

(19) 日本国特許庁(JP)

(12) 特 許 公 報(B2)

(11) 特許番号

特許第3914662号

(P3914662)

(45) 発行日 平成19年5月16日(2007.5.16)

(24) 登録日 平成19年2月9日(2007.2.9)

(51) Int. Cl.

F I

G O 6 F 17/30 (2006.01)

G O 6 F 17/30 3 4 O D

G O 6 F 12/00 (2006.01)

G O 6 F 17/30 1 8 O D

G O 6 F 12/00 5 1 2

G O 6 F 12/00 5 2 O A

請求項の数 4 (全 17 頁)

(21) 出願番号	特願平11-184746	(73) 特許権者	000005108
(22) 出願日	平成11年6月30日(1999.6.30)		株式会社日立製作所
(65) 公開番号	特開2001-14329(P2001-14329A)		東京都千代田区丸の内一丁目6番6号
(43) 公開日	平成13年1月19日(2001.1.19)	(73) 特許権者	000233055
審査請求日	平成16年3月23日(2004.3.23)		日立ソフトウェアエンジニアリング株式会社
			神奈川県横浜市鶴見区末広町一丁目1番43
		(74) 代理人	100100310
			弁理士 井上 学
		(72) 発明者	原 憲宏
			神奈川県川崎市幸区鹿島田890番地 株式会社日立製作所 システム開発本部内

最終頁に続く

(54) 【発明の名称】 データベース処理方法及び実施装置並びにその処理プログラムを記憶した媒体

(57) 【特許請求の範囲】

【請求項1】

データベースに対するユーザからの問合せを解析して対応する実行手順を作成し、当該実行手順にしたがってデータベース処理を行うデータベース処理装置であり、インスタンスの集まりで構成され、個々のインスタンスを添字にて確定することが可能なデータ型を扱うデータベース処理装置において、

問合せ中に前記データ型の列に対する指定を少なくとも二つ含み、かつ前記列に対する条件の指定を含む述語もしくは関数を含む場合、上記述語もしくは関数に指定されている列に対してインデクスが作成されているかどうかを判定し、インデクスが作成されていない場合には、テーブルデータをアクセスし上記述語もしくは関数を評価する様に実行手順を作成する問合せ解析処理部と、

作成した実行手順に従い、テーブルデータを取得し、前記述語に指定された前記データ型の複数列のデータを構成するインスタンスの内、同一添字にて確定されるインスタンスの組みを取得し、そのそれぞれのインスタンスに対し前記述語に指定された条件を用いて評価するテーブルデータ管理部と、
を備えることを特徴とするデータベース処理装置。

【請求項2】

データベースに対するユーザからの問合せを解析して対応する実行手順を作成し、当該実行手順にしたがってデータベース処理を行うデータベース処理装置であり、インスタンスの集まりで構成され、個々のインスタンスを添字にて確定することが可能なデー

10

20

タ型を扱うデータベース処理装置において、

行データの挿入処理、削除処理、または更新処理により挿入、削除、更新される前記データ型の列を少なくとも二つ構成列に持つインデクスに対するメンテナンス処理を行うように実行手順の作成を行う問合せ解析処理部と、

作成した実行手順に従い、行データから上記インデクスを構成する前記データ型の複数列のデータを構成するインスタンスの内、同一添字にて確定されるインスタンスの組みを添字数分取得し、その取得したインスタンスの組みと、そのインスタンスを含む行を識別するための行識別子とで添字数分のインデクスエントリを作成し、その作成されたインデクスエントリを用いて、上記インデクスをメンテナンスするインデクス管理部と、を備えることを特徴とするデータベース処理装置。

10

【請求項3】

データベースに対するユーザからの問合せを解析して対応する実行手順を作成し、当該実行手順にしたがってデータベース処理を行うデータベース処理装置であり、インスタンスの集まりで構成され、個々のインスタンスを添字にて確定することが可能なデータ型を扱うデータベース処理装置としてコンピュータを機能させる為のプログラムを記録した媒体において、

問合せ中に前記データ型の列に対する指定を少なくとも二つ含み、かつ前記列に対する条件の指定を含む述語もしくは関数を含む場合、上記述語もしくは関数に指定されている列に対してインデクスが作成されているかどうかを判定し、インデクスが作成されていない場合には、テーブルデータをアクセスし上記述語もしくは関数を評価する様に実行手順

20

を作成する問合せ解析処理部と、作成した実行手順に従い、テーブルデータを取得し、前記述語に指定された前記データ型の複数列のデータを構成するインスタンスの内、同一添字にて確定されるインスタンスの組みを取得し、そのそれぞれのインスタンスに対し前記述語に指定された条件を用いて評価するテーブルデータ管理部として、コンピュータを機能させる為のプログラムを記録したことを特徴とする媒体。

【請求項4】

データベースに対するユーザからの問合せを解析して対応する実行手順を作成し、当該実行手順にしたがってデータベース処理を行うデータベース処理装置であり、インスタンスの集まりで構成され、個々のインスタンスを添字にて確定することが可能なデータ型を扱うデータベース処理装置としてコンピュータを機能させる為のプログラムを記録した媒体において、

30

行データの挿入処理、削除処理、または更新処理により挿入、削除、更新される前記データ型の列を少なくとも二つ構成列に持つインデクスに対するメンテナンス処理を行うように実行手順の作成を行う問合せ解析処理部と、

作成した実行手順に従い、行データから上記インデクスを構成する前記データ型の複数列のデータを構成するインスタンスの内、同一添字にて確定されるインスタンスの組みを添字数分取得し、その取得したインスタンスの組みと、そのインスタンスを含む行を識別するための行識別子とで添字数分のインデクスエントリを作成し、その作成されたインデクスエントリを用いて、上記インデクスをメンテナンスするインデクス管理部として、

40

【発明の詳細な説明】

【0001】

【発明の属する技術分野】

本発明は、データベース処理方法および装置に適用可能な技術に関するものである。

【0002】

【従来の技術】

現実世界において多値従属となる情報、すなわちある項目に対して複数の情報を扱う場面は多く存在する。リレーショナルデータベースでは、このような情報を正規形データモデルで扱うのが一般的なアプローチである。すなわち、複数テーブルとその間の関連で実現

50

するアプローチである。

【0003】

学生の試験の成績に関する情報を成績表212、学生のEmailアドレスなどに関する情報を学生名簿表211の二つの表に正規化した例を図1に示す。学生名簿表211、成績表212にそれぞれ学籍番号を有し、それらがキーとなって関連が張られている。この場合、キーとなる情報(学籍番号)が重複するため、その分のデータベース容量が必要となる。

【0004】

英語の点数が60以下の成績の学生に英会話教室の案内を電子メールを用いて発送するために対象学生のEmailアドレスを取得したい。その場合、これらの表に対する問合せは次のような複雑なものとなる。また、結合処理を必要とするため処理に時間を要する。

【0005】

```
SELECT 学生名簿.Emailアドレス
FROM 学生名簿,成績
WHERE
```

```
成績.科目='英語' AND
```

```
成績.点数<=60 AND
```

```
学生名簿.学籍番号=成績.学籍番号
```

以上のように正規化データモデルのアプローチでは、社員番号が二つの表に重複するためデータベース容量が大きくなる。また問合せ処理が複雑となり、結合処理を必要とするため処理に時間を要する。

【0006】

一方、非正規化表にて実現する別のアプローチも存在する。先の2つの表を非正規化して実現したのが図1の成績表213である。この場合の問合せ文は以下のように簡単になる。

【0007】

```
SELECT DISTINCT Emailアドレス
FROM 成績
```

```
WHERE 科目='英語' AND 点数<=60
```

しかし、科目、点数以外の列(学籍番号、氏名、学部、Emailアドレス)のデータを冗長に持つことになり、データベースが肥大化する。問合せ文は簡単になるが、非正規化表であるため行の件数も多く、結合処理こそは必要とはしないが、DISTINCTを用いて重複排除しなければならず必ずしも高速な検索とはいかない。

【0008】

データベース問合せ言語SQLでも、情報の集まりを効率よく扱うためオブジェクト指向拡張の機能の一つとしてコレクション型が導入された。また、公知な技術として日立: "XDM E2系 プログラム作成の手引き(XDM/RD E2)", pp27, 1997記載の繰り返し列を用いることにより、複数の情報を配列のように1つの列データとして格納することができる。

【0009】

【発明が解決しようとする課題】

コレクション型も繰り返し列も、列の要素間同士の関連が無いいため、前記のような複数条件を有する問合せを行うアプリケーションを構築しようとする、先に説明した複数の正規化表間の関連によるアプローチもしくは非正規化表によるアプローチのどちらかにならざるを得ない。そして、データベース容量の増大、複雑な問合せ文、検索処理でのパフォーマンスの劣化という問題がある。

【0010】

本発明の目的は上記問題を解決し、比較可能なインスタンスの集まりを管理するデータベースに対するアクセスを好適に行うことが可能な技術を提供することにある。

【0011】

【課題を解決するための手段】

10

20

30

40

50

本発明のデータベース処理システムでは、インスタンスの集まりで構成され、個々のインスタンスを添字にて確定することが可能なデータ型を扱う。本発明のデータベース処理システムでは、前記データ型の列に対する指定を少なくとも二つ含み、かつ前記列に対する条件の指定を含む述語もしくは関数を含む問合せを受け取り解析し、上記述語もしくは関数に指定されている列に対してインデクスが作成されているかどうかを判定する。

【0012】

インデクスが作成されている場合には、上記インデクスをアクセスする様に実行手順を決定する。そしてデータベース処理実行時に、実行手順に従いインデクスをアクセスすることにより、上記述語もしくは関数に指定された条件に合致する同一添字により確定されるインスタンスの組を有するテーブルデータの識別子を取得する。

10

【0013】

また、インデクスが作成されていない場合には、テーブルデータをアクセスする様に実行手順を決定する。そしてデータベース処理実行時に、実行手順に従いテーブルデータをアクセスし、そのテーブルデータの前記述語に指定された前記データ型の列のデータを構成するインスタンスの内、同一添字にて確定されるインスタンスの組みを取得し、その取得した少なくとも二つの前記データ型の列から取得したインスタンスの組みのそれぞれのインスタンスに対し前記述語もしくは関数に指定された条件を用いて評価する。

【0014】

上記インデクスは、インスタンスの集まりで構成され個々のインスタンスを添字にて確定することが可能なデータ型の列を少なくとも二つインデクス構成列に持ち、前記データ型の列のデータを構成するインスタンスの内同一添字にて確定されるインスタンスの組みと、前記データ型の列以外の列データと、行データを識別するための行識別子とから成るインデクスエントリより構成される。そして、テーブルデータの挿入処理、削除処理、更新処理の際、上記インデクスエントリを生成しそれを用いてメンテナンスされる。

20

【0015】

以上の様に本発明のデータベース処理システムによれば、ある同じデータ型のインスタンスの集まりを列のデータとして1つの表に格納管理し、複数のその列の要素を組みとして評価するので、表に関するデータベース容量を最小限に抑えることができ、かつ集合を扱う列に対する検索を簡単な問合せ記述によって高速に行うことが可能である。

【0016】**【発明の実施の形態】**

以下に同じデータ型のインスタンスの集まりをデータとして扱うデータベースに対する処理を高速に行う一実施形態のデータベース処理システムについて説明する。

30

【0017】

まず、本発明の概念を図3を用いて簡単に説明する。

【0018】

本実施形態のデータベース管理システムでは、ある同じデータ型のインスタンスの集まりを列データとして管理する。この同じデータ型のインスタンスの集まりからなるデータ型を以降「繰返し型」、「繰返し型」をデータ型とする列を以降「繰返し列」と呼ぶことにする。また、個々のインスタンスを「繰返し列」の要素と呼ぶことにする。

40

【0019】

図3の成績表300において科目列および点数列が繰返し列として定義されており、成績表のイメージを図示している。繰返し列を有する本成績表のテーブルの定義文の一例を以下に示す。

【0020】

```
CREATE TABLE 成績 (  
  学籍番号      CHAR(8),  
  氏名          MVARCHAR(10),  
  学部          MVARCHAR(10),  
  Emailアドレス VARCHAR(32),
```

50

科目 MVARCHAR(16) ARRAY[32],
 点数 INTEGER ARRAY[32])

ここで、繰返し列はARRAYというキーワードで指定されている。科目列は、定義最大長16バイトのMVARCHAR型データを最大要素数32個まで格納管理可能であることを示している。同様に点数列は、INTEGER型データを最大要素数32まで格納可能であることを示している。

【0021】

繰返し列は、問合せ文において繰返し列名に続く角括弧["および"]の間に指定された値により、繰返し列内に格納管理される要素の一つを確定することを可能とする。例えば、科目列の要素の一つの値で「1」により確定する要素の値を射影する問合せ文は以下のようになる。

【0022】

SELECT 科目[1] FROM 成績

角括弧["および"]の間に指定される数値Nは、科目列内のN番目の要素としての意味を持たせることが多いと考えられる。以降繰返し列名に続く角括弧["および"]の間に指定された値を「添字」と呼ぶ。

【0023】

次のような例を用いて説明する。

【0024】

英語の点数が60以下の成績の学生に英会話教室の案内を電子メールを用いて発送するために対象学生のEmailアドレスを取得したい。その場合の成績表に対する問合せを実現するために以下のような問合せ4をデータベース管理システムに対して行う。

【0025】

SELECT Emailアドレス

FROM 成績

WHERE ARRAY(科目,点数)[ANY](科目='英語' AND 点数<=60)

本実施例では、「英語の点数が60以下」という評価を実現するためにARRAY述語を用いている。ARRAY述語は、「ARRAY」に続く括弧(" , ")内に指定される複数の繰返し列を添字が同じ要素の組みとして複数項目の繰返しと見なし、そのいずれかの要素が"[ANY]"に続く括弧(" , ")内に指定される条件を満たす場合、真を返す述語である。「ARRAY」に続く括弧(" , ")内の指定繰返し列が複数ではなく一つである場合、ARRAY述語は指定繰返し列のいずれかの要素が"[ANY]"に続く括弧(" , ")内に指定される条件を満たす場合に真を返す。

【0026】

よって、ARRAY述語を含めた上記問合せ文は、科目列の要素の値が'英語'であり、かつその要素に対応する点数列の要素の値が60以下であるテーブル行のEmailアドレスの値を取得するための問合せである。

【0027】

上記問い合わせ4をユーザから受け取ったデータベース管理システムは、成績表の各行301~305の科目列、点数列に対してARRAY述語を評価する。その結果、英語の点数が52である要素を有する行302および、英語の点数が59である要素を有する行303を条件を満足する検索結果行として選定し、結果5をユーザに返す。

【0028】

以上により、従来のリレーショナルモデルで複数表の関連にて実現していたインスタンスの集合を、集合を扱える列の値として1つの表に格納でき、データベース容量を最小限に抑えることができる。また、複数の繰返し列の要素を組みとして評価することにより、集合を扱う列に対する検索を簡単な問合せ記述によって実現することができる。その問合せ文は上記1つの表に対するものであり、アクセス領域最小限に抑えられることから、問合せ実行処理も効率的に実現することができる。問合せ実行に関しては、複数の繰返し列の要素の組みをキーとしたインデクスを適用することにより、さらなる高速実行を実現する

10

20

30

40

50

ことができる。

【 0 0 2 9 】

次に図 4 に本実施形態のデータベース管理システムの概略構成を示す。図 4 に示す様に本実施形態のデータベース管理システム 1 は、問合せ解析処理部 1 0、データベース処理実行制御部 2 0、テーブルデータ管理部 3 0、インデクス管理部 4 0、データベース領域アクセス処理部 5 0、定義処理・ディクショナリ管理部 6 0 を有する。

【 0 0 3 0 】

問合せ解析処理部 1 0 は、ユーザあるいはアプリケーションプログラムからの問合せ要求 4 を受け付け解析し、データベースに対するアクセスのための適切な実行手順を生成する。また、問合せ解析処理部 1 0 は、データベースアクセスにより取得した問合せ要求 4 の実行結果データを問合せ元に返す機能も有する。問合せ要求 4 が、データ格納、削除、更新、検索などのデータ操作処理(DML)要求の場合は、処理制御をデータベース処理実行制御部 2 0 に渡す。テーブル定義やインデクス定義などのデータ定義処理(DDL)の場合には、処理制御を定義処理・ディクショナリ管理部 6 0 に渡す。問合せ解析処理部 1 0 は、問合せ要求 4 を解析する際に、問合せ文中にARRAY述語が含まれている場合、そのARRAY述語を評価するためのインデクスが利用可能であるかを決定する処理 1 1 を行い、その結果を実行手順作成処理 1 2 において実行手順に反映する。

10

【 0 0 3 1 】

データベース処理実行制御部 2 0 は、問合せ解析処理部 1 0 により生成された実行手順に従ってテーブルデータ管理部 3 0 やインデクス管理部 4 0 などを用いデータベースアクセスの実行制御を行い、アクセスの結果取得したデータを問合せ要求元に返すために問合せ解析処理部 1 0 に渡す。

20

【 0 0 3 2 】

テーブルデータ管理部 3 0 は、データベース領域 2 に格納されているテーブルデータ 2 1 を管理する。具体的には、テーブルデータ 2 1 に対しデータの格納、削除、更新、検索処理をデータベース領域アクセス処理部 5 0 を経由し行う。テーブルデータ 2 1 には、図 3 を用いて説明した繰返し列を有するテーブルデータも含まれている。テーブルデータ管理部 3 0 は、問合せ文中にARRAY述語が含まれ、その評価にインデクスを使用しないと実行手順により指示されている場合、テーブルの行データに対しARRAY述語評価処理 3 1 を行う機能を有している。

30

【 0 0 3 3 】

インデクス管理部 4 0 は、データベース領域 2 に格納されているインデクス 2 2 を管理する。具体的には、テーブルデータ 2 1 の更新に伴い、それに関連するインデクス 2 2 に対し更新処理をデータベース領域アクセス処理部 5 0 を経由し行うインデクス更新処理 4 2 の機能を有する。また、インデクス 2 2 をアクセスすることにより、ARRAY述語に対して真であるようなテーブル行の検索を高速に行うためのインデクス検索処理 4 1 の機能を有する。

【 0 0 3 4 】

データベース領域アクセス処理部 5 0 は、前述のようにテーブルデータ管理部 3 0 あるいはインデクス管理部 4 0 からの要求に従い、データベース領域 2 に格納されているテーブルデータ 2 1 あるいはインデクス 2 2 に対するアクセスを行う。

40

【 0 0 3 5 】

定義処理・ディクショナリ管理部 6 0 は、ディクショナリ 3 に格納されているスキーマ情報、テーブル情報などの各種定義情報を管理する。ディクショナリ 3 に格納されている定義情報にはデータベース領域 2 に格納されているインデクス 2 2 に関する定義情報であるインデクス 3 1 も含んでいる。定義処理・ディクショナリ管理部 6 0 は、問合せ要求 4 がデータ定義処理(DDL)の場合に、問合せ解析処理部 1 0 の解析結果である定義情報をディクショナリ 3 に登録する。

【 0 0 3 6 】

データベース管理システム 1 を問合せ解析処理部 1 0、データベース処理実行制御部 2 0

50

、テーブルデータ管理部 30、インデクス管理部 40、データベース領域アクセス処理部 50、定義処理・ディクショナリ管理部 60として機能させる為のプログラムは、それぞれ CD-ROM等の記録媒体に記録され磁気ディスク等に格納された後、メモリにロードされて実行されるものとする。なお前期プログラムを記録する媒体は CD-ROM以外の他の媒体でも良い。また、ネットワーク上の別コンピュータシステムを経由してメモリのロードされて実行されるものでも良い。

【0037】

図2は本実施形態のコンピュータシステムのハードウェア構成の一例を示す図である。コンピュータシステム 2000は、CPU 2002、主記憶装置 2001、磁気ディスク装置等の外部記憶装置 2003及び多数の端末 2004で構成される。主記憶装置 2001 10
上には、図4を用いて先に説明したデータベース管理システム1が置かれ、外部記憶装置 2003上にはデータベース管理システム1が管理するテーブルデータ 21とインデクス 22を含むデータベース領域 2が格納される。また、ディクショナリ 3も外部記憶装置 2003上に格納される。さらに、データベース管理システム1を実現するプログラムも外部記憶装置 2003上に格納される。

【0038】

図5はテーブルデータ 21として格納される個々の行データの格納構造の一形態を示す図である。行データは、データレコード 530という形態にて、入出力の単位であるページ 500の中に格納される。ページ 500はデータベース領域 2の構成単位である。1つのページ 500内には、複数のデータレコード 530が格納可能である。データレコード 5 20
30のページ内の格納位置は、スロット 520により指示される。スロット 520の領域には指示するデータレコード 530が格納されているページ 500の先頭からの格納位置が記憶される。ページ制御情報 510は、スロットの割当て状況などのスロット管理およびページ内の使用状況などの領域管理を行うためのものである。データレコード 530は、繰返し列のデータが記憶される繰返し列データフィールド 532を含む。また、データレコード 530は、繰返し列とは別のカラムデータフィールド 533と、データレコード 530内のこれらデータフィールドを管理するためのレコードヘッダ 531を有する。

【0039】

繰返し列データフィールド 532には、繰返し列のデータを構成する要素が配列構造にて格納されている。よって、要素へのアクセスは添字を用いて容易に可能な形態になっている。本形態では、繰返し列データフィールド 532は、データレコード 530内に直接格納されているが、別レコードして実現しデータレコードから論理的にポイントされる形態でもかまわない。また、配列構造である必要はなく、添字により格納位置が確定しさえすればよい。 30

【0040】

図6は本実施形態のインデクス 22の構成例を示す図である。図6では、データベース領域 2に格納されるインデクス 22の構成の一形態を表している。図6に示す様に、インデクス 22はインデクスエントリ 221から成る。

【0041】

インデクスエントリ 221は、インデクスキー 222と、その行データの実体であるデータレコード 530をアクセスするためのレコード識別子 223により構成される。 40

【0042】

インデクスキー 222は、行データ内のインデクスが定義されている繰返し列のデータの内、ある添字により確定される要素の値により生成される。インデクスを複数の繰返し列に対して定義した場合、その複数の繰返し列のデータの内ある添字により確定される要素の値の組みによってインデクスキー 222は生成される。

【0043】

図6の例では、成績表の2つの繰返し列(科目、点数)に対して定義されたインデクスを示している。本例のインデクスキー 222は、科目列の要素の値+点数列の要素の値というデータ構造を採っている。 50

【 0 0 4 4 】

インデックスの定義は、例えば以下のようなSQLに類似した要求文により行われる。

【 0 0 4 5 】

```
CREATE INDEX ARRAY_IDX1 ON 成績 (科目,点数)
```

ここで、ARRAY_IDX1は定義されたインデックスの対して付けたインデックス名称である。

【 0 0 4 6 】

レコード識別子223は、データレコード530が格納されるページ500を一意に識別するためのページ識別子224と、ページ内のデータレコード格納位置を特定するためのスロット520を示すスロット番号225から成る。スロット番号225は、ページ格納構造においてページ制御情報510側から順次番号付けされる。図6では、「ページ識別子+スロット番号」という構造を採っているが、「スロット番号+ページ識別子」でもなんら問題はない。レコード識別子223を用いてデータレコードをアクセスする。具体的には、レコード識別子223を構成するページ識別子224を用いて格納ページをアクセスし、スロット番号225に対応するスロットに記録されているデータレコード格納位置を取得することによってデータレコード530に対するアクセスは行われる。

10

【 0 0 4 7 】

1つの行データの繰返し列は複数の要素を有するため、1行データに対し要素数分のインデックスキー222すなわちインデックスエントリ221が存在する。すなわち、同一レコード識別子223が複数のインデックスエントリ221内に存在することとなる。図6の例では、{Page25,2},{Page30,4}というレコード識別子が2つのインデックスエントリ221に

20

存在している。

【 0 0 4 8 】

インデックスは、複数の繰返し列に対してのみではなく、繰返し型以外の列と繰返し列との組み合わせに対して定義してもかなわない。

【 0 0 4 9 】

図13、図14、図15にインデックスを構成する列のバリエーションを例として示す。

【 0 0 5 0 】

図13は、一つの繰返し列にインデックスを定義した場合のインデックスエントリ221の例を示している。繰返し列C3に対してインデックス定義された表1300の行データ1301(レコード識別子:row13)に対応したインデックスエントリ221を表1300の右側に

30

【 0 0 5 1 】

図14は、二つの繰返し列にインデックスを定義した場合のインデックスエントリ221の例を示している。繰返し列C3およびC4に対してインデックスが定義された表1400の行データ1401(レコード識別子:row14)に対応したインデックスエントリ221を表1400の右側に

40

【 0 0 5 2 】

図15は、二つの繰返し列および一つの繰返し列以外の列にインデックスを定義した場合のインデックスエントリ221の例を示している。繰返し列以外の列C2、繰返し列C3およびC4に対してインデックスが定義された表1500の行データ1501(レコード識別子:row15)に対応したインデックスエントリ221を表1500の右側に

50

【 0 0 5 3 】

本インデクス 2 2 は、ARRAY 述語に指定されている条件に合致する列値を有するデータレコードのレコード識別子を取得するためのものであり、本構成例のようにインデクスエントリ 2 2 1 がテーブル構造のように実装されている必要はなく、図 7 に示すような B 木インデクス等のデータアクセス高速手段として実現することが多い。

【 0 0 5 4 】

図 7 は本実施形態のインデクス 2 2 のデータ構造を B 木インデクスとしての実現した場合の例を示す図である。B 木インデクス 2 2 は、ルートノード 7 0 1、中間ノード 7 0 2、リーフノード 7 0 3 により構成される。

【 0 0 5 5 】

ルートノード 7 0 1 を頂点に木構造を形成している。木構造の末端のリーフノード 7 0 3 に、インデクスエントリ 2 2 1 が格納管理されている。ルートノード 7 0 1 および中間ノード 7 0 2 には下位に位置するノードへのポインタを有するエントリが格納管理される。

【 0 0 5 6 】

ルートノード 7 0 1、中間ノード 7 0 2、リーフノード 7 0 3 は、入出力の単位であるページにより実現されることが多い。図 4 を用いて説明したインデクス検索処理 4 1 において、ARRAY 述語に指定された条件より生成したキーを用いて、ルートノード 7 0 1 より中間ノード 7 0 2 を経由しリーフノード 7 0 3 まで木構造を辿る。ルートページ 7 0 1 および中間ノード 7 0 2 はメモリ常駐することにより I/O 回数を削減することが可能になり高速なアクセスを実現することができる。

【 0 0 5 7 】

図 8 は本実施形態の問合せ解析処理部 1 0 の処理手順を示すフローチャートである。図 8 では問合せ解析処理部 1 0 で問合せ 4 内に ARRAY 述語が出てきた場合の ARRAY 述語対応インデクス使用決定処理 1 1 の処理内容を表している。

【 0 0 5 8 】

ユーザからの問合せ 4 の解析にて、ディクショナリ 3 に登録してあるインデクス定義情報 3 1 を基に、ARRAY 述語評価にインデクスを使用する実行手順を生成するか、インデクスを使用しない実行手順を生成するか決定する。まず、ステップ 8 0 1 において ARRAY 述語部を解析し ARRAY 述語に指定されているカラム名を取得する。次にステップ 8 0 2 にて取得したカラム名に関連したインデクス定義情報 3 1 を取得する。このインデクス定義情報 3 1 の取得処理は、定義処理・ディクショナリ管理部 6 0 に要求して実現する。ステップ 8 0 3 にて関連したインデクス定義情報が取得できた場合、ステップ 8 0 4 に進みそのインデクスを ARRAY 述語評価に使用するよう実行手順を生成することを決定し、ステップ 8 0 6 にて処理を終了する。

【 0 0 5 9 】

ステップ 8 0 3 にて関連したインデクス定義情報が取得できなかった場合、使用可能なインデクスは定義されないと判断し、ステップ 8 0 5 にて ARRAY 述語評価にはインデクスは使用せずデータレコードをアクセスし繰り返し列データを評価するように実行手順を生成することを決定し、ステップ 8 0 6 にて処理を終了する。

【 0 0 6 0 】

図 9 は本実施形態のインデクス管理部 4 0 の処理手順を示すフローチャートである。図 9 では、問合せ文中に ARRAY 述語が含まれ、その評価のために利用可能なインデクスが存在し、それを使用することを実行手順により指示されている場合のインデクス検索処理 4 1 の処理内容を表している。

【 0 0 6 1 】

まず、ステップ 9 0 1 において ARRAY 述語に指定されている条件よりインデクス検索キーの生成を行う。図 3 の説明で使用した問い合わせ文では、科目='英語'および点数=60を用いてインデクス検索キー{'英語',60}を生成する。次にステップ 9 0 1 で生成したインデクス検索キーを用いてインデクスエントリのサーチを行う。ここでは、条件「科目='英語' AND 点数<=60」に合致するインデクスエントリをサーチする。そして、ステップ 9 0 3

10

20

30

40

50

において条件に合致するインデクスエントリ見つかった場合、ステップ904に進みインデクスエントリ内のレコード識別子を取得し、ステップ905にて取得レコード識別子をデータベース処理実行制御部30に対し返却し、ステップ907にてインデクス検索処理を終了する。ここで返却されたレコード識別子は、データベース処理実行制御部30にて、射影処理等のためのデータレコードアクセスに用いられる。

【0062】

ステップ903において条件に合致するインデクスエントリが見つからなかった場合、ステップ908において検索結果なしとデータベース処理実行制御部30に返却し、ステップ907にてインデクス検索処理を終了する。

【0063】

図10および図11は本実施形態のインデクス管理部40の処理手順を示すフローチャートである。図10および図11ではインデクスが定義された繰返し列を含むテーブルデータ21の更新に伴う場合のインデクス更新処理42の処理内容を表している。

【0064】

問合せ解析処理部10で生成した実行手順にて更新対象であると指示されたインデクスに対して更新を行う。まず、ステップ1001において問合せ解析処理部10で生成した実行手順を参照し、テーブルデータ21に施された処理の種別を判定する。判定の結果、テーブルデータに施された処理が挿入処理である場合、ステップ1002に進む。

【0065】

ステップ1002において挿入されたテーブルデータの内更新対象インデクスを構成する列データを取得する。メモリ上に存在する挿入テーブルデータから取得するのが効率的である。繰返し列の場合、要素の一つを挿入データとして取得する。次にステップ1003においてステップ1002で取得したデータを用いてインデクスキーの作成を行う。そして、ステップ1004において先に作成したインデクスキーを用いてインデクス内をサーチしインデクスエントリ更新位置を見つける。ステップ1005においてサーチ対象インデクスキーに対応するインデクスエントリが見つかったかどうか判定し、インデクスエントリが見つかりすでに存在する場合ステップ1006に進む。そしてステップ1006において見つかったインデクスエントリにレコード識別子を追加する。ステップ1005における判定にてインデクスエントリが見つからなかった場合、ステップ1007において新規インデクスエントリを作成し、インデクスに追加する。

【0066】

ステップ1006およびステップ1007において1インデクスエントリの更新処理を終了した後、ステップ1008において追加データがさらに存在するかどうかを判定する。すなわち、繰返し列のさらなる要素が存在するかどうかを判定する。追加データがさらに存在する場合、ステップ1002に戻りインデクスに対しレコード識別子の追加あるいは新規インデクスエントリの追加を前述のように行う。追加データがすでに無い場合、繰返し列の要素数分の更新処理が完了としてステップ1026にてインデクス更新処理を終了する。

【0067】

戻ってステップ1001において判定結果が削除処理である場合、ステップ1009に進む。ステップ1009では削除されたテーブルデータの内更新対象インデクスを構成する列データを取得する。テーブルデータを削除する際、メモリ上にインデクス更新に必要な列データを保持しておく。繰返し列の場合、要素の一つを削除データとして取得する。次にステップ1010においてステップ1009で取得したデータを用いてインデクスキーの作成を行う。そして、ステップ1011において先に作成したインデクスキーを用いてインデクス内をサーチしインデクスエントリ更新位置を見つけ、ステップ1012においてインデクスエントリからレコード識別子を削除する。ここでレコード識別子の削除によってインデクスエントリ無いのレコード識別子の数が0個になった場合、インデクスエントリの削除を行ってもよい。その後ステップ1013において削除データがさらに存在するかどうかを判定する。すなわち、繰返し列のさらなる要素が

10

20

30

40

50

存在するかどうかを判定する。削除データがさらに存在する場合、ステップ1009に戻りインデクスに対しレコード識別子の削除を前述のように行う。削除データがすでに無い場合、繰返し列の要素数分の更新処理が完了したとしてステップ1026にてインデクス更新処理を終了する。

【0068】

戻ってステップ1001において判定結果が更新処理である場合、1014からステップ1015に進む。ステップ1015以降は図11に示してある。ステップ1015では更新されたテーブルデータのインデクスを構成する更新前の列データを取得する。テーブルデータを更新する際、メモリ上にインデクス更新に必要な列データを更新前列データとして保持しておく。繰返し列の場合、要素の一つを更新前データとして取得する。次にステップ1016においてステップ1015で取得したデータを用いてインデクスキーの作成を行う。そして、ステップ1017において先に作成したインデクスキーを用いてインデクス内をサーチしインデクスエントリ更新位置を見つけ、ステップ1018においてインデクスエントリからレコード識別子を削除する。ここでレコード識別子の削除によってインデクスエントリ無いレコード識別子の数が0個になった場合、インデクスエントリの削除を行ってもよい。その後、ステップ1019において、更新されたテーブルデータのインデクスを構成する更新後の列データを取得する。

10

【0069】

メモリ上に存在する更新テーブルデータから取得するのが効率的である。繰返し列の場合、要素の一つを更新後データとして取得する。次にステップ1020において先に取得したデータを用いてインデクスキーの作成を行う。そして、ステップ1021において先に作成したインデクスキーを用いてインデクス内をサーチしインデクスエントリ更新位置を見つける。ステップ1022においてサーチ対象インデクスキーに対応するインデクスエントリが見つかったかどうか判定し、インデクスエントリが見つかりすでに存在する場合ステップ1023に進む。そして、ステップ1023において見つかったインデクスエントリにレコード識別子を追加する。ステップ1022における判定にてインデクスエントリが見つからなかった場合、ステップ1024において新規インデクスエントリを作成し、インデクスに追加する。

20

【0070】

ステップ1023およびステップ1024において1インデクスエントリの更新処理を終了した後、ステップ1025において更新データがさらに存在するかどうかを判定する。すなわち、繰返し列のさらなる要素が存在するかどうかを判定する。更新データがさらに存在する場合、ステップ1015に戻る。更新データがすでに無い場合、繰返し列の要素数分の更新処理が完了したとしてステップ1026にてインデクス更新処理を終了する。

30

【0071】

図12は本実施形態のテーブルデータ管理部30の処理手順を示す処理フローである。図12では問合せ文中にARRAY述語が含まれ、その評価にインデクスを使用しないと実行手順により指示されている場合のテーブルの行データに対するARRAY述語評価処理31の処理内容を表している。

【0072】

インデクス等を用いて取得したARRAY評価対象のレコード識別子を用いてデータレコードをアクセスし、指定列データに対しARRAY述語を評価する。まず、ステップ1201においてインデクス等を用いて絞り込んだ検索結果候補である行のレコード識別子を取得する。そしてレコード識別子内のページ識別子よりデータレコードが格納されているページをアクセスする。さらにレコード識別子内のスロット番号によりデータレコードをアクセスし、ステップ1204において繰返し列データフィールドをアクセスする。そしてステップ1205において繰返し列から要素データを1つ取り出し、ステップ1208において要素データに対しARRAY述語内の条件の判定を行う。ARRAY述語内の条件の判定結果が真の場合、ステップ1208にてARRAY述語評価処理を終了する。その場合、その後射影列データの取り出し処理などを行う。

40

50

【 0 0 7 3 】

戻ってステップ 1 2 0 8 において要素データに対するARRAY述語内の条件の判定結果が偽の場合、さらに次の要素データに対するARRAY述語内の条件の判定を行うためにステップ 1 2 0 7 に進み、次の要素データが存在するか判定する。次の要素データが存在する場合、ステップ 1 2 0 5 に戻る。すでに要素データが無い場合、最終的なARRAY述語の判定結果が偽としてARRAY述語評価処理を終了する。その場合、次のデータレコードに対するARRAY述語評価処理を行う。

【 0 0 7 4 】

以上示したフローチャートの処理は、図 4 で例として示したコンピュータシステムにおけるプログラムとして実行される。しかし、そのプログラムは図 2 の例の様にコンピュータシステムに物理的に直接接続される外部記憶装置に格納されるものと限定はしない。ハードディスク装置、フロッピーディスク装置等のコンピュータで読み書きできる記憶媒体に格納することができる。また、ネットワークを介して図 2 のコンピュータシステムとは別のコンピュータシステムに接続される外部記憶装置に格納することもできる。

10

【 0 0 7 5 】

本実施形態では、繰返し列に対する評価機能をARRAY述語で実現しているが、ARRAY述語である必要はなく他の述語でもよい。さらに、関数であってもかまわない。本実施例では、ARRAY述語を実現する機能において複数の繰返し列の要素の値を組みで扱うようになっているが、必ずしもARRAY述語の機能である必要はない。例えば、表定義文であるCREATE TABLE文におけるオプション指定という実施例も有りうる。

20

【 0 0 7 6 】

以上説明した様に本実施形態のデータベース処理システムによれば、ある同じデータ型のインスタンスの集まりを繰返し列と呼ぶ列のデータとして1つの表に格納管理するので、表に関するデータベース容量を最小限に抑えることができる。また、複数の繰返し列の要素を組みとして評価することにより、集合を扱う列に対する検索を簡単な問合せ記述にて高速に行うことが可能である。

【 0 0 7 7 】

【 発明の効果 】

本発明によれば、比較可能なインスタンスの集まりを繰返し列と呼ぶ列のデータとして管理することにより、データベースへのアクセスを好適に行うことが可能となる。

30

【 図面の簡単な説明 】

【 図 1 】 従来と同じデータ型のインスタンスの集まり扱う場合のテーブルを示す図である。

【 図 2 】 本実施形態のコンピュータシステムのハードウェア構成の一例を示す図である。

【 図 3 】 本発明の概念図である。

【 図 4 】 本実施形態のデータベース処理システムの機能ブロックを示す図である。

【 図 5 】 本実施形態のテーブルデータレコード格納ページ構造の一例を示す図である。

【 図 6 】 本実施形態のインデクスの構成の一例を示す図である。

【 図 7 】 本実施形態のインデクスの構造の一例を示す図である。

【 図 8 】 本実施形態の問合せ解析処理部 1 0 の処理手順を示すフローチャートである。

40

【 図 9 】 本実施形態のインデクス管理部 4 0 のインデクス検索処理の処理手順を示すフローチャートである。

【 図 1 0 】 本実施形態のインデクス管理部 4 0 のインデクス更新処理の処理手順を示すフローチャートである。

【 図 1 1 】 本実施形態のインデクス管理部 4 0 のインデクス更新処理の続きの処理手順を示すフローチャートである。

【 図 1 2 】 本実施形態のテーブルデータ管理部 3 0 の処理手順を示すフローチャートである。

【 図 1 3 】 本実施形態のインデクスキー構成の一例を示す図である。

【 図 1 4 】 本実施形態のインデクスキー構成の一例を示す図である。

50

【図15】本実施形態のインデクスキー構成の一例を示す図である。

【符号の説明】

1 ... データベース管理システム、2 ... データベース領域、3 ... ディクショナリ、4 ... 問合せ、10 ... 問合せ解析処理部、20 ... データベース処理実行制御部、30 ... テーブルデータ管理部、40 ... インデクス管理部、50 ... データベース領域アクセス処理部、60 ... 定義処理・ディクショナリ管理部、21 ... テーブルデータ、22 ... インデクス、31 ... インデクス定義情報、11 ... ARRAY述語対応インデクス使用決定処理、12 ... 実行手順作成処理、31 ... ARRAY述語評価処理、41 ... インデクス検索処理、42 ... インデクス更新処理、222 ... インデクスキー、223 ... レコード識別子、224 ... ページ識別子、225 ... スロット番号、532 ... 繰返し列データフィールド。

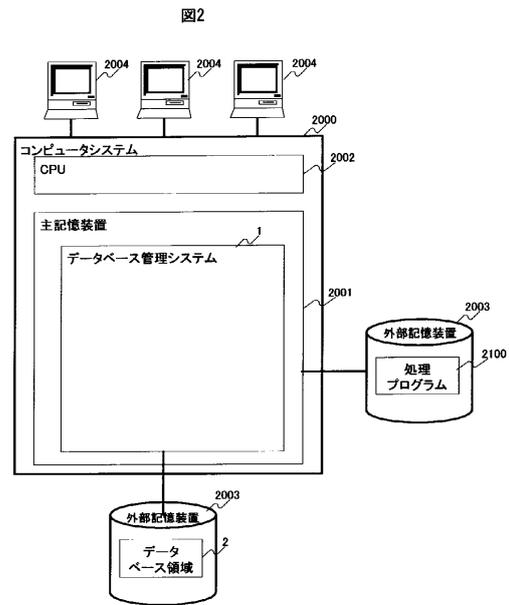
【図1】

図1

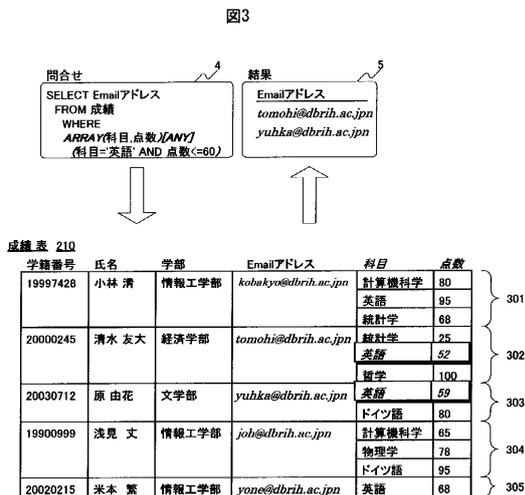
学生名簿表 211				成績表 212		
学籍番号	氏名	学部	Emailアドレス	学籍番号	科目	点数
19997428	小林 清	情報工学部	kobakyo@dbrih.ac.jp	19900999	計算機科学	65
20000245	清水 友大	経済学部	tomohi@dbrih.ac.jp	19997428	計算機科学	80
20030712	原 由花	文学部	yuhka@dbrih.ac.jp	19997428	統計学	68
19900999	浅見 丈	情報工学部	joh@dbrih.ac.jp	20000245	統計学	25
20020215	米本 繁	情報工学部	yone@dbrih.ac.jp	20000245	英語	52
				20020215	英語	68
				20030712	英語	59
				19997428	英語	95
				19900999	物理学	78
				20000245	哲学	100
				20030712	ドイツ語	80
				19900999	ドイツ語	95

成績表 213					
学籍番号	氏名	学部	Emailアドレス	科目	点数
19997428	小林 清	情報工学部	kobakyo@dbrih.ac.jp	計算機科学	80
19997428	小林 清	情報工学部	kobakyo@dbrih.ac.jp	英語	95
19997428	小林 清	情報工学部	kobakyo@dbrih.ac.jp	統計学	68
20000245	清水 友大	経済学部	tomohi@dbrih.ac.jp	統計学	25
20000245	清水 友大	経済学部	tomohi@dbrih.ac.jp	英語	52
20000245	清水 友大	経済学部	tomohi@dbrih.ac.jp	哲学	100
20030712	原 由花	文学部	yuhka@dbrih.ac.jp	英語	59
20030712	原 由花	文学部	yuhka@dbrih.ac.jp	ドイツ語	80
19900999	浅見 丈	情報工学部	joh@dbrih.ac.jp	計算機科学	65
19900999	浅見 丈	情報工学部	joh@dbrih.ac.jp	物理学	78
19900999	浅見 丈	情報工学部	joh@dbrih.ac.jp	ドイツ語	95
20020215	米本 繁	情報工学部	yone@dbrih.ac.jp	英語	68

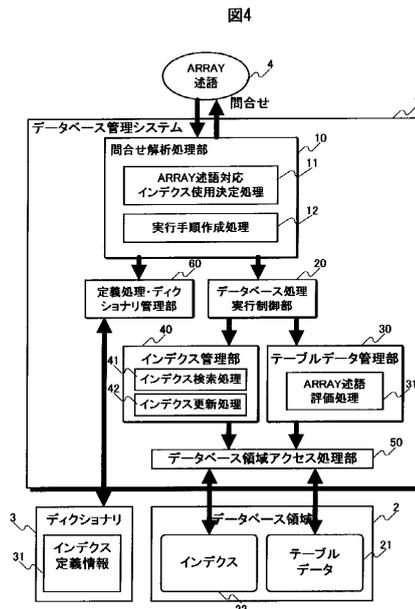
【図2】



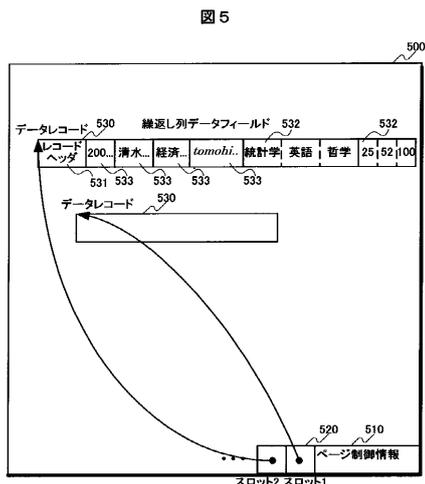
【 図 3 】



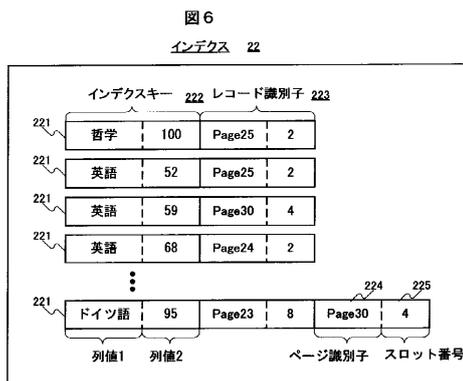
【 図 4 】



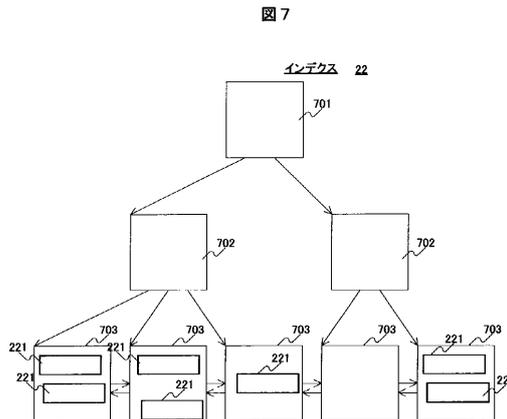
【 図 5 】



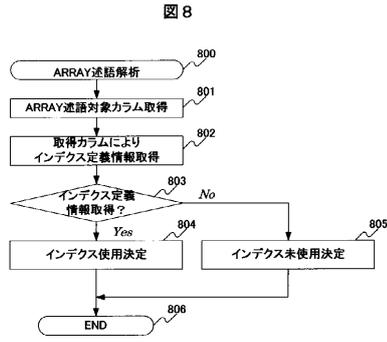
【 図 6 】



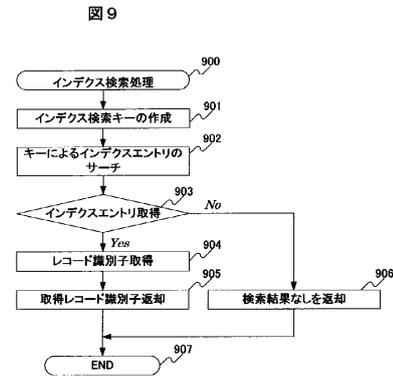
【 図 7 】



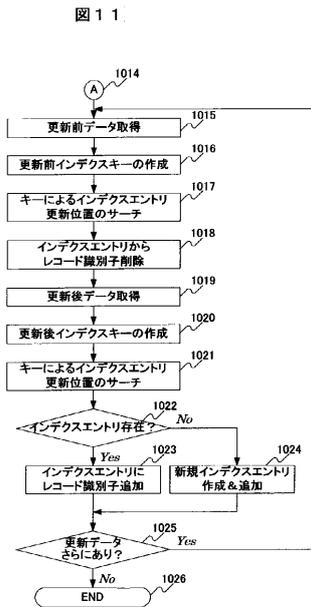
【 図 8 】



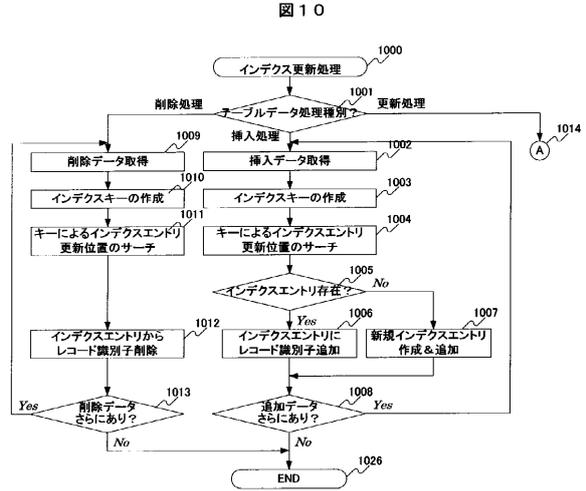
【 図 9 】



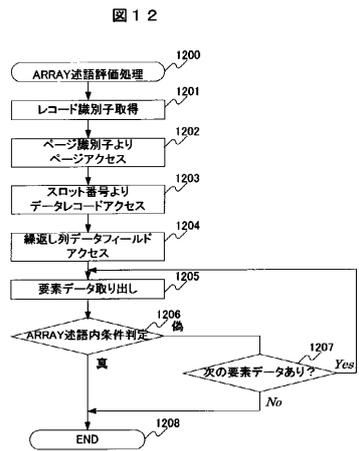
【 図 1 1 】



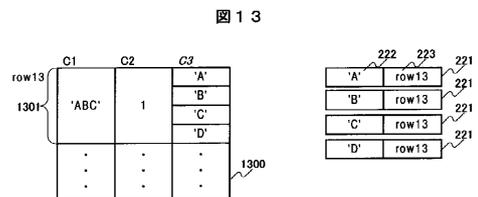
【 図 1 0 】



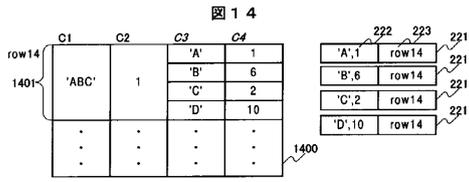
【 図 1 2 】



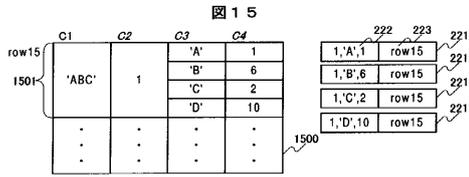
【 図 1 3 】



【 図 1 4 】



【 図 1 5 】



フロントページの続き

- (72)発明者 河村 信男
神奈川県川崎市幸区鹿島田890番地 株式会社日立製作所 システム開発本部内
- (72)発明者 中山 知浩
神奈川県横浜市中区尾上町6丁目81番地 日立ソフトウェアエンジニアリング株式会社内
- (72)発明者 広畠 清美
神奈川県横浜市戸塚区戸塚町5030番地 株式会社日立製作所ソフトウェア事業部内

審査官 池田 聡史

- (56)参考文献 特開平06-324933(JP,A)
特開平07-098711(JP,A)
特開平04-250568(JP,A)
HITAC プログラムプロダクト VOS3 データマネジメントシステム XDM E2系
プログラム作成の手引 - 第2分冊 - (XDM/RD E2), 日本, 株式会社日立製作所
, 1999年 3月31日, 第5版, pp.27

- (58)調査した分野(Int.Cl., DB名)
G06F 17/30
G06F 12/00