

(19) 日本国特許庁(JP)

(12) 特許公報(B2)

(11) 特許番号

特許第3622443号  
(P3622443)

(45) 発行日 平成17年2月23日(2005.2.23)

(24) 登録日 平成16年12月3日(2004.12.3)

(51) Int. Cl.<sup>7</sup>

G06F 12/00

F I

G06F 12/00 520A

請求項の数 9 (全 13 頁)

(21) 出願番号	特願平9-257239	(73) 特許権者	000004226 日本電信電話株式会社 東京都千代田区大手町二丁目3番1号
(22) 出願日	平成9年9月22日(1997.9.22)	(74) 代理人	100070150 弁理士 伊東 忠彦
(65) 公開番号	特開平11-96058	(72) 発明者	岡田 敏 東京都新宿区西新宿三丁目19番2号 日本電信電話株式会社内
(43) 公開日	平成11年4月9日(1999.4.9)	(72) 発明者	黒岩 淳一 東京都新宿区西新宿三丁目19番2号 日本電信電話株式会社内
審査請求日	平成12年11月21日(2000.11.21)	(72) 発明者	梅田 昌義 東京都新宿区西新宿三丁目19番2号 日本電信電話株式会社内

最終頁に続く

(54) 【発明の名称】 T木インデックス構築方法及び装置及びT木インデックス構築プログラムを格納した記憶媒体

(57) 【特許請求の範囲】

【請求項1】

データベースをメモリ上に常駐させ、高速なデータアクセスを実現するデータベース管理システムで使用するT木インデックスを構築するT木インデックス構築方法において、T木インデックスを該T木インデックスの階層構造の情報をも有するノードからなるT木本体、検索時に条件比較を行うキー値情報、一つキーに対して複数のレコードが対応する重複キー情報の3つの領域に分けて管理し、

前記T木インデックスのノード情報をキーを構成するカラム数、長さ及びキーの重複とは独立に構築し、

分割した前記3つの領域の中でT木本体を構成するノードのみを、ディスク上にバックアップを取得することを特徴とするT木インデックス構築方法。

10

【請求項2】

データベースをメモリ上に常駐させ、高速なデータアクセスを実現するデータベース管理システムで使用するT木インデックスを構築するT木インデックス構築装置において、T木インデックスを該T木インデックスの階層構造の情報をも有するノードからなるT木本体を格納するT木本体格納領域と、

検索時に条件比較を行うキー値情報を格納するキー値格納領域と、

一つキーに対して複数のレコードが対応する重複キー情報を格納する重複キー格納領域と、

前記T木本体格納領域、前記キー値格納領域及び前記重複キー格納領域の3つの領域を管

20

理する管理手段とを有することを特徴とするT木インデックス構築装置。

【請求項3】

前記T木本体格納領域は、  
前記T木インデックスのノード情報をキーを構成するカラム数、長さ及びキーの重複とは独立に構築する請求項2記載のT木インデックス構築装置。

【請求項4】

前記管理手段は、  
分割した前記T木本体格納領域、前記キー値格納領域及び前記重複キー格納領域の3つの領域の中でT木本体を構成するノードのみを、ディスク上にバックアップを取得するバックアップ取得手段を含む請求項2及び3記載のT木インデックス構築装置。

10

【請求項5】

前記T木本体格納領域に格納されるノードは、  
前記ノード内要素に前記キー値格納領域内に格納されているキー値の位置を指すポインタを含む請求項2記載のT木インデックス構築装置。

【請求項6】

前記キー値格納領域に格納されるキー値は、  
1つのキーに対して1つのレコードが対応している場合には、各キー値に対応するレコードの位置を指すポインタを保持し、  
1つのキーに対して複数のレコードが対応している場合には、前記重複キー格納領域のポインタの集合内の一つの要素を指すポインタを保持する請求項2記載のT木インデックス構築装置。

20

【請求項7】

前記重複キー格納領域に格納される前記重複キー情報は、  
前記重複キーに対応するレコードの位置を指すポインタを有する請求項2記載のT木インデックス構築装置。

【請求項8】

前記管理手段は、  
前記T木本体格納領域、前記キー値格納領域及び前記重複キー格納領域内の要素をポインタで結合する手段を含む請求項2、5、6、及び7記載のT木インデックス構築装置。

【請求項9】

データベースをメモリ上に常駐させ、高速なデータアクセスを実現するデータベース管理システムで使用するT木インデックスを構築するT木インデックス構築プログラムを格納した記憶媒体であって、  
T木インデックスを該T木インデックスの階層構造の情報を有するノードからなるT木本体、検索時に条件比較を行うキー値情報、一つキーに対して複数のレコードが対応する重複キー情報の3つの領域に分けて管理する管理プロセスを有し、  
前記管理プロセスは、  
前記T木インデックスのノード情報をキーを構成するカラム数、長さ及びキーの重複とは独立に構築し、分割した前記3つの領域の中でT木本体を構成するノードのみを、ディスク上にバックアップを取得するプロセスを有することを特徴とするT木インデックス構築プログラムを格納した記憶媒体。

30

40

【発明の詳細な説明】

【0001】

【発明の属する技術分野】

本発明は、T木インデックス構築方法及び装置及びT木インデックス構築プログラムを格納した記憶媒体に係り、特に、データベースをメモリ上に常駐させることで高速なデータアクセスを実現する、メモリ常駐データベース管理システム等におけるT木インデックスの構築方法及び装置及びT木インデックス構築プログラムを格納した記憶媒体に関する。

【0002】

50



【発明が解決しようとする課題】

しかしながら、上記従来は、T木の各ノードの内部にキー値（重複キーを含む）を埋め込んでいたため、以下のような問題がある。

1．キーを構成するカラムの数、長さによって、ノードのサイズが変化するため、T木インデックスの構築がキー構成に影響され、T木インデックスの構築が複雑となり、構築に時間がかかる。

【0010】

2．ノード内要素数は固定であるため、キー値のサイズが大きく、空きとして管理される領域が多い場合には、T木インデックスとして使用するメモリ量が増加する。

3．キーの挿入/削除時にキー値の移動が生じるため、レコードの挿入/削除の処理速度が遅い。 10

【0011】

4．重複キーの重複度が増加するにつれて検索対象となるキーの数が増加し、多くのキー値との比較を行わなくてはならないため検索時の処理速度が悪化する。

5．インデックスに含まれる全ての情報についてバックアップを取得しているため、バックアップに要する時間が長い。

【0012】

本発明は、上記の点に鑑みなされたもので、T木インデックス構築時間短縮、メモリ使用量削減、キー挿入/削除時の処理速度の向上、キーの重複時の検索速度の向上及びバックアップ時間の短縮を可能とするT木インデックス構築方法及び装置及びT木インデックス構築プログラムを格納した記憶媒体を提供することを目的とする。 20

【0013】

【課題を解決するための手段】

本発明は、データベースをメモリ上に常駐させ、高速なデータアクセスを実現するデータベース管理システムで使用するT木インデックスを構築するT木インデックス構築方法において、

T木インデックスを該T木インデックスの階層構造の情報を有するノードからなるT木本体、検索時に条件比較を行うキー値情報、一つキーに対して複数のレコードが対応する重複キー情報の3つの領域に分けて管理し、

T木インデックスのノード情報をキーを構成するカラム数、長さ及びキーの重複とは独立に構築し、 30

分割した3つの領域の中でT木本体を構成するノードのみを、ディスク上にバックアップを取得する。

【0014】

図1は、本発明の原理構成図である。

本発明は、データベースをメモリ上に常駐させ、高速なデータアクセスを実現するデータベース管理システムで使用するT木インデックスを構築するT木インデックス構築装置において、

T木インデックスを該T木インデックスの階層構造の情報を有するノードからなるT木本体を格納するT木本体格納領域100と、 40

検索時に条件比較を行うキー値情報を格納するキー値格納領域200と、

一つキーに対して複数のレコードが対応する重複キー情報を格納する重複キー格納領域300と、

T木本体格納領域100、キー値格納領域200及び重複キー格納領域300の3つの領域を管理する管理手段400とを有する。

【0015】

上記のT木本体格納領域100は、T木インデックスのノード情報をキーを構成するカラム数、長さ及びキーの重複とは独立に構築する。

上記の管理手段400は、分割したT木本体格納領域100、キー値格納領域200及び重複キー格納領域300の3つの領域の中でT木本体を構成するノードのみを、ディスク 50

上にバックアップを取得するバックアップ取得手段410を含む。

【0016】

上記のT木本体格納領域100に格納されるノードは、ノード内要素にキー値格納領域内に格納されているキー値の位置を指すポインタを含む。

上記のキー値格納領域200に格納されるキー値は、1つのキーに対して1つのレコードが対応している場合には、各キー値に対応するレコードの位置を指すポインタを保持し、1つのキーに対して複数のレコードが対応している場合には、重複キー格納領域のポインタの集合内の一つの要素を指すポインタを保持する。

【0017】

上記の重複キー格納領域300に格納される重複キー情報は、重複キーに対応するレコードの位置を指すポインタを有する。

上記の管理手段400は、T木本体格納領域、キー値格納領域及び重複キー格納領域内の要素をポインタで結合する手段を含む。

本発明は、データベースをメモリ上に常駐させ、高速なデータアクセスを実現するデータベース管理システムで使用するT木インデックスを構築するT木インデックス構築プログラムを格納した記憶媒体であって、

T木インデックスを該T木インデックスの階層構造の情報を有するノードからなるT木本体、検索時に条件比較を行うキー値情報、一つキーに対して複数のレコードが対応する重複キー情報の3つの領域に分けて管理する管理プロセスを有し、

管理プロセスは、

T木インデックスのノード情報をキーを構成するカラム数、長さ及びキーの重複とは独立に構築し、分割した3つの領域の中でT木本体を構成するノードのみを、ディスク上にバックアップを取得するプロセスを有する。

【0018】

上記のように、本発明は、T木インデックスを以下の3領域に分割して格納する各領域内の要素をポインタで結合することにより、データへのアクセスを可能とする。

T木本体については、T木インデックスの各ノードを格納し、ノード間の階層構造の情報を保持する。

【0019】

また、キー値格納領域については、キー値のみを格納し、管理する。

さらに、重複キー格納領域については、重複キーのみを管理する。

バックアップについては、T木本体のみを取得する。

これにより、従来の方式に比較して以下が可能となる。

1. キーを構成するカラムの数、長さとは独立にT木インデックスのノードの作成、ノード間の階層構造が構築できるため、インデックス構築の処理速度が向上する。

【0020】

2. ノード内にキー値を埋め込まず、キー値を格納する領域は必要最小限に抑えることができるため、T木インデックス全体のメモリの使用量を少なくできる。

3. キー値の挿入/削除における、T木インデックスの成長/衰退がノード間のキー値の移動を伴わないで実施でき、キー値の挿入/削除時の処理速度が向上する。

【0021】

4. 重複キーを扱う場合に重複キーの重複度とは独立にT木インデックスの階層構造が構築でき、T木インデックスの検索時の処理速度が向上する。

5. バックアップのデータ量を削減でき、バックアップ取得に要する時間を短縮できる。

【0022】

【発明の実施の形態】

図2は、本発明のT木インデックスの構成を示す。

同図は、本発明の構築方式を使用したT木インデックス10における各領域の相関関係を示している。分割している各領域の内容は、以下の通りである。

1. T木本体領域11:

10

20

30

40

50

T木インデックス10構築のために必要なノード111を格納する領域である。各ノード111のノード内要素112は、キー値格納領域12内に格納されているキー値の121の位置を指すポインタ114を保持している。

#### 【0023】

##### 2. キー値格納領域12:

検索時に使用するキー値121を格納する領域である。1つのキーに対して1レコードのみが対応している場合には、各キー値121は、キー値121に対応するレコード202の位置を指すポインタ122を保持している。1つのキーに対して複数のレコード202が対応する重複キーを扱う場合には、各キー値121は、重複キー格納領域13内のポインタ132の集合131内の一つの要素を指すポインタ123を保持している。格納して

10

#### 【0024】

##### 3. 重複キー格納領域13:

この領域は、重複キーを扱う場合にのみ作成される。重複キーに対応するレコード202の位置を指すポインタ132を管理している領域である。

図3は、本発明のT木インデックスのノードとキー値の構成を示す。

同図は、キー値121を構成するカラム201とT木インデックス10の関係を例示している。CLM1とCLM2の2つかラムでキー値121を構成する場合と、CLM1のみでキー値121を構成する場合のT木本体11の各ノード111のサイズはどちらも同一となる。そのため、新しいノードの作成時、1つのテーブルに複数のインデックスを張る

20

#### 【0025】

また、各ノード111の要素112内でキー値121へのポインタ114を格納していない空き115が生じててもこの空き115に対応するキー値121の領域は、キー値格納領域12内に確保しないため、T木インデックス10としてのメモリの使用量を少なく抑えることができる。

#### 【0026】

##### 【実施例】

以下、図面と共に、本発明の実施例を説明する。

30

前述の図2、図3の構成に基づいて以下の実施例を説明する。

図4は、本発明の一実施例の重複キーの増加を示す。同図は、重複キー133の重複度とT木本体11の関係を例示している。テーブルへのレコードの追加によりKEY3の重複度が2から3に増加した場合には、KEY3のポインタ123が指している重複キー格納領域13内のポインタの集合131に挿入されたレコードの位置を指すポインタ132を追加する。

#### 【0027】

T木本体11、キー値格納領域12は重要度が増加しても構成は変わらない。そのため、重複キーによるキー値121検索の処理速度の低下を防ぐことができ、重複のないキー値121に対する検索の処理速度が向上する。

40

図5は、本発明の一実施例のキー値の挿入を示す。同図は、T木インデックス10にキー値121を挿入する方法を例示している。T木インデックス10にKEY6を挿入する例を以下に示す。

#### 【0028】

1. KEY6をキー値格納領域12に挿入する。

2. T木本体11のノード111内でKEY6を指すポインタ114を挿入する位置を見つける。キーの順序性より、KEY5, KEY6, KEY7の順序となるように挿入するため、ポインタPt5とポインタPt7の間にKEY6を指すポインタPt6を挿入する。このとき、ポインタPt6を挿入するための領域がないため、ポインタPt4, ポインタPt5についてポインタの張り替えを行い、ポインタPt6を挿入する領域を確保する

50

。

【0029】

3. KEY 6を指すポインタ Pt 6を挿入する。

図6は、本発明の一実施例のキー値の削除を示す。同図は、T木インデックス10からキー値121を削除する方法を例示している。T木インデックス10からKEY 6を削除する例を以下に示す。

1. KEY 6を指すポインタ114 Pt 6を検索する。

【0030】

2. キー値格納領域12からKEY 6を削除する。

3. ポインタ Pt 6をノード111から削除する。

4. ポインタ Pt 6を削除した領域が空き115となり、順序性が保てなくなるため、ポインタ Pt 7、ポインタ Pt 8についてポインタ114の張り替えを行う。

【0031】

図7は、本発明の一実施例のバックアップを示す。同図は、バックアップ30を例示している。T木インデックス10を構成するT木本体11、キー値格納領域12、重複キー格納領域13の3領域のうち、バックアップ30をディスク上に取得するのは、T木本体11のみとする。バックアップ30をT木本体11のみとすることで、バックアップ30のデータ量を削減でき、通常運用の処理速度に影響を与えるバックアップ取得時のディスクへの書き込み時間を短縮することができる。

【0032】

障害からのリカバリ時には、T木本体11のバックアップ30をディスクからメモリ上にロードする。次に、データベース内のテーブル20のバックアップを参照して、キー値121を取り出し、T木本体11の各ノード111の各要素112とポインタ114で結合することで、キー値格納領域12、重複キー格納領域13を作成し、レコード202へのアクセスが可能な状態とする。

【0033】

また、本発明は、上記において、T木インデックスを該T木インデックスの階層構造の情報をもつノードからなるT木本体、検索時に条件比較を行うキー値情報、一つキーに対して複数のレコードが対応する重複キー情報の3つの領域に分けて管理し、T木インデックスのノード情報をキーを構成するカラム数、長さ及びキーの重複とは独立に構築し、分割した3つの領域の中でT木本体を構成するノードのみを、ディスク上にバックアップを取得する処理をプログラムとして構築し、ディスク装置や、フロッピーディスク、CD-ROM等の可搬記憶媒体に格納し、データベースをメモリ上に常駐させる場合に利用することにより汎用的に利用することが可能である。

【0034】

なお、本発明は、上記の実施例に限定されることなく、特許請求の範囲内で種々変更・応用が可能である。

【0035】

【発明の効果】

上述のように、本発明によれば、メモリ常駐データベース管理システムのT木インデックスにおいて、キー値を構成するカラムの数、長さとは独立にT木本体を構成でき、T木の成長、衰退時に、ノード間でキー値を移動することなく、ポインタの張り替えのみで対応できるため、T木の構築が高速化される。

【0036】

また、キー値を格納する領域を最小限に抑えることができるため、メモリ使用量が削減できる。重複キーを扱う場合にもキー値は一つであるため、重複度が高いデータでも効率良く検索できる。

さらに、バックアップについては、T木本体のみであるため、バックアップに要する時間が削減できる。

【図面の簡単な説明】

10

20

30

40

50

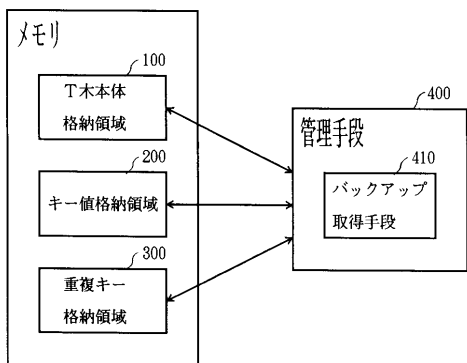
- 【図 1】本発明の原理を説明するための図である。  
 【図 2】本発明の T 木インデックスの構成図である。  
 【図 3】本発明の T 木インデックスのノードとキー値の構成図である。  
 【図 4】本発明の一実施例の重複キーの増加を示す図である。  
 【図 5】本発明の一実施例のキー値の挿入を示す図である。  
 【図 6】本発明の一実施例のキー値の削除を示す図である。  
 【図 7】本発明の一実施例のバックアップを示す図である。  
 【図 8】従来の T 木インデックスの構成図である。  
 【図 9】従来の T 木インデックスにおけるノード構成図である。  
 【図 10】従来の T 木インデックスにおける重複キーの増加を示す図である。  
 【図 11】従来の T 木インデックスにおけるキー値の挿入を示す図である。  
 【図 12】従来の T 木インデックスにおけるキー値の削除を示す図である。  
 【図 13】従来の T 木インデックスのバックアップを示す図である。

【符号の説明】

- |       |                         |    |
|-------|-------------------------|----|
| 1 0   | T 木インデックス               |    |
| 1 1   | T 木本体                   |    |
| 1 2   | キー値格納領域                 |    |
| 1 3   | 重複キー格納領域                |    |
| 2 0   | テーブル                    |    |
| 1 0 0 | T 木本体格納領域               | 20 |
| 1 1 1 | ノード                     |    |
| 1 1 2 | ノード内要素                  |    |
| 1 1 3 | ノード間ポインタ                |    |
| 1 1 4 | ノード内要素とキー値間ポインタ         |    |
| 1 1 5 | 空き                      |    |
| 1 1 6 | ノード内要素とレコード間ポインタ        |    |
| 1 2 1 | キー値                     |    |
| 1 2 2 | キー値とレコード間ポインタ           |    |
| 1 2 3 | キー値と重複キーポインタ集合間ポインタ     |    |
| 1 3 1 | 重複キーポインタ集合内要素とレコード間ポインタ | 30 |
| 1 3 2 | 重複キーポインタ集合内要素とレコード間ポインタ |    |
| 1 3 3 | 重複キー                    |    |
| 2 0 0 | キー値格納領域                 |    |
| 3 0 0 | 重複キー格納領域                |    |
| 4 0 0 | 管理手段                    |    |
| 4 1 0 | バックアップ取得手段              |    |

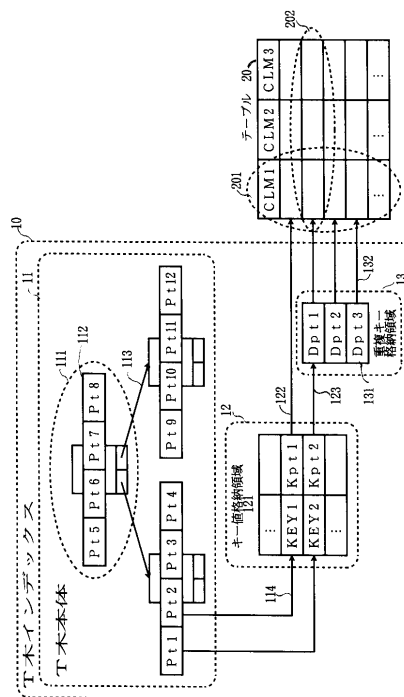
【 図 1 】

本発明の原理構成図



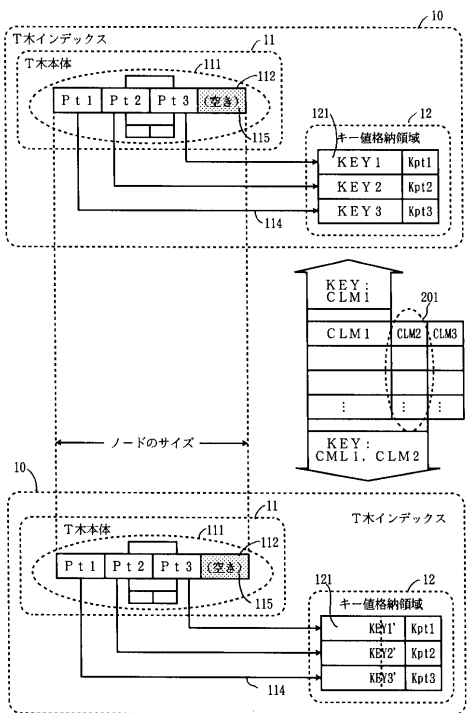
【 図 2 】

本発明のT木インデックスの構成図



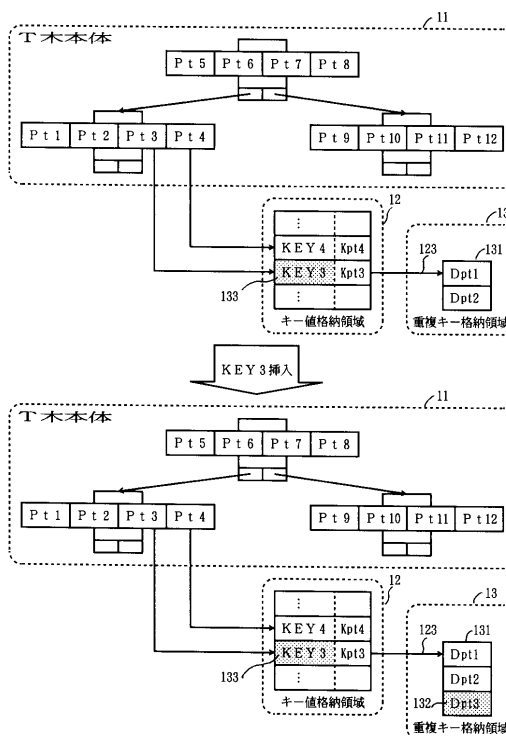
【 図 3 】

本発明のT木インデックスのノードとキー値の構成図



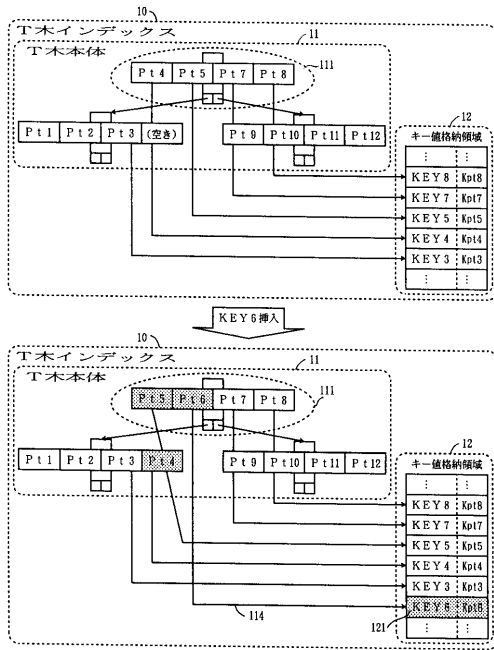
【 図 4 】

本発明の一実施例の重複キーの増加を示す図



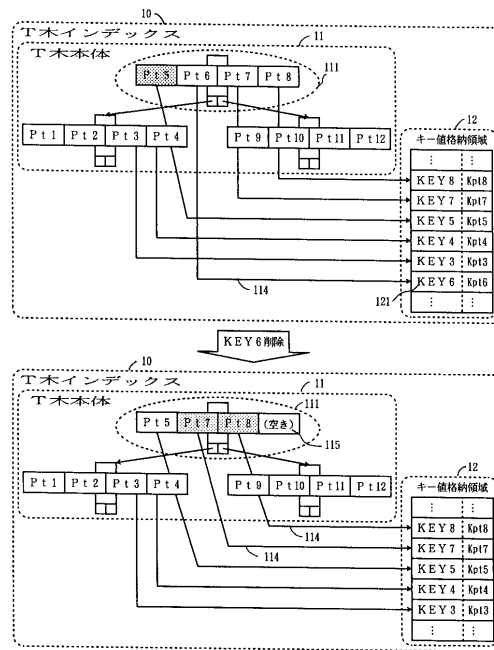
【図5】

本発明の一実施例のキー値の挿入を示す図



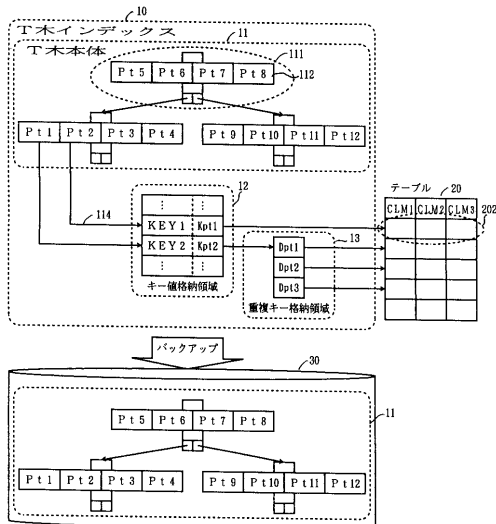
【図6】

本発明の一実施例のキー値の削除を示す図



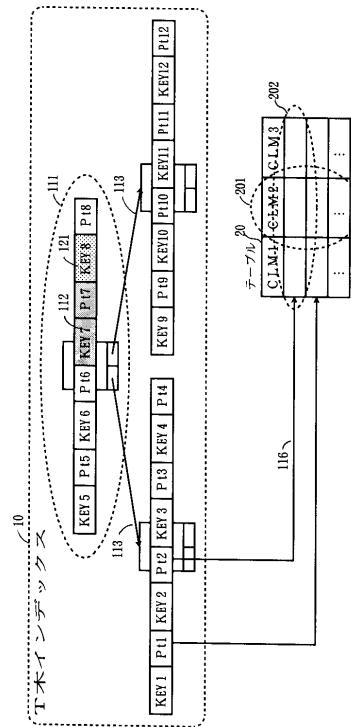
【図7】

本発明の一実施例のバックアップを示す図



【図8】

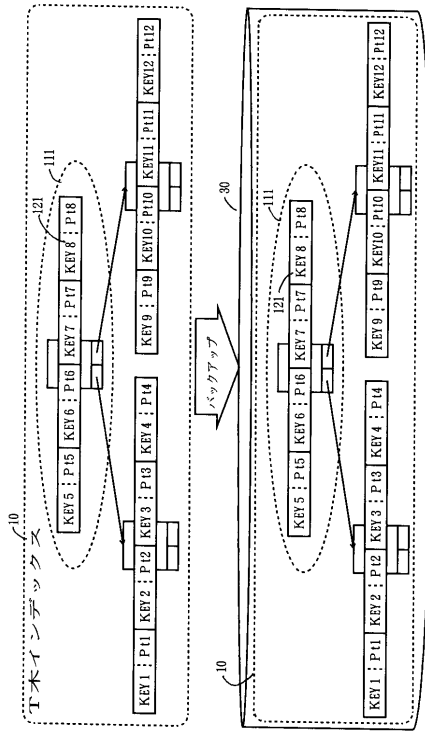
従来のT木インデックスの構成図





【 図 1 3 】

従来のT木インデックスのバックアップを示す図



---

フロントページの続き

審査官 平井 誠

(56)参考文献 特表2000-501861(JP,A)

奥村徹、内徳俊夫、岡田敏、Memory Resident Indexを用いたデータベースの高速化とそのバックアップ機構に関する研究、電子情報通信学会技術研究報告(IN95-13)、日本、社団法人電子情報通信学会、1995年7月14日、Vol.95, No.143, p.1-7  
都司達夫、関係データベースの基礎とISAMシステム、インターフェース、日本、CQ出版株式会社、1991年7月1日、VOL.17, NO.7, p.201-218

(58)調査した分野(Int.Cl.<sup>7</sup>, DB名)

G06F 12/00