

(12) 特許協力条約に基づいて公開された国際出願

(19) 世界知的所有権機関
国際事務局

(43) 国際公開日
2012年6月21日(21.06.2012)



(10) 国際公開番号
WO 2012/081165 A1

- (51) 国際特許分類:
G06F 17/30 (2006.01) G06F 12/00 (2006.01)
- (21) 国際出願番号: PCT/JP2011/006221
- (22) 国際出願日: 2011年11月7日(07.11.2011)
- (25) 国際出願の言語: 日本語
- (26) 国際公開の言語: 日本語
- (30) 優先権データ:
特願 2010-280397 2010年12月16日(16.12.2010) JP
- (71) 出願人(米国を除く全ての指定国について): 日本電気株式会社(NEC CORPORATION) [JP/JP]; 〒1088001 東京都港区芝五丁目7番1号 Tokyo (JP).
- (72) 発明者; および
- (75) 発明者/出願人(米国についてのみ): 佐々木 盛朗(SASAKI, Shigero) [JP/JP]; 〒1088001 東京都港区芝五丁目7番1号日本電気株式会社内 Tokyo (JP).
- (74) 代理人: 速水 進治(HAYAMI, Shinji); 〒1410031 東京都品川区西五反田7-9-2 五反田T Gビル9階 Tokyo (JP).
- (81) 指定国(表示のない限り、全ての種類の国内保護が可能): AE, AG, AL, AM, AO, AT, AU, AZ, BA, BB, BG, BH, BR, BW, BY, BZ, CA, CH, CL, CN, CO, CR, CU, CZ, DE, DK, DM, DO, DZ, EC, EE, EG, ES, FI, GB, GD, GE, GH, GM, GT, HN, HR, HU, ID, IL, IN, IS, JP, KE, KG, KM, KN, KP, KR, KZ, LA, LC, LK, LR, LS, LT, LU, LY, MA, MD, ME, MG, MK, MN, MW, MX, MY, MZ, NA, NG, NI, NO, NZ, OM, PE, PG, PH, PL, PT, QA, RO, RS, RU, RW, SC, SD, SE, SG, SK, SL, SM, ST, SV, SY, TH, TJ, TM, TN, TR, TT, TZ, UA, UG, US, UZ, VC, VN, ZA, ZM, ZW.
- (84) 指定国(表示のない限り、全ての種類の広域保護が可能): ARIPO (BW, GH, GM, KE, LR, LS, MW, MZ, NA, RW, SD, SL, SZ, TZ, UG, ZM, ZW), ユーラシア (AM, AZ, BY, KG, KZ, MD, RU, TJ, TM), ヨーロッパ (AL, AT, BE, BG, CH, CY, CZ, DE, DK, EE, ES, FI, FR, GB, GR, HR, HU, IE, IS, IT, LT, LU, LV, MC, MK, MT, NL, NO, PL, PT, RO, RS, SE, SI, SK, SM, TR),

[続葉有]

(54) Title: DATABASE MANAGEMENT DEVICE AND DATABASE MANAGEMENT METHOD

(54) 発明の名称: データベース管理装置及びデータベース管理方法

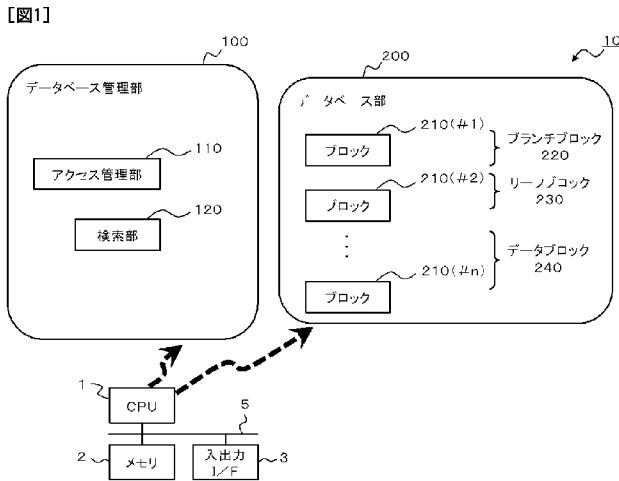


FIG. 1:
 2 Memory
 3 Input/output interface
 100 Database management unit
 110 Access management unit
 120 Search unit
 200 Database unit
 210(#1), 210(#2), 210(#n) Block
 220 Branch block
 230 Leaf block
 240 Data block

(57) Abstract: A database management device of the present invention is provided with: a plurality of index blocks, such that the index blocks have each of at least one index entry for identifying one row datum constituting table data and another index block, and an access counter, and have a tree structure; and access management means which, by referring to each of the index entries in response to a data operation upon the table data, updates each of the access counters of the plurality of index blocks that have been accessed along the tree structure.

(57) 要約: データベース管理装置は、表データを構成する1つの行データ又は他のインデックスブロックを特定するための少なくとも1つのインデックスエントリ、及び、アクセスカウンタをそれぞれ有し、ツリー構造を持つ複数のインデックスブロックと、表データに対するデータ操作に応じて各インデックスエントリを参照することによりツリー構造に沿ってアクセスされた複数のインデックスブロックの各アクセスカウンタをそれぞれ更新するアクセス管理手段と、を備える。



WO 2012/081165 A1

OAPI (BF, BJ, CF, CG, CI, CM, GA, GN, GQ, GW, ML, MR, NE, SN, TD, TG). 添付公開書類:

— 国際調査報告 (条約第 21 条(3))

明 細 書

発明の名称：データベース管理装置及びデータベース管理方法 技術分野

[0001] 本発明は、ツリー構造のインデックスが付加されたデータベースの管理技術に関する。

背景技術

[0002] 大量のデータから少量のデータを高速に検索するために、データにインデックスを付与するのが一般的である。検索処理が多く実行されるのはデータベースであり、最も広く用いられているデータベースの一つは、表形式でデータを管理する関係データベースである。表は、行と列を持ち、例えば、一つの行に一つの取引に関するデータを格納する。行は、複数の列から構成され、例えば、ある列に日付データを、別の列に売上金額を格納する。

[0003] この場合、特定の日付における総売上金額を得る方法には、フルスキャン、インデックススキャン等がある。フルスキャンでは、全ての行に関して日付が一致するかどうかを調べ、一致する行の売上高を加算する。インデックススキャンでは、日付が一致する行をインデックスによって特定し、特定された行から売上高の総和を得る。インデックススキャンは、全行数に比べて日付が一致する行の数が十分に小さいときに効率的である。

[0004] Bツリーは、インデックスを格納するデータ構造の一つとして知られている。例えば、非特許文献1には、Bツリーを使った検索、Bツリーへのデータの挿入、Bツリーへのデータの削除の各アルゴリズムが記載されている。非特許文献2には、広く用いられている関係データベースのORACLE（登録商標）におけるBツリーの説明が記載されている。

[0005] このようなBツリーインデックスを用いることにより、行数を n とすると、「 $\log n$ 」に比例する計算量でデータを検索することができる。なお、フルスキャンの計算量は、 n に比例する。

先行技術文献

特許文献

- [0006] 特許文献1：特開2008-15810号公報
特許文献2：特開2004-295790号公報
特許文献3：特開2008-225575号公報
特許文献4：米国特許出願公開第2008/222090号明細書
特許文献5：特開2009-122850号公報
特許文献6：特開2010-140362号公報
特許文献7：特開平07-200376号公報

非特許文献

- [0007] 非特許文献1：Comer, D. "Ubiquitous B-Tree", ACM Computing Surveys, vol. 11, no. 2, p. 121-137, June 1979
非特許文献2："Oracle Database概要"、10g リリース2、部品番号：B19215-02、2006年3月（図5-7）、http://otndnld.oracle.co.jp/document/products/oracle10g/102/doc_cd/server.102/B19215-02.pdf
非特許文献3："Oracle Databaseリファレンス"、10g リリース2、部品番号：B19228-04、2009年6月（図5-7）、http://otndnld.oracle.co.jp/document/products/oracle10g/102/doc_cd/server.102/B19215-02.pdf
非特許文献4：木脇高太郎、「おら、オラ、オラクル」、翔泳社、2003年5月（pp. 199）
非特許文献5：Drepper, U. What Every Programmer Should Know about Memory, November, 2007. <http://people.redhat.com/drepper/cpumemory.pdf>

発明の概要

- [0008] Bツリーインデックスが付加されたデータベースでは、一般的には、検索は高速であるが、データの挿入及び削除は低速になる。Bツリーインデックスが付加されていないデータベースに対するデータの挿入及び削除は、単に表として記録されたデータを操作すればよい。しかしながら、Bツリーインデックスが付加された表に対するデータの挿入及び削除は、Bツリーに記録されたデータも操作しなければならないからである。更に、Bツリーでは、

操作を加えるブロックの空き領域の過不足によって、ブロック自体の追加及び削除が発生する場合もある。この場合、そのブロックの上位のブロックのデータも操作する必要がある。

[0009] また、CPU (Central Processing Unit) 速度とメモリ速度との差の拡大が、データベースの性能に影響を与えている。特に、ランダムアクセス時のメモリのレイテンシは大きく、メモリ上のデータへのアクセスを待っている間、CPUはストールすることになる。このようなランダムアクセスは、例えば、リストをたどる処理実行時に発生し、Bツリーインデックスが利用されるデータベースにおいては、そのリストをたどる処理が頻繁に発生する。

[0010] このようにデータベースの性能に影響を与える要因は、処理手法、ハードウェア等、様々であるため、現在、データベースの性能を最適化するのは困難となっている。加えて、データベースの性能を最適化するのに有用な情報が存在しないのが現状である。

[0011] 本発明の目的は、ツリー構造のインデックスが付加されたデータベースの性能を最適化するためのデータベース管理技術を提供することにある。

[0012] 本発明の各態様では、上述した課題を解決するために、それぞれ以下の構成を採用する。

[0013] 第1の態様は、表データを格納するデータブロックを含むデータベースを管理するデータベース管理装置に関する。第1の態様に係るデータベース管理装置は、表データを構成する1つの行データ又は他のインデックスブロックを特定するための少なくとも1つのインデックスエントリ、及び、アクセスカウンタをそれぞれ有し、ツリー構造を持つ複数のインデックスブロックと、表データに対するデータ操作に応じて各インデックスエントリを参照することによりツリー構造に沿ってアクセスされた複数のインデックスブロックの各アクセスカウンタをそれぞれ更新するアクセス管理手段と、を備える。

[0014] 第2の態様は、表データを格納するデータブロックを含むデータベースを管理するデータベース管理方法において、表データを構成する1つの行デー

タ又は他のインデックスブロックを特定するための少なくとも1つのインデックスエントリ、及び、アクセスカウンタをそれぞれ有し、ツリー構造を持つ複数のインデックスブロックを備えるコンピュータが、表データに対するデータ操作に応じて各インデックスエントリを参照し、その参照によりツリー構造に沿ってアクセスされた複数のインデックスブロックの各アクセスカウンタをそれぞれ更新する。

[0015] なお、本発明の他の態様としては、上記構成をコンピュータに実現させるプログラムであってもよいし、このプログラムを記録したコンピュータが読み取り可能な記録媒体であってもよい。

[0016] 上記各態様によれば、ツリー構造のインデックスが付加されたデータベースの性能を最適化するためのデータベース管理技術を提供することができる。

図面の簡単な説明

[0017] [図1]図1は、第1実施形態におけるDBシステムの構成例を示す概念図である。

[図2]図2は、第1実施形態におけるインデックスブロックの構成例を概念的に示す図である。

[図3]図3は、第1実施形態におけるDBシステムの動作例を示すフローチャートである。

[図4]図4は、ブランチブロック、リーフブロック及びデータブロックの関係の例を概念的に示す図である。

[図5]図5は、第2実施形態におけるDBシステムの構成例を示す概念図である。

[図6]図6は、第2実施形態におけるインデックスブロックの構成例を概念的に示す図である。

[図7]図7は、第2実施形態におけるDBシステムのデータ挿入時の動作例を示すフローチャートである。

[図8]図8は、第2実施形態におけるDBシステムのデータ削除時の動作例を

示すフローチャートである。

[図9]図9は、第3実施形態におけるDBシステムの構成例を示す概念図である。

[図10]図10は、第3実施形態におけるインデックスブロックの構成例を概念的に示す図である。

[図11]図11は、第4実施形態におけるDBシステム10の構成例を示す概念図である。

発明を実施するための形態

[0018] 以下、本発明の実施の形態について説明する。

[0019] 本実施形態に係るデータベース管理装置は、表データを格納するデータブロックを含むデータベースを管理する。このデータベース管理装置は、表データを構成する1つの行データ又は他のインデックスブロックを特定するための少なくとも1つのインデックスエントリ、及び、アクセスカウンタをそれぞれ有し、ツリー構造を持つ複数のインデックスブロックと、上記表データに対するデータ操作に応じて各インデックスエントリを参照することによりツリー構造に沿ってアクセスされた複数のインデックスブロックの各アクセスカウンタをそれぞれ更新するアクセス管理手段と、を備える。

[0020] 上記データベース管理装置では、ツリー構造を持つ各インデックスブロックにはアクセスカウンタがそれぞれ設けられており、データブロック内の行データに対する1つのデータ操作に応じてツリー構造に沿ってアクセスされた複数のインデックスブロックの各アクセスカウンタがそれぞれ更新される。

[0021] 従って、上記データベース管理装置によれば、ツリー構造のインデックスが付加されたデータベースにおいて、各インデックスブロックに関しアクセスされた回数情報をそれぞれ得ることができる。インデックスブロックへのアクセス回数は、データブロック及びインデックスブロックを含めた全てのブロックへのアクセス回数に対して大きな割合を占めることが多い。インデックスブロックは、通常メモリ上に保持される。そして、CPU (Central P

rocessing Unit) とメモリとの速度差のために、インデックスブロックへのアクセスはボトルネックになりがちである。従って、インデックスブロックへのアクセスに関する情報を提供することは、データベースの性能を最適化する上で極めて重要である。

[0022] 以下、上述の実施形態について更に詳細を説明する。以下の各実施形態は、上述のデータベース管理装置の構成を含むデータベースシステム（以降、DBシステムと表記する）の例である。なお、以下に挙げた各実施形態はそれぞれ例示であり、本発明は以下の各実施形態の構成に限定されない。

[0023] [第1実施形態]

[システム構成]

図1は、第1実施形態におけるDBシステム10の構成例を示す概念図である。DBシステム10は、図1に示すように、ハードウェア構成として、CPU1、メモリ2（RAM（Random Access Memory）、ROM（Read Only Memory）、ハードディスク（HDD）等）、入出力インタフェース3等を有する。これら各ハードウェア要素は例えばバス5により接続される。入出力インタフェース3は、外部のコンピュータ等と通信を行うためのネットワークインタフェースやユーザインタフェース等を含む。

[0024] DBシステム10は、例えば、メモリ2に格納されるプログラムがCPU1により読み出され実行されることにより、以下のような各処理部を実現する。DBシステム10は、データベース管理部（以降、DB管理部と表記する）100、データベース（以降、DBと表記する）部200等を有する。

[0025] なお、図1の例では、DBシステム10が1台のコンピュータとして実現される例を示すが、複数のコンピュータで実現されてもよい。また、図1の例では、1つのCPU1のみが示されるが、複数のプロセッサ（例えば、CPU、DSP（Digital Signal Processor）等）を有していてもよい。例えば、DB管理部100とDB部200とが異なるコンピュータ又はCPU上で実現されてもよい。本実施形態は、DBシステム10のハードウェア構成を限定しない。

- [0026] DB部200は、メモリ2上に実現され、複数のブロック210（図1では210（#1）、210（#2）、210（#n）として表記）を含む。各ブロック210は、例えば、所定容量（例えば、4キロバイト（KB））の連続した記憶領域として確保される。ブロック210間は、連続領域として確保されてもよいし、非連続な各領域として確保されてもよい。本実施形態は、ブロック210の領域サイズを限定しない。
- [0027] 各ブロック210は、インデックスを格納するブロック（以降、インデックスブロックとも表記される）と、データを格納するブロック（以降、データブロックとも表記される）240とに大別される。データブロック240は、任意の表（テーブル）を格納する。
- [0028] インデックスブロックは、ツリー構造（例えば、Bツリー（Balanced Tree）構造）を持ち、更に、ブランチブロック220と、リーフブロック230とに大別される。ブランチブロック220及びリーフブロック230の各々は、インデックスとなる少なくとも1つのエントリを格納する。各インデックスブロックに格納されるエントリは、インデックスエントリと呼ぶこともできる。インデックスエントリは、検索対象となるキー値と識別子（ID）とを含む。
- [0029] リーフブロック230は、インデックスブロックの中の最下層のブロックである。リーフブロックのエントリに含まれる識別子は、データブロックのいずれか1つの行データを特定するためのデータである。その識別子は、例えば、特定すべき行データの先頭を指すポインタとその行データのサイズとから構成される。
- [0030] ブランチブロック220は、リーフブロック230又は他のブランチブロック220へのリンクを持つブロックである。このリンクは、インデックスエントリの識別子によって実現される。即ち、ブランチブロック220のエントリに含まれる識別子は、いずれか1つのブランチブロック220又はいずれか1つのリーフブロック230を特定するポインタである。以降、最上位のブランチブロック220をルートブロックと表記する場合もある。ここ

で、最上位とは、検索時に最初に検索されることを意味する。

[0031] 以降、インデックスエントリの識別子によりリンクされた各インデックスブロックにおいて、そのインデックスエントリを含むブランチブロック220を親ブロックと表記し、そのインデックスエントリの識別子で特定されるリーフブロック230又は他のブランチブロック220を子ブロックと表記する場合もある。

[0032] 図2は、第1実施形態におけるインデックスブロックの構成例を概念的に示す図である。図2に示すように、各インデックスブロックは、ブロックヘッダと少なくとも1つのインデックスエントリとをそれぞれ含む。インデックスエントリは、上述したように、キー値（図2の「key」）と識別子（図2の「id」）とを含む。ブロックヘッダには、ブロック種、アクセスカウンタ等が設定される。ブロック種は、リーフブロック、ブランチブロック等を識別するための情報である。アクセスカウンタは、インデックスブロックへのアクセスの数をカウントする。

[0033] 図2では、インデックスブロック内の連続領域に、ブロックヘッダ内の各値及び各エントリが格納される例を示したが、ブロックヘッダ内の各値と各エントリとは、相互に関連付けられていれば、必ずしも連続した領域に格納されなくともよい。

[0034] DB管理部100は、アクセス管理部110、検索部120等を含む。DB管理部100を構成するこれら各処理部も、メモリ2に格納されるプログラムがCPU1により読み出され実行されることにより、ソフトウェア要素として実現される。

[0035] 検索部120は、検索キーを取得し、この検索キーに対応する列（フィールド）のデータを含む行をデータブロック240から抽出する。検索キーは、他の装置から通信を介して取得されてもよいし、CPU1により実行されているプロセス等のような他の処理部から取得されてもよいし、ユーザインタフェースを介してユーザにより入力されてもよい。

[0036] 検索部120は、データブロック240における抽出すべき行を特定する

ために、インデックス検索を行う。このインデックス検索において、検索部 120 は、まず、ルートブロックにアクセスする。検索部 120 は、ルートブロックに含まれるエントリのうち、検索キー以下の最大のキー値を持つエントリ、又は、検索キー以上の最小のキー値を持つエントリを検索する。

[0037] 検索部 120 は、抽出されたエントリの識別子により特定される他のブランチブロック 220 又はリーフブロック 230 を読み出す。検索部 120 は、他のブランチブロック 220 が読み出された場合には、上記のルートブロックに対する検索と同様に特定のエントリを抽出する。一方、検索部 120 は、リーフブロック 230 が読み出された場合には、検索キーの条件を満たすエントリを抽出し、そのエントリの識別子で特定される行データをデータブロック 240 から抽出する。

[0038] アクセス管理部 110 は、検索部 120 からの指示に応じて動作し、対象のインデックスブロックのブロックヘッダのアクセスカウンタを増加させる。具体的には、アクセス管理部 110 は、検索部 120 がインデックス検索を行うことによりインデックスエントリの識別子により特定されるブランチブロック 220 又はリーフブロック 230 を読み出す際に、その読み出されたインデックスブロックのブロックヘッダのアクセスカウンタを増加させる。但し、ルートブロックは最初に参照されるブロックであるため、検索部 120 は、アクセスした際に、ルートブロックのブロックヘッダのアクセスカウンタを増加させる。

[0039] アクセス管理部 110 は、検索部 120 がルートブロックのいずれかのエントリの識別子が示すブランチブロック 220 を読む際に、そのブランチブロック 220 のアクセスカウンタを 1 増加させる。更に、アクセス管理部 110 は、検索部 120 がこのブランチブロック 220 からたどるリーフブロック 230 にアクセスするときに、リーフブロック 230 のアクセスカウンタを 1 増加させる。

[0040] [動作例]

以下、第 1 実施形態における DB システム 10 の動作例について図 3 及び

図4を用いて説明する。図3は、第1実施形態におけるDBシステム10の動作例を示すフローチャートである。図3の例では、完全一致検索を行う場合の動作が示される。図4は、ブランチブロック220、リーフブロック230及びデータブロック240の関係の例を概念的に示す図である。ここでは、図4の例において、文字列の大小を辞書順に従って比較することを想定し、検索部120が、検索キー「g o」を列データに含む行を検索する場合を例に挙げて、当該動作例を説明する。

[0041] 検索部120は、検索キーを取得すると、まず、ルートブロックを読み出す(S10)。図4の例では、検索部120は、検索キー「g o」を取得すると、キー値「e」、「m」、「t」を含む各インデックスエントリを持つルートブロックを読み出す。

[0042] アクセス管理部110は、検索部120がルートブロックにアクセスした場合に、そのルートブロックのアクセスカウンタを1増加させる(S11)。

[0043] 検索部120は、ルートブロック内のインデックスエントリのうち、当該検索キー以下の最大のキー値を持つエントリ、又は、当該検索キー以上の最小のキー値を持つエントリを特定する。いずれの特定方法を取っても一般性を失わないので、ここでは、検索部120は、当該検索キー以下の最大のキー値を持つエントリを特定するものとする(S12)。このエントリの特定は、ブロック中の全てのエントリを比較することにより実現されてもよいし、ブロック中のエントリをキー値でソートした上で一部のエントリを比較することにより実現されてもよい。

[0044] 図4の例では、簡略化されているが、キー値「e」については、例えば、キー値「e」を持つエントリと「e」より小さい値を示すキー値(例えば、NULL)を持つエントリとが含まれる。ここで、検索キー「g o」以下の最大のキー値は「e」であり、検索キー「g o」は、キー値「e」より大きくキー値「m」よりも小さいので、左から2番目のブランチブロックを特定する識別子を含むエントリが特定される。

- [0045] 続いて、検索部120は、そのように特定されたエントリの識別子によって特定されるインデックスブロックを読み出す(S13)。
- [0046] アクセス管理部110は、検索部120がインデックスブロックを読み出すと、そのインデックスブロックのアクセスカウンタを1増加させる(S14)。
- [0047] 検索部120は、ブロックヘッダに設定されるブロック種により、その読み出されたインデックスブロックがリーフブロック230かブランチブロック220かを判定する(S15)。検索部120及びアクセス管理部110は、その読み出されたインデックスブロックがブランチブロック220である場合には(S15; NO)、その読み出されたブランチブロック220に関し上記(S12)、(S13)及び(S14)の処理を実行する。
- [0048] 検索部120は、その読み出されたインデックスブロックがリーフブロック230である場合には(S15; YES)、その読み出されたリーフブロック230内のエントリの中から、検索キーを含む検索条件に一致するエントリを特定する(S16)。ここでは、その検索条件は、検索キーを列データに含む行(完全一致)の検索を示す。図4の例では、キー値「go」を含むインデックスエントリが特定される。
- [0049] このインデックスエントリの特定は、リーフブロック230中の全てのエントリを比較することにより実現されてもよいし、リーフブロック230中のエントリをキー値でソートした上で一部のエントリを比較することにより実現されてもよい。
- [0050] 検索部120は、特定されたインデックスエントリに含まれる識別子で特定される行データを抽出する(S17)。図4の例では、キー値「go」を含むインデックスエントリの識別子により特定されるデータブロック240の行が抽出される。例えば、その識別子に含まれるポインタで、行データの先頭が特定され、その先頭アドレスから行の大きさ(バイト数)分のデータが読み出される。
- [0051] [第1実施形態の作用及び効果]

このように第1実施形態では、検索部120によるツリー構造に沿ったインデックス検索に応じて、データブロック240の少なくとも1つの行データが抽出されるまでにたどられた検索経路に含まれるインデックスブロックの各アクセスカウンタがアクセス管理部110によりそれぞれ更新される。

[0052] 従って、第1実施形態によれば、検索時にブランチブロック220及びリーフブロック230へのアクセスの回数がブロック毎に記録されるため、各インデックスブロックのアクセスカウンタを参照することにより、ブロック毎のアクセス負荷を容易に見積もることができる。各ブロックのアクセス負荷を見積もることができれば、例えば、その負荷に応じてCPUキャッシュに置くべきインデックスブロックを決める等、データベースの性能を最適化することも可能である。

[0053] なお、第1実施形態では、データブロック240へのアクセスの回数はカウントされない。これは、データブロック240におけるブロック当たりのアクセス数がインデックスブロックのそれに比べて少ないからである。第1実施形態では、データベース性能を最適化する上で特に有用な情報として、インデックスブロックのアクセス回数がカウントされる。このようにして、第1実施形態では、アクセス回数のカウント処理で使われる負荷を必要最小限としている。もちろん、本実施形態において、データブロック240へのアクセスの回数をカウントするようにしてもよい。

[0054] [第2実施形態]

[システム構成]

図5は、第2実施形態におけるDBシステム10の構成例を示す概念図である。第2実施形態におけるDBシステム10は、図5に示すように、第1実施形態の構成に加えて、データ挿入部130及びデータ削除部140を更に有する。以下、第2実施形態におけるDBシステム10について、第1実施形態と異なる内容を中心に説明し、第1実施形態と同一の内容についての説明は適宜省略する。データ挿入部130及びデータ削除部140についても、メモリ2に格納されるプログラムがCPU1により読み出され実行され

ることにより、ソフトウェア要素として実現される。

- [0055] データ挿入部130は、挿入する行データを受け取り、データブロック240にその行データを書き込む。このとき、データ挿入部130は、その挿入された行データに基づいて、インデックスブロック内のインデックスを更新する。データブロック240への書き込みは、例えば、挿入すべき行データよりも大きな空き領域を持つブロックのいずれか1つに書き込まれる。
- [0056] データ挿入部130は、インデックスの更新において、少なくとも、対応するリーフブロック230に、その挿入された行データに対応する新たなインデックスエントリを追加する。この新たなインデックスエントリは、キーが設定される列であって挿入された行に含まれる列データをキー値として含み、更に、その行を特定するための識別子を含む。
- [0057] インデックスの更新では、リーフブロック230へのエントリの追加に加えて、ブランチブロック220にも新たなインデックスエントリを追加することが要求される場合がある。それは、挿入対象のリーフブロック230に、新たなインデックスエントリを追加するための十分な空き領域がない場合（オーバーフローが起きた場合）である。
- [0058] このようにオーバーフローが起きた場合には、データ挿入部130は、ブロック分割を行うことによって空き領域を作る。ブロック分割では、データ挿入部130は、所定容量の空き領域がなくなったリーフブロック230のデータの一部を新たに確保したリーフブロック230に移すことで空き領域を作り出す。ブロック分割は、リーフブロック230のみでなくブランチブロック220においても同様に処理される。ブロック分割の詳細については動作例の項で詳述する。
- [0059] データ削除部140は、削除する行データを特定するための値を受け取り、データブロック240から行データを削除し、削除された行データに対応するインデックスエントリをリーフブロック230から削除する。例えば、削除する行データを特定するための値は、検索部120の検索処理の結果として取得される。

- [0060] インデックスエントリをリーフブロック230から削除することにより、リーフブロック230の空き領域が所定サイズ以上となった場合（アンダーフローが起きた場合）、データ削除部140は、このリーフブロック230を空きブロックにする。このアンダーフロー発生時の処理については動作例の項において詳述する。
- [0061] 図6は、第2実施形態におけるインデックスブロックの構成例を概念的に示す図である。第2実施形態では、図6に示すように、アクセスカウンタとして、リードカウンタとライトカウンタとが設けられる。リードカウンタは、リードアクセスの数をカウントし、ライトカウンタは、ライトアクセス（エントリ削除も含む）の数をカウントする。
- [0062] これにより、第2実施形態におけるアクセス管理部110は、第1実施形態で説明したデータ検索時にはリードカウンタを増加させる。更に、アクセス管理部110は、行データ挿入時においてデータ挿入部130により新たなインデックスエントリが追加されると、そのインデックスエントリが追加されたインデックスブロックのライトカウンタを1増加させる。このとき、アクセス管理部110は、そのインデックスエントリが追加されたインデックスブロックを検索するまでにアクセスされた各インデックスブロックのリードカウンタをそれぞれ1増加させる。
- [0063] 同様に、行データ挿入時においてブロック分割が行われる場合には、アクセス管理部110は、そのブロック分割処理においてアクセスされたインデックスブロックのリードカウンタを増加させ、かつ、ブロック分割処理でエントリが移動させられる度に移動先のインデックスブロックのライトカウンタを1増加させる。
- [0064] また、アクセス管理部110は、データ削除部140がインデックスブロックからエントリを削除した場合には、そのインデックスブロックのライトカウンタを増加させる。更に、アクセス管理部110は、データ削除部140がインデックスブロックを空きブロックにする場合には、そのインデックスブロックを空きブロックにするために移動されたエントリの移動先となる

他のインデックスブロックのライトカウンタを増加させる。

[0065] [動作例]

以下、第2実施形態におけるDBシステム10の動作例について図7及び図8を用いて説明する。図7は、第2実施形態におけるDBシステム10のデータ挿入時の動作例を示すフローチャートである。

[0066] データ挿入部130は、挿入すべき行データを取得すると、この行データを空き領域のある任意のデータブロック240に書き込む(S20)。

[0067] データ挿入部130は、挿入された行データに対応するインデックスエントリを挿入すべきリーフブロック230を特定する(S21)。このリーフブロック230の特定は、挿入された行データにおけるキーが設定された列の値を検索キーとしたインデックス検索によりリーフブロック230を特定する場合と同様の手法で実現される。図3の動作例によれば、S10、S11、S12、S13、S14及びS15の処理が実行されることにより、挿入すべきリーフブロック230が特定される。

[0068] 従って、挿入された行データに対応するリーフブロック230を特定するために読み出されたインデックスブロックのリードカウンタは、アクセス管理部110によりカウントアップされる。

[0069] 続いて、データ挿入部130は、特定されたリーフブロック230に挿入すべき新たなインデックスエントリを生成する(S22)。新たなインデックスエントリは、挿入された行データにおけるキーが設定された列の値をキー値として含み、挿入された行データを特定するための識別子を含む。

[0070] データ挿入部130は、特定されたリーフブロック230が新たなインデックスエントリの追加によりオーバーフローするか否かを判定する(S23)。ここで、オーバーフローとは、新たなインデックスエントリを追加するとブロックの空き領域が所定の比率を下回ることである。

[0071] データ挿入部130は、オーバーフローしないと判定すると(S23; NO)、その特定されたリーフブロック230にその新たに生成されたインデックスエントリを書き込む(S24)。このとき、アクセス管理部110は

、書き込まれたリーフブロック230のライトカウンタを1増加させる（S25）。

[0072] 続いて、データ挿入部130は、ブロック分割がなされたか否かを判定する（S26）。データ挿入部130は、ブロック分割がされていない場合には（S26；NO）、処理を終了し、ブロック分割がなされている場合には（S26；YES）、ブロック分割がなされたブロックの上位のブランチブロックを特定する（S27）。この上位ブランチブロックの特定は、再帰関数的な処理を施すことにより実現してもよいし、ブロックヘッダに上位ブロックの識別子を格納することで実現してもよい。

[0073] 以降、データ挿入部130は、その特定された上位ブランチブロックを対象ブロックとして処理（S22）に戻って各処理を実行する。

[0074] 一方、データ挿入部130は、オーバーフローすると判定した場合には（S23；YES）、その特定されたブロックに対してブロック分割処理を実行する。この処理において、まず、データ挿入部130は、オーバーフローすると判定されたブロックがルートブロックか否かを判定する（S30）。

[0075] データ挿入部130は、オーバーフローすると判定されたブロックがルートブロックである場合（S30；YES）、新たなルートブロックとするための新たな空きブロックを取得する（S31）。このとき、アクセス管理部110は、取得された空きブロックのリードカウンタを1増加させる（S32）。

[0076] データ挿入部130は、オーバーフローすると判定された現在のルートブロック内の最小のキー値及び現在のルートブロックを特定するための識別子を含むインデックスエントリを生成する（S33）。データ挿入部130は、このように生成されたインデックスエントリを、新たなルートブロックとするために取得された空きブロックに書き込む（S34）。言い換えれば、生成されたインデックスエントリは、新たなルートブロックから元のルートブロック（その後のブランチブロック）へリンクするためのデータとなる。

[0077] このとき、アクセス管理部110は、そのインデックスエントリが書き込

まれたブロックのライトカウンタを1増加させる（S35）。

[0078] 続いて、データ挿入部130は、ルートブロックを変更する（S36）。
具体的には、データ挿入部130は、オーバーフローすると判定された元のルートブロックから新たに取得されたブロックに、ルートブロックを変更する。例えば、この変更は、ブロックヘッダのブロック種の変更により実現される。

[0079] 次に、データ挿入部130は、空きブロックを更に取得する（S37）。
データ挿入部130は、そのオーバーフローすると判定されたブロック内の移動させるべきエントリを決定し、決定された各エントリを上記取得された空きブロックに順次移動させる（S38）。例えば、データ挿入部130は、予め閾値を保持しており、オーバーフローすると判定されたブロックのエントリのうち、当該閾値以上のキー値を持つエントリを移動エントリに決定する。その閾値には、中央値などのような、1つ以上のエントリのキー値よりも小さい値を用いる。

[0080] このとき、アクセス管理部110は、エントリを空きブロックに移すたびに、空きブロックのライトカウンタを1増加させる（S39）。

[0081] その後、データ挿入部130は、エントリの移動によりできた空き領域に処理（S22）で生成されたインデックスエントリを書き込む（S24）。
データ挿入部130及びアクセス管理部110は、上述した処理（S25）以降の各処理を実行する。

[0082] 図8は、第2実施形態におけるDBシステム10のデータ削除時の動作例を示すフローチャートである。

[0083] データ削除部140は、削除する行データを特定するための値を受け取り、データブロック240から行データを削除する（S40）。行データの削除は、行のデータを消去することで実現してもよいし、その行データに無効を示す値を付加することで実現してもよい。

[0084] データ削除部140は、削除された行データを特定するインデックスエントリを有するリーフブロック230を特定する（S41）。このリーフブ

ック230の特定は、削除された行データにおけるキーが設定された列の値を検索キーとしたインデックス検索によりリーフブロック230を特定する場合と同様の手法で実現される。図3の動作例によれば、S10、S11、S12、S13、S14及びS15の処理が実行されることにより、当該リーフブロック230が特定される。

[0085] データ削除部140は、特定されたリーフブロック230における、削除された行データに対応するインデックスエントリを削除する(S42)。このとき、アクセス管理部110は、そのインデックエントリが削除されたリーフブロック230のライトカウンタを増加させる(S43)。

[0086] データ削除部140は、エントリを削除することによりリーフブロック230がアンダーフローするか否かを判定する(S44)。データ削除部140は、アンダーフローが起きていなければ(S44; NO)、処理を終了する。

[0087] データ削除部140は、アンダーフローすると判定された場合には(S44; YES)、アンダーフローすると判定されたリーフブロック230に残るエントリを、他のリーフブロック230に移動させ、そのリーフブロック230を空きブロックにする(S45)。このとき、アクセス管理部110は、エントリを移動させた先のリーフブロック230のライトカウンタを1増加させる(S46)。

[0088] なお、移動先ブロックは、一つであってもよいし、複数あってもよく、任意の方法で選択してもよい。データ削除部140は、空きブロックとなったリーフブロック230のアクセスカウンタ(リードカウンタ及びライトカウンタ)を0に設定する。空きブロックとすることで、このブロックはデータ挿入部130によって再利用される。また、空きブロック化されたリーフブロック230のアクセスカウンタは、それぞれ移動先のブロックのアクセスカウンタに合算されてもよい。

[0089] 続いて、データ削除部140は、ブロック統合が可能か否かを判定する(S47)。具体的には、データ削除部140は、空きブロック化されたブロッ

クの親ブロックにぶら下がる全子ブロックの全エントリをその親ブロックに入れることができるか否かを判定する（S 47）。

[0090] データ削除部140は、ブロック統合が不可能と判定すると（S 47；N O）、処理を終了する。一方、データ削除部140は、ブロック統合が可能と判定すると（S 47；Y E S）、それらブロックを統合する（S 48）。つまり、子ブロックの全エントリを親ブロックに移しつつ、親ブロックに存在したエントリは削除する。結果、その親ブロックにぶら下がる全ての子ブロックを空きブロックにする。

[0091] このとき、アクセス管理部110は、上位ブロックのライトカウンタを1増加させる（S 49）。なお、統合される子ブロックのアクセスカウンタの値は、親ブロックのアクセスカウンタに合算されてもよい。

[0092] 〔第2実施形態の作用及び効果〕

このように第2実施形態では、行データがデータブロック240に挿入された場合、その行データを特定するためのインデックスエントリがリーフブロック230に追加されると共に、そのリーフブロック230のライトカウンタが更新される。

[0093] 更に、そのインデックスエントリを追加すべきリーフブロック230を特定するためのインデックス検索において参照された各インデックスブロックのリードカウンタが更新される。また、その行データ挿入により、ブロック分割が生じた場合には、新たに取得された空きブロックのリードカウンタが更新され、エントリ移動の度にそのブロックのライトカウンタが逐次更新される。

[0094] また、第2実施形態では、データブロック240から行データが削除されると、その行データに対応するインデックスエントリがリーフブロック230から削除されると共に、そのインデックスエントリが削除されたリーフブロック230のライトカウンタが更新される。更に、エントリ削除によりリーフブロック230がアンダーフローすると判定された場合には、その空きブロック化に伴い移動されたエントリの移動先のリーフブロック230のラ

イトカウンタが更新される。更に、ブロックが統合される場合においても、統合される上位ブロックのライトカウンタが更新される。

[0095] このように、第2実施形態によれば、各ブロックへのアクセス回数をリードとライトとを区別してそれぞれ管理することができる。リードアクセスとライトアクセスとは、同じ1つのアクセスでも負荷が異なる。それは、アクセスに要する時間に差があり、かつ、異なる種類のロックをかける必要があるからである。

[0096] 従って、第2実施形態によれば、各インデックスブロックについて、アクセス種別毎のアクセス回数情報をそれぞれ管理することにより、データベース性能を最適化する上で一層有用な情報を提供することができる。そのような情報によれば、各インデックスブロックにおける、アクセス負荷や子ブロックへのアクセスの偏り等を正確に見積もることができ、更に、インデックスを付加することで効率化を図れる行とそうでない行とを特定することも可能となる。

[0097] 具体的には、第2実施形態で管理される、各インデックスブロックのリードカウンタ及びライトカウンタを用いれば、行単位でインデックスを付加すべきか否かを定めることができる。ツリーインデックスが付加されると検索は高速化されるが、更新時には、インデックス操作が必要となるため、効率が低下する。これにより、ライトカウンタの数が所定数より多いインデックスブロックのエントリで特定される行にはインデックスを付与しないほうが効率がよいと判断することができる。

[0098] [第3実施形態]

第3実施形態では、各ブロックが格納される記憶領域を区別して各ブロックのアクセスカウンタを管理する。以下、第3実施形態におけるDBシステム10について、上述の各実施形態と異なる内容を中心に説明し、上述の各実施形態と同一の内容についての説明は適宜省略する。

[0099] [システム構成]

図9は、第3実施形態におけるDBシステム10の構成例を示す概念図で

ある。第3実施形態では、図9に示すように、複数の記憶領域（記憶領域A300及び記憶領域B400）が区別される。例えば、記憶領域A300は、メモリ2に含まれるアクセス速度の速いメモリ装置上で実現され、記憶領域B400は、メモリ2に含まれる記憶領域A300よりもアクセス速度の遅いメモリ装置上で実現される。

[0100] 例えば、記憶領域A300は一次記憶上に実現され、記憶領域B400はハードディスク等のような二次記憶上に実現される。なお、記憶領域の区別の仕方は、このような形態に制限されず、CPUキャッシュ、フラッシュメモリ、ハードディスク等のように記憶装置の種別毎に区別されてもよい。

[0101] 第3実施形態におけるDBシステム10は、第2実施形態の構成に加えて、記憶領域管理部150を更に有する。記憶領域管理部150についても、メモリ2に格納されるプログラムがCPU1により読み出され実行されることにより、ソフトウェア要素として実現される。

[0102] 記憶領域管理部150は、区別された各記憶領域A300及び記憶領域B400を管理する。本実施形態では、記憶領域A300がキャッシュとして扱われ、記憶領域B400には、DB部200の全てが格納される。記憶領域管理部150は、検索部120、データ挿入部130及びデータ削除部140がこれら記憶領域を区別することなく処理できるように制御する。

[0103] 記憶領域管理部150は、記憶領域B400内のDB部200の一部のブロックの複製を記憶領域A300に格納する。なお、記憶領域管理部150によるこのメモリ制御は、周知のキャッシュ技術が用いられればよいので、ここでは詳細な説明を省略する。

[0104] 図10は、第3実施形態におけるインデックスブロックの構成例を概念的に示す図である。第3実施形態におけるインデックスブロックでは、図10に示すように、アクセスカウンタとして、記憶領域毎に、ライトカウンタとリードカウンタとの組がそれぞれ設けられる。

[0105] ライトカウンタA及びリードカウンタAは、記憶領域A300に格納されたブロックのアクセス（キャッシュヒット）をカウントし、ライトカウンタ

B及びリードカウンタBは、記憶領域A300にはなかったために記憶領域B400から複製されたブロックへのアクセス（キャッシュミス）をカウントする。

[0106] アクセス管理部110は、アクセスしたブロックの記憶領域を区別し、区別された記憶領域のためのアクセスカウンタを更新する。例えば、アクセス管理部110は、記憶領域管理部150からのキャッシュミスの発生の通知に応じて、どの記憶領域に格納されたブロックにアクセスしたのかを判別する。なお、本実施形態は、メモリ装置の種別の判別方法をこのような方法に限定するものではないため、周知の他の判別方法が利用されてもよい。

[0107] [第3実施形態の作用及び効果]

このように第3実施形態では、記憶領域（メモリ装置）毎に、アクセスカウンタが設けられ、アクセス時にそのインデックスブロックが格納されていた記憶領域に対応するアクセスカウンタが更新される。

[0108] 従って、第3実施形態によれば、例えば、ハードディスクなどの二次記憶に格納されているブロックが一次記憶上に複製されてから操作される場合の、一次記憶以外の記憶媒体へのアクセスの影響を考慮できるようになる。アクセス処理に要する時間は、インデックスブロックが格納されている記憶領域のタイプに応じて異なるため、第3実施形態によれば、上述の各実施形態に比べてより正確にアクセス負荷を見積もることができる。例えば、第3実施形態では、次のようにアクセス負荷Lを算出することができる。

$$L = a_r \times a_{rc} + a_w \times a_{wc} + b_r \times b_{rc} + b_w \times b_{wc}$$

[0109] ここで、 a_r 及び a_w は、記憶領域A300上でのリード負荷及びライト負荷を示し、 a_{rc} 及び a_{wc} は、記憶領域A300上でのリードアクセス数及びライトアクセス数を示し、 b_r 及び b_w は、記憶領域B400上でのリード負荷及びライト負荷を示し、 b_{rc} 及び b_{wc} は記憶領域B400上でのリードアクセス数及びライトアクセス数を示す。

[0110] [第4実施形態]

上述の各実施形態では、ブロック毎のアクセス回数を管理する形態の例を

示したが、第4実施形態では、そのアクセス回数を利用する形態の例を示す。以下、第4実施形態におけるDBシステム10について、上述の各実施形態と異なる内容を中心に説明し、上述の各実施形態と同一の内容についての説明は適宜省略する。

[0111] [システム構成]

図11は、第4実施形態におけるDBシステム10の構成例を示す概念図である。第4実施形態におけるDBシステム10は、第2実施形態の構成に加えて、アクセス負荷評価部310、行特定部320、アクセス偏り算出部330を更に有する。これら各処理部についても、メモリ2に格納されるプログラムがCPU1により読み出され実行されることにより、ソフトウェア要素として実現される。

[0112] アクセス負荷評価部310は、任意のインデックスブロックを特定するためのデータを取得した場合に、そのデータで特定されるインデックスブロックのアクセス負荷を算出し、算出されたアクセス負荷を示す情報を出力する。アクセス負荷の算出の際に、アクセス負荷評価部310は、特定されたインデックスブロックに関する所定のリード負荷及び所定のライト負荷を取得し、かつ、そのインデックスブロックのブロックヘッダからアクセスカウンタを読み出す。

[0113] アクセス負荷評価部310は、所定のリード負荷とリードアクセス数との積、及び、所定のライト負荷とライトアクセス数との積の和を算出し、この算出された値をアクセス負荷として出力する。ここで、リード負荷及びライト負荷は、1つのリードアクセスと1つのライトアクセスの処理にかかる各時間をそれぞれ示す。アクセス処理に要する時間にはばらつきがあるが、例えば、それらの平均値や中央値などの代表値がメモリ2に格納されており、その代表値がリード負荷及びライト負荷として利用される。

[0114] 任意のインデックスブロックを特定するためのデータは、他の装置から通信を介して取得されてもよいし、CPU1により実行されているプロセス等のような他の処理部から取得されてもよいし、ユーザインタフェースを介し

てユーザにより入力されてもよい。

- [0115] アクセス負荷評価部 310 は、任意のインデックスブロックを特定するためのデータを取得することなく、インデックスブロックの識別子とインデックスブロックのアクセス負荷との組がアクセス負荷の高い順に並べられたリストデータを出力するようにしてもよい。
- [0116] 行特定部 320 は、インデックスを付加することで効率化を図れると推定される行を特定する。具体的には、行特定部 320 は、ライトアクセス数が全アクセス数に占める割合が所定値よりも高いリーフブロック 230 を特定する。行特定部 320 は、このように特定されたリーフブロック 230 に含まれる各インデックスエントリにより特定される行をインデックスを付加することで効率化を図れない行として特定する。
- [0117] 行特定部 320 は、インデックスを付加することで効率化を図れると推定される行に対する識別子のリストを出力する。また、行特定部 320 は、インデックスを付加することで効率化を図れると推定される行のみにインデックスを付加し、それ以外の行からはインデックスを削除する処理を実行するようにしてもよい。また、行特定部 320 は、ライトアクセス数が全アクセス数に占める割合をその割合が高い順に指定された数分並べたリストを出力するようにしてもよい。
- [0118] アクセス偏り算出部 330 は、アクセスに偏りがあるブランチブロック 220 を特定する。ブランチブロック 220 は、通常、複数の子ブロックを持つ。子ブロックに対するアクセスの偏りは、各子ブロックのアクセスカウンタを利用することにより把握することができる。そこで、アクセス偏り算出部 330 は、リードアクセス数、ライトアクセス数、全アクセス数に関して、特定の子ブロックに偏ってアクセスを発生させたブランチブロック 220 を偏りが大きいものから所定数分特定する。
- [0119] なお、本実施形態は、偏りの算出手法自体を限定せず、周知の様々な手法が利用されればよい。例えば、全子ブロックへのアクセスの a% が、特定の子ブロックに対するアクセスであり、当該特定の子ブロックへのアクセス数

は全子ブロックに対するアクセス数の $x\%$ ($=100 - a$)であった場合には、この偏りは、 a/x により算出されてもよい。具体例では、ブランチブロック220が10個の子ブロックを持ち、10個のうちの2個の子ブロックへのアクセスが、10個の子ブロックへのアクセスの80%を占めていた場合、偏りは、 $4 (=80/20)$ と算出される。

[0120] 偏りの算出手法としては、 a と x とのいずれか一方を固定するのが最も単純な方法であるが、($75 \leq a \leq 90$)等のように一方の値の範囲を指定して算出する手法が利用されてもよい。このような手法では、範囲指定された一方の値に対してもう一方の値を決め、求めた組に対して(a/x)を算出し、その最大値が偏りとして用いられる。

[0121] アクセス負荷評価部310、行特定部320及びアクセス偏り算出部330により出力された情報は、入出力インタフェース3を介して、他の装置(コンピュータ)に送られてもよいし、表示装置等のユーザインタフェースから出力されてもよいし、ファイル等に格納された状態でメモリ2に保存されてもよい。本実施形態は、それら情報の出力形態を限定しない。

[0122] [第4実施形態の作用及び効果]

このように第4実施形態では、上述の各実施形態において管理されるブロック毎のアクセスカウンタを利用することにより、データベースの性能の適正化を図る上で有効となる新たな情報が出力される。具体的には、アクセス負荷評価部310により、各インデックスブロックのアクセス負荷の情報が出力され、行特定部320により、インデックスを付加することで効率化を図れると推定される行の情報が出力され、アクセス偏り算出部330により、アクセスに偏りがあるブランチブロック220の情報が出力される。

[0123] 例えば、上述のような出力情報を用いれば、行単位でインデックスを付加すべきかそうでないかを定めることができる。これは、検索、挿入及び削除といったデータベースに対する全ての操作を考慮した上でのデータベース性能の最適化に繋がる。また、アクセスの偏りに関する情報やアクセス負荷情報を用いれば、インデックスブロックのツリー構造の再設計や、キャッシュ

に優先的に置くべきインデックスブロックの特定等のようなデータベース性能の最適化を行うことができる。

[0124] 更に、アクセスの偏りに関する情報を用いれば、或るブランチブロック220の次にアクセスされる確率の高い他のブランチブロック220又はリーフブロック230、即ち、連続してアクセスされる確率の高いブロック関係を特定することができる。これにより、例えば、連続してアクセスされる可能性の高い2つのインデックスブロックを連続する記憶領域に配置することにより、それらが非連続の記憶領域に配置される場合と比較して処理を高速化することができる。

[0125] 即ち、第4実施形態によれば、データベース性能の最適化を行う上での有益な情報を提供することができる。

[0126] [変形例]

上述の第2実施形態、第3実施形態及び第4実施形態では、アクセスカウンタとして、リードカウンタとライトカウンタとがそれぞれ設けられたが、上述の第2実施形態、第3実施形態及び第4実施形態においても、第1実施形態のようにリードカウンタとライトカウンタとを統合して1つのアクセスカウンタで管理されるようにしてもよい。また、上述の各実施形態では、アクセスカウンタ（リードカウンタ及びライトカウンタを含む）は1ずつ増やされたが、この増加幅は、アクセス種別等に応じて適宜変えてもよい。

[0127] また、上述の各実施形態では、ブランチブロック220内のインデックスエントリに含まれる各エントリは、いずれか1つの他のブランチブロック220又はいずれか1つのリーフブロック230を特定するための1つの識別子を含む例を示したが、複数の識別子が含まれるようにしてもよい。例えば、ブランチブロック220内の最小のキー値を持つエントリについては、そのキー値より小さいキー値に対応する子ブロックを特定するための識別子と、そのキー値以上で他のエントリのキー値よりも小さいキー値に対応する子ブロックを特定するための識別子とを含むようにしてもよい。

[0128] なお、上記各実施形態の説明は、複数のフローチャートを用いており、そ

れぞれに複数のステップ（処理）を順番に記載しているが、本実施形態は、各ステップの順番を図示される順番に限定するものではない。本実施形態では、図示される処理ステップの順番を内容的に支障しない範囲で変更することができる。また、上述した各実施形態及び変形例は、その内容が相反しない範囲で組み合わせることができる。

[0129] 上記の各実施形態及び変形例の一部又は全部は、以下の付記のようにも記載され得る。但し、各実施形態及び各変形例が以下の記載に限定されるものではない。

[0130] （付記1）表データを格納するデータブロックを含むデータベースを管理するデータベース管理装置において、表データを構成する1つの行データ又は他のインデックスブロックを特定するための少なくとも1つのインデックスエントリ、及び、アクセスカウンタをそれぞれ有し、ツリー構造を持つ複数のインデックスブロックと、表データに対するデータ操作に応じて各インデックスエントリを参照することによりツリー構造に沿ってアクセスされた複数のインデックスブロックの各アクセスカウンタをそれぞれ更新するアクセス管理手段と、を備えることを特徴とするデータベース管理装置。

[0131] （付記2）上記各インデックスブロック内のインデックスエントリを上記ツリー構造に沿って検索することにより、上記データブロック内の少なくとも1つの行データを抽出する検索手段、を更に備え、上記アクセス管理手段は、上記検索手段が上記複数のインデックスブロックのいずれか1つにアクセスする際に、そのアクセスされたインデックスブロックのアクセスカウンタを増加させ、続いて、そのアクセスされたインデックスブロック内のインデックスエントリにより特定される他のインデックスブロックにアクセスする際に、当該他のインデックスブロックのアクセスカウンタを増加させる、ことを特徴とする付記1に記載のデータベース管理装置。

[0132] （付記3）上記データブロックに新たな行データを挿入すると共に、上記データブロックに挿入された行データを特定する新たなインデックスエントリを上記複数のインデックスブロックのうちの挿入対象のインデックスブロ

ックに挿入し、この挿入対象のインデックスブロックに当該新たなインデックスエントリを挿入することができない場合には空のインデックスブロックを取得するデータ挿入手段、を更に備え、上記アクセスカウンタは、リードカウンタとライトカウンタとを含み、上記アクセス管理手段は、上記データ挿入手段が上記インデックスブロックに上記新たなインデックスエントリを挿入する際には、上記インデックスブロックのライトカウンタを増加させ、上記データ挿入手段が上記空のインデックスブロックを取得する際には、上記空のインデックスブロックのリードカウンタを増加させる、ことを特徴とする付記 1 又は 2 に記載のデータベース管理装置。

[0133] (付記 4) 上記データ挿入手段は、上記挿入対象のインデックスブロック内のインデックスエントリの少なくとも 1 つを上記空のインデックスブロックに移動させると共に、上記新たなインデックスエントリを上記挿入対象のインデックスブロックに挿入し、上記アクセス管理手段は、上記データ挿入手段が上記少なくとも 1 つのインデックスエントリを上記空のインデックスブロックに移動させる場合に、上記空のインデックスブロックのライトカウンタを増加させる、ことを特徴とする付記 3 に記載のデータベース管理装置。

[0134] (付記 5) 上記データブロックから行データを削除すると共に、削除された行データに対応するインデックスエントリを上記複数のインデックスブロックのうちの削除対象のインデックスブロックから削除し、上記削除対象のインデックスブロックを空きブロックにする場合には、上記削除対象のインデックスブロックに残っているインデックスエントリを他のインデックスブロックに移動させるデータ削除手段、を更に備え、上記アクセス管理手段は、上記データ削除手段が上記削除対象のインデックスブロックから上記インデックスエントリを削除した場合には、上記削除対象のインデックスブロックのライトカウンタを増加させ、上記データ削除手段が上記削除対象のインデックスブロックを空きブロックにする場合には、上記他のインデックスブロックのライトカウンタを増加させる、ことを特徴とする付記 3 又は 4 に記

載のデータベース管理装置。

- [0135] (付記6) 上記複数のインデックスブロックのうちの指定インデックスブロックに関するリード負荷情報及びライト負荷情報を取得し、その指定インデックスブロックのリードカウンタとそのリード負荷情報との積、及び、その指定インデックスブロックのライトカウンタとそのライト負荷情報との積の和を、その指定インデックスブロックのアクセス負荷として算出するアクセス負荷評価手段、を更に備えることを特徴とする付記3から5のいずれか1つに記載のデータベース管理装置。
- [0136] (付記7) ライトカウンタが全アクセス数に占める割合が所定値よりも高いインデックスブロックを特定する行特定手段を更に備えることを特徴とする付記3から6のいずれか1つに記載のデータベース管理装置。
- [0137] (付記8) 複数の子インデックスブロックを特定する複数のインデックスブロックをそれぞれ格納する各親インデックスブロックに関し、当該複数の子インデックスブロックのアクセスカウンタに基づいて、全子インデックスブロックへのアクセス回数に対する特定の子インデックスブロックへのアクセス回数の偏りをそれぞれ算出し、偏りの大きいものから所定数分の親インデックスブロックを特定するための情報を出力するアクセス偏り算出手段、を更に備えることを特徴とする付記1から7のいずれか1つに記載のデータベース管理装置。
- [0138] (付記9) 上記アクセスカウンタは、複数の記憶領域のそれぞれに対応して複数存在し、上記アクセス管理手段は、上記複数のインデックスブロックのいずれか1つへのアクセスが発生した場合に、アクセスされたインデックスブロックが有するアクセスカウンタであってアクセス先の記憶領域に対応するアクセスカウンタを更新する、ことを特徴とする付記1から8のいずれか1つに記載のデータベース管理装置。
- [0139] (付記10) 表データを格納するデータブロックを含むデータベースを管理するプログラムにおいて、コンピュータを、上記表データを構成する1つの行データ又は他のインデックスブロックを特定するための少なくとも1つ

のインデックスエントリ、及び、アクセスカウンタをそれぞれ有し、ツリー構造を持つ複数のインデックスブロックと、上記表データに対するデータ操作に応じて各インデックスエントリを参照することにより上記ツリー構造に沿ってアクセスされた複数のインデックスブロックの各アクセスカウンタをそれぞれ更新するアクセス管理手段と、して機能させるためのプログラム。

[0140] (付記 1 1) 上記コンピュータを、上記各インデックスブロック内のインデックスエントリを上記ツリー構造に沿って検索することにより、上記データブロック内の少なくとも 1 つの行データを抽出する検索手段、として更に機能させ、上記アクセス管理手段は、上記検索手段が上記複数のインデックスブロックのいずれか 1 つにアクセスする際に、そのアクセスされたインデックスブロックのアクセスカウンタを増加させ、続いて、そのアクセスされたインデックスブロック内のインデックスエントリにより特定される他のインデックスブロックにアクセスする際に、当該他のインデックスブロックのアクセスカウンタを増加させる、ことを特徴とする付記 1 0 に記載のプログラム。

[0141] (付記 1 2) 上記コンピュータを、上記データブロックに新たな行データを挿入すると共に、上記データブロックに挿入された行データを特定する新たなインデックスエントリを上記複数のインデックスブロックのうちの挿入対象のインデックスブロックに挿入し、その挿入対象のインデックスブロックに当該新たなインデックスエントリを挿入することができない場合には空のインデックスブロックを取得するデータ挿入手段、として更に機能させ、上記アクセスカウンタは、リードカウンタとライトカウンタとを含み、上記アクセス管理手段は、上記データ挿入手段が上記インデックスブロックに上記新たなインデックスエントリを挿入する際には、上記インデックスブロックのライトカウンタを増加させ、上記データ挿入手段が上記空のインデックスブロックを取得する際には、上記空のインデックスブロックのリードカウンタを増加させる、ことを特徴とする付記 1 0 又は 1 1 に記載のプログラム。

- [0142] (付記 1 3) 上記データ挿入手段は、上記挿入対象のインデックスブロック内のインデックスエントリの少なくとも 1 つを上記空のインデックスブロックに移動させると共に、上記新たなインデックスエントリを上記挿入対象のインデックスブロックに挿入し、上記アクセス管理手段は、上記データ挿入手段が上記少なくとも 1 つのインデックスエントリを上記空のインデックスブロックに移動させる場合に、上記空のインデックスブロックのライトカウンタを増加させる、ことを特徴とする付記 1 2 に記載のプログラム。
- [0143] (付記 1 4) 上記コンピュータを、上記データブロックから行データを削除すると共に、削除された行データに対応するインデックスエントリを上記複数のインデックスブロックのうちの削除対象のインデックスブロックから削除し、上記削除対象のインデックスブロックを空きブロックにする場合には、上記削除対象のインデックスブロックに残っているインデックスエントリを他のインデックスブロックに移動させるデータ削除手段、として更に機能させ、上記アクセス管理手段は、上記データ削除手段が上記削除対象のインデックスブロックから上記インデックスエントリを削除した場合には、上記削除対象のインデックスブロックのライトカウンタを増加させ、上記データ削除手段が上記削除対象のインデックスブロックを空きブロックにする場合には、上記他のインデックスブロックのライトカウンタを増加させる、ことを特徴とする付記 1 2 又は 1 3 に記載のプログラム。
- [0144] (付記 1 5) 上記コンピュータを、上記複数のインデックスブロックのうちの指定インデックスブロックに関するリード負荷情報及びライト負荷情報を取得し、その指定インデックスブロックのリードカウンタとそのリード負荷情報との積、及び、その指定インデックスブロックのライトカウンタとそのライト負荷情報との積の和を、その指定インデックスブロックのアクセス負荷として算出するアクセス負荷評価手段、として更に機能させることを特徴とする付記 1 2 から 1 4 のいずれか 1 つに記載のプログラム。
- [0145] (付記 1 6) 上記コンピュータを、ライトカウンタが全アクセス数に占める割合が所定値よりも高いインデックスブロックを特定する行特定手段、と

して更に機能させることを特徴とする付記 12 から 15 のいずれか 1 つに記載のプログラム。

[0146] (付記 17) 複数の子インデックスブロックを特定する複数のインデックスブロックをそれぞれ格納する各親インデックスブロックに関し、当該複数の子インデックスブロックのアクセスカウンタに基づいて、全子インデックスブロックへのアクセス回数に対する特定の子インデックスブロックへのアクセス回数の偏りをそれぞれ算出し、偏りの大きいものから所定数分の親インデックスブロックを特定するための情報を出力するアクセス偏り算出手段、として更に機能させることを特徴とする付記 10 から 16 のいずれか 1 つに記載のプログラム。

[0147] (付記 18) 上記アクセスカウンタは、複数の記憶領域のそれぞれに対応して複数存在し、上記アクセス管理手段は、上記複数のインデックスブロックのいずれか 1 つへのアクセスが発生した場合に、アクセスされたインデックスブロックが有するアクセスカウンタであってアクセス先の記憶領域に対応するアクセスカウンタを更新する、ことを特徴とする付記 10 から 17 のいずれか 1 つに記載のプログラム。

[0148] (付記 19) 表データを格納するデータブロックを含むデータベースを管理するデータベース管理方法において、上記表データを構成する 1 つの行データ又は他のインデックスブロックを特定するための少なくとも 1 つのインデックスエントリ、及び、アクセスカウンタをそれぞれ有し、ツリー構造を持つ複数のインデックスブロックを備えるコンピュータが、上記表データに対するデータ操作に応じて各インデックスエントリを参照し、上記参照により上記ツリー構造に沿ってアクセスされた複数のインデックスブロックの各アクセスカウンタをそれぞれ更新する、ことを特徴とするデータベース管理方法。

[0149] (付記 20) 付記 10 から 18 のいずれか 1 つに記載のプログラムを記録したコンピュータが読み取り可能な記録媒体。

[0150] この出願は、2010年12月16日に提出された日本出願特願2010

－ 2 8 0 3 9 7 号を基礎とする優先権を主張し、その開示の全てをここに取り込む。

請求の範囲

- [請求項1] 表データを格納するデータブロックを含むデータベースを管理するデータベース管理装置において、
- 前記表データを構成する1つの行データ又は他のインデックスブロックを特定するための少なくとも1つのインデックスエントリ、及び、アクセスカウンタをそれぞれ有し、ツリー構造を持つ複数のインデックスブロックと、
- 前記表データに対するデータ操作に応じて各インデックスエントリを参照することにより前記ツリー構造に沿ってアクセスされた複数のインデックスブロックの各アクセスカウンタをそれぞれ更新するアクセス管理手段と、
- を備えることを特徴とするデータベース管理装置。
- [請求項2] 前記各インデックスブロック内のインデックスエントリを前記ツリー構造に沿って検索することにより、前記データブロック内の少なくとも1つの行データを抽出する検索手段、
- を更に備え、
- 前記アクセス管理手段は、前記検索手段が前記複数のインデックスブロックのいずれか1つにアクセスする際に、該アクセスされたインデックスブロックのアクセスカウンタを増加させ、続いて、該アクセスされたインデックスブロック内のインデックスエントリにより特定される他のインデックスブロックにアクセスする際に、該他のインデックスブロックのアクセスカウンタを増加させる、
- ことを特徴とする請求項1に記載のデータベース管理装置。
- [請求項3] 前記データブロックに新たな行データを挿入すると共に、前記データブロックに挿入された行データを特定する新たなインデックスエントリを前記複数のインデックスブロックのうちの挿入対象のインデックスブロックに挿入し、該挿入対象のインデックスブロックに該新たなインデックスエントリを挿入することができない場合には空のイン

デックスブロックを取得するデータ挿入手段、

を更に備え、

前記アクセスカウンタは、リードカウンタとライトカウンタとを含み、

前記アクセス管理手段は、前記データ挿入手段が前記インデックスブロックに前記新たなインデックスエントリを挿入する際には、前記インデックスブロックのライトカウンタを増加させ、前記データ挿入手段が前記空のインデックスブロックを取得する際には、前記空のインデックスブロックのリードカウンタを増加させる、

ことを特徴とする請求項1又は2に記載のデータベース管理装置。

[請求項4]

前記データブロックから行データを削除すると共に、削除された行データに対応するインデックスエントリを前記複数のインデックスブロックのうちの削除対象のインデックスブロックから削除し、前記削除対象のインデックスブロックを空きブロックにする場合には、前記削除対象のインデックスブロックに残っているインデックスエントリを他のインデックスブロックに移動させるデータ削除手段、

を更に備え、

前記アクセス管理手段は、前記データ削除手段が前記削除対象のインデックスブロックから前記インデックスエントリを削除した場合には、前記削除対象のインデックスブロックのライトカウンタを増加させ、前記データ削除手段が前記削除対象のインデックスブロックを空きブロックにする場合には、前記他のインデックスブロックのライトカウンタを増加させる、

ことを特徴とする請求項3に記載のデータベース管理装置。

[請求項5]

前記複数のインデックスブロックのうちの指定インデックスブロックに関するリード負荷情報及びライト負荷情報を取得し、該指定インデックスブロックのリードカウンタと該リード負荷情報との積、及び、該指定インデックスブロックのライトカウンタと該ライト負荷情報

との積の和を、該指定インデックスブロックのアクセス負荷として算出するアクセス負荷評価手段、

を更に備えることを特徴とする請求項3又は4に記載のデータベース管理装置。

[請求項6] ライトカウンタが全アクセス数に占める割合が所定値よりも高いインデックスブロックを特定する行特定手段、

を更に備えることを特徴とする請求項3から5のいずれか1項に記載のデータベース管理装置。

[請求項7] 複数の子インデックスブロックを特定する複数のインデックスブロックをそれぞれ格納する各親インデックスブロックに関し、該複数の子インデックスブロックのアクセスカウンタに基づいて、全子インデックスブロックへのアクセス回数に対する特定の子インデックスブロックへのアクセス回数の偏りをそれぞれ算出し、偏りの大きいものから所定数分の親インデックスブロックを特定するための情報を出力するアクセス偏り算出手段、

を更に備えることを特徴とする請求項1から6のいずれか1項に記載のデータベース管理装置。

[請求項8] 前記アクセスカウンタは、複数の記憶領域のそれぞれに対応して複数存在し、

前記アクセス管理手段は、前記複数のインデックスブロックのいずれか1つへのアクセスが発生した場合に、アクセスされたインデックスブロックが有するアクセスカウンタであってアクセス先の記憶領域に対応するアクセスカウンタを更新する、

ことを特徴とする請求項1から7のいずれか1項に記載のデータベース管理装置。

[請求項9] 表データを格納するデータブロックを含むデータベースを管理するプログラムにおいて、

コンピュータを、

前記表データを構成する1つの行データ又は他のインデックスブロックを特定するための少なくとも1つのインデックスエントリ、及び、アクセスカウンタをそれぞれ有し、ツリー構造を持つ複数のインデックスブロックと、

前記表データに対するデータ操作に応じて各インデックスエントリを参照することにより前記ツリー構造に沿ってアクセスされた複数のインデックスブロックの各アクセスカウンタをそれぞれ更新するアクセス管理手段、

として機能させるためのプログラム。

[請求項10]

表データを格納するデータブロックを含むデータベースを管理するデータベース管理方法において、

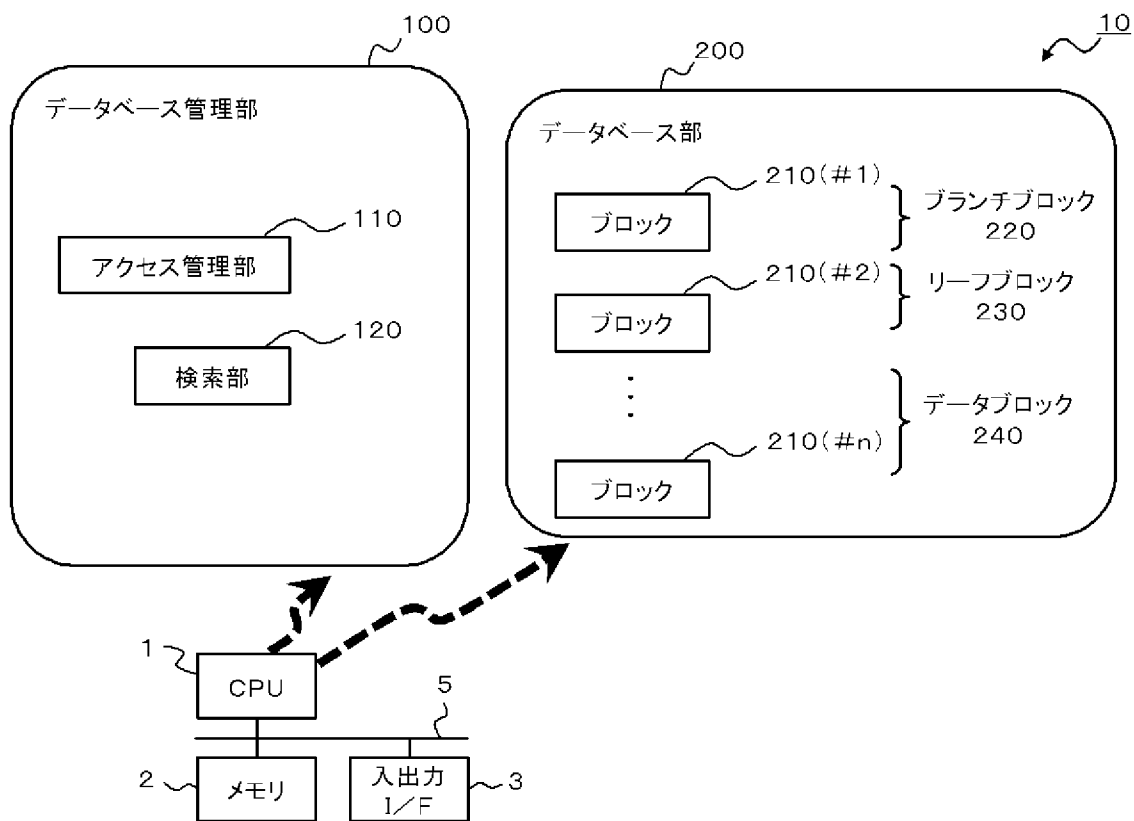
前記表データを構成する1つの行データ又は他のインデックスブロックを特定するための少なくとも1つのインデックスエントリ、及び、アクセスカウンタをそれぞれ有し、ツリー構造を持つ複数のインデックスブロックを備えるコンピュータが、

前記表データに対するデータ操作に応じて各インデックスエントリを参照し、

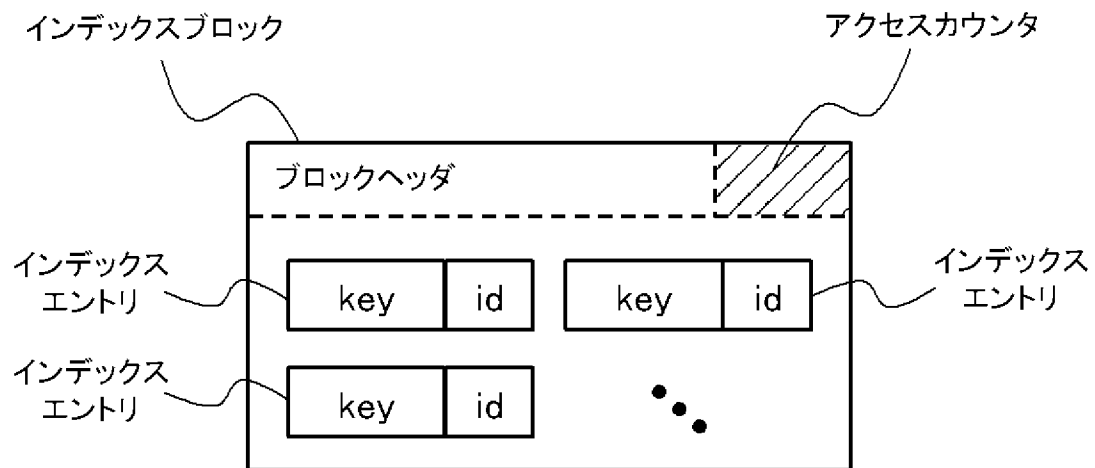
前記参照により前記ツリー構造に沿ってアクセスされた複数のインデックスブロックの各アクセスカウンタをそれぞれ更新する、

ことを特徴とするデータベース管理方法。

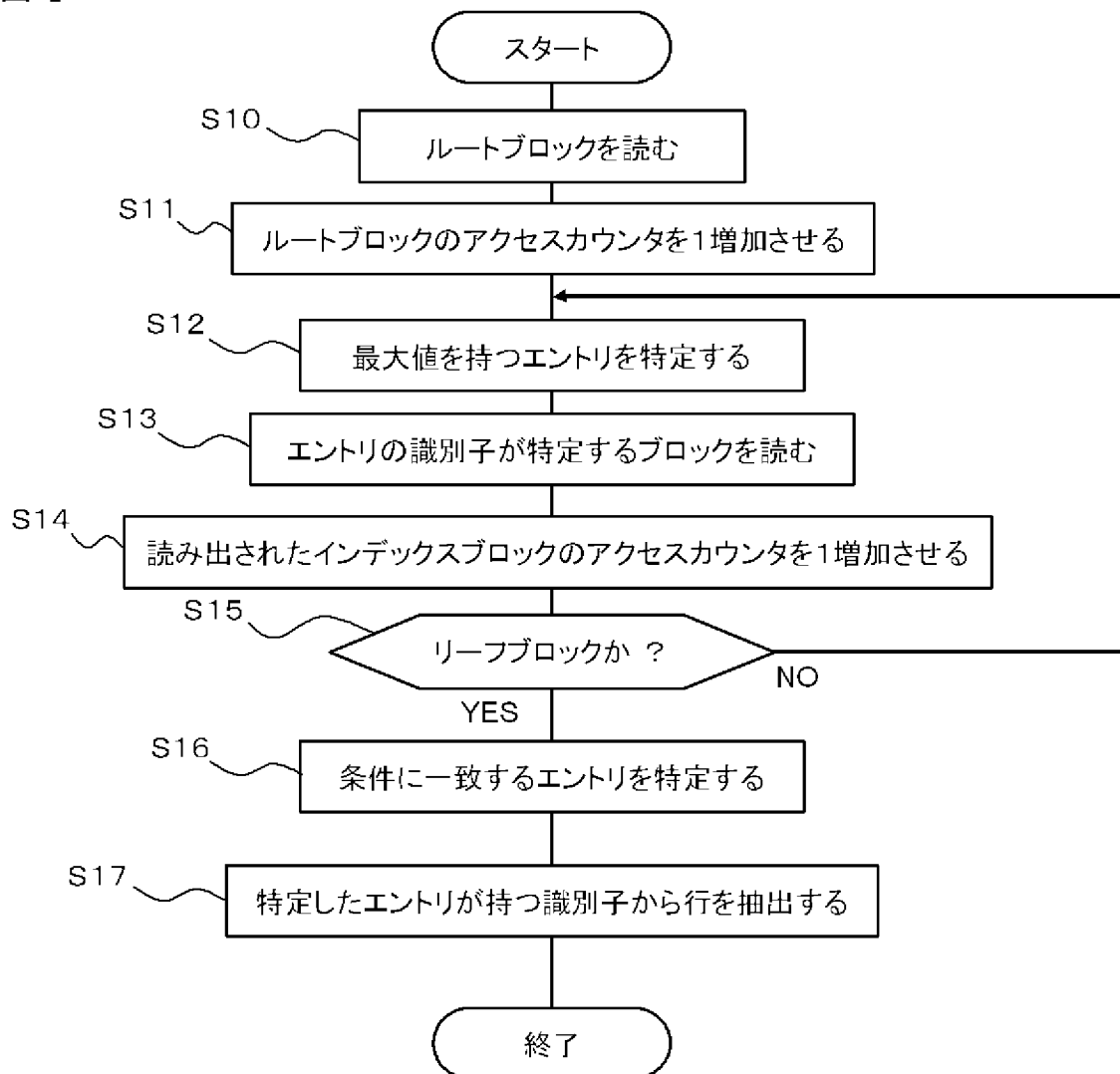
[図1]



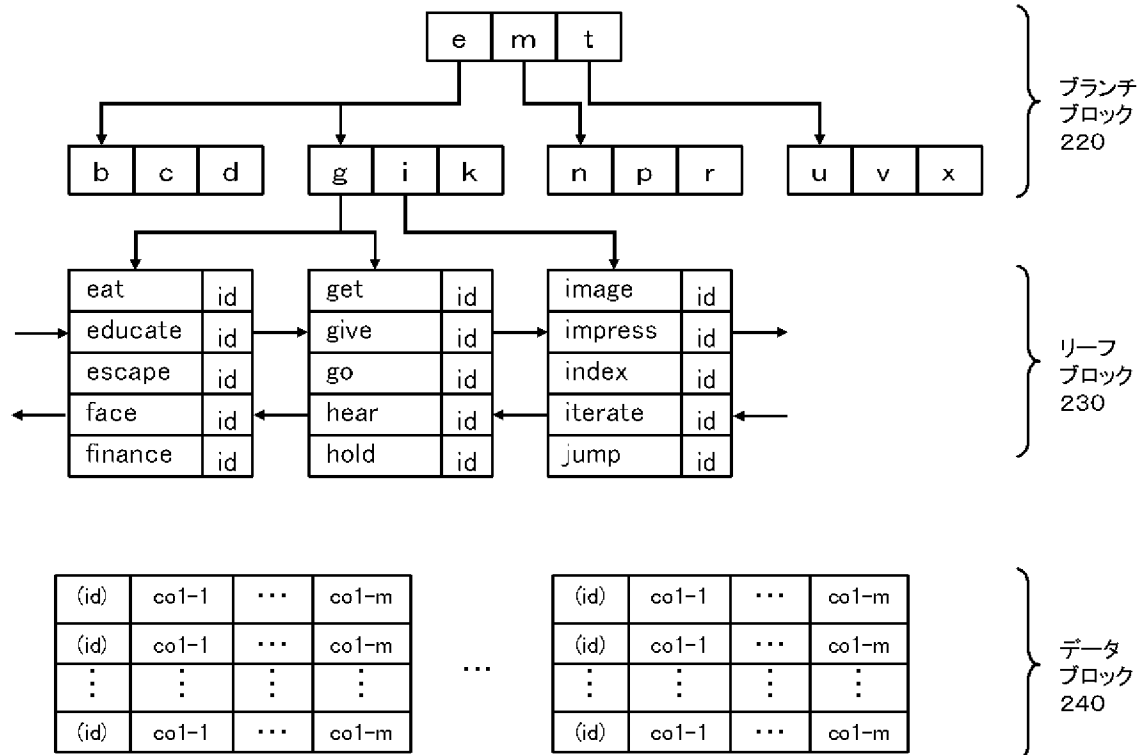
[図2]



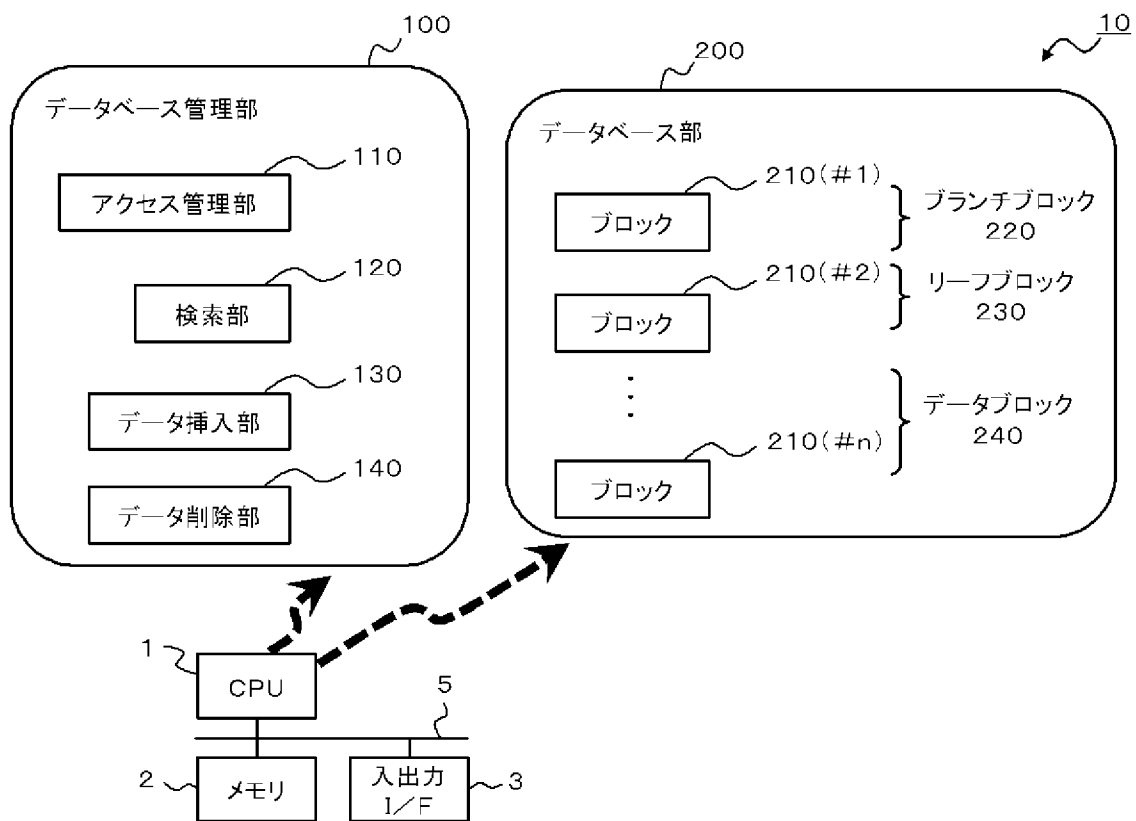
[図3]



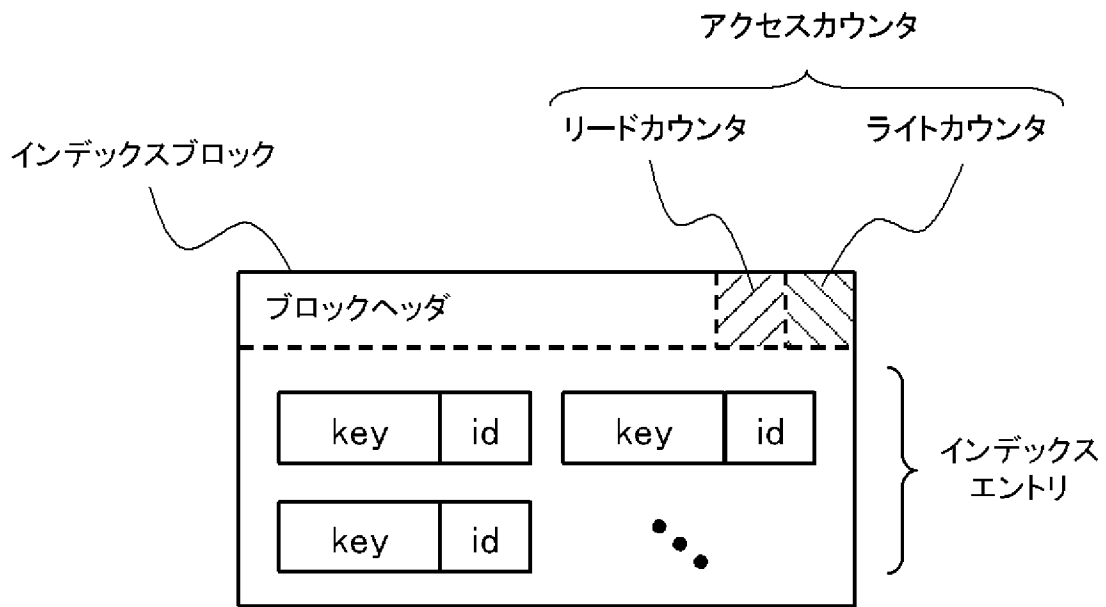
[図4]



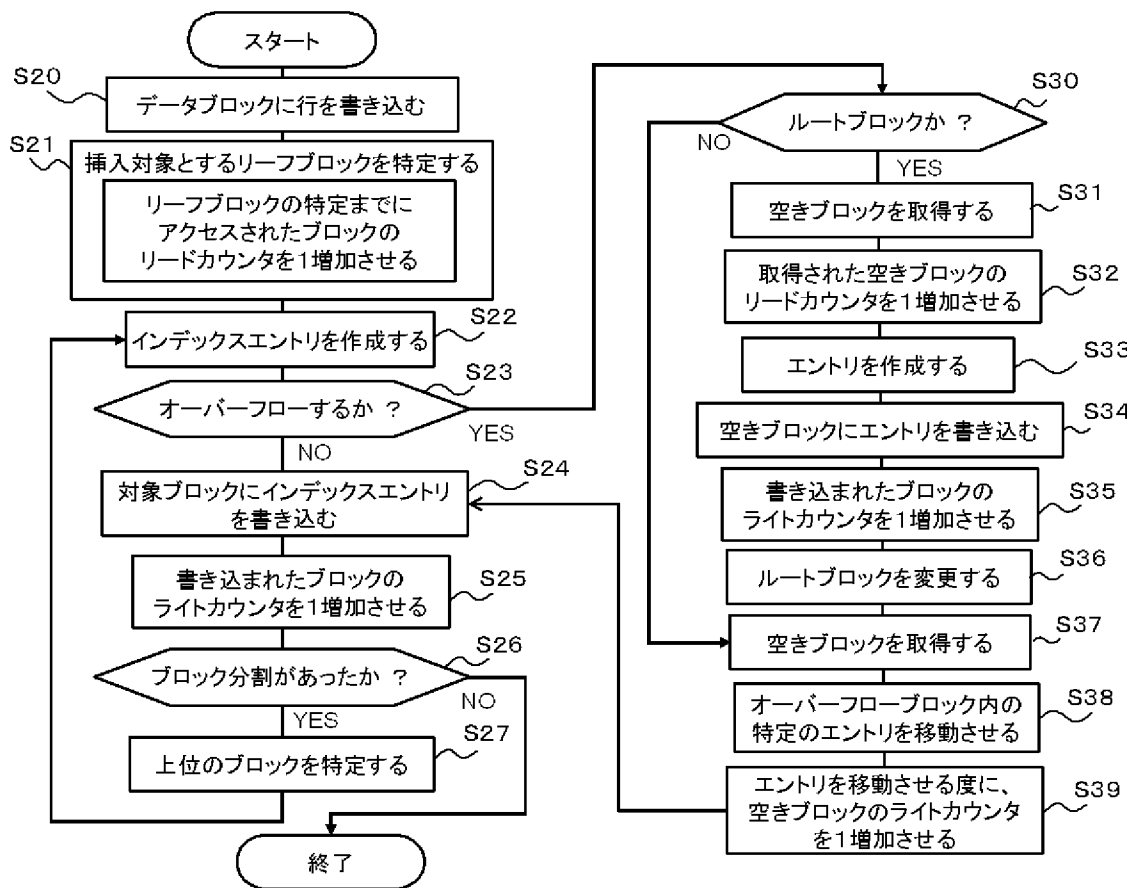
[図5]



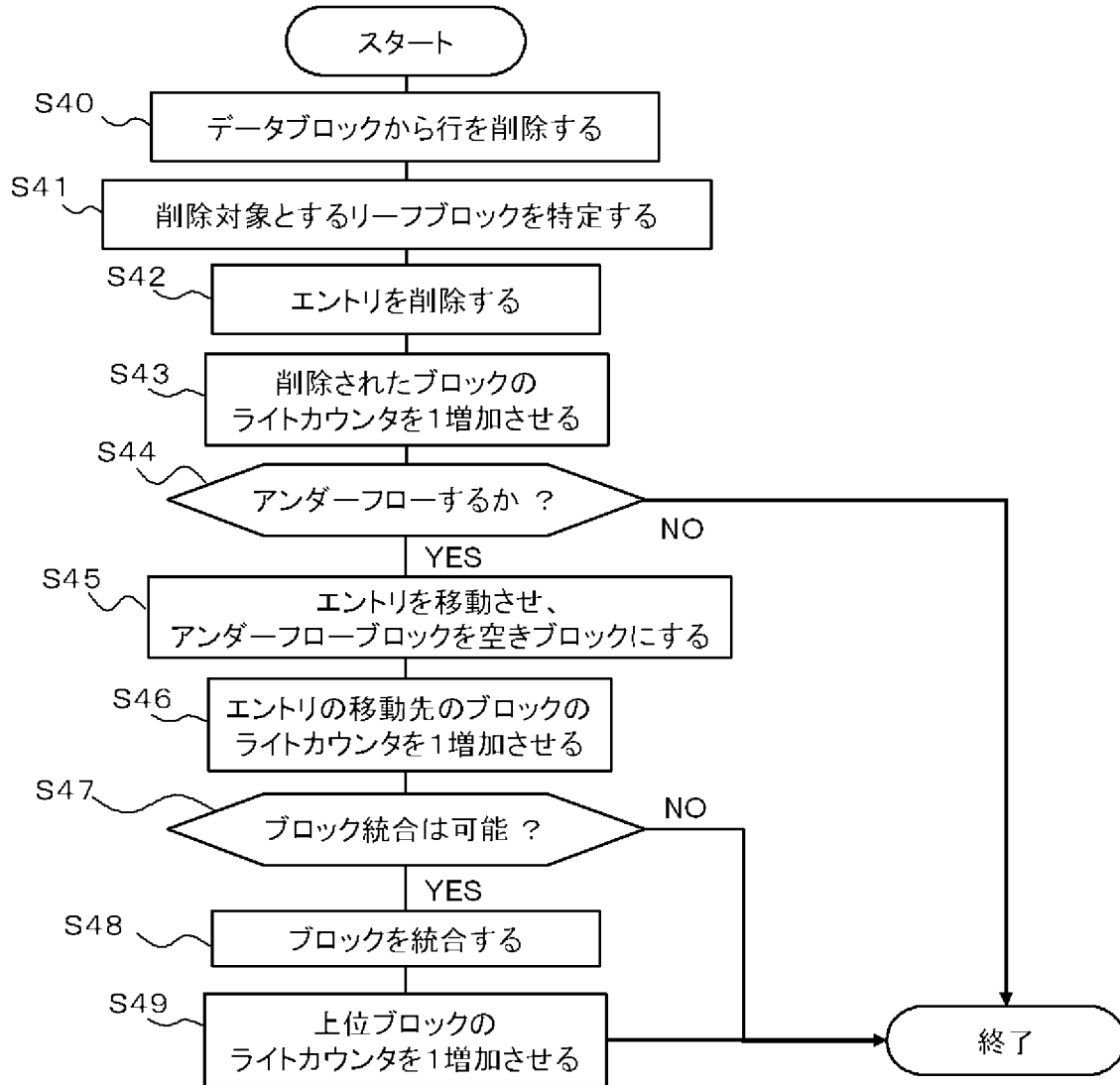
[図6]



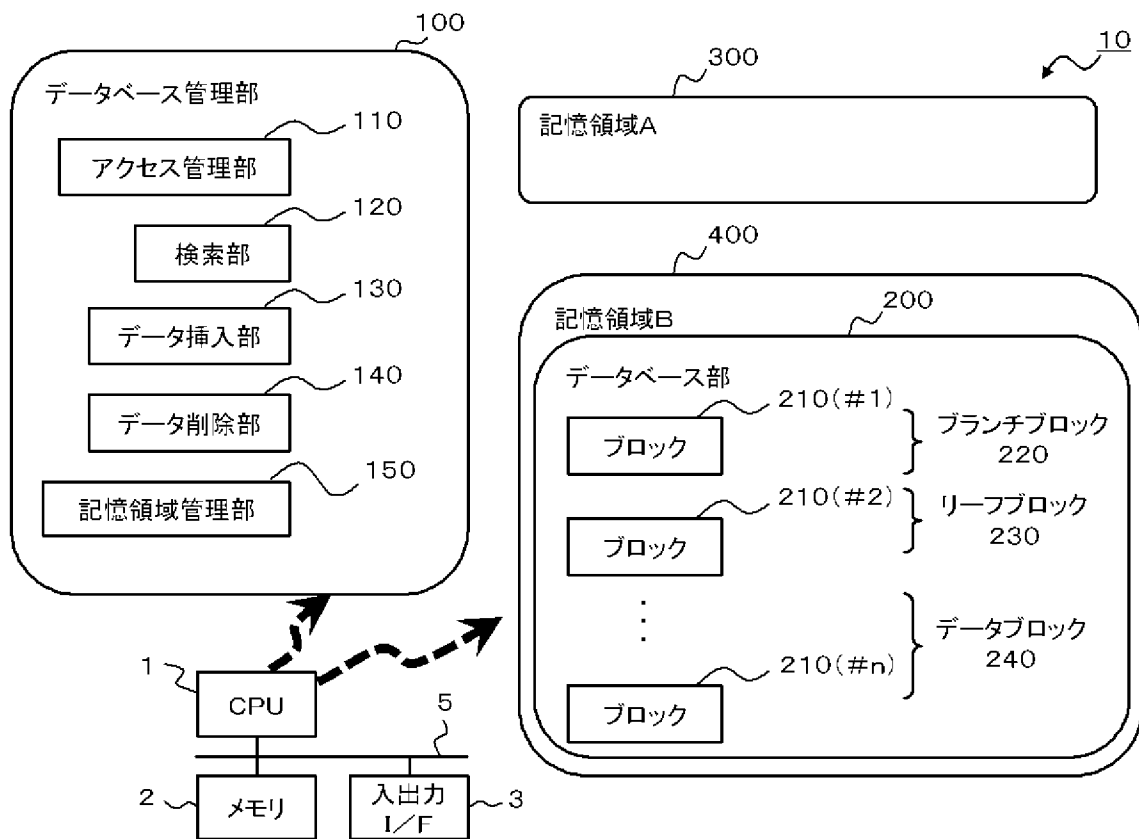
[図7]



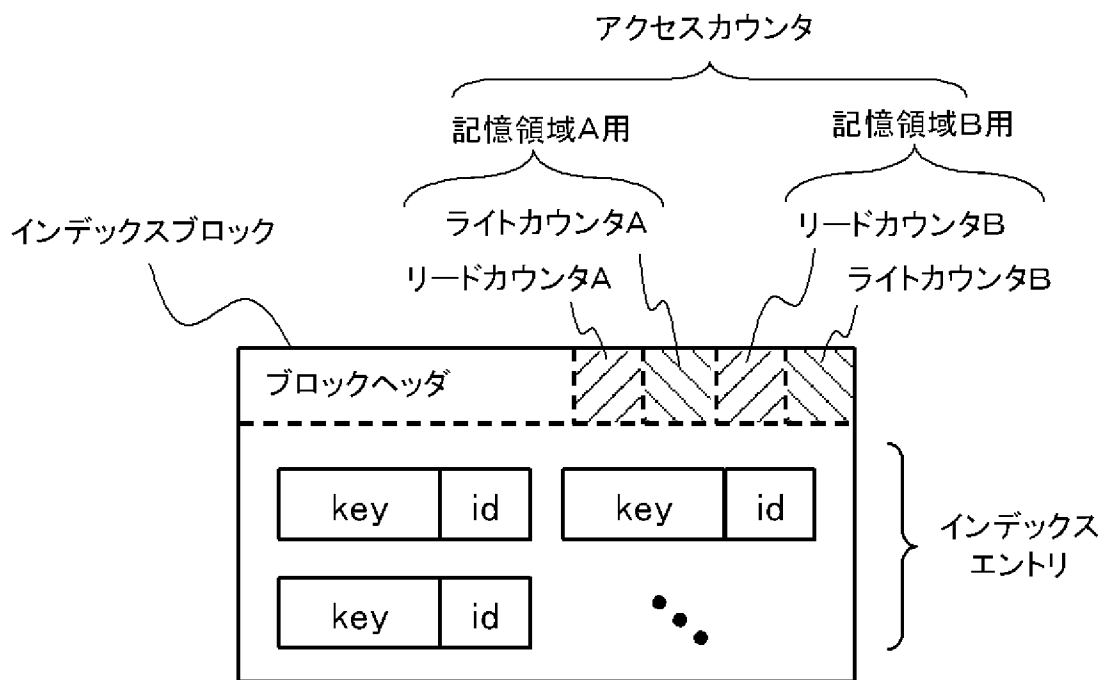
[図8]



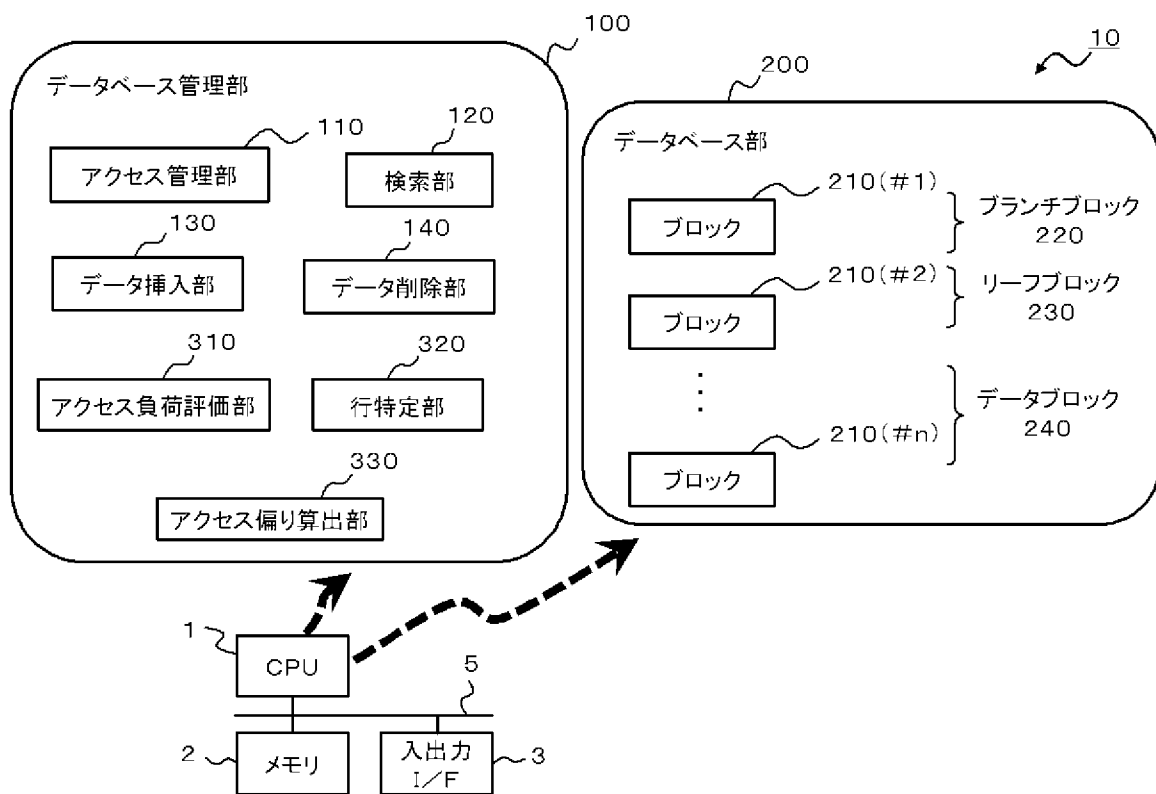
[図9]



[図10]



[図11]



INTERNATIONAL SEARCH REPORT

International application No.

PCT/JP2011/006221

A. CLASSIFICATION OF SUBJECT MATTER

G06F17/30(2006.01)i, G06F12/00(2006.01)i

According to International Patent Classification (IPC) or to both national classification and IPC

B. FIELDS SEARCHED

Minimum documentation searched (classification system followed by classification symbols)

G06F17/30, G06F12/00

Documentation searched other than minimum documentation to the extent that such documents are included in the fields searched

| | | | |
|---------------------------|-----------|----------------------------|-----------|
| Jitsuyo Shinan Koho | 1922-1996 | Jitsuyo Shinan Toroku Koho | 1996-2011 |
| Kokai Jitsuyo Shinan Koho | 1971-2011 | Toroku Jitsuyo Shinan Koho | 1994-2011 |

Electronic data base consulted during the international search (name of data base and, where practicable, search terms used)

C. DOCUMENTS CONSIDERED TO BE RELEVANT

| Category* | Citation of document, with indication, where appropriate, of the relevant passages | Relevant to claim No. |
|-----------|--|-----------------------|
| Y A | Oracle Database Gaiyo 10g Release 2(10.2) Buhin Bango: B19215-02 [online], ORACLE, 2006.03, p.5-25~5-27, [retrieval date 02 December 2011 (02.12.2011)], Internet<URL:http://otndnld.oracle.co.jp/document/products/oracle10g/102/doc_cd/server.102/B19215-02.pdf> | 1-2, 8-10 3-7 |
| Y A | JP 2003-345832 A (NEC Corp.), 05 December 2003 (05.12.2003), entire text; all drawings (Family: none) | 1-2, 8-10 3-7 |
| A | JP 5-334153 A (Nippon Telegraph and Telephone Corp.), 17 December 1993 (17.12.1993), entire text; all drawings (Family: none) | 1-10 |

Further documents are listed in the continuation of Box C.

See patent family annex.

* Special categories of cited documents:

“A” document defining the general state of the art which is not considered to be of particular relevance

“E” earlier application or patent but published on or after the international filing date

“L” document which may throw doubts on priority claim(s) or which is cited to establish the publication date of another citation or other special reason (as specified)

“O” document referring to an oral disclosure, use, exhibition or other means

“P” document published prior to the international filing date but later than the priority date claimed

“T” later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention

“X” document of particular relevance; the claimed invention cannot be considered novel or cannot be considered to involve an inventive step when the document is taken alone

“Y” document of particular relevance; the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents, such combination being obvious to a person skilled in the art

“&” document member of the same patent family

Date of the actual completion of the international search
02 December, 2011 (02.12.11)

Date of mailing of the international search report
13 December, 2011 (13.12.11)

Name and mailing address of the ISA/
Japanese Patent Office

Authorized officer

Facsimile No.

Telephone No.

A. 発明の属する分野の分類 (国際特許分類 (IPC))

Int.Cl. G06F17/30(2006.01)i, G06F12/00(2006.01)i

B. 調査を行った分野

調査を行った最小限資料 (国際特許分類 (IPC))

Int.Cl. G06F17/30, G06F12/00

最小限資料以外の資料で調査を行った分野に含まれるもの

| | |
|-------------|------------|
| 日本国実用新案公報 | 1922-1996年 |
| 日本国公開実用新案公報 | 1971-2011年 |
| 日本国実用新案登録公報 | 1996-2011年 |
| 日本国登録実用新案公報 | 1994-2011年 |

国際調査で使用した電子データベース (データベースの名称、調査に使用した用語)

C. 関連すると認められる文献

| 引用文献の カテゴリー* | 引用文献名 及び一部の箇所が関連するときは、その関連する箇所の表示 | 関連する 請求項の番号 |
|-----------------|---|----------------|
| Y | Oracle Database 概要 10g リリース 2(10.2) 部品番号: B19215-02 [online], ORACLE, 2006.03, p. 5-25~5-27, [検索日 | 1-2, 8-10 |
| A | 2011.12.02], インターネット< URL:http://otndnld.oracle.co.jp/document/products/oracle10g/102/doc_cd/server.102/B19215-02.pdf> | 3-7 |
| Y | JP 2003-345832 A (日本電気株式会社) 2003.12.05, 全文, 全図 (ファミリーなし) | 1-2, 8-10 |
| A | | 3-7 |

C欄の続きにも文献が列挙されている。

パテントファミリーに関する別紙を参照。

* 引用文献のカテゴリー

「A」特に関連のある文献ではなく、一般的な技術水準を示すもの
 「E」国際出願日前の出願または特許であるが、国際出願日以後に公表されたもの
 「L」優先権主張に疑義を提起する文献又は他の文献の発行日若しくは他の特別な理由を確立するために引用する文献 (理由を付す)
 「O」口頭による開示、使用、展示等に言及する文献
 「P」国際出願日前で、かつ優先権の主張の基礎となる出願

の日の後に公表された文献
 「T」国際出願日又は優先日後に公表された文献であって出願と矛盾するものではなく、発明の原理又は理論の理解のために引用するもの
 「X」特に関連のある文献であって、当該文献のみで発明の新規性又は進歩性がないと考えられるもの
 「Y」特に関連のある文献であって、当該文献と他の1以上の文献との、当業者にとって自明である組合せによって進歩性がないと考えられるもの
 「&」同一パテントファミリー文献

国際調査を完了した日

02.12.2011

国際調査報告の発送日

13.12.2011

国際調査機関の名称及びあて先

日本国特許庁 (ISA/JP)
 郵便番号100-8915
 東京都千代田区霞が関三丁目4番3号

特許庁審査官 (権限のある職員)

鈴木 和樹

5M

3252

電話番号 03-3581-1101 内線 3599

| C (続き) . 関連すると認められる文献 | | |
|-----------------------|---|----------------|
| 引用文献の カテゴリー* | 引用文献名 及び一部の箇所が関連するときは、その関連する箇所の表示 | 関連する 請求項の番号 |
| A | JP 5-334153 A (日本電信電話株式会社) 1993. 12. 17, 全文, 全図 (ファミリーなし) | 1-10 |