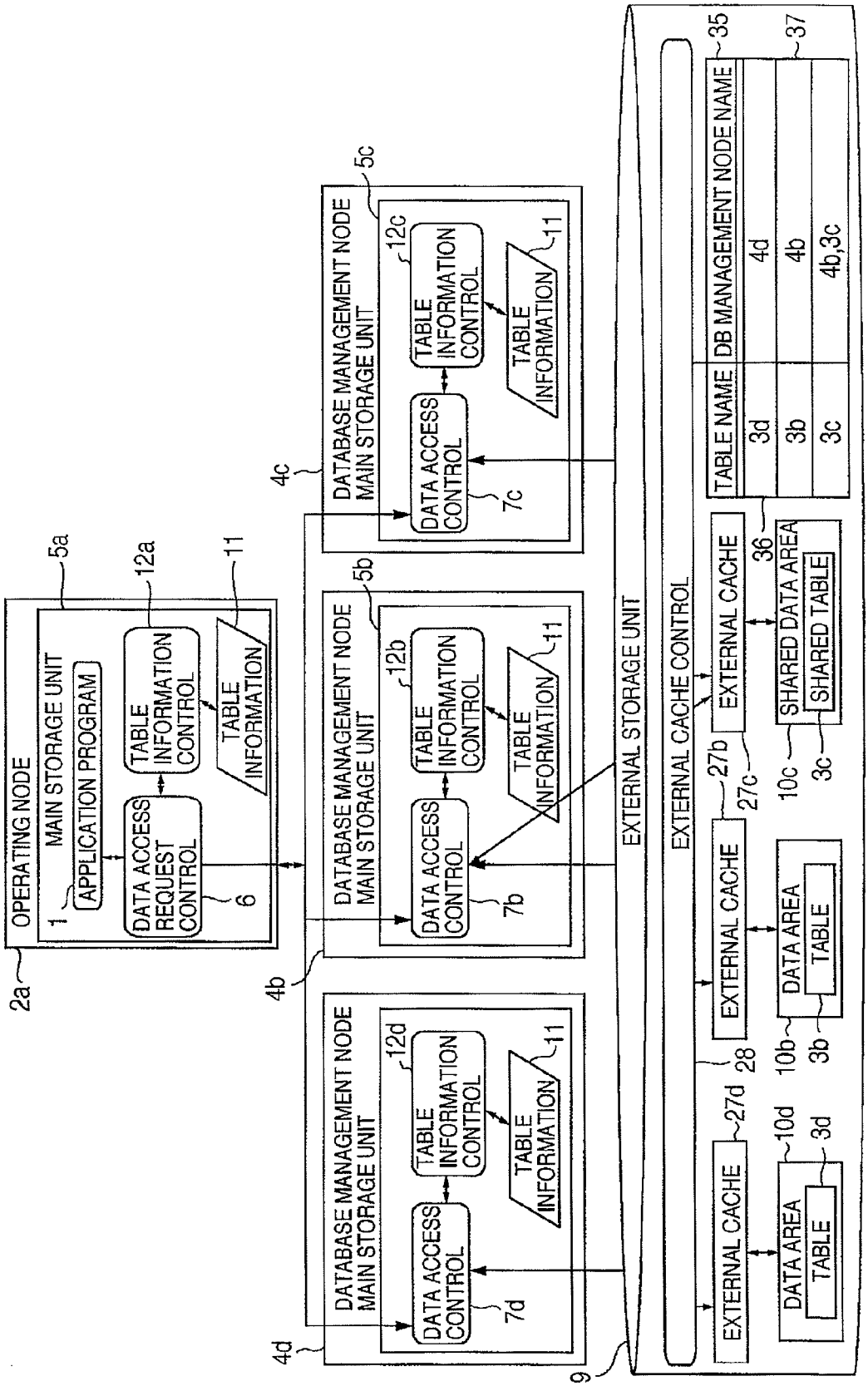

OPERATING NODE
MAIN STORAGE UNIT
APPLICATION PROGRAM
DATA ACCESS REQUEST CONTROL
TABLE INFORMATION CONTROL
TABLE INFORMATION
DATABASE MANAGEMENT NODE MAIN STORAGE UNIT
DATA ACCESS CONTROL
TABLE INFORMATION CONTROL
TABLE INFORMATION
EXTERNAL STORAGE UNIT
EXTERNAL CACHE CONTROL
EXTERNAL CACHE
DATA AREA
TABLE
SHARED DATA AREA
SHARED TABLE
TABLE NAME
DB MANAGEMENT NODE NAME
3d
4d
3b
4b
3c
4b,3c
1
2a
4b
4c
4d
5a
5b
5c
6
7b
7c
7d
9
10b
10c
10d
11
12a
12b
12c
12d
27b
27c
27d
28
35
36
37

FIG.7

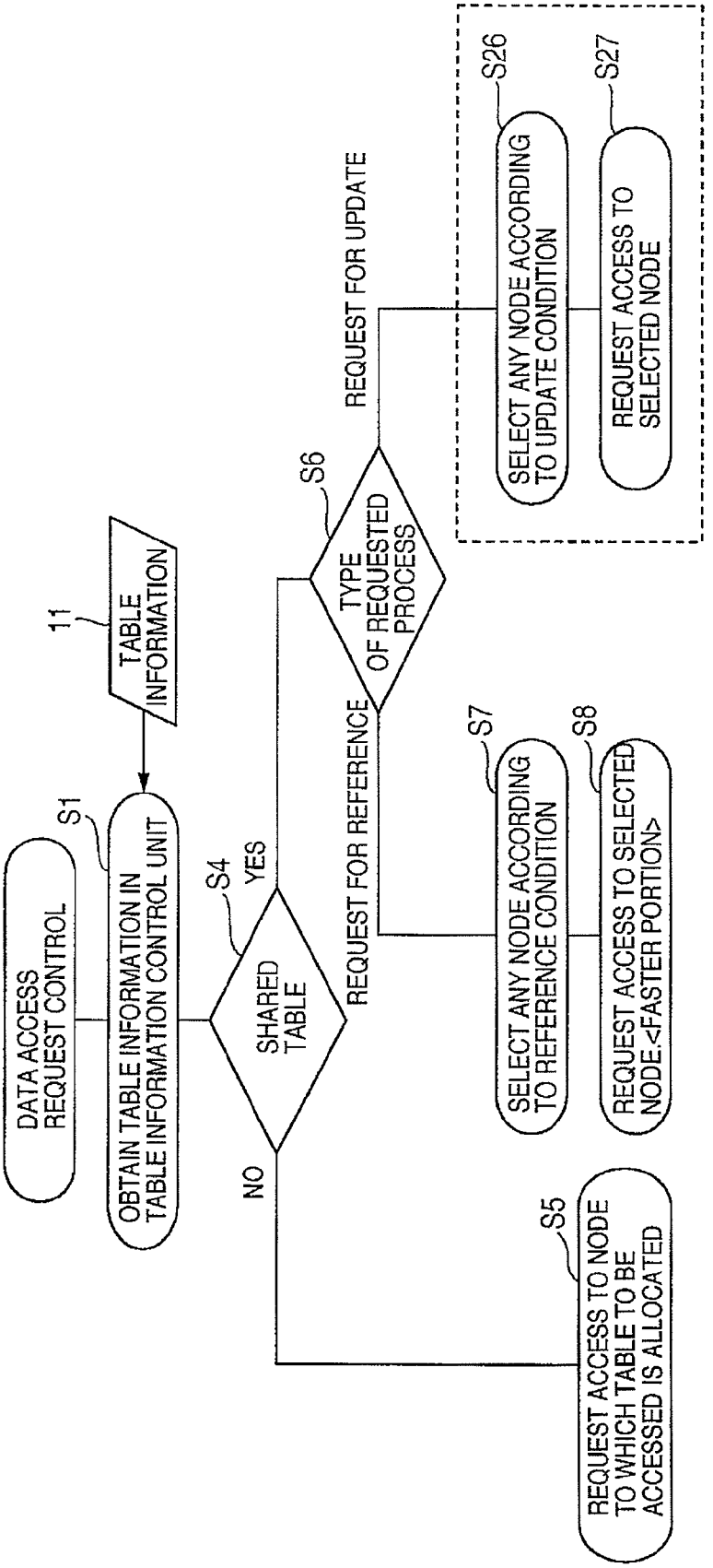

DATA ACCESS REQUEST CONTROL
S1
OBTAIN TABLE INFORMATION IN TABLE INFORMATION CONTROL UNIT
11
TABLE INFORMATION
S4
SHARED TABLE
YES
NO
S5
REQUEST ACCESS TO NODE TO WHICH TABLE TO BE ACCESSED IS ALLOCATED
S6
TYPE OF REQUESTED PROCESS
REQUEST FOR REFERENCE
REQUEST FOR UPDATE
S7
SELECT ANY NODE ACCORDING TO REFERENCE CONDITION
S8
REQUEST ACCESS TO SELECTED NODE.<FASTER PORTION>
S26
SELECT ANY NODE ACCORDING TO UPDATE CONDITION
S27
REQUEST ACCESS TO SELECTED NODE

FIG.8

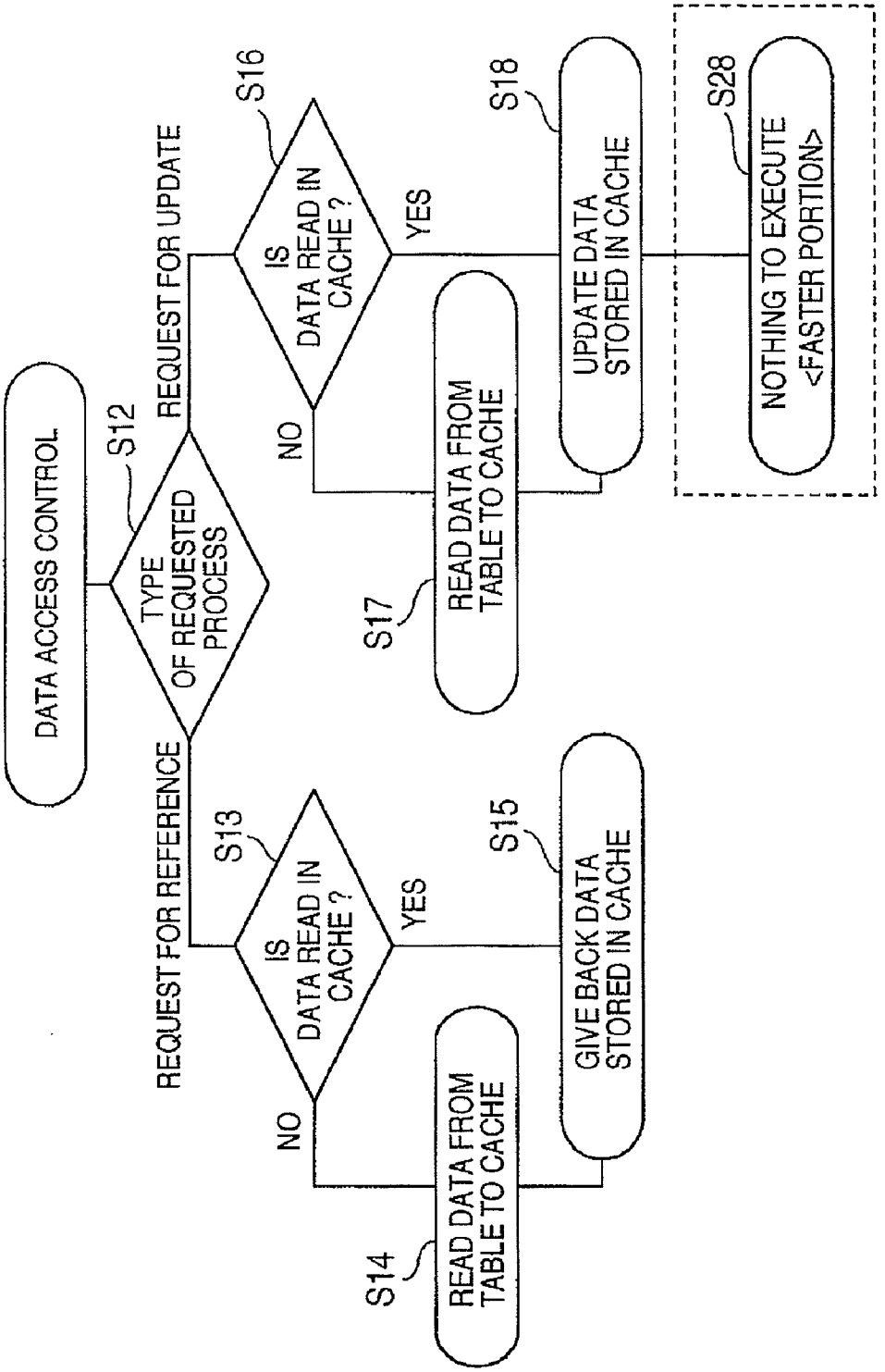

DATA ACCESS CONTROL
S12
TYPE OF REQUESTED PROCESS
REQUEST FOR REFERENCE
REQUEST FOR UPDATE
S13
IS DATA READ IN CACHE ?
YES
NO
S14
READ DATA FROM TABLE TO CACHE
S15
GIVE BACK DATA STORED IN CACHE
S16
IS DATA READ IN CACHE ?
YES
NO
S17
READ DATA FROM TABLE TO CACHE
S18
UPDATE DATA STORED IN CACHE
S28
NOTHING TO EXECUTE <FASTER PORTION>

FIG.9

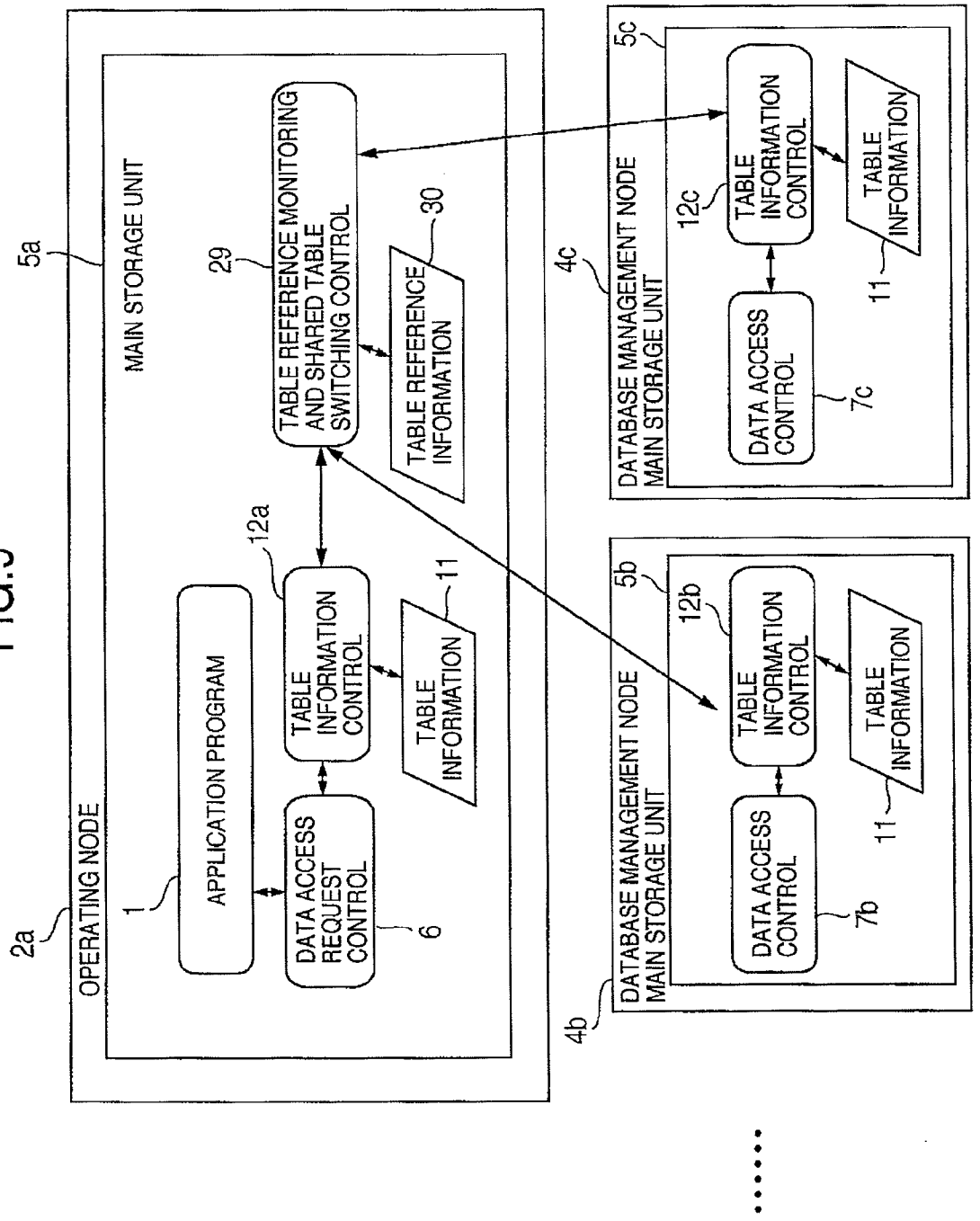
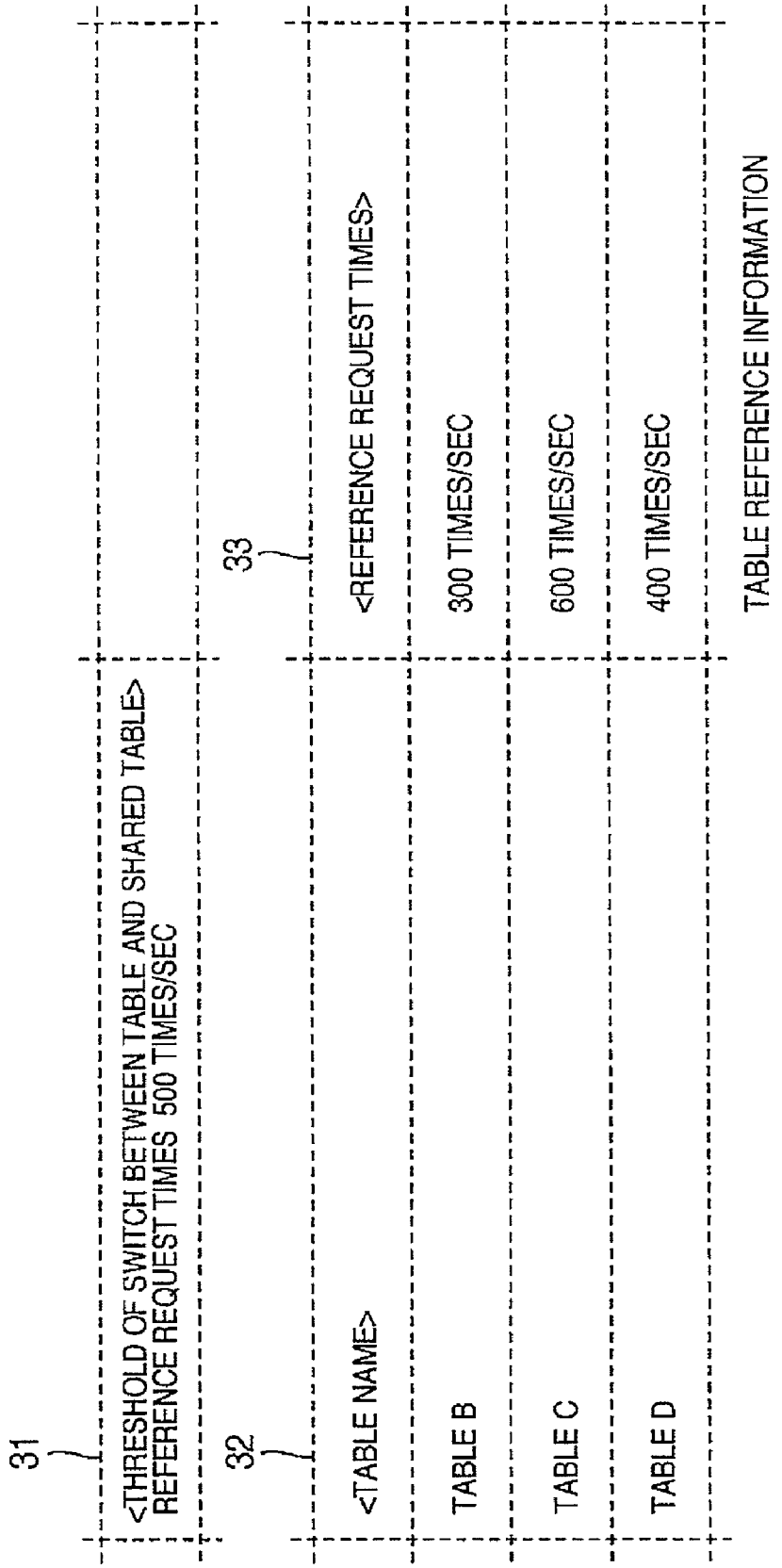

2a
OPERATING NODE
5a
MAIN STORAGE UNIT
1
APPLICATION PROGRAM
DATA ACCESS REQUEST CONTROL
6
12a
TABLE INFORMATION CONTROL
11
TABLE INFORMATION
29
TABLE REFERENCE MONITORING AND SHARED TABLE SWITCHING CONTROL
30
TABLE REFERENCE INFORMATION
4b
DATABASE MANAGEMENT NODE MAIN STORAGE UNIT
5b
12b
TABLE INFORMATION CONTROL
DATA ACCESS CONTROL
7b
11
TABLE INFORMATION
4c
DATABASE MANAGEMENT NODE MAIN STORAGE UNIT
5c
12c
TABLE INFORMATION CONTROL
DATA ACCESS CONTROL
7c
11
TABLE INFORMATION
.....

FIG.10


31
<THRESHOLD OF SWITCH BETWEEN TABLE AND SHARED TABLE>
REFERENCE REQUEST TIMES 500 TIMES/SEC
32
33
<TABLE NAME>
<REFERENCE REQUEST TIMES>
TABLE B
300 TIMES/SEC
TABLE C
600 TIMES/SEC
TABLE D
400 TIMES/SEC
TABLE REFERENCE INFORMATION

# DATA PROCESSING SYSTEM AND METHOD WITH DATA SHARING FOR THE SAME

## CROSS-REFERENCE TO RELATED APPLICATION

[0001] This application is a continuation of U.S. application Ser. No. 10/930,765, filed Sep. 1, 2004, which application claims priority from Japanese application JP2004-157619 filed on May 27, 2004, the content of which is hereby incorporated by reference into this application.

## BACKGROUND OF THE INVENTION

[0002] The present invention relates to a data processing system and method which are arranged to keep loads at data processing nodes well-balanced as much as possible.

[0003] As a method of accessing data in a database system composed of plural nodes, conventionally, two methods, that is, a shared database approach and a shared nothing database approach are disclosed in Sohan DeMel, "Oracle9i Real Application Clusters—Cache Fusion Delivers Scalability," Oracle White Paper, Oracle Corporation, pages 7 to 9 and 12 to 13, May 2001.

[0004] The shared database approach method is arranged to allow each of the nodes to evenly make access to all data. Hence, this method makes it possible to uniformly distribute load among the nodes. This method, however, requires synchronous update among the nodes, which brings about overheads such as an I/O process of reading data again and an inter-node communication for synchronous control.

[0005] On the other hand, the shared nothing database approach method permits each of the nodes to make access to only a data area allocated to each node itself. Hence, if the data location is shifted, the load is focused on one or some nodes. However, this method does not require the synchronous control among the nodes, so that the nodes may be operated in parallel.

## SUMMARY OF THE INVENTION

[0006] Both of the foregoing conventional methods allow the load burdened to the nodes to be distributed by increasing the nodes in number, for the purpose of improving the performance. However, the shared database approach method involves a disadvantage that the performance of the net increased nodes cannot be improved because the overhead is increased with the increase of the nodes. Further, the shared nothing database approach method also requires proper relocation of data according to the number of the increased nodes. In actual, however, if the table with a high reference frequency cannot be relocated by the division or the like, the load is concentrated on the node where that table with a high reference frequency is located. It means that it is impossible to improve the performance according to the number of the increased nodes. This is the disadvantage of the latter method.

[0007] The foregoing problem involved by the shared database approach method may be solved by the following method. That is, if two or more nodes make access to the same data, the transfer of the data from a cache of the node to be synchronized to a cache of another node results in eliminating the necessity of the quite costly I/O process, thereby suppressing the increase of the overhead.

[0008] However, up to date, the foregoing problem involved by the shared nothing database approach method cannot be solved by any effective means.

[0009] It is therefore an object of the present invention to keep the loads at plural database processing nodes well-balanced as much as possible.

[0010] According to an aspect of the present invention, in carrying out the object, a method for managing a database arranged so that each of the database management nodes may make access to only a table in a data area allocated to the accessing node itself, characterized in that a table with a high reference frequency, selected from the tables of the data areas allocated to those nodes, may be specified as a shared table to be referenced by those database management nodes.

[0011] The present invention is effective in keeping the load at the database management nodes well-balanced as much as possible.

## BRIEF DESCRIPTION OF THE DRAWINGS

[0012] FIG. **1** is a block diagram showing an arrangement of a database system according to a first embodiment of the present invention;

[0013] FIG. **2** is a view showing a composition of table information;

[0014] FIG. **3** is a flowchart for illustrating a processing operation of an operating or application or control node included in the first embodiment of the present invention;

[0015] FIG. **4** is a flowchart for illustrating a processing operation of a database management node included in the first embodiment of the present invention;

[0016] FIG. **5** is an explanatory view showing an executing process of the application included in the first embodiment of the present invention;

[0017] FIG. **6** is a block diagram showing an arrangement of a database system according to a second embodiment of the present invention;

[0018] FIG. **7** is a flowchart for illustrating a processing operation of an operating node included in the second embodiment of the present invention;

[0019] FIG. **8** is a flowchart for illustrating a processing operation of a database management node included in the second embodiment of the present invention;

[0020] FIG. **9** is a block diagram for explaining a third embodiment of the present invention which is arranged to collect a reference frequency of each table and then to dynamically switch the table into a shared table or vice versa; and

[0021] FIG. **10** is a view showing an example of table reference information.

## DESCRIPTION OF THE EMBODIMENTS

[0022] Hereafter, the embodiments of the database management method and system according to the present invention will be described in detail with reference to the appended drawings.

[0023] FIG. **1** is a block diagram showing an exemplary arrangement of a database system according to a first embodiment of the present invention. In FIG. **1**, a reference number **1** denotes an application program (including a business program and an object, a process or a thread realizing the function of the program). A reference number **2***a* denotes an operating or application or control node. Reference numbers **3***b* to **3***d* denote tables. Reference numbers **4***b* to **4***d* denote database management nodes. Reference numbers **5***a* to **5***d* denote main storage units. A reference number **6** denotes a data access request control unit (that may be realized by a program, or an object, a process or a thread for realizing the function of the program, or hardware). Reference numbers **7***b* to **7***d* denote data access control units (that may be realized by a program, or an object, a process or a thread for realizing the function of the program, or hardware), Reference numbers **8***b* to **8***d* denote cache memories (simply referred to as caches). A reference number **9** denotes an external storage unit. Reference numbers **10***b* to **10***d* denote data areas (that may be referred also to as a data saving area or a data storage area). A reference number **11** denotes table information. Reference numbers **12***a* to **12***d* denote table information control units (that may be realized by a program, or an object, a process or a thread for realizing the function of the program, or hardware). In the database system to be described as the first embodiment of the present invention, in principle, each of the database management nodes may make access to a data area (for storing a database) allocated to the accessing node itself but may not make access to any data area allocated to any other node. In general, it is a database management system called a shared nothing database approach.

[0024] The database system according to the first embodiment of the present invention includes the operating node **2***a* that executes the application program **1**, a plurality of data areas **10***b* to **10***d*, the external storage unit **9** in which the tables **3***b* to **3***d* composing the database are stored in the data areas respectively, and a plurality of database management nodes **5***b* to **4***d* that make access to the tables **3***b* to **3***d* by executing the application program **1**. In this embodiment, the divisional parts of the database are called the tables. They may be referred to as a partial table, a partial database, a divisional table, a divisional database, or the like. Moreover, the database may be simply called a table. The operating node **2***a* includes the application program **1** in the main storage unit **5***a* inside the node **2***a* itself and the data access request control unit **6** that accepts a request from the application program **1** and distributes a data access request into a proper one of the database management nodes **4***b* to **4***d*. In this embodiment, the operating node **2***a* or the database management nodes **4***b* to **4***d* may be realized by an information processing apparatus, a logical server, a program, an object, a process, a thread, or a hardware unit. Further, the database management nodes **4***b* to **4***d* may be realized by a database management program or a database management apparatus. The data access control units **7***b* to **7***d* may be realized by a program, an object, a process, a thread, or a hardware unit.

[0025] The database management nodes **4***b* to **4***d* include data access control units **7***b* to **7***d*, each of which accepts the request from the data access request control unit **6** of the operating node **2***a* on the corresponding one of the main storage units **5***b* to **5***d* and makes access to the corresponding one of the tables **3***b* to **3***d* allocated to the node **2***a* itself. The data access control units **7***b* to **7***d* make access to the tables **3***b* to **3***d* through the caches **8***b* to **8***d* and then hold the read data on the cache **8***b* to **8***d*, respectively. The data areas **10***b* to **10***d* properly partitioned on the external storage unit **9** correspond with the database management nodes **4***b* to **4***d*, respectively. These data areas **10***b* to **10***d* may be referred by all the database management nodes **4***b* to **4***d*. On each of the data areas **10***b* to **10***d* is created the corresponding one of the tables **3***b* to **3***d*, all of which compose the database. The tables **3***b* to **3***d* are permitted to be accessed by only the database management nodes **4***b* to **4***d* to which the created data areas **10***b* to **10** are allocated respectively.

[0026] In the first embodiment of the present invention, the data area allocated to any database management node may be specified as a shared data area that permits reference from any other database management node. However, only the database management node to which the data area is allocated is permitted to update the data. In the first embodiment of the invention as shown in FIG. **1**, the data area **10***c* allocated to the database is specified as the shared data area, which may be referred by the database management node **4***b*. The data area **10***c* specified as the shared data area is a data area in which the table with a high reference frequency is created. The table is specified as the shared table. In the embodiment shown in FIG. **1**, the table **3***c* is specified as the shared table. Then, on each of the main storage units **5***a* to **5***d* of all the nodes is created table information that includes the information as to whether or not each table is the shared table, the database management node allocated thereto, and the reference-permitted database management node(s). In the main storage units **5***a* to **5***d*, the table information control units **12***a* to **12***d* are provided for accessing the table information.

[0027] In the foregoing first embodiment of the present invention, the operating node **2***a* and the database management nodes **4***b* to **4***d* may be the ordinary information processing apparatus such as well-known workstations or PCs each provided with a CPU (not shown). Then, the execution of the program through the CPU makes it possible to compose various kinds of function units (to be discussed below) required by this embodiment of the invention. Further, the operating node **2***a* and the database management nodes **4***b* to **4***d* or the database management nodes **4***b* to **4***d* may be totally assumed as the virtual computer and built in the computer system composed by one hardware unit. This holds true to the other embodiments of the invention to be discussed below.

[0028] FIG. **2** shows an exemplary composition of the table information **11**. The table information **11** includes a table name **13** positioned as a key, an information item **14** as to whether or not the table is the shared table, an information item **15** of the allocated database management node, and an information item **16** as to the database management node(s) to be referenced.

[0029] FIG. **3** is a flowchart for illustrating the processing operation of the operating node in the first embodiment of the present invention. This processing operation will be described below.

[0030] (1) In the operating node **2**, when the application program **1** requests a database access from the data access request control unit **6**, the data access request control unit **6** performs a data access request control process, requests the

table **11** of the target table from the table information control unit **12***a*, and obtains the table information **11** (step S**1**).

[**0031**] (2) The data access request control unit **6** determines whether or not the target table is the shared table based on the obtained table information **11** and, if not, issues an access request to the data access control unit of the allocated database management node (steps S**4** and S**5**).

[**0032**] (3) If the target table is the shared table in the determination of the step S**4**, the data access request control unit **6** determines if the type of the process to be requested is a reference or an update. If it is the request for reference, the data access request control unit **6** selects any database management node matched to the condition of the request for reference from the database management nodes to be referenced, for the case shown in FIG. **1**, the database management node **4***b* or **4***c*, and issues a request of an access for reference to the data access control unit of the selected database management node (steps S**6** to S**8**).

[**0033**] (4) If the type of request is a request for update in the determination of the step S**6**, the data access request control unit **6** loops all the database management nodes to be referenced, selects the database management node having issued no access request, and then issues a request of an access for update to the data access control unit **7** of the selected database management node (steps S**9** to S**11**).

[**0034**] FIG. **4** is a flowchart for illustrating the processing operation of the database management node included in the first embodiment of the present invention. Then, this processing operation will be described below. The process to be described is a process having been described with reference to FIG. **3**, that is, a process of the database management node having accepted the access request.

[**0035**] (1) The data access control unit of the database management node that has been requested to be accessed performs a data access control process. At first, it is determined if the type of the requested process is a reference or an update (step S**12**).

[**0036**] (2) If the type of the requested process is a request for reference in the determination of the step S**12**, the data access request control unit determines whether or not the data requested for the cache of the node to which the unit belongs has been already read in the cache. If the requested data has been already read in the cache, the data access request control unit gives back the result based on the read data to the requester (steps S**13** and S**15**).

[**0037**] (3) If the data requested for the cache has not been read in the cache in the determination of the step S**13**, the data access request control unit reads the data requested for the cache from the target table or the shared table and saves the data in the cache. Then, the control unit gives back the result based on the saved data to the requester (steps S**14** and S**15**).

[**0038**] (4) If the type of the requested process is a request for update in the determination of the step S**12**, the data access request control unit determines if the data requested to be updated has been already read in the cache. If the data requested to be updated has been already read in the cache, the requested update is reflected on the data saved in the cache (steps S**16** and S**18**).

[**0039**] (5) If the data requested to be updated has not been read in the cache in the determination of the step S**16**, the data access request control unit reads the data requested to be updated from the target table or the shared table and saves the data in the cache. The requested update is reflected on the data saved in the cache (steps S**17** and S**18**).

[**0040**] (6) In succession, the data access request control unit requests the table information **11** of the target table from the table information control unit **12** and then obtains the table information **11** (step S**19**).

[**0041**] (7) The data access control unit determines whether or not the target table is the shared table based on the obtained table information. If it is not the shared table, the data updated in the cache is outputted to the table of the data area (steps S**22** and S**23**).

[**0042**] (8) If the target table is the shared table in the determination of the step S**22**, the data access request control unit determines whether or not the database management node to which the control unit belongs is the database management node to which the shared data area is allocated (step S**24**).

[**0043**] (9) If it is the database management node to which the shared data area is allocated in the determination of the step S**24**, the data updated in the cache is outputted to the shared data area. If it is not, nothing is executed (steps S**23** and S**25**).

[**0044**] According to the first embodiment of the present invention, by executing the foregoing process, in the case of requesting reference to the shared table, the operating node enables to distribute the request into the data access control unit of any database management node in the processes of the steps S**7** and S**8**. If the shared table is a table to be frequently referenced, this process allows the process of reference to be executed from any database management node permitted to refer to the shared table. This prevents the process from being concentrated on the specific database management node.

[**0045**] In a case that the shared table is required to be updated, by issuing an update to the data access control unit of every database management node permitted to refer to the shared table through the processes of the steps S**9**, S**10** and S**11**, the operation is executed to update the data stored in the shared table of the shared data area and the data stored in the cache of every database management node permitted to refer to the shared table.

[**0046**] For the database of the shared database approach, only the cache of the node having executed the update is updated. Hence, if the reference is given from another node, an I/O process takes place for reading the data from the disk again. In this case, in order to guarantee no node being updated for the purpose of preventing the read of the data being updated, the inter-node communication for synchronous control takes place. That is, the database of the shared database approach brings about various kinds of overheads when the plural nodes make access to the same table.

[**0047**] As will be understood from the above description, the database management method and system according to the first embodiment of the invention are arranged to constantly reflect all the update requests onto the data of the cache of every database management node permitted to refer

to the shared table, eliminate the necessity of reading the data from the disk again, and prevent occurrence of any I/O process. Moreover, for preventing the read of the data being updated, the method and system are merely required to guarantee that the database management node required to be referenced is not being updated. This guarantee thus does not depend on the state of any other database management node. It means that no inter-node communication for synchronous control takes place.

[0048] As such, the database management method and system according to the first embodiment of the invention allow plural database management nodes to make access to the shared table without any overhead. This makes it possible for the database management nodes to refer to and update the data in parallel only if the new database management node(s) are added to the reference-permitted node(s) for improving the performance. It results in no increase of overhead, thereby improving the corresponding performance with the increased number of the database management nodes.

[0049] FIG. **5** is a view for explaining the execution of the application program (simply referred to as the application) in the aforementioned first embodiment of the present invention. In turn, the concrete process having been described with reference to FIGS. **1** to **4** will be described with reference to FIG. **5**.

[0050] It is assumed that an application a**17**, an application b**18**, an application c**19**, an application d**20**, and an application e**21** are executed in the operating node **2***a*. The applications a**17**, b**18**, d**20** and e**21** are executed to refer to the shared table c**23** allocated to the database management node c**22**. The application c**19** is executed to update the data of the shared table c**23** into "**20**". The shared table c**23** permits the database management nodes b**24** and c**22** to refer to the table c**23** itself. Before executing the foregoing applications, the data "**10**" is entered into the shared table c**23** but is not still read into the cache memories b**25** and c**26** provided in the database management nodes b**24** and c**22** respectively.

[0051] At first, it is assumed that the applications a**17** and b**18** are executed in succession so that the request for reference given by the applicationa **17** is distributed into the database management node b**24** and the request for reference given by the application b**18** is distributed into the database management node. In this case, the request for reference given by the application a**17** causes the data "**10**" to be read into the cache b**25** of the database management node b**24** and the result to be given back to the application a**17** itself. In parallel to this process, the request for reference given by the application b**18** causes the data "**10**" to be read into the cache c**26** of the database management node c**22** and the result to be given back to the application b**18**. Afterwards, if the application for referring to the shared table c**22** is executed, the data "10" stored in the cache of any distributed database management node is given back to that node itself.

[0052] Afterwards, the application cl**9** is executed to issue a request for changing the data of the shared table c**23** from "10" into "20", when the operating node **2***a* issues a request for update to both of the database management nodes b**24** and c**22**. The database management node b**24** updates the data stored in the cache b**25** from "**10**" to "**20**". Further, the database management node c**22** updates the data stored in the cache c**26** from "**10**" into "**20**" and then outputs the updated data into the shared table c**23**. As described above, the data stored in the cache of every database management node is updated.

[0053] Then, it is assumed that the applications d**20** and e**21** are executed in succession, the request for reference given by the application d**20** is distributed into the database management node b**24**, and the request for reference given by the application e**21** is distributed into the database management node c**22**. In this case, the data "**20**" stored in the cache b**25** of the database management node b**24** is given back to the request for reference given by the application d**20**. In parallel to this process, the data "**20**" stored in the cache c**26** of the database management node c**22** is given back to the request for reference given by the application e**21**. If the application for referring to the shared table c**23** is executed subsequently to the above, the data "**20**" stored in the cache of any distributed database management node is given back to that application.

[0054] As has been understood from the foregoing description, according to the first embodiment of the invention, if the application for updating the shared table is executed while the application for referring to the shared table c**23** is executed repetitively, the operation is executed to update the data stored in the cache of every database management node during the execution of each application. Hence, even if the application having issued the request for reference is distributed into any database management node, the proper data may be given back to that application without having to execute the synchronous control with the update at another database management node and read the data again.

[0055] According to the foregoing first embodiment of the present invention, the data management system and method are arranged to store the tables **3***b* to **3***d* in the data areas **10***b* to **10***d* located on the external storage unit **9** respectively and read the data in the caches **8***b* to **8***d* of the database management nodes **4***b* to **4***d* respectively. The database system of the first embodiment arranged as described above makes it possible for plural database management nodes to refer to the shared table **3***c*. Then, the read data is saved in the cache of every data management node permitted to refer to the shared table **3***c*.

[0056] In the first embodiment of the present invention, therefore, in the case of updating the data for synchronous control between the database management nodes, it is necessary to update the data stored in the cache of every database management node permitted to refer to the shared table **3***c* to be updated through the processes of the steps S**9** to S**11** shown in FIG. **3** and of the steps S**16** to S**18** shown in FIG. **4**. Further, also in the first embodiment, it is necessary to control the database management node **4** for actually updating the shared table **3***c* located on the external storage unit **9** through the processes of the steps S**19** to S**25** shown in FIG. **4**.

[0057] FIG. **6** is a block diagram showing an exemplary arrangement of the database system according to the second embodiment of the present invention. In FIG. **6**, reference numbers **27***b* to **27***d* denote caches of an external storage unit. A reference number **28** denotes a cache control unit of the external storage unit. The other reference numbers

denote the same components as those shown in FIG. **1**. The database system according to the below-described second embodiment of the present invention is arranged to eliminate the necessity of the synchronous control between the database management nodes in the foregoing first embodiment of the present invention.

[**0058**] The second embodiment shown in FIG. **6** is different from the first embodiment in location of the caches **27***b* to **27***d* in the external storage unit **9** in place of the cache located in each database management unit as in the first embodiment and in provision of new table information **35** having a table name **35** and a database management node name **37** for managing the correspondence between the table. The other arrangement of the second embodiment is the same as that of the first embodiment.

[**0059**] The external storage unit cache memories **27***b* to **27***d* (simply referred to as the external caches) located in the external storage unit are allocated to the data areas **10***b* to **10***d* respectively. The external cache control unit **28** manages the allocating correspondence between the external cache memories **27***b* to **27***d* and the data areas **10***b* to **10***d* and refers to or updates the data of the proper external caches **27***b* to **27***d* to the data areas **10***b* to **10***d* requested to be referenced or updated by the database management nodes **4***b* to **4***d*. Further, on the predetermined occasion, the external cache control unit **28** reflects the data updated by the external cache memories **27***b* to **27***d* onto the data areas **10***b* to **10***d* allocated to those caches **27***b* to **27** respectively. The predetermined occasion indicates the times such as when the updated data of each of the external cache memories **27***b* to **27***d* reaches the predetermined amount, when the updating times reach the given times, the predetermined interval of time, or when a request from the database management nodes **4***b* to **4***d* or a command from the external to the external storage unit **9** is inputted. When the database management nodes **4***b* to **4***d* refer to the tables **3***b* to **3***d*, the external cache control unit **28** operates to read the data into the external caches **27***b* to **27***d* allocated to the data areas **10***b* to **10***d* where the tables **3***b* to **3***d* are stored and then stores the data therein. Then, the external cache control unit **28** causes the external caches **27***b* to **27***d* to refer to the database management nodes **4***b* to **34***d*.

[**0060**] In this case, the external cache control unit **28** obtains the allocating relation between the database management node and the table stored in the external storage unit **10** by referring to the table information **35**. In this embodiment, the table **3***d* is allocated to the database management node **4***d*. The table **3***c* is allocated to the database management nodes **4***b* and **4***c*. That is, when the external cache control unit **28** receives an access request from the database management node **4***d*, the external cache control unit **28** refers to the table information **35** and executes the process based on the access request received about the table **3***d* corresponding with the database management node **4***d*. Further, when the database management nodes **4***b* to **4***d* updates the tables **3***b* to **3***d*, the external cache control unit **28** updates the data held in the external cache memories **27***b* to **27***d* allocated to the data areas **10***b* to **10***d* for storing the tables **3***b* to **3***d*. Hence, any one of the database management nodes **4***b* to **4***d* may refer to and update the data stored in the external cache memories **27***b* to **27***d* if requested. Further, on the predetermined occasion, the external cache control unit **28** updates the data areas **10***b* to **10***d* allocated to the external cache memories **27***b* to **27***d* with the data updated in the external cache memories **27***b* to **27***d*. The database management nodes **4***b* to **4***d* thus do not require control of an output of the updated data to the data areas **10***b* to **10***d*.

[**0061**] In the foregoing second embodiment of the present invention, the database management nodes **4***b* to **4***d* may directly refer to and update the data stored in the external cache memories **27***b* to **27***d* through the external cache control unit **28**. Hence, the database management nodes **4***b* to **4***d* does not need to hold the data in the cache of each node. This eliminates the necessity of updating the data stored in the cache memories **8***b* to **8***d* of all the database management nodes **4***b* to **4***d* permitted to refer to the shared table **3***c* to be updated, the update having been executed for synchronous control among the database management nodes **4***b* to **4***d*. Further, it is not necessary to control the database management node **4***c* for actually updating the shared table **3***c* located in the external storage unit **9**. The second embodiment of the present invention does not need the foregoing control, which makes it possible to make access to the table more quickly.

[**0062**] FIG. **7** is a flowchart for illustrating the operating process of the operating node included in the second embodiment of the present invention. Then, the description will be oriented to the operating process of this node.

[**0063**] In the flow shown in FIG. **7**, the process from the steps S**1** to S**8** is identical with the process from the steps S**1** to S**8** shown in FIG. **3**. In the second embodiment of the present invention, it is not necessary to execute the process of updating the data stored in the cache of every database management node permitted to refer to the shared table **3***c* to be updated, which process has been executed in the process from the steps S**9** to S**11** having been described in the flow of FIG. **3**. Hence, in the case of requesting the update for the shared table **3***c*, the operation is executed to select any database management node for the update process (step S**26**) and request the data access control unit of the selected database management node to access the shared table **3***c* (step S**27**).

[**0064**] FIG. **8** is a flowchart for illustrating the operating process of the database management node included in the second embodiment of the present invention. Then, the description will be oriented to the operating process. The process described herein concerns with the process having been described with reference to FIG. **7**, concretely, the process of the database management node having accepted an access request.

[**0065**] In the flow of FIG. **8**, the process of the steps S**12** to S**18** is identical with the process of the steps S**12** to S**18** having been described along the flow of FIG. **4**. In the second embodiment of the invention, the system does not need the process of controlling the database management node **4***c* for actually updating the shared table **3***c* located on the external storage unit **9**. In the flow of FIG. **4**, the process having been executed in the steps S**19** to S**25**. Hence, after the process of the step **18** is executed to update the data stored in the external cache **27***c*, the external cache control unit **28** updates the data area **10***c*. This means that the database management node has nothing to do (step S**28**).

[**0066**] As described above, according to the second embodiment of the present invention, the system provides a

capability of allocating the cache onto each partitioned data area located on the external storage unit and associating the allocated cache with the external storage unit that can be directly controlled by the database management node. This capability makes it possible to execute the application more fast.

[0067] As described above, in the first and the second embodiments of the present invention, the load may be distributed by specifying the table with a high reference frequency as the shared table. However, the system is required to search the reference frequency of the table or study the change of the reference frequency as predicating the change of the using status of the table. Further, the system is required to determine if the table is specified as the shared table on the promotional level.

[0068] Hence, for the purpose of reducing the burden in the search and study of determining if the table is specified as the shared table, the system may have a capability of monitoring the referring status of the table in the execution of the application and dynamically determining if the table is specified as the shared table.

[0069] FIG. **9** is an explanatory view showing the third embodiment of the present invention which makes it possible to collect a reference frequency of each table and dynamically determine if the table is specified as the shared table according to the reference frequency of the table. Then, the description will be oriented to the third embodiment of the present invention. In FIG. **9**, a reference number **29** denotes a table reference monitoring and shared table switching control unit. A reference number **30** denotes table reference information. The other reference numbers indicate the same components as those of FIG. **1**. FIG. **9** shows only the portion required for the description herein. The overall arrangement is the same as that shown in FIG. **1** or **6**.

[0070] In the third embodiment of the present invention shown in FIG. **9**, the data management method and system are arranged to add to the operating node **2***a* the table reference monitoring and shared table switching control unit **29** (simply referred to as the monitoring and switching unit) for monitoring the access to the tables **3***b* to **3***d* (FIG. **1** or **6**) to be executed by the application **1** of the operating node **2***a*, determining if the table is specified as the shared table, and switching the table into the shared table if specified. When the application **1** requires the access to the tables **3***b* to **3***d*, the data access request control unit **6** inevitably issues the request for referring to the table information **11** to the table information control unit **12***a*. In response, the data access request control unit **6** notifies the monitoring and switching unit **29** of the request for reference. Then, the monitoring and switching unit **29** saves each reference frequency of the tables **3***b* to **3***d* in the table reference information **30**.

[0071] FIG. **10** shows an exemplary composition of the table reference information **30**. The table reference information **30** includes a reference frequency threshold value **31** on which the table is switched into the shared table, prespecified by the system or the user, and the reference frequencies **33** of the tables **3***b* to **3***d* with the tale name **32** as a key.

[0072] The monitoring and switching unit **29** calculates each reference frequency of the tables **3***b* to **3***d* based on the notice sent from the data access request control unit **6** and occasionally updates the reference frequency information **33** of the tables **3***b* to **3***d* of the table reference information **30**. Further, the monitoring and switching unit **29** constantly monitors this value. If this value exceeds the threshold value **31** on which each of the tables **3***b* to **3***d* is switched into the shared table, the monitoring and switching unit **29** notifies the table information control unit **12***a* of every node so that the table with the greater value than the threshold value **31** is switched into the shared table, while if the shared table lowers the value, the shared table is switched into the ordinary table. The table information control unit **12***a* updates the information **14** (FIG. **2**) as to whether or not the concerned table name is the shared table in the table information **11** and the reference-permitted database management node(s) **16** (FIG. **2**).

[0073] In a case that the switch of the table into the shared table is notified to the table information control unit **12***a* (FIG. **1**), the table information control unit **12***a* operates to specify the information **14**FIG. **2**) as to whether or not the concerned table name **13** (FIG. **2**) is given to the shared table as YES and then to add to the reference-permitted database management nodes **16** (FIG. **2**) the database management node permitted to refer to the shared table. Further, in a case that the switch of the shared table into the ordinary table is notified to the table information control unit **12**, the table information control unit **12***a* operates to specify the information **14** as to whether or not the concerned table name **13** is given to the shared table as NO and then update the reference-permitted database management nodes **16** to only the allocated database management node(s). Later, the access request from the application **1** is made to be the switched table information **11** (FIG. **1**) given back from the table information control unit **12***a* when the data access request control unit **6** determines if the table is the shared table. The table information control **12***a* is operated according to the switched table information **11**.

[0074] As described above, according to the third embodiment of the present invention, the data management method and system are arranged to dynamically switch the ordinary table into the shared one or vice versa according to the reference frequency of the table. If the reference frequency of the table is increased, the load is distributed with the table as the shared table. On the other hand, if the reference frequency of the shared table is decreased, the shared table is switched into the ordinary table. This operation makes it possible to eliminate the user's necessity of relocating the table(s) and studying if the table is switched into the shared table according to the using status of the table.

[0075] Each process included in each foregoing embodiment of the present invention may be composed of a processing program, which may be supplied as being saved in a recording medium such as a HD (harddisk), a DAT (digital audio tape), a FD (floppy disk), a MO (magneto-optical) disk, a DVD-ROM (digital versatile disk-read-only memory), or a CD-ROM (compact disk-read-only memory).

[0076] As described above, the first embodiment of the present invention concerns with the database system in which each of the database management nodes is permitted to make access to only the data area allocated to each node. If the relocation of the table with a high reference frequency is not made possible by division or the like, the table may be

specified as the shared table. Hence, any node may make access to the shared table on any timing, so that the performance of the database system may be improved according to the number of the increased nodes.

[0077] Moreover, according to the second embodiment of the present invention, the database management method and system are arranged to allocate the cache onto each partitioned data area located on the external storage unit so that the corresponding node may directly control the cache. This thus makes it possible to realize a far faster access to the database.

[0078] Moreover, according to the third embodiment of the present invention, the database management method and system are arranged to monitor the referring status of the table by the application and dynamically switch the table into the shared table or vice versa according to the reference frequency of the table. This thus makes it possible to lessen the user's burden in studying the tables for determining if one table is switched to the shared table.

[0079] It should be further understood by those skilled in the art that although the foregoing description has been made on embodiments of the invention, the invention is not limited thereto and various changes and modifications may be made without departing from the spirit of the invention and the scope of the appended claims.

What we claim is:

1. A database management method for management of a plurality of database management nodes, comprising the steps of:

making an access to a table having a data area allocated to each database management node but not to a table having a data area not allocated to each database management node in a shared nothing system in which each of said plurality of database management nodes makes access to a table stored in a data area allocated to each database management node;

calculating access frequencies of said table stored in said data area allocated to each of said plural database management nodes;

determining if one of said access frequencies is greater than or equal to a predetermined value; and

if one of said access frequencies is more than or equal to said predetermined value, re-allocating said table with the access frequency greater than or equal to the predetermined value as a shared table to be referenced by one or more database management nodes to which said table with the access frequency is not allocated, for distributing load of said disk to said plural database management nodes.

2. The database management method as claimed in claim 1, wherein said shared table is updated by the database management node to which said shared table is allocated, and when said shared table is updated, each of all the database management nodes having a capability of referring to said shared table operates to update data stored in a cache located in each node itself.

3. The database management method as claimed in claim 1, wherein one or more tables and shared tables in data areas allocated to said database management nodes respectively are provided in each of said partitioned data areas located on an external storage unit, a cache is allocated to said external storage unit in a manner to correspond with each of said tables and shared tables, and said cache is controlled by the corresponding database management node.

4. The database management method as claimed in claim 1, further comprising steps of monitoring a referring status of each of said tables and shared tables located in said data areas to which said database management nodes are allocated respectively and dynamically switching said table into said shared table or vice versa according to the referring frequency of said table.

5. A database management system comprising:

a plurality of database management nodes; and

an external storage unit having data areas each having a table;

wherein a database management node makes an access to a table having a data area allocated to each database management node but not to a table having a data area not allocated to each database management node in a shared nothing system in which each of said plurality of database management nodes makes access to a table stored in a data area allocated to each database management node;

wherein said database management node calculates access frequencies of said table stored in said data area allocated to each of said plural database management nodes;

wherein said database management node determines if one of said access frequencies is greater than or equal to a predetermined value; and

wherein said database management node, if it is determined that one of said access frequencies is greater than or equal to a predetermined value, re-allocates said table with the access frequency greater than or equal to the predetermined value as a shared table to be referenced by one or more database management nodes to which said table with the access frequency is not allocated, for distributing load of said disk to said plural database management nodes.

6. The database management system as claimed in claim 5, wherein said shared table is updated by the database management node to which said shared table is allocated, and when said shared table is updated, each of all the database management nodes having a capability of referring to said shared table operates to update data stored in a cache located in each node itself.

7. The database management system as claimed in claim 5, wherein one or more tables and shared tables in data areas allocated to said database management nodes respectively are provided in each of said partitioned data areas located on an external storage unit, a cache is allocated to said external storage unit in a manner to correspond with each of said tables and shared tables, and said cache is controlled by the corresponding database management node.

8. The database management system as claimed in claim 5,

wherein said database management node monitors a referring status of each of said tables and shared tables located in said data areas to which said database management nodes are allocated respectively and

dynamically switching said table into said shared table or vice versa according to the referring frequency of said table.

**9**. A database management program for management of a plurality of database management nodes having computer readable codes arranged to function on a database management system, said management program comprising the steps of:

making an access to a table having a data area allocated to each database management node but not to a table having a data area not allocated each database management node in a shared nothing system in which each of said plurality of database management nodes makes access to a table stored in a data area allocated to each database management node;

calculating access frequencies of said table stored in said data area allocated to each of said plural database management nodes;

determining if one of said access frequencies is greater than or equal to a predetermined value; and

if one of said access frequencies is more than or equal to said predetermined value, re-allocating said table with the access frequency greater than or equal to the predetermined value as a shared table to be referenced by one or more database management nodes to which said table with the access frequency is not allocated, for distributing load of said disk to said plural database management nodes.

**10**. The database management program as claimed in claim 9, wherein said shared table is updated by the database management node to which said shared table is allocated, and when said shared table is updated, each of all the database management nodes having a capability of referring to said shared table operates to update data stored in a cache located in each node itself.

**11**. The database management program as claimed in claim 9, wherein one or more tables and shared tables in data areas allocated to said database management nodes respectively are provided in each of said partitioned data areas located on an external storage unit, a cache is allocated to said external storage unit in a manner to correspond with each of said tables and shared tables, and said cache is controlled by the corresponding database management node.

**12**. The database management program as claimed in claim 9, further comprising steps of monitoring a referring status of each of said tables and shared tables located in said data areas to which said database management nodes are allocated respectively and dynamically switching said table into said shared table or vice versa according to the referring frequency of said table.

* * * * *