access (or locking) to the records or to the portions of the B-Tree indexes to insure that the index node, or record is not changed while the user is attempting to access it. Locking is addressed in "Index Locking and Splitting" IBM Technical Disclosure Bulletin, volume 25, number 7B, December, 1982, pages 3725-3729; and Locking Protocols for Concurrent Operations on B-Trees", IBM Technical Disclosure Bulletin, volume 19, number 10, March, 1977, pages 3887-3889.

The disadvantages to a locking solution is that a lock, while providing access to one user, prevents access by any other user.

Information in index nodes may be altered during access by several users. These nodes contain the key record information that may be deleted or inserted as records are inserted into or deleted from the database table. Information from one node may be moved to another node during a record insert operation because of limited storage space for a node. The inserting transaction may attempt to move key record information from one node to another in order to accomplish the insertion task. This movement of information during concurrent transaction access may generate error conditions within the concurrent transaction accesses because the index information is being changed.

SUMMARY OF INVENTION

It is an object of the invention to ameliorate one or more disadvantages of the prior art.

According to one aspect of the invention there is disclosed in a data processing system, a method executed by a data processor for fetching key record data stored in a data processing memory in a group of record keys by using a portion of a key record through an index tree during a transaction in the data processing system wherein other transactions may concurrently access said record keys through said index tree, said index tree having at least a root node, at least one intermediate level of at least one node, each node having a key record reference to one or more nodes in a next lower ordered level of nodes and having bottom nodes that provide access to said key record data, said method comprising the steps of: (a) traversing across said nodes in said data processing memory

from said root node through said intermediate level of nodes by accessing a node using said key record portion of the root node and traversing to a next node indicated by said key record portion of the accessed node until a bottom node is reached; (b) limiting all but read accesses to the node being accessed and an immediate previously accessed node in said data processing memory, to other concurrent transactions in said data processing system; (c) identifying and accessing said key record in said bottom node; (d) limiting all but read accesses to said key record; (e) removing all access limitations to the accessed nodes including the bottom node except for the accessed key record; (f) fetching key record data; and (g) removing the access limitation to the key record after the record data has been fetched.

According to another aspect of the invention there is disclosed in a data processing system, a method executed by a data processor for inserting a single key record in a group of record keys stored in a data processing memory according to a key record through an index tree during a transaction in the data processing system wherein other transactions concurrently access said record keys through said index tree, said index tree having at least a root node, at least one intermediate level of at least one node, each node having a key record reference to one or more nodes in a next successive level of nodes and having bottom nodes that provide access to said record keys, said method comprising the steps of: (a) limiting all accesses to the key record being inserted; (b) traversing across said nodes in said data processing memory from said root node through said intermediate level of nodes by accessing a node using said key record of said root node and traversing to a next node indicated by said key record of said accessed node until a bottom node is reached; (c) limiting all but read accesses to the node being accessed and not more than one immediately previous accessed node, to other concurrent transactions in said data processing system as said nodes are traversed; (d) identifying a next successively located key record relative to the key record to be inserted in said bottom node and limiting all access to the bottom node; (e) limiting all accesses to said next successively located key record; (f) inserting said key record; thereafter, (g) removing

the access limitation to the next successively located key record and the bottom node; and thereafter, (h) removing all remaining access limitations to the accessed nodes.

According to another aspect of the invention there is disclosed in a data processing system, a method executed by a data processor for deleting a key record in a group of record keys stored in a data processing memory according to a key record through an index tree during a transaction in the data processing system wherein other transactions concurrently access said record keys through said index tree, said index tree having a root node connected to at least one intermediate level of nodes, each node having a key record reference to one or more nodes in a next successive level and having bottom nodes that provide access to said record keys, said method comprising the steps of: (a) traversing across said nodes from said root node in said data processing memory through said intermediate level of nodes by accessing a node according to said key record and traversing to a next node indicated by said key record of said accessed node until a bottom node is reached; (b) limiting all but read accesses to the node being accessed and not more than one immediately previous accessed node, to other concurrent transactions in said data processing system as said nodes are accessed; (c) identifying a next successively located key record relative to the key record to be deleted in said bottom node and limiting all access to the bottom node; (d) limiting all accesses to said next successively located key record; (e) deleting said key record; thereafter, (f) removing all remaining access limitations to the accessed nodes; and thereafter, (g) removing the access limitation to the next successively located key record.

According to another aspect of the invention there is disclosed in a data processing system, a method executed by a data processor for altering an index tree structure that provides access to at least one key record in a group of record keys stored in a data processing memory according to a key record for a transaction in the data processing system wherein other transactions concurrently access said record keys through said index tree, said index tree structure having at least one node, said node having a key record reference to either another node or having a key record reference providing direct

access to said key record, said index tree including an indication when said index tree is being changed said method comprising the steps of: (a) determining that no indication of an index tree change is present in said data processing system; (b) providing the indication of an index tree change that allows concurrent non-altering access to the index tree to said data processing system; (c) altering said index tree node(s) in said data processing memory; (d) allowing concurrent non-altering access to said index tree while said index tree is being altered; and (e) removing the indication of an index tree change after completion of said altering step.

According to another aspect of the invention there is disclosed a method of operating a computer system for effecting simultaneous transactions on a database by multiple users using an index configured as a tree structure having a root node and lower level nodes, each node, except those in lowest level leaf nodes, referencing one or more nodes in the next lower level, with leaf nodes referencing corresponding records in the database, and each node, except leaf nodes, including key information to identify the next level node to be accessed in searching for a particular record, and each leaf node including keys relating to records referenced by that leaf node, the keys being ordered throughout the leaf nodes, said method comprising, for a transaction involving insertion or deletion of a current record in the database and its corresponding key in a leaf node in the index, the steps of: traversing through the nodes of the tree from said root node to the leaf node corresponding to said current record in response to an input request to access said current record, using the key information in said nodes; limiting access to a node being traversed and an immediately previously traversed node such that other users are only allowed read access to said node being traversed and said immediately previously traversed node; inserting or deleting said current record and its corresponding record key; and removing, after said step of inserting or deleting, all the access limitations to the traversed nodes; and characterised by the steps of: prior to said step of inserting or deleting, limiting access to said leaf node such that no other user may access said leaf node; identifying a next successively located key in terms of the ordering of

keys, relative to the key of the current record to be deleted or inserted, and limiting access to the record corresponding to said next successively located key such that no other user may access said record; and removing, after said step of inserting or deleting, the access limitation to the record corresponding to the next successively located key.

According to still another aspect of the invention there is disclosed a digital data processing system for effecting simultaneous transactions on a database by multiple users, including an index configured as a tree structure having a root node and lower level nodes, in which each node, except those in lowest level leaf nodes, references one or more nodes in the next lower level, with leaf nodes referencing corresponding records in the database, and in which each node, except leaf nodes, includes key information to identify the next level node to be accessed in searching for a particular record, with each leaf node including keys relating to records referenced by that leaf node, the keys being ordered throughout the leaf nodes, said system further including means for processing a transaction involving insertion or deletion of a current record in the database and its corresponding key in a leaf node in the index, said processing means comprising: means for traversing through the nodes of the tree from said root node to the leaf node corresponding to said current record in response to an input request to access said current record, using the key information in said nodes; means for limiting access to a node being traversed and an immediately previously traversed node such that other users are only allowed read access to said node being traversed and said immediately previously traversed node; means for inserting or deleting said current record and its corresponding record key; and means for removing, after said inserting or deleting, all of the access limitations to the traversed nodes; and characterised by: means for limiting access to said leaf node such that no other user may access said leaf node, for identifying a next successively located key in terms of the ordering of keys, relative to the key of the current record to be deleted or inserted, and for limiting access to the record corresponding to said next successively located key such that no other user may access said record, prior to said inserting or deleting; and means for removing, after said

inserting or deleting, the access limitation to the record corresponding to the next successively located key.

DESCRIPTION OF THE PREFERRED EMBODIMENTS

An embodiment of the invention will now be described, by way of example, with reference to the accompanying drawings, in which:

Figure 1 is a illustration of a database table;

Figure 2 is a illustration of the database table stored in a computer memory;

Figure 3 is a block diagram illustrating a simple B-Tree index;

Figure 4 is an illustration of the storage of index B-Tree node data in a computer memory;

Figure 5 is a flow chart illustrating an initial search through an index B-Tree;

Figure 6 is a flow chart illustrating a fetch through the index B-Tree;

Figure 7 is a flow chart illustrating an insert through the index B-Tree;

Figure 8 is a flow chart illustrating a delete through the index B-Tree;

Figure 9 is a flow chart illustrating a node splitting algorithm; and

Figure 10 is a flow chart illustrating a node collapsing algorithm.

This invention relates to the accessing of records in a database system. The following example is an overly simplified database access problem provided to enable the reader to more easily understand the invention contained herein. The reader should understand, that, in reality, the number of entries and amount of information that is contained in database tables are much greater than illustrated.

Figure 1 illustrates table 10 containing employee names and numbers. Each employee record includes a single employee number. The names in table 10 are listed in alphabetical order. However, it should be apparent to those skilled in the art that this is not how the records are normally stored in the computer memory.

CLAIMS

1. In a data processing system, a method executed by a data processor for fetching key record data stored in a data processing memory in a group of record keys by using a portion of a key record through an index tree during a transaction in the data processing system wherein other transactions may concurrently access said record keys through said index tree, said index tree having at least a root node, at least one intermediate level of at least one node, each node having a key record reference to one or more nodes in a next lower ordered level of nodes and having bottom nodes that provide access to said key record data, said method comprising the steps of:

(a) traversing across said nodes in said data processing memory from said root node through said intermediate level of nodes by accessing a node using said key record portion of the root node and traversing to a next node indicated by said key record portion of the accessed node until a bottom node is reached;

(b) limiting all but read accesses to the node being accessed and an immediate previously accessed node in said data processing memory, to other concurrent transactions in said data processing system;

(c) identifying and accessing said key record in said bottom node;

(d) limiting all but read accesses to said key record;

(e) removing all access limitations to the accessed nodes including the bottom node except for the accessed key record;

(f) fetching key record data; and

(g) removing the access limitation to the key record after the record data has been fetched.

2. A method according to claim 1 wherein said step (c) further includes the steps of:

(c1) identifying and accessing said record keys in said node;

(c2) determining from said record keys in the node if another node is to be accessed;

(c3) if another node is to be accessed, returning to step (a); and

(c4) if another node is not to be accessed, identifying the key record data according to the portion of the key record.

3. A method according to claim 2 wherein said step (f) includes the steps of:

(f1) retrieving said key record in said node;

(f2) accessing a record according said key record in said node; and

(f3) returning record data from said record.

4. A method according to claim 3 wherein said step (g) further includes waiting until the transaction requiring the fetch of key record data is completed before the access limitation is removed from the key record.

5. In a data processing system, a method executed by a data processor for inserting a single key record in a group of record keys stored in a data processing memory according to a key record through an index tree during a transaction in the data processing system wherein other transactions concurrently access said record keys through said index tree, said index tree having at least a root node, at least one intermediate level of at least one node, each node having a key record reference to one or more nodes in a next successive level of nodes and having bottom nodes that provide access to said record keys, said method comprising the steps of:

(a) limiting all accesses to the key record being inserted;

(b) traversing across said nodes in said data processing memory from said root node through said intermediate level of nodes by accessing a node using said key record of said root node and traversing to a next node indicated by said key record of said accessed node until a bottom node is reached;

(c) limiting all but read accesses to the node being accessed and not more than one immediately previous accessed node, to other concurrent transactions in said data processing system as said nodes are traversed;

(d) identifying a next successively located key record relative to the key record to be inserted in said bottom node and limiting all access to the bottom node;

(e) limiting all accesses to said next successively located key record;

(f) inserting said key record, thereafter,

(g) removing the access limitation to the next successively located key record and the bottom node; and thereafter,

(h) removing all remaining access limitations to the accessed nodes.

6. A method according to claim 5 wherein step (f) includes the steps of:

(f1) determining if the key record can be inserted in said node;

(f2) if the key can be inserted into the node, proceeding to step (f6);

(f3) if the key cannot be inserted into the node, providing an indication that the index tree structure is being changed;

(f4) altering the index tree structure to add a node to receive the key record;

(f5) removing the indication of an index tree structure change; and

(f6) inserting said key record in the node.

7. In a data processing system, a method executed by a data processor for deleting a key record in a group of record keys stored in a data processing memory according to a key record through an index tree during a transaction in the data processing system wherein other transactions concurrently access said record keys through said index tree, said index tree having a root node connected to at least one intermediate level of nodes, each node having a key record reference to one or more nodes in a next successive level and having bottom nodes that provide access to said record keys, said method comprising the steps of:

(a) traversing across said nodes from said root node in said data processing memory through said intermediate level of nodes by accessing a node according to said key record and traversing to a next node indicated by said key record of said accessed node until a bottom node is reached;

(b) limiting all but read accesses to the node being accessed and not more than one immediately previous accessed node, to other concurrent transactions in said data processing system as said nodes are accessed;

(c) identifying a next successively located key record relative to the key record to be deleted in said bottom node and limiting all access to the bottom node;

(d) limiting all accesses to said next successively located key record;

(e) deleting said key record; thereafter,

(f) removing all remaining access limitations to the accessed nodes; and thereafter,

(g) removing the access limitation to the next successively located key record.

8. A method according to claim 7 wherein step (e) includes the steps of:

(e1) determining if the key record to be deleted is the only key record in said node;

(e2) if the key to be deleted is not the only key in the node, deleting said key record in the node and then proceeding to step (f);

(e3) if the key to be deleted is the only key in the node, providing an indication that the index tree structure is being changed;

(e4) deleting said key record in the node;

(e5) altering the index tree structure to remove the node; and

(e6) removing the indication of an index tree structure change.

9. A method according to claim 8 wherein said step (g) further includes waiting until the transaction requiring the deletion of the key record is completed before the access limitation is removed from the key record.

10. In a data processing system, a method executed by a data processor for altering an index tree structure that provides access to at least one key record in a group of record keys stored in a data processing memory according to a key record for a transaction in the data processing system wherein other transactions concurrently access said record keys through said index tree, said index tree structure having at least one node, said node having a key record reference to either another node or having a key record reference providing direct access to said key record, said index tree including an indication when said index tree is being changed said method comprising the steps of:

(a) determining that no indication of an index tree change is present in said data processing system;

(b) providing the indication of an index tree change that allows concurrent non-altering access to the index tree to said data processing system;

(c) altering said index tree node(s) in said data processing memory;

(d) allowing concurrent non-altering access to said index tree while said index tree is being altered; and

(e) removing the indication of an index tree change after completion of said altering step.

11. A method of operating a computer system for effecting simultaneous transactions on a database by multiple users using an index configured as a tree structure having a root node and lower level nodes, each node, except those in lowest level leaf nodes, referencing one or more nodes in the next lower level, with leaf nodes referencing corresponding records in the database, and each node, except leaf nodes, including key information to identify the next level node to be accessed in searching for a particular record, and each leaf node including keys relating to records referenced by that leaf node, the keys being ordered throughout the leaf nodes, said method comprising, for a transaction involving insertion or deletion of a current record in the database and its corresponding key in a leaf node in the index, the steps of:

traversing through the nodes of the tree from said root node to the leaf node corresponding to said current record in response to an input request to access said current record, using the key information in said nodes;

limiting access to a node being traversed and an immediately previously traversed node such that other users are only allowed read access to said node being traversed and said immediately previously traversed node;

inserting or deleting said current record and its corresponding record key; and

removing, after said step of inserting or deleting, all the access limitations to the traversed nodes;

and characterised by the steps of:

prior to said step of inserting or deleting, limiting access to said leaf node such that no other user may access said leaf node, identifying a next successively located key in terms of the ordering of keys, relative to the key of the current record to be deleted or inserted, and limiting access to the record corresponding to said next successively located key such that no other user may access said record; and

removing, after said step of inserting or deleting, the access limitation to the record corresponding to the next successively located key.

12. A method according to claim 11, in which, for inserting said current record, the removal of the access limitation to the record associated with the next successively located key precedes the removal of all access limitations to the traversed nodes.

13. A method according to claim 11, in which, for deleting said current record, the removal of all access limitations to the traversed nodes precedes the removal of the access limitation to the record associated with the next successively located key.

14. A method according to claim 11 or 12 including, for inserting said current record, the steps of:

determining if the key associated with the current record can be inserted in said leaf node; and

if the key can be so inserted, inserting the key and the current record; or

if the key cannot be so inserted:

providing an indication that the index tree structure is to be changed;

altering the index tree to add a node to receive said current record;

removing the indication of the tree structure change; and

inserting the key and the current record.

15. A method according to claim 11 or 13 including, for deleting said current record, the steps of:

determining if the key associated with the current record is the only record in said leaf node; and

if it is not deleting the key and the current record; or, if it is; providing an indication that the index tree structure is to be changed;

deleting the key and the current record;

altering the index tree structure to remove the node; and

removing the indication of the index tree structure change.

16. A digital data processing system for effecting simultaneous transactions on a database by multiple users, including an index configured as a tree structure having a root node and lower level nodes, in which each node, except those in lowest level leaf nodes, references one or more nodes in the next lower level, with leaf nodes referencing corresponding records in the database, and in which each node, except leaf nodes, includes key information to identify the next level node to be accessed in searching for a particular record, with each leaf node including keys relating to records referenced by that leaf node, the keys being ordered throughout the leaf nodes, said system further including means for processing a transaction involving insertion or deletion of a current record in the database and its corresponding key in a leaf node in the index, said processing means comprising:

means for traversing through the nodes of the tree from said root node to the leaf node corresponding to said current record in response to an input request to access said current record, using the key information in said nodes;

means for limiting access to a node being traversed and an immediately previously traversed node such that other users are only allowed read access to said node being traversed and said immediately previously traversed node;

means for inserting or deleting said current record and its corresponding record key; and

means for removing, after said inserting or deleting, all of the access limitations to the traversed nodes;

and characterised by

means for limiting access to said leaf node such that no other user may access said leaf node, for identifying a next successively located key in terms of the ordering of keys, relative to the key of the current record to be deleted or inserted, and for limiting access to the record corresponding to said next successively located key such that no other user may access said record, prior to said inserting or deleting; and

means for removing, after said inserting or deleting, the access limitation to the record corresponding to the next successively located key.

17. A system according to claim 16 including, for inserting said current record, means for determining if the key associated with the current record can be inserted in said leaf node, and, for when the key can not be so inserted, means for providing an indication that the index tree structure is to be changed, means for altering the index tree to add a node to receive said current record, and for removing the indication of the tree structure change prior to insertion of the key and the current record.

18. A system according to claim 16 including, for deleting said current record, means for determining if the key associated with the current record is the only record in said leaf node and, for when it is, means for providing an indication that the index tree structure is to be changed means for deleting the key and the current record, means for altering the index tree structure to remove the node, and for removing the indication of the index tree structure change.

CONCURRENT RECORD ACCESS DATABASE SYSTEM

ABSTRACT

A data base system on a computer uses an index tree (40, 50, 52, 54) containing record keys which reference associated records in the data base. The tree, which comprises a root node extending through intermediate nodes to leaf nodes, provides concurrent accesses to a plurality of users. When changes are to be made to the key structure, action has to be taken to prevent users, other than the user making the change, from corrupting the data or receiving corrupted data. To do this, when one user is inserting or deleting a key and its associated record, all accesses to a node being traversed by this user and the immediately traversed mode are limited to read accesses by other users. At the leaf node, all accesses by others to the key and record being added or deleted are prevented together with accesses to the next successively located record key and its associated record. After the deletion or insertion, the access limitations are removed.

Fig. 4